

**Acknowledgements:** The work was supported by the Russian Foundation for Basic Research, project no. 15-01-07900

For citation:

**Karpova I. P.** Pseudo-Analog Communication in Robot Groups, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 2, pp. 94–101.

DOI: 10.17587/mau/17.94-101

### References

1. **Yogeswaran M., Ponnambalam S. G.** An Extensive Review of Research in Swarm Robotics, *2009 World Congress on Nature and Biologically Inspired Computing, NABIC*, 2009, pp. 140–145.
2. **Jurevich E. I.** *O probleme gruppovogo upravlenija robotami* (About problem of robots group management), *Mekhatronika, Avtomatizatsija, Upravlenie*, 2004, vol. 2, pp. 9–13 (in Russian).
3. **Kaljaev I. A., Gajduk A. R., Kapustjan S. G.** *Modeli i algoritmy kollektivnogo upravlenija v gruppah robotov* (Models and algorithms of management in robots groups), Moscow, Fizmatlit, 2009, 279 p. (in Russian).
4. **Shi Z., Tu J., Zhang Q., Liu L., Wei J.** A survey of swarm robotics system, *3rd International Conference on Swarm Intelligence, ICSI 2012*, Shenzhen, China, 2012, vol. 7331 LNCS, Issue PART 1., pp. 564–572.
5. **Karpov V. E.** *Znak-orientirovannyj mehanizm lokal'nogo vzaimodejstvija mezdu robotami* (The sign-focused mechanism of local interaction between robots), *Integrirrovannye modeli i mjagkie vychislenija v iskusstvennom intellekte*, Proc. VIII Internat. sc.-tech. conf., 2015, vol. 2, pp. 504–514, 388 p. (in Russian).
6. **Panov E. N.** *Metodologicheskie problemy v izuchenii kommunikacii i social'nogo povedenija zhivotnyh* (Methodological problems in studying of communication and social behavior of animals), *Problemy jetologii nazemnyh pozvonochnyh. Itogi nauki i tehniki. Serija "Zoologija pozvonochnyh"*, Moscow, VINITI, 1983, vol. 12, pp. 5–70 (in Russian).
7. **Serebrjakova T. A.** *Zoopsihologija i sravnitel'naja psihologija* (Zoop psychology and comparative psychology), Novgorod, Publishing house of VGIPU, 2011, 328 p. (in Russian).
8. **Panov E. N.** *Ponjatije "signal" v aspekte kommunikacii zhivotnyh. O chem rech'?* (Concept "signal" of aspect of communication of animals. About what the speech?), *Ethology and zoopsychology*, 2012, no. 2 (6), pp. 1–35 (in Russian).
9. **Beer C. G.** Multiple function and gull displays, *Function and evolution of behavior*, Oxford, Clarendon Press, 1975, pp. 16–64.
10. **Ren X.-P., Cai Z.-X., Chen A.-B.** Current research in multi-mobile robots communication system (Review), *Control and Decision*, 2010, vol. 25, iss. 3, pp. 327–332.
11. **Kornienko S., Kornienko S.** IR-based communication and perception in microrobotic swarms, *Proc. of the 7th Workshop on Collective & Swarm Robotics*, 18 November, University of Stuttgart, Germany, 2010.
12. **Agent Communication Language: Part 2.** FIPA 97 Specification, available at: <http://www.fipa.org/specs/fipa00018/OC00018.pdf>, date of access 30.09.2015.
13. **Anders Lyhne Christensen, Rehan O'Grady, Marco Dorigo.** SWARMORPH-script: a language for arbitrary morphology generation in self-assembling robots, *Swarm Intell*, 2008, no. 2, pp. 143–165, DOI 10.1007/s11721-008-0012-6.
14. **Gao Z., Yan G., Ding G., Huang H.** Research of communication mechanism of the multi-agent in multi-agent robot systems, *High Technology Letters*, 2002, vol. 8, iss. 1, pp. 67–71.
15. **Pal A., Tiwari R., Shukla A.** Multi-Robot Exploration in Wireless Environments, *Cognitive Computation*, 2012, vol. 4, iss.e 4, pp. 526–542.
16. **Fridman V. S.** *Rasshifrovka "jazyka zhivotnyh": idei, problemy, perspektivy* (Interpretation of "language of animals": ideas, problems, prospects), available at: <http://ethology.ru/library/?id=304>, date of access 30.09.2015.
17. **Cheney D. L., Seyl'arth R. M.** *How Monkeys See the World: Inside the Mind of Another Species*, Chicago, University of Chicago Press, 1990.
18. **Slabbekoorn H., Pctt M.** Birds sing at a higher pitch in urban noise, *Nature*, 2003, vol. 424, pp. 267.
19. **Karpov V. E.** *Chastnye mehanizmy liderstva i samosoznanija v gruppovoj robototekhnike* (Private mechanisms of leadership and consciousness in a swarm robotics), *Proceedings of the XIII National Conference on artificial intelligence with the international participation (CAI-2012)*, Publishing house of BGTU, 2012, pp. 275–283 (in Russian).
20. **Beinhofer M., Müller J., Burgard W.** Effective landmark placement for accurate and reliable mobile robot navigation, *Robotics and Autonomous Systems*, 2013, vol. 61, iss. 10, pp. 1060–1069.
21. **Jung C., Chung W.** Calibration of kinematic parameters for two wheel differential mobile robots by using experimental heading errors, *International Journal of Advanced Robotic Systems*, 2011, vol. 8, iss. 6, pp. 134–142.
22. **Swarm Communication.** Jasmine: swarm robot platform, available at: <http://www.swarmrobot.org/Communication.html>, date of access 30.09.2015.
23. **Pavlovskiy V. E., Kirkov A. Yu.** *Tonal'naja mul'tichastotnaja akusticheskaja kommunikacija robotov* (Voice-frequency multifrequency acoustic communication of robots), Preprint IPM im. M. V. Kel'dysha, 2013, no. 102, 32 p. (in Russian).

УДК 004.93

DOI: 10.17587/mau.17.101-109

**В. А. Бобков**, д-р техн. наук, зав. лабораторией, bobkov@iacp.dvo.ru,  
Институт автоматизации и процессов управления ДВО РАН, Владивосток,

**В. Ю. Машенцев**, ассистент, v.mashentsev@gmail.com,  
Дальневосточный федеральный университет, Владивосток

## Навигация подводного робота по стереоизображениям\*

Описан метод визуальной навигации автономного подводного робота по потоку стереоизображений, основанный на визуальной одометрии. Новизна предлагаемого метода заключается в реализации адаптивной методики расчета траектории робота, ориентированной на достижение высокой точности вычисляемой локализации, обеспечение режима реального времени при онлайн-обработке и на значительное сокращение вычислительных затрат при постобработке.

