Security of Infocommunication Systems and Internet of Things

2024 Vol 2, No 2

https://doi.org/10.31861/sisiot2024.2.02007

Received 04 December 2024; revised 17 December 2024; accepted 22 December 2024; published 30 December 2024

Scanning of Three-Dimensional Objects by Photogrammetry Methods Using LiDAR and Mobile Computing

Bohdan Romaniuk[1] and Yuliya Tanasyuk[1,*]

[1]Computer Systems and Networks Department, Yuriy Fedkovych Chernivtsi National University, Chernivtsi, Ukraine

*Corresponding author (E-mail: y.tanasyuk@chnu.edu.ua)

ABSTRACT Three-dimensional models of objects are widely used in various fields, including science, construction, healthcare, and entertainment, making the task of creating such models highly relevant. The primary goal of this paper is to develop a mobile application which uses photogrammetry methods for capturing real-world three-dimensional objects or environments. The main advantage of photogrammetry is low hardware requirements with relatively high accuracy of the models obtained. Additionally, the application utilizes smartphone's LiDAR sensor to enhance capture quality, especially for low-textured objects. The LiDAR sensor allows for precise distance measurement between the device and the object, which is crucial for accurately capturing the object's size and position. To build 3D model of the object from a series of images, the application uses Object Capture API, available on iOS, iPadOS, and macOS operating systems. This API fully leverages the build-in GPU and Neural Processing Unit to build and tessellate the point cloud and generate the polygonal mesh model. The application was developed for iPhones that support LiDAR sensor, using Swift programming language, SwiftUI for the user interface, and RealityKit for Object Capture API. The app features three modes: object capturing, model reconstruction and model preview. To simplify the process of capturing the object, the app can automatically take photos of the object, provide the user with guidance and recommendations on optimal lighting conditions, camera positioning, and a preview of the point cloud. Once object capture is complete, the application transitions to the reconstruction mode, which uses captured photos and point cloud data. This process involves image alignment, detailed point cloud generation, polygonal mesh model generation, texture and normal map generation, and model optimization. After reconstruction, the user can obtain model in the USDZ file format and preview it using built-in system tools. Following these steps, two test models of a drum and a garden statue were built with satisfactory quality and accuracy, while maintaining the original size of the scanned objects. The resulting three-dimensional polygonal models can be successfully exported to various 3D editors and programs.

KEYWORDS photogrammetry, object capture, mobile device, swift, 3D models.

I. INTRODUCTION

The rapid development of computer graphics and computing power of GPUs (Graphics Processing Unit) enables the creation of photorealistic virtual environments, which can be used for scientific, constructional, cultural, educational, and entertainment purposes. For example, architects and builders use 3D models of buildings and structures, while CGI (Computer-Generated Imagery) artists create photorealistic 3D models of humans for movies and games.

There are two main methods to obtain 3D model from a physical object. The first one, called photogrammetry (or stereophotogrammetry), uses set of photographs of an object to determine the shape, size and position of the object's points in space by measuring and analyzing these images [1-2]. The primary advantage of this method is its simplicity and low hardware requirements. Because of this, photogrammetry can use images from any camera, such as smartphone, DSLR (Digital Single-Lens Reflex), or a drone. However, in certain scenarios, like when dealing with transparent or low-textured objects, the quality of the photogrammetry output can deteriorate significantly.

The second method involves LiDAR (Light Detection and Ranging) scanners, which determine distances by targeting an object with a laser and measuring the time it takes for the reflected light to return to the receiver. LiDAR provides high accuracy and copes well with complex objects [3]. However, industrial-grade LiDAR scanners often come at a high cost.

In addition to enhancements in computer graphics and GPU computing power, mobile devices have also experienced substantial development. Modern smartphones are equipped with camera systems that can capture high-quality images suitable for photogrammetry. Moreover, some of the latest iPhones and iPads come with LiDAR scanner and possess enough computing power to improve the quality of the photogrammetry results and perform on-device reconstructions.

There are many software solutions on the market for creating three-dimensional models using photogrammetry methods from different vendors. Among them are Polycam and 3DF Zephyr by 3Dflow.

Polycam [4] is a mobile application that allows users to capture photos to create different types of 3D assets, namely three-dimensional models using photogrammetry, gaussian splatting, rooms scanner, and 360 image creator. The application supports automatic image capturing using smartphone, and once completed, the images are sent to the

cloud to create three-dimensional mesh model or gaussian splat, at a user's choice. When the processing is complete, a user can view the model in the application or export it in GLB file format. The other formats require a subscription. Also, a user can create up to 5 photogrammetric or gaussian splat captures using up to 100 images per capture for free, while subscription ensures unlimited number of captures with up to 2000 images for photogrammetric captures and up to 1000 images for gaussian splat captures.

