Modeling, Identification and Control, Vol. 45, No. 2, 2024, pp. 51–63, ISSN 1890–1328

Stereo Camera-based Free Space Estimation for Docking in Urban Waters

T. Nygård [1] N. Dalhaug [1] R.Mester [2] E.F.Brekke [1] A. Stahl [1]

[1] *Department of Engineering Cybernetics, Norwegian University of Science and Technology, Trondheim, Norway. E-mail: {trym.a.nygard,nicholas.dalhaug,edmund.brekke,annette.stahl}@ntnu.no*

[2] *Department of Computer Science, Norwegian University of Science and Technology, Trondheim, Norway. E-mail: rudolf.mester@ntnu.no*

Abstract

Operating in urban waters with an autonomous vessel can be challenging. The autonomous vessel must be able to react quickly and detect obstacles to avoid collisions and risky maneuvers. Exteroceptive sensors such as LiDAR and RADAR have typically been used with great success in the maritime domain, but the measurements are often too sparse to represent smaller obstacles during docking and other maritime operations. However, other sensor modalities, such as stereo cameras, can provide both appearance and dense depth information. In this paper, we present a stereo camera-based free space estimation method for the maritime domain. The mapping of navigable areas is crucial for path planning and collision avoidance systems. To robustly estimate the free space, we use vertically oriented rectangular segments known as stixels. We utilized both stereo correspondences and a recent image segmentation network trained on a large, generalized dataset to create the stixels. To validate our approach, we analyzed the estimated free space, evaluating both the accuracy and consistency in the estimated depth over time. We demonstrate the approach using a real dataset recorded with a stereo camera mounted on an autonomous ferry and compare its accuracy against measurements from a LiDAR.

Keywords: Maritime autonomy, Free space, Safe navigation, Stereo camera, Docking

1 Introduction

The automotive industry has long employed exteroceptive sensors to map the environment, and this field has matured significantly, largely driven by the increasing focus on autonomous cars. Recently, this area of research has gained interest in the maritime domain. While RADAR and LiDAR have been extensively utilized in maritime applications, stereo cameras, especially for docking purposes, have received relatively little attention.

Docking with an autonomous vessel presents unique challenges and requires reliable environmental information to ensure passenger safety. An autonomous ferry must continuously assess its safe navigation paths and be able to always know where it can move safely without the risk of collision, this is often referred to as free space (Badino et al., 2007). Free space is crucial for path planning, collision avoidance, and localization. Volden et al. (2022) addresses some of these challenges by presenting a stereo camera-based positioning system for docking. Whereas Helgesen et al. (2023) use a camera mounted on an autonomous ferry for collision avoidance. However, to the authors' knowledge, no stereo camera-based free space estimation method has been developed with maritime vessels and docking in mind. The water surface itself introduces new challenges such as reflections and waves, making stereo matching more complicated due to the difficult task of matching corresponding pixels across two images,

doi:10.4173/mic.2024.2.2

which is required to get accurate depth estimates.

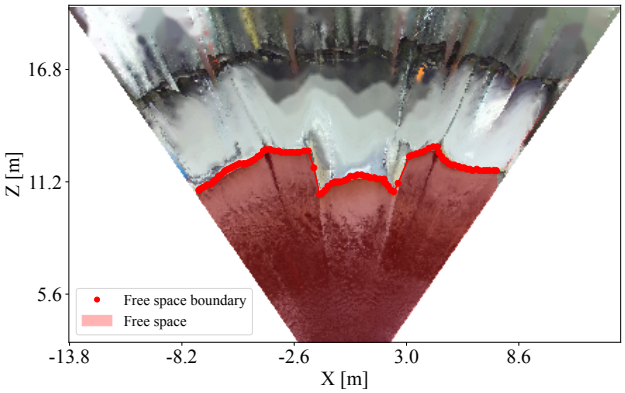
In this paper, we use vertically oriented rectangles known as stixels to robustly estimate the free space boundary, which we represent in a bird's eye view (fig. 1). Stixels were originally developed to aid autonomous cars and have been widely used in the automotive industry (Badino et al., 2009; Pfeiffer and Franke, 2011). Each pixel within a stixel is assumed to have the same depth value. Instead of working with single pixels that are sensitive to noise, we can instead evaluate all the pixels within a stixel to get a more accurate representation of the actual depth. Stixels-based methods have been employed successfully in several automotive applications e.g., they have been extended to handle moving objects (Pfeiffer and Franke, 2010) or to detect hazardous road clutter (Pinggera et al., 2016).

However, due to the difficulty of stereo matching, relying on the estimated depth alone might still be insufficient. Schneider et al. (2016) address this problem, by incorporating semantic segmentation into their stixel approach. In the maritime domain, it is important that the semantic segmentation method can reliably segment and label the water surface, which is non-trivial due to reflections on the water surface.

Instead, we attempt to address these challenges by combining classical techniques with learning-based image segmentation to refine the free space boundary around potential obstacles. We use a recent image segmentation network trained on the large SA-1B dataset used in Kirillov et al. (2023) to provide segmentation masks for obstacles.

The contributions of this paper are threefold: (1) We present a method for estimating navigable free space for docking operations using a short baseline stereo camera. (2) We propose a late fusion approach with image segmentation to reliably estimate the free space in areas where the depth information is insufficient. (3) We analyze and compare the depth measurements from the stereo camera with the ones from the LiDAR using a real dataset collected with a sensor rig mounted on an autonomous ferry.

(a) Image from the current scene showing the dock located close to Brattøra. The dock is built for the autonomous ferry used for the experiment in this paper.



(b) Computed free space for the scene shown in fig. 1a.

Figure 1: Bird's eye view map (b) showing both the free space and the free space boundary obtained with the presented method.

2 Related work

Free space estimation techniques have been studied extensively over the years. 2D occupancy grid maps were proposed in 1989 (Elfes, 1989) and have since been proven successful in a variety of applications for their explicit representation of free space in a probabilistic manner. Badino et al. (2007) uses a stereo camera to create a stochastic occupancy grid and solve a dynamic programming problem to find the optimal boundary of the free space. Soquet et al. (2007) use a stereo camera, but estimate the free space by means of a U and V disparity map, where a histogram over the disparity values is defined for each column and row respectively. The V-disparity image is commonly used to extract the ground surface from its horizontal extent in the disparity image, whereas the U-disparity is used to detect obstacles based on its vertical extent in the disparity image.