**Ключевые слова:** автономный подводный робот, визуальная навигация, адаптивный метод, стереоизображения

\* Работа выполнена при финансовой поддержке РФФИ (проект № 15-07-00341) и Программы "Дальний Восток" (проект 15-1-4-025).

## Введение

Подводная робототехника применяется сегодня для решения достаточно широкого круга прикладных задач, включая поисковые и спасательные работы, океанологические и экологические измерения, инспекцию подводных коммуникаций, поиск источников экологического загрязнения акваторий и др. Важной задачей при этом является задача точной локализации и навигации автономных подводных роботов (АПР) в априори неизвестной сложной подводной обстановке. В отличие от наземных автономных роботов и летающих беспилотных аппаратов, в случае АПР нет возможности использования систем GPS/ГЛОНАСС или предварительно подготовленных цифровых карт местности. Как правило, для навигационных целей применяются гидроакустические навигационные системы [1], у которых есть свои ограничения. Для получения географической привязки АПР с повышенной точностью применяют также системы развертываемых маяков, однако они требуют значительных усилий и времени для их предварительной установки. Перспективной альтернативой и дополнением к традиционному подходу с использованием гидроакустических приборов для АПР является подход, основанный на обработке видеoinформации. Его привлекательность состоит в том, что при локальном маневрировании, связанном с выполнением требующих высокой точности практических операций АПР, получаемая по видеoinформации точность может быть выше традиционной гидроакустической точности в силу особенностей режима маневрирования. На первом этапе исследований визуального подхода к навигации рассматривались постановки задачи по разработке SLAM-алгоритмов (Simultaneous Localization and Mapping) с обработкой 2D-информации, получаемой одной видеокамерой. Например, в работе [2] успешно была решена задача определения параметров движения автономного подводного аппарата на основе анализа 2D-снимков морского дна с использованием детектора Харриса и предложенного авторами метода выделения границ видимых объектов. Аналогичные методы сохраняют свою актуальность и сегодня в приложениях, где достаточно использования монокамеры. Так, в работе [3] представлен алгоритм, применявшийся при инспекции корпусов судов автономным подводным аппаратом, оснащенным монокамерой. В последнее время стали активно развиваться методы, основанные на обработке стереоизображений, получаемых с помощью стереокамер. Многие из первых и последующих работ, относящихся к подводным аппаратам, основывались на применении фильтра Калмана [4–6]. Как альтернатива применению фильтра Калмана с появлением возможности обработки стереоизображений получил развитие подход, основанный на визуальной одометрии, преимущество которого состоит в совместной обработке 2D- и 3D-видеоданных и возможности вычислять пара-

метры пространственного движения робота (6 степеней свободы). Например, в работе [7] он успешно применялся при решении актуальной задачи реконструкции больших протяженных сцен морской подводной обстановки. Несмотря на имеющиеся сегодня убедительные результаты по разработке visual-SLAM применительно к автономным роботам различного назначения, с разной оснащенностью сенсорами и в разных постановках задачи, остается актуальной задача как повышения точности локализации АПР, так и снижения вычислительных затрат, что особенно важно для обеспечения режима реального времени. Для АПР наземного и летающего типа повышению точности способствует применение лазерных сенсоров типа Kinect [8], однако такие технические возможности для подводных АПР пока не получили должного развития.

В предлагаемой авторами настоящей статье сделан акцент на снижении трудоемкости и повышении точности локализации за счет адаптивного расчета траектории АПР. Рассматривается авторская реализация метода визуальной одометрии и его модификации, направленные на повышение точности навигации АПР и скорости вычислений. В данной работе получили развитие ранее полученные авторами в работах [9, 10] результаты. Предлагаемый метод основывается на обработке последовательности стереоизображений. Основным результатом авторов состоит в разработке адаптивной методики расчета траектории АПР, обеспечивающей оптимизацию достигаемой точности локализации и вычислительных затрат, а также в получении оценок эффективности предложенных алгоритмических решений.

## 1. Постановка задачи

Задача заключается в разработке адаптивного метода расчета параметров пространственной (6 степеней свободы) траектории АПР при его движении в априори неизвестной подводной обстановке по видеoinформации, получаемой с помощью непрерывной съемки стереокамерой. Адаптивный метод, базирующийся на подходе визуальной одометрии, должен обеспечить существенное снижение вычислительных затрат в сравнении с традиционной реализацией метода визуальной навигации за счет оптимального выбора расчетных позиций АПР в зависимости от характера траектории (высоты траектории над морским дном, скорости движения АПР, степени перекрытия соседних видов камеры) и от требований к алгоритмической эффективности. Также адаптивный метод должен обеспечивать два различных режима расчета траектории — режим реального времени и режим постобработки, каждый из которых должен отвечать своим требованиям, вытекающим из необходимости решения соответствующих практических задач. Разрешение получаемых снимков — от  $600 \times 400$  до  $1360 \times 1024$ . Для оценки эффективности предложенного адаптивного метода визуальной навигации необходимо провести вычислительные эксперименты с модифи-

кациями программной реализации метода, построенными на применении в основной вычислительной схеме возможных альтернативных известных и предлагаемых алгоритмических локальных решений.

## 2. Базовый метод

В рассматриваемой постановке АПР оснащен стереокамерой, с помощью которой он выполняет съемку дна (25 Гц) при движении по траектории в априори неизвестной среде. Задача заключается в точном вычислении параметров траектории АПР с одновременным построением модели среды по получаемой последовательности кадров. Согласно методу визуальной одометрии решение строится на последовательном вычислении локальных перемещений аппарата, описываемых соответствующими локальными матрицами геометрических преобразований. Локальными будем называть перемещения АПР между соседними позициями. Объединение локальных матриц всех предшествующих позиций позволяет получить матрицу преобразования, связывающую мировую систему координат (МСК) с локальной системой координат (ЛСК), привязанной к аппарату в каждой текущей позиции его траектории. Применяемый в настоящей работе метод вычисления матрицы локального преобразования следует уже сложившемуся подходу, основанному на совместном использовании 2D-особенностей на изображениях и 3D-облаков точек, получаемых на базе стереопар. Вычислительную схему определения параметров текущей позиции АПР кратко можно представить следующим образом:

1. Выделение общего множества особенностей на четверке изображений двух стереопар.
2. Генерация и фильтрация двух облаков 3D-точек.
3. Вычисление локального геометрического преобразования, связывающего локальные СК двух соседних позиций.
4. Вычисление параметров текущей позиции АПР путем объединения локальных преобразований предшествующих позиций.

