

Навигация с применением RANSAC фильтрации точечных особенностей

Машенцев Владимир Юрьевич, Институт автоматики и процессов управления ДВО РАН, Владивосток, Россия,
v.mashentsev@gmail.com

Аннотация

Предложен метод визуальной навигации подводного аппарата по потоку видеоизображений, синхронно захватываемому двумя камерами. Метод основан на использовании оптического потока, прослеживании точечных особенностей среды и обработке соответствующих 3D облаков с применением RANSAC алгоритма. Приведены результаты вычислительных экспериментов на модельных данных.

Ключевые слова: Визуальная навигация, оптический поток, RANSAC, точечные особенности среды, 3D облака.

1. ВВЕДЕНИЕ

Задача восстановления положения подводного аппарата (ПА) на основе характерных особенностей среды, получаемых с изображений с фото или видеокамер, установленных на аппарате, активно исследуется на протяжении длительного времени. На данный момент, предложены алгоритмы с использованием различных техник, относящихся как к обработке исходных данных (оптический поток, характерные особенности среды, выделение объектов и т.д.), так и непосредственно к решению задачи навигации, основанных на оптимизационных методах, робастных схемах семейства «генерация-тестирование», преобразовании Хафа и др.

Извлечение и последующее отслеживание характерных особенностей среды накладывает некоторые ограничения на используемые методы. Одной из причин таких ограничений является наличие ошибок в отслеживании особенностей от кадра к кадру, причём ошибки могут иметь разную природу – ошибка локализации и ошибка измерения. Ошибка измерения (особенность на кадрах, полученных в разные моменты времени, отслежена верно, но некоторые параметры вычислены с ошибкой) позволяет нам использовать такие особенности при решении, однако ошибка локализации (некорректное сопоставление между кадрами – выбросы – outliers) приводит к неверному решению.

Одной из распространённых и успешно применяемых техник, которые позволяют отсеять «плохие» особенности, являются методы, основанные на устойчивых схемах типа «генерация-тестирование». Первой работой, в которой был предложен алгоритм RANSAC (Random Sample Consensus), является работа [1]. Принцип работы данного алгоритма заключается в итеративной выборке минимального подмножества элементов, необходимого для вычисления параметров модели, оценке данных параметров с помощью некоторого критерия и по завершению всех итераций – выборе наилучшего набора параметров и соответствующего им подмножества. Данный метод снискал большую популярность благодаря сочетанию простоты и эффективности, позволяющему даже при большом количестве выбросов давать приемлемые результаты. С тех пор данный метод регулярно используется для решения различных задач, где существует необходимость поиска

«хороших» подмножеств из всего набора входных данных, наиболее соответствующий некоторой модели.

Для решения задачи навигации RANSAC использовался в различных работах. Например, в работе [2] было предложено улучшение работы RANSAC алгоритма путём предварительной генерации фиксированного числа гипотез и выбора наиболее значимых на ограниченном наборе данных. Такой подход позволяет увеличить производительность и может применяться в системах реального времени, однако потенциально может давать худшие результаты по сравнению с классическим подходом в случае, когда отношение «хороших» данных ко всему набору достаточно большое. В работах [3] и [4] использовались более сложные зрительные подсистемы (LIDAR устройства, трифокальные установки) для извлечения особенностей использовались. В работе [5] вводились ограничения на неголономность механической системы, что позволяло уменьшить размер пробных подмножеств до одной точки. Также RANSAC схемы применялись как часть более сложной системы с использованием фильтра Калмана [4].

Несмотря на то, что во многих работах были получены приемлемые результаты, они имеют под собой ряд ограничений, а значит, своей актуальности проблема не потеряла и требует исследования новых подходов в зависимости от практического приложения.

В данной работе предложен алгоритм восстановления положения ПА по последовательности изображений, получаемых со стереопары, и предназначенный для работы в режиме реального времени. В основе алгоритма лежит модифицированная версия метода RANSAC с использованием 3D облаков особенностей. Для извлечения и прослеживания 2D множества особенностей среды используется KLT-трекер.

2. ПОСТАНОВКА ПРОБЛЕМЫ

Ставится задача восстановления траектории ПА с использованием видеоинформации, поступающей с бортовых камер. Одним из возможных подходов к решению данной задачи является применение алгоритмов вычисления оптического потока с использованием характерных особенностей среды (features).

