Alma Mater Studiorum
University of Bologna

Stereo Robot Navigation

Course: Image Processing and Computer Vision

Presented by Priti Kumari Gupta

Contents

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

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 The main task of the project requires the following steps: [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 (with respect to 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:

$$z(\text{mm}) = \frac{b(\text{mm}) \cdot f(\text{pixel})}{\text{d}_{\text{main}}(\text{pixel})}$$

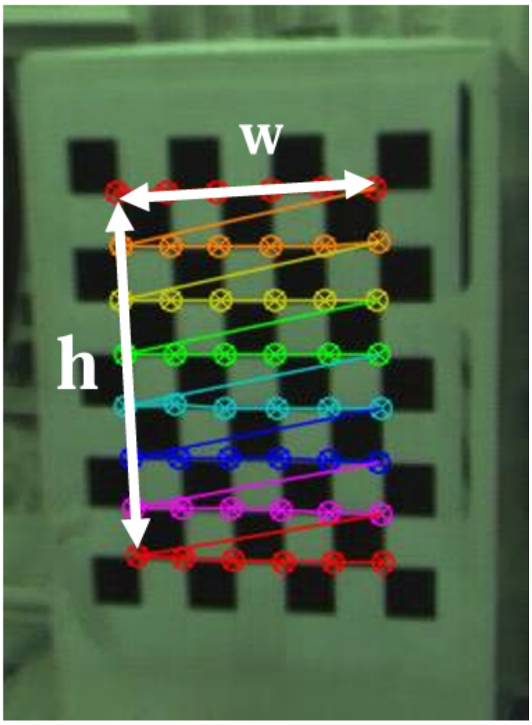
4. Generate a suitable output to convey to the user, in each pair of frame, the information related to the distance (converted in meters) from the camera to the obstacle. Moreover, an alarm should be generated whenever the distance turns out below 0.8 meters.
5. Compute the real dimensions in mm (W,H) of the chessboard pattern present in the scene. 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:

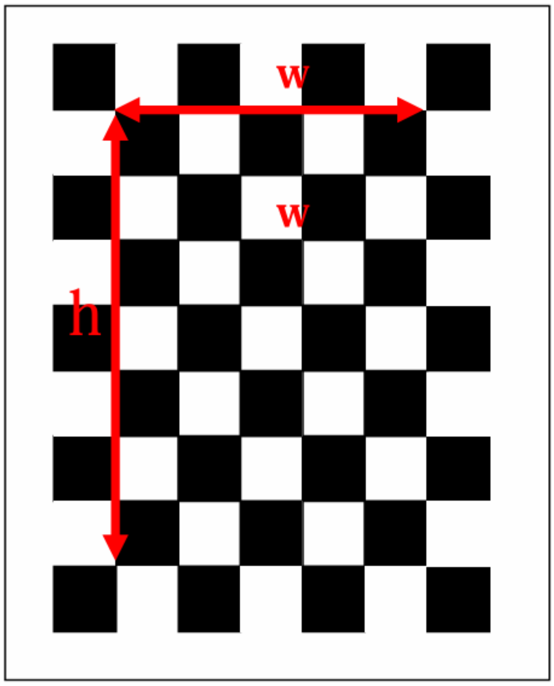
$$W(\text{mm}) = \frac{z(\text{mm}) \cdot w(\text{pixel})}{f(\text{pixel})}$$

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

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.



(a) OpenCV function cvDrawChessboardCorners



(b) Real W and H dimensions of the chessboard

4 Environment and Libraries

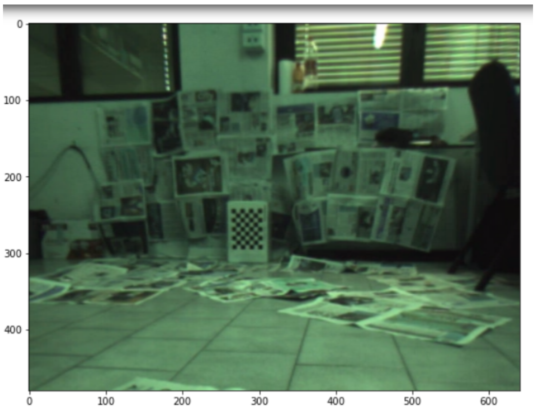
The project has been developed using jupyter Notebook. The python interpreter used is python 3.10.The libraries used for this project are following:

