

## ВИЗУАЛЬНАЯ НАВИГАЦИЯ АВТОНОМНОГО НЕОБИТАЕМОГО ПОДВОДНОГО АППАРАТА С ГЛОБАЛЬНЫМ ПОИСКОМ СВЯЗЕЙ МЕЖДУ ИЗОБРАЖЕНИЯМИ

А.Н. Камаев<sup>1</sup>, Д.А. Карманов<sup>1</sup>

<sup>1</sup> Вычислительный центр ДВО РАН, Хабаровск, Россия

### Аннотация

В статье рассматривается задача навигации автономного необитаемого подводного аппарата. Изображения, получаемые бортовой стереокамерой, используются для построения облаков точек, привязанных к конкретным положениям аппарата. Квантованные дескрипторы точек хранятся в метрическом дереве для организации процедуры быстрого поиска с использованием алгоритма *best bin first*. Связи для нового облака точек ищутся в небольшой группе облаков, которые имеют наибольшее число похожих дескрипторов, сохранённых в дереве. Таким образом, новое облако может быть позиционировано относительно других облаков без какой-либо априорной информации о положении АНПА и точности, с которой это положение известно. Это приводит к повышению надёжности системы навигации, делает систему нечувствительной к пропускам данных, однотонным участкам дна и длинным проходам без пересечений собственной траектории. В статье рассматриваются алгоритмы построения облаков точек, сопоставления этих облаков, построения групп потенциально связанных облаков для ускорения процедуры глобального поиска связей. Также предлагается основной алгоритм навигации, состоящий из трёх выполняемых параллельно подпрограмм: добавление изображений в систему, обновление дерева поиска, оптимизация. Предложенный алгоритм навигации тестируется как на реальных, так и на синтетических данных. Тесты на реальных данных показывают, что траектория может быть построена даже для последовательностей с 60% пропущенных изображений и малым или нулевым перекрытием последовательных снимков. Тесты на синтетических данных демонстрируют, что построенная траектория близка к истинной даже для длительных миссий. Средняя скорость работы навигационной системы составляет 3 кадра в секунду на процессоре среднего ценового диапазона.

**Ключевые слова:** АНПА, SLAM, особые точки, счисление пути, сопоставление изображений.

**Цитирование:** Камаев, А.Н. Визуальная навигация автономного необитаемого подводного аппарата с глобальным поиском связей между изображениями / А.Н. Камаев, Д.А. Карманов // Компьютерная оптика. – 2018. – Т. 42, № 3. – С. 457-467. – DOI: 10.18287/2412-6179-2018-42-3-457-467.

### Введение

Использование фото- и видеоданных с бортовых камер позволяет наиболее точно позиционировать автономный необитаемый подводный аппарат (АНПА) вблизи дна. Существует большое количество методов построения навигационных систем (НС), опирающихся на визуальную информацию. Наиболее простой метод – это счисление пути, он использует найденные трансформации между соседними кадрами видеопоследовательности для вычисления положения АНПА [1]. Этому методу свойственны все недостатки инерциальной НС, но в ряде случаев он позволяет дольше сохранять приемлемую точность.

Наиболее распространённым подходом к визуальной навигации с одновременным построением карты (SLAM – *Simultaneous Localization and Mapping*) является использование расширенного фильтра Калмана (ЕКФ – *Extended Kalman Filter*), когда вектор состояния включает положение и ориентацию АНПА и положение ориентиров (*landmarks*) на карте [2, 3]. Новые ориентиры добавляются по мере получения новых изображений. При повторном посещении уже записанного ориентира неопределённость вектора параметров уменьшается (закрытие разрыва). Для закрытия разрыва используются точечные особенности на изображениях и эллипсоид поиска, ограничиваю-

щий количество потенциально соответствующих ориентиров. Так как матрица ковариаций вектора параметров при использовании ЕКФ хранится в явном виде, эллипсоид поиска всегда можно оценить. Проблемой использования ЕКФ является квадратичный рост сложности задачи по мере добавления новых ориентиров к вектору параметров. Эту проблему можно частично решить, разбив карту на локальные связанные зоны, каждая из которых строится с использованием ЕКФ, тогда как сами зоны позиционируются путём решения задачи минимизации на графе [4].

Другим способом справиться с квадратично растущей сложностью ЕКФ является использование расширенного фильтра информации (EIF – *Extended Information Filter*) [5]. В отличие от ЕКФ, EIF оперирует матрицей информации, обратной к матрице ковариаций. Она является разреженной, что позволяет значительно сократить вычисления, необходимые для обновления вектора состояний и самой матрицы. В [6] показано, как можно оценить ковариации отдельных элементов вектора параметров для решения задачи закрытия разрывов, не вычисляя при этом матрицу, обратную к матрице информации.

Наиболее точным способом решения задачи навигации является подход, основанный на построении графа состояний АНПА, но он применяется значи-

тельно реже из-за сложности реализации в режиме реального времени. В [7] рассматривается оптимизация поиска связей между изображениями при использовании такого подхода. Оптимизация заключается в том, что для поиска отбираются изображения из некоторой окрестности предполагаемого положения только в том случае, если они имеют выраженный и достаточно уникальный текстурный рисунок. В [8] рассматривается подход, основанный на графе применительно к стереоизображениям.

Несмотря на то, что для решения задачи визуальной навигации АНПА могут быть применены самые разные подходы, их все объединяет один важный недостаток – для закрытия разрывов требуется сопоставлять текущее изображение со всеми попадающими в эллипсоид неопределённости положения. Для дна с хорошим текстурным покрытием и при условии большого пересечения последовательных снимков неопределённость положения невелика и для закрытия разрывов достаточно перебрать небольшое число изображений. При однотонной донной поверхности, малом перекрытии, пропуске кадров, длительных проходах без пересечения траектории неопределённость положения возрастает настолько, что задача закрытия разрывов становится нерешаемой. Чтобы иметь возможность закрывать разрывы независимо от знания о положении АНПА и неопределённости этого положения, в статье предлагается использовать глобальный поиск дескрипторов особых точек при сопоставлении изображений. Рассматривается система навигации по стереоизображениям, основанная на графе, использующая глобальный поиск связей между изображениями и способная работать в режиме реального времени.

**Обозначения.** Векторы и матрицы обозначены полужирным шрифтом:  $\mathbf{A}$  – матрица,  $\mathbf{a}$ ,  $\mathbf{a}^T$  – векторы столбец и строка. Для обозначения поворота в  $\mathbb{R}^3$  используется либо матрица поворота  $\mathbf{R}: \mathbb{R}^3 \rightarrow \mathbb{R}^3$ , либо вектор поворота  $\mathbf{r} \in \mathbb{R}^3$ , такой что  $|\mathbf{r}| \leq \pi$  – угол, а  $\mathbf{r}/|\mathbf{r}|$  – ось поворота. Функции  $\Gamma(\mathbf{R})$  и  $R(\mathbf{r})$  отвечают за перевод матрицы поворота в вектор и обратно. Через  $\text{diag}(\mathbf{a})$ ,  $\text{diag}(a_0, a_1, \dots)$  обозначены диагональные матрицы с элементами  $a_0, a_1, \dots$  на диагонали.

### 1. Постановка задачи

На корпусе АНПА установлены две камеры, которые могут быть описаны с использованием модели камеры – обскуры (все нелинейные искажения должны быть устранены заранее). Будем считать, что матрицы внутренних параметров камер  $\mathbf{K}_1^c$  и  $\mathbf{K}_2^c$  известны. Также известны матрицы поворота камер  $\mathbf{R}_1^c$ ,  $\mathbf{R}_2^c$  и их положения  $\mathbf{t}_1^c$  и  $\mathbf{t}_2^c$  в системе координат АНПА. Известны  $\mathbf{t}_0^g$  и  $\mathbf{r}_0^g$  – положение и ориентация АНПА в глобальной системе координат в начальный момент времени. На аппарате работает инерциальная НС, позволяющая определить положение и ориентацию аппарата в момент получения  $k$ -го изображения:  $\mathbf{t}_k^l$ ,  $\mathbf{r}_k^l$ .

Инерциальная НС даёт накапливающуюся со временем ошибку. Обозначим через  $\sigma_t^l$  и  $\sigma_r^l$  векторы среднеквадратических отклонений положений и ориентаций, вычисляемых инерциальной НС за 1 секунду.

Необходимо, используя визуальную информацию с камер, определять положение и ориентацию АНПА в момент получения  $k$ -го изображения  $\mathbf{t}_k^g$ ,  $\mathbf{r}_k^g$ ,  $k = 1, 2, \dots, M-1$ , где  $M$  – количество изображений, получаемых АНПА за миссию. В режиме фотообследования АНПА движется с небольшой скоростью, и для обеспечения достаточного перекрытия снимков требуемая частота съёмки составляет от 0,5 до 2 кадров в секунду. На участках, где удалённость или характер дна не позволяет использовать визуальные особенности, может быть использована инерциальная НС.

