GPS-level accurate camera localization with HorizonNet

Bertil Grelsson, Andreas Robinson, Michael Felsberg and Fahad Khan

GPS-level Accurate Camera Localization with HorizonNet

Bertil Grelsson[1,2] **Andreas Robinson**[1] **Michael Felsberg**[1]

Fahad Shahbaz Khan[1,3]

[1]Computer Vision Laboratory, Linköping University, Sweden

[2]Saab Dynamics, Linköping, Sweden

[3]Inception Institute of Artificial Intelligence, Abu Dhabi, UAE

{bertil.grelsson,andreas.robinson,michael.felsberg,fahad.khan}@liu.se

Abstract

This paper investigates the problem of position estimation of unmanned surface vessels (USVs) operating in coastal areas or in the archipelago. We propose a position estimation method where the horizon line is extracted in a 360° panoramic image around the USV. We design a CNN architecture to determine an approximate horizon line in the image and implicitly determine the camera orientation (the pitch and roll angles). The panoramic image is warped to compensate for the camera orientation and to generate an image from an approximately level camera. A second CNN architecture is designed to extract the pixelwise horizon line in the warped image. The extracted horizon line is correlated with digital elevation model (DEM) data in the Fourier domain using a MOSSE correlation filter. Finally, we determine the location of the maximum correlation score over the search area to estimate the position of the USV. Comprehensive experiments are performed in field trials conducted over three days in the archipelago. Our approach provides excellent results by achieving robust position estimates with GPS-level accuracy in previously unvisited test areas.

1 Introduction

In recent years, unmanned systems such as Unmanned Aerial Vehicles (UAVs), Unmanned Ground Vehicles (UGVs), and Unmanned Surface Vessels (USVs) have become increasingly popular providing safe and secure operations in remote environments. Within unmanned systems, the aim of USVs is to perform various ocean sensing tasks in a variety of cluttered sea environments. Generally, these USVs (autonomous or tele-operated) are reliant on accurate position measurements provided by the Global Positioning System (GPS) for safe navigation. However, the GPS signal is not always available and reliable. GPS outages are rare (William J. Hughes Technical Center, 2014), but they do occur and they need to be accounted for in the USV navigation system. Perhaps a more severe issue, in one of the early test trials conducted for this paper with a USV, we experienced a case with a short time period of completely erroneous GPS measurements. The GPS receiver repeatedly computed position estimates which were located south of the equator, and not in southern Sweden where the trials were actually performed. The GPS receiver fed the USV autopilot with these faulty position estimates, and the autopilot completely lost control of its true position and the appropriate heading to proceed to the next waypoint in the planned mission. The pilot, being standby to tele-operate the USV, had to save the situation. The onboard USV navigation system must be capable of handling such erroneous position estimates from the GPS. Moreover, in hostile ocean scenarios, the GPS signal can be unreliable or not available at all. All in all, these potential shortcomings of the GPS create the need for alternative and complementary position sensors. In this paper, we tackle the problem of providing accurate position measurements for a USV operating in challenging coastal areas or in the archipelago.

Terrain navigation, without any need for externally controlled signals and sensors, is suitable as an alternative position sensor in these scenarios. Terrain navigation is a family of techniques where measurements of the surrounding terrain are correlated with a terrain spatial database to provide position measurements. Terrain navigation is well established as a position sensor and the underlying sensor measurements could originate from various techniques such as radars, sonars, altimeters, lidars, and cameras (Vaman, 2012; Han et al., 2016; Melo and Matos, 2017). For the position estimation in our scenario, the USV can utilize information obtained from an omnidirectional camera, a digital compass, and a high-resolution digital elevation model (DEM) of the operational area (see Figure 1).

In recent years, deep learning and convolutional neural networks (CNNs) have significantly boosted the level of performance for various computer vision tasks including image classification, object segmentation, and object tracking. Generally, a CNN consists of two main parts. The first part, the encoder, generates powerful

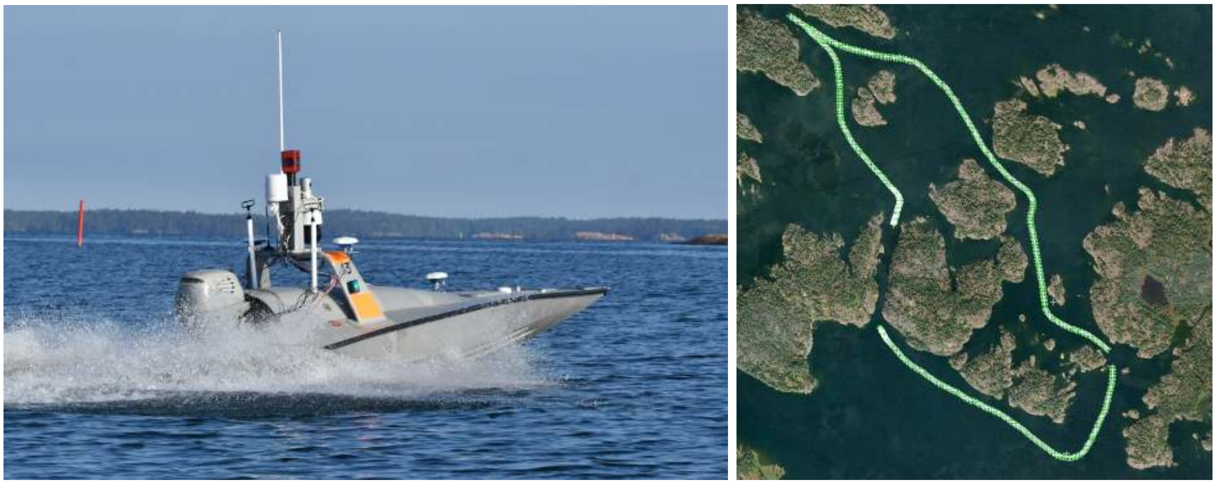

Figure 1: Left: The USV equipped with a Ladybug omnidirectional camera. Right: Test image position estimates with the GPS (green) and the proposed method (white).

feature descriptors to represent the input image at various levels of detail. The second part utilizes these image representations and, depending on the application, is trained to output e.g. the image object class, a pixelwise segmentation, the location of a tracked object, or the camera location. For unmanned systems, CNNs have e.g. been used for object recognition in UAV images (Radovic et al., 2017), road lane guidance for autonomous cars (Nugraha et al., 2017), and collision avoidance manoeuvers for USVs (Xu et al., 2017). In this paper, we employ CNNs to extract terrain information from USV images to generate accurate position measurements for the USV.

To aggregate terrain information from consecutive images and to align the image content with DEM data for USV position estimation, object tracking and registration are essential components. In recent years, the best performing visual object trackers apply a discriminatively trained correlation filter (DCF) on top of multidimensional features (Kristan et al., 2017). The foundation for DCF-based trackers is the MOSSE correlation filter (Bolme et al., 2010). A filter is trained to model the appearance of the tracked object in some example images. To search for the object in the next frame or, in our case, to register the object with DEM data, a correlation score is computed over the search area. For computational speed, the correlation is performed in the Fourier domain.

For terrain navigation of a manned surface vessel in coastal areas and in the archipelago, humans would intuitively use readily observable characteristic landmarks in a few directions, project the directions on a sea chart or map, and use cross bearing to determine the vessel position. The horizon line most often constitutes a spatially extended characteristic landmark for islands and the shore. Matching of the complete horizon line around the vessel with a map is mathematically a more robust cross bearing measurement than just using a few directions. To obtain highly accurate position estimates, the horizon line must be captured with high angular resolution, which camera sensors can provide. These insights are the motivations behind our proposed position estimation method.

Our proposed method requires that an approximate position is known to limit the search area and to make the method real-time capable, which is necessary if it is to be used as a position sensor for onboard navigation of the USV. The approximate position could e.g. be obtained from continuous position tracking over time using the proposed method from mission start, GPS measurements (when available and reliable), cross bearing measurements based on detection of seamarks/landmarks, a large scale position estimation method as in (Baatz et al., 2012), or position estimates from the proposed method at a coarser scale.

We propose a position estimation method where two CNNs are employed to extract the camera orientation and the horizon line, respectively, in a 360° panoramic image around the USV. The horizon line is correlated with DEM data in the Fourier domain using a MOSSE correlation filter. Finally, we determine the location of the maximum correlation score over the search area to estimate the position of the USV. The core of the proposed method has previously been presented in (Grelsson et al., 2018). In this paper, we provide new comprehensive field trials performed over three days in different locations of the Swedish east-coast archipelago. The results demonstrate that our method can be trained on previously captured image data from one region and achieves a global position accuracy of 2.72 ± 1.58 meters relative to the GPS ground truth data when evaluated on images from a previously unvisited area. To reduce the search time, we provide evidence that our method can be used in a multi-grid approach. We verify that our method works at a coarser scale to generate a slightly less accurate position estimate, which is then refined at a finer scale. We also show that our method can be used in applications with narrower field of view (FOV) images than a full 360° panoramic image. The evaluation shows that the position accuracy of our method degrades gracefully when narrowing the FOV. The field trials and the results achieved are described in section 5. Figure 1 shows an image containing the USV with the omnidirectional camera (left) and the position estimates (right) obtained with the GPS (green) and the proposed approach (white).

Our contributions are: 1) The proposed method for USV terrain navigation, 2) CNNs designed for camera orientation estimation and horizon segmentation in a marine environment, 3) Horizon line registration with a MOSSE correlation filter, 4) Comprehensive field trials that demonstrate the GPS-level accuracy of the proposed method.

