

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

Работа выполнена в СКБ «Интеллектуальные технологии»

СОГЛАСОВАНО

Начальник отдела ОНИПКРС
Е.М. Димитриади

(подпись)

« 20 » 06 2023 г.

УТВЕРЖДАЮ

Проректор по научной работе
А.В. Космынин

(подпись)

« 20 » 06 2023 г.

Декан

И.А. Трещёв

И.А. Трещёв

(подпись)

« 20 » 06 2023 г.

«Камера глубины»

Комплект проектной документации

Руководитель СКБ

(подпись)

20.06.2023

(подпись, дата)

Г.В. Москалец

Руководитель проекта

(подпись)

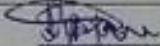

20.06.2023

(подпись, дата)

Г.В. Москалец

Комсомольск-на-Амуре 2023

Карточка проекта

| | | | |
|-----------------|-----------------------|---|--------------------------|
| Название | Камера глубины | | |
| Тип проекта | Тип проекта: | техническое | творчество |
| Исполнители | Студент |  | В.В. Покровский – 0ВТ6-1 |
| | Студент |  | Е.И. Монастырная – 0ИБ-1 |
| Срок реализации | 01.01.2023-30.03.2023 | | |

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

ЗАДАНИЕ
на разработку

Название проекта: «Камера глубины»

Назначение: Предназначен для обнаружения и отображения маски объектов на видеопотоке с камеры Intel RealSense, для их дальнейшей классификации.

Область использования: Предназначен для решения различных задач в области компьютерного зрения, робототехники, мониторинг и безопасность, автоматическое отслеживание.

Функциональное описание проекта: Класс MaskRCNN загружает модель Mask RCNN, устанавливает параметры модели и генерирует цвета для масок объектов. Метод detect_objects_mask выполняет обнаружение объектов и масок на кадре видеопотока, а методы draw_object_mask и draw_object_info отрисовывают маски объектов и отображают информацию о глубине объектов на кадре.

Класс RealsenseCamera настраивает камеру Intel RealSense, запускает видеопоток и возвращает текущий цветовой и глубинный кадры. Основной код использует оба класса для обработки видеопотока с камеры. Он получает кадры, выполняет обнаружение объектов с помощью Mask RCNN и отображает маску объектов на видеопотоке с камеры Intel RealSense.

Техническое описание устройства: Программа состоит из двух классов MaskRCNN и RealsenseCamera. Класс MaskRCNN выполняет обнаружение и отображение масок объектов на видеопотоке, а класс RealsenseCamera обеспечивает взаимодействие с камерой Intel RealSense.

Требования: Камера Intel RealSense для получения видеопотока. Модель Mask RCNN и предварительно обученные данные для обнаружения и классифика

ции объектов. Библиотеки OpenCV и NumPy для обработки видеопотока и работы с массивами. Установленная среда разработки Python для выполнения кода.

План работ:

| Наименование работ | Срок |
|-----------------------------------|---------|
| Подготовка библиотек | 01.2023 |
| Разработка класса MaskRCNN | 01.2023 |
| Разработка класса RealsenseCamera | 02.2023 |
| Интеграция классов в основной код | 02.2023 |
| Тестирование и отладка | 03.2023 |

Комментарии:

Перечень графического материала:

1. Листинги;
2. Изображения;

Руководитель проекта



20.08.2023

(подпись, дата)

Г.В. Москалец

Министерство науки и высшего образования Российской Федерации

Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

ПАСПОРТ
«Камера глубины»

Руководитель проекта



20.06.2023

Г.В. Москалец

(подпись, дата)