Free space estimation techniques have also been studied in the maritime domain. Plenge-Feidenhans'l and Blanke (2021) trained a convolutional neural network (CNN) to divide images into smaller sub-regions that are classified as either water, partial water, or not water. The open-water detection method was later used to estimate the free space in confined waters using LiDAR, RADAR and camera (Plenge-Feidenhans'l, 2023). Yao et al. (2015) use a monocular camera and formulate the free space estimation problem as a 1D Markov random field (MRF) inference problem. Then they solve the inference problem with dynamic programming using only appearance information, edges,

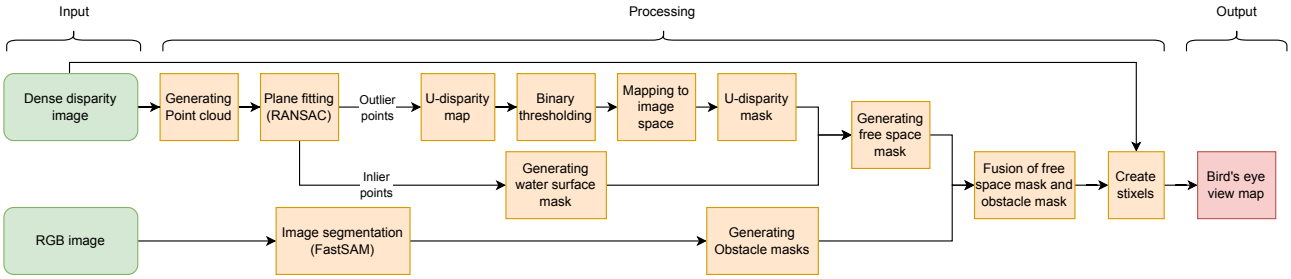


Figure 2: Block diagram of all components involved in the proposed free space estimation method. The final output of the method is a bird’s eye view representation of the environment.

and temporal smoothness constraints.

Water surface segmentation is closely related to free space estimation. Ideally, if one manages to accurately segment out the water surface, one will have a visual clue about which areas are navigable and not. Bovcon and Kristan (2022) train a CNN using an Inertial Measurement Unit (IMU) as a prior on the horizon to reliably segment out the water. However, there is no guarantee that areas that appear behind obstacles are safe to navigate, despite being classified as water. In this paper, we instead define the free space, as navigable area within the first obstacles in sight, that protrude from the water surface.

To detect the obstacles protruding from the water surface, we must first identify the water surface. The assumption that the water surface can be well represented as a plane is a common assumption. Muhovič et al. (2020), propose a plane fitting algorithm to track obstacles with an unmanned surface vessel. They use a modified RANSAC approach, utilizing an IMU and image segmentation to robustly fit a plane to the water surface. Griesser et al. (2023) fit a plane to the water surface and use the plane parameters to estimate roll and pitch from the water surface.

In this paper, we show that we can reliably estimate the free space by combining classical techniques such as plane fitting, stixels and depth estimates from a stereo camera with a refinement step leveraging a recent image segmentation network.

3 Method overview

In this paper, we only consider the first obstacles in the line of sight, meaning that the location of the obstacles will explicitly define the boundary of the free space. With the assumption that most man-made objects can be represented as a set of planar surfaces, we use stixels to represent the obstacles protruding from the water surface. The free space estimation method consists of two parts, one classical approach (sec. 3.1) for finding the free space mask, and a refinement step leveraging a segmentation network to obtain object masks (sec. 3.2). A block diagram with a full overview of all components that are involved in the free space method can be seen in fig. 2.

The proposed free space estimation method relies on the ZED SDK from Stereolabs to obtain reliable, dense, and smooth disparity images (fig. 3), but any dense stereo matching method will serve the purpose. We also used rectified images with the provided intrinsic and extrinsic camera parameters for 3D reconstruction.

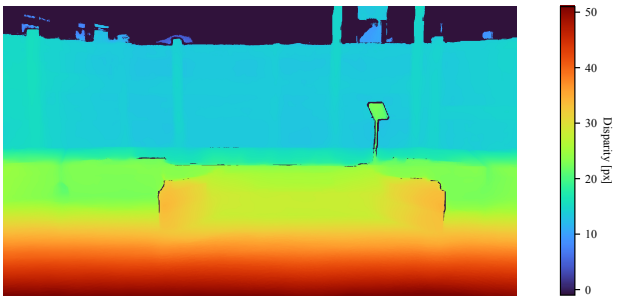
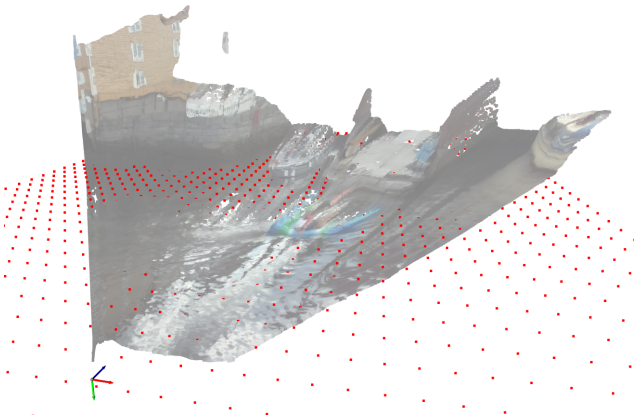


Figure 3: Dense disparity image acquired with the ZED SDK taken from the same video sequence seen in fig. 1a. Assuming that the images are rectified, each disparity value in the image represents the horizontal displacement between two corresponding pixels in an image pair.

3.1 Estimating free space

To estimate the free space that defines the navigable area, we must know where the water surface is located. From the dense disparity images and camera intrinsic and extrinsic parameters, we can reconstruct the 3D points to obtain a point-cloud representation of the scene. The surface of the water was obtained by fitting a plane by solving a regression problem (fig. 4), using Random Sample Consensus (RANSAC) (Fischler

(a) Image of the current scene. In the background we can see two moving obstacles (speedboat and kayak).



(b) Point cloud of the scene seen in fig. 4a plotted with the water surface plane (red points).