2 Related work

Our proposed method for camera localization includes two CNNs for fast estimation of the camera orientation and segmentation of the horizon line in the image. Any fast and accurate segmentation method would fit into our proposed method. In this section, however, we focus on CNN-based methods. Automatic extraction of the horizon line and water line (which is the first water to land/sky transition seen from the vessel) in the image is equivalent to segmentation of the image into sky/land/water groups. Previous works (Lee et al., 2017; Verbickas and Whitehead, 2014) have investigated the use of CNNs to determine straight lines and sky segmentation in images. In (Lee et al., 2017), the authors propose an approach where a CNN is trained to find straight lines in order to extract semantically meaningful information from the image. A CNN-based method for sky segmentation is proposed in (Verbickas and Whitehead, 2014). In their approach a simple two convolutional layer network is trained from scratch on the authors' own dataset. A convolutional and deconvolutional network is employed in (Porzi et al., 2016) to determine the horizon line in full size, i.e. not in a downsampled image. A comparison of deep learning methods for horizon/sky line segmentation is presented in (Ahmad et al., 2017). However, their evaluation is only performed on sky/mountain images. To the best of our knowledge, we have not encountered any public benchmark or CNN-based segmentation method with respect to marine images and USVs.

In recent years, several surveys of image-based localization methods have been published by (Piasco et al., 2018) for urban environments, (Wu et al., 2018) focussing on unknown environments and different types of SLAM methods, and (Brejcha and Čadík, 2017) for city-scale and natural environments. In the latter survey, the localization methods are classified based on the reference data used. The authors refer to two main classes of methods, *Image-based* methods and methods utlizing *Multiple modality data*, e.g. having a terrain model as a reference. A flow chart of localization methods in accordance with this classification is shown in Figure 2.

The *Image-based* methods require a large database of geo-tagged images from the test area. In an urban environment, the database can often be made available from public photographs or street-view images taken from cars. The database enables e.g. *image retrieval* methods for localization. The location of a query image is inferred by retrieving similar images from the database using various matching algorithms including Bag-of-Words and hashing approaches (Arandjelovic et al., 2016; Sattler et al., 2017; Chum et al., 2009). Another option for localization is *Train and regress* methods where the image database is used to train a classifier and then directly regress the location of the query image (Kendall et al., 2015; Weyand et al.,

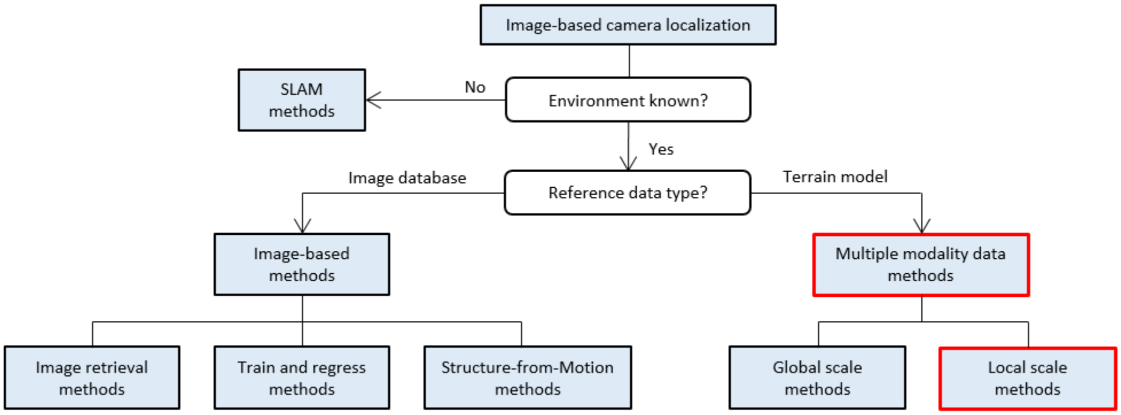

Figure 2: Flowchart of image-based localization methods. Our proposed method is a local scale method utilizing multiple modality data.

2016; Felsberg and Hedborg, 2007). An image database also enables 3D reconstruction of the scene using *Structure-from-Motion* (SfM). Various techniques have been proposed to align the query image with the 3D model to infer the camera location (Irschara et al., 2009; Sattler et al., 2011; Li et al., 2012).

In a natural environment covering large areas without any infrastructure or road network to guide your movements within the area, these image databases are rarely available. Often you do not have access to any images from the test area. This calls for cross-domain matching of the query image with *Multiple modality data*. Terrain models are commonly used as reference data since they are easily accessible and readily generated worldwide with large scale coverage using satellite imagery or radars (SRTM, 2019). For cross-domain registration of images, features such as horizon lines and edge maps are often utilized. The class of *multiple modality data* methods can be further divided into global scale methods (Brejcha and Čadík, 2017), striving to estimate a coarse position within a large search area, and local scale methods aiming at accurate position estimates within a smaller search region. Our proposed method is a local scale method which is capable to provide accurate position estimates in a previously unvisited test area.

To the best of our knowledge, the first large scale localization method using *multiple modality data* was presented by (Baatz et al., 2012). They segment the horizon line in the image and extract contour word descriptors called contourlets. They use a DEM to generate a database of contours from a 360° view. Then they employ a Bag-of-Words approach to search for a contour in the database that matches all contourlets in the query image to infer the camera location. They correctly locate 88% of the mountainous test images within a 1km radius searching over the whole country of Switzerland. A similar large scale method is presented by (Tzeng et al., 2013). They propose an alternative feature descriptor based on the concavity of the horizon line. They use geometric hashing to find candidate matches with synthesized horizon lines from a DEM to localize the query image.

Most previous works on horizon registration on the local scale, where a rough position is already known, have been performed in the spatial domain. Woo et al. compute the curvature of mountain peaks in the image plane and on DEM data (Woo et al., 2007). They use a Markov Chain Monte Carlo method to generate position hypotheses and to find the best match over the search area. Ramalingam et al. use an omnidirectional camera in order to estimate the position of car in a city (Ramalingam et al., 2009). They segment the skyline in the image and use graph cuts to find the best match with the skyline generated from DEM data. Dumble and Gibbens precompute reference horizon profiles from DEM data in the Alps in a set of 3D grid points (Dumble and Gibbens, 2015). They extract the reference profile at the grid point closest to the assumed position of their aerial vehicle. To refine the estimated location, they use gradient descent to iteratively minimize the error between the horizon line in the image and the transformed reference profile. The method requires a large horizon profile variance, which prevents its use in our scenario in the archipelago with low altitude islands. A method for accurate registration of low variance horizon profiles is proposed by (Grelsson et al., 2016), but they only estimate the camera orientation and not the position. Another method suitable for low variance horizon lines is proposed by (Chiodini et al., 2017), where a Mars rover is localized by matching the detected skyline with DEM data. For position estimation, they do a grid search over the location and the viewing angle, and minimize the least-square error between the detected and the rendered skyline.

There are some previous works on registration of the horizon line with DEM data in the Fourier domain. To align and annotate mountain pictures captured at a known position, Baboud et al. detect edges in the image and match with silhouettes from the DEM data (Baboud et al., 2011). The matching to find the orientation angles is performed using spherical cross correlation in the Fourier domain. The orientation estimates obtained are very accurate. The processing time (corresponding to a GeForce GTX 1080 Ti) is around 15s per image, which is why the method would not be suitable for online navigation of a USV searching over an area with multiple position hypotheses. The work of Brejcha and Čadík builds on the previous method and they complement the edge lines with semantic information to make the registration more robust (Brejcha and Cadík, 2018). The registration to find the camera orientation is performed with spherical cross correlation in the Fourier domain. The introduction of the MOSSE correlation filter (Bolme et al., 2010) showed that image object tracking with adaptive correlation filters in the Fourier domain is significantly faster and also more robust to variations in target appearance than previous trackers working in the spatial domain.

3 Classical methods for position estimation

The approach of our position estimation method is based on registration of the horizon line with DEM data. In this section, we sketch an algorithm with classical computer vision methods for position estimation. This algorithm is used to create the target labels for the CNNs in the proposed method. It also provides a baseline for the position estimate accuracy that can be obtained by registration in the spatial domain. An algorithm flowchart is shown in Figure 3. For each step in the algorithm, an example image is included in the flowchart to illustrate the output from that step. Note that the illustrations are cropped images for better visibility.

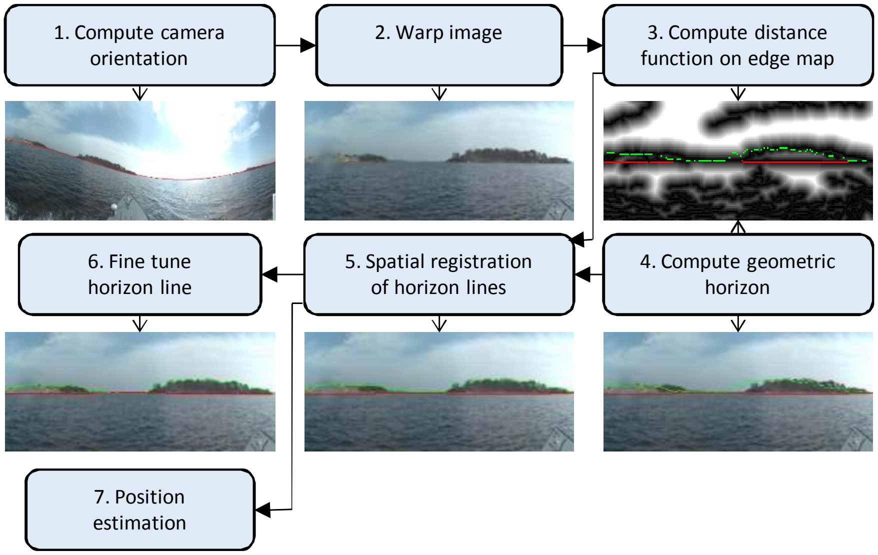
1. Compute camera orientation
2. Warp image
3. Compute distance function on edge map
4. Compute geometric horizon
5. Spatial registration of horizon lines
6. Fine tune horizon line
7. Position estimation

Figure 3: Flowchart for position estimation with classical methods. Predicted horizon line (red) overlaid on image (1). Geometric horizon line (green) and water line (red) overlaid on distance function (3) and warped image (4) before registration, after registration (5), and after fine tuning (6).