### 2. Построение облака точек

На изображениях, получаемых со стереокамер АНПА – стереопарах, выполняется корректировка освещения [9]. Рассмотрим работу с  $k$ -й стереопарой (при обозначении объектов, локальных для  $k$ -й стереопары, будем опускать индекс  $k$ ). На изображениях стереопары извлекаются точечные особенности  $\mathbf{x}_{1l} = (x_{1l,1}, x_{1l,2}, 1)^T$ ,  $\mathbf{x}_{2p} = (x_{2p,1}, x_{2p,2}, 1)^T$  и их SIFT-дескрипторы  $\mathbf{d}_{1l}$ ,  $\mathbf{d}_{2p}$  [10], где  $l = 1, 2, \dots, L$ ;  $p = 1, 2, \dots, P$ ;  $L$  и  $P$  – количество особенностей, обнаруженных на первом и втором изображении соответственно.

#### 2.1. Модель связи

Соответствующие точки первого и второго изображения стереопары  $\mathbf{x}_1$ ,  $\mathbf{x}_2$  связаны эпиполярным ограничением  $\mathbf{x}_2^T \mathbf{F} \mathbf{x}_1 = 0$ , где  $\mathbf{F}$  – фундаментальная матрица [11],

$$\mathbf{F} = \mathbf{K}_2^{c-T} \mathbf{R}_2^{cT} \mathbf{R}_1^c \left[ \mathbf{R}_1^{cT} \mathbf{t}_2^c - \mathbf{R}_1^{cT} \mathbf{t}_1^c \right]_{\times} \mathbf{K}_1^{c-1}.$$

Согласно эпиполярному ограничению точка на втором изображении  $\mathbf{x}_2$ , соответствующая точке  $\mathbf{x}_1$ , лежит на линии, заданной коэффициентами  $\mathbf{F} \mathbf{x}_1$ .

Определим функцию расстояния от точки до прямой

$$d\left((a_0, a_1, a_2)^T, (x_0, x_1, 1)^T\right) = (a_0 x_0 + a_1 x_1 + a_2) / \sqrt{a_0^2 + a_1^2}.$$

Точка первого изображения стереопары с номером  $i$  соответствует точке второго изображения стереопары с номером  $j$ , если выполнены следующие условия:

$$d_{ij} = d(\mathbf{F} \mathbf{x}_{1i}, \mathbf{x}_{2j}) < d_{\max}, \tag{1}$$

$$|\mathbf{d}_{1i} - \mathbf{d}_{2j}| < d_{ij} \frac{m_{1a} - m_{1b}}{d_{\max}} + m_{1b}, \tag{2}$$

$$\frac{|\mathbf{d}_{1i} - \mathbf{d}_{2j}|}{\min_{p,l} (|\mathbf{d}_{1i} - \mathbf{d}_{2p}|, |\mathbf{d}_{1l} - \mathbf{d}_{2j}|)} < d_{ij} \frac{m_{2a} - m_{2b}}{d_{\max}} + m_{2b}, \tag{3}$$

$$p \in \{1, 2, \dots, P\} \setminus \{j\}, \quad l \in \{1, 2, \dots, L\} \setminus \{i\},$$

где  $d_{\max}$ ,  $m_{1a}$ ,  $m_{1b}$ ,  $m_{2a}$ ,  $m_{2b}$  – параметры алгоритма сопоставления. В параграфе 6 использованы следующие значения этих параметров:  $d_{\max} = 2$ ,  $m_{1a} = 0,3$ ,  $m_{1b} = 0,5$ ,  $m_{2a} = 0,8$ ,  $m_{2b} = 0,9$ . Пары индексов всех соответствующи-

ших точек формируют множество  $P = \{(i \in \{1, 2, \dots, L\}, j \in \{1, 2, \dots, P\}) \mid (1) \wedge (2) \wedge (3)\}$ .

### 2.2. Данные облака

Каждая стереопара позволяет сгенерировать облако точек – множество  $B_k$ . Каждая точка  $b \in B_k$  описывается следующими свойствами:  $b.x = (b.x_0, b.x_1, b.x_2)^T$  – положение в системе координат АНПА,  $b.C$  – матрица ковариаций вектора  $b.x$ ,  $b.R$  – матрица, строки которой являются собственными векторами матрицы  $b.C$ ,  $b.s = (b.s_0, b.s_1, b.s_2)^T$  – соответствующий вектор собственных чисел и  $b.d$  – SIFT-дескриптор.

Задача определения координат точки  $b.x$  по проекциям  $x_{1i}$ ,  $x_{2j}$ ,  $(i, j) \in P$  и известным матрицам камеры  $K_k^c (R_k^{cT} \mid -R_k^{cT} t_k^c)$ ,  $k=1,2$ , называется триангуляцией и может быть решена различными способами [11]. Авторы для определения начального приближения использовали direct linear transformation (DLT) метод [11], а для уточнения результата – метод Ньютона.

Матрица

$$b.C = (J_1 \mid J_2) \Sigma_x^{-1} (J_1 \mid J_2)^T$$

где  $J_1 = (J_{1,lp})$ ,  $J_2 = (J_{2,lp})$ ,  $J_{1,lp} = (\partial x_{1i,p} / \partial b.x_i)$ ,  $J_{2,lp} = (\partial x_{2j,p} / \partial b.x_i)$ ,  $l=0, 1, 2$ ,  $p=1, 2$ , а  $\Sigma_x$  – матрица ковариаций вектора  $(x_{1i,1}, x_{1i,2}, x_{2j,1}, x_{2j,2})$ . Для тестирования алгоритма в параграфе 6 выберем  $\Sigma_x = \sigma_x^2 E$ , где  $E$  – единичная матрица, а  $\sigma_x = 2$  пикселя – среднеквадратическое отклонение измерений координат особых точек.

Дескриптор  $b.d$  рассчитывается как средний вектор между дескрипторами соответствующих точек на стереопаре:  $b.d = (d_{1i} + d_{2j})/2$ . Каждой паре сопоставленных точек  $(i, j) \in P$  соответствует одна точка в множестве  $B_k$ . Таким образом,  $|P| = |B_k|$  – количество точек в  $k$ -м облаке.

### 2.3. Компактное представление данных

При большом количестве изображений в системе актуальным становится вопрос хранения данных облаков в оперативной памяти. Большая часть данных может храниться на диске и загружаться при необходимости, но для алгоритма поиска потенциально связанных облаков точек требуется присутствие в памяти дескрипторов  $b.d$ ,  $b \in B_k$ ,  $k=0, 1, \dots, M-1$ .

Эффективным способом уменьшения объема памяти, занимаемого дескрипторами  $b.d$ , является квантование. Компоненты дескрипторов предлагается квантовать по группам (*product quantization*) [12]. Приемлемая точность поиска связанных облаков (параграф 4.1) может быть обеспечена, если разбить вектор SIFT-дескриптора на 16 групп по 8 компонент в каждой. При таком разбиении достаточно словаря из 256 слов для каждой группы. Таким образом, 128-мерный SIFT-дескриптор  $b.d$  может быть представлен 16-мерным вектором  $b.d'$ , каждый компонент которого представлен однобайтным индексом в словарь соответствующей группы. Операция построения словарей может быть выполнена с использованием алгоритма К-средних до выполнения миссии. При этом

для построения выбираются дескрипторы с изображений, типичных для акватории, в которой будет выполняться миссия. Подробно процесс квантования описан в работе [12].

### 3. Вычисление связи между облаками точек

Два облака точек  $B_k$  и  $B_l$  могут содержать некоторое количество пар соответствующих точек (имеющих одинаковое положение в глобальной системе координат). Наличие таких точек позволяет рассчитать преобразование, связывающее точки из облака  $B_l$  с точками из облака  $B_k$ .

#### 3.1. Определение точечных соответствий

Рассмотрим точки  $b_k \in B_k$  и  $b_l \in B_l$ ,  $k \neq l$ . Близость этих точек характеризует функция асимметричного расстояния  $\tilde{d}(b_l.d, b_k.d')$  (ADC) [12]. Асимметричное расстояние, вычисляемое между полным дескриптором и его квантованной версией, согласно [12] точнее аппроксимирует истинное расстояние  $|b_l.d - b_k.d|$ , чем симметричное расстояние, вычисляемое между двумя квантованными дескрипторами.