Согласно схеме на первом этапе анализируются две стереопары изображений, относящиеся к двум последовательным во времени позициям АПР. Применительно к ним осуществляется поиск и сопоставление единого множества точечных особенностей, одновременно наблюдаемых на всех четырех изображениях. Для каждой из двух стереопар выполняется построение 3D-точек-особенностей (3D-облако) методом триангуляции, так как известна полная калибровка используемых камер. На втором этапе вычисляется перемещение (шесть степеней свободы) робота из предыдущей позиции в текущую позицию методом решения оптимизационной задачи, где в качестве целевой функции рассматривается минимизация суммарных расхождений по всем точкам двух 3D-облаков.

В рассматриваемом подходе на получаемую точность визуальной навигации негативно влияют

ошибочные сопоставления особенностей и недостаточное число точек в 3D-облаках. Поэтому для устранения влияния ложных сопоставлений применяется фильтрация ошибок как на этапе 2D-обработки (пороговое отсечение, кросс-проверка, эпиполярные ограничения в каждой стереопаре), так и на этапе построения 3D-облаков (итерационная фильтрация точек на основе структурной когерентности облаков или/и на основе критерия связности поверхности). Поддержание гарантированного числа точек в 3D-облаках обеспечивается предлагаемой авторами и описываемой ниже адаптивной методикой.

### 2.1. Сопоставление особенностей и фильтрация

Сопоставление точечных особенностей выполняется на четырех изображениях — двух стереопарах (1—2 и 3—4), соответствующих двум последовательным позициям АПР на траектории (рис. 1, см. вторую сторону обложки). Большое число правильных сопоставлений обеспечивает высокую точность вычисления локальных перемещений. Ошибки при сопоставлении возникают из-за наличия шумов (различие в освещенности изображений, слабая текстурированность или регулярность текстуры), что приводит к значительной доле ложных сопоставлений и, в конечном счете, к малому числу правильно сопоставленных особенностей после фильтрации. Поэтому было рассмотрено и проверено несколько схем получения единого множества особенностей для четверки изображений двух стереопар с альтернативным использованием известных детекторов SURF, FREAK, FAST, а также алгоритма вычисления диспарантности для калиброванных очищенных пар. При программной реализации использовалась библиотека OpenCV. Возможны несколько схем обхода и применения фильтров при получении единого множества особенностей для четверки изображений. Выбор был сделан в пользу двух схем: а) последовательное сопоставление пар изображений 1—3, 3—4, 4—2 с выделением общего множества особенностей. На всех трех этапах работает SURF. Для каждой пары изображений выполняется сопоставление слева направо и справа налево (cross-checking). Для исключения ложных сопоставлений применяется эпиполярный фильтр для пары 3—4 и для пары 1—2; б) вначале выполняется сопоставление пары 1—3 детектором SURF, затем применяется алгоритм определения диспарантности для калиброванных пар 1—2 и 3—4. Эпиполярный фильтр основывается на известном из геометрии двух изображений эпиполярном ограничении. Согласно эпиполярной геометрии на положение точек  $m_1$  и  $m_2$ , являющихся проекциями одной и той же точки  $M$  пространственной сцены на первое и второе изображение, соответственно, накладывается ограничение

$$m_2^T F m_1 = 0$$

(здесь  $m_1, m_2$  — однородные координаты точек,  $F$  — фундаментальная матрица), которое заключается в том, что обе точки должны лежать на соответствующих эпиполярных линиях. При фиксации одной из точек эпиполярная линия для второй вычисляется из приведенного соотношения.

Матрица  $F$  вычисляется через известные калибровочные матрицы  $K_1$  и  $K_2$  и существенную матрицу  $E$ . В свою очередь, существенная матрица  $E$  вычисляется как  $E = [t]_x R$ , где  $t$  — вектор переноса камеры,  $R$  — матрица поворота камеры, а  $[t]_x =$

$$= \begin{bmatrix} 0 & -t_z & t_y \\ t_z & 0 & -t_x \\ -t_y & t_x & 0 \end{bmatrix} \text{ — матрица векторного произведения.}$$

Процесс сопоставления проиллюстрирован на рис. 1 (см. вторую сторону обложки). На одном из шагов на изображениях 1, 2, 3, 4 были сгенерированы, соответственно, 1123, 1100, 1183, 1146 особенности. На каждом этапе 1—3, 3—4, 4—2 были сопоставлены, соответственно, 578, 427, 333 особенности. Единое множество на четверке изображений — 333 особенности. Это позволяет построить на следующем этапе два сопоставленных облака 3D-точек, отвечающих двум стереопарам (двум позициям АПР).

## 2.2. Расчет параметров траектории

Траектория движения АПР представляет собой последовательность позиций, в каждой из которых необходимо вычислить координаты положения и ориентации аппарата в МСК. В качестве МСК без нарушения общности можно выбрать ЛСК АПР в начальной позиции. Тогда, согласно методу визуальной одометрии, мировые координаты очередной позиции траектории можно получить путем объединения локальных перемещений. В свою очередь, изменение в положении и ориентации АПР при локальном перемещении можно вычислить, опираясь на сравнение двух сопоставленных 3D-облаков, одно из которых построено в СК первой стереопары (предыдущая позиция), а другое — в СК второй стереопары (текущая позиция). Пространственная точка  $P$  облака стереопары реконструируется по двум образам точечной особенности следующим образом.

Пусть  $p_1 = (x_1, y_1)$  — точечная особенность на левом снимке;  $p_2 = (x_2, y_2)$  — соответствующая точечная особенность на правом снимке,  $f$  — фокусное расстояние левой и правой камеры,  $A$  — матрица, преобразующая точки из СК правой камеры в СК левой камеры,  $o = (0, 0, 0)$  — начала СК левой и правой камер.

$p_1^{3D} = (x_1, y_1, f)$  — точка  $P$  на левом экране для первой особенности,

$p_2^{3D} = (x_2, y_2, f)$  — точка  $P$  на правом экране для второй особенности.

Построим луч в левой СК, проходящий через первую особенность:

$R_1 = (o_{ray1}, v_{ray1})$ , здесь  $o_{ray}$  — точка начала луча,  $v_{ray}$  — вектор направления луча, при этом

$$o_{ray1} = o, v_{ray1} = \frac{p_1^{3D}}{\|p_1^{3D}\|}.$$

Аналогичным образом строится луч в правой СК, проходящий через вторую особенность:

$$R_2 = (o_{ray2}, v_{ray2}), \text{ здесь } o_{ray2} = o, v_{ray2} = \frac{p_2^{3D}}{\|p_2^{3D}\|}.$$

Луч  $R_2$  переводится в левую СК, обозначим его через  $R_{21} = (o_{ray21}, v_{ray21})$ :

$$o_{ray21} = o_{ray2} \cdot A, v_{ray21} = v_{ray2} \cdot A.$$

Теперь точка  $P$  строится в левой СК как пересечение лучей  $R_1$  и  $R_{21}$ .

