

РОБАСТНЫЙ АЛГОРИТМ ИНЕРЦИАЛЬНОЙ ВИДЕОНАВИГАЦИИ С ИСПОЛЬЗОВАНИЕМ ОТНОСИТЕЛЬНЫХ ОЦЕНОК ПОЛОЖЕНИЯ

Представлен слабосвязанный метод построения интегрированной навигационной системы, обеспечивающий улучшение качества оценок состояния в автономной инерциальной навигации за счет оценки относительного движения с использованием видеоданных. Применена новая методика оценки положения на основе минимизации энтропийно-подобной функции стоимости, по определению робастной к наличию шума и выбросов в визуальном изображении. Приводится экспериментальное подтверждение эффективности этой методики в сравнении с алгоритмом, используемым в настоящее время. В рамках стохастического клонирования для навигации используется непрямой фильтр Калмана. Робастная относительная оценка положения, полученная при использовании новой методики, передается в навигационный фильтр в качестве значения относительного положения. Представлены результаты моделирования, проведено их сравнение с результатами, полученными с помощью классического алгоритма прямого линейного преобразования на основе RANSAC.

Введение

Для инерциальных навигационных систем характерен дрейф, вызванный различными факторами, в частности погрешностями инерциальных датчиков. Для коррекции навигационных систем обычно используются вспомогательные датчики – GPS, барометры или доплеровские лазерные измерители скорости. Реальная альтернатива этим вспомогательным системам – видеосистема, которая оценивает движение камеры, а значит и объекта, обрабатывая поток последовательных изображений. Использование видеоданных в навигации часто называют визуальной одометрией. Основная проблема визуальной одометрии (видеонавигации) в неизвестной обстановке состоит в оценивании местоположения и ориентации объекта. Это является завершающим этапом алгоритма, состоящего из: 1) выявления значимых элементов обстановки по изображениям с камеры, 2) отслеживания элементов в последовательных кадрах, 3) оценивания положения. Методики, используемые для оценки положения камеры, различаются в основном способами реализации трех вышеперечисленных этапов. Наиболее сложным вопросом в решении задачи оценки положения является наличие шума и выбросов данных. Выбросы возникают в основном из-за неточного согласования ключевых точек и/или их отслеживания в левом и правом изображениях (при стереозрении) и между последовательными моментами времени. Задача отбраковки выбросов часто решается методами минимизации (L_2, L_∞ и т.д.) [1] или итерационного уточнения [2–4] при предварительной или постобработке изображений. Известные робастные алгоритмы типа RANSAC [2, 3], применяющиеся при оценке положения камеры, не гарантируют оптимально-

Ди Корато Франческо. Аспирант, старший лаборант Университета Пизы (г. Пиза, Италия).

Инноченти Марио. Профессор Университета Пизы.

Поллини Лоренцо. Доцент (assistant professor) Университета Пизы.

Научный редактор перевода д.т.н. А.В.Небылов.

сти. Почти во всех схемах, предложенных в литературе, отбраковка выбросов выполняется на этапе предварительной или постобработки в основном за счет минимизации квадрата нормы ошибки оценивания. В последнее время появились интегрированные видеоинерциальные навигационные системы, которые различаются главным образом подходом к выбору связи между видеосистемой и инерциальной навигационной системой. В работах [5, 6] используется сильно-связанный подход, т. е. каждая накопленная ключевая точка добавляется в навигационный фильтр состояний и участвует в процессе оценивания. В слабо-связанных системах [7, 8], напротив, в навигационный фильтр поступают только значения относительного местоположения. В [7] представлен метод *стохастического клонирования* (stochastic cloning), а оценка относительного положения вычисляется путем минимизации по классическому методу наименьших квадратов. Аналогичный подход использован в [8], однако в фильтр поступают измерения положения, полученные с помощью робастного подхода с применением минимизации второй нормы, при которой используется функция Хубера [9] в рамках М-оценки. В работе [10] в качестве основного навигационного датчика используется система стереозрения, а обработанные измерения инерциального измерительного модуля (ИИМ) подаются для коррекции ориентации в расширенный фильтр Калмана.

В данной статье предлагается слабосвязанная система, включающая систему стереозрения и ИИМ. Относительные оценки положения, полученные от видеосистемы, рассчитываются с использованием энтропийно-подобной (*Entropy-Like*) функции стоимости, которая по определению робастна к выбросам данных. Таким образом, оценки положения подаются в качестве значений относительного положения в непрямой навигационный фильтр Калмана в рамках *стохастического клонирования*, как описано в [7].

Основная идея данной статьи состоит в том, чтобы показать, что использование предлагаемого робастного алгоритма оценки положения, устойчивого к широкому классу возмущений, обеспечивает повышение точности навигации, а также, при более полном использовании специфических характеристик алгоритма, создает возможности для дальнейшего ее повышения.

1. Математическое обоснование

В данном разделе дается математическое обоснование и вводится терминология, используемая в работе как для инерциальной навигации, так и для видеооценки положения. Мы использовали измерения бесплатформенной инерциальной системы и косвенный (error-state), или, как часто его называют, навигационный фильтр Калмана. В качестве ИИМ использован блок с волоконно-оптическими гироскопами и МЭМС-акселерометрами. Точность этого типа ИИМ обычно достаточна для оценки ориентации, но не для навигации [10], [15]: большие дрейфы оценок местоположения возникают вследствие неточностей акселерометров и изменяющихся во времени уходов нулевых сигналов. Обычно ИИМ дополняются приемниками сигналов GPS или другими внешними датчиками для обеспечения коррекций ошибок местоположения и оценки уходов нулей акселерометров. В данной статье предлагается использовать видеосистему вместо спутниковой навигационной системы GPS или когда её сигналы недоступны. Ориентация оценивается только с помощью инерциальных измерений, а скорость и местоположение – с помощью данных от видеосистемы.

1.1. Инерциальная навигация

Алгоритм инерциальной системы использует следующие переменные: матрица направляющих косинусов C_b^n поворота из корпусной (b) в навигационную (n) систему координат (СК), вектор скорости $V^n = [V_{North} \ V_{East} \ V_{Down}]^T$ (в навигационной СК) и вектор местоположения $r^e = [\phi \ \lambda \ h]^T$, включающий широту, долготу и высоту навигационной СК относительно СК, связанной с Землей (e). Можно использовать любые навигационные СК и СК, связанные с Землей, в данной работе мы использовали СК с направлением осей «Север, Восток, Вниз» (NED) и геоцентрическую СК, связанную с Землей (ECEF).

При таких допущениях инерциальные уравнения можно разделить на две части: уравнение ориентации и уравнение местоположения и скорости:

$$\begin{cases} \dot{C}_b^n = C_b^n \left[(\omega_{ib}^b - C_b^n \omega_{in}^n) \wedge \right], \\ \dot{x} = f(x, C_b^n, f_{ib}^b), \end{cases} \quad (1)$$

где $x = [V^n, r^e]$, f_{ib}^b и ω_{ib}^b – измерения акселерометров и гироскопов, а ω_{in}^n – скорость объекта. Полное описание вывода уравнений можно найти, например, в работе [11], здесь оно не приводится.