Figure 4: The water surface plane was obtained using plane parameters obtained with RANSAC.

and Bolles, 1987). For implementation, we used the RANSAC regressor in sklearn to obtain the plane parameters (Pedregosa et al., 2011). To ensure a good fit, the point cloud was cropped using the horizon found in eq. 6. An initial guess for the cropping threshold is used to fit the plane in the first iteration, whereas the new estimate is used for the following iterations.

After plane fitting, we are left with two sets of points, inliers and outliers. Inlier points will ideally lie on the water surface, whereas the outlier points will mainly represent the obstacles protruding from the water's surface. We assume that the plane complies with the water surface, and project both sets of points back onto the image plane. The points belonging to the water surface will later be used as a water surface mask I_{ws}, while the obstacle points are used to create a U-disparity map (fig. 7b). The U-disparity map can be seen as an obstacle map of objects with a significant vertical extent, where we for each vertical column in the disparity image, compute a histogram with a bin for each disparity value. More specifically, we count the number of times the same disparity value appears along each column in the disparity image (fig. 5).

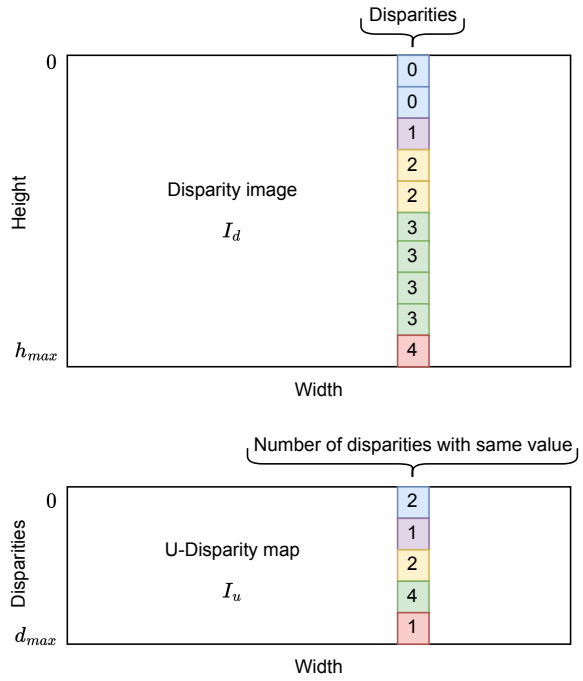


Figure 5: Computing U-disparity values for one column in the disparity image. Each colored block represents a pixel, whereas its color illustrates its connection to the corresponding pixel in the U-disparity map.

More formally, we can for one selected image column c define the U-disparity $I_u(d, c)$ by iterating through each row r up to image height h, to count the number of occurrences of a disparity d by the means of a Kronecker delta function $\delta(I_d(r, c), d)$:

$$I_u(d, c) := \sum_{r=0}^{h} \delta(I_d(r, c), d), \tag{1}$$

with

$$\delta(I_d(r, c), d) := \begin{cases} 1 & I_d(r, c) = d \\ 0 & \text{otherwise.} \end{cases} \tag{2}$$

The Kronecker delta function is set to one if the disparity value d exists in the disparity image I_d at the current row index. To filter out values of an insignificant size and to convert the U-disparity map I_u to a binary map, we perform binary thresholding with a threshold value T:

$$I_{bin}(d, c) := \begin{cases} 1 & I_u(d, c) > T \\ 0 & \text{otherwise.} \end{cases} \tag{3}$$

The threshold value T is empirically found and can be adjusted depending on the scenario. A low threshold value makes it more sensitive to waves and other disturbances in the water surface, but makes it more likely to detect smaller obstacles.

Before creating a binary obstacle mask from I_{bin}, we must map $I_{bin}(d, c)$ to the image space $I'_{bin}(r, c)$ (fig. 7c). If we project the fitted water surface plane onto the image plane (a-b in fig. 6), and plot one column as a function of its disparity values (c in fig. 6), the water surface will appear as a line. With the assumption that we only consider obstacles that protrude from the water surface, the mapping is a matter of finding the row r for the corresponding disparity d of an obstacle.

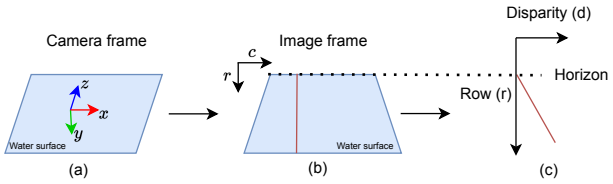


Figure 6: Using the plane parameters in the camera frame to map the U-disparity values to image space. The red line illustrates one column in the disparity image plotted as a function of the disparity and represents the water surface profile.

We use the plane parameters to compute the slope and horizon of this line. For a set of 3D coordinates x, y, z and plane parameters α, β, γ we can define a plane:

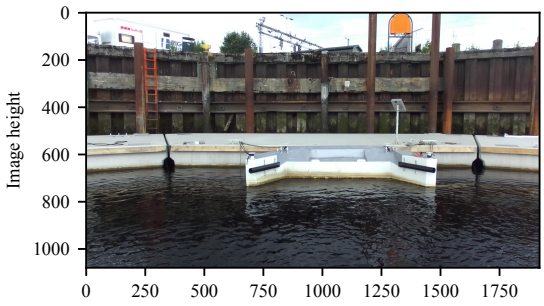
$$\alpha x + \beta y + \gamma = z. \tag{4}$$

This plane is defined in the camera coordinate frame, whereas the U-disparity map is defined in the image coordinate frame. Knowing the intrinsic and extrinsic parameters of the stereo camera, we can use the projection formulas to express the coordinates in the camera frame as the coordinates in the image frame. With baseline b, focal length f and principal point (c_x, c_y) we can use the projection formulas for computing a 3D scene point (x, y, z) from an image point (c, r, d) (Badino et al., 2007).

