

Министерство науки и высшего образования Российской Федерации
Томский государственный университет систем управления и
радиоэлектроники

Захлебин А.С.

СИСТЕМЫ СОЗДАНИЯ И ОТОБРАЖЕНИЯ 3D-ИНФОРМАЦИИ

Учебно-методическое пособие по практическим занятиям, лабораторным
работам и самостоятельной работе

Томск 2025

УДК 004.9, 528.8
ББК 26.113
3-38

Рецензент:

Курячий М. И., профессор кафедры телевидения и управления ТУСУР, канд. техн. наук.

Захлебин Александр Сергеевич

3-38 Системы создания и отображения 3D-информации: Учебно-методическое пособие по практическим занятиям, лабораторным работам и самостоятельной работе / Захлебин А.С. – Томск: гос. ун-т систем упр. и радиоэлектроники, 2025. – 18 с.

Учебно-методическое пособие предназначено для работы на практических занятиях, лабораторных работах и для самостоятельной работы студентов по дисциплине «Системы создания и отображения 3D-информации».

Целью изучения дисциплины «Системы создания и отображения 3D-информации» является подготовка бакалавров в теории и технике современных средств отображения 3D-информации на основе информационных моделей различных видов.

Одобрено на заседании каф. ТУ протокол № 3 от 13.02.2025

УДК 004.9, 528.8
ББК 26.113

© Захлебин А.С., 2025
© Томск: гос. ун-т систем упр.
и радиоэлектроники, 2025

Оглавление

Практическая работа №1. «Получение и визуализация стереопары».	4
Практическая работа №2. «Калибровка камеры. Вычисление внутренних параметров».	5
Практическая работа №3. «Построение 3D-модели объекта по серии его изображений».	7
Практическая работа №4. «Импорт и просмотр 3D-модели в VR-сцене».	8
Практическая работа №5. «Построение 3D-модели объекта из видеозаписи».	9
Лабораторная работа №1. «Определение расстояния до объекта по стереопаре».	10
Лабораторная работа №2. «Построение цифровой 3D-модели объекта по его изображениям с использованием контрольных точек».	12
Лабораторная работа №3. «Построение цифровой 3D-модели объекта по его изображениям без использования контрольных точек».	14
Лабораторная работа №4. «Расчёт объема насыпи по цифровой 3D-модели».	16
Темы для самостоятельного изучения.	18

Практическая работа №1. «Получение и визуализация стереопары».

Тема «Получение и визуализация стереопары».

Цель работы.

Освоить принципы получения стереопары с использованием камеры смартфона и реализовать её визуализацию в гарнитуре виртуальной реальности с применением Unity.

Задачи.

1. Получить стереопару с помощью камеры смартфона.
2. Подготовить изображения для загрузки в Unity.
3. Создать в Unity сцену, отображающую левый и правый каналы изображения отдельно на каждый глаз.
4. Просмотреть стереопару в стереогарнитуре.

Порядок выполнения работы.

1. Найдите неподвижный объект (например, статую, вазу, предмет на столе).
2. Сделайте два снимка с горизонтальным смещением камеры примерно на 6–7 см между кадрами (вправо/влево).
3. Убедитесь, что объект съёмки остаётся неподвижным, а камера остаётся на одном уровне (лучше использовать штатив или опору).
4. Назовите изображения left.jpg и right.jpg.
5. Создайте новый 3D-проект в Unity.
6. Установите в проект: XR Plugin Management (в Project Settings → XR Plugin Management → включите OpenXR).
7. Поместите left.jpg и right.jpg в папку Assets/Textures/
8. В Unity создайте два Quad объекта (GameObject → 3D Object → Quad), расположите их на расстоянии друг от друга (например, -0.03 и +0.03 по оси X от центра)
9. Назначьте на каждый Quad соответствующую текстуру: левый глаз — left.jpg, правый глаз — right.jpg.
10. Для каждого Quad установите Layer: Создайте слои LeftEyeLayer и RightEyeLayer. Присвойте левому Quad слой LeftEyeLayer, правому — RightEyeLayer.
11. В XR Origin → Camera настройте: Stereo Target Eye: Both. В скрипте (или с помощью Canvas per eye) скройте отображение левого Quada для правого глаза и наоборот.
12. Подключите Oculus Quest 3 к ПК через Oculus Link. Запустите сцену — убедитесь, что левому и правому глазу соответствуют разные изображения. Настройте положение камер/Quad'ов для комфортного восприятия объёма. При необходимости подстройте расстояние между изображениями.