Another great example of photogrammetry software is 3DF Zephyr by 3Dflow [5]. 3DF Zephyr is a commercial photogrammetry and 3D modelling software. It is a desktop Windows application, but user can import photos from any camera, e.g. smartphone's camera, DSLR, or drone's camera. The application also can utilize video cards to speed up reconstruction process, but only Nvidia GPUs that use CUDA are supported. Otherwise, the application will use only CPU cores. Like Polycam, 3DF Zephyr comes both in free and paid versions. The free one is limited to only 50 images per capture, supports a single Nvidia GPU, and imposes editing and exporting restrictions. The full version of 3DF Zephyr, available as a monthly subscription or one-time payment, allows one to capture an unlimited number of images, restricted only by the available system RAM, and utilize multiple Nvidia GPUs.

Considering the software features outlined above, the primary goal of this paper is the development of a mobile iOS application designed for scanning three-dimensional objects using photogrammetry methods and a smartphone, equipped with a camera and LiDAR sensor. This combination allows for high-performance object capture and 3D model reconstruction entirely on a mobile device. The relevance of the research is emphasized by the fact that resulted 3D models can be used across various fields, such as virtual reality, gaming, product design, cultural heritage preservation, and e-commerce.

Since the application uses smartphone as the main capturing and processing device, it allows users to take pictures, view, and export three-dimensional models for free without an Internet connection and cloud services. Considering these advantages, the developed application can be compared with alternative software solutions using Table 1.

II. PRINCIPLES OF SCANNING THREE-DIMENSIONAL OBJECTS USING PHOTOGRAMMETRY METHODS

Photogrammetry is a technique that allows one to reconstruct three-dimensional objects from two-dimensional images by analyzing and interpreting spatial relationships between these images. This technique relies on image overlap, which is essential for achieving accurate and reliable 3D reconstruction by identifying key points. Thus, the photogrammetry methods require a large set of images of the same object, captured from different angles (Fig. 1). After capturing the images, software identifies key points (for example, object boundaries or texture highlights) using obtained images (e.g., I_1, I_2, I_3). By means of these identified key points from each image (a_1, b_1, and c_1; a_2, b_2, and c_2; a_3, b_3, and c_3), the system can combine them into real 3D points. Since each key point of $a_1b_1c_1$ correlates with $a_2b_2c_2$ and $a_3b_3c_3$, the system can match them accordingly. Therefore, by knowing the position and orientation of the cameras (P_1, P_2, and P_3) at the time of capturing, the software can determine the geometric coordinates of the ABC points of the object as intersections of the corresponding lines [2].

However, one should take into account that using only photographic images from the camera can lead to inaccuracies in identifying key points, especially when scanning low-textured objects. In such cases, it can be challenging to identify key points due to the lack of distinct surface features. To address this issue, application uses the device's LiDAR scanner to capture additional metadata, such as depth information, which can enhance the accuracy of the 3D reconstruction during the processing stage [3].

The application workflow is divided into several stages, each executed sequentially as follows:

Stage 0. The user points the smartphone at the object that to be scanned.

Stage 1. The user presses the capture button, prompting a bounding box to appear around the object (Fig. 2).

The user can manually adjust the size of the bounding box or reset it. If necessary, the user can also cancel the operation, returning to Stage 0.

Stage 2. The user presses the capture button again to begin the scanning process. To complete this stage, the user needs to move around the object, keeping it in the frame.

TABLE 1. Photogrammetry applications comparison.

Application	Developed app	Polycam	3DF Zephyr
Platform	Mobile	Mobile + Cloud	Desktop
Operating System	iOS	iOS and Android	Windows
Hardware requirements	iPhone or iPad with LiDAR sensor	iOS or Android smartphone with camera and at least 3.5 GB RAM	PC with at least dual core CPU, 16 GB RAM, and 10 GB storage. For GPU – Nvidia video card with at least 1GB of VRAM and DirectX 9.0c support
Maximum number of images	200	100 for free tier 2000 for Pro and higher tiers	50 for free version Unlimited for full version
Export formats	USDZ	GLB for free version USDZ, STL, OBJ, DAE, FBX, and others for Pro and higher tiers	OBJ, STL, and PLY
Licensing	Free	Free with limitations Monthly or annual subscriptions	Free with limitations Monthly subscription One-time purchase

FIG. 1. Obtaining the key points of the object from images I_1, I_2, and I_3 taken by cameras P_1, P_2, and P_3.