Для соответствующих точек  $b_k$  и  $b_l$  важно, чтобы каждая из них не образовывала пар с другими точками с похожим расстоянием, т.е. следующая функция должна быть по возможности меньше:

$$d_r(b_k, b_l) = \frac{\tilde{d}(b_l.d, b_k.d')}{\min_{b_j, b_i} (\tilde{d}(b_j.d, b_k.d'), \tilde{d}(b_l.d, b_i.d'))}$$

$$b_j \in B_l \setminus \{b_l\}, b_i \in B_k \setminus \{b_k\}.$$

Множество пар потенциально соответствующих точек  $b_k \in B_k$  и  $b_l \in B_l$

$$B'_{kl} = \{(b_k, b_l) \mid \tilde{d}(b_l.d, b_k.d') < m_{1b} \wedge d_r(b_k, b_l) < m_{2b}\}.$$

#### 3.2. Фильтрация ложных соответствий

Множество  $B'_{kl}$  может содержать ложные соответствия. Применение стандартной процедуры RANSAC [13] для фильтрации ложных соответствий затруднено тем, что для вычисления границ, в рамках которых точка может считаться инлайнером для подобранной модели, требуются затратные по времени вычисления. Отфильтровать ложные соответствия можно опираясь на тот факт, что расстояние между любыми двумя точками одного облака должно быть равно расстоянию между соответствующими точками другого облака. Рассмотрим функцию – счётчик от четырёх точек  $(b_k, b_l), (c_k, c_l) \in B'_{kl}$

$$I(b_k, c_k, b_l, c_l) = \begin{cases} 1, & \Lambda(b_k, c_k, b_l, c_l) \leq \Delta(b_k, c_k, b_l, c_l), \\ 0, & \Lambda(b_k, c_k, b_l, c_l) > \Delta(b_k, c_k, b_l, c_l), \end{cases}$$

$$\Lambda(b_k, c_k, b_l, c_l) = \|b_k.x - c_k.x - |b_l.x - c_l.x|\|,$$

$$\Delta(b_k, c_k, b_l, c_l) = \delta\left(b_k, c_k, \frac{b_k.x - c_k.x}{|b_k.x - c_k.x|}\right) +$$

$$+ \delta\left(b_l, c_l, \frac{b_l.x - c_l.x}{|b_l.x - c_l.x|}\right),$$

$$\delta(b, c, \mathbf{v}) = t(b, \mathbf{v}) + t(c, \mathbf{v}),$$

где  $t(b, \mathbf{v})$  – расстояние вдоль  $\mathbf{v}$  от точки  $b$  до точки пересечения с границей эллипсоида ковариаций вектора  $b \cdot \mathbf{x}$ , построенного на расстоянии в  $s$  раз превышающем дисперсию  $b \cdot \mathbf{x}$  (предположив нормальное распределение ошибки в определении  $b \cdot \mathbf{x}$ , согласно правилу трёх сигм, выберем  $s = 3^2$ ). Функцию  $t(b, \mathbf{v})$ ,  $|\mathbf{v}| = 1$ , можно определить, подставив точку  $t(b, \mathbf{v}) \cdot b \cdot \mathbf{Rv}$  в уравнение эллипсоида и выразив

$$t(b, \mathbf{v}) = \frac{\sqrt{s \cdot b \cdot s_0 \cdot b \cdot s_1 \cdot b \cdot s_2 \cdot \mathbf{g} \cdot \text{diag}(b \cdot \mathbf{Rv}) b \cdot \mathbf{Rv}}}{\mathbf{g} \cdot \text{diag}(b \cdot \mathbf{Rv}) b \cdot \mathbf{Rv}},$$

где  $\mathbf{g} = (b \cdot s_1 \cdot b \cdot s_2, b \cdot s_0 \cdot b \cdot s_2, b \cdot s_0 \cdot b \cdot s_1)$ . Используя функцию  $I(b_k, c_k, b_l, c_l)$ , можно для каждой пары соответствующих точек определить количество других пар точек, расстояния до которых в обоих облаках одинаково:

$$n(b_k, b_l) = \sum_{(c_k, c_l) \in B'_{kl}} I(b_k, c_k, b_l, c_l), \quad (b_k, b_l) \in B'_{kl}.$$

Введём множество

$$B_{kl}(N) = \{(b_k, b_l) \in B'_{kl} \mid n(b_k, b_l) \geq N\},$$

тогда множество истинных соответствий

$$B_{kl} = B_{kl}(\min_{N \geq |B_{kl}(N)|} N).$$

Рассмотрим зависимость  $|B_{kl}|$  от  $|B'_{kl}|$  для связанных облаков точек и несвязанных, где точечные соответствия определены случайным образом. Рис. 1 отображает зависимость, построенную для 20000 связанных и 20000 несвязанных облаков. Из рис. 1 видно, что эти множества могут быть разделены прямой линией.

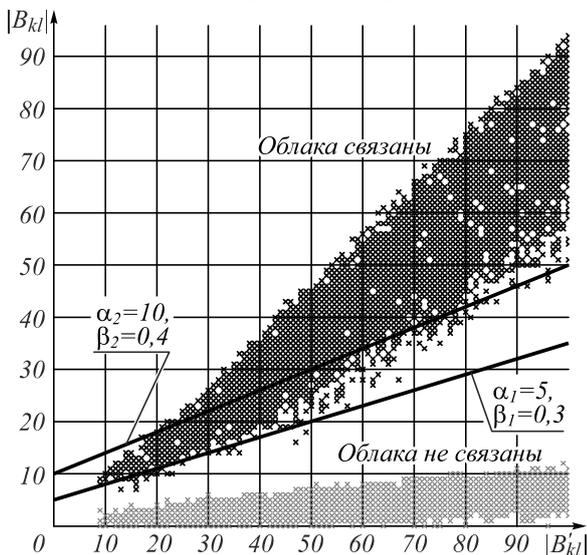


Рис. 1. Соотношение истинных и ложных соответствий для связанных и несвязанных облаков точек

Факт наличия связи можно установить, если выполняется условие  $f(\alpha_1, \beta_1) = |B_{kl}| - \alpha_1 - \beta_1 |B'_{kl}| > 0$ , где  $\alpha_1 = 5$  и  $\beta_1 = 0,3$  (такие значения позволяют отфильтровать ложные и неустойчивые связи). Надёжность связи опишем величиной

$$V_{kl} = \max\left(0,05, \min\left(1, \frac{f(\alpha_1, \beta_1)}{f(\alpha_1, \beta_1) - f(\alpha_2, \beta_2)}\right)\right) \times \min\left(1, \frac{|B_{kl}|}{K_s}\right),$$

где  $K_s$  – минимальное количество точек, соответствующее надёжной связи. В параграфе 6 используются следующие значения:  $\alpha_2 = 10$  и  $\beta_2 = 0,4$  и  $K_s = 25$ .

### 3.3. Модель связи

Для  $(b_k, b_l) \in B_{kl}$  справедливо соотношение  $b_k \cdot \mathbf{x} = \mathbf{R}_{kl} b_l \cdot \mathbf{x} + \mathbf{t}_{kl}$ ,

где  $\mathbf{t}_{kl}$  – вектор смещения между облаками, а  $\mathbf{R}_{kl}$  – матрица поворота. Чтобы найти  $\mathbf{R}_{kl}$ , воспользуемся алгоритмом Кабша [14]. Для этого рассчитаем:

$$\bar{\mathbf{x}}_k = \frac{1}{|B_{kl}|} \sum_{(b_k, b_l) \in B_{kl}} b_k \cdot \mathbf{x}, \quad \bar{\mathbf{x}}_l = \frac{1}{|B_{kl}|} \sum_{(b_k, b_l) \in B_{kl}} b_l \cdot \mathbf{x},$$

$$\mathbf{A} = \sum_{(b_k, b_l) \in B_{kl}} (b_k \cdot \mathbf{x} - \bar{\mathbf{x}}_k)(b_l \cdot \mathbf{x} - \bar{\mathbf{x}}_l)^T.$$

Если  $\mathbf{A} = \mathbf{UDV}^T$  – сингулярное разложение матрицы  $\mathbf{A}$ , то матрица  $\mathbf{R}_{kl} = \mathbf{U} \text{diag}(1, 1, \det(\mathbf{UV}^T)) \mathbf{V}^T$ , а вектор  $\mathbf{t}_{kl} = \bar{\mathbf{x}}_k - \mathbf{R}_{kl} \bar{\mathbf{x}}_l$ . Хранить поворот будем в виде вектора поворота  $\mathbf{r}_{kl} = \mathbf{r}(\mathbf{R}_{kl})$ .