В этой статье используется навигационный фильтр Калмана, и основным источником погрешностей для рассматриваемого класса ИИМ служат акселерометры. В соответствии с методикой [11], сначала были определены следующие случайные ошибки: ошибка скорости δV^n , ошибка местоположения δr^e и ошибка акселерометра (или смещения нуля) δf_{ib}^b , которые задают соотношение между реальными значениями, их оценками (обозначенными знаком $\wedge$) или измеренными значениями (с тильдой):

$$\begin{cases} \hat{V}^n = V^n + \delta V^n, \\ \hat{r}^e = r^e + \delta r^e, \\ \tilde{f}_{ib}^b = f_{ib}^b + \delta f_{ib}^b + v_f, \end{cases} \quad (2)$$

где v_f – гауссовский процесс с нулевым средним значением и дисперсиями $E[v_f v_f^T] = Q_f$, представляющий шум акселерометра. Динамика случайной ошибки сформулирована следующим образом:

$$\delta \dot{x}(t) = \begin{bmatrix} \delta \dot{V}^n \\ \delta \dot{r}^e \\ \delta \dot{f}_{ib}^b \end{bmatrix} = \begin{bmatrix} F_{VV} & F_{Vr} & C_b^n \\ F_{rV} & F_{rr} & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta V^n \\ \delta r^e \\ \delta f_{ib}^b \end{bmatrix} + \begin{bmatrix} C_b^n & 0 \\ 0 & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} v_f \\ v_{rf} \end{bmatrix} = F(t) \delta x(t) + G(t) u_c(t), \quad (3)$$

где изменяемые во времени компоненты матрицы F являются определителями Якоби динамики ошибки, а динамика смещения нуля акселерометров моделирована как броуновское движение с фиктивной динамикой: $\delta \dot{f}_{ib}^b = v_{rf}$ с $E[v_{rf} v_{rf}^T] = Q_{rf}$.

В уравнении (3) представлена аппроксимация нелинейной динамики ошибки линейной системой с переменными параметрами; линейное выражение хорошо подходит для использования в расширенном фильтре Калмана.

Далее при рассмотрении дискретных калмановских фильтров будем использовать выражение динамики (3) с дискретизацией по времени:

$$\delta x_{k+1} = \Psi_k \delta x_k + \Gamma_k u_{c,k}, \quad (4)$$

Матрицы в системе уравнений (4) можно получить несколькими способами; мы использовали метод интегрирования Эйлера с интервалом ΔT и получили $\Psi_k = (I - F(t_k) \Delta T)$ и $\Gamma_k = G(t_k) \Delta T$.

1.2. Оценка положения с помощью системы стереозрения

В задаче видеоинерциальной навигации с использованием относительных измерений требуется оценивать движение камеры между последовательными кадрами. В системе стереозрения каждая пара изображений (левое и правое) создает облако из N трехмерных ключевых точек $\{P_{i,k}\}$, полученных путем выявления, согласования и триангуляции двух наборов (левого и правого изображений) N двумерных элементов $\{p_{i,k}\}$.

Отслеживание двумерных элементов в последовательные моменты времени t_k и t_{k+h} позволяет получить два облака трехмерных ключевых точек $P_{i,k}$ и $P_{i,k+h}$, связанных соотношениями перемещения. Эти соотношения включают движение камеры, сдвиг (поступательное перемещение) T_k^{k+h} и поворот C_k^{k+h} между t_k и t_{k+h} , т.е.:

$$P_{i,k+h} = C_k^{k+h} P_{i,k} + T_k^{k+h} = g_k^{k+h} P_{i,k}, \quad (5)$$

где $g_k^{k+h} = \{C_k^{k+h}, T_k^{k+h}\} \in SE(3)$ – преобразование положения камеры в момент t_k в положение в момент t_{k+h} , $g_k^{k+h} P_{i,k}$ – поворот и сдвиг (C_k^{k+h} и T_k^{k+h}) точки $P_{i,k}$. При трехмерной параметризации матрицы поворота матрицу преобразования в уравнении (5) можно записать: $g_k^{k+h}(\theta_p)$, $\theta_p \in \Theta_p \subset \mathbb{R}^6$, где Θ_p – набор всех возможных параметров (угловые смещения и сдвиги). Задача оценки положения сводится к оцениванию вектора параметров неизвестного движения θ_p по двум облакам N элементов в моменты t_k и t_{k+h} . Задача решается путем использования минимизации невязок оценок

$$E_{i,k+h} = P_{i,k+h} - g_k^{k+h}(\theta_p) P_{i,k}, \quad (6)$$

т.е.

$$\hat{\theta}_p = \arg \min_{\theta_p} \sum_{i=1}^N \mathcal{L}\{P_{i,k+h} - g_k^{k+h}(\theta_p) P_{i,k}\}, \quad (7)$$

где $\mathcal{L}\{\cdot\}$ – функция стоимости невязки оценки (6).

Из-за наличия погрешностей триангуляции и калибровки требуется отслеживать как минимум $N > 3$ нейтральных точек вдоль траектории движения камеры. Выбор элементов и отслеживание во время внешних экспериментов, описанных в статье, проводились с помощью стереозрения и алгоритма масштабно-инвариантного преобразования элементов (Scale Invariant Feature Transform

(SIFT)) [12], традиционного для таких приложений. Алгоритм начинается с получения пары стереоизображений и вырабатывает трехмерные точки с учетом различий элементов методом триангуляции. Алгоритм SIFT применяется для выявления, выбора и согласования элементов из левого и правого изображений. Стереосогласование ключевых точек левого и правого изображений выполняется путем сравнения квадрата расстояния между дескрипторами каждой точки в двух изображениях и выбора пары, где это расстояние минимально. Для триангуляции и отслеживания используются только те точки, которые есть и в левом, и в правом изображениях. Такой же метод позволяет отслеживать ключевые точки в текущем и прошедшем изображениях, что обеспечивает возможность выбора точек $P_{i,k}$ и $P_{i,k+h}$.

2. Робастное оценивание положения камеры

Желательно, чтобы детектор элементов мог точно отслеживать искаженные участки изображений, если они деформируются по мере изменения точки обзора. Поэтому ковариантные детекторы элементов, например SIFT, *исключают* эффект преобразований определенной группы. В алгоритме SIFT элементы *канонизируются* относительно сдвига, поворота и масштаба, иными словами, эффект этих преобразований компенсируется и элементы становятся *инвариантными* к ним. Процесс канонизации приводит к некоторой потере данных в обнаруженных элементах, что может вызвать неопределенность. На рис. 1 представлен пример, когда потеря данных приводит к рассогласованию: два выделенных участка изображения выглядят практически одинаково, хотя находятся далеко друг от друга; единственное различие состоит в том, что один является повернутой версией другого. После прохождения процесса канонизации, связанного с извлечением инвариантных элементов, детектор может посчитать, что они соответствуют одному участку изображения, что приведет к рассогласованию. В результате целый набор элементов, собранных на этапах сбора, согласования и отслеживания, может подвергнуться влиянию выбросов данных.



Рис. 1. Пример возможной неопределенности согласования, связанной с использованием ковариантных детекторов элементов (например, SIFT)

Для целей данной работы будет удобно пересмотреть определение выбросов в части эффекта, производимого ими, на описание движения. При рассогласовании ключевых точек между левым и правым кадром (при использовании стереозрения), некорректное отслеживание ключевых точек в последовательные моменты времени, вызванное наличием элементов, принадлежащих движущимся объектам, приводит к образованию точек, которые движутся в плоскости изображения иначе, чем *правильные точки*. Поэтому есть необходимость в робастной методике, гарантирующей определенную степень *нечувствительности* к набору возможных отклонений от номинальной модели. Совершенно очевидно, что наблюдения будут иметь разброс относительно номинальной модели, если при обработке будет использовано некоторое количество выбросов. Далее

для разработки робастного алгоритма оценки положения будет использована методика измерения дисперсии разброса данных.