FIG. 2. Example of bounding box that appears for captured object.

Although the system will automatically take photos, the user can also manually capture images if needed. Additionally, the user can monitor the number of photos taken.

Step 3. After the initial scan, the user is prompted to perform up to two additional scans, which may involve repositioning the object or capturing it from different heights. If the user chooses to continue, the app returns to Stage 2 for each additional scan. Otherwise, the app transitions to Stage 4.

Stage 4. Once the object capture is complete, the reconstruction process begins. This step utilizes the device's built-in GPU and NPU (Neural Processing Unit) to accelerate the reconstruction through parallel computing. The system aligns the images and generates a point cloud of the model, using the collected data (i.e. photos, depth maps, LiDAR data, and other metadata). The point cloud is then tessellated, with textures and normal maps generated, followed by model optimization to ensure smooth performance on mobile devices.

Stage 5. The final stage involves saving the created 3D model. The model is stored in the USDZ file format, allowing the user to preview it in 3D or augmented reality (AR) modes using system tools. The file can also be imported into various 3D applications and editors.

An UML activity diagram (Fig. 3) effectively represents the workflow of the object capture process, clearly displaying user actions and system responses.

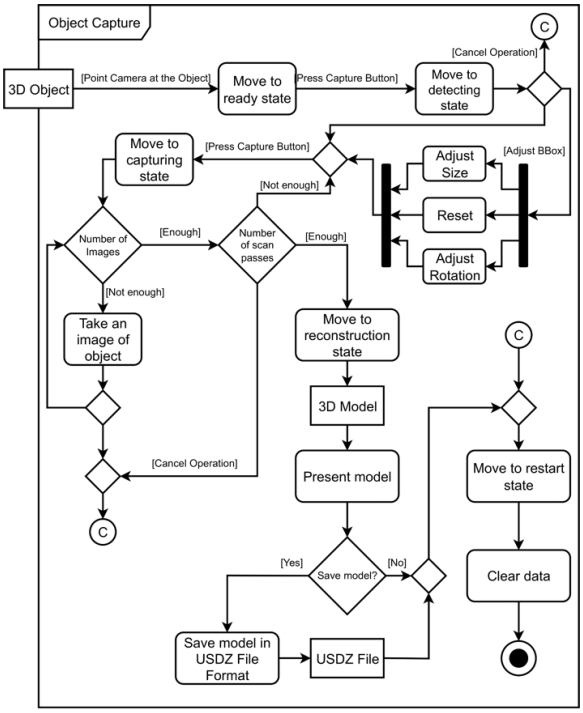


FIG. 3. UML activity diagram for object capture process.

III. SOFTWARE IMPLEMENTATION OF THE IOS APPLICATION TO SCAN AND RECONTRUCT THREE-DIMENSIONAL OBJECTS

The iOS application was developed using the Swift programming language, which was created and introduced by Apple in 2014. Swift is a high-level general-purpose programming language that compiles into machine code using LLVM-based compiler [6-7]. LLVM is a set of compiler and toolchains that can be used to develop both frontend for any language and backend for any ISA (instruction set architecture) [8].

The user interface was designed and implemented with the SwiftUI framework, which is used to develop applications for iOS, iPadOS, macOS and other Apple operating systems. SwiftUI allows developers to build user interfaces in declarative way, using pure Swift code without requiring markup languages [9]. For example, Fig. 4 shows a simple SwiftUI View containing text and a button. Due to SwiftUI's reactive nature, the Text View automatically updates when the user presses the button and the count value changes.

```
import SwiftUI

struct ContentView: View {
    @State private var count: Int = 0

    var body: some View {
        VStack {
            Text("Hello, SwiftUI!")

            Button {
                count += 1
            } label: {
                Text("Counter: \(count)")
            }
            .buttonStyle(.borderedProminent)
        }
        .padding()
    }
}
```

FIG. 4. Basic example of SwiftUI View that contains Text and Button Views.

