

УДК 519.673

DOI: [10.26102/2310-6018/2023.41.2.009](https://doi.org/10.26102/2310-6018/2023.41.2.009)

## 3D реконструкция объектов по видеопотоку в динамической сцене

Е.М. Логачев

*Дальневосточный федеральный университет,  
Владивосток, Российская Федерация  
[logachev.em1997@gmail.com](mailto:logachev.em1997@gmail.com)*

**Резюме.** Статья посвящена задаче 3D реконструкции объектов в динамических сценах по стереоизображениям. При выполнении каких-либо сложных задач автономными роботами (ремонтные работы, обследование морского дна) появляется необходимость одновременного восстановления траектории движения автономного робота и построения 3D модели окружающей обстановки по видеоинформации. Данные о траекториях движения роботов и информация об окружающей обстановке необходимы специалистам для дальнейшей работы по корректировке работы дронов и отслеживания прогресса выполняемых работ. Существующие на данный момент решения по идентификации объектов позволяют восстанавливать геометрию динамических объектов с накладываемыми ограничениями, которые не позволяют с необходимой точностью реконструировать всю сцену. Также существующие методы не предполагают детальную визуализацию всей 3D сцены по заранее неизвестным точечным данным и не включают в себя восстановление невидимых участков поверхностей объектов. Предложен подход к решению задачи идентификации и 3D реконструкции объектов по видеоинформации применительно к динамическим сценам. Описан базис программной системы, реализующий предложенные алгоритмические и архитектурные решения. Приведены данные по модельным сценам и особенностям объектов сцены. Обсуждены результаты вычислительных экспериментов с виртуальными сценами. Рассмотрены закономерности, выявленные в результате проведения тестов, влияющие на точность реконструкции моделей.

**Ключевые слова:** динамическая сцена, идентификация объектов, OpenGL, 3D-реконструкция, визуализация, эпиполярные ограничения, метод Делоне.

**Для цитирования:** Логачев Е.М. 3D реконструкция объектов по видеопотоку в динамической сцене. *Моделирование, оптимизация и информационные технологии.* 2023;11(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1334> DOI: 10.26102/2310-6018/2023.41.2.009

## 3D reconstruction of objects by video stream in a dynamic scene

E.M. Logachev

*Far Eastern Federal University,  
Vladivostok, the Russian Federation  
[logachev.em1997@gmail.com](mailto:logachev.em1997@gmail.com)*

**Abstract.** The article is devoted to the problem of 3D reconstruction of objects in dynamic scenes by stereo images. When performing any complex tasks by autonomous robots (repair work, inspection of the seabed), there is a need to simultaneously restore the trajectory of the autonomous robot and build a 3D model of the environment using video information. Data on the trajectories of robots and information about the environment are necessary for specialists to further correct drone operation and track the progress of work performed. Currently existing object identification solutions help to restore the geometry of dynamic objects with imposed restrictions that prevent from reconstructing the entire scene with the necessary accuracy. Also, the existing methods do not involve detailed visualization of the entire 3D scene using previously unknown point data and do not include the restoration of invisible parts

of object surfaces. An approach to solving the problem of identification and 3D reconstruction of objects based on video information in relation to dynamic scenes is proposed. The basis of the software system implementing the proposed algorithmic and architectural solutions is described. Data on model scenes and features of scene objects are given. The results of computational experiments with virtual scenes are discussed. The regularities discovered as a result of tests affecting the accuracy of model reconstruction are considered.

**Keywords:** dynamic scene, object identification, OpenGL, 3D reconstruction, visualization, epipolar constraints, Delaunay method.

**For citation:** Logachev E.M. 3D reconstruction of objects by video stream in a dynamic scene. *Modeling, Optimization and Information Technology*. 2023;11(2). URL: <https://moitvvt.ru/ru/journal/pdf?id=1334> DOI: 10.26102/2310-6018/2023.41.2.009 (In Russ.).

## Введение

Задача одновременного восстановления траектории движения автономного робота и построения 3D модели окружающей обстановки по видеоинформации является одной из важных в науке. Особенно актуальна данная задача в компьютерном зрении и робототехнике, которые активно используются в различных отраслях с применением распознавания объектов в естественной среде, где существует ряд ограничений для распознавания объектов. В настоящее время существует множество решений для обработки статических сцен. Однако разработанные методы не позволяют работать с динамическими сценами (сценами с движущимися объектами) с достаточной точностью, либо позволяют работать, но с рядом ограничений. Также существующие методы не предполагают детальную визуализацию всей 3D сцены по заранее неизвестным точечным данным и не включают в себя восстановление невидимых участков поверхностей объектов. Данный аспект моделирования важен для расчетов размеров объектов и дальнейшего вычисления маршрута робота.

Существующие на данный момент решения по идентификации объектов позволяют восстанавливать геометрию динамических объектов с накладываемыми ограничениями, которые не позволяют с необходимой точностью реконструировать всю сцену. Большинство существующих работ базируются на контролируемом окружении, где известен фон и используются калиброванные и фиксированные камеры. Однако существуют и более общие подходы, которые ограничиваются только сложностью сцены и уровнем реконструкции [1-4].

Представляется эффективный в плане алгоритмической простоты, сравнительно небольших вычислительных затрат и возможности использовать стандартное графическое ускорение подход, основанный на точечном представлении объектов и рассмотренный в работе [5].

Наиболее эффективные подходы, которые позволяют точнее реконструировать сцены, используют в своей работе темпоральную когерентность и multi-view сегментацию, а также вычисление кривизны поверхностей [6-7].

В настоящей статье описываются результаты по реализации базиса программной системы 3D-реконструкции и визуализации объектов в динамических сценах с восстановлением невидимых участков поверхностей объектов. В основу программной реализации системы положена алгоритмическая база и вычислительная схема, предложенные в [8-15]. Методология работы основывается также на работах по реконструкции 3D-сцен на пучках эпиполярных плоскостей стереоизображений и реконструкции 3D-сцен по неректифицированным стереоизображениям с прореживающей фильтрацией [16-17] с применением алгоритмов построения триангуляции Делоне [18-19].

## Материалы и методы

Задача заключается в разработке программной системы 3D-реконструкции и визуализации объектов в динамических сценах. Входной информацией для системы является видеопоток, фиксируемый стереокамерой, установленной на автономном подводном роботе (АПР). Сложность задачи заключается в том, что в процессе работы АПР в неизвестной среде необходимо вычислять траекторию движения не только АПР, но и траектории движения динамических объектов (ДО) окружающей среды с присущими им геометрическими формами, которые заранее неизвестны. Разрабатываемая система основывается на подходе и алгоритмической базе, предложенной в [10]. Согласно подходу для описания и опознавания объектов применяется 3D точечное представление объектов. Для оценки движения и восстановления траекторий по изображениям применяется метод визуальной навигации (визуальная одометрия):

- установление соответствий 2D особенностей на парах последовательных снимков, соответствующих двум соседним расчетным позициям на траектории;
- построение по сопоставленному на стереопаре текущей позиции множеству особенностей соответствующего множества 3D точек (3D облако). Аналогичным образом строится 3D облако для стереопары предшествующей позиции. Каждое 3D облако координируется в локальной системе координат позиции;
- вычисление по сопоставленным 3D облакам матрицы геометрического преобразования, описывающего относительное перемещение камеры / робота (6 DOF (degrees of freedom)) в локальной системе координат.