- opencv version 4.9.0
- matplotlib version 3.4.3
- numpy version 1.20.3

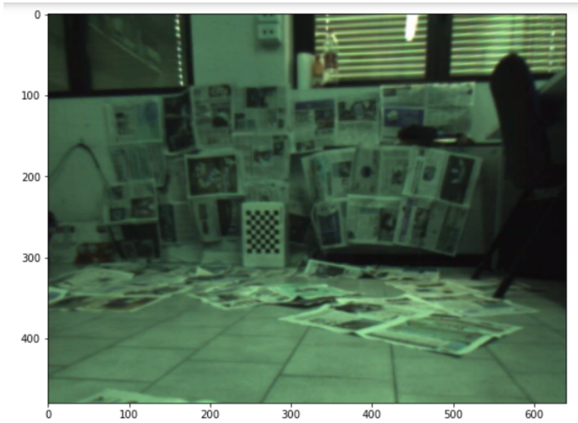
5 Implementation

5.1 PreProcessing

After importing all the required libraries, two video streams, `robotR.avi` and `robotL.avi`, captured by the right and left cameras respectively, were loaded using OpenCV's `VideoCapture`. The code ensures that both video streams are successfully opened. Additionally, the program reads frames from the right (`video_R`) and left (`video_L`) stereo video streams using the `read()` function. The returned values, `retR` and `retL`, represent the success of frame retrieval, while `frameR` and `frameL` contain the actual frames. In the process of calculating the disparity, the initial step involves converting the color frames to grayscale using OpenCV's cvtColor function.In the context of computing a disparity Grayscale images focus on intensity variations rather than color differences, leading to more accurate depth estimates.



(a) FrameL



(b) FrameR

5.2 First Task: Compute Disparity Map

After the initial preprocessing steps, including loading stereo video frames and converting them to grayscale, the first task implemented is to compute the disparity map. The StereoSGBM algorithm, provided by OpenCV, is utilized for this purpose. The algorithm is initialized with specific parameters using the `StereoSGBM_create` function, and subsequently, the `compute` function is employed to calculate the disparity map based on the grayscale images from the left and right cameras.Parameters used to improve the disparity map are following:

- minDisparity: Minimum possible disparity value.In our case it is set to '0'.
- numDisparities: Maximum disparity minus minimum disparity. In our case maximum disparity is set to '128'.
- blockSize: The size of the block used for matching. In our case it is set to be 15.
- P1,P2: parameter controlling the disparity smoothness.
- speckleWindowSize:150
- speckleRange:32

5.3 Second Task: Estimate a main disparity

After computing the disparity map, to focus our anlysis on a specific region within the disparity map, a window frame of 100x100 pixels was extracted from the central area of the reference frame. Later the mean or average disparity are calculated to obtain the distance of the obstacle wrt to the moving vehicle based on the main disparities.

5.4 Third Task: Determine the distance of the obstacle

With given focale = 567.2 pixel and baseline = 92.226 mm it is possible to determine the distance(in meter) of the obstacle wrt to the moving vehicle based on the main disparities (in pixel) which we calculated in the task 2 by applying the formula.

$$z(\text{mm}) = \frac{b(\text{mm}) \cdot f(\text{pixel})}{\text{d}_{\text{main}}(\text{pixel})}$$

where z is denoted as distance.

5.5 Fourth Task: Alarm generated whenever the distance turns out below 0.8 meters.

The distance calculated from the above-mentioned formula is converted to meters from millimeters and then the distance in meters is displayed on the frame using OpenCV's cv2.putText function. The text is positioned at coordinates (10,50) with a font scale of 0.5, and colored green (RGB: (0, 255, 0)).

Figure 3: Distance showing in the video

Additionally, an alarm system is implemented based on the calculated distance. If the calculated distance is less than or equal to 0.8 meters, the Alarm flag is set to True, indicating a potential collision scenario. In such cases, the text "Alarm" is displayed prominently on the frame at coordinates (320,210).

