# Project work: Perception and Situation Understanding

Part: Prof. Dr.-Ing. Stache

## 0. Fill out the following:

Name: Semyon Kondratev

Matrikel-Nr. / Student-ID: 207612

I confirm that the work I handed in is my own work. All sources are cited.   
I know that my work will be checked for plagiarism, and I accept that any occurrence of plagiarism and/or non-cited sources will lead to a grade of 5.0 (failed / nicht bestanden). The due date for this work to upload to ILIAS is July 30, 2021, time: 23:55, uploads after this due are not accepted.

## 1. Download the camera data from ILIAS and unzip the files (e.g. use 7zip)

The data contain images from a stereo rig (left and right camera) which have already been rectified.

The data of the virtual rectified camera is in the file camera.txt. The parameters in the section [EXTERNAL] describe the extrinsic parameters of the left rectified camera, which is the reference camera.

## 2. Check Image image0110\_c0.pgm (in the folder of the left images)

Note: This task can be done by reading the pixel coordinates by hand and doing the calculations with MATLAB or Python.



Enter the pixel position of the marked corner in the left image:

Pixel, („x-coordinate in the image”)

Pixel, („y-coordinate in the image”)

A picture containing text, outdoor, sky, way

Description automatically generated

Do the same for the corresponding right image:

Pixel,

Pixel

A picture containing text, outdoor, way, road

Description automatically generated

Enter the pixel-accurate disparity

Pixel

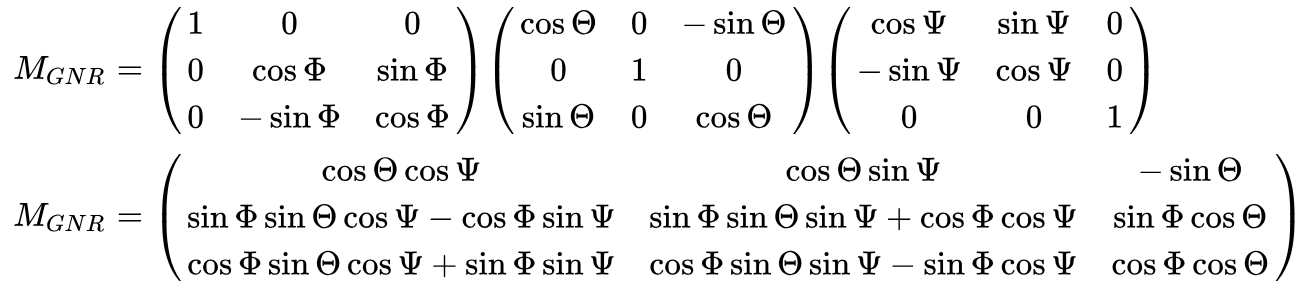
Where is this point located in the world (use the external camera parameters from camera.txt to locate the left rectified camera in the world frame)?

Note: To create a rotation matrix out of tilt, yaw and roll angle, assume the following:

- rotations around fixed rotation axes

- order of rotation: first apply yaw angle, second is tilt , and finally the roll angle

🡪 this results in a rotation matrix of:

  
(Source: [https://de.wikipedia.org/wiki/Eulersche\_Winkel#Standard-x-Konvention\_.28z.2C\_x.E2.80.B2.2C\_z.E2.80.B3.29](https://de.wikipedia.org/wiki/Eulersche_Winkel), 19.12.2016)

Note that in the world frame the x-Axis points to the front, the y-Axis points to the left and the z-Axis points to the top (according to DIN70000):

(x-coordinate in world);

(y-coordinate in world);

(z-coordinate in world);

Write down all processing steps (you shall enter equations and paste Matlab/Python printout below this line in the document).

**Please write down how you validated your calculations.**



,

where sy – pixel size Y-axis

,

K =

Table

Description automatically generated

Text, letter

Description automatically generated

Euler angles:

Graphical user interface, text

Description automatically generated

Graphical user interface, text

Description automatically generated

Text

Description automatically generated with medium confidence

Text, letter

Description automatically generated in Pixels

Text

Description automatically generated X, Y, Z coordinates in meters.

3. Compute a disparity map using normalized cross correlation for the images   
 image0110\_c0.pgm and image0110\_c1.pgm. Try to find an optimal window size.

Paste the disparity image here (use colormap jet for coloring the image):

(References used: https://www.programmersought.com/article/50045374193/)

A map of the world

Description automatically generated with medium confidence

I used the window-size 51x51 pixels

## 4. Feature matching and ICP

1. Use image0110\_c0.pgm to find corner-like features in the image
2. Compute the 3D-World coordinates for the corners (you may use the disparity image form before)
3. Do the same (step 1+2) for image0112\_c0.pgm
4. Try to find a match between the point clouds in the world by using ICP
5. Insert a plot of the 3D-Points matched by ICP from top perspective. Draw the points of image 13 in red and the points of image 14 in blue.
6. How much has the vehicle moved?
7. Hand in matlab or python file / or insert code below (note: all code must be executable!).

Note: If you find a better pair of images in the sequence to show matching and ICP, feel free to use it.

Detecting key points using ORB detector as SIFT is patented

(Source: <https://docs.opencv.org/4.5.2/d1/d89/tutorial_py_orb.html>)

***All of code is in .ipynb file handed. The results for 4 task also there.***