Контрольные вопросы

1. Что такое стереопара?
2. Почему необходимо горизонтальное смещение камеры при съёмке?
3. Какие ошибки могут привести к плохому стереоэффекту?
4. Как обеспечить отдельную подачу изображений на левый и правый глаз в Unity?

Практическая работа №2. «Калибровка камеры. Вычисление внутренних параметров».

Тема «Калибровка камеры. Вычисление внутренних параметров».

Цель работы.

Научиться проводить калибровку камеры смартфона с использованием стандартных шаблонов и инструментов компьютерного зрения, получить матрицу внутренних параметров камеры и коэффициенты дисторсии.

Задачи.

1. Подготовить калибровочный шаблон (шахматную сетку).
2. Получить серию изображений с камеры смартфона под разными ракурсами.
3. Выполнить калибровку камеры с помощью инструментария OpenCV.
4. Получить параметры калибровки: фокусное расстояние, главный точку (principal point), коэффициенты искажения.
5. Интерпретировать полученные результаты.

Порядок выполнения работы.

1. Распечатайте шаблон шахматной доски.
2. Прикрепите шаблон на жёсткую, плоскую поверхность (на доску или стену).
3. Сделайте 10–20 снимков шаблона с разных ракурсов (наклоны, повороты, разное расстояние), но чтобы вся доска была видна.
4. Назовите изображения последовательно: img1.jpg, img2.jpg, ...
5. Скопируйте их на компьютер в отдельную папку.
6. В Python установите библиотеки: opencv, numpy.
7. Используйте следующий шаблон кода:

```
import cv2

import numpy as np

import glob

# размеры шахматной доски
chessboard_size = (9, 6)

square_size = 0.025 # в метрах (или других единицах)

objp = np.zeros((chessboard_size[0]*chessboard_size[1], 3), np.float32)

objp[:, :2] = np.mgrid[0:chessboard_size[0], 0:chessboard_size[1]].T.reshape(-1, 2)

objp *= square_size

objpoints = []

imgpoints = []

images = glob.glob('path_to_images/img*.jpg')

for fname in images:
```

```

img = cv2.imread(fname)
gray = cv2.cvtColor(img, cv2.COLOR_BGR2GRAY)
ret, corners = cv2.findChessboardCorners(gray, chessboard_size, None)
if ret:
    objpoints.append(objp)
    imgpoints.append(corners)
ret, mtx, dist, rvecs, tvecs = cv2.calibrateCamera(objpoints, imgpoints, gray.shape[:-1],
None, None)
print("Камера откалибрована:")
print("Матрица камеры:\n", mtx)
print("Коэффициенты дисторсии:\n", dist)
8. Сохраните параметры в файл camera_calib.txt

```

Контрольные вопросы.

1. Зачем нужна калибровка камеры?
2. Что такое параметры внутреннего ориентирования камеры?
3. Какие искажения корректируются при калибровке?
4. Почему важно делать снимки под разными углами?
5. Как параметры калибровки используются в 3D-реконструкции?

Практическая работа №3. «Построение 3D-модели объекта по серии его изображений».

Тема «Построение 3D-модели объекта по серии его изображений».

Цель работы.

Освоить базовые принципы фотограмметрии и научиться строить 3D-модель объекта по множеству фотографий, снятых под разными ракурсами, с помощью программного обеспечения Metashape.

Задачи.

1. Провести съемку объекта.
2. Импортировать изображения в Metashape.
3. Выполнить выравнивание камер.
4. Построить плотное облако точек
5. Построить сетку (mesh) и текстурировать модель
6. Проанализировать качество результата и экспортировать 3D-модель

Порядок выполнения работы.