Задача вычисления матрицы геометрического преобразования для локального перемещения сводится к решению переопределенной системы уравнений, где данными являются координаты наблюдаемых точек, а неизвестными являются элементы матрицы геометрического преобразования.

Пусть перемещение АПР из позиции  $p_1$  траектории в позицию  $p_2$  определяется некоторым неизвестным преобразованием  $H$ , состоящим из вектора переноса и кватерниона вращения. Будем искать это преобразование, основываясь на имеющемся взаимоднозначном сопоставлении двух облаков 3D-точек, наблюдаемых парой камер, соответственно, в позициях 1 и 2. Первое облако  $C^1(x, y, z)$  задано в СК первой стереопары, второе облако  $C^2(x, y, z)$  — в СК второй стереопары. Задача нахождения матрицы  $H$  решается методом градиентного спуска (используется библиотека MATLAB). В качестве параметров оптимизации используются три координаты вектора переноса и четыре координаты кватерниона  $Q$ , определяющего вращение в  $H$ . Ограничение задается условием — норма  $Q$  равна 1.

Целевая функция  $F = \sum \|c_k^2 - c_k^1 H\|$ ,  $k$  — номер точки в облаке.

Здесь  $\{c_k^1\}$  — множество точек в первом облаке и  $\{c_k^2\}$  — множество точек во втором облаке. Тогда с учетом вычисленной матрицы  $H$  новое положение АПР вычисляется через предыдущее как  $p_2 = p_1 H$ . Поскольку нам нужны координаты точек траектории в МСК, необходимо вычислить преобразование из МСК в СК  $i$ -й позиции траектории. Если в качестве МСК выбрана СК первой пары в последовательности, то  $H_0$  — единичная матрица (для произвольной МСК необходимо знать  $H_0$ ). С учетом знания локальных преобразований для всех предшествующих позиций траектории преобразование из МСК в СК  $i$ -й позиции вычисляется как последовательное произведение локальных преобразований:  $H W_i = H_0 H_1 \dots H_i$ . Соответственно, мировые

координаты  $i$ -й позиции траектории АПР вычисляются как  $p_i = p_0 H W_i^{-1}$ .

В качестве альтернативы применению MATLAB для вычисления локального преобразования был реализован вариант с использованием RANSAC-метода и библиотеки MRPT (Mobile Robot Programming Toolkit).

### 3. Адаптивный алгоритм расчета траектории

Расчет реальной траектории АПР по изображениям необходим при решении двух практических задач: а) навигация робота в режиме реального времени и б) 3D-реконструкция подводной сцены по последовательности видов в режиме постобработки (в этом случае вычисление шести степеней свободы позиций АПР (камеры), по сути, является внешней калибровкой изображений, которая необходима для объединения отдельных видов в единую 3D-модель наблюдаемой сцены). В режиме постобработки для работы алгоритма доступна вся последовательность кадров съемки в каждый момент времени. В режиме реального времени (обработка осуществляется бортовым компьютером синхронно с движением робота) естественным ограничителем при выборе алгоритмом шага обработки траектории является то число кадров, которое было накоплено в процессе съемки к данному моменту времени.

Согласно методу визуальной одометрии вычисление текущей позиции АПР основывается на результатах обработки всех предшествующих позиций. Поэтому для соответствия режиму реального времени необходимо, чтобы скорость алгоритмических вычислений была достаточной для того, чтобы время компьютерной обработки поступающих визуальных данных не превышало времени движения АПР по траектории на текущий момент. Время расчета параметров траектории определяется трудоемкостью обработки одной позиции (в нашем случае она определяется в основном значительной вычислительной трудоемкостью детектора SURF). Поэтому скорость расчета в целом потенциально можно повысить за счет сокращения числа обсчитываемых на траектории позиций. Заметим, что в этом случае при расчете траектории АПР нужно удовлетворять двум противоречивым требованиям одновременно. С одной стороны, для обеспечения высокой точности навигации необходимо близкое соседство обсчитываемых позиций, что приводит к увеличению числа позиций (шагов по траектории) и, соответственно, к увеличению счетного времени. С другой стороны, чтобы сократить общее время счета, необходимо уменьшать число обсчитываемых позиций. Уточним, что здесь и далее под термином "шаг" понимается время (с) перемещения аппарата в очередную позицию или эквивалентное ему число кадров, которое можно подсчитать исходя из известной частоты съемки. Очевидно, что методика с фиксированным шагом не эффективна по ряду причин: не гарантируется достаточное число сопоставленных особенностей, не учитывается форма траектории и ее вы-

сота, динамика движения АПР, число накапливаемых в буфере памяти кадров съемки при реальном движении аппарата не будет соответствовать фиксированному шагу. Поэтому была разработана адаптивная схема расчета позиций траектории, ориентированная на минимизацию счетного времени без ущерба для точности вычисления траектории (при условии, что число обсчитываемых позиций на траектории является достаточным для системы управления АПР). Адаптивность реализуется тремя совместно работающими механизмами: а) выбор размера очередного шага в зависимости от степени перекрытия зон видимости стереокамеры для двух соседних позиций; б) уменьшение шага, если число сопоставленных особенностей меньше заданного порога; в) отсечение тех частей изображения, которые не относятся к перекрытию зон видимости. Оно выполняется непосредственно перед генерацией и сопоставлением особенностей в каждом из четырех исходных изображений.

Первый механизм позволяет задавать максимальное значение очередного шага, гарантирующее заданный размер общей (для двух позиций) зоны видимости, что направлено на минимизацию числа обсчитываемых позиций и, соответственно, на уменьшение времени счета. Уменьшение или увеличение размера общей области видимости позволяет увеличивать или, соответственно, уменьшать максимальный шаг. Зависимость между перемещением камеры и размером общей зоны видимости определяется следующими геометрическими соотношениями (рис. 2). Пусть  $\alpha$  — вертикальный угол обзора камеры (по движению камеры),  $R$  — среднее расстояние камеры до дна,  $H$  — линейный размер зоны видимости камеры в проекции на плоскость дна,  $P$  — заданный процент перекрытия зон видимости двух камер,  $f$  — частота съемки,  $S$  — перемещение камеры из предыдущей позиции в текущую позицию при заданном проценте перекрытия зон видимости.