Ключом к решению поставленной задачи является получение преобразования между 3D облаками особенностей в моменты времени t_i и t_j , и использование полученного преобразования для вычисления координат ПА. Для обнаружения и отслеживания особенностей используется алгоритм KLT-трекер, с помощью которого можно получать множество особенностей и отслеживать его на последовательности изображений. 3D координаты особенностей вычисляются с использованием геометрии стереопары. Вопросы, связанные с работой стереопары, в данной работе не рассматриваются.

Итак, ставится задача нахождения значений дискретно заданной функции $F(t_i)$.

$$F(t_i) = \vec{p}_i = M_i \vec{p}_0,$$

где p_i – положение ПА в i -й момент времени, p_0 – положение ПА в начальный момент времени, M_i – преобразование, связывающее положение ПА в момент времени i , и начальное положение.

Положение \vec{p}_0 берётся положением первого кадра. Для получения преобразования в произвольной мировой системе координат (МСК) необходимо знание преобразования из начала координат МСК в положение \vec{p}_0 .

Траектория ПА рассматривается в виде последовательности серий изображений.

$$t_0 < \dots < t_i < t_j < \dots < t_N, \text{ где } i < j.$$

Матрица преобразования M_i получается путем комбинирования преобразований каждой серии. $M_i = M_{10}M_{21}M_{32} \dots M_{ji}$. Каждая из матриц M_{ji} находится способом, предложенным в работе [6] и связывает 3D облака особенностей C_i, C_j .

$$C_i = M_{ji}C_j,$$

Поскольку особенности, отслеживаемые с помощью KLT-трекера зашумляются, то, по сути, мы имеем C'_j – облако особенностей в моменты времени t_j , но вычисленные с ошибкой (особенности на первом кадре каждой серии будем считать идеальными).

$$C_i = M'_{ji}C'_j + E, \text{ где } M'_{ji} = M_{j,i} + \theta, C'_j = C_j + \varphi$$

Таким образом, необходимо найти преобразование M'_{ji} наиболее близкое к идеальному M_{ji} , т.е. с минимальной ошибкой θ .

3. ВЫЧИСЛЕНИЕ ОПТИЧЕСКОГО ПОТОКА

В данной работе для вычисления оптического потока используется KLT-трекер, предложенный в работе [7, 8]. С помощью данного алгоритма производится поиск (извлечение) и прослеживание характерных особенностей среды на изображениях с левых камер стереопары. Прослеживание осуществляется только в пределах серии кадров. В работе исследовались два возможных способа использования KLT-трекера применительно к последовательности изображений. В пределах одной серии трекер можно запускать:

1. на каждом кадре серии (период выборки 1 кадр);
2. выборочно (период выборки 2, 4 и т.д. кадра).

Преимуществом покадрового подхода является минимизация вероятности ошибки локализации, поскольку отслеживание особенностей происходит на малых смещениях. Однако недостатком является время обработки всех кадров серии, которое может превысить реальное время движения ПА.

При выборочном подходе количество запусков трекера снижается и, соответственно, количество ресурсоёмких вычислений снижается, поэтому данный подход более простой для выполнения в реальном времени, однако при больших значениях смещений проекций особенностей на изображениях повышается вероятность появления выбросов и уменьшается количество особенностей в облаках, что будет отрицательно сказываться на дальнейших результатах.

4. ПРЕДЛАГАЕМЫЙ ПОДХОД

Поскольку взаимно-однозначное соответствие особенностей облаков C_i, C'_j известно (результат работы KLT-трекера), то для получения преобразования M'_{ji} с минимальной ошибкой θ необходимо исключить из C'_j особенности-выбросы.

Для решения проблемы поиска «хороших» подмножеств $G_i \subseteq C'_i, G'_j \subseteq C'_j$ в данной работе используется алгоритм RANSAC. Работа данного алгоритма основывается на случайных выборках из исходных множеств, построение модели по выборкам и оценке полученной модели относительно исходных множеств.

Под моделью будем понимать преобразование M'_{ji} . Критерием оценки модели будем считать среднее отклонение особенностей 3D облака G_i и особенностей 3D облака G'_j , преобразованного в соответствии с моделью.