1. Разместите объект на однотонном фоне (по возможности — без бликов и теней).
2. Снимите не менее 30–40 кадров с разных углов, обходя объект по кругу и сверху.
3. Фокусировка и экспозиция должны быть стабильными — отключите автоэкспозицию/автофокус.
4. Скопируйте фотографии на ПК в одну папку.
5. Откройте Metashape.
6. Создайте новый проект File → New.
7. Импортируйте изображения: Workflow → Add Photos → выберите все фото
8. Сохраните проект: File → Save As.
9. Выравнивание камер. Workflow → Align Photos. Accuracy: High. Generic Preselection: Enabled. Key point limit: 40,000. Tie point limit: 4,000. Подождите завершения, проверьте распределение камер и точек. Вы должны видеть облако точек и камеры.
10. Фильтрация лишних точек. Используйте инструмент Rectangle Selection для удаления точек фона.
11. Построение плотного облака точек. Workflow → Build Dense Cloud. Quality: Medium или High (в зависимости от производительности ПК). Depth filtering: Moderate.
12. Построение сетки. Workflow → Build Mesh. Surface type: Arbitrary. Source data: Dense cloud. Face count: Medium. Interpolation: Enabled.
13. Построение текстуры. Workflow → Build Texture. Mapping mode: Generic. Blending mode: Mosaic.
14. Экспорт 3D-модели. File → Export → Export Model. Форматы: .obj, .fbx, .ply и т.д.

Контрольные вопросы.

1. Какой принцип лежит в основе фотограмметрии?
2. Зачем нужно перекрытие между изображениями?
3. Чем отличаются облако точек и сетка (mesh)?
4. В чем заключается процесс выравнивания изображений?
5. Какие ошибки могут привести к плохой 3D-реконструкции?

Практическая работа №4. «Импорт и просмотр 3D-модели в VR-сцене».

Тема «Импорт и просмотр 3D-модели в VR-сцене».

Цель работы.

Научиться импортировать 3D-модель, полученную из Metashape, в Unity, настроить сцену для просмотра в виртуальной реальности и запустить её в стереогарнитуре.

Порядок выполнения работы.

1. Откройте Unity Hub, создайте новый проект 3D.
2. В меню File → Build Settings → Android. Выберите платформу Android и нажмите Switch Platform.
3. Перейдите в Edit → Project Settings → XR Plug-in Management: Установите XR Plug-in Management.
4. Активируйте Oculus в разделе Android.
5. Настройка XR и сцены. В Project Settings → Player → Android → Other Settings: Убедитесь, что Graphics API → Vulkan отключён (используйте OpenGL ES3). Установите Minimum API Level на Android 10 (API Level 29) или выше. Установите необходимые пакеты через Package Manager: XR Interaction Toolkit Input System.
6. Импорт 3D-модели. Скопируйте экспортированную модель (например, .fbx) в папку Assets/Models. Перетащите модель в сцену. Настройте масштаб, положение и освещение: добавьте Directional Light. Убедитесь, что модель не «проваливается» под пол.
7. Добавление VR-камеры. Удалите стандартную Main Camera. Добавьте XR Origin (VR) из XR → Device Based → XR Origin (VR). Убедитесь, что XR Origin находится над поверхностью сцены. Добавьте плоскость (Plane) для «пола» под камерой.
8. Построение проекта и запуск. Подключите Oculus Quest по кабелю или настройте Air Link. В Build Settings добавьте текущую сцену. Нажмите Build And Run.
9. Наденьте гарнитуру — сцена должна загрузиться. Как пользователь вы сможете оглядываться и приближаться к 3D-модели.

Контрольные вопросы.

1. В каких форматах можно импортировать 3D-модели в Unity?
2. Что такое XR Origin и зачем он нужен в сцене?
3. Каковы основные требования к сцене для корректной работы в VR?
4. Как обеспечить правильный масштаб импортированной модели?
5. Какие существуют способы передачи приложения на стереогарнитуру?

Практическая работа №5. «Построение 3D-модели объекта из видеозаписи».

Тема «Построение 3D-модели объекта из видеозаписи».

Цель работы.

Ознакомиться с возможностью построения 3D-модели из видеозаписи путём извлечения кадров и их обработки в Metashape. Освоить полный цикл создания модели и сравнить её качество с моделью, построенной на основе фотографий.

