

УДК 519.688

ПОЗИЦИОНИРОВАНИЕ ИЗОБРАЖЕНИЙ МОРСКОГО ДНА, ПОЛУЧЕННЫХ С ПОМОЩЬЮ АНПА

А. Н. Камаев

Вычислительный центр ДВО РАН¹

Рассматривается задача определения планарных координат изображений, получаемых с АНПА, в системе координат с центром в одном из этих изображений. Предлагается способ быстрого обнаружения пересекающихся пар изображений в больших массивах изображений на основе визуальных особенностей. Также рассматривается возможность выполнения быстрого поиска связанных пар в условиях ограниченной памяти.

ВВЕДЕНИЕ

Подводное дно – интересный и вместе с тем труднодоступный для изучения объект. Перспективным методом исследования дна является его фотографирование с помощью автономных необитаемых подводных аппаратов (АНПА) [1]. Поскольку глубоко под водой освещение отсутствует, для получения качественных снимков необходимо производить съемку на небольшом удалении от поверхности дна. Это приводит к тому, что каждый снимок охватывает малый участок территории, следовательно, для покрытия больших участков необходимо большое количество снимков. Так как работать с общей фотографической картой исследуемого участка подводного дна значительно удобнее, чем с набором из тысяч разрозненных изображений, актуальной представляется задача сшивки подводных изображений в такие карты. Кроме того, задача сшивки подводных изображений тесно пересекается с задачей навигации АНПА по фото- или видеоинформа-

ции. Обе эти задачи требуют позиционирования снимков относительно друг друга.

Одна из первых систем автоматической сшивки подводных изображений представлена в работе [2]. В данной системе рассматриваются изображения с заранее известным положением. Успешная попытка использования последовательной видеоинформации для расчета траектории аппарата была предпринята позже в работе [3]. Подход, предложенный в данной работе, заключается в обнаружении и сопоставлении особых точек на последовательных изображениях. Для увеличения производительности процедуры сопоставления используется фильтр Калмана, с помощью которого предсказываются положения особенностей на каждом следующем изображении. Для вычисления траектории (смещения между особыми точками на последовательных изображениях) используется обобщенное преобразование Хафа. В публикациях [4, 5, 6] рассматриваются похожие на [3] подходы. Отличия в основном заключаются в способе установления связи

между последовательными изображениями. В [7] описан подход к построению мозаик на основе различных последовательностей изображений, полученных на разных высотах над дном. Мозаика, собранная из изображений, полученных с большой высоты, используется для построения мозаик из изображений, полученных с малой высоты. Очевидно, такой подход хорошо подходит для небольших глубин, на которые проникает свет и где можно производить съемку с большой высоты.

Современный подход к построению фотографических карт дна или мозаик состоит из четырех основных этапов: сопоставление последовательных изображений, сопоставление остальных пересекающихся пар изображений, вычисление движения камер и смещение изображений, спроецированных на поверхность карты. При этом процесс сопоставления непоследовательных пар ис-

¹ 680000, Хабаровск, ул. Ким Ю Чена, 65. Тел.: 8 (924) 211-10-16. E-mail: Kamaev_an@mail.ru

пользует приближённую структуру движения, полученную на основе сопоставления последовательных пар. Эта структура движения позволяет с некоторой точностью определить места, в которых пересекаются непоследовательные пары изображений. После обнаружения набора дополнительных пар структура движения уточняется, и процесс поиска непоследовательных пересекающихся пар может быть выполнен повторно для уточнённой структуры. Подобный подход реализован в работе [8]. Похожий подход, но использующий дополнительную информацию о положении АНПА из навигационной системы, представлен в [9]. Недостатком подхода является то, что сшивка последовательных изображений далеко не всегда определяет структуру движения с достаточной точностью. Точности рассчитанной структуры может не хватить для обнаружения пересекающихся непоследовательных пар изображений. Более того, иногда последовательную сшивку осуществить невозможно.

Рассмотрим три случая, когда последовательная сшивка изображений невозможна либо неточна. Первый случай – обследование большой зоны, когда между пересечениями своей траектории АНПА проходят большие расстояния. Накапливающаяся при последовательной сшивке ошибка легко может достигнуть 10 м и более, что потребует сравнения большого числа пар изображений между собой для замыкания траектории. Второй случай – недостаточно пересекающиеся последовательные изображения. Когда АНПА идёт быстро или редко выполняет фотографирование, между многими соседними снимками не будет обнаружено соответствий, что

приведёт к более быстрому накоплению ошибки. Третий случай – особенности дна. На дне могут быть большие однородные зоны, не содержащие особенностей для сопоставления последовательных снимков.

Чтобы сделать процесс построения фотографических карт возможным для любых последовательностей изображений, независимо от разрывов и пропусков данных, а также от размера исследуемой территории, в статье предлагается механизм обнаружения положения каждого изображения, не предъявляющий никаких требований к порядку этих изображений. Рассматриваются процедура быстрого сопоставления каждого изображения со всем доступным набором изображений, а также простая процедура позиционирования изображений на плоскости. Планарные координаты, рассчитанные с помощью описанной процедуры, могут быть использованы для инициализации нелинейной задачи вычисления полной модели движения АНПА, учитывающей все шесть степеней свободы.

Предложенная процедура обнаружения связей в полном наборе изображений может быть полезна не только при составлении фотографических карт дна, но и в задаче навигации по изображениям. Если АНПА находится близко к пересечению своей траектории после длительного прохода без пересечений либо после серии пропущенных данных, то для замыкания петли может быть использован предложенный метод.

1. Постановка задачи

АНПА движется над поверхностью дна на некотором расстоянии и непрерывно его фотографирует. Один из вари-

антов возможной траектории аппарата представлен на рис. 1.

Здесь жирной линией показана возможная траектория движения АНПА, а тонкой – границы снимков, сделанных АНПА во время движения. На рисунке показан идеальный случай, когда все соседние снимки имеют хорошее (достаточное для установления связи, но не избыточное) перекрытие. В действительности перекрытие соседних снимков может варьироваться от полного до нулевого, что не позволяет использовать стандартные методы [8, 9].

