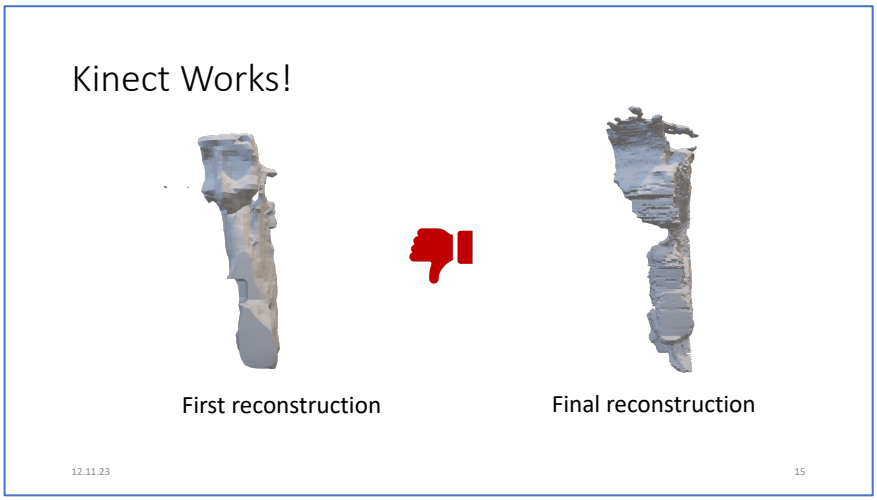
Report week 7-8

Progress report for the MOT & 3D object reconstruction bachelor's thesis.

spraypaintFEDONA

During the previous meeting I presented the results of running the BundleSDF code on a dataset, spraypaintFEDONA, made by me recording the images with the Microsoft Kinect camera and SDK recorder of a spray paint can.
The results were terrible since during the run of the code there were major issues with the pose estimation, and the pose reconstruction was sparse and full of holes.



My conclusions of this first run were that probably the high resolution of the input images and the used object where not optimal as input for the BundleSDF code.
Right after the meeting I understood to have made some significant and obvious mistakes: the RGB input images and the depth images had different resolutions! Especially I completely forgot to consider that the depth camera is a completely different camera from the RGB one, and this has a different calibration matrix.
What I should have better done was to exploit the SDK library to transform the depth images into the RGB camera calibration.

Coding

Successively I decided to clone the SDK github homepage to study the code examples for producing my own code for recording RGBD images with the same calibration (and make it possible to view what I am currently recording, since the normal sdkrecorder does not allow to record and view at the same time).
I found the c/c++ library to be fairly complicated and opted to use instead the respective python wrapper "pyk4a".
With a small python script, I am now able to view and record RGBD images, but I still have to correct this code since I am having issues with the saved files: RGB images have a blue tint, I believe that the red and blue channels are being somehow switched.

Next objectives

I am going to correct the python script to be able to finally record a correct version of spraypaintFEDONA. Successively, for the sake of curiosity, I wanted to try comparing the bundleSDF reconstruction of both correct and wrong datasets on the same RGB images.
Moreover, I am eager to start recording different and more complicated objects, as well as testing object material and object obstructions.

Pyk4a code

```
from argparse import ArgumentParser

import pyk4a
from helpers import colorize
from pyk4a import Config, ImageFormat, PyK4A, PyK4ARecord
import numpy as np
import cv2
import os
from PIL import Image

parser = ArgumentParser(description="pyk4a recorder")
parser.add_argument("--device", type=int, help="Device ID", default=0)
args = parser.parse_args()

print(f"Starting device #{args.device}")
config = Config(
            color_resolution=pyk4a.ColorResolution.RES_720P,
            depth_mode=pyk4a.DepthMode.NFOV_UNBINNED,
            synchronized_images_only=False,
            )
device = PyK4A(config=config, device_id=args.device)
device.start()

rgb_folder = "rgb_images"
depth_folder = "depth_images"

os.makedirs(rgb_folder, exist_ok=True)
os.makedirs(depth_folder, exist_ok=True)
imagesCount = 0

try:
    print("Recording... Press CTRL-C to stop recording.")
    while True:
        capture = device.get_capture()

        if np.any(capture.color):
            cv2.imshow("k4a", capture.color[:, :, :3])
            #cv2.imshow("Transformed Depth",
colorize(capture.transformed_depth, (None, 5000)))
            key = cv2.waitKey(10)

            rgb = capture.color[:, :, :3]
            rgb = (rgb * 255).astype(np.uint8)
            current_rgb = Image.fromarray(rgb, "RGB")
            #current_rgb = Image.fromarray(capture.color[:, :, :3])
```

```
            current_depth =
Image.fromarray(colorize(capture.transformed_depth, (None, 5000)))

            path_1 = os.path.join(rgb_folder,
f"image_{imagesCount}.jpg")
            path_2 = os.path.join(depth_folder,
f"image_{imagesCount}.jpg")

            current_rgb.save(path_1)
            current_depth.save(path_2)

            imagesCount = imagesCount + 1
            if key != -1:
                cv2.destroyAllWindows()
                break

except KeyboardInterrupt:
    print("CTRL-C pressed. Exiting.")
    device.stop()

print(f"{imagesCount} frames written.")
```