Порядок выполнения работы.

1. Запишите непрерывное видео, двигаясь вокруг объекта (тот же, что и в Практике №1) по кругу, держа камеру на одном уровне. Рекомендуемая продолжительность: 20–30 секунд. Разрешение: Full HD (1920x1080) или выше. Старайтесь избегать резких движений и тряски.
2. Извлечение кадров. В Metashape создайте новый проект → Workflow → Add Video... Укажите путь к видеофайлу. В появившемся окне установите параметры извлечения: Интервал: 5–15 кадров (зависит от плавности видео). Лимит количества: 100–200 кадров. Нажмите ОК — Metashape автоматически добавит изображения.
3. Произведите выравнивание фотографий. Workflow → Align Photos. Accuracy: High, Generic preselection: Enabled. Построение плотного облака. Workflow → Build Dense Cloud. Quality: Medium.
4. Построение сетки (Model). Workflow → Build Mesh. Source: Depth Map.
5. Построение текстуры. Workflow → Build Texture. Blending mode: Mosaic.
6. Оценка результата. Визуально оцените плотность и детализацию модели.

Проверьте: замкнутость модели, отсутствие артефактов, корректность текстуры.

7. Сравните модель с моделью из Практики №1: геометрию, точность воспроизведения формы, детализацию текстур.

Контрольные вопросы.

1. Какие преимущества и недостатки имеет использование видео по сравнению с фотосъёмкой?
2. Какие параметры извлечения кадров из видео наиболее сильно влияют на результат?
3. Почему важно избегать резких движений при съёмке?
4. Какие этапы обработки в Metashape являются критичными для качества модели?
5. Чем может быть полезна предварительная стабилизация видео перед извлечением кадров?

Лабораторная работа №1. «Определение расстояния до объекта по стереопаре».

Тема «Определение расстояния до объекта по стереопаре».

Цель работы.

Научиться вычислять расстояние до объекта по его изображениям, полученным со смещением (стереопара), используя принципы стереозрения. Освоить ручной расчёт глубины и сравнить его с автоматическим построением карты глубины с помощью библиотеки OpenCV.

Задачи.

1. Получить стереопару с использованием камеры смартфона.
2. Провести калибровку камеры и получить параметры.
3. Рассчитать расстояние вручную по формулам геометрии.
4. Построить карту глубины с использованием OpenCV.
5. Проанализировать результаты и точность.

Порядок выполнения работ.

1. Получение стереопары. Закрепите смартфон и сделайте два снимка: один — сместите камеру немного вправо (на ~6–10 см). Постарайтесь не менять наклон, высоту и фокус. Объект должен быть статичным, хорошо видимым, находиться ближе к центру.
2. Калибровка камеры (если не сделана ранее)

Используйте шаблон шахматной доски и сделайте 10–15 фото под разными углами. Примените OpenCV-функции `cv2.findChessboardCorners()` и `cv2.calibrateCamera()`. Сохраните фокусное расстояние f и коэффициенты искажения.

3. Ручной расчёт глубины. Найдите одну и ту же точку на левом и правом изображениях. Измерьте координаты x_L и x_R — горизонтальные координаты точки.

Вычислите диспаратность: $d = x_L - x_R$.

Найдите расстояние: $Z = (f \cdot B) / d$. Фокусное расстояние должно быть в пикселях, B — в метрах.

4. Автоматическое построение карты глубины. Пример кода:

```
import cv2
import numpy as np
from matplotlib import pyplot as plt

# Загрузка изображений
imgL = cv2.imread('left.jpg', cv2.IMREAD_GRAYSCALE)
imgR = cv2.imread('right.jpg', cv2.IMREAD_GRAYSCALE)

# Создание объекта StereoBM
stereo = cv2.StereoBM_create(numDisparities=64, blockSize=15)

# Построение карты диспаратности
```

```
disparity = stereo.compute(imgL, imgR)
```

```
# Отображение
```

```
plt.imshow(disparity, 'gray')
```

```
plt.title('Disparity Map')
```

```
plt.colorbar()
```

```
plt.show()
```

Если известны f и B , можно перевести карту диспаратности в глубину:

```
f = 700 # фокусное расстояние в пикселях (пример)
```

```
B = 0.07 # базовая линия в метрах
```

```
# Чтобы избежать деления на 0
```

```
disparity[disparity == 0.0] = 0.1
```

```
depth = f * B / disparity
```

Контрольные вопросы.