2.1. Алгоритм оценивания с использованием энтропийно-подобной штрафной функции

Недавно в качестве альтернативы для оценивания по методу наименьших квадратов был предложен робастный метод [13]. Целью такого алгоритма оценивания является представление дисперсии невязок, он построен на концепции энтропии Гиббса [14]. акой оценщик получил название алгоритма оценивания с использованием энтропийно-подобной штрафной функции (*Least-Entropy Like (LEL)*).

Имея N наблюдений $\tilde{y}_i$, $i = 1, \dots, N$ общей нелинейной системы, параметризуемых по вектору θ :

$$\tilde{y}_i = f(u, \theta) + \epsilon_i, \quad i = 1, 2, \dots, N, \quad (8)$$

алгоритм LEL можно построить по *квадратам относительных невязок* π_i , определенных как:

$$\pi_i = \frac{r_i^2}{\sum_{j=1}^N r_j^2}, \quad (9)$$

$$r_i = \tilde{y}_i - \hat{y}_i$$

где u – известный входной вектор системы, ϵ_i – нормально распределенная невязка, $\hat{y}_i = f(u, \hat{\theta})$ – прогноз выходных данных системы при оценке $\hat{\theta}$ вектора неизвестного параметра, и

$$\pi_i \in [0, 1], \quad \sum_{i=1}^N \pi_i = 1 \quad (10)$$

Задаваясь уравнением (9), нормализованную энтропийно-подобную штрафную функцию стоимости, можно определить следующим образом:

$$H = \begin{cases} 0 & \text{если } \sum_{j=1}^N r_j^2 = 0 \\ -\frac{1}{\log N} \sum_{i=1}^N \pi_i \log \pi_i & \text{иначе} \end{cases}, \quad (11)$$

где условие $\sum_{j=1}^N r_j^2 = 0$ вводится для математической полноты и представляет наилучшую аппроксимацию (в этой ситуации оптимизация на основе второй нормы обеспечивает точное согласование).

В работе [13] предлагается следующий алгоритм оценивания:

$$\hat{\theta}_{LEL} = \arg \min_{\theta} H. \quad (12)$$

Суть этого алгоритма состоит в том, чтобы распределить квадраты относительных невязок максимально неравномерно. В таком случае большинство невязок *малы* (относительно функции $\sum_{j=1}^N r_j^2$), а несколько *велики* [13]. Точки данных, соответствующие этим большим невязкам, являются потенциальными выбросами.

2.2. Робастное оценивание положения с помощью алгоритма LEL

Когда невязки r_i в уравнении (9) определяются с помощью ошибки оценки положения, как в уравнении (6), задача робастной оценки положения может быть решена с помощью алгоритма LEL:

$$\hat{\theta}_{LEL} = \arg \min_{\theta} \left(-\frac{1}{\log N} \sum_{i=1}^N \pi_i \log \pi_i \right), \quad (13)$$

где

$$\pi_i = \frac{\psi(r_i)}{\sum_{j=1}^N \psi(r_j)}, \quad (14)$$

$$r_i = p_{i,k+h} - K_c \text{proj} \left\{ g_k^{k+h}(\theta_p) P_{i,k} \right\}. \quad (15)$$

Здесь $\psi(\cdot)$ – функция, подобная функции Хубера [9]:

$$\psi(a) = \begin{cases} a^2, & \text{если } a \leq d \\ d^2, & \text{иначе} \end{cases}, \quad (16)$$

а K_c и $\text{proj}\{\cdot\}$ – матрица калибровки модели камеры с точечной диафрагмой и оператор проецирования перспективы, обычно принятый в системах компьютерного зрения.

Такая формулировка алгоритма *Least-Entropy Like* для решения задачи оценивания положения является улучшенной версией по отношению к версии, предложенной авторами в [15] и [16]. В этих работах относительные невязки π_i в (14) были определены с использованием стандартной трехмерной формы невязок, как в (6). Триангулируемые элементы обычно имеют неопределенную глубину вследствие собственного шума изображения. Эта неопределенность растет пропорционально относительному расстоянию от камеры до точек элемента и сильно снижает точность оценивания. Один из способов улучшения результатов оценивания с сохранением информации о масштабе (по схемам оценки положения между трехмерными точками) – использовать схему ошибки перепроецирования, так как она включает только координаты изображения, инвариантные к изменениям глубины [17]. Поэтому в данной работе невязку оценки положения преобразуем в ошибку перепроецирования, что повышает точность оценивания. Применение функции, подобной функции Хубера, позволяет избежать плохой обусловленности энтропийно-подобной функции стоимости за счет ограничения верхней границы знаменателя (14) и, таким образом, предотвращения неконтролируемого роста суммы невязок вследствие численной чувствительности алгоритма.

До настоящего момента не проводилась параметризация гомогенной матрицы $g_k^{k+h}(\theta_p)$. Параметризовать можно только матрицу поворота C_k^{k+h} , используя для этого разные способы (углы Эйлера, единичные повороты, кватернионы и т.д.). Желательно, чтобы параметризация в процессе нелинейной оптимизации не приводила к увеличению числовой чувствительности [18], уже имеющейся в

задаче. В данной статье (см также [16]) использовалось решение, предложенное в [19]: параметризуется нормированный кватернион $q \in \mathbb{Q}$ с трехмерными векторами типа $v \in \mathbb{V}$, $q = B(v)$, $B(\cdot): \mathbb{V} \rightarrow \mathbb{Q}$ – нелинейное преобразование, как определено в [19]. Применяя такой подход, кватернион можно описать с помощью сокращенного набора параметров – компонентов трехмерного вектора $v = [v_1 \ v_2 \ v_3]^T$. Хотя этот алгоритм параметризации не лишен особенностей, в [19] показано, что ее можно успешно применить в задачах минимизации, где изменения кватерниона на каждом такте алгоритма невелики. Подход также позволяет изменять вектор v свободно и не налагая ограничения единичной нормы (которое всегда выполняется соответствующим кватернионом $q = B(v)$). Таким образом, появляется возможность выполнять безусловную минимизацию.

2.3. Алгоритм оптимизации

Энтропийно-подобная штрафная функция нелинейна, и может существовать множество локальных минимумов. Поэтому при расчете минимизации следует уделять особое внимание начальным условиям. Нелинейная оптимизация обычно выполняется после этапа предварительной (обычно линейной) оптимизации путем нахождения решений, близких к оптимальным значениям. Затем эта приближенная оценка служит начальным условием для нелинейного уточнения. В соответствии с задачами данной работы цель состоит в получении приемлемой локальной оценки движения по инерциальным измерениям, выполняемым за короткий промежуток времени (между двумя последовательными кадрами). Это осуществимо, так как в течение такого промежутка измерения ускорения достаточно хорошо описывают реальное движение системы. Поэтому можно начать нелинейное оценивание с параметра $\hat{\theta}_{p,0}$, извлекаемого с помощью наилучшей доступной оценки относительного преобразования

$$\hat{g}_k^{k+h} = (\hat{g}_{k+h}^-)^{-1} \hat{g}_k^+ \rightarrow \hat{\theta}_{p,0}, \quad (17)$$

где $\hat{g}_k^+$ – наилучшая оценка навигационного параметра (ранее рассчитанная фильтром) в момент t_k , $\hat{g}_{k+h}^-$ – прогноз навигационного параметра в текущий момент t_{k+h} . Символ $\rightarrow$ в уравнении (17) означает, что значение $\theta_{p,0}$ извлекается путем преобразования $\hat{g}_k^{k+h}$.