Кроме самих изображений входными данными для задачи являются параметры навигационной системы АНПА в момент получения каждого i -го снимка, $i=1, 2, \dots, m$, где m – общее количество изображений. К таким параметрам можно отнести курс аппарата γ_i и высоту аппарата над дном h_i . Параметры γ_i и h_i могут быть определены достаточно точно и позволяют повысить скорость и надёжность процедуры сопоставления изображений. Кроме курса и высоты над дном, навигационная система может содержать информацию о предполагаемых координатах аппарата в момент фотографирования: $\mathbf{x}_i = (x_i, y_i)$. Эта информация является необязательной, но может помочь в случае, когда m велико (больше 10000). В разделе 7 будет рассмотрен способ использования данной информации.

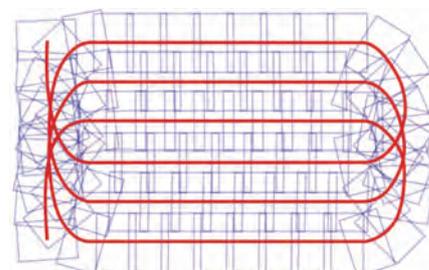


Рис. 1. Пример траектории АНПА

Результатом процесса позиционирования являются координаты центров изображений $c_i = (c_{ix}, c_{iy})$, наилучшим образом согласующиеся с визуальной информацией на этих изображениях. Система координат, в которой задаются c_i – это система координат с центром в одном из изображений. Таким образом, необходимо найти положения центров всех изображений относительно одного изображения. При позиционировании в данной статье не рассматривается положение каждого снимка по 3-й вертикальной координате. Движение АНПА считается плоским, а отклонения крена и дифферента несущественными.

Решение поставленной задачи состоит из нескольких этапов: выделение визуальных особенностей на изображениях, сопоставление пар изображений, вычисление координат c_i на основе сопоставленных пар. Далее каждый из представленных этапов рассматривается подробно.

2. Визуальные особенности на изображениях

Чтобы определить, имеют ли два изображения пересечения, необходимо выделить на этих изображениях некоторые особенности и определить, есть ли среди выделенных особенностей попарно схожие. Особенности будем называть такие точки изображения, которые хорошо выделяются в своей окрестности и с высокой вероятностью повторяются на других изображениях одного и того же объекта, полученных в различных условиях съемки. Условия съемки определяются положением АНПА над фотографируемым объектом, высотой над поверхностью дна, курсом и освещением.

Таким образом, одни и те же особенности должны сохраняться на различных изображениях одного объекта, сделанных с различного положения, с разным масштабом, ориентацией и освещением. Данному критерию удовлетворяют SIFT особенности [10].

Кроме определения самих особенностей важна процедура их сравнения, в результате которой выявляются схожие особенности. Чтобы к особенности могла быть применена операция сравнения, ей в соответствие должен быть поставлен уникальный вектор – дескриптор, описывающий данную особенность. Чем больше размерность дескриптора, тем точнее может быть описана особенность, но тем сложнее будет процедура сравнения дескрипторов разных особенностей. Кроме того, дескриптор, как и сама особенность, должен быть инвариантен к изменению положения съёмки, масштаба, ориентации и освещения.

Одним из наиболее известных в компьютерном зрении дескрипторов, удовлетворяющих предъявляемым требованиям, является SIFT дескриптор [10]. Данный дескриптор является 128-компонентным вектором и составляется на основе градиентов яркости в точках окрестности особенности и полностью инвариантен к перемещению, масштабу, плоскому повороту и линейному изменению освещения. Также SIFT дескриптор частично инвариантен к аффинным искажениям и нелинейному изменению освещения. Поскольку для подводного фотографирования используется освещение, привязанное к АНПА, для получаемых изображений характерны сильные нелинейные изменения освещения, поэтому в [11] рассмотрена процедура предварительного

выравнивания освещения перед составлением SIFT дескриптора.

После того как все особенности определены и их дескрипторы составлены, степень схожести любых двух особенностей может быть определена как евклидово расстояние между их дескрипторами. Чем расстояние меньше, тем больше похожи особенности и, наоборот, чем расстояние больше, тем сильнее особенности различаются.

3. Сопоставление пар изображений

Рассмотрим два изображения с номерами i и j из набора, полученного с АНПА. Будем считать, что у каждого изображения есть собственная система координат с началом в центре изображения и осями, параллельными сторонам изображения. Координате некоторой точки $x_j = (x_j, y_j)$ в системе координат изображения j будет соответствовать точка $x_i = (x_i, y_i)$ в системе координат изображения i :

$$\begin{pmatrix} x_i \\ y_i \end{pmatrix} = \frac{h_j}{h_i} \cdot B \cdot \begin{pmatrix} x_j \\ y_j \end{pmatrix} + \begin{pmatrix} t_x \\ t_y \end{pmatrix}, \quad (1)$$

где $B = \begin{pmatrix} \cos(\gamma_j - \gamma_i) & \sin(\gamma_j - \gamma_i) \\ -\sin(\gamma_j - \gamma_i) & \cos(\gamma_j - \gamma_i) \end{pmatrix}$.

Здесь $t_{ij} = (t_x, t_y)$ – вектор смещения центра j -го изображения относительно центра i -го изображения. Данный вектор имеет масштаб i -го изображения и является единственным неизвестным в формуле (1). Процесс расчёта параметров $\alpha_{ij} = \gamma_i - \gamma_j$, $s_{ij} = h_j/h_i$ и вектора t_{ij} на основе SIFT особенностей подробно описан в [11], где рассмотрена процедура вычисления данных параметров для пар изображений, полученных в тяжёлых условиях съемки: слабая не-

линейная освещённость, отсутствие чётких, контрастных деталей, большая повторяемость однотипных объектов. Для изображений, не имеющих пересечений или имеющих слишком малую площадь пересечения, алгоритм из [11] определит отсутствие связи.

Поскольку в рассматриваемой задаче параметры a_{ij} и s_{ij} также могут быть найдены на основе данных навигационной системы АНПА, они могут быть использованы для контроля корректности рассчитанной связи с использованием метода [11]. Если параметры навигационной системы и параметры, рассчитанные в [11], различаются на величину, большую погрешности навигационной системы, то обнаруженную связь не следует рассматривать как корректную.