Комсомольск-на-Амуре 2023

Содержание

| | |
|--|----|
| Общие положения | 7 |
| 1.1 Наименование изделия | 7 |
| 1.2 Наименования документов, на основании которых ведется проектирование изделия | 7 |
| 1.3 Перечень организаций, участвующих в разработке изделия | 7 |
| 1.4 Сведения об использованных при проектировании нормативно-технических документах..... | 8 |
| 2 Назначение и принцип действия | 9 |
| 2.1 Назначение изделия | 9 |
| 2.2 Области использования изделия | 9 |
| 2.3 Принцип действия изделия..... | 9 |
| 3 Состав изделия и комплектность | 10 |
| 4 4 Технические характеристики | 11 |
| 4.1 Основные технические характеристики камеры Realsense D455 | 11 |
| 5 Устройство и описание работы изделия | 12 |
| 5.1 Описание работы изделия | 12 |
| 6 Условия эксплуатации | 15 |
| 6.1 Правила и особенности размещения изделия..... | 15 |
| 6.2 Меры безопасности..... | 15 |
| 6.3 Правила хранения и транспортирования | 16 |
| ПРИЛОЖЕНИЕ А | 17 |
| ПРИЛОЖЕНИЕ Б..... | 22 |

Общие положения

Настоящий паспорт является документом, предназначенным для ознакомления с основными техническими характеристиками, устройством, правилами установки и эксплуатации устройства «Камера глубины» (далее «изделие»).

Паспорт входит в комплект поставки изделия. Прежде, чем пользоваться изделием, внимательно изучите правила обращения и порядок работы с ним. В связи с постоянной работой по усовершенствованию изделия, повышающей его надежность и улучшающей условия эксплуатации, в конструкцию могут быть внесены изменения, не отраженные в данном издании.

1.1 Наименование изделия

Полное наименование изделия – «Камера глубины».

1.2 Наименования документов, на основании которых ведется проектирование изделия

Проектирование «Камера глубины» осуществляется на основании требований и положений следующих документов:

- задание на разработку.

1.3 Перечень организаций, участвующих в разработке изделия

Заказчиком проекта «Камера глубины» является Федеральное государственное бюджетное образовательное учреждение высшего образования «Комсомольский-на-Амуре государственный университет» (далее заказчик), находящийся по адресу: 681013, Хабаровский край, г. Комсомольск-на-Амуре, Ленина пр-кт., д. 17.

Исполнителями проекта «Камера глубины» являются участники студенческого конструкторского бюро «Интеллектуальные технологии», студенты групп ОВТб-1 Покровский Виктор Владимирович, ОИБ-1 Монастырняя Елизавета Игоревна.

1.4 Сведения об использованных при проектировании нормативно-технических документах

При проектировании использованы следующие нормативно-технические документы:

ГОСТ 2.001-2013. Единая система конструкторской документации.

Общие положения.

ГОСТ 2.102-2013. Единая система конструкторской документации.

Виды и комплектность конструкторских документов.

ГОСТ 2.105-95. Единая система конструкторской документации.

Общие требования к текстовым документам.

ГОСТ 2.610-2006. Единая система конструкторской документации.

Правила выполнения эксплуатационных документов.

ГОСТ 2.004-88. Единая система конструкторской документации.

Общие требования к выполнению конструкторских технологических документов на печатающих и графических устройствах вывода ЭВМ.

ГОСТ 2.051-2006. Единая система конструкторской документации.

Электронные документы. Общие положения.

ГОСТ 2.052-2006. Единая система конструкторской документации.

Электронная модель изделия. Общие положения.

ГОСТ 2.601-2013. Единая система конструкторской документации.

Эксплуатационные документы.

2 Назначение и принцип действия

2.1 Назначение изделия

Камера глубины

В состав изделия входят:

- Паспорт,
- Камера RealSense D455
- Программная реализация.

2.2 Области использования изделия

Изделие может применяться для решения различных задач в области компьютерного зрения, робототехники, мониторинга и обеспечения безопасности, автоматическое отслеживание.