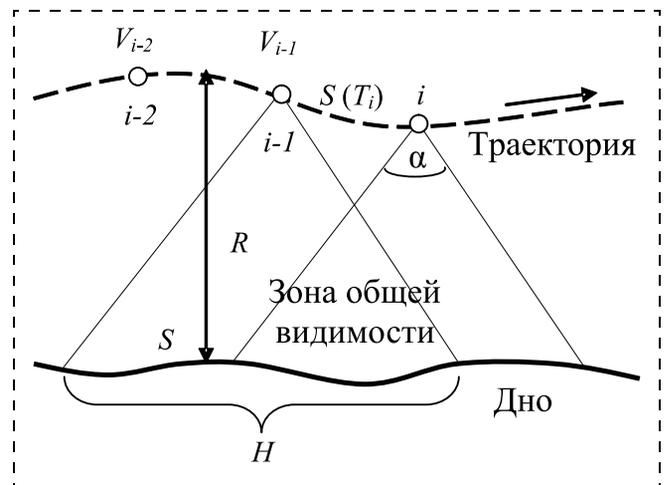


Рис. 2. Вычисление адаптивного шага в зависимости от зоны общей видимости

Тогда

$$H = 2\text{tg}\left(\frac{\alpha}{2}\right) R,$$

$$S = H\left(1 - \frac{P}{100}\right).$$

При вычислении максимального размера очередного шага будем учитывать динамику АПР, т. е. на прямолинейных участках движения, где скорость высокая, шаг может быть большим, а на криволинейных участках, где скорость движения снижается, шаг должен уменьшаться. Обозначим  $T_i$  — искомое время перемещения на очередном шаге  $i$ ;  $T_{pre}$  — время перемещения на предыдущем шаге  $i - 1$ ;  $V_{i-1}$  — скорость АПР в позиции  $i - 1$ ;  $V_{i-2}$  — скорость АПР в позиции  $i - 2$ . Все величины, относящиеся к предыдущему шагу, известны. Величина  $A = (V_{i-1} - V_{i-2})/T_{pre}$  определяет ускорение на предыдущем шаге. Соответственно, скорость на текущем участке  $[i - 1, i]$  будет изменяться от  $V_{i-1}$  до величины  $V_{i-1} + AT_i$ . Для подсчета пути, пройденного за время  $T_i$ , можно исходить из средней скорости  $V_{i-1} + AT_i/2$ . Тогда можно получить выражение для искомой величины  $T_i$ :

$$T_i = S/(V_{i-1} + AT_i/2).$$

Оно эквивалентно квадратному уравнению

$$AT_i^2 + 2V_{i-1}T_i - S = 0 \quad (1)$$

с учетом особых случаев:

1) если  $A = 0$ , то  $T_i = \frac{S}{V_{i-1}}$ ;

2) если  $A < 0$  и дискриминант  $D = V_{i-1}^2 + 2AS < 0$ ,

то  $T_i = -\frac{V_{i-1}}{A}$ .

В общем случае  $T_i = (-V_{i-1} + \sqrt{D})/A$ . Для пересчета размера шага в число кадров полученную величину умножаем на  $f$ . Таким образом, обеспечивается адаптивное вычисление текущего шага с учетом расстояния до дна, размера желаемого перекрытия зон видимости и динамики движения АПР.

Второй механизм — уменьшение шага — направлен на поддержание числа особенностей на

уровне не ниже заданного порога, что необходимо для обеспечения высокой точности вычисления матрицы локального преобразования. Если число сопоставленных особенностей меньше заданного порога, то делается повторное сопоставление после деления шага пополам (в экспериментах порог варьировался от 30 до 100 точечных особенностей). Процесс итеративный, продолжающийся до тех пор, пока не выполнится условие или размер шага не достигнет одного кадра. Этот же алгоритм обеспечивает и последующее постепенное восстановление размера шага до максимального расчетного путем его удвоения.

Механизм отсекающей части изображения, не относящейся к общей зоне видимости, потенциально дает два преимущества. Первое — сокращение времени обработки на этапе сопоставления особенностей за счет меньших размеров обрабатываемых изображений. Второе основное преимущество — эта процедура служит также и фильтром, поскольку естественным образом исключает возможные ошибочные сопоставления особенностей на неперекрывающихся областях, что, в конечном счете, приводит к повышению точности вычислений локальных преобразований. Последующие эксперименты подтвердили это предположение. Относительную величину  $k$  отсекаемой части изображения можно получить из анализа зон видимости для двух соседних позиций (рис. 2) и с учетом параметров, получаемых с помощью вышеописанной методики вычисления адаптивного шага:  $k = S/H$ , здесь  $S$  и  $H$  — как и выше, перемещение камеры и линейный размер зоны видимости. Согласно формуле (1)  $S = VT = (V_{pre} + AT/2)T$ , здесь  $V$  — средняя скорость на текущем шаге,  $V_{pre}$  — скорость в предыдущей позиции,  $T$  — время текущего шага,  $A$  — ускорение на предыдущем шаге. Тогда отсекаемая часть изображения  $cutimage = himage \cdot k$ , здесь  $himage$  — размер изображения по оси в направлении движения. В изображениях стереопары предыдущей позиции отсекается часть снизу, а в изображениях стереопары текущей позиции отсекается часть сверху.

Аналогичным образом можно исключать из рассмотрения особенности, которые находятся в части изображения, не относящейся к общей зоне видимости между камерами стереопары.

#### 4. Обсуждение результатов экспериментов

Вычислительные эксперименты проводили для ряда модельных сцен с траекториями разной сложности, длины и высоты ее прохождения над дном (рис. 3), а также с разными формами рельефа и текстурами дна. Использовался компьютер PC Intel® Core™i5 CPU @280GHz с графической платой NVIDIA GeForce GTX 470 в среде ОС WINDOWS. Базовое расстояние между камерами стереопары