Кратко схему алгоритма можно записать в следующем виде:

1. На вход подаются 3D облака C_i, C'_j .
2. Производится выборка «пробных» подмножеств (облака-гипотезы) соответствующих особенностей X_i, X_j из C_i, C'_j соответственно.
3. Вычисляется модель-гипотеза $H: X_i = HX_j$ методом, описанным в работе [6].
4. Вычисляется оценка соответствия особенностей модели: на основе H вычисляется облако $C'_i = HC'_j$ и проверяется соответствие особенностей C_i, C'_i гипотезе на основе функции оценки $\|c_{i,l} - c'_{i,l}\|_{L_2} < \xi, c_{i,l} \in C_i, c'_{i,l} \in C'_i, l = \overline{1, A}$, где ξ – порог задаваемый пользователем, $A = |C_i - C'_i|$. В результате все особенности помечаются либо «хорошими»: $c_{i,l} \in G_i, c'_{i,l} \in G'_i$, либо выбросами: $c_{i,l} \notin G_i, c'_{i,l} \notin G'_i$.
5. На основе количества особенностей в облаках G_i, G'_i и оценки критерия $\sigma = \frac{1}{B} \sum \|g_{i,l} - g'_{i,l}\|_{L_2}, g_{i,l} \in G_i, g'_{i,l} \in G'_i, l = \overline{1, B}, B = |G_i - G'_i|$ гипотеза помечается лучшей и вычисляется обновленная модель по всем особенностям G_i, G'_j , в противном случае отбрасывается.
6. Повторяем пункты 2-5 k раз, k – количество итераций, задаваемое пользователем.

После выполнения данного алгоритма на выходе получаем наилучшую модель M'_{ji} с минимальной ошибкой θ , полученную за ограниченное время.

5. ПАРАМЕТРЫ RANSAC АЛГОРИТМА

Слабым местом RANSAC подхода в приложениях реального времени, является эвристический поиск параметров количества итераций и количества точек в облаке, на основании которого мы будем строить модель-гипотезу.

При первоначальном анализе проблемы возникают следующие рассуждения: чтобы уменьшить время работы алгоритма необходимо уменьшить общее количество всевозможных выборок, что должно позволить получить оптимальный по времени результат, и нехитрые рассуждения приводят к следующему факту – для того, чтобы уменьшить количество выборок, нужно брать облако минимального размера [1]. Однако даже при минимальном количестве особенностей в облаке, число всевозможных подмножеств достаточно большое, чтобы можно было все их «попробовать» в режиме реального времени. А также немаловажен тот факт, что чем меньше точек в облаке, тем больший вклад вносят их погрешности в модель-гипотезу [9]. С другой стороны при увеличении размера облака повышается вероятность включения точек с большой

погрешностью и их значительный вклад в ошибку модели-гипотезы. Многочисленные эксперименты показали корректность данных предположений. При 10 независимых запусков, размер облака в 3 точки давал неустойчивые результаты вне зависимости от допустимого количества итераций. Однако при увеличении размера облака расхождение результатов всех запусков уменьшалось при допустимом количестве итераций.

На конечный результат также влияет выбор порогового значения ξ . На первый взгляд, для получения лучшей модели, нужно уменьшать порог ξ (в идеальном случае мы можем получить $|\sigma - \varepsilon| = 0$ для любого бесконечно малого $\varepsilon > 0$), однако, такое утверждение неверно и полученные результаты это подтверждают.

Проблема заключается в том, что облако C'_i получено с ошибкой. Распишем функцию оценки (далее не будем указывать норму, будем подразумевать $L2$):

$$\|C_i - C'_i\| = \|C_i - HC'_j\| = \|C_i - HC_j - H\varphi\| < \xi$$

$\Rightarrow H = M_{ji}$, где M_{ji} – идеальная матрица. Перепишем полученную оценку:

$$\|C_i - M_{ji}C_j - M_{ji}\varphi\| = \|M_{ji}\varphi\| < \xi$$

Поскольку шум φ в рамках одного момента времени является постоянным $\varphi = Const \neq 0$, то для выполнения оценки, при изменении порога ξ , нужно изменять M_{ji} , отдаляясь от идеальной, что противоречит исходной задаче.

Малые значения ξ отсеивают большую часть точек, что приводит к уменьшению размера 3D облака и увеличению значения вклада погрешности каждой особенности в модель-гипотезу. Однако большие значений порога ξ допускают наличие особенностей-выбросов в 3D облаке или особенности с большой погрешностью, которые дадут значительный вклад в смещение модели-гипотезы.