The camera used in our field trials generates panoramic images in a cylindrical projection. First, to determine an approximate camera orientation from the image, we use Canny edge detection (Canny, 1986) and Hough voting (Hough, 1962). We search for the approximate horizon plane on the unit cylinder, which will be an S-shaped curve in the panoramic view. We adapt the method in (Grelsson et al., 2016) and vote for the normal vector of the horizon plane parameterized with the pitch and roll angles.

The second step in the algorithm is to warp the image to compensate for the camera orientation. The warping will create an image corresponding to an approximately level camera, suitable for the subsequent registration process with the geometric horizon. As a third step, to prepare for the registration, we compute a Canny edge image on the warped image and a distance function D based on the edge image. Another input to the registration is the geometric horizon line from the digital elevation model. The DEM for the test site was provided by Vricon[1]. The DEM is computed from recent satellite imagery and has a pixel resolution on the ground of 0.5m. The altitude accuracy is in the same order as the pixel resolution. The DEM of the test area is illustrated in Figure 4.

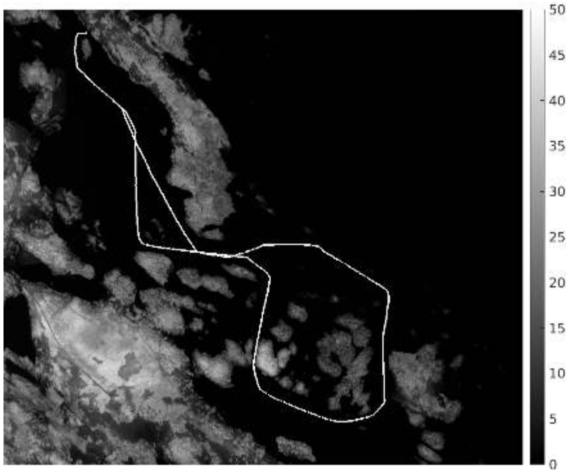

Figure 4: Digital elevation model of the test area with GPS trajectory (white) from one trial day overlaid. The bar shows the altitude above sea level in meters.

To create the target labels for training the CNNs, we use the position given by the GPS and the heading angle from the digital compass or, if not available, the tangent vector from the logged GPS trajectory. The geometric horizon is generated by ray-tracing using DEM data. For the desired number of image columns around the vessel, we extract the altitude profile from the DEM. In each direction, we compute the elevation angle of all objects along the ray. The maximum elevation angle is taken as the horizon point in that direction. The water line point, the first water to land/sky transition in each direction, is taken as the vertical viewing angle to the point where the DEM makes a step larger than a 0.2m threshold along the ray direction. The search radius is set to 6km in all directions. The ideal horizon, assuming a flat and spherical earth, is at 3.2km for a camera at 1.0m height, but we add some margin to cope with the topography. When overlaid on the image, the geometric horizon may be slightly off-set due to small errors in the pitch, roll, and heading angle estimates.

Ideally, the complete geometric horizon line would be located where the distance function D is zero. To find the horizon line in the image, we search for a rotation of the geometric horizon line points on the unit cylinder, such that when projected onto the image, their summed distance function values will be a

[1] https://www.vricon.com/

minimum, i.e. we minimize the score

$$s = \underset{\theta,\phi,\psi}{\operatorname{argmin}} \sum_i \mathrm{D}\left\{\mathbf{R}(\psi)\pi(\mathbf{R}(\theta,\phi)\pi^{-1}(h_i))\right\} \tag{1}$$

where h_i are the horizon line points, π is the projection from the unit cylinder to the image surface, θ, ϕ, and ψ are the pitch, roll and heading angles, $\mathbf{R}$ is a rotation matrix and D is the distance function.

For registration, we perform a grid search over the pitch, roll and heading angles. For the first two angles we need to compute the rotation matrix and project the transformed points onto the image plane. The heading angle rotation simply corresponds to a horizontal shift on the image. The step size in pitch and roll is set to 0.25° and we search over the range ±2°. We extract the rotation angles for the minimum score and project the geometric horizon line and water line onto the warped image after transformation with the said rotation angles. In general, there is a good fit between the geometric horizon line and water line with the image content, but occasionally there are small deviations. The main reason is that the DEM is not a perfect representation of the real world. To adjust for these discrepancies we perform a final tuning step of the geometric horizon line in the image. For final tuning we first convolve the warped image with a Sobel filter to enhance gradients in the vertical direction. For each image column we search for a local maximum of the gradient, exceeding a threshold, in a small region close to the horizon line obtained in the registration step. If no gradient maximum is detected in the search region, the horizon line from the registration is retained. Image columns occluded by sensors and antennas on the vessel are excluded from the tuning process.

For position estimation, the algorithm is run for various positions over an XY grid and the minimum score obtained according to (1) is recorded for each position. We extract the location of the minimum score over the search area. To refine the position estimate, the scores of the nearest neighbors to the minimum are extracted in the X and Y directions, see Figure 5. A second order polynomial fit is applied in the X and Y directions to obtain a refined subgrid position estimate.

The main drawback with this classical method for position estimation is that it is prohibitively slow for real-time applications. In our implementation, the spatial registration took more than 1s per position grid point, whereas our proposed method below achieved more than 40 grid points per second.

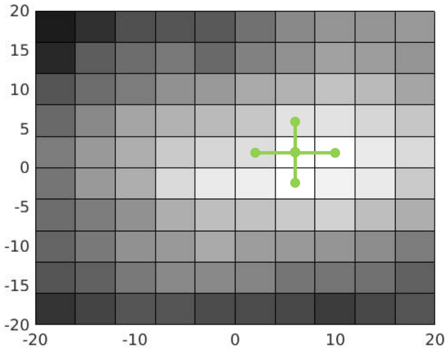

Figure 5: Correlation scores are extracted from the nearest neighbors to the minimum score over the search area in the X and Y directions to compute the subgrid position estimate.

4 CNN-based position estimation method

Prior to designing our proposed method, we experimented with an end-to-end CNN to output the position estimate directly from the input image. We tried a modified version of Posenet (Kendall et al., 2015), which we adapted to our image input format. We trained it to learn the camera position and orientation, but the training failed completely and did not converge to anything useful. The reason for this behavior is straightforward. Posenet was designed for images on land. In our images, only a small part (the land objects) contain information that is useful for position estimation. The sky and sea change appearance over time and will only distract the position estimation if using the full image content. Hence, prior knowledge about what is the relevant part of the image for position estimation is required.

This insight motivates the design of our proposed method for position estimation, which is similar to the classical method in its architecture. The proposed algorithm consists of seven steps, which are explained in detail in this section and shown in a flow chart in Figure 6.

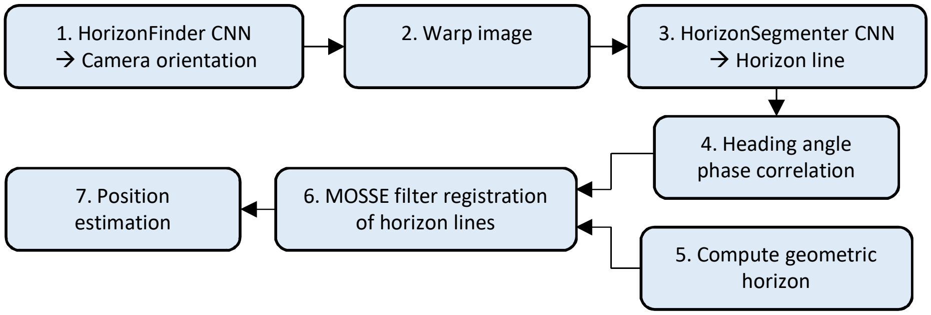

Figure 6: Flowchart for position estimation with the proposed CNN-based method.

4.1 HorizonFinder CNN

To determine a rough camera orientation (pitch and roll angles), i.e. to find an approximate horizon line in the panoramic image, we employ a convolutional neural network called HorizonFinder. This CNN replaces the Canny detector and the Hough voting in the classical method.

We use a ResNet50 network (He et al., 2016) pretrained on ImageNet to provide feature descriptors. We then add two stacks with a convolutional layer, vertical pooling, and a Leaky Rectified Linear Unit (Lrelu) (Maas et al., 2013) activation function. Finally, we have a fully connected layer to output the camera orientation angles. We found that the network training was more accurate when the output was the cosine and sine of the pitch and roll angles, θ and ϕ, rather than having the orientation angles by themselves as output. Since we had relatively few training images, only the weights of the new network layers (after ResNet50) were trained. The network design is shown in Figure 7.

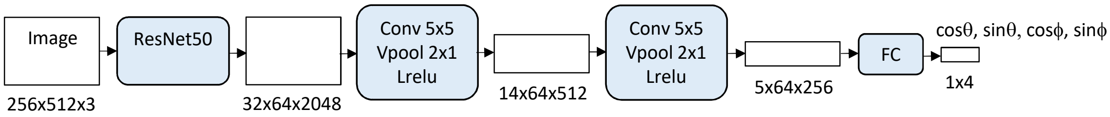

Figure 7: Network design for HorizonFinder. The numbers denote the input and output size (H×W×C) for each network layer.