1. Какова роль базовой линии в стереозрении?
2. Почему важно калибровать камеру?
3. Как связаны диспаратность и расстояние до объекта?
4. В каких случаях автоматическая карта глубины будет неточной?
5. Каковы преимущества и ограничения ручного расчёта?

Отчёт по работе должен быть оформлен в соответствии с ОС ТУСУР и содержать:

1. Стереопара, полученная студентом;
2. Измеренные координаты точек и расчёт глубины вручную;
3. Построенная карта диспаратности;
4. Изображения карты глубины;
5. Сравнение расстояний, полученных ручным и автоматическим методами.

Лабораторная работа №2. «Построение цифровой 3D-модели объекта по его изображениям с использованием контрольных точек».

Тема «Построение цифровой 3D-модели объекта по его изображениям с использованием контрольных точек».

Цель работы.

Освоить процесс построения цифровой 3D-модели фасада здания на основе серии фотографий, сделанных с разных точек, с использованием заранее нанесённых контрольных точек с известными координатами для повышения точности моделирования.

Порядок выполнения работы.

1. Съёмка фасада. Перед началом необходимо самостоятельно выполнить съёмку фасада здания по следующим правилам:
 - Съёмка производится фронтально или с небольшим смещением (до 10–15 градусов).
 - Камера располагается на одном уровне с фасадом (по возможности).
 - Общее количество снимков: не менее 40–50, с перекрытием.
2. Импорт фотографий в Metashape. Создайте новый проект в Metashape: File → New. Импортируйте фотографии: Workflow → Add Photos. Убедитесь, что фотографии загружены корректно, отображаются в панели “Photos”.
3. Выравнивание изображений. Выберите Workflow → Align Photos. Параметры: Accuracy: High. Key point limit: 40,000. Generic preselection: Enabled.

По завершении выравнивания должна появиться сцена с облаком точек и позицией камер.

4. Добавление контрольных точек. Перейдите в панель “Markers”. Щёлкните правой кнопкой на изображении в области, где находится контрольная точка, и выберите Add Marker. Назначьте этой точке уникальное имя (например, CP1). Повторите для всех контрольных точек. Перейдите во вкладку Reference → Markers. Введите известные координаты (или импортируйте файл с координатами точек через Import Reference).
5. После ввода координат: убедитесь, что камеры и маркеры включены (галочки в Reference). Выполните Tools → Optimize Cameras. Проверьте RMS-ошибки в окне Reference. ВАЖНО, чтобы они не превышали 1–3 пикселя (или соответствующую точность в метрах/сантиметрах).
6. Построение плотного облака и 3D-сетки. Workflow → Build Dense Cloud. Quality: Medium или High. Workflow → Build Mesh. Source: Depth maps. Surface type: Arbitrary. Workflow → Build Texture. Mapping mode: Generic. Blending mode: Mosaic.
7. Анализ результатов. Осмотрите полученную модель: есть ли искажения, артефакты. Проверьте видимость и точность расположения контрольных точек. Сравните размеры элементов фасада с реальными (по контрольным точкам или другим известным размерам).

Отчёт по работе должен быть оформлен в соответствии с ОС ТУСУР и содержать:

1. Тему и цель работы;
2. Краткое описание съёмки;
3. Исходные фотографии (миниатюры);
4. Расположение камер;
5. Облако точек;

6. Модель фасада;
7. Таблицу ошибок (RMS) по контрольным точкам.

Контрольные вопросы.

1. Что такое контрольная точка и зачем она используется?
2. Какие параметры влияют на точность фотограмметрической модели?
3. В чём заключается этап оптимизации камер?
4. Почему важно, чтобы контрольные точки были видны на нескольких изображениях?
5. Как интерпретировать RMS-ошибку привязки?