В литературе по минимизации в аналогичных задачах часто полагается, что форма штрафной функции стоимости вблизи локального минимума может считаться квазипараболической; т.о., оптимизацию можно производить локально методом Левенберга – Маркара. Затем вектор оценки $\theta_{p,k} \in \mathbb{R}^6$ итеративно корректируется следующим аппроксимирующим уравнением:

$$\hat{\theta}_{p,k} = \hat{\theta}_{p,k-1} - \gamma_k \left(\nabla_{\theta_p} H^T \nabla_{\theta_p} H + \mu_k \text{diag} \{ \nabla_{\theta_p} H^T \nabla_{\theta_p} H \} \right)^{-1} \nabla_{\theta_p} H^T. \quad (18)$$

Откорректированное уравнение (18) влияет на решение задачи оптимизации в направлении уменьшения градиента $\nabla_{\theta_p} H \in \mathbb{R}^{1 \times 6}$ в зависимости от значения

аппроксимированной матрицы Гесса $\nabla_{\theta_p}^2 H \approx \nabla_{\theta_p} H^T \nabla_{\theta_p} H \in \mathbb{R}^{6 \times 6}$. Градиент энтропийно-подобной функции можно рассчитать численным или аналитическим методом; в данном случае мы выбрали первый вариант и получили приемлемые результаты. Затем алгоритм применяется, пока вектор оценки $\hat{\theta}_{p,k}$ не достигнет стационарной точки с допуском ϵ , так что $\|\hat{\theta}_{p,k} - \hat{\theta}_{p,k-1}\| < \epsilon$.

3. Экспериментальные результаты оценивания положения

В работе [16] авторы показали, что предлагаемый метод работает лучше, чем стандартный метод на основе второй нормы. В этом разделе предлагаемый алгоритм сравнивается с алгоритмом прямого линейного преобразования (Direct Linear Transformation, DLT) на основе RANSAC [20] с нелинейным уточнением путем блочного уравнивания (Bundle adjustment) [21]. Для сравнения проводится несколько экспериментов по автономному оцениванию положения (без комплексования с ИИМ). Алгоритм RANSAC, используемый в DLT, построен на основе *всех* имеющихся сведений о шуме изображения, чтобы получить наилучший результат для алгоритма.

Для натурных экспериментов мы использовали ручную стереокамеру Firewire; было выбрано разрешение 516×388 пикселей, так как оно обеспечивало оптимальное соотношение скорости обработки изображений с точностью выбора и согласования элементов. Для сбора тестовых видеоданных использовался компьютер с тактовой частотой 1,6 ГГц и ОЗУ 1 Гб; затем видеокadres обрабатывались автономно с помощью алгоритма оценивания.

3.1. Инициализация алгоритма без использования инерциальных данных

При автономном оценивании положения камеры (без применения инерциальных данных) не было возможности произвести оценку локального движения. В таких условиях инициализация алгоритма производится только по измерениям изображений. Для этой цели хорошо подходит восьмиточечный алгоритм [20], включающий десять этапов алгоритма RANSAC, который и был использован для получения приближенных оценок положения. В алгоритме применяются только измерения на пространстве монокулярного изображения, таким образом, возможно восстановить сдвиг до масштабного коэффициента (благодаря перспективной проекции). Мы выбрали этот алгоритм, так как он обеспечивает быстрое аналитическое решение задачи, а выработанную оценку положения можно считать достаточно близкой к оптимальной. В частности, такая процедура инициализации позволяет с большой точностью восстановить матрицу поворота. Мы выяснили, что, задавая один параметр между сдвигом или поворотом, по направлению к его оптимальному значению, можно добиться, чтобы минимум энтропийно-подобной штрафной функции стоимости относительно других был вблизи ожидаемого оптимального значения. Метод оказался очень полезным, ибо позволил получить данные о повороте с достаточно высокой точностью. Единственный недостаток заключается в неопределенности масштаба, которая влияет на результаты оценивания восьмиточечным алгоритмом. Чтобы преодолеть эту неопределенность и подстроить реальный масштаб преобразования, использовалось расстояние между двумя случайно выбранными точками

в пространстве, расстояние между которыми по глубине известно за счет триангуляции отслеживаемых элементов.

Полученная начальная оценка положения использовалась для инициализации нелинейного уточнения путем минимизации энтропийно-подобной функции. Минимизация проводилась при использовании всего набора собранных данных.

3.2. Результаты эксперимента

На приведенных ниже рисунках показаны результаты нескольких экспериментов. На рис. 3 и 6 представлена отсортированная вторая норма невязок перепроецирования:

$$\|e_{r,i}\|^2 = \left\| p_{i,2} - K_c \text{proj} \left\{ g_1^2 \left(\hat{\theta}_p^{LEL, DLT} \right) P_{i,1} \right\} \right\|^2,$$

полученная при использовании параметров движения $\hat{\theta}_p$, оцененных двумя методами – LEL и робастным методом DLT. Матрица калибровки камеры K_c определялась экспериментально. $P_{i,1}$ – трехмерные ключевые точки, триангулируемые в первом положении камеры (в момент t_1); $p_{i,2}$ – координаты пространства изображения (в пикселях) соответствующих элементов на плоскости изображения, полученного в конечном положении камеры (в момент t_2). На рис. 2 и 5 показаны повторные проекции трансформированных точек $g_1^2 \left(\hat{\theta}_p^{LEL} \right) P_{i,1}$ на плоскости второго изображения при оценивании методом LEL, т.е. значения

$$\hat{p}_{i,2} = K_c \text{proj} \left\{ g_1^2 \left(\hat{\theta}_p^{LEL} \right) P_{i,1} \right\}.$$



Рис. 2. Натурный эксперимент №1. Пары изображений с соответствиями точек и результатами оценки. Точки слева – согласованные элементы SIFT, кружки справа – элементы, перепроецированные с помощью оценки по методу LEL

На рис. 5 показаны оценки только тех измерений, которые соответствуют модели; на рис. 2, наоборот, выделены рассогласования между измеренной проекцией и ее оценкой после оптимального преобразования выбросов. На рис. 2 один из выбросов помечен знаком $\times$, соединенным с выбросом пунктирной ли-

нией. После получения результатов оценки двумя методами можно рассчитать квадраты нормированных невязок $\hat{\pi}_i$ (из уравнения (14)) оценок, полученных по методам LEL и робастного DLT, затем были рассчитаны члены

$$H_i(\hat{\theta}_j) = -\hat{\pi}_{i,j} \log \hat{\pi}_{i,j}, i = 1, \dots, N, \quad j = \{LEL, DLT\}$$

и нанесены на график относительно $\hat{\pi}_{i,j}$ на рис. 4. Выбросы соответствуют тем невязкам $\hat{\pi}_{i,j}$, значение которых очень близко к единице. Более того, виден разброс дисперсии этих невязок относительно других, относящихся к точкам, соответствующим модели, и находящихся ближе к нулю. Алгоритм робастного DLT работает примерно так же, так как его значение было рассчитано путем извлечения выбросов и самых «шумных» точек в наборе данных. Таким образом, можно утверждать, что предлагаемый метод (работающий на всем наборе данных) работает не хуже метода на основе второй нормы, если набор данных очищен от выбросов и неопределенных точек. Впервые такой вывод был сделан в [13]; полученные результаты подкрепляют его. Хотя это и не подтверждено аналитически, в общем, алгоритм LEL работает не хуже, чем алгоритм робастного DLT с нелинейным уточнением (оптимально настроенным), который использовался в качестве эталона. В некоторых экспериментах невозможно указать абсолютную точность методов вследствие отсутствия опорных данных для сравнения алгоритмов. Например, в *натурном эксперименте №2* представлен случай, анализ которого можно посчитать неопределенным. Базовая линия сильно меняется (за счет широкого движения камеры), и полученные изображения демонстрируют множество рассогласований, как показано на рис. 5. Оба метода демонстрируют достаточную точность оценки изменения положения, но, поскольку опорные данные (от дополнительного высокоточного датчика) недоступны, невозможно установить, какой алгоритм работает лучше. Чтобы выполнить качественное сравнение, мы определили два набора *возможных измерений, соответствующих модели*, выбрав точки, у которых квадрат ошибки перепроецирования ниже порога в два квадратных пиксела. Учитывая только эти результаты, оценка алгоритмом LEL обеспечивает квадрат ошибок перепроецирования немного ниже уровня ошибок алгоритма робастного DLT, примерно до невязки $I = 40$. Отсюда можно предположить, что алгоритм LEL работает немного лучше своего конкурента, например в ситуации, показанной на рис. 5.