### 3.4. Уточнение параметров связи

Для пары точек  $(b_k, b_l) \in B_{kl}$  определим вектор  $\boldsymbol{\varepsilon}_{b_k, b_l}(\mathbf{r}_{kl}, \mathbf{t}_{kl}) =$

$$= \left( (\mathbf{R}(\mathbf{r}_{kl}) b_l \cdot \tilde{\mathbf{x}} + \mathbf{t}_{kl} - b_k \cdot \mathbf{x})^T, (b_l \cdot \tilde{\mathbf{x}} - b_l \cdot \mathbf{x})^T \right),$$

где  $b_l \cdot \tilde{\mathbf{x}}$  – истинное положение точки  $b_l$ . Допуская нормально распределённую погрешность в определении  $b_k \cdot \mathbf{x}$  и  $b_l \cdot \mathbf{x}$  с соответствующими матрицами ковариаций  $b_k \cdot \mathbf{C}$  и  $b_l \cdot \mathbf{C}$ , запишем ошибку связи

$$E_{B_{kl}}(\mathbf{r}_{kl}, \mathbf{t}_{kl}) = \sum_{(b_k, b_l) \in B_{kl}} \boldsymbol{\varepsilon}_{b_k, b_l}(\mathbf{r}_{kl}, \mathbf{t}_{kl}) \mathbf{W}_{b_k, b_l} \boldsymbol{\varepsilon}_{b_k, b_l}(\mathbf{r}_{kl}, \mathbf{t}_{kl})^T,$$

$$\mathbf{W}_{b_k, b_l} = \text{diag}(b_k \cdot \mathbf{C}^{-1}, b_l \cdot \mathbf{C}^{-1}).$$

Уточнить параметры связи  $\mathbf{r}_{kl}$  и  $\mathbf{t}_{kl}$  можно, минимизировав ошибку  $E_{B_{kl}}(\mathbf{r}_{kl}, \mathbf{t}_{kl})$  относительно параметров  $\mathbf{r}_{kl}, \mathbf{t}_{kl} \in \mathbb{R}^3$  и координат точек  $b_l \cdot \tilde{\mathbf{x}} \in \mathbb{R}^3$ ,  $(b_k, b_l) \in B_{kl}$ . Такую задачу можно решить, используя метод bundle adjustment (BA) [11], выбрав в качестве начального приближения параметры связи, рассчитанные в подпараграфе 3.3 и точки  $b_l \cdot \tilde{\mathbf{x}} = b_l \cdot \mathbf{x}$ . Далее под параметрами связи  $\mathbf{p}_{kl} = (\mathbf{r}_{kl}^T, \mathbf{t}_{kl}^T)$  будем понимать параметры, уточненные в результате решения задачи минимизации. На заключительной итерации ВА рассчитывается матрица  $\boldsymbol{\Sigma}_{\mathbf{p}_{kl}}$  – матрица ковариаций вектора  $\mathbf{p}_{kl}$ .

Для повышения скорости вычислений и уменьшения машинной ошибки вместо матрицы  $\boldsymbol{\Sigma}_{\mathbf{p}_{kl}}^{-1}$  в качестве веса связи будем использовать вектор  $\mathbf{w}_{kl} = (w_{kl,i})$ ,

$$w_{kl,i} = V_{kl} \min\left(\frac{\bar{w}_i}{\sqrt{\boldsymbol{\Sigma}_{\mathbf{p}_{kl}, ii}}}, 1\right), \quad i = 0, 1, \dots, 5, \quad (4)$$

где  $\Sigma_{p_i,ii}$  – диагональный элемент  $\Sigma_{p_i}$ ,  $V_{kl}$  определена в подпараграфе 3.2, а  $\bar{w}_i$  – среднеквадратическое отклонение  $i$ -го параметра связи, при котором она может считаться надёжной. Возьмем  $\bar{w}_{0,1,2} = 10^{-3}$  радиан и  $\bar{w}_{3,4,5} = 10^{-2}$  метров для компонент, соответствующих вектору поворота и вектору перемещения.

**4. Поиск потенциально связанных облаков**

Для поиска связей в режиме реального времени для каждого нового облака точек необходимо сформировать группу из  $M' \in \{10, 11, \dots, 30\}$  облаков, потенциально связанных с ним. Потенциально связанными будем считать облака, имеющие наибольшее количество точек с похожими дескрипторами.

**4.1. Структура данных для хранения дескрипторов точек**

Для быстрого поиска похожих дескрипторов точек (задача поиска  $k$  ближайших соседей) будем использовать метрическое дерево [15] совместно с алгоритмом приближённого поиска best bin first [16, 17]. Алгоритм best bin first позволяет сократить время поиска, ограничив количество просматриваемых листовых узлов дерева до константы  $E$ . Квантование дескрипторов (подпараграф 2.3) позволяет ускорить поиск за счёт быстрого вычисления расстояния между дескрипторами. Рассмотрим, насколько эффективно работает структура данных (СД) при поиске среди  $10^6$  дескрипторов точек морского дна – рис. 2. Будем искать  $k=10$  ближайших соседей и оценивать количество дескрипторов  $\bar{k}$ , входящих в 10 ближайших, но не найденных при поиске. Параметр  $G$  определяет количество групп квантования дескриптора (подпараграф 2.3). Из рис. 2 видно, что  $\bar{k}$  резко падает при переходе от 8 до 16 групп, а далее снижается медленнее. Значение  $G=16$  является хорошим компромиссом между объемом занимаемой памяти и точностью поиска. Увеличение  $E$  линейно увеличивает время, при этом значительное повышение точности поиска наблюдается в интервале значений [50; 100]. Параметр  $L$  определяет количество дескрипторов в листовых узлах дерева. Параметры  $E$  и  $L$  не влияют на потребляемую память, поэтому могут выбираться исходя из требуемого соотношения точности и времени поиска. В параграфе 6 используются значения  $E=100, L=20$ .

**4.2. Обновление СД**

Первое дерево поиска строится лишь тогда, когда в СД накопится достаточно большое количество дескрипторов, например, более 5000, до этого поиск выполняется полным перебором. Когда дерево построено, новые дескрипторы добавляются в него посредством записи в соответствующий листовой узел. Если количество дескрипторов в листовом узле превышает  $L$ , то он помечается не листовым и делится на два новых листовых узла гиперплоскостью, перпендикулярной направлению максимальной дисперсии дескрип-

торов таким образом, чтобы количество дескрипторов в новых узлах отличалось не более чем на один.

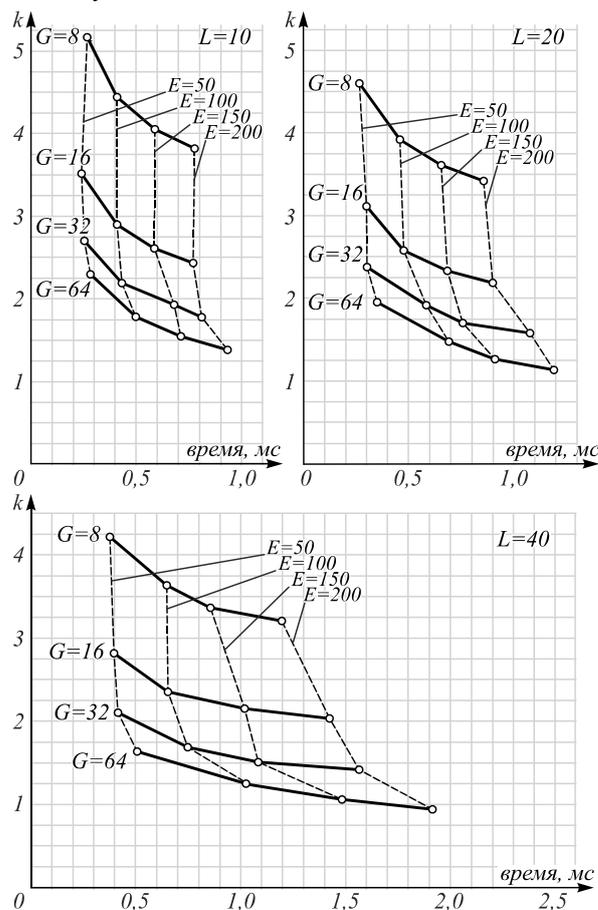


Рис. 2. График зависимости количества найденных дескрипторов от времени при поиске 10 ближайших соседей в базе данных из  $10^6$  SIFT-дескрипторов (результаты усреднены для 10000 запросов)

Для сохранения оптимальной глубины дерево периодически перестраивается. Запишем алгоритм обновления СД (АОСД). Через  $N$  обозначим количество дескрипторов в СД, через  $avg\_d$  – усреднённую максимальную глубину посещаемых листьев по 1000 последних запросов.

- Шаг 1:** Если  $N < 5000$ , или новое дерево строится в настоящий момент, или  $avg\_d - \lfloor \log_2(N/L) + 1 \rfloor < 2$ , то закончить алгоритм.
- Шаг 2:** Присвоить  $N' \leftarrow N$ , построить новое дерево для  $N'$  дескрипторов, во время построения поисковые запросы обрабатывает старая СД.
- Шаг 3:** Пока  $N' \neq N$ , добавлять в новое дерево дескриптор с номером  $N'$  (предполагается нумерация с нуля) и увеличивать  $N'$ :  $N' \leftarrow N' + 1$ .
- Шаг 4:** Заблокировать возможность добавления дескрипторов в СД, повторить шаг 3, заменить прошлую СД новым деревом, разблокировать возможность добавления дескрипторов в СД.

АОСД выполняется в отдельном потоке и блокирует работу НС лишь на шаге 4. Для организации ал-

горитма удобно объединить квантованные дескрипторы всех облаков в непрерывном массиве данных. Тогда любой дескриптор в системе может быть найден по номеру  $i$ :  $\mathbf{d}'_i$ , где  $i=0, 1, \dots, N-1$ , а  $N$  – количество дескрипторов в системе. Дерево хранит в листовых узлах лишь индексы дескрипторов, а облака точек хранят указатели на группу дескрипторов облака в массиве. СД также хранит номера облаков, которым принадлежат дескрипторы –  $\text{img}(i)$ .