As a loss function we use the L1 loss between the predicted and target labels for the parameters $\cos\theta$, $\sin\theta$, $\cos\phi$, and $\sin\phi$. This network design and loss function gave satisfactory results as judged by viewing the backprojected horizon line on the panoramic image. Since we did not have access to any exact ground truth for the camera orientation from external sensors, we could not perform any quantitative comparison for different network designs and loss functions. We trained the network for 100 epochs. The initial learning rate was set to 0.0001 and it was then reduced by a factor of two every nine epochs.

4.2 Image warping

Based on the predicted pitch and roll angles from the HorizonFinder CNN, we warp the panoramic image to compensate for the camera orientation. Since the horizon now will be almost vertically centered in the image, we only warp the central part of the image. The size of the original panoramic image is 2048×1024, whereas the warped image is 2048×384.

4.3 HorizonSegmenter CNN

To predict the location of the horizon line and water line in the warped image, we employ a second CNN called HorizonSegmenter, which was proposed in our previous paper (Grelsson et al., 2018). The water line, when seen from a camera at low height with small gracing angles to the sea, was found to not improve the position accuracy and it is, in contrast to our previous paper (Grelsson et al., 2018), no longer used for registration in the proposed method nor in the spatial registration method.

The HorizonSegmenter CNN has no one-to-one counterpart in the classical method. We use a similar network design as for HorizonFinder. We start with a pretrained ResNet50 to generate feature descriptors. We use three stacks of layers, each comprising a convolutional layer, vertical pooling, and a Lrelu activation function. To obtain the same horizontal resolution as the target labels, we insert two horizontal upsampling layers with bilinear interpolation. Finally, a fully connected layer is added to output the vertical pixel location of the horizon line and water line for all image columns. As a loss function we use the absolute pixel difference between the predicted and target horizon line and water line summed over the training image. The network design is shown in Figure 8.

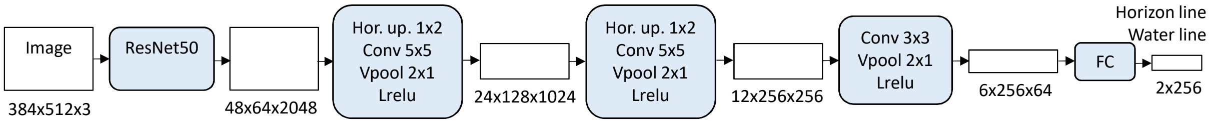

Figure 8: Network design for HorizonSegmenter. The numbers denote the input and output size (H×W×C) for each network layer.

To avoid considerable overfitting during training due to relatively few images, we randomly generate training images from the original warped images. We make a random horizontal crop of a 512×384 image from the original image during training. We used the same learning rate scheme as for the HorizonFinder network, and only the weights of the new network layers were trained.

4.4 Phase correlation for relative heading angle measurements

For the registration with the MOSSE correlation filter in the next step, we need the camera heading angle for each image in a global coordinate system as an input. The absolute heading angle of the first image in a sequence can be obtained from a digital compass (without relying on GPS) with an accuracy better than 2° (Airmar GH2183 specification , 2019). In our field trial we did not have access to a digital compass. Instead, we replaced the heading for the first image with the ground truth heading from the GPS trajectory plus some noise simulating the digital compass.

To find the relative change in heading angles from one video frame to the next, we employ phase correlation matching (Meneghetti et al., 2015) of the horizon line output from the HorizonSegmenter network for successive images. We denote the Fourier transform of the complete 360° horizon line for two consecutive images with I and J respectively. For this image pair, we compute the signal

$$s = \mathscr{F}^{-1}\left\{\frac{J^{\star} \cdot I}{\|J^{\star} \cdot I\|}\right\} \quad , \tag{2}$$

where * is the complex conjugate and $\cdot$ denotes element-wise multiplication. The phase angle of s is a measure of the heading angle change between the two images. The estimated heading angle ψ_i of image i in a sequence is given by

$$\psi_{i+1} = \quad \mathrm{mod}\ (\arg(s) + \psi_i, 2\pi) \quad , \tag{3}$$

where ψ_0 is the heading angle taken from the digital compass.

4.5 Compute geometric horizon

The geometric horizon is computed in exactly the same manner as described in the classical position estimation method.

4.6 MOSSE correlation filter

We adapt the MOSSE (Minimum Output Sum of Squared Error) correlation filter (Bolme et al., 2010), originally developed for visual object tracking. The MOSSE filter is designed to generate a desired output signal (typically a Gaussian) shifted to the temporal or spatial location most closely corresponding to a set of learned reference signals. For efficiency, the filter is evaluated in the Fourier domain. The filter is trained with multiple references to improve its robustness against changes in appearance and noise.

In our case we use the segmented horizon lines from an image sequence projected onto the unit cylinder as the reference. We align the segmented horizon lines in the spatial domain in accordance with their estimated heading angle. In the spatial domain, they are all centered around the estimated mean heading angle of the images in the sequence. In general, we use a sequence of ten consecutive images to compute the MOSSE filter. Figure 9 shows ten segmented horizon lines (green) from an image sequence, and the geometric horizon line (black) computed from one position hypothesis. We select the target signal to be a one-dimensional Gaussian signal g that makes one revolution on the unit cylinder, see Figure 9. The Gaussian target signal

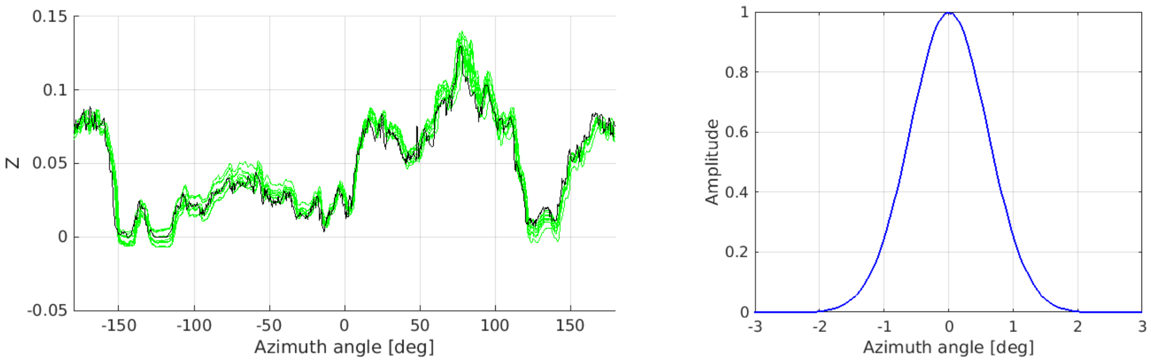

Figure 9: Segmented horizon lines (green) from ten consecutive images and the geometric horizon line (black) projected onto the unit cylinder (left), from position D in Figure 11. Gaussian target signal on the unit cylinder (right).

is also centered around the estimated mean heading angle of the images. The bandwidth of the target signal is chosen to be around 1°. In the Fourier transform domain, we denote the segmented horizon lines in the sequence with F_i and the Gaussian target signal with G. We want to find the MOSSE filter K, which minimizes

$$K = \underset{K}{\operatorname{argmin}} \sum_i |F_i \cdot K^\star - G_i|^2 . \tag{4}$$

We compute the MOSSE filter K as

$$K^\star = \frac{\sum_i G \cdot F_i^\star}{\sum_i F_i \cdot F_i^\star} \quad , \tag{5}$$

and the MOSSE filter signal response as

$$r = \mathscr{F}^{-1}\left\{H \cdot K^\star\right\} \quad , \tag{6}$$

where H is the Fourier transform of the geometric horizon line from the DEM. Ideally, r will be the Gaussian target signal with a zero phase shift if the heading angle estimate is correct. An erroneous heading angle estimate will generate an angular shift of the MOSSE filter signal response. Since the digital compass gives the absolute heading with an accuracy around 2°, we search for the peak signal response within a ±5° band from the center to have some margin. As a quality measure of the signal response, i.e. our MOSSE filter correlation score, we use the peak-to-output-energy ratio (Javidi and Wang, 1994). We suppress the response within the expected Gaussian signal bandwidth around the detected peak and compute the average energy over the remainder of the signal response. The score is the peak signal over the square root of the average energy.

4.7 Position estimation

To generate a position estimate, we compute the MOSSE filter correlation score for an image sequence in various positions over an XY grid. We extract the maximum score over the search area and apply a second order polynomial fit, in the same manner as for the classical method, to obtain a refined subgrid position estimate.

5 Experiments and Results

5.1 Field trials

Field trials were performed on three days in the archipelago outside Västervik in Sweden. The first two days were consecutive days and the third trial day occurred two weeks later. The three days are denoted day 1, 2 and 3 in the sequel. In the field trials, omnidirectional images were captured with a Ladybug3 camera (FLIR Ladybug3 specification, 2019) mounted on a tele-operated USV (4m long), see Figure 10. Each day about 25k images were captured at 10 fps during a 40-45 minute trial. The USV position was measured with a U-blox EVK-8 GPS receiver (U-blox EVK-8 specification, 2019) acquired at 1 fps. The relative positions between the camera and the GPS antenna are illustrated in a local USV coordinate system in Figure 10. The USV trajectories during the field trials, as measured by the GPS, are shown in Figure 11.