Вычислительная схема подразумевает многоэтапную обработку, которая включает выделение групп точек, принадлежащих «статическим» и ДО, расчет траектории движения камеры и траекторий движения ДО, а также 3D реконструкцию объектов с восстановлением невидимых участков поверхностей объектов.

С учетом вышесказанного разработка программной системы требует решения следующих основных задач:

1. Разработка структуры и функциональной схемы системы.
2. Программная реализация функционала базиса системы с применением метода визуальной навигации, алгоритмов 3D реконструкции и прослеживания движения ДО.
3. Тестирование основных функций системы с оценкой их эффективности.

Реализация подхода происходит в два этапа, на каждом из которых происходит обработка точечных данных сцены (включая ДО и объекты статики) на каждом из анализируемых кадров. На первом этапе делается разбиение точек на группы, предположительно относящиеся к «статике» и к ДО. Отбор точек осуществляется применением двух критериев: критерия «связности» и критерия «жесткости». Критерий «связности» позволяет оценивать непрерывность поверхности объекта, а критерий «жесткости» – оценивать движение объекта как твердого тела. Выделение статической части сцены позволяет вычислить траекторию движения камеры (т. е. получить матрицу «статики»). Прослеживание движения ДО во времени обеспечивается алгоритмом с применением критерия сходства движений.

Критерий «жесткости» строится исходя из двух положений:

1. Расстояние между двумя точками недеформируемого / жесткого объекта сохраняется при его движении.
2. В качестве точки, заведомо принадлежащей конкретному объекту, будет рассматриваться его «затравочная» точка.

Обработка точек исходного множества 3D точек начинается с затравочной точки (заведомо принадлежащей объекту). Проверка критерия «связности» состоит из

нескольких этапов. На первом этапе сравниваются расстояния от камеры до каждой пары соседних точек в текущей позиции (одна из данных точек принадлежит какой-либо группе точек объекта).

На втором этапе происходит объединение групп точек, сформированных на первом этапе. Данный шаг позволяет определить, относятся ли группы точек к одному объекту или к разным. Это необходимо для четкого разделения объектов статики и ДО, с восстановлением траекторий их движения.

Для вычисления локальных матриц, необходимых для определения собственного движения каждого из объектов сцены, используется уже вычисленная матрица «статики». Реализующую данный подход вычислительную схему можно представить следующим образом:

- 1) построение множества 3D точек в текущей позиции по результатам работы детектора SURF;
- 2) генерация затравочных точек в ячейках пространственной сетки видимого камерой пространства;
- 3) начальное разделение точек исходного 3D множества на группы алгоритмом с применением критерия «жесткости»;
- 4) идентификация группы «статика» соответствующим алгоритмом;
- 5) объединение схожих групп точек. Вычисление матрицы движения камеры по точкам группы «статика»;
- 6) разделение точек исходного множества на два множества «статика» и «динамика» соответствующим алгоритмом;
- 7) фильтрация множества точек «статика» и множества точек «динамика»;
- 8) выделение групп точек, относящихся к разным динамическим объектам (ДО).

При реализации алгоритма идентификации точек применялся метод сортировки массива слиянием (merge sort) и поиск элемента в сортированном массиве методом деления отрезка пополам. Реконструкция выполнялась для недеформируемых ДО. Все используемые 3D множества исходных точек и матрица движения камеры на каждом шаге – идеальные. Таким образом, в программе моделирования соответствующими алгоритмами формируются точечные модели ДО. По ним алгоритмом ИСР на каждом шаге для каждого ДО вычисляется комплексная матрица, представляющая совместное движение камеры и ДО (вычисляется по 3D точкам, принадлежащим ДО), и, соответственно из соотношения

$$H_{i, i+1}^{joint, id\_gr} = H_{i, i+1}^{camera} \cdot H_{i, i+1}^{id\_gr} \quad (1)$$

вычисляется матрица собственного движения ДО-матрица:

$$H_{i, i+1}^{id\_gr} = H_{i, i+1}^{joint, id\_gr} \cdot (H_{i, i+1}^{camera})^{-1}, \quad (2)$$

где:

$H_{i, i+1}^{camera}$  – матрица движения камеры;

$H_{i, i+1}^{joint, id\_gr}$  – матрица комплексного преобразования;

$H_{i, i+1}^{id\_gr}$  – матрица собственного движения.

При работе над реализацией базиса программной системы по 3D-реконструкции и визуализации использован программный интерфейс для графических устройств OpenGL и модифицированные стандартные шаблоны приложений WinAPI. В дополнение к вышеописанному была добавлена библиотека утилит для приложений под OpenGL – GLUT (OpenGL Utility Toolkit).

В результате анализа вычислительной схемы в программной реализации выделены следующие базисные функции для идентификации точек сцены:

1. Функция формирования групп точек предназначена для создания отображения входного видеопотока в последовательность рациональных чисел.

2. Функция группировки точек по схожим признакам предназначена для формирования групп с похожими или одинаковыми признаками. В дальнейшем группы будут использоваться для создания моделей объектов.

3. Функция отслеживания точек / объектов предназначена для отображения положения каждой точки и каждого объекта на каждом кадре.

При решении задачи визуализации точечных данных необходима возможность работы с системой частиц и интерпретации 2D проекций. Однако существующие на данный момент решения предполагают построение моделей на основе сплайнов, примитивов. При работе с точками нет возможности в автоматическом режиме без настройки скриптов системы интерпретировать и объединять их как объекты. Однако существуют возможности задания «анимации» поведения данных частиц. Чаще всего такие системы частиц используются для имитации каких-либо процессов. Примером является система Autodesk 3ds Max.

Поэтому в настоящей работе разработан базис, который включает в себя основные модули по идентификации пространственных точек (отслеживание точек на каждом из кадров, объединение их в группы) и работе с ними (визуализация точечных данных). Данный базис не содержит сложных функций (например, реконструкция невидимых участков 3D моделей объектов) и предназначен для выявления закономерностей и отлаживания работы основных алгоритмов с последующим дополнением сложными функциями.

## Результаты

Основная задача разрабатываемой программной системы – реконструкция объектов динамических сцен по точечным данным, полученным обработкой видеопотока, и визуализация полученного результата в 3D пространстве. В процессе визуализации в 3D пространстве необходимо создавать поток из кадров и преобразовывать их в формат интерактивной сцены. Интерактивная сцена – это сцена с применением возможностей графического движка, которая позволяет в процессе просмотра потоков кадров регулировать положение камеры специалиста.