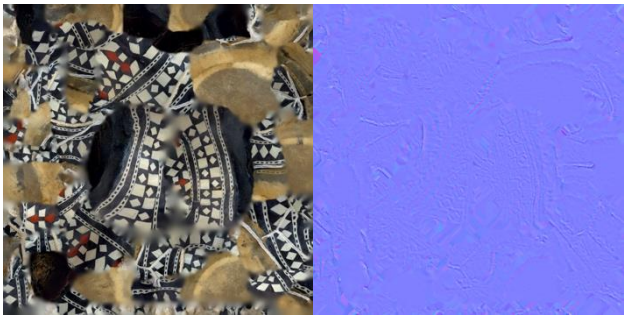
For object capturing and model reconstruction, the application uses Object Capture API, a component of the RealityKit framework. RealityKit provides high-performance 3D simulation and rendering capabilities by utilizing the device's built-in GPU [10]. When using the Object Capture API, the framework can leverage the Apple Neural Engine (Apple's proprietary neural processing unit) to significantly accelerate compute vision algorithms, which are essential for recognizing key points of the object in the captured images [11-12].

IV. REVIEW OF THE RESULTS OF THE DEVELOPED MOBILE APPLICATION

Using the developed mobile application, several three-dimensional models of real-world objects were generated through photogrammetry and on-device reconstruction. All capturing and rendering processes were conducted on a device powered by Apple A17 Pro chipset, featuring 6-core CPU (4 efficiency cores and 2 performance cores), 6-core GPU, 16-core NPU, and 8 GB of LPDDR5 RAM. The device's main camera was equipped with a 48 MP sensor, 1.22 µm pixel size, and an f/1.78 aperture. Additionally, the device was equipped with a LiDAR sensor, enhancing depth perception, improving object reconstruction accuracy, and facilitating low-textured objects capturing. Both objects under investigation were captured outdoors under diffuse lighting in cloudy weather, ensuring uniform illumination. Blender was used to render the resulting models, as well as to extract textures and normal maps [13].

The first three-dimensional model of a drum (Fig. 5) was generated from 109 images taken at various angles around the object. The capture process was divided into three stages, each of which took approximately 1-2 minutes, with the drum repositioned between stages to ensure full-angle coverage. The reconstruction process then took around 5-6 minutes. The resulting mesh was saved in the USDZ file format, consisting of 14 258 vertices and 25 000 faces, with a final size of 10.4 MB. Additionally, a 2K (2048 × 2048px) texture and a normal map (Fig. 6) were generated and embedded in the final file.

FIG. 5. The 3D model of the drum generated with the developed mobile application.



a) b)

FIG. 6. Generated 2K texture (a) and normal map (b) for the drum model (Fig. 5).

To evaluate the performance of the developed application, we compared it with Polycam and 3DF Zephyr using images obtained with the presented software during capturing process. However, since drum was flipped several times, both Polycam and 3DF Zephyr accepted only limited sets of taken images from the first scan. Furthermore, 3DF Zephyr only created a texture, while Polycam and the developed application created both a texture and a normal map. The detailed comparison between the developed app and its alternatives is presented in Table 2.

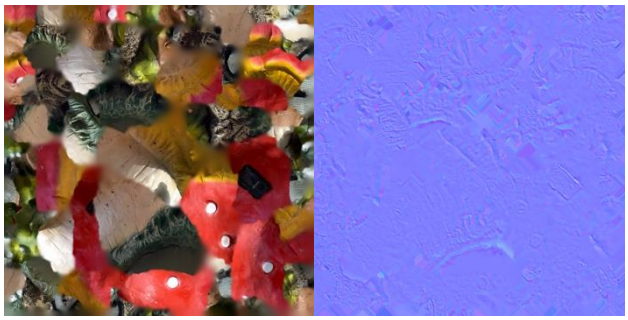
TABLE 2. Drum model reconstruction comparison.

Application	Developed app	Polycam	3DF Zephyr
Number of images	109	40	40
Number of vertices	14 258	18 442	5 266
Number of faces	25 000	32 222	10 324
Texture	2048px	4096px	6704px
Normal map	2048px	4096px	N / A
File format	USDZ	GLB	OBJ/MTL
File size	10.4 MB	4.1 MB	4.4 MB

The second three-dimensional model created was a garden figure (Fig. 7), generated from 91 images taken at various angles around the object. Similar to the drum model reconstruction, the capture process was divided into three stages, each lasting approximately 1-2 minutes. However, unlike the drum, the garden figure was difficult to reposition. To achieve full-angle coverage, the additional pictures were taken with the device positioned at varying polar angles – one at a higher angle and another one at a lower angle. The reconstruction process also took approximately 5-6 minutes. The resulting mesh, saved in the USDZ file format, contains 14 417 vertices and 25 000 faces, and its final size is 10.6 MB. Similar to the drum model, a 2K (2048 × 2048px) texture and a normal map (Fig. 8) were generated and embedded in the final file.