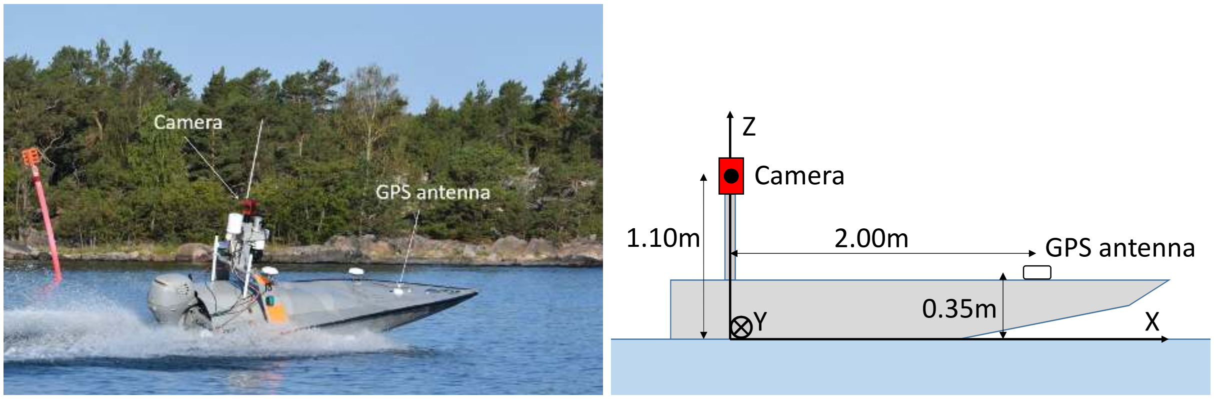

Figure 10: Left: Photo of the Piraya USV with Ladybug camera and GPS antenna. Right: Local USV coordinate system with nominal heights above the water surface.

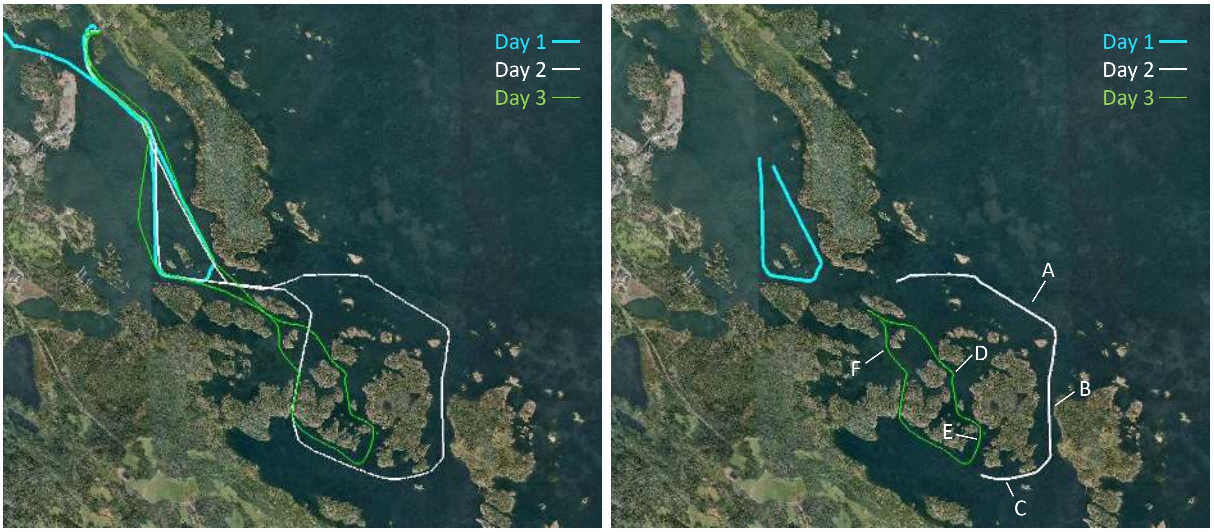

Figure 11: Left: USV trajectory on day 1 (cyan), day 2 (white), and day 3 (green). Right: Sections of the trajectories selected for training and test data from day 1 (cyan), day 2 (white), and day 3 (green). The letters denote positions referred to in the evaluation of the CNNs.

5.2 CNNs for horizon detection and segmentation

Image data from the three trial days were selected such that they originate from three distinct areas, see Figure 11. Due to this selection, the CNNs could be trained on image data from two days/areas and tested on image data from a third day/area. In the evaluation, we first trained the two CNNs on images from days 1 and 2, and tested on images from day 3. Second, we trained on images from days 1 and 3, and tested on images from day 2. From the map in Figure 11, it can be concluded that in the selected sections from days 1 and 2, the USV is surrounded by islands/land in all directions. In the selected section from day 2, one part is outside the islands where the camera will see open water in a large part of the panoramic view. Example images from the test areas are shown below in the evaluation of the two CNNs, see Figures 12 and 14. In these images, it can be seen that a small part of the view (roughly 7°) was occluded by a radio communication antenna on the USV.

The selected sections from each day contain in the order of 6000-7000 images. Image data were grouped as image sequences, each containing 100 consecutive images acquired at 10 fps. We selected about 15 image sequences from each day for training of the CNNs. Target labels for the training data were generated with the methods described in Section 3. The evaluation of the CNNs, and the complete position estimation method, were performed on all images from the respective test areas.

5.2.1 HorizonFinder

Evaluating the HorizonFinder CNN in inference mode on the test images shows that it robustly locates the horizon line in the panoramic image. The test error relative to the target labels is in general less than 0.2° on both the pitch and the roll angles. Figure 12 shows the results for six test images at locations indicated in Figure 11.

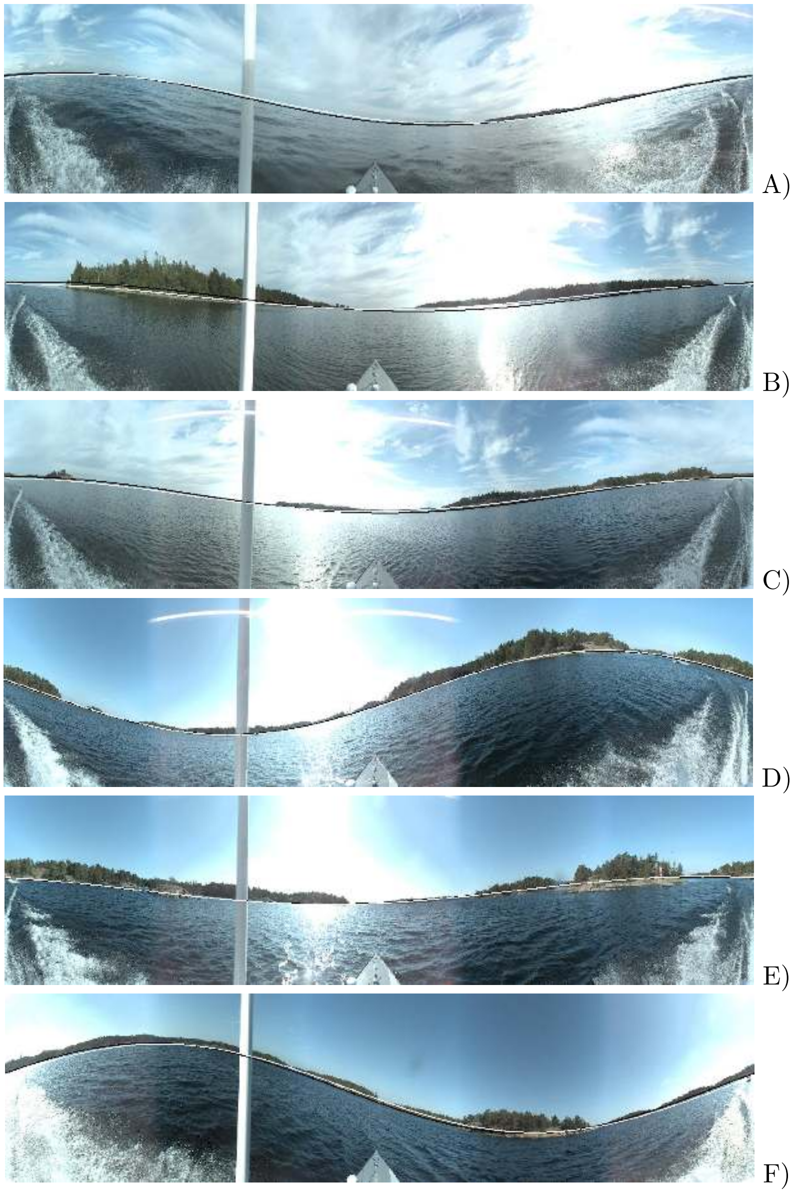

Figure 12: HorizonFinder results: CNN output (white) and target labels (black). HorizonFinder predicts the approximate horizon line well. The letters denote the positions shown in Figure 11.

One observation that we made is that the HorizonFinder network in many cases tends to give a visually better approximation of the horizon line than the target labels generated with Canny detection and Hough voting, see e.g. Figure 12 B) and C). The reason is probably that the network has learned to generalize and average the errors produced in the target label generation. Since we have no ground truth for the camera orientation from other sensors, we cannot provide a quantitative evidence for this observation.

We also noted some failure cases and discrepancies between the results for HorizonFinder and the classical method with Canny detection and Hough voting, see Figure 13. In the first image, Figure 13 a), a large part of the horizon is outside the image and naturally both methods fail to predict the horizon line correctly. In the second image, Figure 13 b), HorizonFinder underestimates the pitch angle. The reason is that in the training data from days 1 and 2, there were no training images with such a large tilt angle (almost 30°). Hence, the CNN will output a too small tilt angle in inference mode. The remedy would be to ensure that the training images would span the complete tilt angle envelope for the USV in test mode. In the third image, Figure 13 c), HorizonFinder robustly detects the horizon line whereas the classical method has been deceived by edges on the water surface.