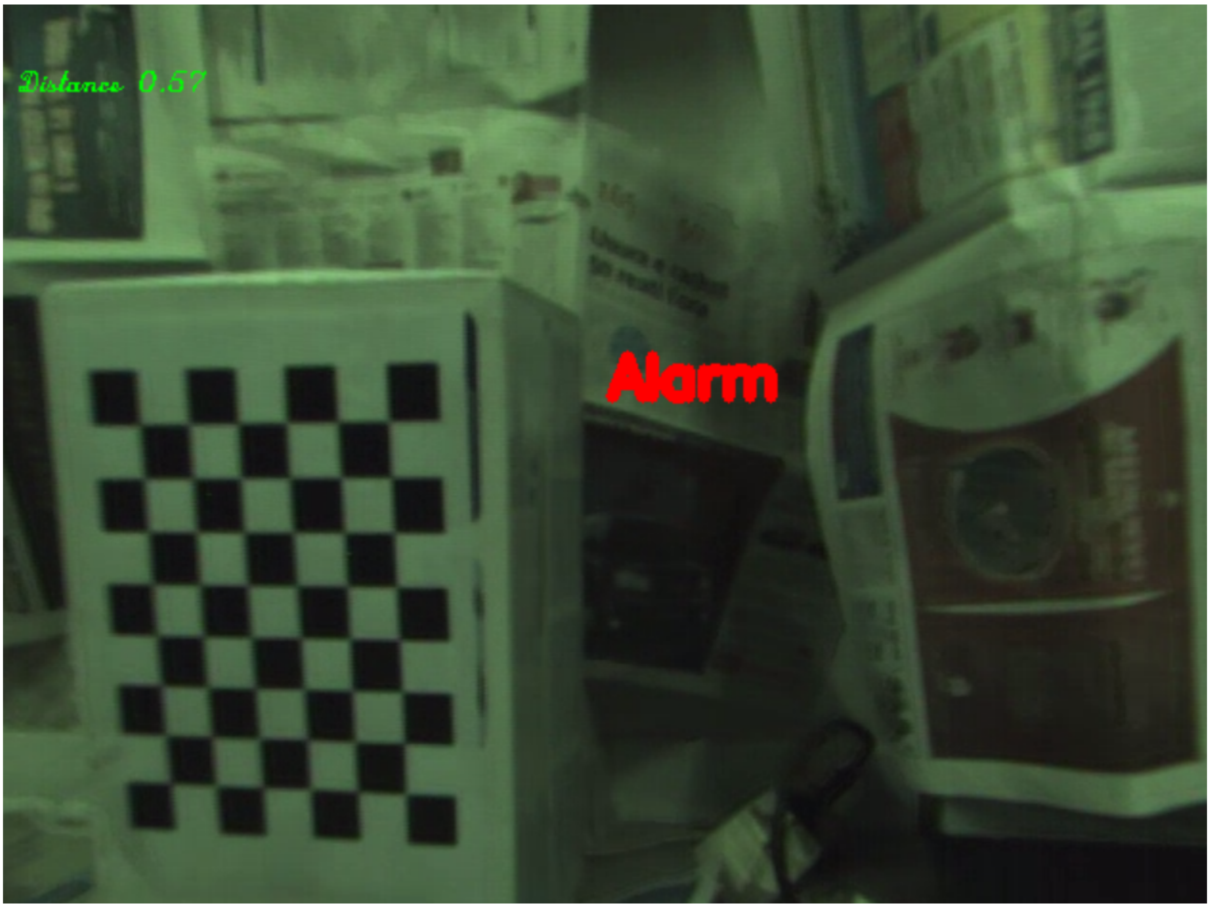


Figure 4: Alarm showing in the video

5.6 Fifth Task:Compute the real dimensions of the chessboard pattern present in the scene.

To accurately determine the dimensions of the chessboard pattern in the image, the OpenCV functions cvFindChessboardCorners and cvDrawChessboardCorners is deployed to 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 is obtained from their pixel dimensions (w,h) by the following formulas:

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

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

After detecting the the corners of a chessboard pattern in the grayscale image, the variable found is used to check whether the corners were successfully located Additionally, the cornerSubPix function is employed to refine the corner locations for increased accuracy.

```
#Corners Detector
pattern_size=(8,6)


found,corners = cv2.findChessboardCorners(GrayImageL,pattern_size)

if found:
    criteria = (cv2.TERM_CRITERIA_EPS + cv2.TERM_CRITERIA_MAX_ITER, 30, 1)
    cv2.cornerSubPix(GrayImageL, corners, (5,5), (-1,-1), criteria)
```

Figure 5: code snippet.

As the given chessboard pattern is (8,6), the corners of the chessboard pattern are identified based on their position within the corner array which is essential for calculating the width and height of the chessboard pattern. Later Using the calculated distances and the known focal length (focale) of the camera, the actual height (H) and width (W) of the chessboard pattern are estimated in millimeters. Finally, the detected chessboard corners are visualized on the grayscale image using the OpenCV function drawChessboardCorners.