The evaluation of the garden statue model reconstructed from the images captured with the developed application has been performed. In this case study, Polycam accepted all 91 images, since the real-world object wasn't moved, and all images were taken by changing the position relative to the object. However, 3DF Zephyr took only 50 images due to free version limitations.

FIG. 7. Generated 3D model of the garden figure.



a) b)

FIG. 8. Generated 2K texture (a) and (b) for the garden figure model (Fig. 7).

The detailed comparison between the developed app and alternative applications is given in Table 3.

TABLE 3. Garden statue model reconstruction comparison.

Application	Developed app	Polycam	3DF Zephyr
Number of images	91	91	50
Number of vertices	14 417	30 244	12 756
Number of faces	25 000	49 999	25 179
Texture	2048px	4096px	6704px
Normal map	2048px	4096px	N / A
File format	USDZ	GLB	OBJ/MTL
File size	10.6 MB	6.4 MB	7.7 MB

On balance, two 3D models presented proved to be of good quality and compatible with the considered commercial software products. However, it is worth noting, that the models produced by both Polycam and 3DF Zephyr are not watertight and include the elements of their surrounding environment as a part of the model geometry. This may require additional processing depending on the use case. In addition, through the use of LiDAR, the results obtained with the developed application retain accurate dimensions, without the need to use reference values. For example, the actual dimensions of the drum are 204 mm in height, 139 mm in top diameter, and 92 mm in bottom diameter. According to Blender, these dimensions mostly coincide with those of the output 3D model, namely: 202 mm in height, 138 mm in top diameter, and 92 mm in bottom diameter (Fig. 9).

FIG. 9. Dimensions of the generated model of the drum, defined in Blender.

The same applies to the garden statue, which has a height of 390 mm and a diameter width of 300 mm. According to Blender, the height of the model obtained is 383 mm, and the diameter width is 292 mm (Fig. 10).

FIG. 10. Dimensions of the generated model of the garden statue, defined in Blender

V. CONCLUSION

The iOS mobile application has been developed to scan and reconstruct 3D models of objects using photogrammetry methods, built-in device's camera, LiDAR sensors, NPU, and GPU.

The iOS mobile application was developed using the Swift programming language, the SwiftUI framework for UI and the RealityKit framework, which includes the Object Capture API. This API generates 3D object models using photogrammetry methods and enables real-time capturing with mobile devices equipped with a LiDAR sensor to enhance model quality, even in complex cases, such as low-textured models or poor lightning conditions. Available in recent versions of iOS, iPadOS, and macOS, the API leverages the device's GPU and NPU to significantly improve efficiency and reduce processing time.

The 3D models created with the developed mobile application possess satisfactory quality and accuracy, while maintaining the original dimensions of the scanned objects. Each model is proved to be manifold, ensuring continuous, gap-free surface ideal for simulations, 3D printing, and further processing. Moreover, the resulting 3D models can be successfully exported to the widely used visualization environments and, thanks to the presence of textures and normal maps, are well-suited for high-quality renders.

AUTHOR CONTRIBUTIONS

B.R. – conceptualization, software development, original draft preparation, visualization; Y.T. – draft review and editing, supervision.

COMPETING INTERESTS

The authors declare no conflict of interest.

REFERENCES

[1] Nvidia, "What Is Photogrammetry?" [Online]. Available: https://blogs.nvidia.com/blog/what-is-photogrammetry.

[2] T. Luhmann, S. Robson, S. Kyle, and J. Boehm, *Close-Range Photogrammetry and 3D Imaging*, 4th ed. Berlin, Germany: De Gruyter, 2023.

[3] "The Basics of LiDAR – Light Detection and Ranging – Remote Sensing," [Online]. Available: https://www.neonscience.org/resources/learning-hub/tutorials/lidar-basics.

[4] Polycam, "3D Scanner, LiDAR, 360," [Online]. Available: https://poly.cam.

[5] 3Dflow, "3DF Zephyr – the photogrammetry software solution," [Online]. Available: https://www.3dflow.net/3df-zephyr-photogrammetry-software/.

[6] The Swift Project, "Swift Language Documentation," [Online]. Available: https://www.swift.org/documentation.