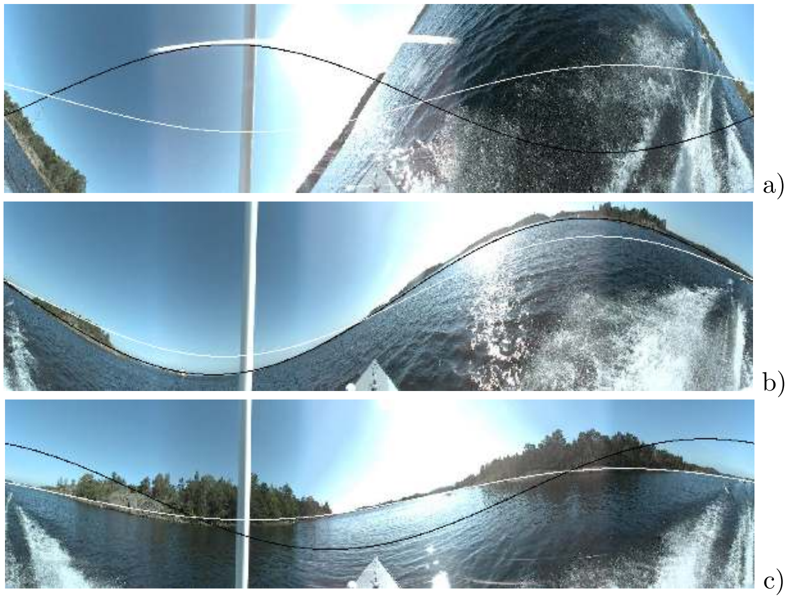

Figure 13: Failure cases and differences between the HorizonFinder output (white) and target labels (black).

Based on the predicted pitch and roll angles from the HorizonFinder CNN, the panoramic images are warped to compensate for the camera orientation and to generate an image from an approximately level camera. The warped image is then fed into the HorizonSegmenter CNN.

5.2.2 HorizonSegmenter

The evaluation of the HorizonSegmenter CNN on the test images shows that the network segments the horizon line well in the warped image, but the segmentation is not perfect. Figure 14 shows the segmentation results for the same six test images as for HorizonFinder. HorizonSegmenter learns to predict the general shape of the horizon line but it does not catch all the high frequency variations. Single trees and other thin structures in the true horizon line are missed. This observation must be kept in mind when using the segmented horizon line in the registration process for position estimation, which is our main goal.

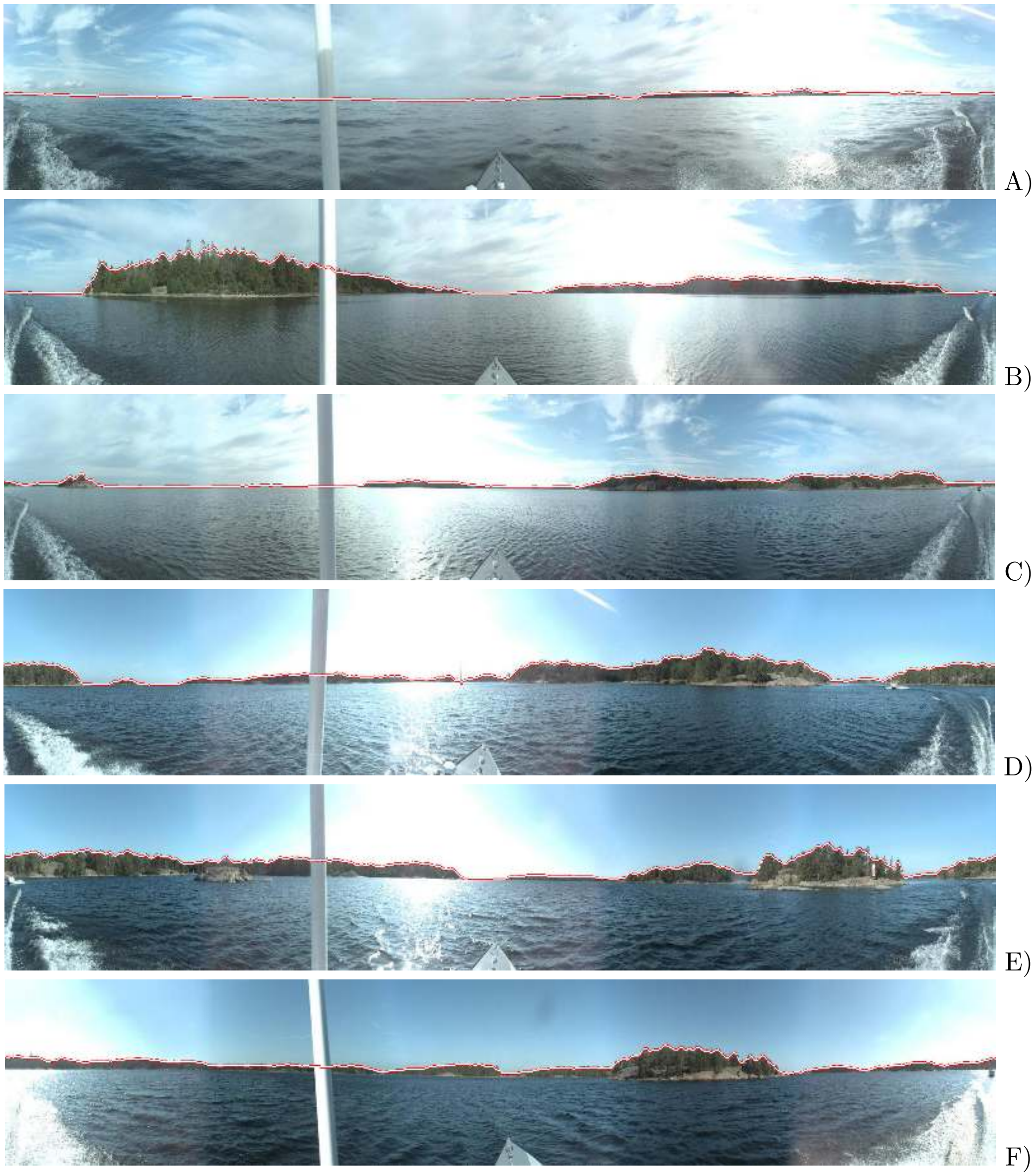

Figure 14: Predicted horizon line by HorizonSegmenter (red/white/red). HorizonSegmenter predicts the general shape of the horizon line well. The letters denote the positions shown in Figure 11.

The results show that the segmentation network can be trained on images from one area and then be applied to images from another area with good results. Although all training and test images are from the same archipelago, the results indicate that the network learns to generalize and segment this type of terrain and does not learn the segmentation of specific images. If the two CNNs, HorizonFinder and HorizonSegmenter, are to be employed in other archipelagos with different types of terrain and texture on the islands, they most likely need to be retrained or finetuned using images from the new area. Moreover, the CNNs need to be trained with images from more than three trial days to work robustly in different weather conditions.

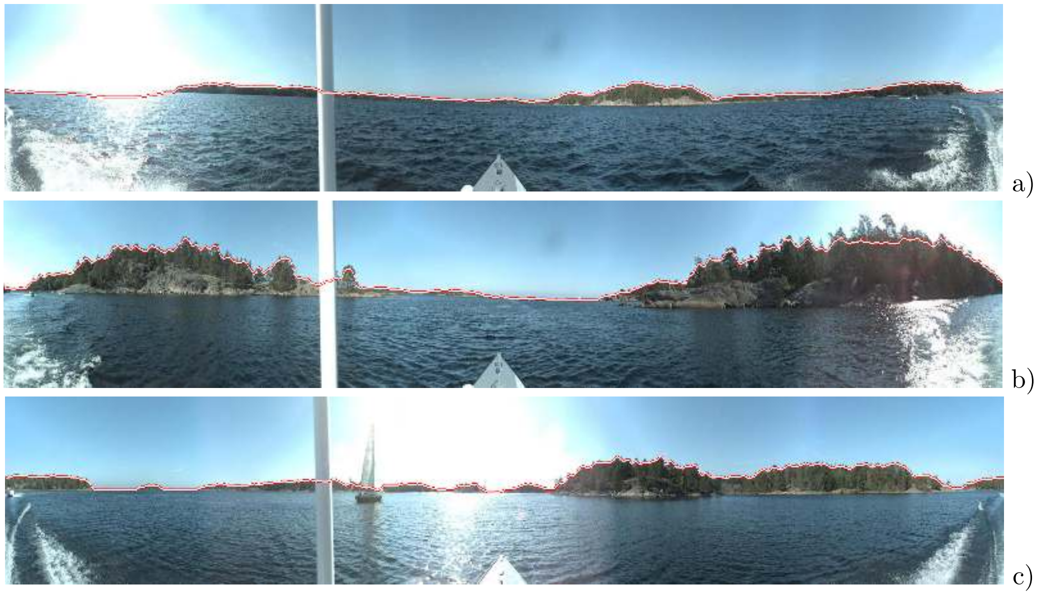

Figure 15: Failure cases in the predicted horizon line (red/white/red). a) The horizon line makes a small bump into the water. b) Individual trees are missed close to the image border. c) The sailing boat is missed.

Some failure cases were noted in the segmentation results, see Figure 15. In the first image, Figure 15 a), the bright sunlight is reflected in the water and there is no distinct transition between the sky and the water. Hence, the predicted horizon line makes a small bump into the water. In the second image, Figure 15 b), the CNN fails to segment individual trees correctly. The effect is more severe close to the image border. One reason is that the training data contained too few images with the horizon line close to the upper image border, and hence the predicted horizon line is pushed down towards the center of the image in inference mode. The mitigation would be to include more training images with the horizon line close to the upper image border. In the third image, Figure 15 c), the sailing boat is missed in the segmentation simply because there were no white sails towards the sky in the training images.

5.3 Position estimates

For position estimation we have evaluated the results achieved with the proposed method (Section 4) and make a comparison with baseline results obtained with registration in the spatial domain (Section 3). Second, we have synthetically decreased the camera FOV to evaluate how the accuracy of the position estimate degrades when limiting the available information on the horizon line. Third, we have evaluated the proposed method at various scales, i.e. we have computed the MOSSE filter correlation score at different grid sizes over the search area.