2.3 Принцип действия изделия

Выполняется загрузка модели Mask RCNN, предварительно обученной на наборе данных COCO, которая способна обнаруживать и классифицировать различные объекты, настройка камеры Intel RealSense для получения цветового и глубинного потоков с определенным разрешением и частотой кадров. В цикле while приложение получает текущие кадры с камеры, выполняя обработку в реальном времени. Для каждого кадра приложение вызывает метод detect_objects_mask класса MaskRCNN для обнаружения объектов на цветовом кадре, отрисовывает маску объектов на цветовом кадре с помощью метода draw_object_mask класса MaskRCNN, отображает информацию о глубине объектов на цветовом кадре с помощью метода draw_object_info класса MaskRCNN, отображает измененный цветовой кадр с маской объектов на экране. Приложение проверяет нажатие клавиши на клавиатуре. Если нажата клавиша Esc, цикл прерывается, и приложение завершается.

3 Состав изделия и комплектность

В комплект поставки входит:

- Паспорт,
- Камера RealSense D455
- Программная реализация.

4 4 Технические характеристики

4.1 Основные технические характеристики камеры Realsense D455

Основные технические характеристики камеры Intel Realsense D455 приведены в таблице 1.

Таблица 1 – Технические характеристики камеры Intel Realsense D455

| Наименование параметра | Значение |
|---|-------------------|
| Разрешение кадра RGB | 1280 × 800 |
| Частота кадров RGB | 30 fps |
| Поле зрения датчика RGB (H× W) | 90 × 65° |
| Разрешение датчика RGB | 1 MP |
| Технология камеры глубины | Стереоскопическая |
| Глубина поля зрения (FOV) | 87° × 58° |
| Минимальное расстояние по глубине (Min-Z) при максимальном разрешении | -52 см. |
| Выходное разрешение глубины | 1280 × 720 |
| Частота кадров глубины | 90 fps |
| Интерфейсы | USB-C* 3.1 Gen 1 |
| Питание, В | 5 |

5 Устройство и описание работы изделия

5.1 Описание работы изделия

Данный код представляет собой два класса: MaskRCNN и RealsenseCamera. Класс MaskRCNN отвечает за обнаружение и отображение маски объектов на кадре видеопотока.

В конструкторе класса выполняются следующие действия:

- загрузка модели Mask RCNN с использованием файлов `frozen_inference_graph_coco.pb/mask_rcnn_inception_v2_coco_2018_01_28.pbtxt`;
- установка предпочтительных параметров выполнения модели, таких как бэкенд и целевой уровень вычислений (в данном случае, процессор CPU);
- генерация разных цветов для масок объектов;
- задание пороговых значений для обнаружения объектов и масок.

Метод `detect_objects_mask` принимает BGR-кадр и выполняет следующие действия:

- преобразование кадра в blob формат с использованием функции `cv2.dnn.blobFromImage`;
- установка входных данных для модели с помощью `net.setInput(blob)`;
- получение результата обнаружения объектов и масок с помощью `net.forward(["detection_out_final", "detection_masks"]);`
- обработка результатов обнаружения, включая извлечение координат боксов объектов, классов, центров и контуров объектов;
- Возвращение списков боксов, классов, контуров и центров объектов.

Метод `draw_object_mask` принимает BGR-кадр и выполняет следующие действия: отрисовка маски объектов на кадре путем перебора боксов, классов и контуров объектов, рисование контуров объектов с использованием функции `cv2.drawContours`, заливка областей объектов на кадре с использованием функции `cv2.fillPoly`, возвращение измененного BGR-кадра.

Метод `draw_object_info` принимает BGR-кадр и карту глубины и выполняет следующие действия: отображение информации о глубине объектов на кадре, рисование линий, соединяющих центры объектов с соответствующими границами, рисование прямоугольника и текста, содержащего класс объекта, возвращение измененного BGR-кадра.

Класс `RealsenseCamera` отвечает за взаимодействие с камерой Intel RealSense. В конструкторе класса выполняются следующие действия:

- настройка конфигурации камеры для получения цветовой и глубинного потоков с разрешением 1280x720 и частотой 30 кадров в секунду;
- запуск видеопотока с помощью `pipeline.start(config)`;
- выравнивание кадров цвета и глубины с использованием `align`.