Для оценки эффективности базиса системы был проведен ряд экспериментов с виртуальными сценами. В сценах построены 3D модели, которые удовлетворяют требуемым параметрам, а также 2D снимки данных моделей в разрешениях 500x500 и 1024x768 пикселей. В рассматриваемых областях с сопоставленным размером 5x5 км расположены объекты статики различной сложности и полигональности. Эксперименты проводились на PC с конфигурацией:

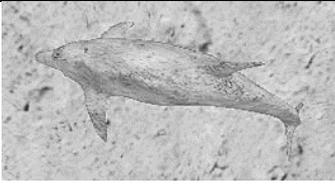
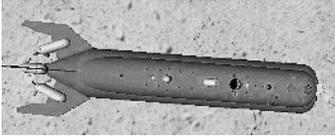
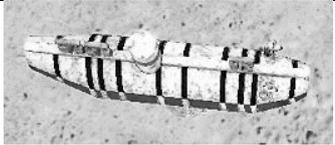
Процессор – Intel(R) Core(TM) i7-7700K CPU @ 4.20GHz 4.20 GHz

Оперативная память – 16,0 ГБ

Видеокарта – NVIDIA GeForce GTX 1080 Ti и NVIDIA GeForce GTX 1650

В тестах использовались 3 динамических объекта (2 малополигональных объекта и 1 многополигональный) и статическая среда на протяжении 127 кадров (Таблица 1).

Таблица 1 – Данные об объектах  
Table 1 – Data about objects

| Визуализация                                                                       | Данные                                                                                             |
|------------------------------------------------------------------------------------|----------------------------------------------------------------------------------------------------|
|   | Название: Дельфин<br>Идентификатор: Объект 3<br>Тип: деформируемый<br>Полигоны: 23746              |
|   | Название: Подводная лодка<br>Идентификатор: Объект 1<br>Тип: недеформируемый<br>Полигоны: 2968     |
|   | Название: Батискаф<br>Идентификатор: Объект 2<br>Тип: недеформируемый<br>Полигоны: 852             |
|  | Название: Морское дно<br>Идентификатор: Объекты статики<br>Тип: недеформируемый<br>Полигоны: 27052 |

На Рисунке 1 представлена модельная подводная сцена. На Рисунке 2 приведена структурная схема области сцены. Данная модель необходима для тестирования в условиях «насыщенной» статики. Здесь объекты статики обладают высокой полигональностью и текстурами высокого разрешения.

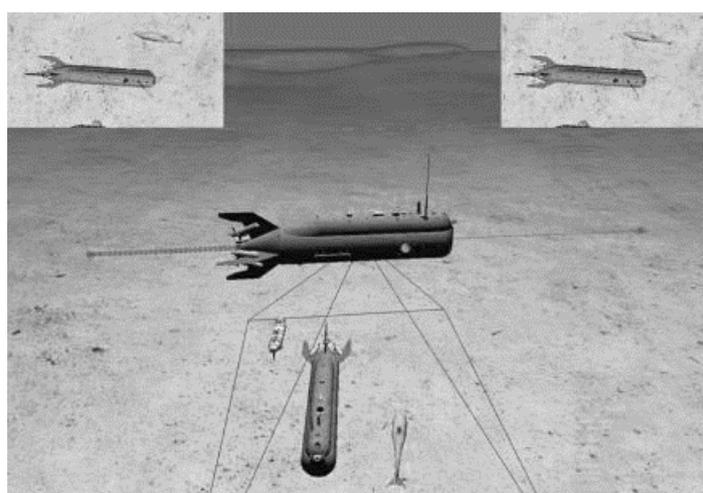


Рисунок 1 – Модельная подводная сцена в одной из позиций траектории камеры АПР: в поле зрения камеры находится 3 динамических объекта на фоне рельефа дна

Figure 1 – Model underwater scene in one of the APR camera trajectory positions: in the field of camera view, there are 3 dynamic objects against the background of the bottom topography

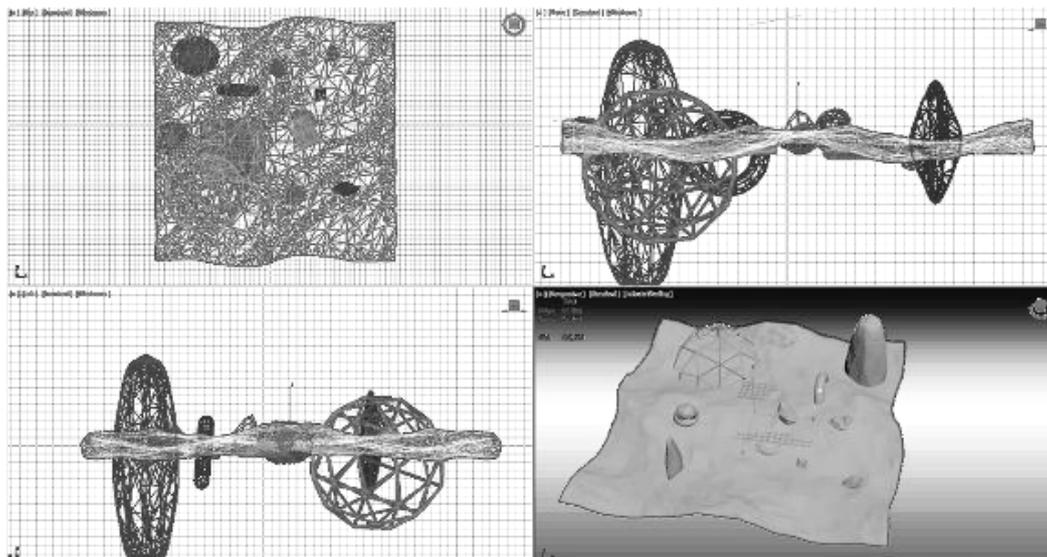


Рисунок 2 – Структурная схема области сцены  
Figure 2 – Survey plan of the scene area

В результате тестирования были выявлены некоторые закономерности и особенности работы алгоритма.

Полученные данные по отдельным кадрам приведены в Таблице 2, которая содержит информацию о номере кадра видеопотока, количестве групп точек (на последнем этапе идентификации), числе точек, которые не удалось идентифицировать однозначно (неизвестно, к какому объекту относятся), информацию о точках статики и точках динамических объектов.

Из таблицы видно, что идентификация объектов происходит на втором кадре после их появления (кадры 9-10 и 19-20). В момент появления объектов фиксируется резкий рост числа неопределенных точек.

Другие результаты показывает многополигональный (критический) объект (кадр 25). Данный объект не вызывает рост числа неопределенных точек и идентифицируется сразу. Далее эта ситуация будет рассмотрена подробнее.

