Enhanced visualization: assignment 5

Stereopsis

I. Theoretical part

A. 3D point coordinate from image buffer coordinate.

1. Give the relationship between the disparity in the image buffers of the following stereo system:
   - The cameras have the same intrinsic parameters, $f_x = -f_y$, $f_\theta = 0$ and o_x, o_y.
   - The left camera reference frame coincides with the world frame, while the right camera optical axis is parallel to the left camera optical axis. The right camera is translated on the x axis by t_x with respect to the left camera.

II. Practical part

A. Simulation of a stereo system. In this part we are going to simulate in Python how a 3D point is projected in the image buffers of a stereo system. After simulation, we will retrieve the 3D point coordinates from the projections on the image buffers.

In all the following simulations we consider the 3D point $\mathbf{u}_w = [3\,1\,-2]^\mathrm{T}$. Both cameras have focal lengths in pixels $f_x = -f_y = 50$, their resolutions are both 500×500 pixels and the centers of the image buffers are considered to be on the top left.

1. For the simplified stereo system of the theoretical part, write the intrinsic matrices, extrinsic matrices and camera matrices of both cameras.
2. Evaluate the projection of the point $\mathbf{u}_w$ on the image buffers and draw the image buffers using pyplot command imshow.
3. Retrieve the 3D point coordinate from the image buffer disparity using the result of the theoretical part.
4. Retrieve the 3D point using the general linear approach.
5. Add zero mean Gaussian noise on the observed image buffer to simulate small errors on the matching procedure. Consider a standard deviation of $\sigma = 0.1$ for the noise on both coordinates. Retrieve the 3D point using the general linear approach. What do you observe?

6. Simulate what is observed on the image buffers by considering that the right camera has also a rotation matrix R corresponding to a rotation of 45° around the z axis.

7. Retrieve the 3D point coordinate using the general linear approach both with a perfect matching (no noise) and with Gaussian noise as previously considered.

B. Stereo matching and reconstruction with pyOpenCV

1. In *moodle* you can find two pictures *aloeL.jpg* and *aloeR.jpg* obtained with a simplified stereo system as the one considered in the theoretical part. The two pictures are shown in Figure 1. Download the two pictures and the Python notebook `stereo_aloe`*.ipynb* (also from *moodle*).

   This code is a commented version of the Python Open CV tutorial that you can find here `https://github.com/opencv/opencv/blob/master/samples/python/stereo_match.py`.

2. Read the code and test its cells. After reading, write a short summary of the main steps for 3D reconstruction with a stereo system.

3. If you want to see a colored 3D mesh from the point cloud in the file *aloe.ply*, you can download Meshlab at `http://www.meshlab.net/` and then import the *aloe.ply* file by clicking on "Import Mesh" (in Linux you may need to open Meshlab with the terminal command LC_ALL=C meshlab if the .ply file is not imported properly).

Figure 1: Left (top) and right images taken with a simple stereo system.