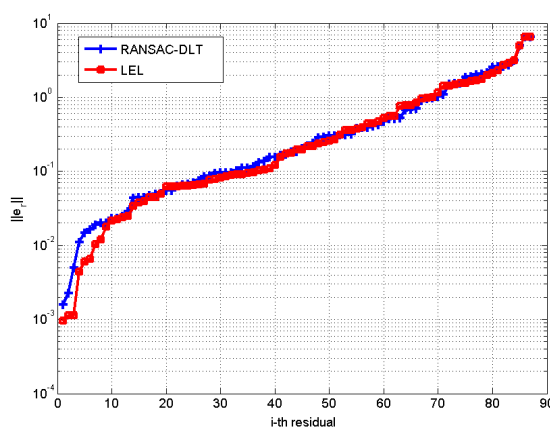


Рис. 3. Натурный эксперимент №1. Отсортированные ошибки перепроецирования

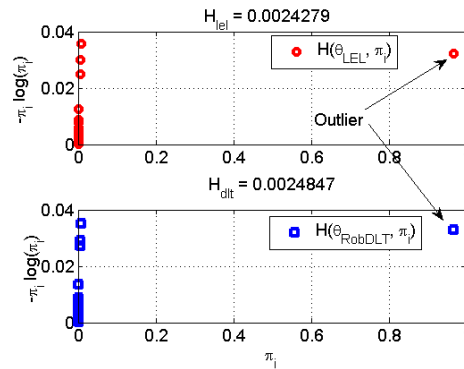


Рис. 4. Натурный эксперимент №1. Оценка энтропийно-подобной функции с помощью оценок по двум тестируемым методам. При оценке предлагаемым методом значение энтропийно-подобной функции меньше: имеет место немного большая дисперсия относительных невязок при минимизации LEL

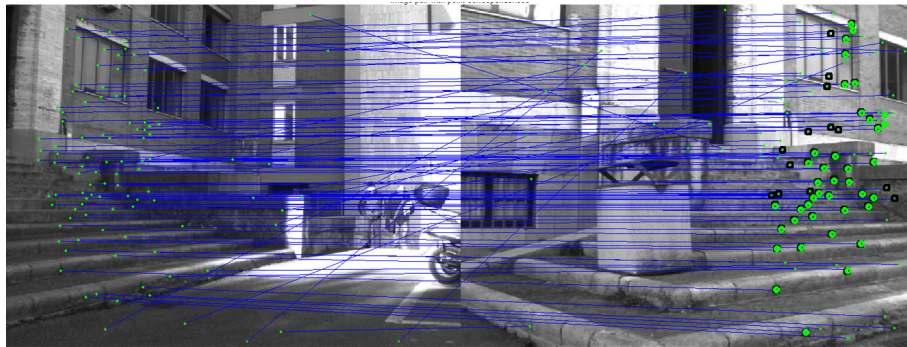


Рис. 5. Натурный эксперимент №2. Пары изображений с соответствиями точек и результатами оценки. Точки слева – согласованные элементы SIFT; светлые кружки справа – точки, соответствующие модели, оцененные по методу робастного DLT, черные квадратики (за кружками) – точки, соответствующие модели, оцененные по методу LEL

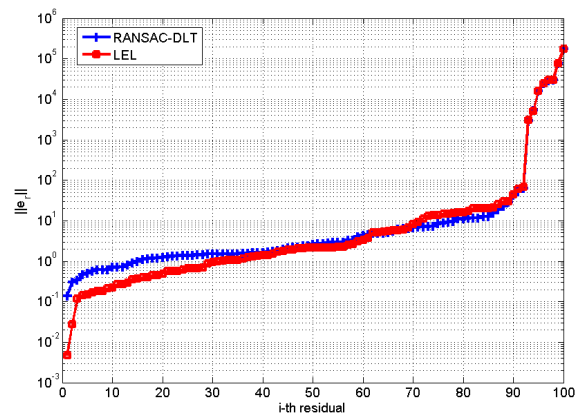


Рис. 6. Натурный эксперимент №2. Отсортированные ошибки перепроецирования

Чтобы численно оценить оба алгоритма, мы провели моделирование по методу Монте-Карло: работа алгоритмов LEL и DLT сравнивалась в 50 задачах оценки положения по 100 элементов в каждой; элементы были случайным образом распределены по конусу обзора камеры, а также были добавлены искусственные выбросы пяти различных величин: 10, 15, 20, 25 и 30% для 250 контрольных примеров. Значения среднего и среднеквадратического отклонений

распределения квадрата ошибки оценивания (отдельно для местоположения и пространственного положения) были записаны и нанесены на график в зависимости от уровня выбросов. На рис. 7 и 8 четко видно, что точность алгоритма LEL превосходит точность метода робастного DLT.

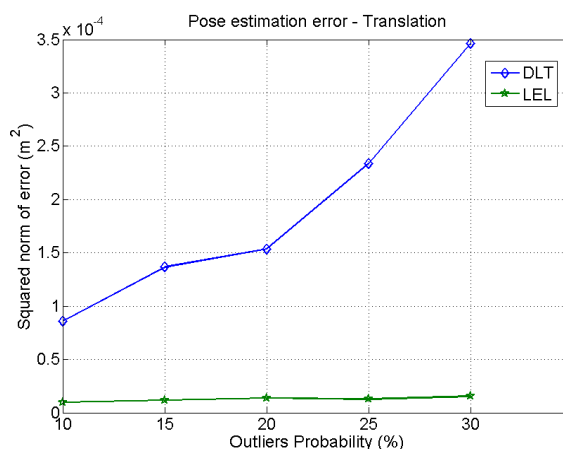


Рис. 7. Сравнение ошибки оценки положения при сдвиге.
 Pose estimation error-translation – Ошибка оценки положения – сдвиг
 Squared norm of error – Квадрат нормы ошибки (м²)
 Outliers Probability – Вероятность выбросов

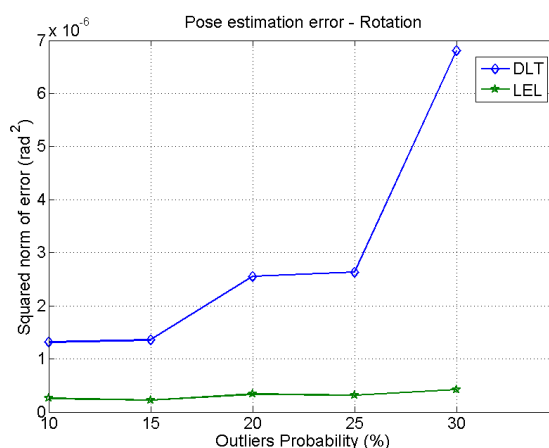


Рис. 8. Сравнение ошибки оценки положения при повороте:
 Pose estimation error-rotation – Ошибка оценки положения – поворот
 Squared norm of error – Квадрат нормы ошибки (рад²)
 Outliers Probability – Вероятность выбросов

4. Использование фильтра Калмана для комплексирования визуальных и инерциальных измерений

Для данного слабосвязанного метода видеоинерциальной навигации характерен дрейф, так как алгоритм навигации построен на принципе счисления [7, 10]. Это вызвано тем, что корректирующие данные, будучи по природе относительными, могут рассматриваться как *псевдоскорости*. Например, смещение нуля или почти постоянная ошибка в каждом псевдоизмерении будут приводить к отклоняющимся оценкам [17]. А поскольку при использовании только

относительных измерений невозможно уйти от уходов оценок, задача навигационного фильтра – максимально снизить дрейф [8].