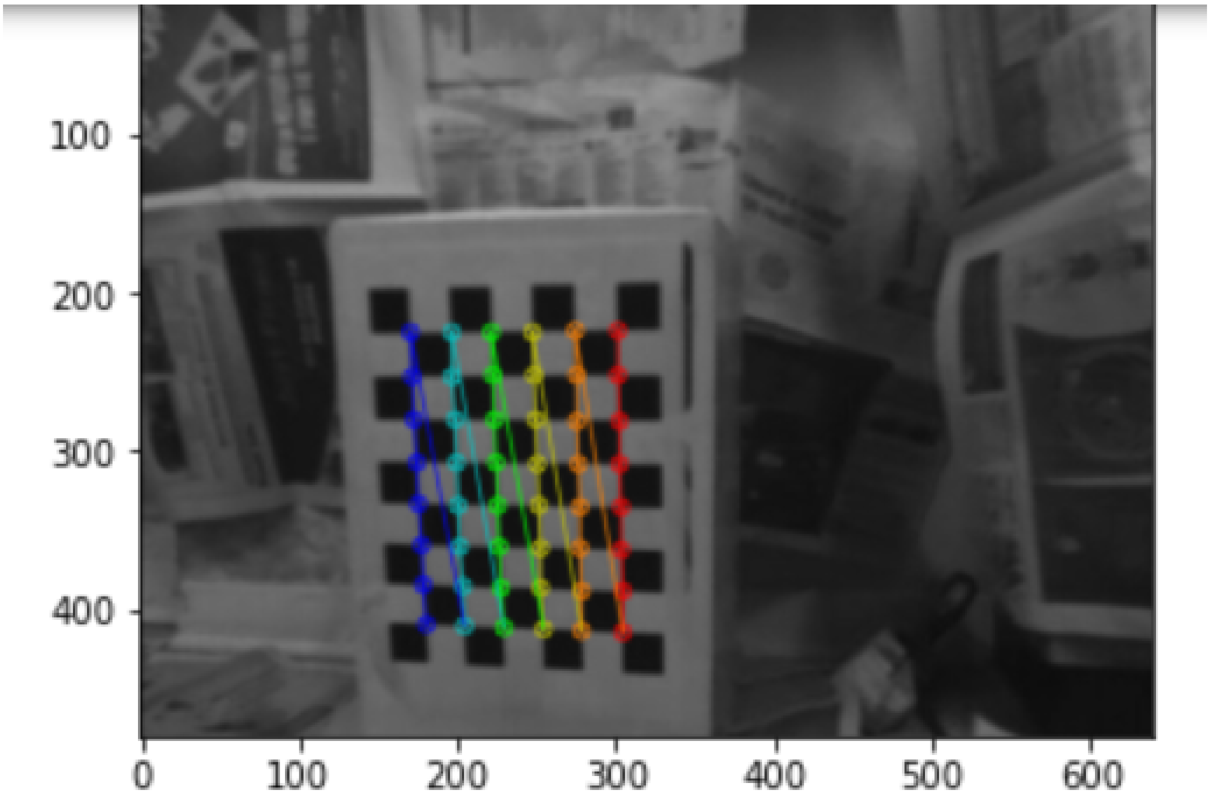
```
top_right = corners[0,0,:]
bottom_right = corners[7,0,:]
top_left = corners[40,0,:]
bottom_left = corners[47,0,:]

h_mm = np.linalg.norm(bottom_left-top_left)
w_mm = np.linalg.norm(top_left-top_right)

H=(distance*h_mm)/focale
W=(distance*w_mm)/focale

draw = cv2.cvtColor(GrayImageL,cv2.COLOR_GRAY2BGR)
cv2.drawChessboardCorners(draw, pattern_size, corners, found)
plt.imshow(cv2.cvtColor(draw,cv2.COLOR_BGR2RGB))
plt.show()
```

Figure 6: code snippet.




```
W - estimated: 127.14mm   real: 125mm
H - estimated: 178.66mm   real: 178mm
Main disparity: 96.98
distance_in_m: 0.5394125327928236
```

Figure 7: drawChessboardCorners.