Таблица 2 – Данные об эффективности распознавания групп точек  
Table 2 – Data on the efficiency of point group recognition

| № кадра | Количество групп точек | Неопределённые точки | Объекты статики | Объект 1 | Объект 2 | Объект 3 | Общее количество точек |
|---------|------------------------|----------------------|-----------------|----------|----------|----------|------------------------|
| 0       | 1                      | 13                   | 300             | 0        | 0        | 0        | 313                    |
| 1       | 1                      | 0                    | 305             | 0        | 0        | 0        | 305                    |
| 2       | 1                      | 0                    | 306             | 0        | 0        | 0        | 306                    |
| 9       | 1                      | 499                  | 313             | 0        | 0        | 0        | 812                    |
| 10      | 2                      | 16                   | 305             | 1164     | 0        | 0        | 1485                   |
| 11      | 2                      | 9                    | 309             | 1669     | 0        | 0        | 1987                   |
| 12      | 2                      | 12                   | 306             | 1968     | 0        | 0        | 2286                   |

Таблица 2(продолжение)  
Table 2 (extended)

| № кадра | Количество групп точек | Неопределённые точки | Объекты статики | Объект т 1 | Объект т 2 | Объект т 3 | Общее количество точек |
|---------|------------------------|----------------------|-----------------|------------|------------|------------|------------------------|
| 18      | 2                      | 41                   | 313             | 11536      | 0          | 0          | 11890                  |
| 19      | 2                      | 200                  | 300             | 13154      | 0          | 0          | 13654                  |
| 20      | 3                      | 47                   | 312             | 13259      | 298        | 0          | 13916                  |
| 21      | 3                      | 134                  | 303             | 13364      | 776        | 0          | 14577                  |
| 22      | 3                      | 52                   | 305             | 13633      | 3152       | 0          | 17142                  |
| 23      | 3                      | 74                   | 301             | 13954      | 3956       | 0          | 18285                  |
| 24      | 3                      | 83                   | 295             | 14357      | 4109       | 0          | 18844                  |
| 25      | 4                      | 33                   | 296             | 14664      | 4043       | 5298       | 24334                  |
| 26      | 4                      | 57                   | 295             | 14692      | 3969       | 8699       | 27712                  |
| 27      | 4                      | 75                   | 298             | 14664      | 3910       | 12006      | 30953                  |
| 28      | 4                      | 103                  | 291             | 14677      | 3848       | 38444      | 57363                  |
| 29      | 4                      | 68                   | 296             | 14811      | 3875       | 46394      | 65444                  |
| 63      | 3                      | 309                  | 287             | 14530      | 0          | 114559     | 129685                 |
| 64      | 3                      | 192                  | 279             | 14428      | 0          | 113999     | 128898                 |

Также были проведены оценки работы по ряду критериев:

- 1) производительность реализованных программных решений;
- 2) количество обрабатываемых точек сцены;
- 3) число кадров, необходимых для корректной реконструкции.

Если рассматривать данные о формировании групп точек на разных этапах их объединения (см. Материалы и методы), то получим следующие данные (Рисунок 3).

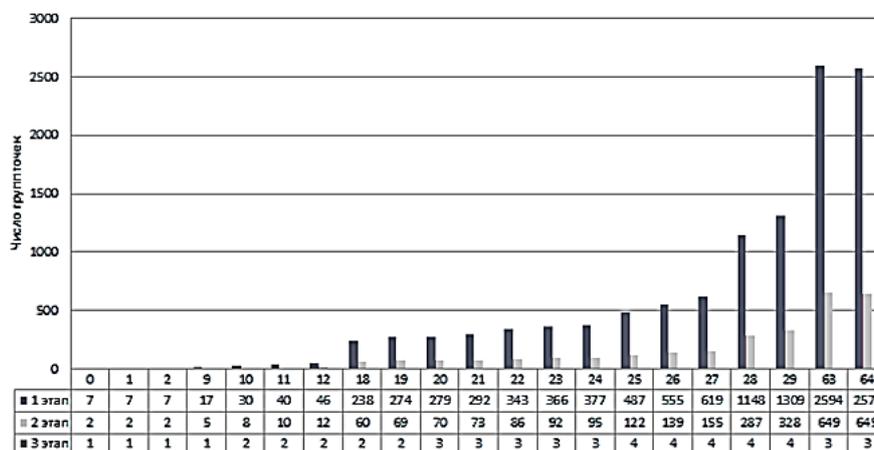


Рисунок 3 – Данные о формировании групп точек на каждом из этапов  
Figure 3 – Data on point group formation at each stage

Из данных показателей видно, что на каждом из кадров обрабатываемого видеопотока зависимость основных этапов идентификации (этапы 1 и 2) в среднем составляет  $\frac{1}{4}$ . Для данной задачи это является хорошим результатом, так как данные показатели иллюстрируют то, что даже в условиях полного отсутствия какой-либо информации о внешней среде удается сгруппировать больше половины точек видеоряда.

### Обсуждение

Для определения причины резкого скачка числа идентифицированных точек, а также основных закономерностей идентификации точек, необходимо детально рассмотреть кадры, на которых происходит появление объектов.

Первое появление **Объекта 1** можно наблюдать на **кадре 9** (Таблица 2 / Рисунок 4). Однако идентификация этого объекта происходит на **кадре 10** (Таблица 2 / Рисунок 5).

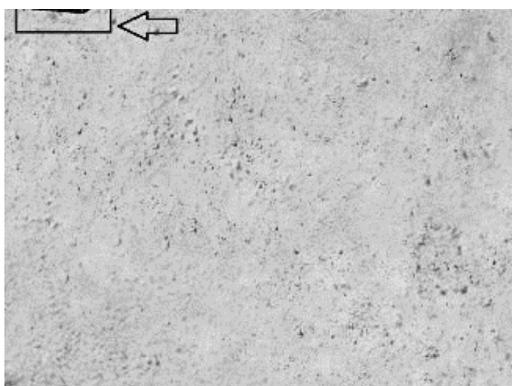


Рисунок 4 – Кадр 9  
Figure 4 – Frame 9

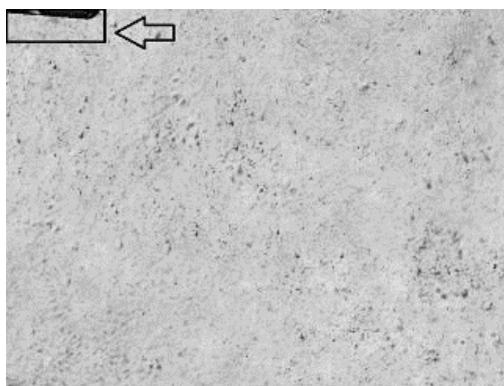


Рисунок 5 – Кадр 10  
Figure 5 – Frame 10

Такая же ситуация происходит и с **Объектом 2** (Таблица 2 / Рисунок 6). Его можно наблюдать впервые на **кадре 19**, но его идентификация происходит на **кадре 20** (Таблица 2 / Рисунок 7).

