Feasibility of Basic Visual Navigation for Small Robotic Sailboats

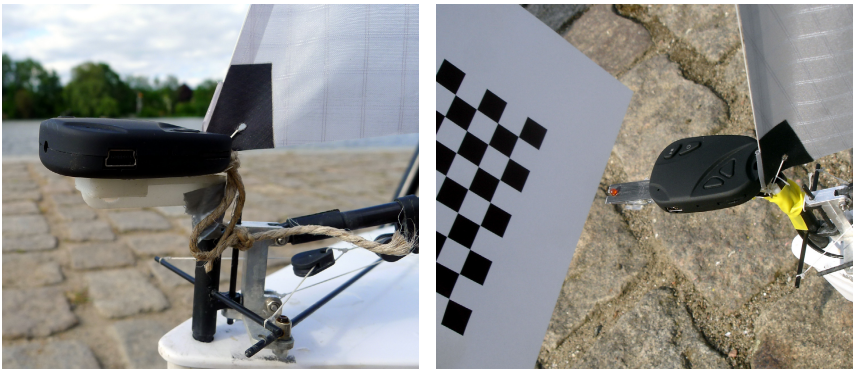
Tobias Neumann and Alexander Schlaefer

Abstract. Image based navigation is a key research focus for many robotic applications. One complication for small sailing robots is their limited buoyancy and rather rapid motion. We studied whether it would still be feasible to use video data for basic navigation in an inshore race course scenario. Particularly, we considered methods for detecting the horizon and buoys, as well as estimating rotations via optical flow. All methods have been tested on a set of manually annotated scenes representing different sailing and lighting conditions. The results show that detection rates of more than 80 % for the horizon and more than 94 % for buoys can be achieved. Moreover, a comparison of the average optical flow with compass data indicates that rotations of the boat can be estimated. Hence, the methods should be considered in addition to other sensors.

1 Introduction

Keeping a constant lookout at all times is part of the traditional navigation approach, e.g., to avoid collisions. While communication based systems like the Automatic Identification System (AIS) are potentially simpler and more robust, vision based methods may be more versatile and a number of approaches have been proposed [3, 7, 4]. Building a general purpose system to understand images in maritime environments is a challenging task, and we consider a more limited scenario with small sailing robots [8]. Our boats have limited buoyancy and allow for only very small and lightweight cameras to be installed. They are also moving substantially, even in very small waves. While we typically use an onshore server to communicate

Alexander Schlaefer · Tobias Neumann
Institute for Robotics and Cognitive Systems,
University of Luebeck,
Ratzeburger Allee 160, D-23562 Luebeck
e-mail: schlaefer@rob.uni-luebeck.de,
neumann@informatik.uni-luebeck.de



(a) (b)

Fig. 1 A small 15 g GUNCAM mounted to the bow of an rrMM boat (a). Before sailing, the camera is calibrated (b).

position data [1], one advantage of a vision based approach could be a better ability to detect near field collisions and to augment sensor data with information obtained from images. We present methods for detection of horizon and buoys and we propose to augment sensor data, e.g., from compass, accelerometer, or gyroscope, with motion information obtained from the images.

2 Material and Methods

We obtained test data from a small camera (GUNCAM, www.guncam.de) with a weight of only 15 g mounted to the bow of a robotic racing Micro Magic (rrMM) sailing robot (Fig. 1). The camera has an image resolution of 720 x 480 pixels and a frame rate of 30 fps. All video data is directly recorded to a microSD card, which in our setup can store approximately 1 h of data.

So far, three different objectives have been considered: the detection of the horizon, the detection of buoys, and the estimation of the optical flow.

2.1 Horizon Detection

Finding the horizon is a typical task in maritime scene analysis. We consider an inshore setting where the horizon is typically defined by the shoreline. While this avoids sea and sky being indistinguishable, new issues arise, e.g., due to reflections. Initially, we used the following steps to find candidate lines in the images. First, the images are filtered with a 9x9 Gaussian kernel and a Canny edge detector [2] was applied. Second, a Hough transform was used to detect lines in the preprocessed images. The resulting set of lines C is ordered by descending length, i.e., the first line returned by the algorithm contains the largest number of edge pixels. We call C the

candidate set. Obviously, if the horizon goes through the whole images, it is likely to be among the long lines. However, given possible reflections and small waves, other lines can actually be longer. To reduce the likelihood of misclassification, we considered the following additional assumptions:

1. The horizon remains relatively stable over time

2. The line representing the horizon is similar in two consecutive frames

3. The horizon runs typically close to the image center for our camera setup

Each line in C is represented by a point P and an angle m between the line and the x-axes. Consider two lines l_1 and l_2 represented by P_1, m_1 and P_2, m_2. Then we define the dissimilarity h between the two lines as

$$h(l_1, l_2) = \alpha \|P_1 - P_2\| + |m_1 - m_2| \tag{1}$$

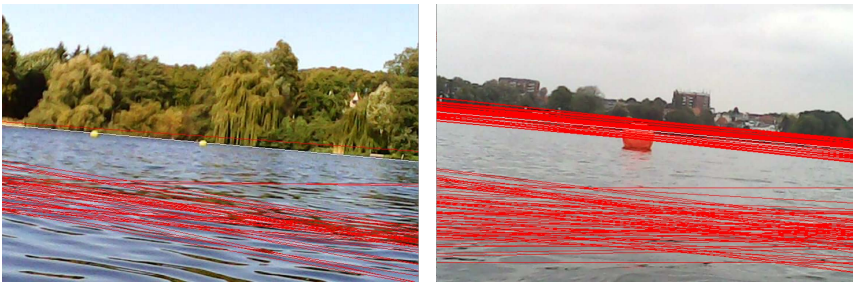
where α is typically 0.5 and denotes the relative importance of a difference in the points and in the angle. Note that the points returned by the method are unique for the same line.

We now consider k consecutive video frames such that C_k is the set of candidate lines for the current frame. Moreover, we restrict $|C_i|$ to be at most n, i.e., only the n longest candidates in C_i are retained. For $c \in C_k$ we sort $\bigcup C_i | i = 1, 2, ...k-1$ according to h, i.e., such that the most similar lines come first. We then compute the mean dissimilarity for c and the first m items of the resulting set. The candidate c from the current frame with minimal mean dissimilarity is considered the horizon, as similar long lines have frequently occurred in the previous frames. To account for the observation that the horizon is typically a line close to the image center, we additionally consider the distance to center when evaluating the candidates.

In case no line is detected in the current video frame or if the dissimilarity of all candidates exceeds a threshold, the horizon line from the last frame is added to the candidate set and returned. Figure 2 illustrates the assumptions underlying the approach, i.e., that the horizon is typically close to the image center and more stable than, e.g., lines induced by waves.

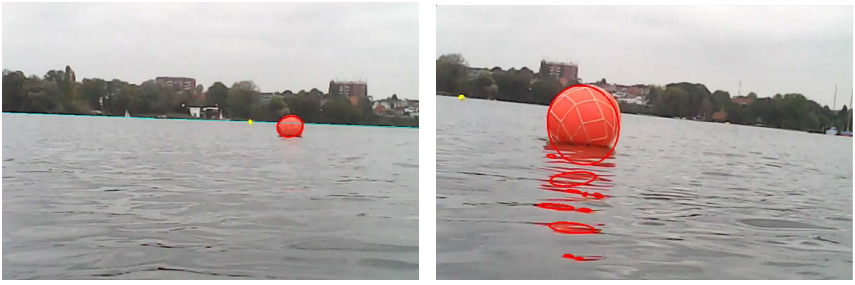
2.2 Buoy Detection

Typically, buoys have a distinctive shape and color to be easily detectable. Our main purpose is the detection of marks when following a race course. Hence, we can assume that the colors are exceptionally bright, e.g., yellow, orange, or red. To use color for object detection, we first convert the images to the hue-saturation-value (HSV) color space. We then applied a threshold and a morphological closing operation for cleaning contours and used OpenCV to label each object in the resulting image. Subsequently a minimally enclosing ellipse was determined for each potential object, which in case of a buoy allows to estimate the center point and the radius.



(a) (b)

Fig. 2 Typically the horizon is close to the image center (a). The image in (b) shows a set of candidates from 30 consecutive frames, illustrating that the actual horizon is fairly stable.



(a) (b)

Fig. 3 An example for detecting a red and yellow buoy (a). Reflections can substantially complicate the detection (b). Note that bright sunshine and calm would lead to even more reflections.

Figure 3 illustrates the method and also highlights that reflections can cause substantial problems. Again, we can argue that reflections due to waves change over time. Using an approach similar to that used for the horizon, we consider a number of consecutive frames to identify stable objects. All objects are represented by the center point and radius of the respective ellipse. Clearly, as the boat can be very close to the buoy, the change in its position can be rapid. Hence we use a dissimilarity function that weights the position with 0.1 and the radius with 0.9.