6 RESULT

6.1 Comparision between Distance and Disparity

The relationship between disparity (d), depth (Z), focal length (f), and baseline (b) can be described by the following equation.

$$Z = \frac{d}{f \cdot b}$$

This means disparity and distance are inversely proportional: smaller disparities correspond to larger distances, and larger disparities correspond to smaller ones.

```
distance_in_m: 2.513042753439248
Main disparity: 20.89
distance_in_m: 2.5037418663359725
Main disparity: 21.09
distance_in_m: 2.480035796631122
Main disparity: 21.15
distance_in_m: 2.4734381043657665
Main disparity: 21.39
distance_in_m: 2.44507770676017
Main disparity: 21.63
distance_in_m: 2.4186615785515304
Main disparity: 21.91
distance_in_m: 2.387412207372319
Main disparity: 22.11
distance_in_m: 2.365948360613636
Main disparity: 22.23
```

```
Main disparity: 38.46
distance_in_m: 1.360048091556447
Main disparity: 38.93
distance_in_m: 1.3437926285379087
Main disparity: 39.18
distance_in_m: 1.3351556052485516
Main disparity: 39.65
distance_in_m: 1.319278117320326
Main disparity: 40.95
distance_in_m: 1.277283724525924
Main disparity: 42.51
distance_in_m: 1.2306610498694215
Main disparity: 42.92
distance_in_m: 1.2186928810766804
Main disparity: 42.58
distance in m: 1.2284455360313156
```

Figure 8: The figure reports the values of the disparity and the distance (depth) for the first approach manoeuvre.

```
Main disparity: 53.75
distance_in_m: 0.9732882661969566
Main disparity: 51.47
distance_in_m: 1.0163390282668987
Main disparity: 49.70
distance_in_m: 1.0524290663469067
Main disparity: 49.85
distance_in_m: 1.0493792887219082
Main disparity: 48.91
distance_in_m: 1.0694876571686343
Main disparity: 47.50
distance_in_m: 1.1013695427166355
Main disparity: 46.88
distance_in_m: 1.1158904493722839
Main disparity: 47.36
distance_in_m: 1.1044903093227656
Main disparity: 46.06
distance_in_m: 1.1356046550880081
Main disparity: 45.27
```

```
distance_in_m: 1.955956875793225
Main disparity: 26.60
distance_in_m: 1.966830513569839
Main disparity: 26.45
distance_in_m: 1.977573014223221
Main disparity: 26.45
distance_in_m: 1.9774193083205753
Main disparity: 26.48
distance_in_m: 1.9756171156422888
Main disparity: 26.15
distance_in_m: 2.000549789630362
Main disparity: 26.11
distance_in_m: 2.003193703994641
Main disparity: 26.13
distance_in_m: 2.001716289773478
Main disparity: 26.36
distance_in_m: 1.984555955431603
Main disparity: 26.49
```

Figure 9: The figure reports the values of the disparity and the distance (depth) for the second approach manoeuvre.

From the above few results for the first approach, we can observe that as the disparity decreases (from 42 to 20), the distance increases. This inverse relationship is consistent with the principle of stereo vision, where smaller disparities correspond to larger distances and vice versa. While there is an overall trend of decreasing distance with decreasing disparity, the relationship is not strictly linear. This is because the disparity-to-distance relationship is influenced by factors such as the focal length and baseline of the stereo camera setup, as well as the geometry of the scene being captured. For Example, In the case of distance 1.3600, the disparity is 38.93 whereas for the distance 1.343, the main disparity is 39.18.