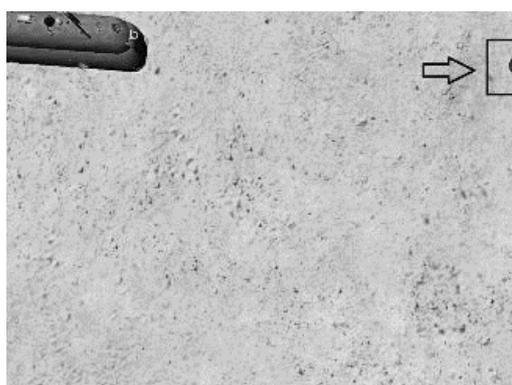


Рисунок 6 – Кадр 19  
Figure 6 – Frame 19

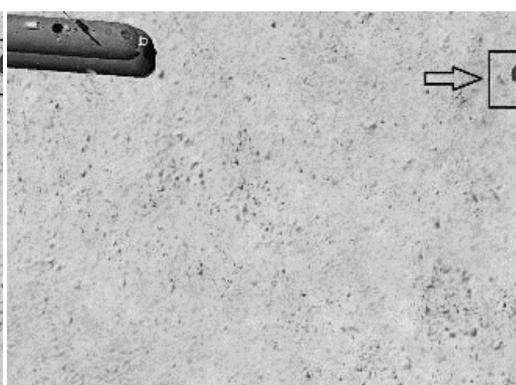


Рисунок 7 – Кадр 20  
Figure 7 – Frame 20

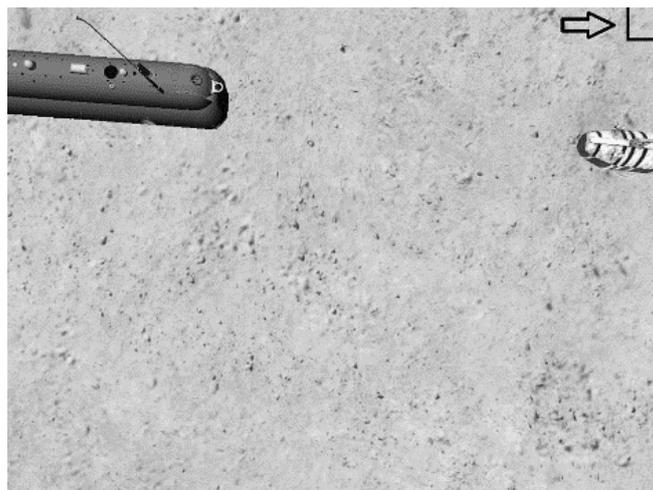


Рисунок 8 – Кадр 25  
Figure 8 – Frame 25

Другая ситуация происходит с **Объектом 3**. Данный объект появляется в **кадре 25** и в этом же кадре идентифицируется программой (Таблица 2 / Рисунок 8).

Из данных можно сделать вывод, что это связано с тем, что Объект 1 и Объект 2 менее детализированные или малополигональные модели (определено меньше точек) в отличие от Объекта 3. При появлении Объекта 1 и Объекта 2 наблюдается резкое увеличение количества неопределенных точек (см. Таблицу 2). В первом случае наблюдается разница в 499 точек, во втором – 159 точек. При появлении же Объекта 3 разница всего составляет 50 точек. Это свидетельствует о том, что для системы для идентификации группы точек как объекта большую роль играет количество точек. Именно это утверждение подтверждает Объект 3 (многополигональный объект). При первом появлении данного объекта на кадре 25 количество определенных точек равно 5298. Этот показатель на порядок выше, чем у Объекта 1 и Объекта 2, 499 и 159 соответственно.

Если рассматривать рост числа точек, то также видно, что рост числа точек у многополигональной модели идет значительно быстрее (Рисунок 9).

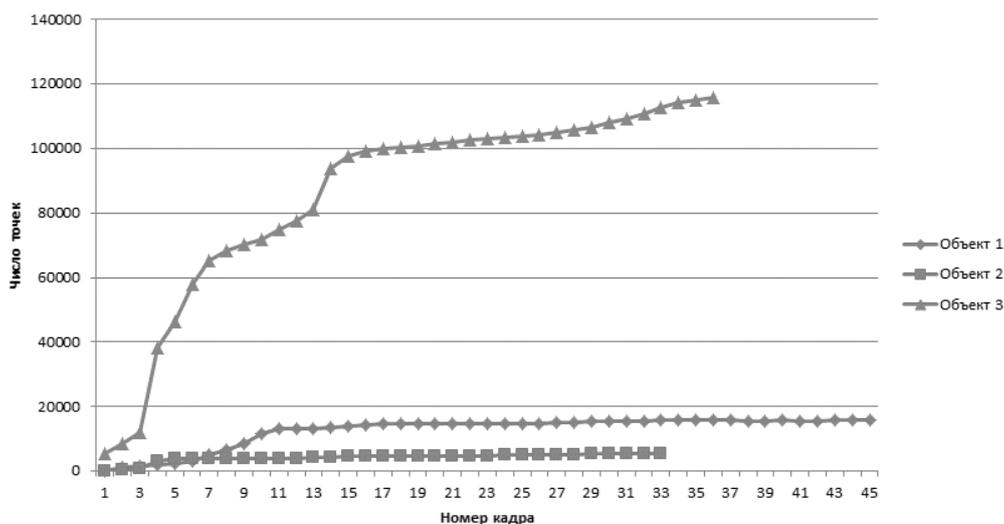


Рисунок 9 – Рост числа точек  
Figure 9 – Increase in the number of points

Прослеживая данные по неопределенным точкам также можно сделать вывод, что их рост фиксируется на кадрах с появлением новых объектов. Скачки в этом показателе наблюдаются в разные моменты времени, но связаны в основном с изменением положений объектов и угла обзора камеры. Сравнение данных приведено на Рисунке 2.

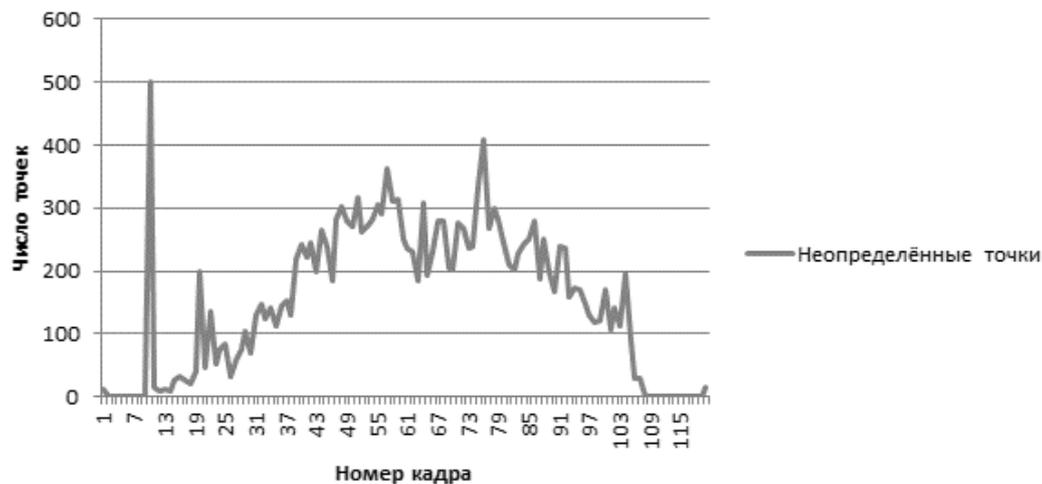


Рисунок 10 – Неопределенные точки  
Figure 10 – Undefined points

Примером изменения показателей, зависящих от положения объекта, могут служить кадры 51 (Рисунок 11), 57 (Рисунок 12). На них можно увидеть изменение положения плавника многополигональной модели, что влечет за собой переопределение данных о большом числе точек.