**4.3. Алгоритм поиска потенциально связанных облаков точек**

Пусть в навигационную систему добавлено  $M$  изображений с номерами  $m=0, 1, \dots, M-1$ . Для нового изображения с номером  $M$  требуется найти  $M' \leq M$  изображений, потенциально с ним связанных. Введём алгоритм построения вспомогательного множества  $K(\mathbf{d})$  для некоторого дескриптора  $\mathbf{d}$ :

- Шаг 1:** Присвоить  $K(\mathbf{d}) \leftarrow \emptyset$ .
- Шаг 2:** Для всех  $\mathbf{d}'_i, i \in S(\mathbf{d}) \subseteq \{0, 1, \dots, N-1\}$ , где  $S(\mathbf{d})$  – подмножество индексов дескрипторов, близких к  $\mathbf{d}$ , определяемое СД: если  $\exists k \in K(\mathbf{d}), \text{img}(i) = \text{img}(k.n)$ , то выполнять шаг 3а, иначе – шаг 3б.
- Шаг 3а:** Если  $k.d_1 > \tilde{d}(\mathbf{d}, \mathbf{d}'_i)$ , то присвоить:  
 $k.d_2 \leftarrow k.d_1, k.d_1 \leftarrow \tilde{d}(\mathbf{d}, \mathbf{d}'_i), k.n \leftarrow i$ ,  
 иначе,  
 если  $k.d_2 > \tilde{d}(\mathbf{d}, \mathbf{d}'_i)$ ,  
 то присвоить  $k.d_2 \leftarrow \tilde{d}(\mathbf{d}, \mathbf{d}'_i)$ .
- Шаг 3б:** Присвоить:  $l.d_1 \leftarrow \tilde{d}(\mathbf{d}, \mathbf{d}'_i), l.d_2 \leftarrow \sqrt{2}, l.n \leftarrow i$ .  
 Если  $|K(\mathbf{d})| < 10$ , то  $K(\mathbf{d}) \leftarrow K(\mathbf{d}) \cup \{l\}$ ,  
 иначе:  
 $p = \arg \max_{k \in K(\mathbf{d})} k.d_1$ ,  
 если  $l.d_1 < p.d_1$ ,  
 то  $K(\mathbf{d}) \leftarrow K(\mathbf{d}) \setminus \{p\} \cup \{l\}$ .

Алгоритм поиска потенциально связанных облаков точек (АППСОТ):

- Шаг 1:** Присвоить  $v_m \leftarrow 0$ .
- Шаг 2:** Для всех  $b \in B_M$  и  $k \in K(b.\mathbf{d})$  присвоить  

$$v_{\text{img}(k.n)} \leftarrow v_{\text{img}(k.n)} + w_a(k.d_1)w_b \left( \frac{k.d_1}{k.d_2} \right).$$

- Шаг 3:** Построить такое множество индексов  $I(M) \subseteq I = \{0, 1, \dots, M-1\}$ , что  

$$\min_{i \in I(M)} v_i \geq \max_{k \in I \setminus I(M)} v_k \wedge |I(M)| = M'.$$
 $I(M)$  – множество индексов изображений, потенциально связанных с изображением с номером  $M$ . Задача весовых функций  $w_a(x)$  и  $w_b(x)$  – назначить больший вес дескрипторам, имеющим большие шансы соответствовать дескриптору запроса. В статье использованы следующие весовые функции:

$$w_a(x) = e^{-5x^2/9}, \quad w_b(x) = \min \left( 1, \left( 1 - \frac{x-0,7}{0,3} \right)^2 \right).$$

**5. Алгоритм навигации**

Основной операцией алгоритма навигации является поиск состояния АНПА  $\mathbf{p}_l^g = ((\mathbf{r}_l^g)^T, (\mathbf{t}_l^g)^T)^T$  в момент получения  $l$ -го изображения на основе обнаруженной связи  $\mathbf{p}_{kl} = (\mathbf{r}_{kl}^T, \mathbf{t}_{kl}^T)$  (подпараграф 3.4),  $k < l$ :  $\mathbf{p}_l^g = \mathbf{f}(\mathbf{p}_k^g, \mathbf{p}_{kl})$ , где

$$\mathbf{f}(\mathbf{p}_k^g, \mathbf{p}_{kl}) = \left( \mathbf{r}(\mathbf{R}(\mathbf{r}_k^g)\mathbf{R}(\mathbf{r}_{kl}))^T, (\mathbf{R}(\mathbf{r}_k^g)\mathbf{t}_{kl} + \mathbf{t}_k^g)^T \right).$$

НС поддерживает список связей изображений  $L$ , в момент запуска  $L = \emptyset$ . Запишем алгоритм, выполняемый НС при добавлении  $l$ -го изображения,  $l > 0$  (алгоритм добавления изображения – АДИ):

- Шаг 1:** Вычислить  $B_l$  (параграф 2), построить  $I(l)$  (подпараграф 4.3, АППСОТ).
- Шаг 2:**  $I'(l) \leftarrow \emptyset$ , для  $k \in I(l)$ , если есть связь между облаками  $B_k$  и  $B_l$ : рассчитать  $\mathbf{p}_{kl}$  и  $\mathbf{w}_{kl}$  (4) (параграф 3), присвоить  $I'(l) \leftarrow I'(l) \cup \{k\}$ .
- Шаг 3:** Если  $l-1 \notin I'(l)$ , построить связь  $\mathbf{p}_{l-1,l} = (\mathbf{r}_{l-1,l}^T, \mathbf{t}_{l-1,l}^T)$ , где  $\mathbf{r}_{l-1,l} = \mathbf{r}(\mathbf{R}(\mathbf{r}_{l-1}^l)^T \mathbf{R}(\mathbf{r}_l^l))$ , а  $\mathbf{t}_{l-1,l} = \mathbf{R}(\mathbf{r}_{l-1}^l)^T (\mathbf{t}_l^l - \mathbf{t}_{l-1}^l)$ , рассчитать  $\mathbf{w}_{kl}$  (4), взяв  $V_{l-1,l} = 1$  и  $\Sigma_{\mathbf{p}_{l-1,l}} = \text{diag}(\Delta_{l-1,l} \sigma_r^T, \Delta_{l-1,l} \sigma_t^T)^2$ , где  $\Delta_{l-1,l}$  – время между получением изображений  $l-1$  и  $l$ , присвоить  $I'(l) \leftarrow I'(l) \cup \{l-1\}$ .
- Шаг 4:** Рассчитать  $\mathbf{p}_l^k = (\mathbf{r}_l^{kT}, \mathbf{t}_l^{kT}) = \mathbf{f}(\mathbf{p}_k, \mathbf{p}_{kl})$ ,  $k \in I'(l)$ .  
 Относительно параметров  $\mathbf{r}_l^g, \mathbf{t}_l^g \in \mathbb{R}^3$  решить задачу  $\sum_k \epsilon_l^k \text{diag}(\mathbf{w}_{kl}) \epsilon_l^{kT} \rightarrow \min$ ,  
 где  $\epsilon_l^k = (\mathbf{r}(\mathbf{R}(\mathbf{r}_l^k)^T \mathbf{R}(\mathbf{r}_l^k)))^T, (\mathbf{t}_l^g - \mathbf{t}_l^k)^T$ .
- Шаг 5:** Рассчитать  $D_l = \max_{k \in I'(l)} |\mathbf{t}_l^g - \mathbf{t}_l^k|$ .
- Шаг 6:** Дополнить список связанных изображений:  $L \leftarrow L \cup \{(l, k) | k \in I'(l)\}$ .
- Шаг 7:** Заменить дескрипторы точек облака  $B_l$  на квантованные (подпараграф 2.3) и поместить их в СД, выполнить АОСД (подпараграф 4.2).

Рассчитанная на шаге 5 АДИ величина  $D_l$  отслеживается процессом, работающим параллельно и отвечающим за корректировку траектории в случае, когда это необходимо согласно следующему алгоритму (алгоритм корректировки траектории – АКТ):

- Шаг 1:** Бездействовать, пока  $D_i \leq D_{\max}, i = 1, 2, \dots, M-1$ , где  $M$  – количество изображений в системе, а  $D_{\max}$  – допустимая погрешность связи.
- Шаг 2:** Блокировать АДИ, присвоить  $M' \leftarrow M$  и  $L' \leftarrow \{(i, j) \in L | j \geq k\}$ , где  $k = \min_{D_i > D_{\max}, (i, j) \in L} j$ ,  
 разблокировать АДИ.