Перебрав все пары изображений ij : $i, j=1, 2, \dots, m$ и применив к ним алгоритм из [11], обнаружим n связанных пар изображений $i(l) j(l)$, для которых известны векторы $\mathbf{t}_l = \mathbf{t}_{i(l)j(l)}$, где $l=1, 2, \dots, n$.

4. Позиционирование

В рассматриваемой задаче будем считать, что все изображения связаны (граф, построенный из вершин – изображений и рёбер – известных связей $i(l) j(l)$, связан). Если это не так, то следует из множества изображений выделить подмножества связанных изображений и решать задачу позиционирования для каждого подмножества отдельно. При этом связь между несвязными компонентами может быть обнаружена только на основе дополнительной информации, например, данных о координатах из навигационной системы АНПА.

Задача позиционирования заключается в нахождении

планарных координат центров всех изображений $\mathbf{c}_k = (c_{kx}, c_{ky})$, $k=1, 2, \dots, m$ в системе координат с центром в некотором изображении $p \in \{1, 2, \dots, m\}$. Переведём все векторы $\mathbf{t}_l = \mathbf{t}_{i(l)j(l)}$ (раздел 3) в общую систему координат, домножив их на масштаб, определяемый высотой АНПА над дном $h_{i(l)}$ и повернув на курс $\gamma_{i(l)}$ (2).

$$\mathbf{d}_l = \frac{h_{i(l)}}{f} \cdot \begin{pmatrix} \cos \gamma_{i(l)} & \sin \gamma_{i(l)} \\ -\sin \gamma_{i(l)} & \cos \gamma_{i(l)} \end{pmatrix} \cdot \mathbf{t}_l.$$

Здесь \mathbf{d}_l – вектор смещения центра изображения $j(l)$ относительно центра изображения $i(l)$, заданный в общей системе координат, а f – фокусное расстояние камеры АНПА в пикселях. Использование фокусного расстояния позволяет получить масштаб векторов \mathbf{d}_l , соответствующий масштабу, в котором измерены высоты $h_{i(l)}$. Каждый вектор \mathbf{d}_l даёт два уравнения относительно неизвестных координат центров изображений \mathbf{c}_k : $\mathbf{d}_l = \mathbf{c}_{j(l)} - \mathbf{c}_{i(l)}$. Совокупность уравнений для $l=1, 2, \dots, n$ и известные координаты $\mathbf{c}_p = (0, 0)$ составляют систему линейных алгебраических уравнений. Эффективный способ решения систем подобного вида рассмотрен в [12].

5. Проблемы обнаружения связанных пар изображений

Сопоставление пар изображений – самый затратный по времени этап, так как необходимо проводить процедуру сопоставления для очень большого числа пар. Если с АНПА было получено m фотографий, то из них можно составить N пар (3) изображений:

$$N = \frac{m^2 - m}{2}.$$

Если с АНПА было получено 10000 изображений, то для

обнаружения связанных пар потребуется выполнить $N \approx 5 \cdot 10^7$ процедур сопоставления. На современном компьютере процедура сопоставления, описанная в [11], занимает порядка 0.1–0.7 секунд (в зависимости от используемого количества особенностей на каждом изображении: от 500 до 1000) на одном ядре процессора Intel Core2 Quad Q9550. Это означает, что сопоставление $5 \cdot 10^7$ пар потребует от 58 до 405 суток на ядро, а для пар, составленных из 50000 изображений, потребуется в среднем около 15 лет. Чтобы решить задачу за приемлемое время, необходимо уменьшить количество сопоставляемых пар.

В основе процедуры сопоставления пар изображений [11] лежит процесс сравнения дескрипторов особых точек. Чем больше дескрипторов одного изображения оказались близкими к дескрипторам другого, тем больше вероятность, что для пары будет определена связь. Таким образом, связь следует искать лишь для пар таких изображений, количество близких дескрипторов особенностей между которыми больше, чем некоторое пороговое число.

Обозначим дескриптор j -й особенности i -го изображения за v_{ij} , $i=1, 2, \dots, m, j=1, 2, \dots, \lambda(i)$, где $\lambda(i)$ – количество особенностей на i -м изображении. Предположим, что существует быстрый способ для дескриптора v_{ij} , $i \in \{1, 2, \dots, m\}, j \in \{1, 2, \dots, \lambda(i)\}$ найти вектор \mathbf{l}_{ij} , состоящий из номеров изображений, содержащих первые k особенностей с дескрипторами, максимально близкими к дескриптору v_{ij} : $\mathbf{l}_{ij} = (l_1^j, l_2^j, \dots, l_k^j)$. Подробно этот способ описан в разделе 6.

Зная \mathbf{l}_{ij} , можно составить списки предполагаемых кандидатов на сопоставление для

каждого изображения. Для этого введём матрицу $W=(w_{pb})$, $p,b=1, 2, \dots, m$, где w_{pb} – количество похожих дескрипторов между парой изображений p и b . Изначально все w_{pb} приравниваются к нулю. Далее перебираются все элементы l_n^{ij} , $i=1, 2, \dots, m, j=1, 2, \dots, \lambda(i), n=1, 2, \dots, k$ и компоненты матрицы W : $w_{i,j}^{ij}$ и $w_{j,i}^{ij}$ увеличиваются на единицу. В список предполагаемых кандидатов на сопоставление каждого изображения $S_p=(s_{p1}, s_{p2}, \dots, s_{pn})$, $p=1, 2, \dots, m$ помещаются η номеров столбцов наибольших элементов из p -й строки матрицы W .

После вычисления списков S_p процедуру сопоставления нужно выполнять не для (3), а всего лишь для $N=\eta \cdot m$ пар. Каждое p -е изображение сопоставляется лишь с изображениями из списка S_p . Влияние параметров η и k на процесс поиска связанных изображений рассмотрено в разделе 8.