Метод `get_frame_stream` возвращает текущий цветовой и глубинный кадры: ожидание и получение кадров с использованием `pipeline.wait_for_frames()` и `align.process(frames)`, применение фильтров для заполнения отверстий и улучшения изображения глубины, преобразование кадров в массивы NumPy, создание карты цветов для отображения глубины объектов, возвращение флага успеха, цветовой кадра и карты глубины.

Метод `release` останавливает работу камеры Intel RealSense с помощью `pipeline.stop()`. Основной код использует оба класса для обработки видеопотока с камеры. В цикле `while`: получается цветовой и глубинный кадры с помощью `rs.get_frame_stream()`, выполняется обнаружение объектов на цветовой кадре с помощью `mrcnn.detect_objects_mask`, отрисовывается маска объектов на цветовой кадре с помощью `mrcnn.draw_object_mask`, отображается информация о глубине объектов на цветовой кадре с помощью `mrcnn.draw_object_info`, отображается измененный цветовой кадр с маской объектов с помощью `cv2.imshow`.

Проверяется нажатие клавиши на клавиатуре. Если нажата клавиша Esc (код 27), цикл прерывается. После выхода из цикла освобождаются ресурсы с помощью `rs.release()` и `cv2.destroyAllWindows()`.

6 Условия эксплуатации

Изделие выпускается в климатическом исполнении УХЛ 4.2 по ГОСТ 15150-69 и предназначен для использования в стационарных условиях в закрытых помещениях при соответствующих климатических условиях:

- интервал температур от +10 до +35 °С;
- относительная влажность воздуха до 80 % при температуре +25 °С;
- высота над уровнем моря не более 2000 м;
- атмосферное давление от 86,6 до 106 кПа (от 650 до 800 мм рт. ст.).

В помещении, где используется изделие не должно возникать условий для конденсации влаги (выпадения росы). *Изделие является электронным прибором, требующим бережного обращения.*

Для обеспечения безотказной работы, сохранения точности и его сбережения необходимо соблюдать следующие правила:

- изучить паспорт, прежде чем приступить к работе с изделием;
- предохранять изделие от ударов и повреждений;
- *при необходимости указать дополнительные пункты*
- не допускать самостоятельную разборку изделия.

6.1 Правила и особенности размещения изделия

Изделие должно быть расположено на расстоянии не менее 1 м от нагревательных приборов.

ВНИМАНИЕ! При эксплуатации изделия запрещается проводить самостоятельно какие-то либо работы по извлечению и установке внутренних компонентов изделия.

6.2 Меры безопасности

Необходимо соблюдать требования техники безопасности и следующие меры предосторожности:

- *не оставлять изделие включенным без наблюдения;*
- *внутренние осмотры и ремонт изделия должны производиться только квалифицированными специалистами;*

6.3 Правила хранения и транспортирования

Транспортирование изделия в упакованном виде может производиться железнодорожным, автомобильным (в закрытых транспортных средствах), воздушным, речным и морским видами транспорта в соответствии с правилами перевозок грузов, действующих на транспорт данного вида. Условия транспортирования изделия по части воздействия климатических факторов должны соответствовать группе 5 по ГОСТ 15150.

После транспортирования изделие должно быть выдержано не менее 2 часов в транспортной таре при температуре 20 ± 5 °С и относительной влажности воздуха не более 80 %.

Распакованное изделие должно храниться в отапливаемом и вентилируемом чистом помещении при температуре от +5 до +40 °С и относительной влажности воздуха не более 60 %. При температуре ниже 25 °С допускается увеличение относительной влажности до 80 %. Воздух в помещении не должен содержать примесей, вызывающих коррозию металлов, налеты на поверхностях оптических деталей.

