Stereo Robot Navigation

1) Objective

Given a video sequence taken by a stereo camera mounted on a moving vehicle, project's objective is to sense information concerning the space in front of the vehicle which may be deployed by the vehicle navigation system to automatically avoid obstacles.

2) Dataset

The input data consist of a pair of synchronized videos taken by a stereo camera (*robotL.avi, robotR.avi*), with one video concerning the left view (*robotL.avi*), the other the right view (*robotR.avi*). Moreover, the parameters required to estimate distances from stereo images are provided below:

- focale **f** = 567.2 pixel
- baseline **b** = 92.226 mm

In case of problems associated with reading the video files, students may try to install an Indeo CODEC, e.g. after downloading it from the following URL:

http://downloadcenter.intel.com/Detail_Desc.aspx?strState=LIVE&ProductID=355&DwnldID=2846

3) Functional Specifications

Sensing of 3D information related to the obstacles in front of the vehicle should rely on the stereo vision principle. Purposely, students should develop an area-based stereo matching algorithm capable of producing a dense disparity map for each pair

of synchronized frames and based on the SAD (Sum of Absolute Differences) dissimilarity measure.

For each pair of candidate corresponding points, the basic stereo matching algorithm consists in comparing the intensities belonging to two squared windows centred at the points. Such a comparison involves computation of either a dissimilarity (e.g. SAD, SSD) or similarity (e.g. NCC, ZNCC) measure between the two windows. As the matching process is carried out on rectified images, once a reference image is chosen (e.g. the left view), the candidates associated with a given point need to be sought for along the same row in the other image (right view) only and, usually, within a certain disparity range which depends on the depth range one wishes to sense. Accordingly, given a point in the reference image, the corresponding one in the other image is selected as the candidate minimizing (maximizing) the chosen dissimilarity (similarity) measure between the windows. As such, the parameters of the basic stereo matching algorithm consist in the size of the window and the disparity range. In this project, students should choose the former properly, while the latter is fixed to the interval [0, 128].

The main task of the project requires the following steps:

1. Computing the disparity map in a central area of the reference frame (e.g. a squared area of size 60x60, 80x80 o 100x100 pixels), so to sense distances in the portion of the environment which would be travelled by the vehicle should it keep a straight trajectory.

2. Estimate a *main disparity* (d_{main}) for the frontal (wrt the camera) portion of the environment based on the disparity map of the central area of the reference frame computed in the previous step, e.g. by choosing the average disparity or the most frequent disparity within the map.

3. Determine the distance (z, in *mm*) of the obstacle wrt to the moving vehicle based on the *main disparities* (in pixel) estimated from each pair of frames:

$$z(mm) = \frac{b(mm) \cdot f(pixel)}{d_{main}(pixel)}$$

4. Generate a suitable output to convey to the user, in each pair of frame, the information related to the distance (converted in meters) from the camera to the obstacle. Moreover, an alarm should be generated whenever the distance turns out below 0.8 meters.

5. Compute the real dimensions in *mm* (**W,H**) of the chessboard pattern present in the scene.

Purposely, the OpenCV functions `cvFindChessboardCorners` and `cvDrawChessboardCorners` may be deployed to, respectively, find and display the pixel coordinates of the internal corners of the chessboard. Then, assuming the chessboard pattern to be parallel to the image plane of the stereo sensor, the real dimensions of the pattern can be obtained from their pixel dimensions (**w,h**) by the following formulas:

$$W(mm) = \frac{z(mm) \cdot w(pixel)}{f(pixel)}$$

$$H(mm) = \frac{z(mm) \cdot h(pixel)}{f(pixel)}$$

Moreover, students should compare the estimated real dimensions to the known ones (*125 mm x 178 mm*) during the first approach manoeuvre of the vehicle to the pattern, so to verify that accuracy becomes higher as the vehicle gets closer to the pattern. Students should also comment on why accuracy turns out worse during the second approach manoeuvre.
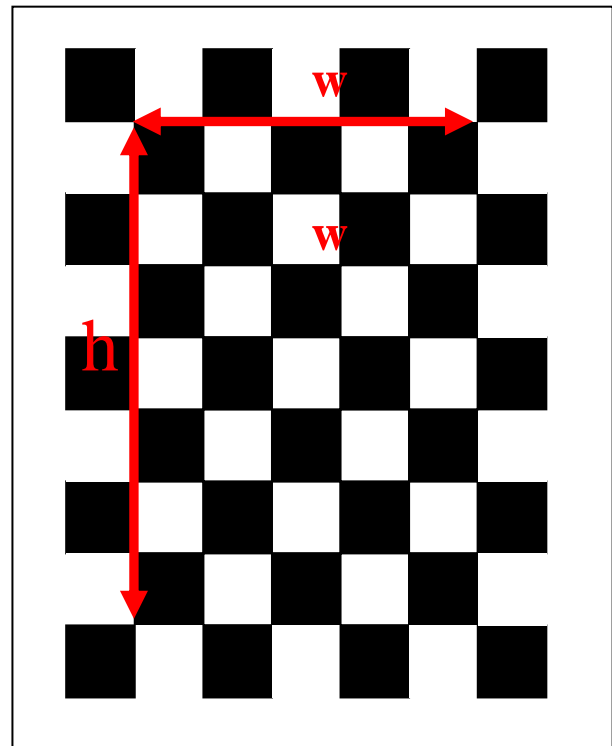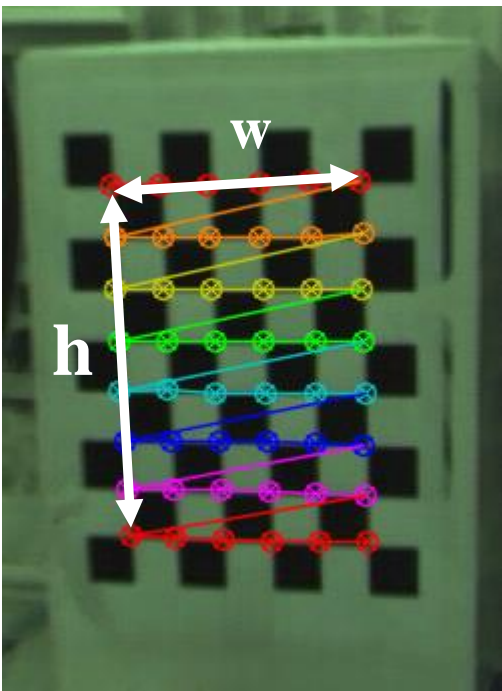
4) Possible improvements

Optionally, students may modify their software system to implement one (or more) of the following improvements.

- Modify the matching algorithm so to deploy a smaller disparity range, such as e.g. [0+o, 64+o], with o being a suitable horizontal offset. This offset can be computed as that value allowing the main disparity, d_{main}, computed at the previous time instant to lie at the centre of the disparity range. Accordingly, as the vehicle gets closer to the obstacle, the horizontal offset increases, thus avoiding the main disparity to exceed the disparity range (and conversely, when the vehicle goes away from the obstacle).

- Instead of just a single main disparity, compute a set of disparities associated with the different areas of the obstacle, so to then estimate the distance from the camera for each part of the obstacle. For example, the image may be divided into a few large vertical stripes (5 in the exemplar left picture below), assuming the vertical lines parallel to the obstacle to be parallel to the image plane of the stereo sensor, and then estimate for each stripe a main disparity value (as done previously). Accordingly, one may create a planar view of the obstacle computing the angle between the horizontal lines parallel to the obstacle and the image plane of the stereo sensor (see the exemplar right picture below).
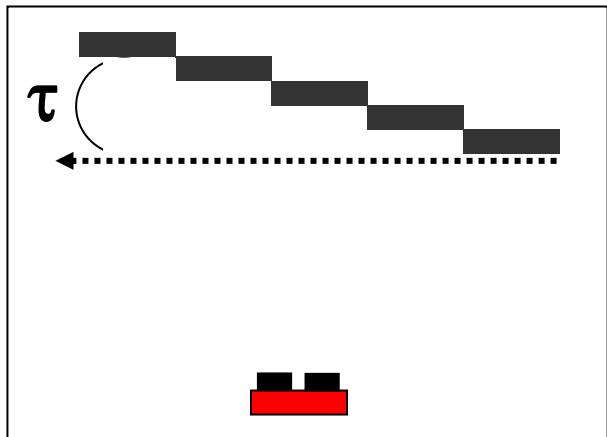
- Develop a more robust approach to estimate the main disparity (disparities) in order to filter out outliers. For example, ambiguous disparity measurements may be detected and removed by analysing the function representing the dissimilarity (similarity) measure along the disparity range or by computing disparities only at sufficiently textured image points (e.g, as highlighted by the well-known Moravec interest operator).