6. Поиск похожих дескрипторов особенностей

Ключом к быстрому обнаружению потенциально пересекающихся пар изображений является процедура быстрого поиска дескрипторов особенностей, близких к заданному дескриптору. В качестве дескрипторов будем рассматривать 128-мерные вектора – SIFT дескрипторы [10], а в качестве меры близости – евклидово расстояние между этими векторами.

Запрос на поиск выполняется для некоторого дескриптора q в наборе дескрипторов $V=\{v_i\}$, $i=1, 2, \dots, n$, где n – количество особенностей, найденных на изображениях. Результатом поиска будет k дескрипторов из V , максимально близких к q . Основная проблема данного вида поиска – высо-

кая размерность используемых дескрипторов.

Для поиска в высокоразмерных пространствах необходимо организовать элементы множества V таким образом, чтобы возможно было построить способ поиска k дескрипторов, максимально близких к q , за время быстрее линейного.

Можно перечислить следующие наиболее известные способы организации дескрипторов и методы поиска: kd-деревья [13], best bin first [14], локально чувствительное хеширование (LSH) [15], метрические деревья [16], sr -деревья и гибридные sr -деревья [17]. Согласно исследованиям, описанным в статье [17], время поиска в sr -деревьях и метрических деревьях в разы меньше времени поиска в других структурах. Недостатками sr -деревьев является большое количество требуемой памяти и неопределённая глубина дерева, тогда как метрическим деревьям нужно значительно меньше памяти, однако время поиска в них больше. Гибридные sr -деревья являются промежуточным вариантом между sr и метрическими деревьями, позволяющим выбирать между объёмом потребляемой памяти и временем поиска.

6.1. Метрические и sr -деревья

Метрическое дерево – бинарное дерево, узлы которого представляют множества дескрипторов. При этом корневой узел представляет целиком множество V .

Рассмотрим внутренний узел дерева t . Множество точек, представляемое данным узлом $V(t)$, делится на два подмножества $V(t.lc)$ и $V(t.rc)$, представляемые двумя дочерними узлами t : $t.lc$ и $t.rc$. При этом $V(t)=V(t.lc) \cup V(t.rc)$, также для

метрического дерева $V(t.lc) \cap V(t.rc) = \emptyset$. Обратим внимание, что для sr -дерева, в отличие от метрического, последнее равенство не всегда справедливо. Узлы самого нижнего уровня содержат некоторое минимальное количество дескрипторов и далее не делятся.

Рассмотрим стратегию деления множеств $V(t)$ на множества $V(t.lc)$ и $V(t.rc)$. Сначала необходимо выбрать в $V(t)$ два дескриптора $t.lpv$ и $t.rpv$, расстояние между которыми максимально: $|t.lpv-t.rpv|=\max_{a,b \in V(t)} |a-b|$. Выбор таких точек имеет вычислительную сложность $O(n^2)$, поэтому в качестве точек $t.lpv$ и $t.rpv$ будем брать точки не с максимальным, но с близким к максимальному расстоянием $|t.lpv-t.rpv|$. Для этого выберем случайным образом точку $p \in V(t)$, затем выберем точку $t.lpv$ как самую дальнюю от p , а точку $t.rpv$ как самую дальнюю от $t.lpv$. Введём вектор $u=t.rpv-t.lpv$. Спроецируем все дескрипторы $V(t)$ на этот вектор. Далее найдём медианную точку A среди проекций. Через точку A построим гиперплоскость L , перпендикулярную вектору u . Плоскость L разделит множество дескрипторов $V(t)$ на два подмножества. Под плоскостью окажутся дескрипторы $V(t.lc)$, а над плоскостью – $V(t.rc)$. Рисунок 2, а демонстрирует процесс деления узла метрического дерева в двумерном пространстве.

Для осуществления процедуры поиска для каждого узла t также понадобится хранить гиперсферу $B(t.center, t.r)$, где $t.center$ – центр гиперсферы, а $t.r$ – её радиус. Основное требование, предъявляемое к данной гиперсфере, – это то, что она должна содержать внутри себя все дескрипторы из

$V(t)$, при этом $t.r$ должен быть по возможности минимальным.

Процедуры построения метрического и sr -дерева отличаются способом деления множества $V(t)$ на множества $V(t.lc)$ и $V(t.rc)$. Процесс деления узла в sr -дереве повторяет аналогичный процесс в метрическом дереве до момента вычисления делящей гиперплоскости L . После вычисления L строятся две дополнительные плоскости, параллельные L : LL на расстоянии τ от L в направлении u и LR на расстоянии τ в направлении $-u$. Далее множество $V(t)$ делится следующим образом: $N(t.lc) = \{x | x \in V(t), d(x, LR) + 2\tau > d(x, LL)\}$, $N(t.rc) = \{x | x \in V(t), d(x, LL) + 2\tau > d(x, LR)\}$, где $d(x, P)$ – Евклидово расстояние от дескриптора x до гиперплоскости P . Процесс деления узла в sr -дереве показан на рис. 2, б.

Если в рамках построения одного дерева дескрипторы в одних узлах делятся по правилам метрического дерева, а в других – по правилам sr -дерева, то такое дерево называют гибридным sr -деревом [17]. Именно гибридным деревьям было отдано предпочтение при реализации процедуры быстрого поиска похожих дескрипторов.

6.2. Поиск дескрипторов в гибридном sr -дереве

Рассмотрим процедуру поиска k дескрипторов, максимально похожих на дескриптор запроса q в гибридном sr -дереве. Дополнительным параметром поиска будет целое число E – максимальное количество листовых узлов, которое следует просмотреть при поиске. Для точного алгоритма поиска $E = \infty$. Для приближённого поиска *best bin first* [14] параметр E может варьироваться от 50 до 200 и бо-

лее. Чем больше E , тем точнее поиск, но тем медленнее работает алгоритм.

Для поиска будем использовать приоритетную очередь Q , изначально пустую. В данную очередь будем заносить узлы дерева. Роль приоритета каждого узла играет наибольшее из двух расстояний: расстояние до границы гиперсферы этого узла V или расстояние до плоскости деления L . Всегда из очереди будет извлекаться узел с минимальным расстоянием. Для обозначения расстояния, назначенного каждому узлу очереди t , будем использовать функцию $d(t)$.