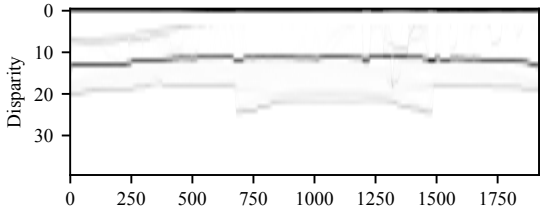
$$x = \frac{(c - c_x)b}{d}, \quad y = \frac{(r - c_y)b}{d}, \quad z = \frac{fb}{d}. \tag{5}$$

Substituting x, y, and z with the equations in eq. 5 we can formulate an expression for the line describing the water surface profile as a function of disparity d:

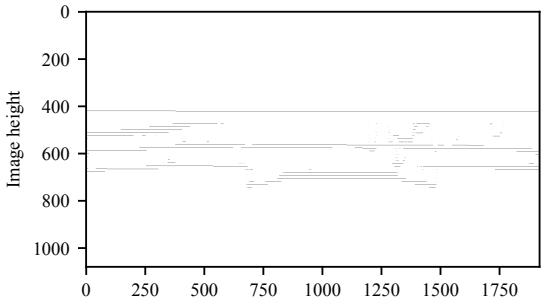
$$r(d, c) = \underbrace{-\frac{\gamma}{\beta b}}_{\text{slope}} d + \underbrace{\left(\frac{(f - \alpha(c - c_x))}{\beta} + c_y\right)}_{\text{horizon}}. \tag{6}$$



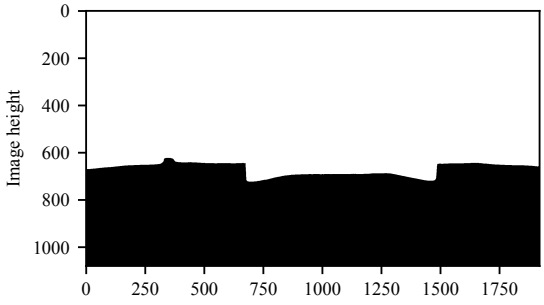
(a) Image of the dock.



(b) U-disparity map (I_u).



(c) Mapped U-disparity image (I'_{bin}).



(d) Free space mask (I_{fs}).

Figure 7: Each step of the process of creating a free space mask from a U-disparity map. The binary images are inverted such that black represents 1 and white represents 0 to make the lines more distinguishable from the background. Be aware that the U-disparity map is not in image space and has a different vertical axis than the other images.

The slope and the horizon are then used to map the disparity indices in the binary U-disparity map to its corresponding row indices r in image space. There is a risk of dividing by zero due to β being in the denominator. However, a small β would mean that the vessel is either facing straight up or down, which in practice is unlikely to happen. We also do not make any bold assumptions about roll as we compute eq. 6 for each column c in the U-disparity image.

The final step of the method is to create a free space mask I_{fs} representing the navigable area in image coordinates. Starting from the top of I'_{bin} seen in fig. 7c, a binary obstacle mask can be created by searching for the last non-zero occurrence, and set subsequent values to one. From the water surface mask I_{ws}, we already have a set of points that are more likely to belong to the water surface. To ensure that all values belonging to the water surface are included in the mask, we perform a bit-wise or-operation between the binary obstacle mask and the water surface mask to obtain the final free space mask (fig. 7d). From the free space mask I_{fs}, the free space boundary in image coordinates representing the starting point of each stixel can be obtained by searching for the last non-zero occurrence in each column, starting from the bottom of the image.

3.2 Obstacle masks with image segmentation

The dense disparity image obtained with ZED neural depth mode tends to over-smooth the disparity image. The contour of obstacles that are floating in the water can therefore end up blending in with the water's surface, making it more difficult to accurately estimate the free space boundary around those obstacles. To address these challenges, we use the recent image segmentation network FastSAM to segment out the water surface and potential obstacles (Zhao et al., 2023). FastSAM was chosen based on its generalization capability and fast performance. After segmentation, several masks are available as seen in fig. 8.

FastSAM does not provide labels, and manual classification is necessary to classify the water surface and potential objects. The water surface mask is in this paper not used for the refinement as it is not reliable enough in itself, leaving us with only the obstacle masks. To classify a mask as a potential water surface, we use the water surface mask I_{ws} found in sec. 3.1. However, only when a potential water surface mask has a sufficient amount of overlap with the water surface mask in the previous image it is classified as water. Masks belonging to potential obstacles are confirmed obstacle masks only when their bounding boxes in two consecutive frames have a significant overlap. The amount of overlap is evaluated based on the intersection over union (IOU). The IOU approach also works as a temporal filter and filters out spurious masks that only appear in a single frame. After all masks have been labeled as either obstacle or water, obstacle masks are then combined into one obstacle mask. A final free space mask is obtained by performing a bitwise and-operation between an inverted version of the obstacle mask with the free space mask I_{fs}. The free space boundary can then be obtained in the same way as described in sec. 3.1.

Figure 8: FastSAM used to segment out the water surface and all potential obstacles.

3.3 Representing obstacles with stixels

Stixels are vertically oriented rectangular segments with a fixed width that are used to represent objects as vertical planar surfaces. Representing objects with stixels has several advantages. It reduces the information from millions of pixels to a few hundred stixels. Each pixel within a stixel is also assumed to have the same depth value. Instead of working with single pixels that are sensitive to noise, we can instead take the median of all the values within a stixel to get a more accurate representation of the actual depth. The approach presented in this paper only represents the first obstacles that appear in front of the camera and are in contact with the water surface with stixels, we refer to this as first-level stixels. The starting point of each stixel is defined by the free space boundary computed from the final free space mask described in sec. 3.2.

The stixel height is estimated iteratively. Starting at the bottom of a stixel with the first row, the mean of all disparity values within the row is added to a list (fig. 9). For each value added to the list, the standard deviation is computed. If the standard deviation after the most recently added stixel row exceeds a threshold value, the algorithm will stop. The height of the stixel is then defined as the current row index at the time the

algorithm stops.

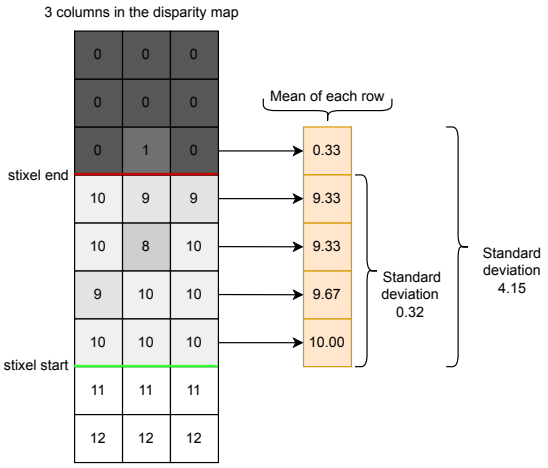


Figure 9: Computing the height of a stixel from the disparity image. Pixels in the disparity image are represented as gray-colored squares. The value of each pixel is the disparity.