4.1. Определение ошибки относительных псевдоизмерений положения

Обычно все абсолютные датчики производят абсолютные измерения переменных (относительно известных и постоянных эталонов) (например, GPS измеряет $\tilde{r}_e$ и $\tilde{V}_e$), а при визуальной одометрии используются только относительные измерения движения (т.е. измеряется относительное смещение между двумя последовательными изображениями). В данном разделе описываются уравнения, используемые для комплексирования относительных измерений движения, выработанных алгоритмом оценки положения, и инерциальных данных. Без потери общности можно допустить, что инерциальная система изначально выставлена относительно геодезической системы координат (углы Эйлера равны нулю), а в рамках данной работы эквивалентно допущению о знании начального пространственного положения.

Положение камеры в момент времени t_k относительно начального местоположения (в момент t_0) может быть выражено инерциальными состояниями (местоположение и ориентация) следующим образом:

$$g_k^0 = \{R_k^0, T_k^0\}, \quad (19)$$

где $R_k^0 = C_b^n(t_k)$ и T_k^0 – местоположение начала навигационной/корпусной СК в момент t_k , если смотреть от начальной навигационной СК (в момент t_0). В рамках данной работы допустим, что относительное смещение (по широте и долготе) между последовательными изображениями настолько мало, что ориентацию навигационной СК (Север-Восток-Вниз) можно считать неизменной относительно геоцентрической СК, связанной с Землей. Следовательно, можно использовать простую плоскостную проекцию (аппроксимацию плоской поверхности) для описания движения вблизи стартовой точки. Таким образом можно определить местоположение камеры:

$$T_k^0 = \xi(r^e(t_k)) = \begin{bmatrix} P_n(\phi_k - \phi_0) \\ P_n \cos \phi_k (\lambda_k - \lambda_0) \\ h_k \end{bmatrix}, \quad (20)$$

где $r^e(0) = [\phi_0 \quad \lambda_0 \quad 0]$ – местоположение объекта (широта и долгота) в момент начала решения навигационной задачи (t_0), P_n – радиус кривизны, перпендикулярный к поверхности эллипсоида в точке касания на данной широте ϕ [11].

Если имеются две стереопары в последовательные моменты времени t_k и t_{k+h} , относительное движение, рассчитываемое по данным видеосистемы, g_k^{k+h} связано с абсолютными положениями в моменты t_k и t_{k+h} следующим соотношением:

$$g_k^{k+h} = \{C_k^{k+h}, T_k^{k+h}\} = \{C_0^{k+h} C_k^0, C_0^{k+h} (T_k^0 - T_{k+h}^0)\}.$$

Изображения, используемые для оценки относительного движения, связаны с двумя разными моментами времени, которые могут быть удалены друг от друга как во времени, так и в пространстве.

Предлагаемая редуцированная версия косвенного фильтра Калмана оценивает ошибки по скорости (δV^n) , положению в геоцентрической СК, связанной с Землей (δr^e) и смещениям нулей акселерометров $(\delta f_{ib,b}^b)$. Вектор состояния выглядит следующим образом:

$$\delta x_k = \begin{bmatrix} \delta V_k^n & \delta r_k^e & \delta f_{ib,b}^b \end{bmatrix}. \quad (21)$$

Теперь необходимо определить выходные данные фильтра, которые можно использовать для построения невязки измерений совместно с данными видеосистемы. Сначала оценим ошибку навигационного положения с помощью оператора плоской проекции $\xi(\cdot)$:

$$\delta y_k = H_k \delta x_k \triangleq \delta T_k^0,$$

где

$$H_k = C_{n,k}^b \begin{bmatrix} 0 & \left. \frac{\partial T_k^0}{\partial r_k^e} \right|_{\hat{r}_k^e} & 0 \end{bmatrix}.$$

Следует отметить, что ненулевой элемент в матрице H_k является определителем Якоби функции $\xi(r_k^e)$ относительно оценки положения в геоцентрической СК, связанной с Землей около $\hat{r}_k^e$. Тогда можно оценить относительную навигационную ошибку $\Delta \delta y_{k+h}$ между моментами времени t_{k+h} и t_k :

$$\Delta \delta y_{k+h} = H_{k+h} \delta x_{k+h} - H_k \delta x_k$$

Отметим, что $\Delta \delta y_{k+h}$ – оценка навигационной ошибки значения $\hat{T}_k^{k+h}$, рассчитанного по инерциальным данным.

Поскольку в реальности видеосистема измеряет $\tilde{g}_k^{k+h} = \{\tilde{C}_k^{k+h}, \tilde{T}_k^{k+h}\}$ (член поворота не учитывается, так как считается, что используются гироскопы навигационного класса точности), можно рассчитать псевдомеру ошибки относительного положения по измеренному относительному положению $\tilde{T}_k^{k+h}$ и его оценке $\hat{T}_k^{k+h}$, восстановленным из навигационных уравнений, как функцию состояния фильтра. В данном случае ошибку можно записать следующим образом:

$$\Delta \delta y_{k+h}^* = \tilde{T}_k^{k+h} - \hat{T}_k^{k+h}, \quad (22)$$

если оценка относительного движения

$$\hat{T}_k^{k+h} = C_0^{k+h} (\hat{T}_k^0 - \hat{T}_{k+h}^0)$$

и измеренное относительное движение по определению равно реальной величине, включающей шум v :

$$\begin{aligned}
 \tilde{T}_k^{k+h} &= T_k^{k+h} + \mathbf{v}_k \\
 &= C_0^{k+h} (T_k^0 - T_{k+h}^0) + \mathbf{v}_k \\
 &= C_0^{k+h} (\hat{T}_k^0 - \delta T_k^0 - \hat{T}_{k+h}^0 + \delta T_{k+h}^0) + \mathbf{v}_k \\
 &= C_0^{k+h} (\hat{T}_k^0 - \hat{T}_{k+h}^0) - C_0^{k+h} (\delta T_k^0 - \delta T_{k+h}^0) + \mathbf{v}_k.
 \end{aligned}$$

Таким образом, псевдомера ошибки относительного положения может быть переписана как функция состояний (текущего и прошлого) косвенного фильтра Калмана:

$$\Delta \delta y_{k+h}^* = C_0^{k+h} (\delta T_{k+h}^0 - \delta T_k^0) + \mathbf{v}_k.$$

Один из важных аспектов заключается в том, что приведенная выше формулировка ошибки измерения местоположения зависит от переменных движения, относящихся к текущему времени (через $\hat{r}_{k+h}^e$ и C_0^{k+h}) и к некоторым шагам в прошлом (через $\hat{r}_k^e$). Значит, нужно добавить в состояние фильтра память о прошлом; это позволит отслеживать взаимную ковариацию оценок между двумя моментами времени [15].