Также для хранения результата поиска введём список P . Этот список предназначен для хранения дескрипторов, близких к q . Изначально список P пуст. Максимальное число элементов в P : $|P|_{max} = k$. Для P определена величина $P.d = \max_{p \in P} |p - q|$. Если $P = \emptyset$, то $P.d = \infty$. Операция добавления дескриптора p в P осуществляется следующим образом: если $|P| < k$, то p добавляется в P , иначе, если $|p - q| < P.d$, из P сначала удаляется дескриптор $w \in P$ такой, что $|w - q| = P.d$, и только затем добавляется p . В случае, если $|P| = k$ и $|p - q| \geq P.d$, содержание P не изменяется.

Алгоритм поиска состоит из двух частей: прямой и обратный поиск.

Алгоритм прямого поиска:

Шаг 1. Присвоить текущему узлу t корневой узел дерева.

Шаг 2. Если узел t – листовый, перейти к шагу 8.

Шаг 3. Если $q < L$ (q лежит под плоскостью деления узла t), перейти к шагу 6.

Шаг 4. Если узел t не имеет области перекрытия ($\tau = 0$), поместить $t.lc$ в Q .

Шаг 5. Присвоить $t = t.rc$, перейти к шагу 2.

Шаг 6. Если узел t не имеет области перекрытия ($\tau = 0$), поместить $t.rc$ в Q .

Шаг 7. Присвоить $t = t.lc$, перейти к шагу 2.

Шаг 8. Поместить дескрипторы p в список P , $p \in V(t)$, $p \neq q$.

Обратный поиск работает, пока очередь Q не будет пуста, и состоит из следующих шагов:

Шаг 1. Присвоить $e = 1$

Шаг 2. Извлечь из Q узел t или закончить алгоритм, если Q пуста или $e \geq E$.

Шаг 3. Если $d(t) \geq P.d$, закончить алгоритм.

Шаг 4. Если t – листовый узел, перейти к шагу 10.

Шаг 5. Если $q < L$ (q лежит под плоскостью деления узла t), перейти к шагу 8.

Шаг 6. Если узел t не имеет области перекрытия ($\tau = 0$), поместить $t.lc$ в Q .

Шаг 7. Присвоить $t = t.rc$, перейти к шагу 4.

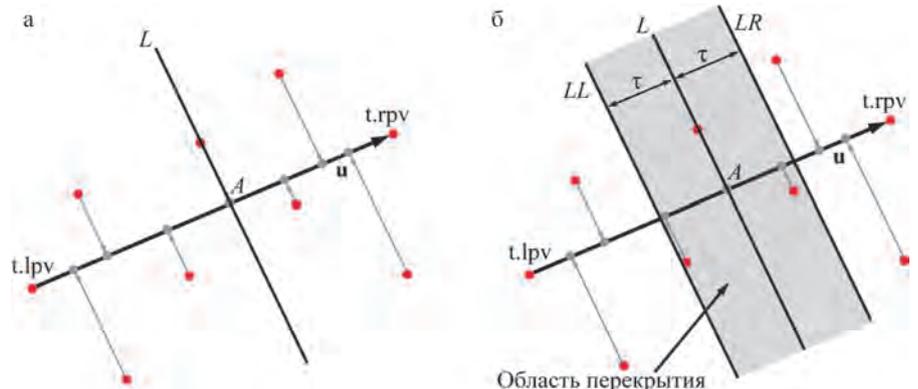


Рис. 2. Деление узла метрического дерева (а) и sr -дерева (б)

Шаг 8. Если узел t не имеет области перекрытия ($\tau=0$), поместить $t.rc$ в Q .

Шаг 9. Присвоить $t=t.lc$, перейти к шагу 4.

Шаг 10. Поместить дескрипторы p в список P , $p \in V(t)$, $p \neq q$.

Шаг 11. Увеличить e на единицу и перейти к шагу 2.

Результатом выполнения прямого и обратного поиска будет список P из k дескрипторов, максимально близких к q . Поскольку номер изображения, к которому принадлежит каждый дескриптор, известен, P легко преобразуется в список номеров изображений, на которых были найдены особенности с дескрипторами, похожими на q . Именно такой список нужно было получить для организации быстрого сопоставления пар в разделе 5.

Обратим внимание на то, от чего зависит эффективность по времени и памяти описанного алгоритма. На этапе прямого и обратного поиска один из дочерних узлов просматриваемого узла t помещается в очередь Q , если t не является узлом с перекрытием. Чем больше очередь, тем дольше длится обратный поиск. С другой стороны, чем больше область перекрытия узла t , тем больше дубликатов одинаковых дескрипторов появится в дереве и тем глубже оно станет и тем больше потребует памяти. Ограничивающим значением для t является максимально допустимое расстояние между похожими дескрипторами. Если присвоить t меньшее значение, то есть риск пропустить близкие к q узлы. Таким образом, необходимо найти баланс между доступной для построения дерева памятью и желаемой скоростью выполнения алгоритма. Если количество дескрипторов невелико, а памяти много, то имеет смысл использовать много узлов с перекрытием

в дереве. Если же дескрипторов много и памяти не хватает, придется пожертвовать производительностью и отказаться от перекрывающихся узлов.

7. Сопоставление пар в больших наборах изображений

Самым существенным ограничением при сопоставлении пар с помощью поиска похожих дескрипторов в гибридных деревьях является оперативная память. Рассмотрим, какой объем памяти необходим для хранения дерева. Для этого перечислим, какую информацию нужно обязательно хранить в каждом узле дерева:

129 коэффициентов делящей плоскости L – 132 байта.

128 координата центра узла (центр гиперсферы B) – 128 байт.

Радиус гиперсферы B – 4 байта.

Количество дескрипторов (для листовых узлов) – 4 байта.

Указатель на индексы дескрипторов (для листовых узлов) – 4 байта.

Флаг, является ли узел узлом с перекрытием – 1 байт.

Указатели на два дочерних узла – 8 байт.