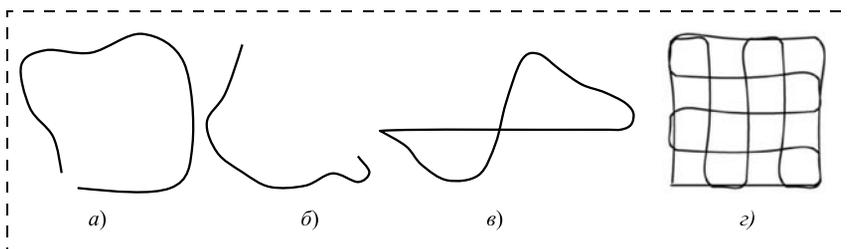


Рис. 3. Тестируемые траектории АПР в модельных сценах (в проекции на плоскость морского дна). Указаны размеры по осям  $X$  и  $Y$  и расстояния по оси  $Z$  до дна:  $a$  — сцена: размеры  $11 \times 13$  м, расстояние  $4...10$  м, время движения  $130$  с;  $b$  — сцена 2: размеры  $80 \times 80$  м, расстояние  $3,5...8$  м, время движения  $240$  с;  $c$  — сцена 3: размеры  $40 \times 20$  м, расстояние  $3,7...10,8$  м, время движения  $200$  с;  $d$  — сцена 4: размеры  $40 \times 40$  м, расстояние  $2,3...9,8$  м, время движения  $1035$  с

Варианты реализации метода визуальной навигации			Усредненная по всей траектории абсолютная ошибка смещения (м) (время счета (с))			
			Сцена 1	Сцена 2	Сцена 3	Сцена 4
Базовый вариант (фиксированный шаг = 16 кадров)			0,077 ( $T = 181 >$ времени движения)	0,102 ( $T = 279 >$ времени движения)	0,121 ( $T = 210 >$ времени движения)	0,336 ( $T = 1473 >$ времени движения)
Адаптивный шаг	Режим пост-обработки	Без отсечения С отсечением	0,072 (18) 0,042	0,088 (62) 0,075	0,119 (22) 0,102	0,252 (328) 0,194
	Режим реального времени	Без отсечения С отсечением	0,062 ( $T <$ времени движения) 0,056 ( $T <$ времени движения)	0,092 ( $T <$ времени движения) 0,081 ( $T <$ времени движения)	0,117 ( $T <$ времени движения) 0,098 ( $T <$ времени движения)	0,329 ( $T <$ времени движения) 0,207 ( $T <$ времени движения)

виртуального АПР бралось равным 40 см. Захват кадров осуществлялся с частотой 25 кадров/с. Разрешение захватываемых камерой изображений составляло  $600 \times 400$  и  $800 \times 600$  пикселей. Степень перекрытия для алгоритма адаптивного шага устанавливалась в 50...75 %. Точность метода оценивали путем сравнения параметров вычисленной и истинной траектории для разных вариантов алгоритмической реализации. В качестве исходного варианта для сравнения рассматривался метод, описанный в разделе "Базовый метод" настоящей статьи, который близок к традиционной реализации.

Оценивали абсолютную ошибку точности (отклонение параметров вычисленной траектории от истинной в МСК) и локальную (ошибку при перемещении АПР между двумя соседними вычисляемыми позициями АПР). Подготовку модельных сцен выполняли с помощью разработанного авторами программного моделирующего комплекса [11]. На этапе вычисления матрицы локального преобразования по двум 3D-облакам применяли средства оптимизации библиотеки MATLAB и, в качестве альтернативы, аналогичные средства из библиотеки MRPT в комбинации с RANSAC. На рис. 4 (см. вторую сторону обложки) показан вид одной из модельных сцен. Результаты вычислительных экспериментов для этих сцен в разных режимах приведены в таблице. Заметим, что указываемый в таблице режим реального времени адаптивного шага соответствует условиям реальной работы программы на борту АПР (при вычислении очередной позиции траектории число доступных для обработки кадров ограничивается числом накопленных кадров в буфере к данному моменту времени). В таблице приведе-

ны усредненные по траектории значения ошибки вычисления пространственного положения АПР и значения времени расчета траектории на компьютере. Заметим, что наиболее вычислительно затратным этапом работы метода является этап сопоставления особенностей — он занимает около 90 % от общего времени счета.

На рис. 5 показаны графики поведения абсолютной и локальной ошибок в вычислении положения и ориентации для метода с фиксированным и адаптивным шагом в режиме реального времени на примере сцены 1. В качестве ошибки ориентации оценивалось усредненное расхождение по углам ориентации камеры. Здесь используется адаптивный шаг без отсечения изображений. Поскольку для фиксированного шага (базовый вариант) и

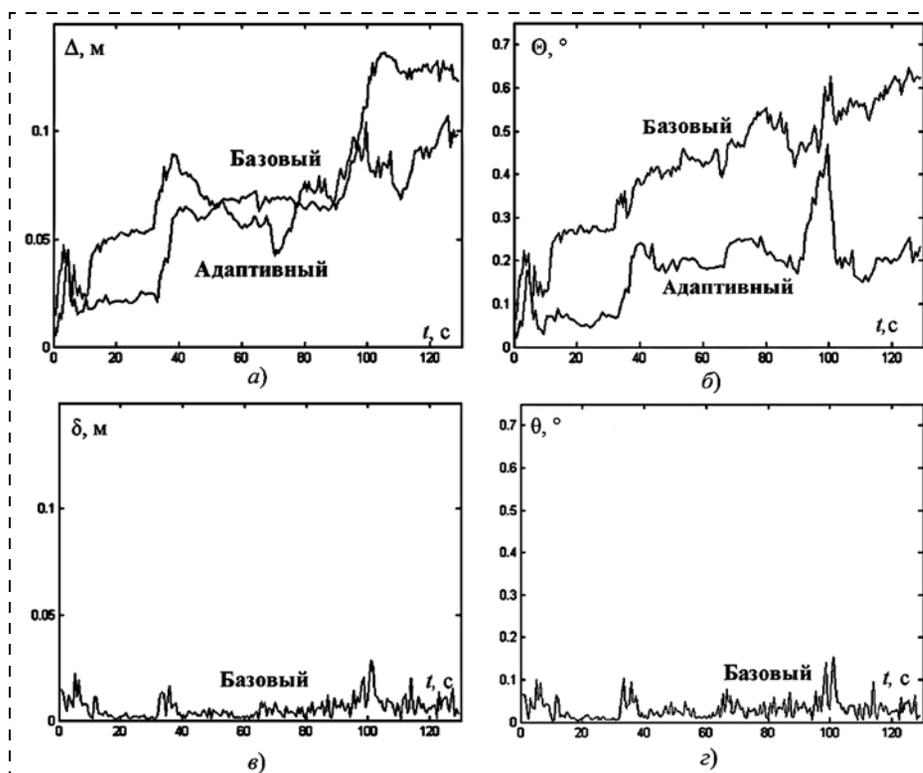


Рис. 5. Точность вычисления траектории для метода с фиксированным и адаптивным шагом в режиме реального времени в зависимости от времени движения АПР (сцена 1): а — абсолютная ошибка  $\Delta$  в вычислении положения; б — абсолютная ошибка  $\Theta$  в вычислении ориентации; в — локальная ошибка  $\delta$  в вычислении положения; г — локальная ошибка  $\theta$  в вычислении ориентации

адаптивного шага локальные ошибки смещения и ориентации близки, на рис. 5 приведены графики только для базового варианта. Число сопоставленных особенностей при вычислении локальных перемещений изменялось в диапазоне "пороговое значение — несколько сот". Сравнительные результаты измерений для разных сцен с разными режимами и модификациями программы навигации приведены в таблице.