Состояние навигационного фильтра ошибок дополняется одной точной копией $\delta \tilde{x}_k$ самого себя при получении ключевого кадра. Предположим, новый опорный ключевой кадр поступает в момент времени t_k , при состоянии фильтра Калмана

$$\delta \bar{x}_k = \begin{bmatrix} \delta \tilde{x}_k \\ \delta x_k \end{bmatrix} \quad (23)$$

и матрице ковариации состояния

$$\bar{P}_k = \begin{bmatrix} P_k & P_k \\ P_k & P_k \end{bmatrix} \quad (24)$$

при $P_k = E \{ \delta x_k \delta x_k^T \}$. Копии состояния $\delta \tilde{x}_k$ присваивается начальное значение δx_k , которое постоянно удерживается в течение работы фильтра, в то время как вектор состояния δx_k распространяется в соответствии с динамикой ошибок.

4.2. Распространение навигационных уравнений ошибок

На каждом такте выполняются расчеты по инерциальным данным для получения новой оценки состояния объекта (местоположение, скорость и смещения нулей акселерометров):

$$\hat{x}_{k+1}^- = f(\hat{x}_k, u_k)$$

где $f(\cdot)$ – дискретизованная версия стандартных данных ИИМ, которая отображает скорректированные навигационные переменные состояния на состоя-

ния на следующем такте; переменная $\hat{x}_{k+1}^-$ – оценка местоположения и скорости объекта (в момент t_{k+1}) до учета поправок, выработанных фильтром Калмана (если таковые имеются) (т.е. *априорная оценка*).

Этап предсказания косвенного фильтра Калмана выполняется с помощью выражения

$$\begin{bmatrix} \delta \tilde{x}_{k+1}^- \\ \delta x_{k+1}^- \end{bmatrix} = \begin{bmatrix} I & 0 \\ 0 & \Psi_k \end{bmatrix} \begin{bmatrix} \delta \tilde{x}_k^- \\ \delta x_k^- \end{bmatrix} + \begin{bmatrix} 0 \\ \Gamma_k \end{bmatrix} w_k = \bar{\Psi}_k \delta \bar{x}_k + \bar{\Gamma}_k w_k. \quad (25)$$

Уравнение прогноза для ковариационной матрицы, как и для стандартного фильтра Калмана:

$$\bar{P}_{k+1} = \bar{\Psi}_k \bar{P}_k \bar{\Psi}_k^T + \bar{\Gamma}_k Q \bar{\Gamma}_k^T, \quad (26)$$

где Q – матрица ковариации шума.

После h тактов (по прошествии времени, необходимого для получения второго изображения) ковариационная матрица приобретает вид:

$$\bar{P}_{k+h} = \begin{bmatrix} P_k & P_k \left(\prod_{i=1}^h \Psi_{k+i} \right)^T \\ P_k \prod_{i=1}^h \Psi_{k+i} & P_{k+h} \end{bmatrix}. \quad (27)$$

Недиagonальные элементы представляют взаимную корреляцию между навигационными ошибками в моменты t_k и t_{k+h} . После получения от видеосистемы нового измерения относительного положения выполняется этап коррекции:

$$\begin{aligned} S_{k+h} &= \begin{bmatrix} H_k & H_{k+h} \end{bmatrix} \bar{P}_{k+h}^- \begin{bmatrix} H_k^T \\ H_{k+h}^T \end{bmatrix} + R_k, \\ K_{k+h} &= \bar{P}_{k+h}^- \begin{bmatrix} H_k^T \\ H_{k+h}^T \end{bmatrix} S_{k+h}^{-1}, \\ \Delta \delta y_{k+h}^* &= \tilde{T}_k^{k+h} - \hat{T}_k^{k+h}, \\ \delta x_{k+h}^+ &= \delta x_{k+h}^- + K_{k+h} \big|_{\delta x_{k+h}} \left(\Delta \delta y_{k+h}^* - \Delta \delta y_{k+h} \right), \\ \hat{x}_{k+h}^+ &= \hat{x}_{k+h}^- + \delta x_{k+h}^+, \\ \bar{P}_{k+h}^+ &= \bar{P}_{k+h}^- - K_{k+h} \begin{bmatrix} H_k & H_{k+h} \end{bmatrix} \bar{P}_{k+h}^-, \end{aligned} \quad (28)$$

где R_k – матрица ковариации шума измерений, переменная $\hat{x}_{k+h}^+$ – оценка местоположения и скорости объекта (в момент t_{k+h}) с учетом коррекций фильтра Калмана (т.е. *апостериорная оценка*).

5. Результаты моделирования

В данном разделе представлено моделирование с использованием предлагаемого робастного метода оценки положения в рамках инерциальной навигации. Проводится сравнение метода с методом DLT на основе RANSAC с нелинейным уточнением на основе блочного выравнивания.



Рис. 9. Искусственная среда, используемая при моделировании

При моделировании вырабатывались выборочные (зашумленные) ускорения и *незашумленные* угловые скорости. В результате была выработана траектория камеры с шестью степенями свободы. Движения при ускорениях и угловых скоростях имитировались с помощью имитатора летательного аппарата (ЛА) с четырьмя несущими винтами с вертикальным взлетом и посадкой. На рис. 9 продемонстрирован снимок искусственной среды [22] для моделирования траектории камеры и сбора точек изображений. Вдоль этой траектории было получено и спроецировано на плоскости изображений стереопары несколько трехмерных точек. Полагалось, что процесс инерциальных измерений пройдет в 5 раз быстрее, чем процесс видеоизмерений. Для искажения двумерных элементов использовались шум и некоторое количество выбросов. Пиксельный шум представлял собой белый шум с нулевым средним и дисперсией $0,5 \text{ пкс}^2$. Чтобы симитировать выбросы в данных, к двумерным координатам ключевых точек на пространстве изображения были добавлены случайные цифры. Средняя ошибка, вызванная выбросами, составила $5/40 \text{ пкс}$, а процент выбросов относительно всего набора данных составил $5/50\%$. На каждом такте, соответствующем получению нового изображения, вырабатывалась новая вероятность засоренности выбросами, и в соответствии с этой вероятностью рисовались определенное количество выбросов. На рис. 10 показана абсолютная навигационная ошибка (по местоположению) для алгоритмов LEL и DLT. На рис. 11 представлена ошибка оценки смещения нулей акселерометров для обоих алгоритмов. Совершенно очевидно, что алгоритм LEL работает лучше, чем DLT. Чтобы оценить и сравнить сложность этих алгоритмов, в течение всех экспериментов записывалось время выполнения алгоритмов (см. таблицу).

Сравнение времени выполнения алгоритмов DLT и LEL

	DLT	LEL
Среднее время вычислений (мс)	125,0	116,2
Среднеквадратическое отклонение – 1 сигма (мс)	69,7	51,4

Оба алгоритма реализованы в программной среде Matlab с использованием наиболее эффективных методов кодирования, и мы полагаем, что в данной среде невозможно больше сократить время выполнения алгоритмов. Хотя эти качественные показатели и не могут заменить анализ вычислительной сложности, можно сделать вывод о том, что алгоритмы имеют сопоставимое время выполнения.