Для хранения координат дескрипторов, которые всегда находятся внутри гиперкуба с единичными сторонами, удобно использовать однобайтовый целый тип данных. В связи с этим координаты центра узла занимают 128 байт. 129 коэффициентов делящей плоскости состоят из 128-размерного вектора – нормали к плоскости, которая занимает 128 байт, и числа с плавающей точкой – расстоянием от плоскости до начала координат, занимающим 4 байта. В сумме это даёт 132 байта на делящую плоскость. В итоге общий объем памяти,

занимаемый каждым узлом дерева, $C_t=281$ байт.

Кроме узлов дерева при поиске в памяти должны храниться массивы дескрипторов и индексов на эти дескрипторы. Использование индексов даёт преимущество в скорости (копирование 4-байтных индексов быстрее, чем 128-байтных дескрипторов) и используемой памяти (узлы с перекрытием дублируют индексы, но не сами дескрипторы). Обозначим размер памяти, занимаемый каждым индексом, через $C_{id}=4$, а размер памяти, занимаемый дескриптором, через $C_d=128$.

Для хранения дерева из n дескрипторов необходимо C_{mem} байт оперативной памяти:

$$C_{mem} = n \left(\frac{2C_t}{l} + C_{id} + C_d \right).$$

Здесь l – минимальное количество дескрипторов, при котором узел дерева не делится (максимальное количество дескрипторов, представленное листовыми узлами). Это выражение получено для случая, когда в дереве отсутствуют перекрывающиеся узлы, то есть для дерева, использующего память максимально эффективно. Выразив из него n , получим количество дескрипторов, которое можно представить в виде дерева, имея C_{mem} оперативной памяти:

$$n = \frac{C_{mem}}{2C_t l^{-1} + C_{id} + C_d}.$$

Имея $C_{mem}=2$ Гб оперативной памяти, доступной в 32-разрядных операционных системах, при $l=10$, можно построить дерево из $n \approx 11.4$ миллиона дескрипторов, что, в свою очередь, хватит для представления 10–20 тысяч изображений. Очевидно, что на 64-разрядной системе ограничения на количество изображений обуславливаются лишь наличием физической памяти, но не ограничением самой системы.

Рассмотрим способ решения задачи сопоставления пар изображений для 32-разрядных операционных систем, имеющих жёсткое ограничение по памяти. Для этого будем использовать дополнительную информацию из навигационной системы АНПА. Этой дополнительной информацией являются приближенные координаты АНПА в момент получения каждого снимка: $x_i = (x_i, y_i)$, $i=1, 2, \dots, m$.

Зная x_i , можно по пространственному признаку разбить множество всех изображений на несколько подмножеств. Разбиение будем производить вдоль осевой линии области исследования (области покрытия снимками), так как вдоль этой линии область имеет наибольшую протяжённость. Осевая линия проходит через центр масс $\bar{x} = (\bar{x}, \bar{y})$ точек x_i :

$$\bar{x} = \frac{1}{m} \cdot \sum_{i=1}^m x_i$$

под углом θ к оси абсцисс:

$$\theta = \frac{1}{2} \cdot \arctan \left(\frac{2\mu_{11}}{\mu_{20} - \mu_{02}} \right).$$

Здесь μ_{kp} – дискретный центральный момент:

$$\mu_{kp} = \sum_{i=1}^m (x_i - \bar{x})^k \cdot (y_i - \bar{y})^p.$$

Подмножества изображений образуются путём деления множества всех изображений прямыми, перпендикулярными к осевой линии области исследований (рис. 3). Гибридные sr-деревья строятся для каждого подмножества поочередно. При этом в каждое дерево помещаются не только дескрипторы особых точек с изображений соответствующего подмножества, но и дескрипторы изображений, находящихся на расстоянии, меньшем, чем r , от границ этого подмножества. Величина r зависит от точности координат x_i . Чем точнее эти координаты,

тем меньше может быть величина r . Если максимальная ошибка в координатах x_i составляет p длин участков дна d , охватываемых диагональю одного изображения, то $r \geq d(p+1)$.

Размеры подмножеств изображений выбираются таким образом, чтобы для каждого подмножества хватило памяти для построения дерева и хранения дескрипторов. Запросы на поиск похожих дескрипторов в каждом дереве выполняются исключительно для дескрипторов особенностей изображений подмножества, для которого это дерево было построено, но не для дескрипторов изображений из r окрестности этого подмножества.

Описанное разбиение имеет ряд преимуществ. Вся задача по-прежнему решается в оперативной памяти, нет необходимости хранить дерево на жёстком диске. Размер области перекрытия r практически не влияет на время работы алгоритма сопоставления, поскольку время построения дерева ничтожно мало по сравнению со временем поиска похожих дескрипторов для всех дескрипторов из подмножества. Каждое подмножество имеет не более двух соседей (соответственно зон перекрытия): справа и слева благодаря тому, что разбиение ведётся вдоль одной оси.

8. Полученные результаты

Применим описанную процедуру поиска и сопоставления пар для набора из 3591 изображения, полученного с АНПА. В этом наборе были найдены 2362523 особенности, дескрипторы которых будем использовать для сопоставления. Также в текущем разделе рассмотрим влияние основных параметров поиска на точность и время сопоставления.

Среди параметров, влияющих на точность и время сопоставления, можно выделить следующие: k (разделы 5, 6), E (раздел 6.2) и η (раздел 5). Также рассмотрим влияние параметра l – максимально допустимого количества дескрипторов особенностей в листовых узлах.