ПРИЛОЖЕНИЕ А

(обязательное)



Рисунок А.1 – Камера RealSense D455

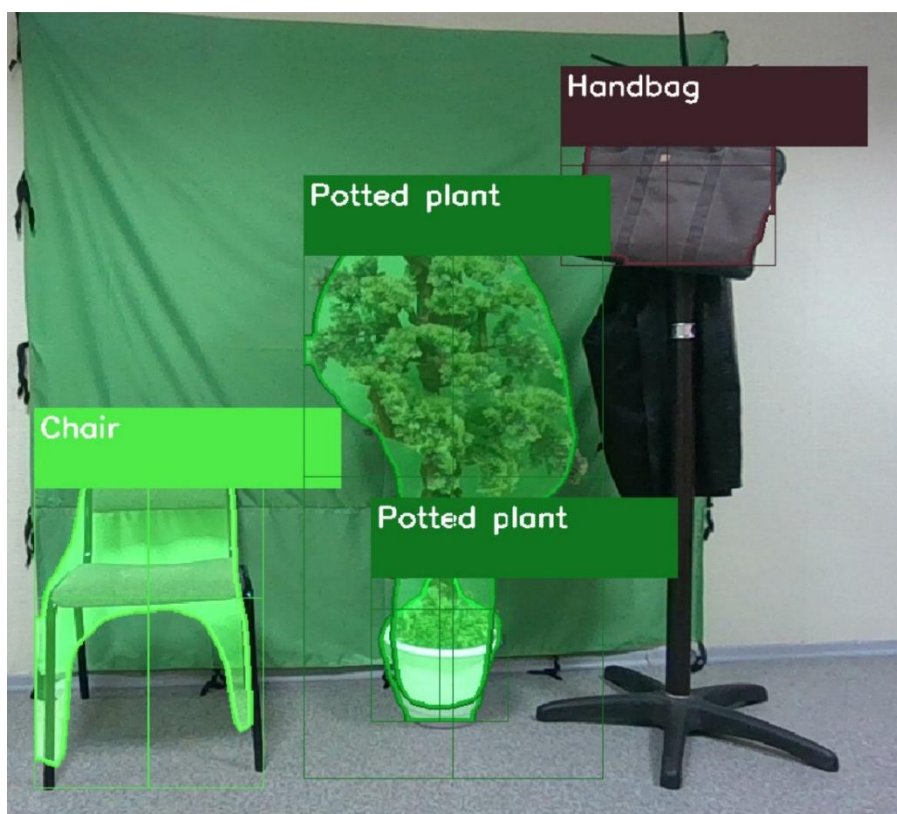


Рисунок А.2 – Распознавание объектов с помощью ИИ

ПРИЛОЖЕНИЕ Б

(обязательное)