6. РЕЗУЛЬТАТЫ

При проведении экспериментов были рассмотрены различные комбинации параметров таких как: размер облаков-гипотез S , порог ξ , период выборки кадров T , количество итераций алгоритма $RANSAC$ k . Каждый набор параметров запускался 10 раз, затем считалось среднее значение для каждого момента времени. Во всех запусках длина серии была принята в 16 кадров. Эксперименты проводились на процессоре Intel Core2Duo E6550 на сцене с «песчаной» текстурой (рис. 3), с перепадами расстояния от центров проекций изображений до поверхности сцены 2-10 метров и размерами изображений 600x400 точек.

В ходе экспериментов в качестве значений ξ брались 0.01, 0.03, 0.05, 0.10. Значения 0.01, 0.03 давали достаточно большой разброс результатов, а количество «хороших» особенностей могло уменьшаться в несколько раз относительно общего количества. При значениях 0.05 и 0.10 результаты получались лучше, однако вследствие того, что при значении 0.10 количество особенностей в «хороших» облаках практически совпадало с общим числом особенностей (попадали особенности с большой ошибкой измерения), результаты при $\xi = 0.05$ м были наилучшими.

Параметры k и S явным образом связаны между собой. В качестве параметра S использовались различные значения: от минимального $S = 3$, до $S = 50$, а параметр k непосредственно отвечает за соответствие времени работы алгоритма в реальном времени. При $S = 3$ работа алгоритма оказывалась неустойчивой даже при максимальных значениях параметра

k , однако, тенденция к уменьшению разброса результатов при увеличении параметра S позволила уменьшать параметр k . Наилучшие результаты были достигнуты при значениях $k = 100, S = 50$.

Для параметра T также пробовались несколько значений 1, 4, 8 (рис. 1). При значении $T = 8$ результаты получались наихудшими, поскольку общее количество точек в облаках уменьшалось, и значительно возрастало количество особенностей с ошибкой локализации. Значения $T = 1$ и $T = 4$ дали интересные результаты: при значении $T = 4$ показатели улучшились по сравнению с $T = 1$. Основными факторами, которые повлияли на поведение алгоритма в данной ситуации, являются характер сцены и движения ПА. При минимальном перемещении особенность может попадать в тот же пиксел изображения камеры, что и на предыдущем шаге, и таким образом давать дополнительную погрешность. При $T = 4$ особенности становятся более «различимы» на изображениях, но при этом расстояние между одной и той же особенностью на кадрах с соседних шагов находится в рамках корректной работы трекара.

	T = 1		T = 4	
	S = 50	Без RANSAC	S = 50	Без RANSAC
Ошибка положения, м.	0.0901	0.18	0.0750	0.1298
Ошибка поворота, м.	0.0991	0.2642	0.0963	0.1381
Время работы (реальное = 128 сек.)	114.7312	94.3912	58.9953	39.1953

Таблица 1: средние величины полученных результатов.

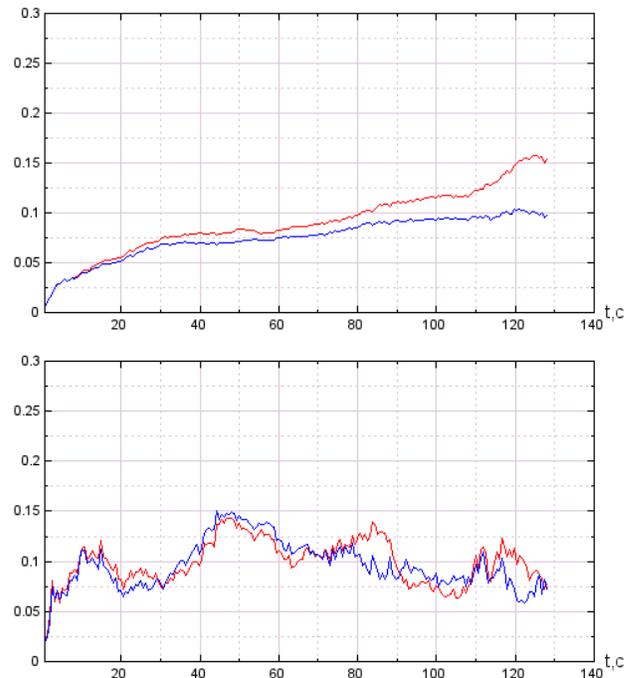


Рис. 1: Ошибка положения ПА в метрах (вверху) и ошибка ориентации в градусах (внизу) с параметрами $S=50, T=1$ (красный график), $T=4$ (синий график) и длиной траектории 128 с.