Определим критерии точности и времени поиска, которые будут использоваться при оценке влияния тех или иных параметров на процедуру сопоставления пар. Для этого введём понятия поиска грубой силой. Данный вид поиска заключается в полном переборе и сопоставлении всех возможных пар изображений. Время, затрачиваемое на такой поиск, как и количество обнаруживаемых соответствующих пар изображений, максимально. Поэтому в качестве временной характеристики будем использовать отношение

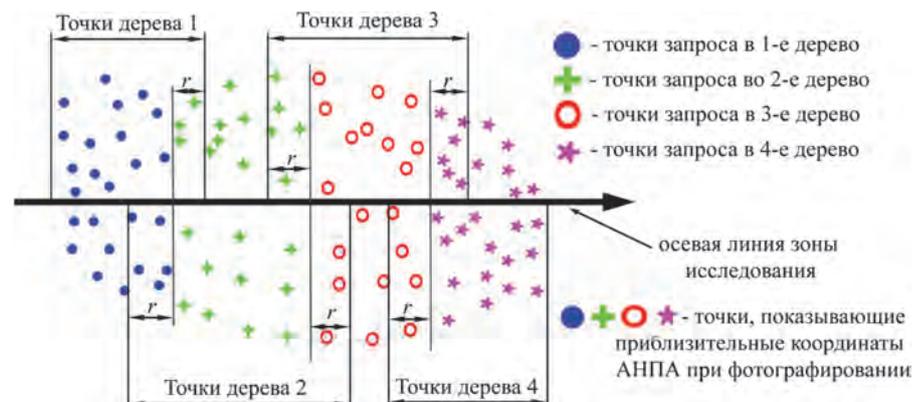


Рис. 3. Деление множества изображений на подмножества

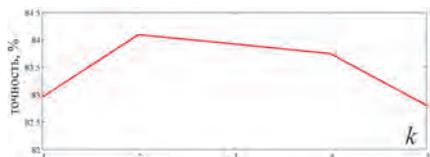


Рис. 4. Зависимость точности сопоставления от k

времени поиска грубой силой k времени поиска описанным алгоритмом. В качестве характеристики точности будем использовать процент найденных пар изображений с помощью описанного алгоритма. За 100% примем количество пар, найденных поиском грубой силой.

Рассмотрим влияние параметра k на точность процедуры сопоставления (рис. 4).

Из графика на рис. 4 видно, что максимальная точность достигается при $k=2$. При дальнейшем увеличении k точность падает. Поскольку влияние параметра k на время процедуры сопоставления незначительно, для всех дальнейших расчётов примем $k=2$.

Параметр l влияет как на точность и время поиска, так и на потребляемую память. От величины параметра l за-

висит глубина гибридного дерева, так как деление узлов дерева прекращается, когда количество дескрипторов в них не превышает l . Поскольку l – максимальное, а не фактическое количество дескрипторов в каждом листовом узле, зависимость точности и времени сопоставления от l носит ступенчатый характер (рис. 5).

Увеличение l приводит к незначительному повышению точности поиска, при этом существенно падает преимущество по времени по сравнению с поиском грубой силой. Вследствие этого предлагается использовать $l=10$. Этого достаточно, чтобы занимаемая деревом память была существенно меньше, чем память, необходимая для хранения дескрипторов. А точность сопоставления будем контролировать другими параметрами, например, E и η .

Параметры E и η одинаково влияют на точность и время сопоставления: чем больше эти параметры, тем выше точность и меньше выигрыш по времени и наоборот. Графики зависимости точности и времени сопо-

ставления от E для 3 значений η представлены на рис. 6.

На рис. 6 также присутствует график, соответствующий непостоянному значению η . Для данного графика рассматривается порог t такой, что для изображения p выбирается такое число $\eta(p)$, что $w_{p,spn(p)} > t$ (раздел 5). Увеличивая значение порога t , можно повышать точность сопоставления и уменьшать время. Порог t является альтернативой фиксированному параметру η .

Стоит отметить, что выигрыш по времени, варьирующийся от 70 до 650, в представленных результатах является характеристикой, зависящей от количества изображений m . Поскольку поиск грубой силой имеет квадратичную вычислительную сложность $O(m^2)$, а описанный алгоритм практически линейную $O(m \times \log_2 m)$, выигрыш по времени будет расти с увеличением числа m .

ЗАКЛЮЧЕНИЕ

В статье была рассмотрена двумерная задача позиционирования изображений, полученных с АНПА. Детально описана подзадача поиска пар соответствующих изображений – ключевая для рассматриваемой задачи. Предложен способ для быстрого обнаружения соответствующих пар изображений.

Данный способ позволяет на обычной современной персональной ЭВМ решать задачу позиционирования для больших наборов изображений за разумное время. Описанный способ использует минимум информации навигационной системы АНПА для выполнения позиционирования. Дополнительная информация о приближённых координатах каждого изображения может существенно увеличить допустимое количество изображений, которое можно обработать на обычном ПК в условиях ограниченной памяти.