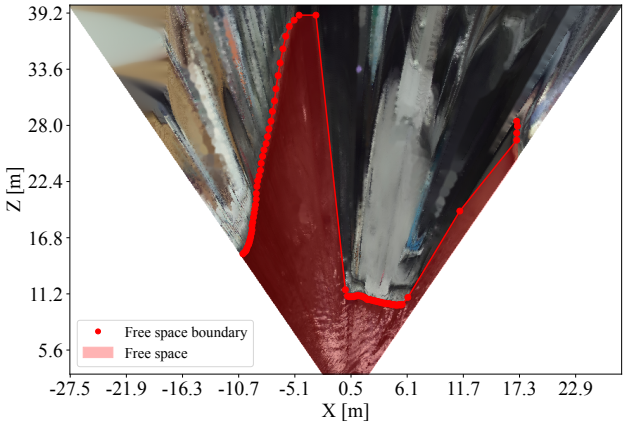
3.4 From stixels to a bird’s eye view representation

The stixel representation in itself provides useful information such as the vertical extent and location of the obstacles in the image coordinate frame. However, for localization and collision avoidance purposes, a top-down view of the reconstructed free space would be more useful. We reconstruct the location of each stixel using eq. 5, and project the points onto the XZ-plane to create a bird’s eye view (BEV) (fig. 10). Instead of relying on single disparity values, we use the median value of all disparity values that belong to a stixel for 3D reconstruction.

4 Evaluation

The dataset used in this paper was recorded using a stereo camera with approximately 12cm baseline. The camera was mounted on an autonomous ferry next to a LiDAR. The resolution of the videos was set to 1080p and the frame rate to 15fps. For evaluation, we used two different video sequences recorded on the same day (fig. 1a and 10a). In all the experiments, a stixel width of 20 pixels was used. The performance of the proposed method was evaluated by looking at the consistency of the reconstructed depth of each stixel over time and comparison of estimated stixel depth with depth measurements from a LiDAR.

(a) Image from the current scene showing the dock located at Ravnkloa.



(b) Computed free space for the scene shown in fig. 10a.

Figure 10: Bird’s eye view map (b) showing both the free space and the free space boundary. The estimated depth of a point can be seen along the vertical axis and is specified in meters.

4.1 Free space refinement with image segmentation

In fig. 11 a transparent inflatable ring is moving from left to right. Due to the smoothing of the disparity map, it is difficult to accurately separate the object boundary from the water surface based on the stereo depth information alone. To create stixels that accurately represent the ring, we utilize the object masks from FastSAM. From fig. 11b we can observe that refinement is necessary to accurately estimate a free space boundary that closely follows the edges of the object.

4.2 Ground truth with LiDAR

We compare the estimated stixel depth with depth from a LiDAR. The LiDAR point cloud is projected into the left camera using both camera intrinsic and camera-LiDAR extrinsic parameters. In fig. 12, one

(a) Without refinement with object segmentation masks

(b) With refinement with object segmentation masks

Figure 11: Objects with a short vertical extent are difficult to distinguish from the water surface and refinement with object masks from FastSAM is necessary.

can see the projected LiDAR points together with a zoomed-in view to easily visualize the number of points that fall within each stixel.

The camera-LiDAR extrinsic parameters, that is the relative pose of the sensors with respect to each other, were found by manual tuning. The LiDAR points that are on the border of an object are distinct from points outside the object both by not following the same line in the image, due to different distance from the camera, and also by color, see fig. 12. Therefore, the manual tuning of extrinsic parameters consisted of changing the position and orientation parameters until the LiDAR points on each object were on the correct object in the image. An Iterative Closest Point (ICP) method was tried but gave worse results with respect to visual verification, the LiDAR points belonging to an object did not match where that object was in the image. Manual tuning was done for different scenes, and especially for different distances to objects. An important finding to ease the tuning was to use close objects to tune the position parameters and use faraway objects to tune the orientation parameters.

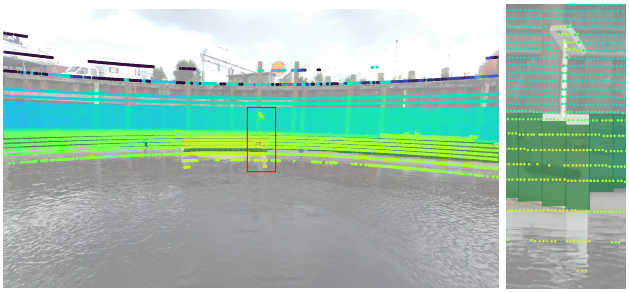


Figure 12: LiDAR points projected onto the image plane. The right image is a zoomed-in view of the red rectangular region visualized in the left image. The blue rectangles in the right image are stixels. The LiDAR points are colored by the distance from the camera.

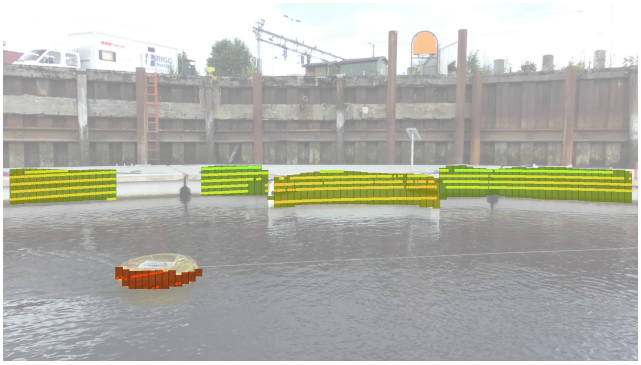
4.3 Accuracy and depth uncertainty

With the LiDAR points projected back into the image, we can compare the estimated stereo depth values with the depth from the LiDAR. However, for the stereo camera, there is a known uncertainty related to the disparity values. This uncertainty is due to the difficulty in matching the corresponding pixels in two images and will significantly impact the accuracy of the estimates. In the image coordinate frame, we can approximate this uncertainty as a Gaussian around each disparity value. However, for 3D reconstructed points, the uncertainty is more complex. The reconstruction process is nonlinear and the resulting 3D scene points will have a distribution depending on the depth. For a stereo camera with baseline b, focal length f, and disparity d, the nonlinear relation between disparity and depth is defined as:

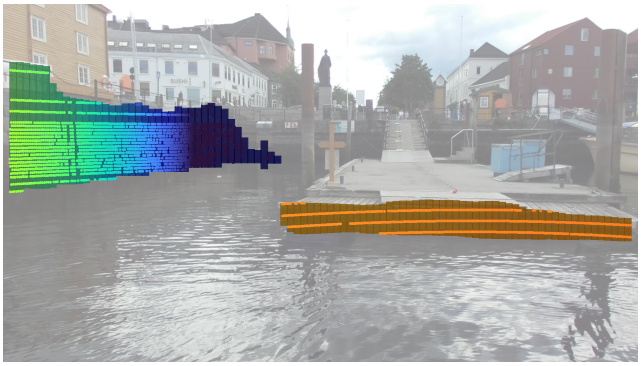
$$z = \frac{fb}{d}. \tag{7}$$

Without making any strong assumptions about Gaussianity, we can estimate the mean and standard deviation of the approximated Gaussian for each 3D point by propagating the disparity uncertainty across the nonlinear function seen in eq. 7. In this paper, this is done by employing an unscented transformation (Wan and Van Der Merwe, 2000).

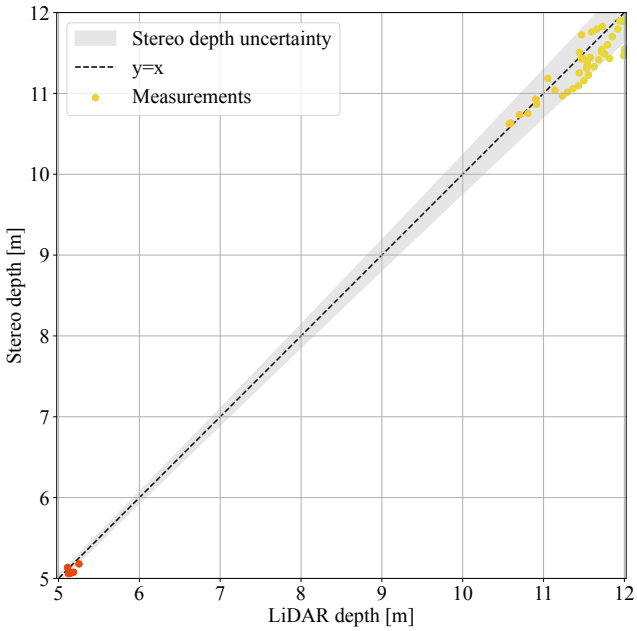
To visualize the results, we plot the stereo and LiDAR depth with the stereo depth uncertainty in a scatter plot (fig. 13). To avoid relying on single measurements that can be sensitive to noise, we use the median value of the LiDAR points and the 3D reconstructed stereo depth values that fall within a stixel. The uncertainty is visualized as an interval and is defined as the standard deviation as a function of the depth. The



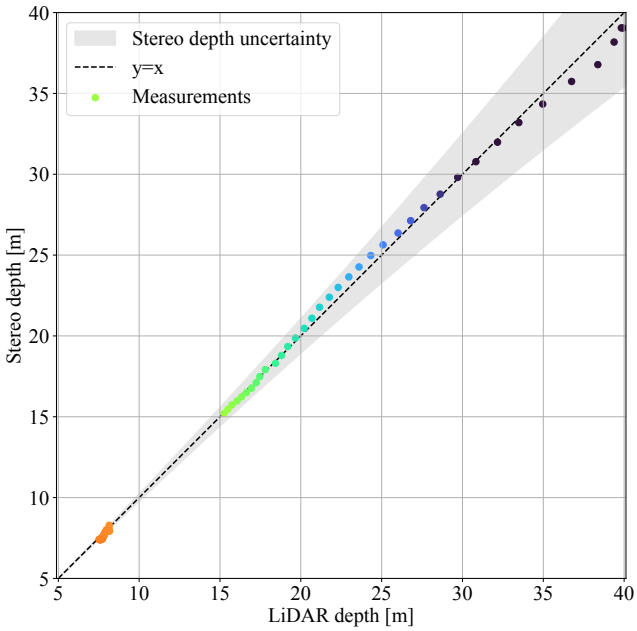
(a) Docking scenario with inflatable ring.



(b) Docking scenario at Ravnkloa.



(c) Scatter plot for docking scenario with inflatable ring.



(d) Scatter plot for docking scenario at Ravnkloa.

Figure 13: In fig. 13a and fig. 13b, stixels are visualized as colored rectangular segments, whereas the LiDAR measurements are plotted as colored points. In fig. 13c and fig. 13d, stereo and LiDAR depth are plotted together with the stereo depth uncertainty in a scatter plot. Measurements are plotted with the same color as its corresponding LiDAR points.

disparity uncertainty used for computing the depth uncertainty was empirically found by increasing it until most points end up within the interval, and was found to be approximately 0.5 pixels based on the scenarios that were used for testing. From fig. 13, one can see the scatter plot for two different scenarios. Ideally, if the depth measurements from the stereo camera are accurate, they will appear close to the dashed black line. If the points are either above or below the line, we either over or underestimate the depth, respectively. Notice in fig. 13c, that the measurements located on the inflatable ring at a distance of about 5m are more accurate than the points belonging to the dock 11−12m away. The same can also be said for the dock seen in fig. 13d. Based on these results it is clear that estimating the depths at far distances reliably with a short baseline stereo camera is not ideal.