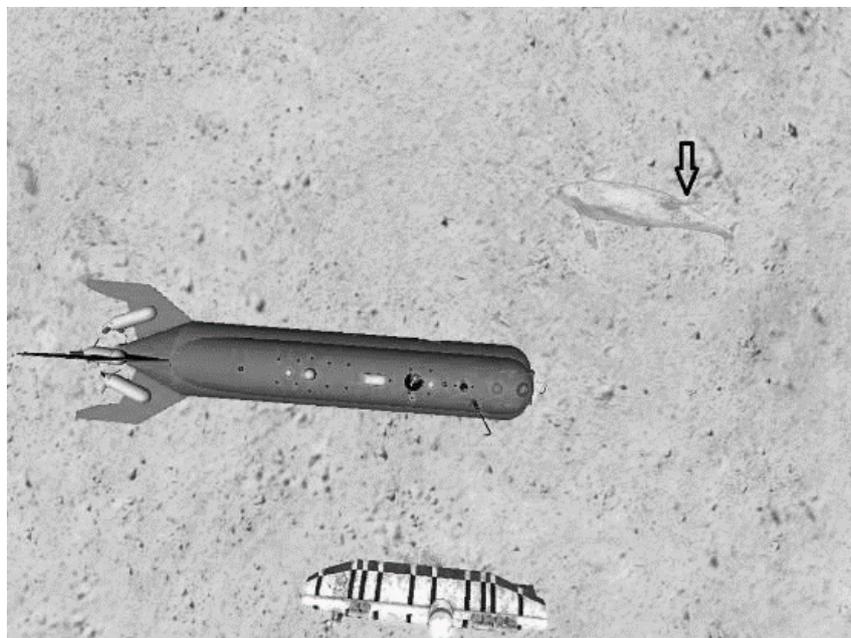


Рисунок 11 – Кадр 51  
Figure 11 – Frame 51

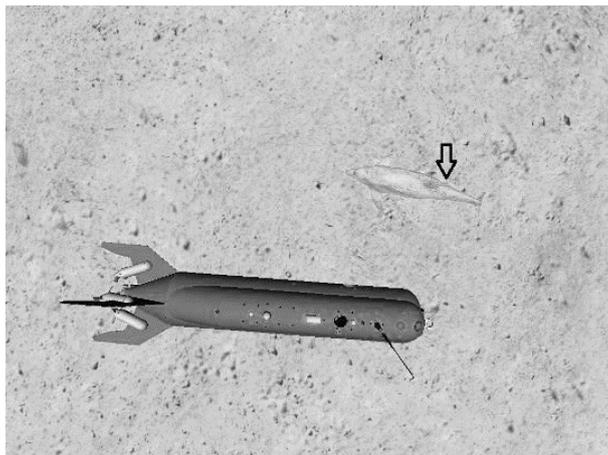


Рисунок 12 – Кадр 57  
Figure 12 – Frame 57

В рамках эксперимента были дополнительно рассмотрены случаи 3D-реконструкции простых примитивов в «чистой» среде, а также сложных объектов в среде, которая приближена к реальной. На Рисунке 13 представлена примитивная 3D-реконструкция сложной сцены, при которой опора шла только на 2 кадра видеопотока. В данном случае были замечены интересные особенности реконструкции. Несмотря на то, что «батискаф» обладает меньшим числом полигонов в контексте 3D модели, чем «дельфин», на изображениях мы можем наиболее четко отследить именно батискаф. Это обусловлено его расположением относительно камеры (он располагается ближе, чем дельфин), его размеры значительно больше, а также он несет в себе наиболее простые геометрические формы, в сочетании с малой полигональностью.

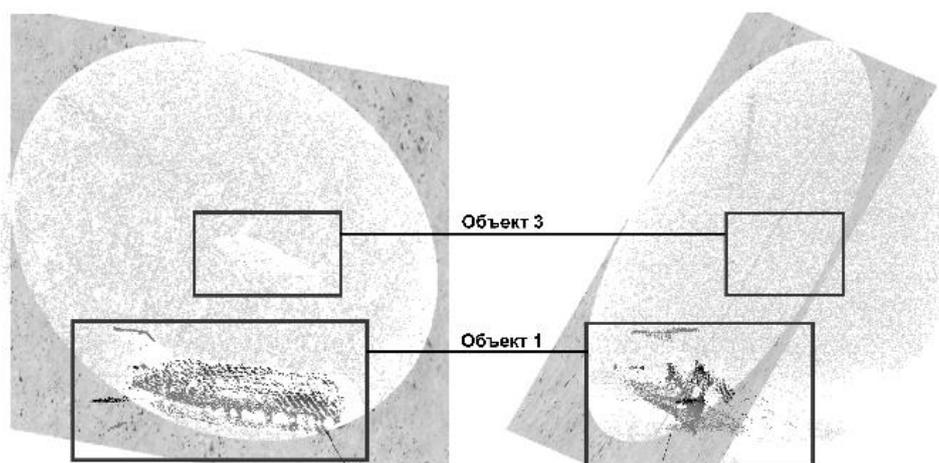


Рисунок 13 – Реконструкция сцены по двум кадрам в разрешении 500x500 (позиция 1, позиция 2)

Figure 13 – Reconstruction of the scene by two frames in a resolution of 500x500 (position 1, position 2)

Также выявилось, что от разрешения изображения зависит качество реконструкции. На Рисунке 14 показаны результаты эксперимента для изображения с большим разрешением. Здесь можно заметить большую детализацию «батискафа», а также более прослеживаемый контур «дельфина».

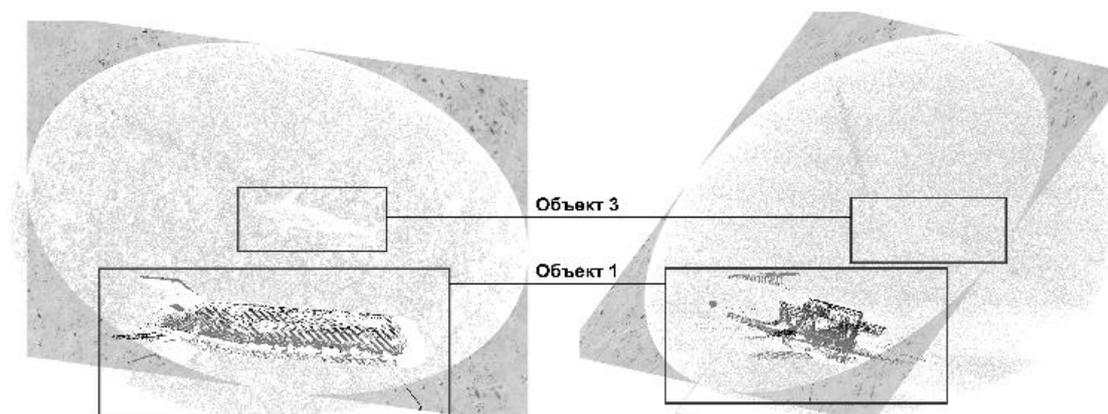


Рисунок 14 – Реконструкция сцены по двум кадрам в разрешении 1024x768 (позиция 1, позиция 2)

Figure 14 – Reconstruction of the scene by two frames in a resolution of 1024x768 (position 1, position 2)

Из анализа результатов реконструкции следует вывод о том, что для качества 3D-реконструкции важно четкое выделение границ объекта на кадрах видеопотока относительно фона.