2.3 Rotation Detection

To detect rotations of the boat we computed the optical flow using the Lucas-Kanade method [6]. First, we used the horizon to partition the image. We only considered

features above the horizon in order to avoid artifacts from waves. For every two consecutive frames the mean flow over all features detected in both images was computed and stored.

2.4 Incorporating Sensor Data

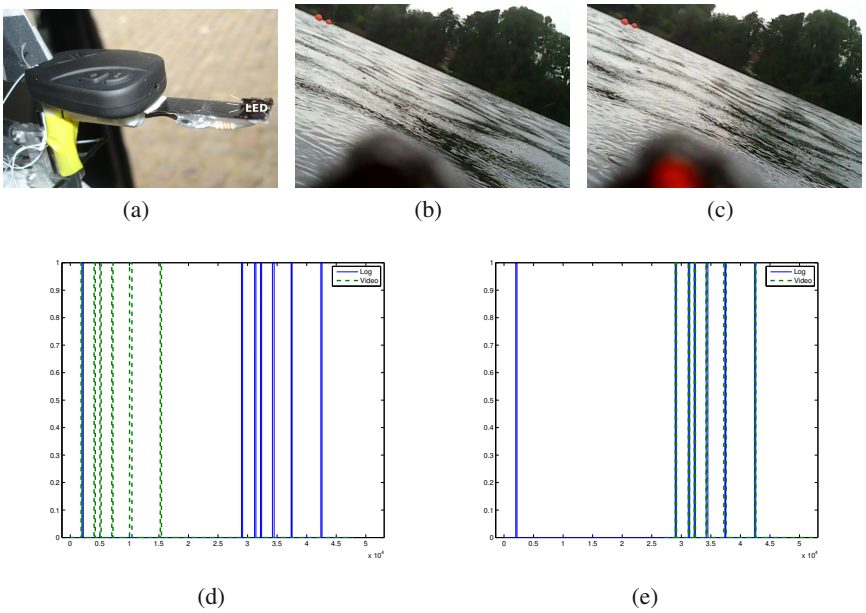
Obviously, it is interesting to study how information from image data and the sensor data available onboard can be considered jointly, e.g., to compute the expected position of horizon and buoys in subsequent image frames or to improve navigation by estimating the boat's motion from the image data. Our current design is limited in that the onboard computing power is not sufficient for video processing or storage. While we are working to integrate more powerful hardware, our current experimental setup was based on independent recordings with the GUNCAM. To synchronize camera and sensor data, a small LED was mounted in front of the camera. It was switched on via the onboard control unit following a Fibonacci sequence and its status was recorded with the log-data. During offline processing, the state of the LED was automatically detected in the video data, and the frames were labeled accordingly. Video and sensor data were then combined by correlating the state of the LED in the two data sets. Figure 4 illustrates the setup, how a certain subset of pixels was used to detect the state of the LED, and how sensor and image data were correlated.

2.5 Data

We collected 6 h of video data on 11 different days and in different weather and light conditions. From this data set we extracted five scenarios of 11 s to 16 s length representing different conditions (Fig. 5). The actual weather conditions and the lengths of the video sequences are summarized in Table 1. Each of the video frames was manually annotated, i.e., from the horizon or buoy candidates the correct one was identified and stored. This data was then used to run tests quantifying the detection rate for horizon, buoys and rotation.

Table 1 Summary of length and wind and weather conditions for the five test scenes.

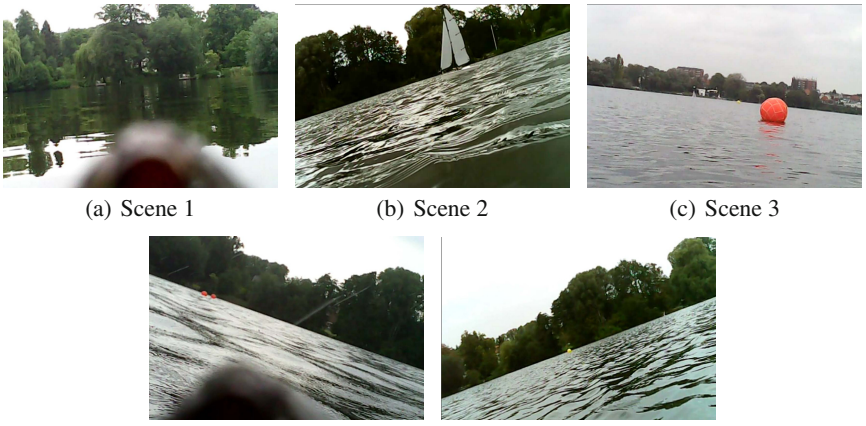
scene	weather	wind (kn)	length (s)	total number of frames
1	sunny	2-3	16	480
2	cloudy	6-12	14	440
3	cloudy	4-8	13	394
4	rainy	8-12	11	341
5	cloudy	6-10	13	409



(a) (b) (c)

(d) (e)

Fig. 4 The camera with LED (a), examples for images with LED switched off (b) and on (c), and the log and video data before and after correlation, respectively (d,e). For the latter two, blue indicates log data while the dotted green lines indicate video data.