6.2 Chessboard Dimension

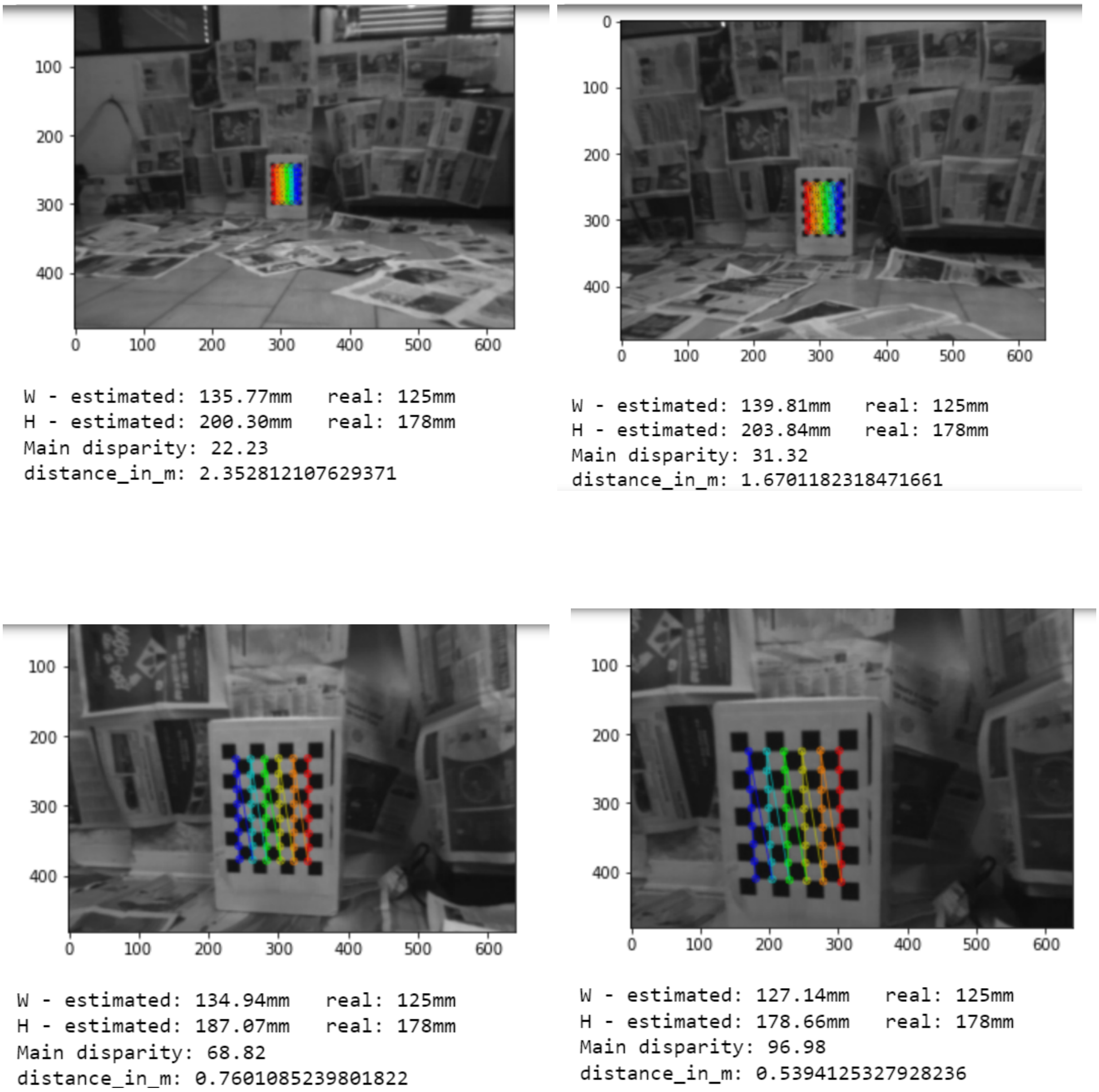


Figure 10: The figure reports the dimension of the chessboard related to distance.

From the above result, we can conclude that The estimated dimensions (W - width, H - height) were larger and inconsistent than the real dimensions at the beginning of the approach where the distance between the vehicle and the chessboard is greater. As the vehicle got closer to the pattern the difference between the estimated and real dimensions decreased. The reason could be the use of a window size of 100x100 pixels, centered on the target (chessboard), implies that at greater distances, the window might include not only the chessboard but also surrounding scene elements, such as the wall. This inclusion of background elements can lead to inaccuracies in the disparity estimation. on the other side, as the vehicle approaches the chessboard, the window gradually captures fewer background scene points and more of the chessboard itself. Consequently, the algorithm's performance improves as it focuses more on the relevant features of the chessboard. This increases in accuracy as the distance decreases. The second reason could be possible for inconsistent values that we might not have properly tuned some parameters that belong to the disparity function.

In the first approach, the vehicle goes closer to the chessboard resulting in increasing accuracy whereas the second approach may have introduced more uncertainty in estimation. Visual cues become less clear as distance increases, leading to less accurate estimations. Figure 10 illustrates the comparison between the estimated dimensions and the actual dimensions of the chessboard. Each set of measurements consists of the width (W) and height (H) of the chessboard, both estimated and real, along with the main disparity and the calculated distance in meters from the stereo vision system. when the vehicle is at a distance of 2.3meters then the estimated dimensions of the chessboard are 135(width) and 200(height) which is much greater than the actual dimension of the chessboard whereas when the vehicle is at a distance of 0.5m than the estimated and actual dimension of the chessboard is almost equal.