Andrea Fedon 12.11.23

Report week 8-9

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

spraypaintFEDONA

During this past weekend I kept working on developing a suitable dataset to feed to the BundleSDF program. I complete the code of a python script that exploits the library "pyk4a" to connect to the Kinect while streaming and recording the images.

RGB and Depth

Testing this code, I encountered the issue that when saving the Kinect RGB and depth captures as PNG files, the streaming was glitchy and the output images were not continuous with each other's. This issue was not present when saving the captures as JPG files. My opinion is that writing and streaming less compressed PNG files was particularly costly and this caused interferences during the program execution. The JPG files were in the order of 100kB of size, while PNGs with the same resolution have size of 2MB.
BundleSDF does not support JPG files as input images, thus I decided to record JPG files (to avoid the "interferences issue"), and then to use a bash script to convert all the files to PNG. This produced PNG files with size of 1MB.

Masks

Regarding the masks, I am producing them from the RGB JPG files using the MiVOS code, and directly saving them as PNG files. To obtain the same resolution as the input images I have to modify the code adding a resize method that scales the image to the correct format. This results in low quality masks that are a bit blurred at the edges.

Result

I do find all this steps to be very laborious and I am thinking about finding another solution for recording the datasets, or to make it automated through bash scripts.
Comparing the files produced with this method to the ones given as samples by the BundleSDF paper, they have all the same PNG, mines have higher resolution quality (1280x720 vs 640x480) but the size of an input RGB frame is much higher (1MB vs 50kB), my masks are a bit blurry at the edges instead of having a sharp edge.

BundleSDF

After having prepared a very small video (150 frames) following the above steps, I ran the BundleSDF code on the dataset without encountering any issue. I cannot present any result yet because I did not have time to wait for the complete execution. Although from the BundleSDF GUI I did observe that it was having issues in estimating the pose of the spray-paint can while I was rotating the object side to side.

Next objectives

The next objectives remain the same as the previous week, since I had not enough time to complete them: compare the last ran of the BundleSDF code with the wrong-calibrated

version; prepare new videos with more complicated objects and test object material and obstructions.
Would it make sense to test the brightness of the RGB images? What about giving flat depth data as input to BundleSDF and check its performance under this condition?