(a) Scene 1 (b) Scene 2 (c) Scene 3

(d) Scene 4 (e) Scene 5

Fig. 5 Randomly chosen images from the five scenes we analyzed. Note another rrMM boat in scene 2. More details on the weather conditions are summarized in Table 1.

3 Results

The actual images were processed offline using OpenCV on a computer with an Intel Q6600 2.40 GHz processor and 4 GB memory running a 64 bit Fedora-Linux.

3.1 Horizon Detection

Table 2 summarizes the performance of horizon detection using $k = 16$, $n = 3$, and $m = 5$. All scenes resulted in candidate lines for each frame, except for scene 1, where only 61 % of all frames yielded at least one candidate line. Taking the frames with at least one candidate line as the basis, the detection rate for the horizon ranged from 81 % for scene 4 to 90 % for scene 1. The mean runtime ranged from 24 ms to 46 ms.

Table 2 Results of the horizon detection tests including the total number of frames, the number of frames for which at least one line was detected, the number of correctly identified horizon lines, and the runtime.

scene	total frames	with lines	with correct horizon	runtime (ms)
1	480	293	265	23.95
2	440	440	376	43.75
3	394	394	337	20.60
4	341	341	277	33.05
5	409	409	343	39.49

3.2 Buoy Detection

Only scenes 3, 4, and 5 contained buoys, with scenes 3 and 4 each containing two. Hence, Table 3 summarizes results for five tests with red and yellow buoys in distances ranging from approximately 5 m to more than 100 m. Candidate objects were visible in slightly more than 50 % for scene 4, and in more than 80 % of the frames for scenes 3 and 5. The yellow buoy for scene 3 was detected in 78 % of the frames actually containing it, while all the other buoys were detected in more than 94 % of the frames. Between 36 % and 68 % of the frames resulted in a stable detection of the correct buoy, i.e., the buoy was correctly labeled in subsequent frames. However, for the red buoy in scene 3, an irrelevant object was incorrectly identified as a stable buoy in 6 % of the frames.

Table 3 Results of the buoy detection tests including the scene, the object to detect, the approximate distance, the total number of frames, the number of frames with the object visible, the number of frames where the object was detected as a candidate, the number of frames where the object was correctly / wrongly classified as stable, and the runtime.

scene	buoy	distance (m)	total frames	visible	candidate	stable	wrong	runtime (ms)
3	red	5-10	394	326	326	212	20	45.69
3	yellow	100	394	333	259	126	0	43.25
4	red (1)	50	341	178	175	64	0	44.87
4	red (2)	50	341	199	195	82	0	43.46
5	yellow	100	409	355	333	242	0	43.96

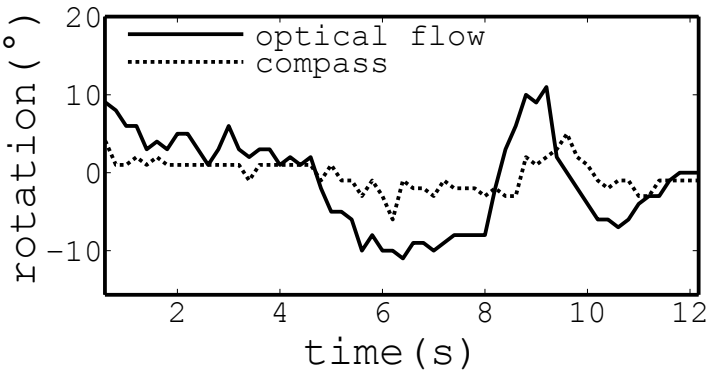


Fig. 6 The plots show the mean optical flow vs. the logged compass data for scene 1.

3.3 Rotation Detection

Figure 6 illustrates the results of the optical flow experiments for scene 1. Clearly, flow and compass values are not the same. However, the trend, i.e., turning port or starboard, is similar and a further calibration may lead to better agreement of the magnitude of change.