Как видно из таблицы, для сцены 1 адаптивный шаг обеспечивает режим реального времени, в отличие от варианта с фиксированным шагом, где время расчета превышает время движения АПР по траектории. В режиме постобработки время счета значительно сокращается за счет "длинных" шагов. Для сцен 1 и 3 это время почти на порядок меньше времени движения АПР. При этом достигаемая точность навигации для адаптивного шага в обоих режимах, как видно из таблицы, выше точности, получаемой при фиксированном шаге. Также в таблице можно увидеть положительный эффект от применения алгоритма отсечения кадров. Очевидным образом, в большей степени он проявляется при длительных перемещениях АПР. Эксперименты с траекторией движения большой длины и сложной формы (сцена 4) показали, что метод визуальной навигации достаточно устойчив к накоплению абсолютной ошибки. Анализ графиков зависимости абсолютной ошибки от времени также показывает, что существует начальный участок траектории с медленным ростом ошибки. Отсюда следует, что локальные маневры небольшой продолжительности могут осуществляться в режиме визуальной навигации с достаточно высокой точностью. Эксперименты показали, что вариант с MATLAB дает лучший результат по точности, чем вариант с применением MRPT + RANSAC (в зависимости от разных сцен — в 1,3...1,6 раза), уступая при этом по времени вычислений 5...10 %. Дополнительные эксперименты были проведены с использованием: а) детектора FREAK, который строит дескриптор, альтернативный дескриптору SURF; б) детектора FAST Corner Detection. Они ускоряют вычисления до 2 раз, снижая при этом точность примерно до полутора раз. Также экспериментально было выявлено, что применение алгоритма вычисления диспарантности вместо SURF внутри каждой из двух стереопар изображений дает несколько лучший конечный результат по точности и по времени счета. Однако этот алгоритм чувствителен к ошибкам калибровки. Во всех вышеописанных модельных экспериментах использовалась версия OpenCV без CUDA-ускорения. Отдельный сравнительный эксперимент показал, что применение CUDA-версии OpenCV сокращает счетное время на этапе сопоставления особенностей примерно в 9 раз для используемых аппаратных средств.

В целом, анализ результатов экспериментов показывает, что:

- адаптивная методика значительно сокращает время вычислений в режиме постобработки (до

10 раз) и позволяет визуальному методу поддерживать режим реального времени за счет оптимальной обработки захватываемых при движении АПР видеок кадров, не ухудшая при этом навигационную точность. В среднем ошибка локальной точности во всех экспериментах меньше 1 см;

- алгоритм отсечения кадров, работая как фильтр, может уменьшить ошибку навигации до 30 % и сократить время вычислений;
- использование аппаратного ускорения графической платы с реализацией параллельных вычислений может существенно ускорить on-line-расчет траектории, давая при этом возможность применять и более вычислительно емкие программно-алгоритмические средства.

### Заключение

Полученные результаты по разработке адаптивного метода визуальной навигации и оценке его эффективности позволяют сделать вывод о том, что предложенный метод может быть полезным дополнением и/или альтернативой традиционным средствам навигации автономных подводных аппаратов, особенно в ситуациях локального маневрирования, в целом способствуя повышению качества решения задачи управления движением АПР.

### Список литературы

1. **Щербатюк А. Ф., Дубровин Ф. С.** Алгоритмы определения местоположения АНПА на основе информации о дальности до одного мобильного гидроакустического маяка // Информационно-измерительные и управляющие системы. 2012. № 9. С. 26—39.
2. **Борейко А. А., Мун С. А., Щербатюк А. Ф.** Определение движения подводного аппарата на основе обработки видео изображений // Мехатроника, автоматизация, управление. 2008. № 8. Приложение. С. 2—8.
3. **Kim A., Eustice R. M.** Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency // IEEE Transactions on Robotics. 2013. N. 29 (3). P. 719—733.
4. **Бобков В. А., Борисов Ю. С.** Навигация подводного аппарата на малых дистанциях по оптической информации // Мехатроника, автоматизация, управление. 2010. № 2. С. 75—78.
5. **Salvi J., Petillot Y., Battle E.** Visual SLAM for 3D Large Scale Seabed Acquisition Employing Underwater Vehicles // IEEE International Conference on Intelligent Robots and Systems. 2008. P. 1011—1016.
6. **Eustice R., Singh H., Leonard J., Walter M.** Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters. International Journal of Robotics Research. 2006. N. 25 (12). P. 1223—1242.
7. **Johnson-Roberson M., Pizarro O., Williams S. B., Mahon I.** Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys // Journal of Field Robotics, Special Issue: Three-Dimensional Mapping. Part 3. 2010. Vol. 27. P. 21—51.
8. **Stoyanov T., Mojtahedzadeh R., Andreasson H., Lilienthal A. J.** Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications // Robotics and Autonomous Systems. 2013. Vol. 61. P. 1094—1105.
9. **Bobkov V. A., Ron'shin Yu. I., Kudryashov A. P., Mashentsev V. Yu.** 3D SLAM from Stereoimages // Programming and Computer Software. 2014. Vol. 40. N. 4. P. 159—165.
10. **Бобков В. А., Машенцев В. Ю.** Визуальная навигация подводного аппарата для целей локального маневрирования // Подводные исследования и робототехника. 2013. № 2 (16). С. 33—37.
11. **Бобков В. А., Морозов М. А., Багницкий А. В., Инзарцев А. В., Павин А. М., Щербатюк А. Ф.** Имитационный моделирующий комплекс для обследовательского автономного подводного робота // электронный журнал "Научная визуализация". 2013. Т. 5. № 4. С. 47—70.