4.4 Consistency in depth over time

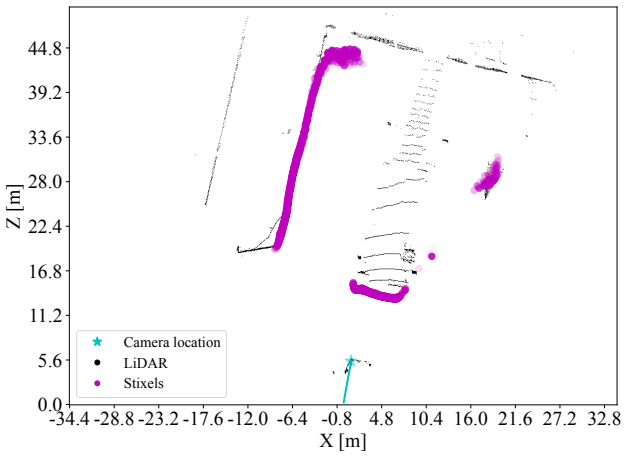
To visualize the consistency in depth over time, we created a BEV map in a world coordinate frame. The estimated free space boundaries from 20 consecutive image frames were overlaid and visualized together with measurements from a LiDAR in the same BEV frame (fig. 14). In order to go from camera coordinate frame c to world coordinate frame w, ego-motion compensation was performed. For a 3D scene point $p^c = [x, y, z]$ in camera frame, we can compute the point in the world coordinate frame p^w using the translation t_c^w and rotation R_c^w:

$$p^w = R_c^w p^c + t_c^w. \tag{8}$$

Different methods can be used to obtain the pose information. We used the Inertial Navigation System (INS) from the autonomous vessel used in the experiment. Notice in fig. 14 how the points are quite consistent when located closer to the camera compared to far away. Considering the depth uncertainty described in sec. 4.3, it is expected that estimates that are far away will vary more than estimates that are close to the camera.

To evaluate the robustness of the method, we tested the method on a more challenging scenario with two moving vessels, strong reflections, and small waves. Looking at fig. 15, we can observe that the vertical extent of each stixel is relatively consistent over time. However, the method does fail in frame 11 in fig. 15k on the kayak. This is possibly due to a missing or incomplete object mask from FastSAM. The short vertical extent of the kayak and its distance away from the camera makes it challenging to reliably estimate the stixels based on the stereo depth alone. The binary threshold for the U-disparity map also plays an important role in the overall performance of the method and might need to be adjusted in more challenging scenarios. A low threshold value will make the method more sensitive to waves and other disturbances in the water surface, but is more likely to be able to detect smaller obstacles.

(a) Image from the current scene showing the dock located at Ravnkloa.

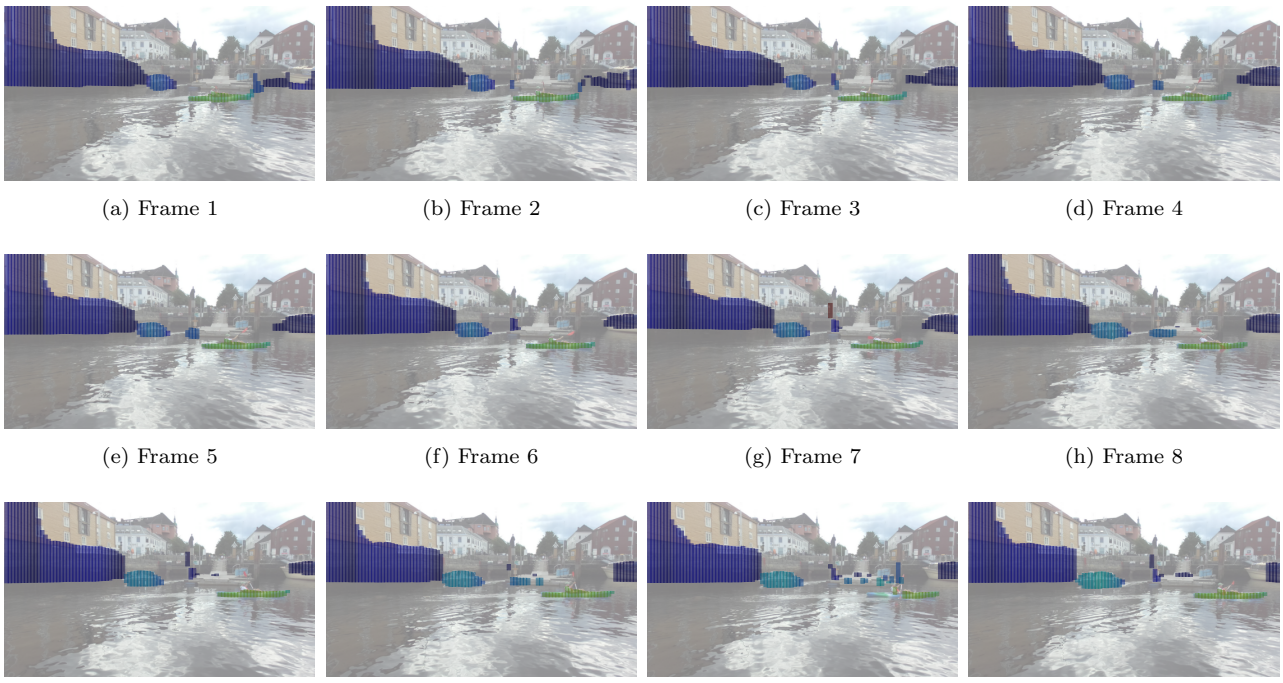


(b) BEV of the dock seen in fig. 14a.

Figure 14: The XZ-location of the free space boundary from 20 consecutive frames plotted together with LiDAR measurements in the same BEV map. Recently added free space boundary points are plotted with a higher opacity than older points. The current location of the camera mounted on the ferry is marked with a star symbol with its trajectory of the past locations plotted in blue.

5 Conclusion

In this paper, we introduced a novel stereo vision-based free space estimation approach for the maritime domain. We aim to address the challenging task of safe and efficient autonomous vessel docking, where safety is paramount for both passengers and the environment.



(a) Frame 1 (b) Frame 2 (c) Frame 3 (d) Frame 4

(e) Frame 5 (f) Frame 6 (g) Frame 7 (h) Frame 8

(i) Frame 9 (j) Frame 10 (k) Frame 11 (l) Frame 12

Figure 15: 12 consecutive frames from the docking sequence at Ravnkloa. Stixels are visualized as colored rectangular segments, where the color illustrates its depth. Refinement using segmentation masks from FastSAM was used in this experiment.

The maritime domain has seen substantial advancements in situational awareness, akin to the automotive industry, although stereo cameras, especially for docking, have been relatively overlooked compared to RADAR and LiDAR technologies. Our work highlights the potential of stereo cameras to improve docking safety and efficiency.

Our method utilizes a stixel representation, allowing for effective differentiation between horizontal surfaces and obstacles. The stixel representation provides a compact and meaningful portrayal of the environment and allows us to evaluate a set of pixels located within a stixel instead of single values to improve the accuracy of the reconstructed depth values.