5.3.1 Proposed method vs classical method

We first assume that we have gained the prior knowledge that the USV is located within a 80m×80m region. We compute the MOSSE filter correlation score every 4m in the X and Y directions within this region. We extract the grid position of the maximum score and perform a second order polynomial fit of the scoring function to interpolate a sub-grid resolution position estimate. To compute the MOSSE filter, we use the segmented horizon lines for 10 consecutive images and correlate it with the geometric horizon computed in the grid points over the search area. The USV travels about 6m during these 10 frames (1s), i.e. slightly more than the grid size. The next position estimate is computed from segmented horizon lines in a new set of 10 consecutive images, i.e. there is no overlap between the sets of 10 images used to generate the position estimates. Typical results for the correlation score are illustrated in Figure 16.

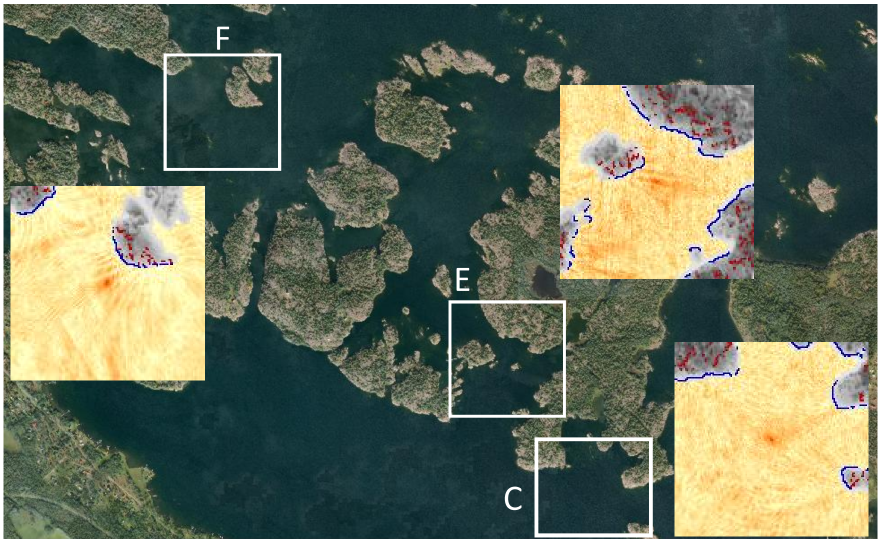

Figure 16: The MOSSE filter correlation score (orange) in the search region for images at positions C, E and F in Figure 11 overlaid on map. The geometric horizon lines are displayed in red and the water lines in blue (as seen from the true position). The search region is extended to 500m×500m for better illustration. The highest correlation score is clearly obtained for the true position in the center of the image.

The position estimates obtained with the proposed method are shown on the map in Figure 17 together with the measured GPS position. The estimated position is the average position of the USV while capturing 10 images. The GPS position measurements have been compensated for the GPS antenna position relative to the camera position, such that they correspond to having GPS mesurements from the camera position.

The average deviation between the position estimates with our proposed method and the GPS measurements is 3.69±2.32m and 2.72±1.58m, respectively, for the test images from days 2 and 3. Assuming uncorrelated position measurements, these deviations imply that our method provides position measurements with GPS-level accuracy, according to e.g. (William J. Hughes Technical Center, 2014). See the Appendix for calculations. The position estimates for the proposed method are highly accurate for test images from day 3 where the USV is surrounded by islands in all directions. The average position estimate accuracy is slightly lower for test images from day 2. The explanation for the small degradation is readlily found looking at the segmented image in Figure 14A. Almost one half of the image is facing open water, where there is no variation in the horizon line usable for registration with the DEM. Still, the horizon line variation in the remaining half of the image is sufficient for our method to provide accurate position estimates.

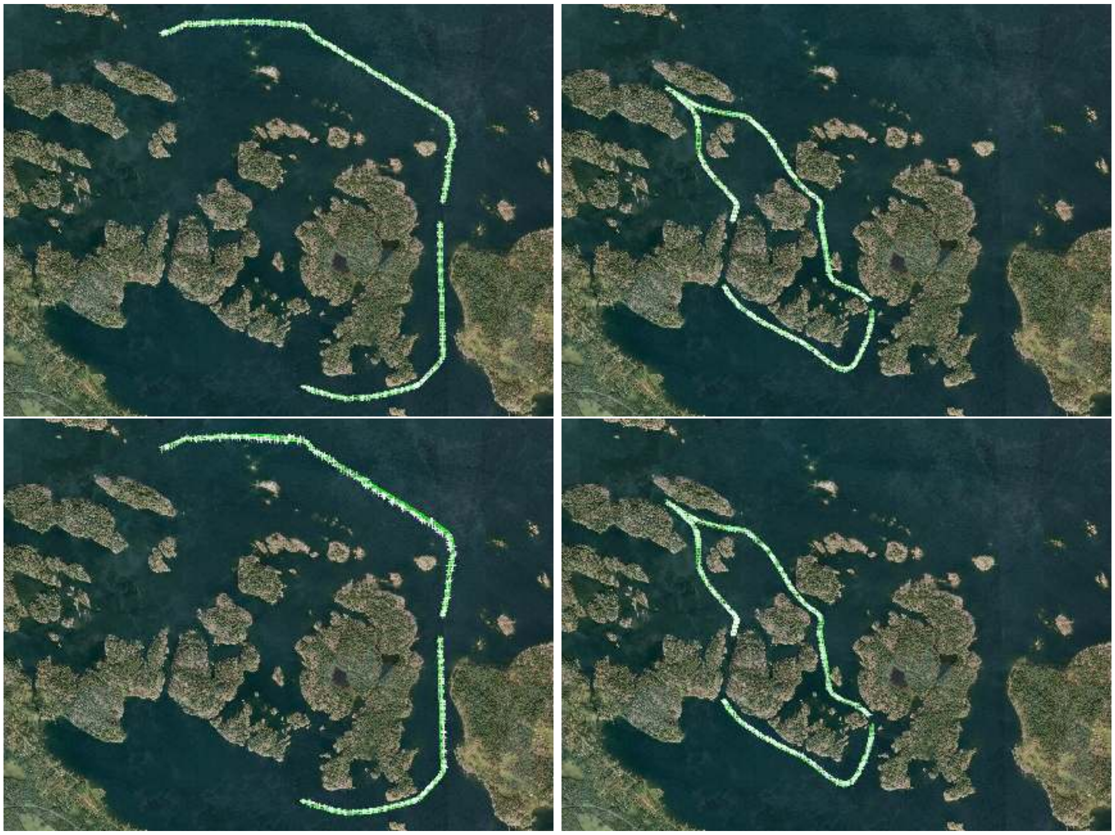

Figure 17: Position estimation results: GPS (green), proposed method (white). MOSSE filter registration (top) and spatial registration (bottom). Test day 2 (left) and test day 3 (right).

The position estimates in the proposed method are refined using a second order polynomial fit of the correlation scores over the XY grid. The accuracy of the position estimates obtained without this refinement is 3.90±2.46m and 3.07±1.72m, respectively, for the test images from days 2 and 3, see Table 1. The average degradation due to quantization effects is 0.2-0.3m, and it is statistically significant.

The key contribution in our proposed method is the adaptation of the MOSSE filter to perform the registration of the horizon line with the DEM data in the Fourier domain. For comparison, we perform the registration in the spatial domain, where we use the method described in Section 3. For each position grid point, we record the lowest summed distance according to (1). We detect the location of the lowest distance point over the grid and make the same sub-grid resolution interpolation as for the proposed method. The average position error obtained with spatial registration is 10.41±6.89m and 3.98±2.27m, respectively, for the two test days/areas. A t-test at 95% significance level shows that the position errors achieved with the proposed method for both test days are significantly lower than those obtained with the spatial registration. The position errors obtained, which are listed in Table 1, are averages over 650 position estimates for each test day.

We also performed registration in the spatial domain when the image was warped with the pitch and roll angles estimated by the CNN HorizonFinder. The average position error obtained with the spatial registration is 10.62±6.98m and 4.07±2.12m, respectively, for the two test days/areas. The position errors obtained are very similar to those obtained with the spatial registration where the image was warped with the pitch and roll angles estimated by the Hough voting, see Table 1. These results further ensure that it is the spatial registration that degrades the position accuracy compared to the MOSSE filter registration.

Table 1: Position estimation accuracy achieved with the MOSSE filter registration and with the spatial registration. The MOSSE filter registration in the Fourier domain outperforms the classical spatial registration.

Registration method	Test day	Mean error (m)	Std (m)	Median error (m)	Test day	Mean error (m)	Std (m)	Median error (m)
MOSSE	2	**3.69**	**2.32**	**3.27**	3	**2.72**	**1.58**	**2.48**
MOSSE - quantized	2	3.90	2.46	3.48	3	3.07	1.72	2.81
Spatial - classic	2	10.41	6.89	8.10	3	3.98	2.27	3.62
Spatial - CNN	2	10.62	6.98	8.33	3	4.07	2.12	3.89

The Fourier method is found to be significantly more robust to deviations between the horizon line in the image and the geometric horizon line from the DEM. As noted, the segmented horizon line from the CNN does not catch all high frequency variations in the image. Furthermore, not even a high-resolution DEM will be a perfect representation of the real world. These discrepancies between the image and the DEM are basically inevitable, and for the Fourier registration, the lower frequency content is sufficient to produce highly accurate position estimates whereas the discrepancies distract the spatial registration more severly.

The difference in position accuracy between the two methods is more pronounced for the test images from day 2 where the camera sees a lot of open water and there is less information in the panoramic image to base the position estimate upon. Small spatial errors in the detected horizon line in the image compared to the geometric horizon line from the DEM generate large position errors with the classical spatial method, whereas the proposed method is significantly more robust to these small spatial deviations. The average position error for the proposed method is 3.69m and increases to 10.41m for the method with spatial registration.