4 Discussion

We considered basic visual information for navigating small sailing robots in inshore race scenarios. More than 80 % accuracy in detecting the correct horizon is far from perfect but still illustrates that a fairly simple approach yields good results. Moreover, adapting the preprocessing of the images – particularly the Gaussian filter – to the sailing conditions could further improve the performance. While one could

argue that technically we identify the shoreline, we would hold that for a small boat with the camera mounted a few centimeters above the water, this close to the actual horizon. A more obvious limitation is the dependency on the illumination, which will require more tests in sunny conditions. However, at least in our home area in northern Germany good sailing conditions often coincide with cloudy weather.

Clearly, illumination also effects the buoy detection. Another problem is the correct labeling of the detected objects over multiple frames, which only worked for two out of five scenes. Note that part of this issue is related to the buoy leaving and entering the image, and a better integration with the boats' sensors may be used to compute the expected position of the buoy. For example, in scene 4 the buoys were frequently out of sight, which relates to the poor performance in stable detection. However, an important goal of buoy detection in our scenario is the rounding of marks, and it is promising that the test scene with a close proximity to the mark resulted in a larger number of consecutive frames where the buoy was correctly detected. Further work will consider the actual size of the buoy for this particular scenario.

The proposed methods allow estimating basic information about changes in the robot's pose. It is straightforward to get the approximate heeling angle from the detected horizon. Likewise, the detected rotation can indicate course changes. While the current results are preliminary and require further calibration, a visual compass has been considered before and presents a promising approach [5]. Particularly as the motions of a small sailing robot are characterized by substantial pitching and rolling even in small waves, which can compromise the compass readings. A further possible use of visual information is estimating the distance to buoy, if their size is known. In fact, we consider the methods most useful in the proximity of buoys, where their estimates may be more accurate than, e.g., GPS data. For example, a close rounding of a mark could be based on image data rather than GPS.

Considering the runtime, horizon and buoy detection could be performed at a rate of 20 Hz to 45 Hz. Although the computations were done offline and separately on a rather powerful computer, we believe that further optimization of the code and more powerful microprocessors will allow for sufficiently fast onboard online processing.

5 Conclusion

Rapid motion and limited buoyancy complicate image guided navigation for small robotic boats. However, our results indicate that basic information can be derived, which is sufficiently robust to augment other sensors. The heeling of the boat and rotations can be estimated and the horizon and buoys can be detected. Further work should address the robustness and runtime of the proposed methods.

References

1. Ammann, N., Hartmann, F., Jauer, P., Krüger, J., Meyer, T., Bruder, R., Schlaefer, A.: Global Data Storage for Collision Avoidance in Robotic Sailboat Racing – the World Server Approach. In: Schlaefer, A., Blaurock, O. (eds.) Robotic Sailing, vol. 79, pp. 157–166. Springer, Heidelberg (2011)

2. Canny, J.: A computational approach to edge detection. IEEE Trans. Pattern Anal. Mach. Intell. 8(6), 679–698 (1986)
3. Gal, O.: Automatic Obstacle Detection for USV's Navigation Using Vision Sensors. In: Schlaefer, A., Blaurock, O. (eds.) Robotic Sailing, vol. 79, pp. 127–140. Springer, Heidelberg (2011)
4. Huntsberger, T., Aghazarian, H., Howard, A., Trotz, D.C.: Stereo vision based navigation for autonomous surface vessels. Journal of Field Robotics 28(1), 3–18 (2011)
5. Labrosse, F.: The visual compass: Performance and limitations of an appearance-based method. Journal of Field Robotics 23(10), 913–941 (2006)
6. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proceedings of the 7th International Joint Conference on Artificial Intelligence - Volume 2, IJCAI 1981, pp. 674–679. Morgan Kaufmann Publishers Inc., San Francisco (1981)
7. Ramirez, R., Guo, Y., Ieng, S.H., Plumet, F., Benosman, R., Gas, B.: Omni-directional camera and fuzzy logic path planner for autonomous sailboat navigation. In: 2011 Iberoamerican Conference on Electronics Engineering and Computer Science (CIIECC 2011), pp. 335–346 (2011)
8. Schlaefer, A., Beckmann, D., Heinig, M., Bruder, R.: A New Class for Robotic Sailing: The Robotic Racing Micro Magic. In: Schlaefer, A., Blaurock, O. (eds.) Robotic Sailing, vol. 79, pp. 71–84. Springer, Heidelberg (2011)