В дальнейшем возможно увеличение детализации реконструкции путем применения данных о точках, полученных дальномером. Это позволит избежать появления ряда дыр на лицевой стороне моделей, а также даст возможность реконструировать наиболее мелкие объекты и детали.

Из результатов 3D-реконструкции также видно, что в случае дефекта на каком-либо из кадров происходит фрагментация получаемой 3D модели. Поскольку в данной реализации базиса программной системы анализ точек изображений выполнялся без использования данных с дальномера, возникала ситуация, когда невозможно определить толщину объекта. Именно поэтому на кадрах реконструкции можно увидеть «шлейф» модели.

Из экспериментов следует, что основными критериями, которые влияют на качество реконструкции сцен, являются: разрешение исходного видеопотока / снимка, количество кадров, количество используемых видов и число идентифицируемых точек сцены. Имеет значение и качество самих снимков / видеопотока (дефект хотя бы на одном снимке может приводить к неточностям при построении 3D модели объекта).

При проведении модельного эксперимента по оценке вычислительных затрат были воспроизведены экстремальные условия, при которых обрабатывался видеопоток с числом идентифицируемых точек более 120000. Заметим, что в реальных условиях число точек существенно меньше (порядка 50000). Полная обработка видеопотока продолжительностью 127 кадров (без критического числа точек) с распараллеливанием потоков требует до 12-19 минут согласно закону Амдала, который иллюстрирует ограничение роста производительности вычислительной системы с увеличением количества вычислителей. Основные затраты времени и ресурсов при этом идут на идентификацию и отслеживание точек, а сама 3D реконструкция занимает несколько секунд, так как на данном этапе происходит только визуализация точечных данных. Распараллеливание может уменьшить время обработки на порядок и более (зависит от числа процессоров). Из результатов экспериментов также следует, что при большом числе идентифицируемых точек требуется меньшее число кадров для реконструкции объектов.

## Заключение

В статье описываются результаты по реализации базиса программной системы 3D-реконструкции и визуализации объектов в динамических сценах с восстановлением невидимых участков поверхностей объектов.

Для повышения качества 3D реконструкции в ближайшей перспективе планируется расширение моделей «дальномерными» данными, которые позволят более точно (не программно) задавать формы и размеры объектов. Будет разработана версия комбинированного алгоритма по «зашивке дыр», основанного на ленточном варианте алгоритма Делоне, tight cocone и методе, использующем октодереву, индексацию треугольников и гибридное вычисление весовой функции с дальнейшим диффузионным размыванием, который представлен в статье. Это позволит повысить детализацию моделей одновременно с уменьшением количества «дыр» в реконструируемых поверхностях объектов.

## СПИСОК ИСТОЧНИКОВ

1. Hasler N., Rosenhahn B., Thormahlen T., Wand M., Gall J., Seidel H.P. Markerless motion capture with unsynchronized moving cameras. *CVPR*. 2009;224–231. DOI: 10.1109/CVPR.2009.5206859.
2. Ballan L., Brostow G.J., Puwein J., Pollefeys M. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics. Proceedings of SIGGRAPH*. 2010;29(4):134–146. DOI: 10.1145/1778765.1778824.
3. Taneja A., Ballan L., Pollefeys M. Modeling dynamic scenes recorded with freely moving cameras. *Conference on Computer Vision*. 2010;613–626. DOI: 10.1007/978-3-642-19318-7\_48.
4. Mustafa A., Kim H., Guillemaut J-Y., Hilton A. General Dynamic Scene Reconstruction from Multiple View Video. *ICCV*. 2015;900–908. DOI: 10.1109/ICCV.2015.109.
5. Keller M., Lefloch D., Lambers M., Izadi S., Weyrich T., and Kolb A. Real-time 3d reconstruction in dynamic scenes using point-based fusion. *Proc. of Joint 3DIM/3DPVT Conference (3DV)*. 2013;1–8. DOI: 10.1109/3DV.2013.9.
6. Mustafa A., Kim H., Guillemaut J-Y., Hilton A. Temporally coherent 4D reconstruction of complex dynamic scenes. *IEEE Conference on Computer Vision and Pattern recognition*. 2016;223–245. DOI: 10.1109/CVPR.2016.504.
7. Lefloch D., Kluge M., Sarbolandi H., Weyrich T., Kolb A. Comprehensive Use of Curvature For Robust And Accurate Online Surface Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. Доступно по: <http://reality.cs.ucl.ac.uk/projects/kinect/lefloch17comprehensive.pdf>. DOI: 10.1109/TPAMI.2017.2648803 (дата обращения: 14.03.2023).
8. Бобков В.А., Кудряшов А.П. Воксельный метод построения триангуляционной поверхности по множеству видов. *Информатика и системы управления*. 2012;2:31–38. Доступно по: [http://media/2012/N32\\_04.pdf](http://media/2012/N32_04.pdf) (дата обращения: 13.06.2021).
9. Бобков В.А., Кудряшов А.П. Построение трёхмерной модели морского дна воксельным методом. *Подводные исследования и робототехника*. 2016;2:13–18. Доступно по: <http://jmtf.febras.ru/journal/2-22-2016/13-18.pdf> (дата обращения: 13.06.2021).
10. Бобков В.А., Кудряшов А.П., Мельман С.В. О восстановлении движения динамических объектов по стереоизображениям. *Программирование*. 2018;3:29–42.