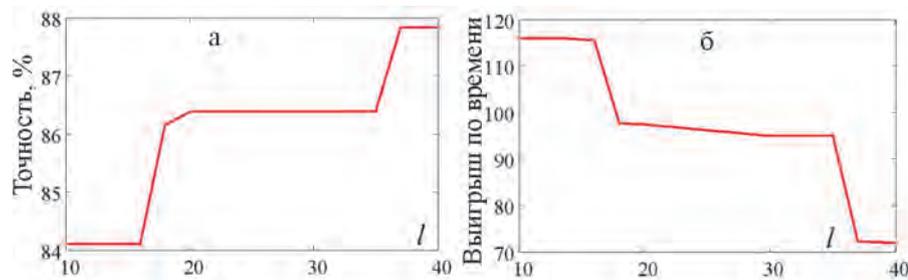


Рис. 5. Зависимость точности (а) и времени (б) сопоставления от l

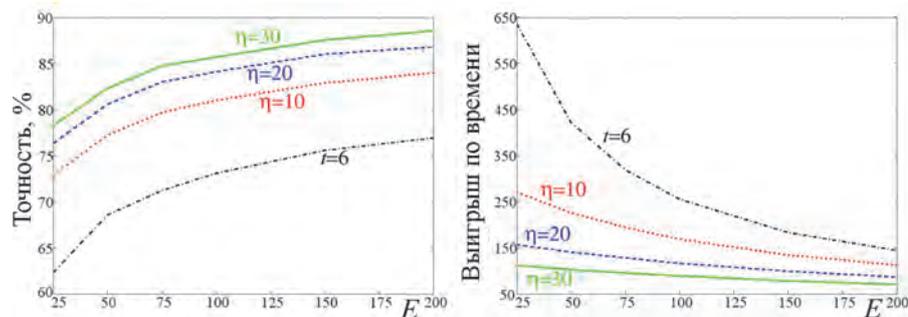


Рис. 6. Зависимость точности (а) и времени (б) сопоставления от E и η

Тестирование алгоритма на наборе подводных снимков, полученных с АНПА, выявило, что для сопоставления одного изображения со всеми изображениями из набора требуется по меньшей мере, 0.5–0.7 секунд. Данные временные интервалы справедливы для 80%-ной точности поиска (по сравнению с полным перебором), параллельного выполнения 4 потоков и 700 особенностей на каждом изображении. Поскольку время сопоставления одного изображения имеет логарифмическую зависимость от количества изображений, приведённые временные рамки охватывают диапазон от 3500 до 100000 изображений. Вместе с тем сравнение двух последовательных изображений при таких же условиях потребует порядка 0.05 секунд, что на порядок меньше, чем 0.5–0.7 секунд. Таким образом, видится перспективным сочетание двух подходов: предложенного в статье и основанного на связанных последовательных изображениях [3–9]. До тех пор пока связи между последовательными изображениями находятся хорошо, предпочтительно использовать стандартный подход, как только появляются разрывы или необходимость замкнуть траекторию, можно воспользоваться предложенным методом. Однако для стандартного подхода есть ещё одно ограничение – каждое новое добавленное изображение уточняет структуру движения, поэтому для точного предсказания пересекающихся пар для новых изображений необходимо периодически корректировать траекторию с учётом добавленных изображений. Процедура коррекции может свести на нет все преимущества по времени для стандартного подхода. В случае, когда процедура периодической коррекции отнимает слишком много времени, в задаче сшивки изображений

предпочтительно использование предложенного подхода.

По-другому обстоят дела с задачами навигации. Поскольку в этой задаче возможен исключительно последовательный доступ к изображениям, а точная траектория в каждый момент времени является целью задачи, комбинирование стандартного и предложенного подхода представляется особенно привлекательным. Пока последовательных снимков достаточно для точного определения координат и на процедуру сопоставления тратится минимальное время, можно использовать оставшуюся вычислитель-

ную мощность на поддержание гибридного *sr*-дерева поиска в актуальном состоянии, чтобы при необходимости можно было им воспользоваться.

Используя предложенный в статье подход, можно добиться надёжного позиционирования изображений независимо от качества перекрытия последовательных изображений, пропусков и однородных зон на дне. Кроме того, данный подход будет полезен и в задаче навигации, когда стандартный подход не может замкнуть траекторию из-за большой накопившейся ошибки.

ЛИТЕРАТУРА

1. Агеев М. Д., Киселёв Л. В., Матвиенко Ю. В. и др. Автономные подводные роботы: Системы и технологии / под общ. ред. М. Д. Агеева. М.: Наука, 2005. – 400 с.
2. Haywood R., Acquisition of a micro scale photographic survey using an autonomous submersible // Proc. of the OCEANS Conference. Washington, 1986. Vol 5. P. 1423–1426.
3. Aguirre F., Boucher J. M., Jacq J. J. Underwater navigation by video sequence analysis // Proc. of the International Conference on Pattern Recognition. Atlantic City, 1990. Vol 2. P. 537–539.
4. Fleischer S. D., Marks R. L., Rock S. M., Lee M. J. Improved Real-Time Video Mosaicking of the Ocean Floor // Proc. of the MTS/IEEE OCEANS Conference. San Diego, 1995. Vol 3. P. 1935–1944.
5. Garcia R., Batlle J., Cufi X., Ama J. Positioning an underwater vehicle through image mosaicking // Proc. of the IEEE International Conference on Robotics and Automation. Seoul, 2001. – Vol 3. P. 2779–2784.
6. Бореико А. А., Воронцов А. В., Кушнерик А. А., Щербатюк А. Ф. Алгоритмы обработки видеоизображений для решения задач управления и навигации автономных необитаемых подводных аппаратов // Подводные исследования и робототехника. 2010. № 1/9. С. 29–39.
7. Gracias N., Negahdaripour S. Underwater mosaic creation using video sequences from different altitudes // Proc. of the MTS/IEEE OCEANS Conference. Washington, 2005. Vol. 2. P. 1295–1300.
8. Gracias N. R., van der Zwaan S., Bernardino A., Santos-Victor J. Mosaic-based navigation for autonomous underwater vehicles // IEEE Journ. of Oceanic Eng. 2003. Vol. 28. № 4. P. 609–624.
9. Eustice R. M., Pizarro O., Singh H. Visually augmented navigation for autonomous underwater vehicles // Journal of Oceanic Engineering. 2008. Vol. 33, № 2. P. 103–122.
10. Lowe D. G. Distinctive image features from scale-invariant keypoints // Int. Journ. of Computer Vision. 2004. Vol. 60, № 2. P. 91–110.
11. Камаев А. Н. Относительное ориентирование снимков низкой контрастности, полученных с АНПА, на основе точечных особенностей // Учёные заметки ТОГУ. 2011. Т. 2, № 2. С. 32–42.
12. Камаев А. Н. Исследование алгоритмов упорядочивания коэффициентов систем линейных алгебраических уравнений, возникающих в задачах компьютерного зрения // Информатика и системы управления. 2013. № 3. С. 32–44.
13. Friedman J. H., Bentley J. L., Finkel R. A. An algorithm for finding best matches in logarithmic expected time // ACM Transactions on Mathematical Software. 1977. Vol. 3, № 3. P. 209–226.
14. Beis J., Lowe D. G. Shape indexing using approximate nearest-neighbor search in high-dimensional spaces // Proc. of the conference on CVPR. Puerto Rico, 1997. P. 1000–1006.
15. Indyk P., Motwani R. Approximate nearest neighbours: towards removing the curse of dimensionality // Proc. of the 30-th ACM Symposium on the STOC. Dallas, 1998. P. 604–613.
16. Ciaccia P., Patella M., Zezula P. M-tree: An efficient access method for similarity search in metric spaces // Proc. of the 23-rd VLDB International Conference. Athens, 1997. P. 426–435.
17. Liu T., Moore A., Gray A., Yang K. An investigation of practical approximate nearest neighbor algorithms // Proc. of the conference on NIPS. Vancouver; Whistler, 2004. P. 825–832.