We also address the uncertainty associated with stereo cameras, particularly the depth uncertainty. We employed an unscented transformation for a comprehensive estimation of the sensor limitations. The evaluation of our approach utilized a dataset recorded with a stereo camera on an autonomous ferry, demonstrating the accuracy and consistency of the method across different distances.

In conclusion, our approach holds promise for improving safety and efficiency in autonomous vessel docking, offering a bird’s eye view of navigable free space. As the maritime industry embraces autonomous technologies, our method contributes to improved situational awareness and decision making, ensuring safer docking operations for the benefit of passengers and the environment.

For future work, we envision refining our method to handle varying environmental conditions, including adverse weather or low-light scenarios, to ensure its robustness in diverse operational settings. Additionally, we want to work towards a real-time implementation and integration with autonomous navigation systems, which would propel this research toward practical deployment.

Acknowledgments

We extend our gratitude to Øystein Kaarstad Helgesen, for his valuable contributions and great help to the field experiment, which significantly influenced the outcome of this research. This work was supported by the Research Council of Norway-funded IKTPLUSS project ”Autosight (project number 333917): Autonomy through stereo vision near the seashore,” enabling the advancement of our research in autonomous maritime operations.

References

Badino, H., Franke, U., and Mester, R. Free Space Computation Using Stochastic Occupancy Grids and Dynamic Programming. In *Workshop on Dynamical Vision, ICCV, Rio de Janeiro, Brazil*, volume 20. page 73, 2007.

Badino, H., Franke, U., and Pfeiffer, D. The Stixel World - A Compact Medium Level Representation of the 3D-World. In J. Denzler, G. Notni, and H. Süße, editors, *Pattern Recognition*, Lecture Notes in Computer Science. Springer, Berlin, Heidelberg, pages 51–60, 2009. doi:10.1007/978-3-642-03798-6_6.

Bovcon, B. and Kristan, M. WaSR—A Water Segmentation and Refinement Maritime Obstacle Detection Network. *IEEE Transactions on Cybernetics*, 2022. 52(12):12661–12674. doi:10.1109/TCYB.2021.3085856.

Elfes, A. Using occupancy grids for mobile robot perception and navigation. *Computer*, 1989. 22(6):46–57. doi:10.1109/2.30720.

Fischler, M. A. and Bolles, R. C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. In M. A. Fischler and O. Firschein, editors, *Readings in Computer Vision*, pages 726–740. Morgan Kaufmann, San Francisco (CA), 1987. doi:10.1016/B978-0-08-051581-6.50070-2.

Griesser, D., Umlauf, G., and Franz, M. O. Visual Pitch and Roll Estimation For Inland Water Vessels. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*. pages 1961–1967, 2023. doi:10.1109/ICRA48891.2023.10160460.

Helgesen, Ø. K., Thyri, E. H., Brekke, E., Stahl, A., and Breivik, M. Experimental validation of camera-based maritime collision avoidance for autonomous urban passenger ferries. *Modeling, Identification and Control: A Norwegian Research Bulletin*, 2023. 44(2):55–68. doi:10.4173/mic.2023.2.2.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y., Dollár, P., and Girshick, R. Segment Anything. 2023. doi:10.48550/arXiv.2304.02643.

Muhovič, J., Mandeljc, R., Bovcon, B., Kristan, M., and Perš, J. Obstacle Tracking for Unmanned Surface Vessels Using 3-D Point Cloud. *IEEE Journal of Oceanic Engineering*, 2020. 45(3):786–798. doi:10.1109/JOE.2019.2909507.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 2011. 12:2825–2830.

Pfeiffer, D. and Franke, U. Efficient representation of traffic scenes by means of dynamic stixels. In *2010 IEEE Intelligent Vehicles Symposium*. pages 217–224, 2010. doi:10.1109/IVS.2010.5548114.

Pfeiffer, D. and Franke, U. Towards a Global Optimal Multi-Layer Stixel Representation of Dense 3D Data. In *the British Machine Vision Conference*. British Machine Vision Association, Dundee, pages 51.1–51.12, 2011. doi:10.5244/C.25.51.

Pinggera, P., Ramos, S., Gehrig, S., Franke, U., Rother, C., and Mester, R. Lost and Found: detecting small road hazards for self-driving vehicles. In *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. pages 1099–1106, 2016. doi:10.1109/IROS.2016.7759186.

Plenge-Feidenhans'l, M. *Robust Free Area Mapping for Autonomous Harbour Navigation*. Ph.D. thesis, Technical University of Denmark, 2023.

Plenge-Feidenhans'l, M. K. and Blanke, M. Open Water Detection for Autonomous In-harbor Navigation Using a Classification Network. *IFAC-PapersOnLine*, 2021. 54(16):30–36. doi:10.1016/j.ifacol.2021.10.069.

Schneider, L., Cordts, M., Rehfeld, T., Pfeiffer, D., Enzweiler, M., Franke, U., Pollefeys, M., and Roth, S. Semantic Stixels: Depth is not enough. In *2016 IEEE Intelligent Vehicles Symposium (IV)*. pages 110–117, 2016. doi:10.1109/IVS.2016.7535373.

Soquet, N., Perrollaz, M., Labayrade, R., and Aubert, D. Free Space Estimation for Autonomous Navigation. *International Conference on Computer Vision Systems*, 2007. doi:10.2390/biecoll-icvs2007-30.

Volden, Ø., Stahl, A., and Fossen, T. I. Vision-based positioning system for auto-docking of unmanned surface vehicles (USVs). *International Journal of Intelligent Robotics and Applications*, 2022. 6(1):86–103. doi:10.1007/s41315-021-00193-0.

Wan, E. and Van Der Merwe, R. The unscented Kalman filter for nonlinear estimation. In *IEEE Adaptive Systems for Signal Processing, Communications, and Control Symposium*. pages 153–158, 2000. doi:10.1109/ASSPCC.2000.882463.

Yao, J., Ramalingam, S., Taguchi, Y., Miki, Y., and Urtasun, R. Estimating Drivable Collision-Free Space from Monocular Video. In *IEEE Winter Conference on Applications of Computer Vision*. pages 420–427, 2015. doi:10.1109/WACV.2015.62.

Zhao, X., Ding, W., An, Y., Du, Y., Yu, T., Li, M., Tang, M., and Wang, J. Fast Segment Anything. 2023. doi:10.48550/arXiv.2306.12156.