[7] Apple Inc., *The Swift Programming Language*, Swift 5.7 ed. Apple Inc., 2022.

[8] The LLVM Project, "Overview and Documentation," [Online]. Available: https://www.llvm.org.

[9] Apple Inc., "SwiftUI Framework Documentation," [Online]. Available: https://developer.apple.com/documentation/SwiftUI.

[10] Apple Inc., "RealityKit Framework Documentation," [Online]. Available: https://developer.apple.com/documentation/realitykit.

[11] Apple Inc., "Object Capture API Documentation," [Online]. Available: https://developer.apple.com/documentation/realitykit/realitykit-object-capture.

[12] E. Hollemans, "The Neural Engine – what do we know about it?" [Online]. Available: https://github.com/hollance/neural-engine.

[13] Blender Foundation, "Blender – a 3D modelling and rendering software," [Online]. Available: https://www.blender.org.

Bohdan Romaniuk

In 2023, graduated from Yuriy Fedkovych Chernivtsi National University with a degree in "Computer Engineering" (bachelor's level). Currently studying at the Chernivtsi National University, majoring in "Computer Engineering of Internet of Things and Cyber-Physical Systems Technologies" (master's level). Research interests are as follows: software engineering, embedded programming, computer graphics.

Yuliya Tanasyuk

PhD, Associate Professor at the Department of Computer Systems and Networks of Physical, Technical and Computer Sciences Institute, Yuriy Fedkovych Chernivtsi National University, Ukraine. Research interests and academic activities are as follows: programming, network information technologies, cybersecurity, IoT, software engineering.

ORCID ID: 0000-0001-8650-0521

Сканування тривимірних об'єктів методами фотограмметрії з використанням LiDAR та мобільних обчислень

Богдан Романюк[1], Юлія Танасюк[1,*]

[1]Кафедра комп'ютерних систем та мереж, Чернівецький національний університет імені Юрія Федьковича, Чернівці, Україна

*Автор-кореспондент (Електронна адреса: y.tanasyuk@chnu.edu.ua)

АНОТАЦІЯ Тривимірні моделі об'єктів широко використовуються в різних сферах, у тому числі в науці, будівництві, медицині та індустрії розваг, що робить завдання створення таких моделей дуже актуальним. Основною метою цієї

роботи є розробка мобільного застосунку, який використовує методи фотограмметрії для сканування об’єктів. Основною перевагою фотограмметрії є низькі вимоги до апаратного забезпечення, але водночас вона забезпечує відносно високу точність отриманих моделей. Крім того, програма використовує сенсор LiDAR смартфона для покращення якості зйомки, особливо для низькотекстурованих об’єктів. Датчик LiDAR дозволяє точно вимірювати відстань між пристроєм і об’єктом, що важливо для отримання інформації про розмір та положення об’єкта. Для побудови 3D-моделі об’єкта із серії зображень програма використовує Object Capture API, доступний в операційних системах iOS, iPadOS та macOS. Цей API повністю використовує вбудовані GPU і блок нейронної обробки (NPU) для побудови та теселяції хмари точок і створення полігональної сітки моделі. Програма була розроблена для iPhone, які оснащені датчиком LiDAR, використовуючи мову програмування Swift, SwifUI для інтерфейсу користувача та RealityKit для Object Capture API. Створений застосунок підтримує три режими роботи: сканування об’єкта, реконструкція моделі та попередній перегляд моделі. Щоб спростити процес зйомки об’єкта, програма може автоматично робити фотографії об’єкта та надавати вказівки і рекомендації користувачеві. Ці рекомендації містять поради щодо оптимальних умов освітлення, розташування камери, а також проміжний вигляд хмари точок. Після завершення зйомки об’єкта програма переходить в режим реконструкції, який використовує отримані, під час зйомки, фотографії та дані хмари точок. Цей процес включає вирівнювання зображення, генерацію детальної хмари точок, генерацію полігональної сітки моделі, генерацію текстури і карти нормалей, а також оптимізацію моделі. Після реконструкції, користувач може отримати модель в форматі файлу USDZ та переглянути її за допомогою системних засобів. Після цих кроків було створено дві тестові моделі барабану та садової фігури із задовільною якістю та точністю, зберігаючи початковий розмір сканованих об’єктів. Отримані тривимірні полігональні моделі можна експортувати в різні 3D-редактори та спеціалізовані програми.

КЛЮЧОВІ СЛОВА фотограмметрія, захоплення об’єкта, мобільний пристрій, swift, 3D модель.