Лабораторная работа №3. «Построение цифровой 3D-модели объекта по его изображениям без использования контрольных точек».

Тема «Построение цифровой 3D-модели объекта по его изображениям без использования контрольных точек».

Цель работы.

Научиться строить цифровую 3D-модель фасада здания на основе серии фотографий, снятых смартфоном, с использованием геотегов и данных ориентации камеры, автоматически полученных с устройства, без применения контрольных точек для привязки.

Задачи.

1. Выполнить съёмку фасада здания с использованием смартфона, обеспечив наличие в EXIF-метаданных GPS-координат и данных ориентации камеры.
2. Импортировать изображения в программу Agisoft Metashape и извлечь пространственные метаданные.
3. Построить цифровую 3D-модель здания по этим данным без использования контрольных точек.
4. Визуализировать модель, оценить геометрию и качество привязки.
5. Сопоставить контрольные точки на изображениях с их известными координатами.
6. Оценить точность пространственной привязки построенной модели по координатным ошибкам контрольных точек.
7. Сделать выводы о применимости метода и влиянии используемых метаданных на точность результата.

Порядок выполнения работы.

1. Подготовка к съёмке. Перед съёмкой убедитесь: что на смартфоне включены GPS и автосохранение местоположения в EXIF данные изображений; не используется широкоугольный режим; что смартфон делает фотографии с меткой ориентации (данные с инерционной системы).
2. Съёмка фасада. Сделать не менее 40–50 перекрывающихся изображений фасада с разных точек. Держать смартфон устойчиво, желательно на уровне глаз, с небольшими изменениями угла съёмки. Обеспечить перекрытие между снимками 60–70%. Не перемещать смартфон резко и не менять фокусировку.
3. Импортируйте изображения и метаданные в Metashape. File → New project. Workflow → Add Photos — загрузите фотографии. Tools → Preferences → General → Coordinate System — проверьте, чтобы была выбрана WGS84 или локальная система координат. Перейдите во вкладку Reference и проверьте: загружены ли координаты и ориентации (в столбцах X, Y, Z и Roll, Pitch, Yaw).
4. Выравнивание фотографий. Workflow → Align Photos. Accuracy: High. Pair preselection: Generic + Reference. Adaptive camera model fitting: Enabled. После выравнивания проверьте, появилась ли сцена с позицией камер и облаком точек.
5. Построение плотного облака точек и модели. Workflow → Build Dense Cloud. Quality: Medium или High. Workflow → Build Mesh. Source: Depth Maps. Surface Type: Arbitrary. Workflow → Build Texture. Mapping Mode: Generic. Blending Mode: Mosaic.
6. Оценка точности с использованием контрольных точек. Этот шаг выполняется только для оценки точности модели, контрольные точки не участвуют в выравнивании!!! Добавьте метки вручную. Перейдите в окно “Photos”. Щёлкните

правой кнопкой мыши по контрольной точке на фото → Add Marker. Назначьте имя (например, GCP1, GCP2...). Перейдите во вкладку Reference → Markers. Введите известные координаты точек. Убедитесь, что для каждой точки задано не менее 3-х измерений на разных изображениях. Снимите галочку напротив этих точек в колонке "Used" (чтобы они не влияли на выравнивание). Metashape рассчитает ошибки позиционирования по контрольным точкам, они отразятся в столбце "Error".

7. Анализ результатов. Посмотрите на ошибки по координатам контрольных точек. Сравните точность с предыдущей лабораторной работой, где использовались контрольные точки в качестве опорных. Обратите внимание на: равномерность распределения ошибок; соответствие масштаба модели; смещения или вращения модели относительно реального объекта.

Отчёт по работе должен быть оформлен в соответствии с ОС ТУСУР и содержать:

1. Тему, цель и краткое описание съёмки;
2. Таблицу с EXIF-данными одной или нескольких фотографий (GPS и ориентация);
3. Скриншоты: положения камер (изображений, облака точек, построенную модель);
4. Таблицу с координатами контрольных точек и ошибками;
5. Сравнить результаты с ЛР2;
6. Выводы.

Контрольные вопросы.