# Navigation of an Underwater Robot by Stereo Images

**V. A. Bobkov**, bobkov@iacp.dvo.ru✉, Institute of Automation and Control Processes of the Far Eastern Branch of the Russian Academy of Sciences, Vladivostok, 690041, Russian Federation  
**V. Yu. Mashentsev**, v.mashentsev@gmail.com, Far Eastern Federal University, Vladivostok, 690091, Russian Federation

Corresponding author: **Bobkov Valery A.**, D. Sc., Head of the Computer Graphics Laboratory, Institute of Automation and Control Processes of the Far Eastern Branch of the Russian Academy of Sciences, Vladivostok, 690041, Russian Federation, e-mail: bobkov@iacp.dvo.ru

Received on May 15, 2015

Accepted on June 09, 2015

The work is devoted to development of a visual navigation method for an autonomous underwater robot (AUR) using stereo vision. The authors describe the basic principles of implementation of the visual odometry method in accordance with the already established concept of computing, the main stages of which are comparison of the images' features and calculation of the geometric transformation between the neighboring positions of AUR. Two possible modes of operation of a computer program are considered: an on-line mode — the program runs directly on the onboard computer when AUR is moving, and it processes the captured video stream as it becomes available; and a post-processing mode — the program runs on a stationary computer and processes the saved video stream of the whole trajectory. In the first case initially we solve the problem of calculation for each position of the trajectory with the highest possible precision of AUR localization on the current limited sequence of images, and then a problem of provision of real time mode in general. In the second case the task is to ensure a high-precision calculation of AUR localization and increase the speed of calculation through the optimal selection and calculation sequence of the key frames in a path, using the existing video on the entire trajectory. Post-processing is essentially a solution to the task of a full calibration of the sequence of the key frames, which is necessary for solving of the subsequent problem of 3D reconstruction of an underwater scene. A modification of the basic method is suggested. It consists in development and application of an adaptive technique for calculation of AUR trajectory aimed at solving the above problems. It guarantees the desired area of the overlap of the adjacent frames, takes into account the shape of the trajectory and the height of its passage over the sea-floor, driving the dynamic model of AUR, and takes into account the dependence of the number of the actual incoming frames during the movement of the vehicle. Three co-operating techniques realize adaptability: a) selection of the size of the next step, depending on the degree of the overlapping areas of visibility of the stereo camera for two adjacent positions; b) reduction of the step, if the number of the matching features is smaller than the predetermined threshold; and c) cutting off those parts of the image which are not related to the overlap zone of visibility. These techniques provide an optimal way for determination of the sequence of the key frames, in which calculation of the geometric transformation for the local movements is performed. Optimality means achievement of the highest possible accuracy of localization of computation of AUR with the greatest step length in order to ensure the desired overlapping of the adjacent visibility positions. Computation of the local transformation matrixes is carried out proceeding from the condition of the minimal divergences of the points of the two matching 3D clouds. Comparative results of the computational experiments on estimation of the effectiveness of the proposed adaptive method are presented and analyzed. Alternative versions of the individual stages of the general computational scheme are also analyzed. In general, the method can be a useful addition to the traditionally used sonar navigation systems, or a good alternative to them in the conditions of local maneuvering.

**Keywords:** autonomous underwater robot, stereo video stream, visual navigation, visual odometry, adaptive method

**Acknowledgements:** This work was supported by RFBR (project № 15-07-00341) and the Far East Program (project 15-1-4-011 o)

For citation:

**Bobkov V. A., Mashentsev V. Yu.** Navigation of an Underwater Robot by Stereo Images, *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2016, vol. 17, no. 1, pp. 101–109.

DOI: 10.17587/mau/17.101-109

## References

1. **Shherbatjuk A. F., Dubrovin F. S.** *Algoritmy opredeleniya mestopolozheniya ANPA na osnove informacii o dal'nosti do odnogo mobil'nogo gidroakusticheskogo majaka* (The algorithms AUV positioning based on information about the range of up to one mobile sonar beacon), *Informacionno-Izmeritel'nye i Upravljajushhie Sistemy*, 2012, no. 9, pp. 26–39 (in Russian).
2. **Borejko A. A., Mun S. A., Shherbatjuk A. F.** *Opredelenie dvizheniya podvodnogo apparata na osnove obrabotki video izobrazhenij* (Positioning underwater vehicle based on the processing of video images), *Mekhatronika, Avtomatizatsiya, Upravlenie, Prilozhenie*, 2008, no. 8, pp. 2–8 (in Russian).
3. **Kim A., Eustice R. M.** Real-Time Visual SLAM for Autonomous Underwater Hull Inspection Using Visual Saliency, *IEEE Transactions on Robotics*, 2013, no. 29 (3), pp. 719–733.
4. **Bobkov V. A., Borisov Ju. S.** *Navigacija podvodnogo apparata na malyh distancijah po opticheskoj informacii* (Underwater vehicle navigation on small distances from optical data), *Mekhatronika, Avtomatizatsiya, Upravlenie*, 2010, no. 2, pp. 75–78 (in Russian).
5. **Salvi J., Petillot Y., Battle E.** Visual SLAM for 3D Large — Scale Seabed Acquisition Employing Underwater Vehicles, *IEEE International Conference on Intelligent Robots and Systems*, 2008, pp. 1011–1016.
6. **Eustice R., Singh H., Leonard J., Walter M.** Visually mapping the RMS Titanic: Conservative covariance estimates for SLAM information filters, *International Journal of Robotics Research*, 2006, no. 25 (12), pp. 1223–1242.
7. **Johnson-Roberson M., Pizarro O., Williams S. B., Mahon I.** Generation and visualization of large-scale three-dimensional reconstructions from underwater robotic surveys, *Journal of Field Robotics*, Special Issue: *Three-Dimensional Mapping*, part 3, 2010, vol. 27, pp. 21–51.
8. **Stoyanov T., Mojtahedzadeh R., Andreasson H., Lilienthal A. J.** Comparative evaluation of range sensor accuracy for indoor mobile robotics and automated logistics applications, *Robotics and Autonomous Systems*, 2013, vol. 61, pp. 1094–1105.
9. **Bobkov V. A., Ron'shin Yu. I., Kudryashov A. P., Mashentsev V. Yu.** 3D SLAM from Stereoimages, *Programming and Computer Software*, 2014, vol. 40, no. 4, pp. 159–165.
10. **Bobkov V. A., Mashentsev V. Ju.** *Vizual'naja navigacija podvodnogo apparata dlja celej lokal'nogo manevrirovaniya* (Visual navigation of underwater vehicle for local maneuvering), *Podvodnye Issledovanija i Robototekhnika*, 2013, no. 2 (16), pp. 33–37 (in Russian).
11. **Bobkov V. A., Morozov M. A., Bagnickij A. V., Inzartsev A. V., Pavin A. M., Shherbatjuk A. F.** *Imitacionnyj modelirujushhij kompleks dlja obsledovatel'skogo avtonomnogo podvodnogo robota* (Simulation system for underwater research vehicle), *Nauchnaja vizualizatsija*, 2013, vol. 5, no. 4, pp. 47–70 (in Russian).