**Шаг 3:** Относительно параметров  $\mathbf{p}_i^g \in \mathbb{R}^6$ ,

$$i = k + 1, k + 2, \dots, M' - 1 \text{ решить задачу минимизации}$$

$$\sum_{(i,j) \in L'} \boldsymbol{\varepsilon}_{ij}^T \text{diag}(\mathbf{w}_{kl}) \boldsymbol{\varepsilon}_{ij} \rightarrow \min$$

$$\boldsymbol{\varepsilon}_{ij} = \begin{pmatrix} \mathbf{r}(\mathbf{R}(\mathbf{r}_i^g)^T \mathbf{R}(\mathbf{r}_j^g) \mathbf{R}(\mathbf{r}_{ji})) \\ (\mathbf{R}(\mathbf{r}_j^g)^T (\mathbf{t}_i^g - \mathbf{t}_j^g) - \mathbf{t}_{ji}) \end{pmatrix}.$$

**Шаг 4:** Блокировать АДИ, применить решение задачи шага 3, пока  $M' \neq M$ : рассчитать  $\mathbf{p}_{M'}^g$  (шаг 4, АДИ), присвоить  $M': M' \leftarrow M' + 1$ . Разблокировать АДИ.

Задача минимизации на шаге 3 может быть сформулирована различным образом, в зависимости от того, какие данные навигационная система АНПА определяет точно (часто это может быть глубина, курс, крен и дифферент), а какие требуют уточнения. Подобная задача может быть эффективно решена, если учитывать разреженность матрицы системы. В работе [18] представлены несколько подходов к решению подобной задачи.

**6. Результаты**

Рассмотрим работу НС на трёх наборах данных: набор из 3591 изображения с АНПА «Клавесин-1Р» и два набора по 11000 и 10006 изображений виртуальной акватории.

Данные с АНПА «Клавесин-1Р» обладают рядом особенностей: 3591 снимок покрывает лишь отдельные участки траектории, в общей сложности не более 40%, последовательные снимки получены с интервалом времени в 2 секунды, что при средней скорости 0,7 м/с приводит к малому или отсутствующему перекрытию, дно на снимках вымощено однотипными панцирями ископаемых морских ежей – рис. 3.



Рис. 3. Пример снимков с АНПА «Клавесин-1Р»

АНПА «Клавесин-1Р» не обладает стереокамерой, поэтому облака точек для каждого изображения строятся на основе данных о расстоянии АНПА до дна в момент получения снимка и предположения о плоском дне в рамках получаемого снимка. На рис. 4а изображена траектория АНПА, полученная комплексированной инерциальной и гидроакустической НС. В режиме постобработки была смоделирована работа предложенной НС – рис. 4б. Связи между изображениями показаны черными линиями: сплошными – связи, полученные в результате сопоставления изображений, пунктирными – связи на основе данных инерциальной и гидроакустической НС. Рассмотрим вектор ошибки связи – вектор с началом в положении АНПА, предсказанном связью, и концом в положении, рассчитанном НС. Векторы ошибок показаны серыми линиями на рис. 4. Точность работы НС можно оценить с помощью длин векторов ошибок связей.

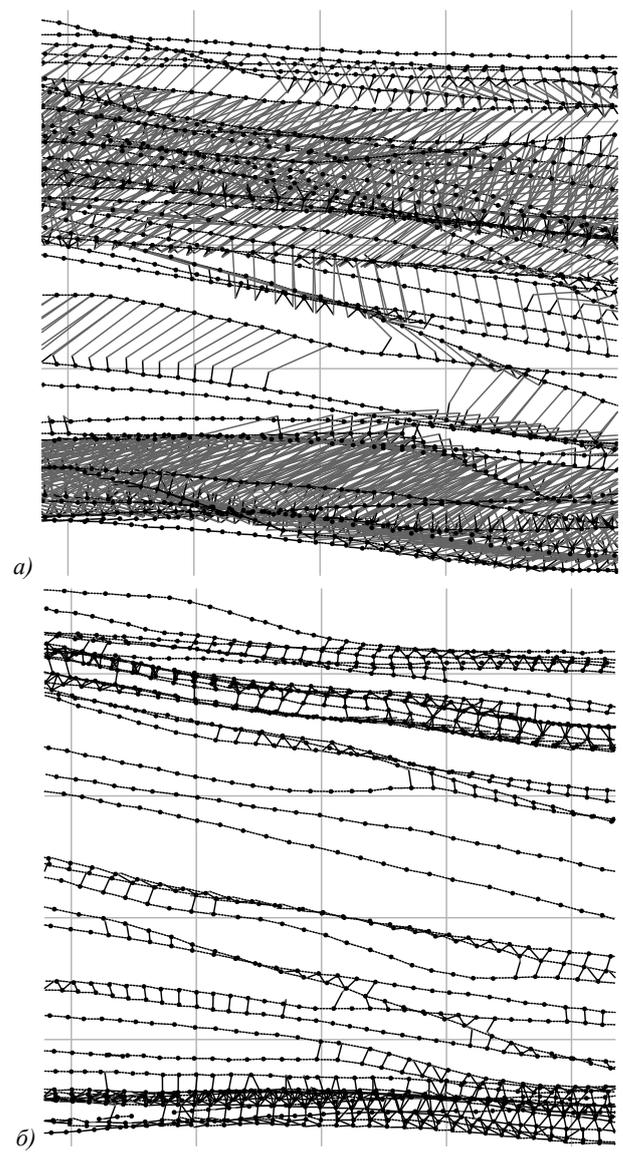


Рис. 4. Участки траектории АНПА «Клавесин-1Р»: полученной инерциальной НС с коррекцией по гидроакустическим маякам (а), полученной с использованием визуальной навигационной системы (б). Размерность сетки – 10×10 метров

Рассмотрим связи, соединяющие непоследовательные изображения (т.е. связи, закрывающие разрывы). В общей сложности было найдено 3179 таких связей. Длины векторов ошибок этих связей варьировались от  $8 \cdot 10^{-4}$  до 0,96 м, средняя длина составила 0,07 м, а среднеквадратическое отклонение – 0,08 м. Если оценить траекторию АНПА, полученную без использования визуальной НС, с точки зрения этих же связей, то длины векторов их ошибок попадут в диапазон от 0,13 до 24,65 м, средняя длина составит 6,69 м, а среднеквадратическое отклонение – 4,18 м.

При построении траектории с помощью предложенной НС удалось достичь скорости 3 кадра в секунду на процессоре Intel Core Quad Q9550, что в шесть раз превысило скорость получения изображений в рассматриваемом примере. При этом разрешенные снимков составляло 768 × 576 пикселей.

Для получения синтетических данных использовался алгоритм построения дна [19] и симулятор [20]. Стереозображения с разрешением  $640 \times 480$  и стереобазой 0,4 м были получены с помощью бортовой камеры виртуального АНПА – рис. 5. Синтетические данные позволяют сопоставить траекторию АНПА, рассчитываемую НС, с истинной траекторией, которая остаётся неизвестной при экспериментах в реальных акваториях.



Рис. 5. Пример снимков с бортовой камеры виртуального АНПА

В первом эксперименте в виртуальной акватории АНПА выполнял движение по окружности с радиусом 100 метров со смещающимся центром. На рис. 6 представлена траектория длиной 33 км, построенная НС при движении аппарата.

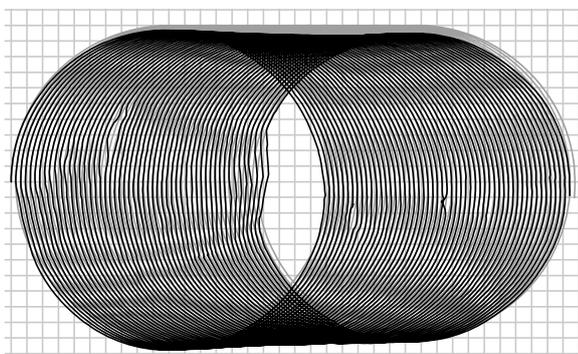


Рис. 6. Движение АНПА по окружности со смещающимся центром. Чёрной линией показана рассчитанная траектория АНПА, а серой – истинная. Сетка имеет размерность  $10 \times 10$  м

В общей сложности для 11000 изображений в ходе работы НС было обнаружено 94114 связей. При реализации АКТ использовались данные о глубине аппарата, поскольку стереобазы в 0,4 м не способна обеспечить приемлемую точность траектории по высоте. Из рис. 6 видно, что рассчитанная траектория (чёрная линия) близка к истинной (серая линия), а графики ошибки НС представлены на рис. 7а, б.

Рис. 7а показывает график зависимости ошибки НС от пройденного расстояния. Зависимость носит циклический характер, так как ошибка опускается к минимальным значениям каждый раз, когда АНПА проходит недалеко от начала координат. График на рис. 7б отображает зависимость ошибки НС от расстояния до начала координат. Данные усреднены по каждому метру расстояния.

Для второго теста в синтетической акватории была выбрана сложная траектория, чередующая зоны с полным покрытием дна и зоны прямолинейного движения – рис. 8. Траектория получена для 10006 изображений на основе 42985 рассчитанных связей. Как

и ожидалось, основная ошибка в таком тесте накапливается на прямолинейных участках пути. На рис. 9 показан график роста ошибки НС в зависимости от пройденного расстояния для второго теста.