1. Какие данные в EXIF метаданных используются для привязки в Metashape?
2. Какую точность дают GPS-данные смартфона?
3. Почему контрольные точки нужны даже при наличии GPS-меток?
4. Что означает RMS-ошибка в оценке модели?
5. Какие ограничения у метода построения без использования контрольных точек?

Лабораторная работа №4. «Расчёт объема насыпи по цифровой 3D-модели».

Тема «Расчёт объема насыпи по цифровой 3D-модели».

Цель работы.

Освоить метод построения цифровой 3D-модели объекта по изображениям и научиться выполнять расчет объема насыпи с использованием программного обеспечения Agisoft Metashape.

Задачи.

1. Импортировать изображения насыпи в программное обеспечение Agisoft Metashape.
2. Выполнить выравнивание фотографий и построить облако точек.
3. Построить цифровую модель поверхности насыпи.
4. Создать полигональную и текстурированную модель.
5. Задать опорную базовую поверхность для расчета объема.
6. Выполнить расчет объема насыпи.
7. Сравнить полученный объём с данными маркшейдерских измерений.
8. Сделать вывод о точности и применимости метода.

Порядок выполнения работы.

1. Импорт изображений. Откройте Metashape. Создайте новый проект. Загрузите изображения через: Workflow → Add Photos или просто перетащите их в рабочее окно.
2. Выравнивание изображений. Выберите: Workflow → Align Photos. Используйте параметры: Accuracy: High; Generic preselection: Enabled; Key point limit: 40 000; Tie point limit: 4 000. После выравнивания проверьте, корректно ли сформировалось облако точек.
3. Построение плотного облака точек. Выберите: Workflow → Build Dense Cloud. Рекомендуемые параметры: Quality: High; Depth Filtering: Mild.
4. Построение 3D-модели. Перейдите к созданию модели: Workflow → Build Mesh. Выберите источник: Dense cloud, Surface type: Arbitrary, Face count: High.
5. Построение текстуры. Workflow → Build Texture. Mapping mode: Generic, Blending mode: Mosaic, Texture size/count: по умолчанию.
6. Задание базовой поверхности. В панели "Model" выберите инструмент Draw Polygon. Обведите основание насыпи по периметру. Назначьте полигон как base surface для расчета объема.
7. Расчет объема. Выделите полигон → правый клик → Measure Area and Volume; Выберите опцию Measure Volume. В появившемся окне отобразится объем насыпи в кубических метрах. Запишите результат.
8. Сравнение с маркшейдерскими данными. Сопоставьте полученный объём с маркшейдерскими расчётами. Проанализируйте расхождение:

Погрешность = $\frac{(|\text{Объём (фотограмметрия)} - \text{Объём (маркшейдер)}|)}{(\text{Объём (маркшейдер)})} \times 100\%$

Отчёт по работе должен быть оформлен в соответствии с ОС ТУСУР и содержать:

1. Тему, цель и задачи работы;
2. Краткое описание методики построения 3D-модели;

3. Снимки экрана ключевых этапов работы;
4. Построенную 3D-модель насыпи;
5. Значение рассчитанного объема;
6. Таблицу сравнения с маркшейдерскими расчетами;
7. Выводы.

Контрольные вопросы.

1. Какие этапы включает построение 3D-модели в Agisoft Metashape?
2. Какие параметры влияют на точность 3D-модели?
3. Какие источники ошибок могут возникать при фотограмметрическом измерении объема насыпи?
4. Как использование разных высот основания влияет на итоговый объем насыпи?
5. В чём преимущества фотограмметрии перед традиционными маркшейдерскими методами?

Темы для самостоятельного изучения

1. Методы построения 3D-моделей: сравнение фотограмметрии и лазерного сканирования.
2. Технологии дополненной реальности.
3. Алгоритмы построения и визуализации облаков точек.
4. Использование ИИ для реконструкции 3D-геометрии из фотографий.
5. Принципы работы IMU и GPS в фотограмметрии.
6. Мобильные приложения для 3D-сканирования.
7. Создание цифровых двойников объектов на основе изображений.
8. Основы освещения и материалов в фотореалистичной визуализации 3D-сцен.
9. Методы измерения точности 3D-моделей, полученных из изображений.