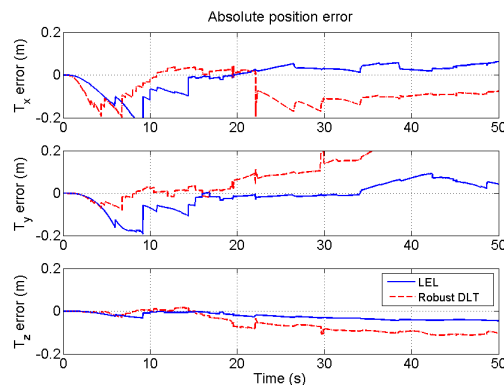


Рис. 10. Погрешности местоположения при моделировании

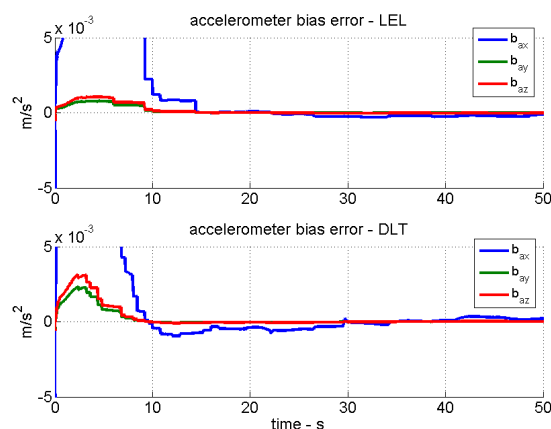


Рис. 11. Погрешности смещения нулей акселерометров при моделировании

6. Выводы

В статье предлагается робастный слабосвязанный метод видеоинерциальной навигации, в котором используется новая штрафная функция стоимости, — *функция энтропии* квадратов относительных невязок. Моделирование и эксперименты показали, что такая функция стоимости робастна к наличию шума и выбросов в визуальных элементах. Результаты моделирования подтвердили, что использование метода LEL для решения задачи оценки положения позволяет уменьшить дрейф по сравнению с робастным методом, основанным на минимизации второй нормы с нелинейным уточнением за счет блочного выравнивания. Кроме того, алгоритм прост в вычислении (время его выполнения сопоставимо со временем выполнения алгоритма прямого линейного преобразования), он не требует больших ресурсов памяти (благодаря слабой связи), а его устойчивость

к выбросам позволяет использовать простые следящие функции.

Предлагаемый интегрированный навигационный метод можно использовать во всех тех случаях, когда ИИМ, имеющий относительно высокую точность для оценки ориентации, нельзя использовать для долговременной оценки местоположения, а сигнал GPS недоступен или пропадает. Фильтр оценки можно модифицировать, включив в него данные об абсолютном местоположении (от GPS или ориентиров).

ЛИТЕРАТУРА

1. **R. Hartley and F. Kahl**, "Optimal algorithms in multiview geometry", in Asian Conference of Computer Vision, 2007.
2. **A. Milella and R. Siegwart**, "Stereo-Based Ego-Motion Estimation Using Pixel Tracking and Iterative Closest Point", in Proc. of the 4-th IEEE International Conference on Computer Vision Systems, 2006.
3. **D. Nistér**, "Preemptive RANSAC for Live Structure and Motion Estimation", in IEEE International Conference on Computer Vision, 2003.
4. **Campa G., Mammarella M., Napolitano M. R., Fravolini M. L., Pollini L., Stolarik B.**, "A comparison of Pose Estimation algorithms for Machine Vision based Aerial Refueling for UAVs", IEEE Mediterranean Conference on Control and Automation, vol. 1, 2006.
5. **E. Jones and S. Soatto**, "Visual-Inertial Navigation, Mapping and Localization: A Scalable Real-Time Causal Approach", International Journal of Robotics Research, Volume 30 Issue 4, 2011.
6. **A. Mourikis, N. Trawny, S. Roumeliotis, A. Johnson, A. Ansar and L. Matthies**, "Vision-Aided Inertial Navigation for Spacecraft Entry, Descent, and Landing", IEEE Transactions on Robotics, vol. 25, no. 2, pp. 264-280, April 2009.
7. **S. Roumeliotis, A. Johnson and J. Montgomery**, "Augmenting inertial navigation with image-based motion estimation", in Proc. of IEEE International Conference on Robotics and Automation, 2002.
8. **J.-P. Tardif, M. George, M. Laverne, A. Kelly and A. Stentz**, "A new approach to vision-aided inertial navigation", in Proc. of International Conference on Intelligent Robots and Systems (IROS), 2010.
9. **P. J. Huber and E. M. Ronchetti**, "Robust Statistics", Second Edition, John Wiley & Sons, Inc, 2009.
10. **K. Konolige, M. Agrawal and J. Solà**, "Large-Scale Visual Odometry for Rough Terrain", Robotics Research; Springer Tracts in Advanced Robotics, vol. 66, pp. 201-212, 2011.
11. **R. Rogers**, "Applied Mathematics in Integrated Navigation Systems", American Institute of Aeronautics and Astronautics, 2007.
12. **D. Lowe**, "Object Recognition from Local Scale-Invariant Features", in Proc. of the International Conference on Computer Vision (ICCV), 1999.
13. **G. Indiveri**, "An Entropy-Like Estimator for Robust Parameter Identification", Entropy, vol. 11, pp. 560-585, 2009.
14. **C. Chakrabarti and K. De**, "Boltzmann-Gibbs entropy: axiomatic characterization and application", Journal of Mathematics and Mathematical Sciences, vol. 23, no. 4, 2000.
15. **F. Di Corato, M. Innocenti and L. Pollini**, "An Entropy-Like Approach to Vision-Aided Inertial Navigation", in Proc. of 18th IFAC World Congress, Milan, Italy, 2011.
16. **F. Di Corato, M. Innocenti, G. Indiveri and L. Pollini**, "An Entropy-Like Approach to Vision Based Autonomous Navigation", in Proc. of the IEEE International Conference on Robotics and Automation, 2011.

17. **D. Nistér, O. Naroditsky and J. Bergen**, "Visual Odometry", in Proc. IEEE Conference on Computer Vision and Pattern Recognition, 2004.
18. **J. Hornegger and C. Tomasi**, "Representation issues in the ML estimation of camera motion", in Proc. of the 7-th IEEE Int. Conference on Computer Vision, 1999.
19. **J. Schmidt and H. Niemann**, "Using Quaternions for Parametrizing 3-D Rotations in Unconstrained Nonlinear Optimization", in Proc. of the Vision Modeling and Visualization Conference, 2001.
20. **R. I. Hartley and A. Zisserman**, "Multiple View Geometry in Computer Vision", Cambridge University Press, 2000.
21. **B. Triggs, P. McLauchlan, R. Hartley and A. Fitzgibbon**, "Bundle adjustment-a modern synthesis", Vision Algorithms: Theory and Practice, Lecture Notes in Computer Science, vol. 1883, pp. 298-372, 2000.
22. **Pollini, L. Innocenti, M.**, "A synthetic environment for dynamic systems control and distributed simulation", IEEE Control Systems Magazine, Vol. 20, No. 2, 2000.

Abstract. The paper presents a loosely coupled approach for the improvement of state estimation in autonomous inertial navigation, augmented via image based relative motion estimation. The proposed approach uses a novel pose estimation technique based on the minimization of an entropy like cost function which is robust by nature to the presence of noise and outliers in the visual features. Experimental evidence of the performance of this approach is given and compared to a state of the art algorithm. An indirect Kalman filter is used for navigation in the framework of *stochastic cloning*. The robust relative pose estimation given by our novel technique is used to feed a relative position fix to the navigation filter. The simulation results are presented and compared with the results obtained via the classical RANSAC based direct linear transform approach.

Key words: robust vision_aided inertial navigation, inertial navigation systems, indirect Kalman filter, pose estimation, entropy like cost function, RANSAC

Материал поступил 31.05.12