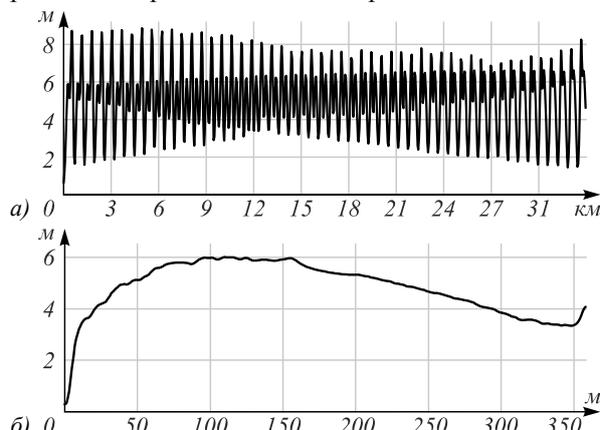


Рис. 7. Зависимость ошибки НС: от пройденного пути (а), от расстояния до начала координат (б)

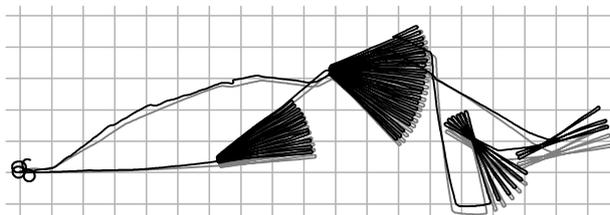


Рис. 8. Траектория АНПА, чередующая зоны с полным покрытием и прямолинейные участки. Чёрной линией показана рассчитанная траектория, а серой – истинная. Сетка имеет размерность  $100 \times 100$  м

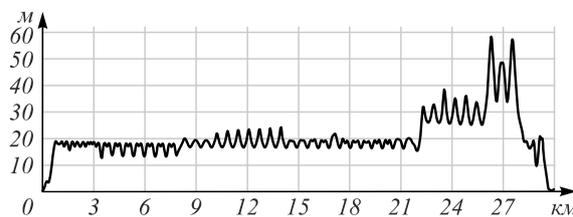


Рис. 9. Зависимость ошибки НС от пройденного расстояния

Как в случае с реальными данными, в экспериментах в виртуальной акватории удалось достичь скорости обработки 3 кадра в секунду. При этом для поиска связей отбирались 20 потенциально связанных изображений ( $M'$  в подпараграфе 4.3). При моделировании работы инерциальной НС использовались следующие параметры:  $\sigma'_t = (0,15, 0,15, 0,15)$  м/с и  $\sigma'_r = (0,01, 0,01, 0,01)$  рад/с. Проведённые эксперименты показали, что глобальный поиск позволяет без труда находить связи между изображениями, даже если это слабоконтрастные, зашумлённые снимки морского дна с однообразными текстурными рисунками. При этом время поиска зависит от количества изображений лишь логарифмически, что позволяет пренебречь этой зависимостью для  $M < 100000$  изображений, получаемых АНПА за миссию.

### Заключение

Алгоритм визуальной навигации, описанный в статье, обладает рядом преимуществ по сравнению с алгоритмами навигации, использующими априорные знания о положении АНПА для поиска связей между изображениями. Самым важным преимуществом является устойчивость к участкам дна, не имеющим каких-либо визуальных особенностей. Даже если большая часть обследуемой территории представлена однородной поверхностью либо значительная часть траектории АНПА проходит слишком высоко для фотосъемки, небольшого количества визуально различных участков будет достаточно для корректировки положения аппарата. Другим преимуществом системы является предсказуемое время работы, так как количество изображений, просматриваемых для установления связей, не зависит от точности, с которой положение АНПА известно в данный момент.

Предложенный алгоритм не требует высокопроизводительных систем для работы. Приемлемая частота обработки в 2–3 кадра в секунду может быть достигнута уже на 4-ядерных процессорах среднего ценового диапазона. Кроме двух параллельных потоков – потока АОСД и потока АКТ, параллельно могут выполняться следующие задачи потока АДИ: расчёт данных облаков, поиск похожих дескрипторов для каждого дескриптора запроса, поиск связей между парой изображений. Таким образом, увеличения скорости работы алгоритма при необходимости можно достичь, увеличив количество процессоров.

Предложенный алгоритм навигации можно рекомендовать к применению на АНПА, оснащённых стереокамерой и предназначенных для решения задачи фотообследования морского дна либо объектов, расположенных на этом дне. Глобальный поиск связей в равной степени может быть применен к НС, основаным на графе состояний АНПА, и к НС, использующим EKF или EIF.

### Благодарности

Работа выполнена при поддержке гранта РФФИ (№ 16-31-00187 мол\_а).

### Литература

1. **Бобков, В.А.** О некоторых алгоритмах визуальной навигации автономного необитаемого подводного аппарата с использованием стереоизображений / В.А. Бобков, С.В. Мельман, А.Ю. Толстоногов, А.Ф. Щербатюк // Подводные исследования и робототехника. – 2016. – № 2(22). – С. 19-24.
2. **Eustice, R.M.** Visually augmented navigation for autonomous underwater vehicles / R.M. Eustice, O. Pizarro, H. Singh // IEEE Journal of Oceanic Engineering. – 2008. – Vol. 33, Issue 2. – P. 103-122. – DOI: 10.1109/JOE.2008.923547.
3. **Salvi, J.** Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles / J. Salvi, Y. Petillot, E. Battle // 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems. – 2008. – P. 1011-1016. – DOI: 10.1109/IROS.2008.4650627.
4. **Aulinas J.** Vision-based underwater SLAM for the SPARUS AUV / J. Aulinas, Y. Petillot, X. Llado, J. Salvi, R. Garcia // Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries. – 2011. – P. 171-180.
5. **Johnson-Roberson, M.** Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys / M. Johnson-Roberson, O. Pizarro, S.B. Williams, I. Mahon // Journal of Field Robotics. – 2010. – Vol. 27, Issue 1. – P. 21-51. – DOI: 10.1002/rob.20324.
6. **Eustice, R.M.** Visually mapping the RMS titanic: Conservative covariance estimates for SLAM information filters / R.M. Eustice, H. Singh J.J. Leonard, M.R. Walter // The International Journal of Robotics Research. – 2006. – Vol. 25, Issue 12. – P. 1223-1242. – DOI: 10.1177/0278364906072512.
7. **Kim, A.** Real-time visual SLAM for autonomous underwater HULL inspection using visual saliency / A. Kim, R.M. Eustice // IEEE Transaction on Robotics. – 2013. – Vol. 29, Issue 3. – P. 719-733. – DOI: 10.1109/TRO.2012.2235699.
8. **Carraso, P.L.N.** Stereo Graph-SLAM for autonomous underwater vehicles / P.L.N. Carraso, F. Bonin-Font, G.O Codina. – In: Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13 / ed. by E. Menegatti, N. Michael, K. Berns, H. Yamaguchi. – Heidelberg, New York, Dordrecht, London: Springer, 2014. – P. 351-360. – DOI: 10.1007/978-3-319-08338-4\_26.
9. **Камаев, А.Н.** Относительное ориентирование снимков низкой контрастности, полученных с АНПА, на основе точечных особенностей / А.Н. Камаев // Учёные заметки ТОГУ. – 2011. – Т. 2, № 2. – С. 32-42.
10. **Lowe, D.G.** Distinctive image features from scale-invariant keypoints / D.G. Lowe // International Journal of Computer Vision. – 2004. – Vol. 60, Issue 2. – P. 91-110. – DOI: 10.1023/B:VISI.0000029664.99615.94.
11. **Hartley, R.** Multiple view geometry in computer vision / R. Hartley, A. Zisserman. – 2<sup>nd</sup> ed. – Cambridge: Cambridge University Press, 2004. – 672 p. – ISBN: 978-0521-54051-3.
12. **Jegou, H.** Product quantization for nearest neighbor search / H. Jegou, M. Douze, C. Schmid // IEEE Transactions on Pattern Analysis and Machine Intelligence. – 2011. – Vol. 33, Issue 1. – P. 117-128. – DOI: 10.1109/TPAMI.2010.57.
13. **Fischler, M.A.** Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography / M.A. Fischler, R.C. Bolles // Communications of the ACM. – 1981. – Vol. 24, Issue 6. – P. 381-395. – DOI: 10.1145/358669.358692.
14. **Kabsch, W.** A solution for the best rotation to relate two sets of vectors / W. Kabsch // Acta Crystallographica Section A. – 1976. – Vol. 32, Issue 5. – P. 922-923. – DOI: 10.1107/S0567739476001873.
15. **Liu, T.** An investigation of practical approximate nearest neighbor algorithms / T. Liu, A.W. Moore, A. Gray, K. Yang // Proceedings of the 17th International Conference on Neural Information Processing Systems. – 2004. – P. 825-832.
16. **Beis, J.** Shape indexing using approximate nearest-neighbour search in high-dimensional spaces / J. Beis, D.G. Lowe // Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97). – 1997. – P. 1000-1006. – DOI: 10.1109/CVPR.1997.609451.
17. **Камаев, А.Н.** Позиционирование изображений морского дна полученных с помощью АНПА / А.Н. Камаев // Подводные исследования и робототехника. – 2013. – № 2(16). – С. 38-47.
18. **Grisetti, G.** A tutorial on graph-based SLAM / G. Grisetti, R. Kümmerle, C. Stachniss, W. Burgard // IEEE Intelligent Transportation Systems Magazine. – 2010. – Vol. 2, Issue 4. – P. 31-43. – DOI: 10.1109/MITS.2010.939925.