Листинг Б.1 – Программная реализация

```
import cv2
import numpy as np

class MaskRCNN:
    def __init__(self):
        # Загрузка Mask RCNN
        self.net =
cv2.dnn.readNetFromTensorflow("dnn/frozen_inference_graph_coco.pb",
"dnn/mask_rcnn_inception_v2_coco_2018_01_28.pbtxt")
        self.net.setPreferableBackend(cv2.dnn.DNN_BACKEND_DEFAULT)
        self.net.setPreferableTarget(cv2.dnn.DNN_TARGET_CPU)

        # Генерация разных цветов для маски
        np.random.seed(2)
        self.colors = np.random.randint(0, 255, (90, 3))

        # Задаем пороги для обнаружения объектов
        self.detection_threshold = 0.7
        self.mask_threshold = 0.3

        self.classes = []
        with open("dnn/classes.txt", "r") as file_object:
            for class_name in file_object.readlines():
                class_name = class_name.strip()
                self.classes.append(class_name)

        self.obj_boxes = []
        self.obj_classes = []
        self.obj_centers = []
        self.obj_contours = []

        # Дистанция (НЕ РАБОТАЕТ, ФИКС)
        self.distances = []

    def detect_objects_mask(self, bgr_frame):
        blob = cv2.dnn.blobFromImage(bgr_frame, swapRB=True)
        self.net.setInput(blob)

        boxes, masks = self.net.forward(["detection_out_final", "detection_masks"])

        # Детектор объектов
        frame_height, frame_width, _ = bgr_frame.shape
        detection_count = boxes.shape[2]

        # Боксы объектов
        self.obj_boxes = []
        self.obj_classes = []
        self.obj_centers = []
        self.obj_contours = []

        for i in range(detection_count):
            box = boxes[0, 0, i]
            class_id = box[1]
            score = box[2]
```

```

        color = self.colors[int(class_id)]
        if score < self.detection_threshold:
            continue

        # Координаты бокса объекта
        x = int(box[3] * frame_width)
        y = int(box[4] * frame_height)
        x2 = int(box[5] * frame_width)
        y2 = int(box[6] * frame_height)
        self.obj_boxes.append([x, y, x2, y2])

        cx = (x + x2) // 2
        cy = (y + y2) // 2
        self.obj_centers.append((cx, cy))

        # Добавляем класс
        self.obj_classes.append(class_id)

        # Контуры
        # Получаем координаты маски
        # Получаем маску
        mask = masks[i, int(class_id)]
        roi_height, roi_width = y2 - y, x2 - x
        mask = cv2.resize(mask, (roi_width, roi_height))
        _, mask = cv2.threshold(mask, self.mask_threshold, 255,
cv2.THRESH_BINARY)
        contours, _ = cv2.findContours(np.array(mask, np.uint8),
cv2.RETR_EXTERNAL, cv2.CHAIN_APPROX_SIMPLE)
        self.obj_contours.append(contours)

    return self.obj_boxes, self.obj_classes, self.obj_contours,
self.obj_centers

    def draw_object_mask(self, bgr_frame):
        # Обнаружение
        for box, class_id, contours in zip(self.obj_boxes, self.obj_classes,
self.obj_contours):
            x, y, x2, y2 = box
            roi = bgr_frame[y: y2, x: x2]
            roi_height, roi_width, _ = roi.shape
            color = self.colors[int(class_id)]

            roi_copy = np.zeros_like(roi)

            for cnt in contours:
                # cv2.f(roi, [cnt], (int(color[0]), int(color[1]),
int(color[2])))
                cv2.drawContours(roi, [cnt], - 1, (int(color[0]),
int(color[1]), int(color[2])), 3)
                cv2.fillPoly(roi_copy, [cnt], (int(color[0]), int(color[1]),
int(color[2])))
                roi = cv2.addWeighted(roi, 1, roi_copy, 0.5, 0.0)
            bgr_frame[y: y2, x: x2] = roi
        return bgr_frame

    def draw_object_info(self, bgr_frame, depth_frame):
        # Проходим обнаружением
        for box, class_id, obj_center in zip(self.obj_boxes,
self.obj_classes, self.obj_centers):
            x, y, x2, y2 = box

            color = self.colors[int(class_id)]
            color = (int(color[0]), int(color[1]), int(color[2]))

```

```

        cx, cy = obj_center

        depth_mm = depth_frame[cy, cx]

        cv2.line(bgr_frame, (cx, y), (cx, y2), color, 1)
        cv2.line(bgr_frame, (x, cy), (x2, cy), color, 1)

        class_name = self.classes[int(class_id)]
        cv2.rectangle(bgr_frame, (x, y), (x + 250, y + 70), color, -1)
        cv2.putText(bgr_frame, class_name.capitalize(), (x + 5, y + 25),
0, 0.8, (255, 255, 255), 2)
        cv2.rectangle(bgr_frame, (x, y), (x2, y2), color, 1)
    return bgr_frame
import cv2
from realsense_camera import *
from mask_rcnn import *

# Загрузка камеры
rs = RealsenseCamera()
mrcnn = MaskRCNN()

while True:
    # Получение кадра в реальном времени с камеры Realsense
    ret, bgr_frame, depth_frame = rs.get_frame_stream()

    # Получение маски объектов
    boxes, classes, contours, centers = mrcnn.detect_objects_mask
(bgr_frame)

    # Отрисовка маски объектов
    bgr_frame = mrcnn.draw_object_mask (bgr_frame)

    # Отображение информации о глубине объектов
    mrcnn.draw_object_info (bgr_frame, depth_frame)

    # Отображение кадра с маской объектов
    cv2.namedWindow ('Realsense OpenCV', cv2.WINDOW_NORMAL)
    cv2.imshow ('Realsense OpenCV', bgr_frame)

    key = cv2.waitKey (1)
    if key == 27:
        break

rs.release()
cv2.destroyAllWindows()
import pyrealsense2 as rs
import numpy as np
import cv2

class RealsenseCamera:
    def __init__(self):
        # Конфигурация камеры и цветов
        print ("Запуск камеры Intel RealSense")
        self.pipeline = rs.pipeline ()

        config = rs.config ()
        config.enable_stream (rs.stream.color, 1280, 720, rs.format.bgr8, 30)
        config.enable_stream (rs.stream.depth, 1280, 720, rs.format.z16, 30)

        # Запуск видеопотока
        self.pipeline.start (config)
        align_to = rs.stream.color
        self.align = rs.align (align_to)