Эксперименты на более длинной траектории (рис. 2) на той же сцене и с теми же значениями параметров показали, что алгоритм дает отклонение траектории в пределах 20 см. и 0.5 гр. на протяжении 350 сек., что может быть приемлемым для решения задачи навигации в условиях локального маневрирования.

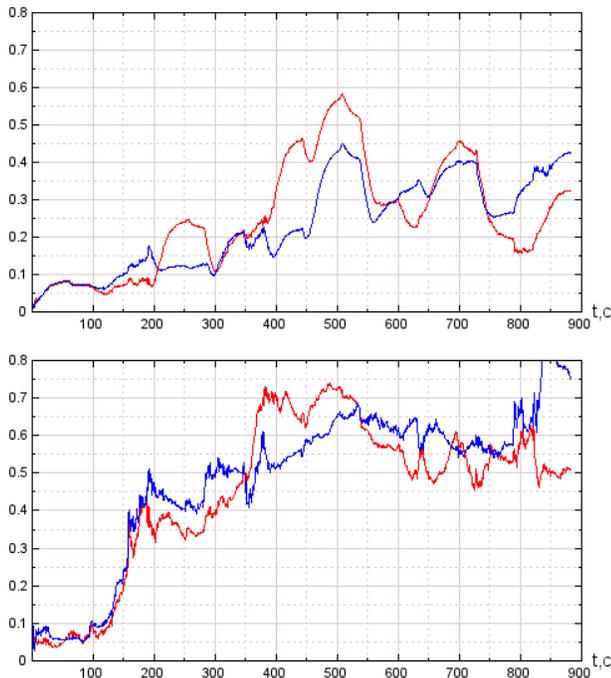


Рис. 2: Ошибка положения ПА в метрах (вверху) и ошибка ориентации в градусах (внизу) с параметрами $S=50$, $T=1$ (красный график), $T=4$ (синий график) и длинной траектории 882 с.

7. ЗАКЛЮЧЕНИЕ

Полученные в вычислительных экспериментах оценки эффективности предложенного метода визуальной навигации показали перспективность его применения для целей локального маневрирования ПА. В дальнейшем планируется повышения вычислительной эффективности метода за счет реализации параллельных вычислений с использованием технологии CUDA, что в полной мере обеспечит его работу в режиме реального времени. Также возможным направлением повышения точности вычисления траектории, является учёт показаний бортовой навигационной системы ПА. Также планируется проведение натурных экспериментов с использованием разработанного программного комплекса.

8. ССЫЛКИ

- [1] M.A.Fischler, R.C.Bolles. Random Sample Consensus: A paradigm for model fitting with applications to image analysis and automated cartography. CACM 24.381-395. 1981.
- [2] D. Nister. Preemptive RANSAC for Live Structure and Motion Estimation, IEEE International Conference on Computer Vision, pp. 199-206, 2003.
- [3] D. Fontanelli, L. Ricciato, and S. Soatto. A fast ransac-based registration algorithm for accurate localization in unknown environments using lidar measurements, in Proc. IEEE Int. Conf. on Automation Science and

Engineering, Scottsdale, AZ, USA, 22-25 September 2007, pp. 597-602.

- [4] B. Kitt, A. Geiger and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme, IEEE Intelligent Vehicles Symposium (IV), pp. 486-492. 2010.
- [5] D. Scaramuzza, F. Fraundorfer and R. Siegwart. Real-time monocular visual odometry for on-road vehicles with 1-point RANSAC, Proc. IEEE Int. Conf. Robot. Autom., pp.4293 -4299 2009.
- [6] K.Berthold, P.Horn. Closed-form solution of absolute orientation using unit quaternions. 1987.
- [7] Carlo Tomasi and Takeo Kanade. Detection and Tracking of Point Features. Carnegie Mellon University Technical Report CMU-CS-91-132, April 1991.
- [8] Jianbo Shi and Carlo Tomasi. Good Features to Track. IEEE Conference on Computer Vision and Pattern Recognition, pages 593-600, 1994.
- [9] O.Chum, J.Matas, J.Kittler. Locally Optimized RANSAC. DAGM 2003.



Рис. 3. Кадр с бортовой камеры с изображением морского дна.