19. **Камаев, А.Н.** Constructing and visualizing three-dimensional sea bottom models to test AUV machine vision systems / A.N. Kamaev, V.A. Sukhenko, D.A. Karmanov // Programming and Computer Software. – 2017. – Vol. 43, Issue 3. – P. 184-195. – DOI: 10.1134/S0361768817030070.
20. **Камаев, А.Н.** Synthetic data for AUV technical vision systems testing / A.N. Kamaev, S.I. Smagin, V.A. Sukhenko, D.A. Karmanov // CEUR Workshop Proceedings. – 2017. – Vol. 1839. – P. 126-140.

#### Сведения об авторах

**Камаев Александр Николаевич**, кандидат технических наук, старший научный сотрудник Вычислительного центра ДВО РАН, лаборатория численных методов в математической физике, Хабаровск, Россия. Область научных интересов: компьютерное зрение, обработка изображений, компьютерная графика. E-mail: [kamaev\\_an@mail.ru](mailto:kamaev_an@mail.ru).

**Карманов Дмитрий Александрович**, старший программист Вычислительного центра ДВО РАН, лаборатория информационно-коммуникационных систем, Хабаровск, Россия. Область научных интересов: компьютерная графика, распределённые вычисления и системы искусственного интеллекта. E-mail: [karmanno@gmail.com](mailto:karmanno@gmail.com).

ГРНТИ: 28.23.15.

Поступила в редакцию 7 февраля 2018 г.

### VISUAL NAVIGATION OF AN AUTONOMOUS UNDERWATER VEHICLE BASED ON THE GLOBAL SEARCH OF IMAGE CORRESPONDENCES

A.N. Kamaev<sup>1</sup>, D.A. Karmanov<sup>1</sup>

<sup>1</sup> Computing Center FEB RAS, Khabarovsk, Russia

#### Abstract

A task of autonomous underwater vehicle (AUV) navigation is considered in the paper. The images obtained from an onboard stereo camera are used to build point clouds attached to a particular AUV position. Quantized SIFT descriptors of points are stored in a metric tree to organize an effective search procedure using a best bin first approach. Correspondences for a new point cloud are searched in a compact group of point clouds that have the largest number of similar descriptors stored in the tree. The new point cloud can be positioned relative to the other clouds without any prior information about the AUV position and uncertainty of this position. This approach increases the reliability of the AUV navigation system and makes it insensitive to data losses, textureless seafloor regions and long passes without trajectory intersections. Several algorithms are described in the paper: an algorithm of point clouds computation, an algorithm for establishing point clouds correspondence, and an algorithm of building groups of potentially linked point clouds to speedup the global search of correspondences. The general navigation algorithm consisting of three parallel subroutines: image adding, search tree updating, and global optimization is also presented. The proposed navigation system is tested on real and synthetic data. Tests on real data showed that the trajectory can be built even for an image sequence with 60% data losses with successive images that have either small or zero overlap. Tests on synthetic data showed that the constructed trajectory is close to the true one even for long missions. The average speed of image processing by the proposed navigation system is about 3 frames per second with a middle-price desktop CPU.

**Keywords:** navigation, AUV, SLAM, feature points, dead reckoning, image matching.

**Citation:** Kamaev AN, Karmanov DA. Visual navigation of an autonomous underwater vehicle based on the global search of image correspondences. Computer Optics 2018; 42(3): 457-467. DOI: 10.18287/2412-6179-2018-42-3-457-467.

**Acknowledgements:** The work was partially funded by the Russian Foundation for Basic Research under the research project 16-31-00187 mol\_a.

#### References

- [1] Bobkov VA, Mel'man SV, Tolstonogov AJu, Shherbatjuk AF. On some algorithms for visual navigation of the autonomous underwater vehicle using stereo images [In Russian]. Underwater Investigations and Robotics 2016; 2(22): 19-24.
- [2] Eustice RM, Pizarro O, Singh H. Visually augmented navigation for autonomous underwater vehicles. IEEE Journal of Oceanic Engineering 2008; 33(2): 103-122. DOI: 10.1109/JOE.2008.923547.
- [3] Salvi J, Petillot Y, Battle E. Visual SLAM for 3D large-scale seabed acquisition employing underwater vehicles. 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems 2008: 1011-1016. DOI: 10.1109/IROS.2008.4650627.
- [4] Aulinas J, Petillot Y, Llado X, Salvi J, Garcia R. Vision-based underwater SLAM for the SPA-RUS AUV. Proceedings of the 10th International Conference on Computer and IT Applications in the Maritime Industries 2011: 171-180.
- [5] Johnson-Roberson M, Pizarro O, Williams SB, Mahon I. Generation and visualization of large-scale three-dimensional

- reconstructions from underwater robotic surveys. *Journal of Field Robotics* 2010; 27(1): 21-51. DOI: 10.1002/rob.20324.
- [6] Eustice RM, Singh H, Leonard JJ, Walter MR. Visually mapping the RMS titanic: Conservative covariance estimates for SLAM information filters. *The International Journal of Robotics Research* 2006; 25(12): 1223-1242. DOI: 10.1177/0278364906072512.
- [7] Kim A, Eustice RM. Real-time visual SLAM for autonomous underwater HULL inspection using visual saliency. *IEEE Transaction on Robotics* 2013; 29(3): 719-733. DOI: 10.1109/TRO.2012.2235699.
- [8] Carraso PLN, Bonin-Font F, Codina GO. Stereo Graph-SLAM for autonomous underwater vehicles. In Book: Menegatti E, Michael N, Berns K, Yamaguchi H, eds. *Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13*. Heidelberg, New York, Dordrecht, London: Springer; 2014: 351-360. DOI: 10.1007/978-3-319-08338-4\_26.
- [9] Kamaev AN. Relative features based orientation of low contrast images obtained from AUV [In Russian]. *Scientists notes PNU* 2011; 2(2): 32-42.
- [10] Lowe DG. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 2004; 60(2): 91-110. DOI: 10.1023/B:VISI.0000029664.99615.94.
- [11] Hartley R, Zisserman A. *Multiple view geometry in computer vision*. 2<sup>nd</sup> ed. Cambridge: Cambridge University Press; 2004. DOI: 10.1023/B:VISI.0000029664.99615.94.
- [12] Jegou H, Douze M, Schmid C. Product quantization for nearest neighbor search. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 2011; 33(1): 117-128. DOI: 10.1109/TPAMI.2010.57.
- [13] Fischler MA, Bolles RC. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM* 1981; 24(6): 381-395. DOI: 10.1145/358669.358692.
- [14] Kabsch W. A solution for the best rotation to relate two sets of vectors. *Acta Crystallographica Section A* 1976; 32(5): 922-923. DOI: 10.1107/S0567739476001873.
- [15] Liu T, Moore AW, Gray A, Yang K. An investigation of practical approximate nearest neighbor algorithms. *Proceedings of the 17th International Conference on Neural Information Processing Systems* 2004: 825-832.
- [16] Beis J, Lowe DG. Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. *Proceedings of the 1997 Conference on Computer Vision and Pattern Recognition (CVPR '97)* 1997: 1000-1006. DOI: 10.1109/CVPR.1997.609451.
- [17] Kamaev AN. Computation of positions of sea floor images obtained from AUV [In Russian]. *Underwater Investigations and Robotics* 2013; 2(16): 38-47.
- [18] Grisetti G, Kümmerle R, Stachniss C, Burgard W. A Tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine* 2010; 2(4): 31-43. DOI: 10.1109/MITS.2010.939925.
- [19] Kamaev AN, Sukhenko VA, Karmanov DA. Constructing and visualizing three-dimensional sea bottom models to test AUV machine vision systems. *Program Comput Soft* 2017; 43(3): 184-195. DOI: 10.1134/S0361768817030070.
- [20] Kamaev AN, Smagin SI, Sukhenko VA, Karmanov DA. Synthetic data for AUV technical vision systems testing. *CEUR Workshop Proceedings* 2017; 1839: 126-140.

#### *Author's information*

**Aleksandr Nikolaevich Kamaev**, candidate of Technical Sciences, senior research fellow of the Computing Center FEB RAS, Numerical Methods in Mathematical Physics laboratory, Khabarovsk, Russia. His research interests are currently focused on computer vision, image processing and computer graphics. E-mail: [kamaev\\_an@mail.ru](mailto:kamaev_an@mail.ru).

**Dmitry Aleksandrovich Karmanov**, senior software developer of the Computing Center FEB RAS, Information and Telecommunication Systems laboratory, Khabarovsk, Russia. His research interests are currently focused on computer graphics, distributed computations and artificial intelligence systems. E-mail: [karmanno@gmail.com](mailto:karmanno@gmail.com).

*Received February 7, 2018.*