11. Кудряшов А.П., Черкашин А.С. Построение единой триангуляционной поверхности по набору видов с зашивкой дыр. *Информатика и системы управления*. 2015;1:36–40. Доступно по: [http://media/2015/N43\\_04.pdf](http://media/2015/N43_04.pdf) (дата обращения: 13.06.2021).
12. Bobkov V.A., Ron'shin Yu.I., Kudryashov A.P., and Mashentsev V.Yu. 3D SLAM from Stereoimages. *Programming and Computer Software*. 2014;40(4):159–165. DOI: 10.1134/S0361768814040021.
13. Bobkov V., Mashentsev V., Tolstonogov A., Scherbatyuk A. Adaptive Method for AUV Navigation Using Stereo Vision. *Proceedings of the 26th ISOPE International Ocean and Polar Engineering Conference*. 2016;562–565.
14. Bobkov V., Melman S., Kudrashov A., Scherbatyuk A. Vision-based navigation method for a local maneuvering of the autonomous underwater vehicle. *IEEE OES International Symposium on Underwater Technology 2017 (UT 2017 Busan)*. Busan; 2017;21–24.
15. Bobkov V.A., Melman S.V., and Kudryashov A.P. Fast Computation of Local Displacement by Stereo Pairs. *Pattern Recognition and Image Analysis*. 2017;3:458–465. DOI: 10.1134/S1054661817030063.
16. Гошин Е.В., Фурсов В.А., Бибииков С.А. Реконструкция 3D-сцен на пучках эпиполярных плоскостей стереоизображений. *Мехатроника, Автоматизация, Управление*. 2013;9:19–24.
17. Гошин Е.В. Реконструкция 3D-сцен по неректифицированным стереоизображениям с прореживающей фильтрацией. *Известия Самарского научного центра РАН*. 2013;15(6):748–753.
18. Скворцов А.В. *Триангуляция Делоне и её применение*. Томск: Изд-во Том. ун-та; 2002. 128 с.
19. Скворцов А.В., Костюк Ю.Л. *Эффективные алгоритмы построения триангуляции Делоне. Геоинформатика. Теория и практика*. Томск: Изд-во Том. ун-та; 1998;1:22–47.

## REFERENCES

1. Hasler N., Rosenhahn B., Thormahlen T., Wand M., Gall J., Seidel H.P. Markerless motion capture with unsynchronized moving cameras. *CVPR*. 2009;224–231. DOI: 10.1109/CVPR.2009.5206859.
2. Ballan L., Brostow G.J., Puwein J., Pollefeys M. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics. Proceedings of SIGGRAPH*. 2010;29(4):134–146. DOI: 10.1145/1778765.1778824.
3. Taneja A., Ballan L., Pollefeys M. Modeling dynamic scenes recorded with freely moving cameras. *Conference on Computer Vision*. 2010;613–626. DOI: 10.1007/978-3-642-19318-7\_48.
4. Mustafa A., Kim H., Guillemaut J-Y., Hilton A. General Dynamic Scene Reconstruction from Multiple View Video. *ICCV*. 2015;900–908. DOI: 10.1109/ICCV.2015.109.
5. Keller M., Lefloch D., Lambers M., Izadi S., Weyrich T., and Kolb A. Real-time 3d reconstruction in dynamic scenes using point-based fusion. *Proc. of Joint 3DIM/3DPVT Conference (3DV)*. 2013;1–8. DOI: 10.1109/3DV.2013.9.
6. Mustafa A., Kim H., Guillemaut J-Y., Hilton A. Temporally coherent 4D reconstruction of complex dynamic scenes. *IEEE Conference on Computer Vision and Pattern recognition*. 2016;223–245. DOI: 10.1109/CVPR.2016.504.
7. Lefloch D., Kluge M., Sarbolandi H., Weyrich T., Kolb A. Comprehensive Use of Curvature For Robust And Accurate Online Surface Reconstruction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. 2017. Available from:

- <http://reality.cs.ucl.ac.uk/projects/kinect/lefloch17comprehensive.pdf>.  
DOI: 10.1109/TPAMI.2017.2648803 (accessed on 14.03.2023).
8. Bobkov V.A., Kudryashov A.P. Voxel method of constructing a triangulation surface by a set of types. *Information Science and Control Systems*. 2012;2:31–38. Available from: [http://media/2012/N32\\_04.pdf](http://media/2012/N32_04.pdf) (accessed on 13.06.2021). (In Russ.).
  9. Bobkov V.A., Kudryashov A.P. Construction of a three-dimensional model of the seabed by the promissory note method. *Underwater Investigations and Robotics*. 2016;2:13–18. Available from: <http://jmtf.febras.ru/journal/2-22-2016/13-18.pdf> (accessed on 13.06.2021). (In Russ.).
  10. Bobkov V.A., Kudryashov A.P., Mel'man S.V. On restoring the motion of dynamic objects from stereo images. *Programming and Computer Software*. 2018;3:29–42. (In Russ.).
  11. Kudryashov A.P., Cherkashin A.S. Construction of a single triangulation surface based on a set of views with hole stitching. *Information Science and Control Systems*. 2015;1:36–40. Available from: [http://media/2015/N43\\_04.pdf](http://media/2015/N43_04.pdf) (accessed on 13.06.2021). (In Russ.).
  12. Bobkov V.A., Ron'shin Yu.I., Kudryashov A.P., and Mashentsev V.Yu. 3D SLAM from Stereoimages. *Programming and Computer Software*. 2014;40(4):159–165. DOI: 10.1134/S0361768814040021.
  13. Bobkov V., Mashentsev V., Tolstonogov A., Scherbatyuk A. Adaptive Method for AUV Navigation Using Stereo Vision. *Proceedings of the 26th ISOPE International Ocean and Polar Engineering Conference*. 2016;562–565.
  14. Bobkov V., Melman S., Kudrashov A., Scherbatyuk A. Vision-based navigation method for a local maneuvering of the autonomous underwater vehicle. *IEEE OES International Symposium on Underwater Technology 2017 (UT 2017 Busan)*. Busan; 2017;21–24.
  15. Bobkov V.A., Melman S.V., and Kudryashov A.P. Fast Computation of Local Displacement by Stereo Pairs. *Pattern Recognition and Image Analysis*. 2017;3:458–465. DOI: 10.1134/S1054661817030063.
  16. Goshin E.V., Fursov V.A., Bibikov S.A. Reconstruction of 3D scenes on bundles of epipolar planes of stereo images. *Mekhatronika, Avtomatizatsiya, Upravlenie*. 2013;9:19–24. (In Russ.).
  17. Goshin E.V. Reconstruction of 3D scenes from unrectified stereo images with thinning filtering. *Izvestia of Samara Scientific Center of the Russian Academy of Sciences*. 2013;15(6):748–753. (In Russ.).
  18. Skvortsov A.V. *Delaunay triangulation and its application*. Tomsk, TUP; 2002.128 p. (In Russ.).
  19. Skvortsov A.V., Kostyuk Yu.L. *Efficient algorithms for constructing Delaunay triangulation. Geoinformatics. Theory and practice*. Tomsk: TUP; 1998;1:22–47. (In Russ.).

#### ИНФОРМАЦИЯ ОБ АВТОРАХ / INFORMATION ABOUT THE AUTHORS

**Логачев Егор Михайлович**, ассистент, Департамент программной инженерии и искусственного интеллекта, Дальневосточный федеральный университет, Владивосток, Российская Федерация.

*e-mail*: [logachev.em1997@gmail.com](mailto:logachev.em1997@gmail.com)

ORCID: [0009-0007-0063-2816](https://orcid.org/0009-0007-0063-2816)

**Egor Mikhailovich Logachev**, Assistant Lecturer, Department of Software Engineering and Artificial Intelligence, Far Eastern Federal University, Vladivostok, the Russian Federation.

*Статья поступила в редакцию 19.03.2023; одобрена после рецензирования 14.04.2023;  
принята к публикации 10.05.2023.*

*The article was submitted 19.03.2023; approved after reviewing 14.04.2023;  
accepted for publication 10.05.2023.*