```

```

def get_frame_stream(self):
    # Ожидание цветовой пары глубины и цвета
    frames = self.pipeline.wait_for_frames ()
    aligned_frames = self.align.process (frames)
    depth_frame = aligned_frames.get_depth_frame ()
    color_frame = aligned_frames.get_color_frame ()

    if not depth_frame or not color_frame:
        # Если кадра нет, то возможно камера не подключена
        print (
            "Ошибка, невозможно получить кадр, убедитесь, что камера In-
tel Realsense правильно подключена")
        return False, None, None

    # Применить фильтр, чтобы заполнить отверстия в изображении глубины
    spatial = rs.spatial_filter ()
    spatial.set_option (rs.option.holes_fill, 3)
    filtered_depth = spatial.process (depth_frame)

    hole_filling = rs.hole_filling_filter ()
    filled_depth = hole_filling.process (filtered_depth)

    # Создадим карту цветов, чтобы показать глубину объектов
    colorizer = rs.colorizer ()
    depth_colormap = np.asanyarray (colorizer.colorize
(filled_depth).get_data ())

    # Преобразование изображений в массивы numpy
    depth_image = np.asanyarray (filled_depth.get_data ())
    color_image = np.asanyarray (color_frame.get_data ())

    # Изменение цветовой карты глубины до того же размера, что и цветное
изображение
    depth_colormap = cv2.resize (depth_colormap, (color_image.shape[1],
color_image.shape[0]))

    return True, color_image, depth_colormap

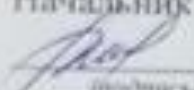
def release(self):
    self.pipeline.stop ()

```

Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное
образовательное учреждение высшего образования
«Комсомольский-на-Амуре государственный университет»

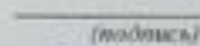
СОГЛАСОВАНО

Начальник отдела ОНиПКРС
Е.М. Димитриади


(подпись)
« 20 » 06 2023 г.

УТВЕРЖДАЮ

Проректор по научной работе
А.В. Космынин


(подпись)
« 20 » 06 2023 г.

Декан
И.А. Трещёв


(подпись)

АКТ

о приемке в эксплуатацию проекта
«Камера глубины»

г. Комсомольск-на-Амуре

« 20 » 06 2023 г.

Комиссия в составе представителей:

со стороны заказчика

- Г.В. Москалец – руководитель СКБ,
- И.А. Трещёв – декан ФКТ

со стороны исполнителя

- Г.В. Москалец – руководитель проекта,
- В.В. Покровский – ОВТб-1
- Е.И. Монастырная – ОИБ-1
- составила акт о нижеследующем:

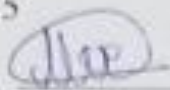
«Исполнитель» передает проект «Камера глубины», в составе:

1. Паспорта

2. Программой реализации

3. RealSense D455

руководитель проекта



20.06.2023

Г.В. Москалец

(подпись, дата)

Исполнители проекта



20.06.2023

В.В. Покровский

(подпись, дата)

Исполнители проекта



20.06.2023

Е.И. Монастырная

(подпись, дата)