One difference between the MOSSE filter correlation method and the spatial method is that the former method uses a sequence of images to compute the position estimate, whereas the spatial method is based on registration of a single image. The position errors for the spatial method could likely be reduced by averaging the position estimates over an image sequence. However, as stated in section 3, the spatial method is already prohibitively slow for real-time applications and adding this averaging over several images would make it even slower.

Missing data. For test day 2, there are some position estimates missing on the right hand side of the trajectory. The reason for the missing data is a short period with frame drops in the data acquisition system. For test day 3, there are also some position estimates missing on the right hand side of the trajectory. This is caused by very large maneouvres (large tilt angles) of the USV where the horizon line is outside the image border and no position estimates could be obtained, see Figure 13a). Further, there are no position estimates provided in the narrow straight. The width of the straight is about 20-30m. The altitude above sea level is 20-25m on both sides of the straight, corresponding to an elevation angle larger than 45° as seen from the vessel. Hence, the true horizon line is outside the image border on a large part of the image and no position estimate can be provided with the proposed method given the camera's vertical FOV.

5.3.2 Proposed method with varying FOV

In our field trial we had access to a camera with a full 360° panoramic FOV. In many applications, images are acquired with a camera having a considerably smaller FOV. This raises the question, how accurately can the USV position be estimated with the proposed method with a narrower FOV?

To investigate how the position estimation accuracy varies with the available FOV, we synthetically decrease the camera FOV to 270°, 180°, and 120°, respectively. We do this by retaining the segmented horizon line in a sector centered in the forward camera direction and setting the horizon line to zero outside this sector.

In this manner, we keep the same number of image points in the Fourier transform and the MOSSE signal output is independent of the FOV used. Retaining the horizon line within the FOV and setting it to zero outside corresponds to having a rectangular window function. We first experimented with a rectangular window but it resulted in severe side lobe effects in the MOSSE signal output, generating large position errors. To alleviate the side lobe effects, we employ a Hamming window over the available FOV and set the horizon line to zero outside this sector.

The position estimates obtained with the various FOVs are shown in Figure 18 and they are listed in Table 2. We have also included results when introducing the Hamming window over the full 360° FOV to see the effect of the pure window function on the position estimates. The window was centered in the forward direction.

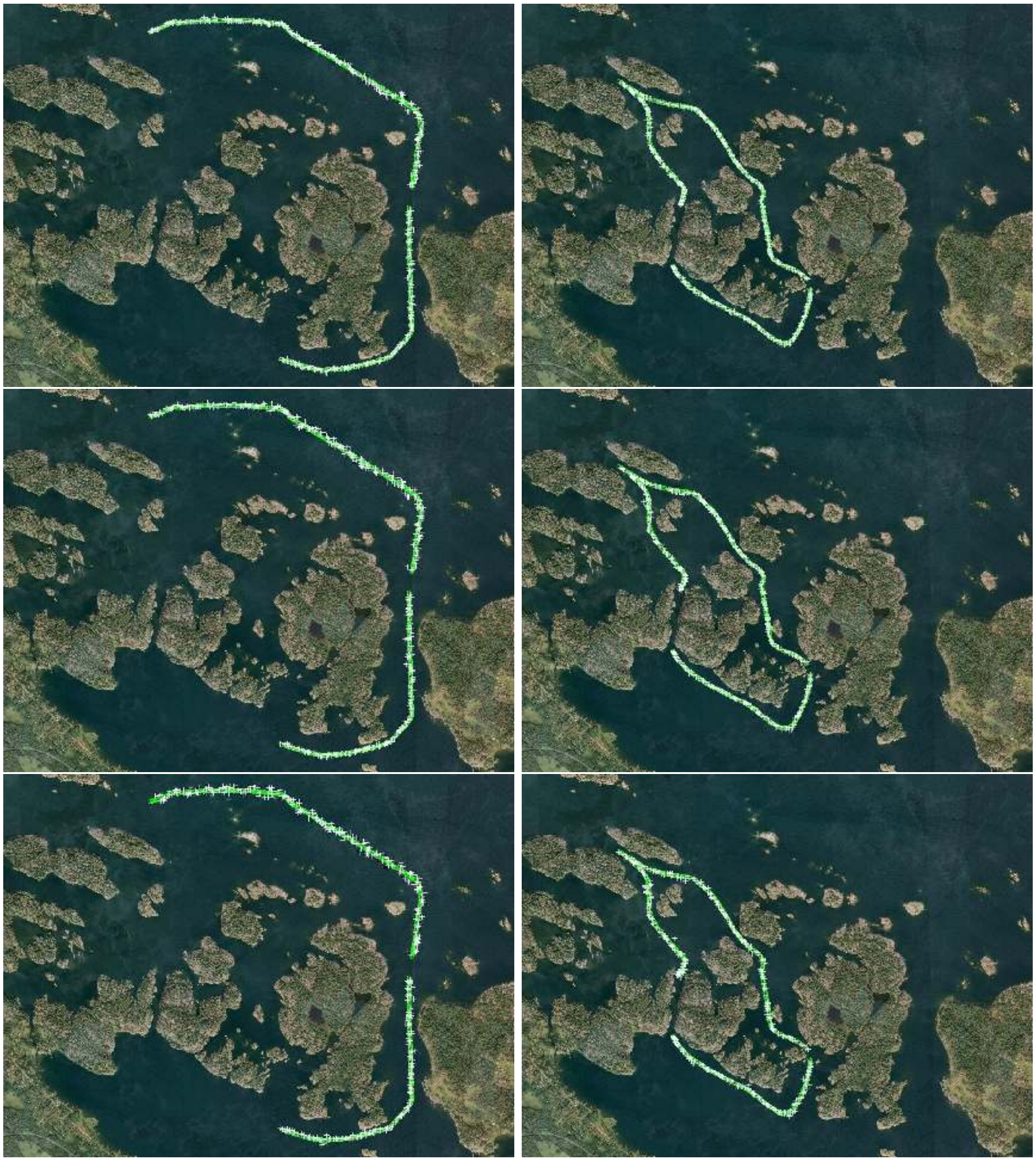

Figure 18: Position estimation results with proposed method (white) and GPS (green). Synthetic FOV 270° (top), 180° (middle), and 120° (bottom). Test day 2 (left) and test day 3 (right). The position estimation accuracy degrades with a narrower FOV as expected.

As expected, the position estimate accuracy continuously degrades when decreasing the available FOV, and the variance of the position estimates increases. The reason for the degradation is simply that there is less information to base the position estimates upon. Also, for narrower FOVs some obvious outliers in the position estimates start to appear. These outliers could easily be detected and suppressed if the position estimates were filtered over time taking a USV motion model into account. Now the presented position estimates are viewed as completely individual measurements. In such a position filter, the maximum MOSSE

Table 2: Position estimation accuracy achieved with the proposed method with various FOV. The ”w” in the FOV column denotes the Hamming window.

FOV (°)	Test day	Mean error (m)	Std (m)	Median error (m)	Test day	Mean error (m)	Std (m)	Median error (m)
360	2	3.69	2.32	3.27	3	2.72	1.58	2.48
360w	2	6.55	6.75	4.67	3	3.73	2.50	3.09
270w	2	8.80	8.62	5.76	3	4.55	3.51	3.63
180w	2	12.22	10.33	8.82	3	6.43	6.13	4.72
120w	2	18.96	13.49	15.36	3	10.67	9.85	7.47

filter correlation score over the search region could be used as a confidence measure of the position estimate.

The degradation with a narrower FOV is more pronounced for the test images from day 2, especially in the area outside the islands where the camera sees a lot of open sea. This larger degradation is natural since the remaining FOV will contain a larger fraction of level, i.e. uninformative horizon line. It has the same effect as a further reduction in FOV. This observation is expected and also goes hand in hand with when accurate position estimates are needed. When navigating within the archipelago, surrounded by islands, a high position accuracy is required for safe navigation and can be provided by the proposed method. At more open water, the position accuracy degrades, but the requirement on the position accuracy for safe navigation also decreases.

5.3.3 Proposed method at coarser scale

Our proposed method requires that an approximate position is known to limit the search area and to make the method real-time capable, which is necessary if it is to be used as a sensor for navigation of the USV. Without limiting the search area, our method is slightly below real-time speed, see section 5.4. The approximate position could e.g. be obtained from continuous position tracking over time using the proposed method from mission start, GPS measurements (when available and reliable), cross bearing measurements based on detection of seamarks/landmarks, a large scale position estimation method as in (Baatz et al., 2012), or position estimates from the proposed method at a coarser scale.

We investigated the accuracy of the proposed method at a coarser scale with a grid size of 8m, 16m and 32m respectively. Since we need the USV to travel about the same distance as the grid size, we use sequences of 20, 40 and 80 images when computing the MOSSE filter. If the image sequence is too short, the actual MOSSE correlation score peak may fall between grid points. Aggregating information from a long image sequence, on the other hand, may imply that the Fourier spectrum of the segmented horizon lines varies too much over the sequence and does not generate a distinct correlation peak.

We compute the MOSSE filter correlation score over a 21×21 point grid, corresponding to a search area of 160m×160m, 320m×320m and 640m×640m for the three grid sizes. We make sure that the measured GPS position is deliberately set *between* grid points in the search area. Employing this worst-case scenario in the evaluation, we ensure that the MOSSE correlation peak would generate a high score irrespective of how the grid pattern is located relative to the true position. The position estimate results obtained when varying the grid size are shown in Figure 19 and they are listed in Table 3. Note that the values in the table are averages after removal of outliers as described below.

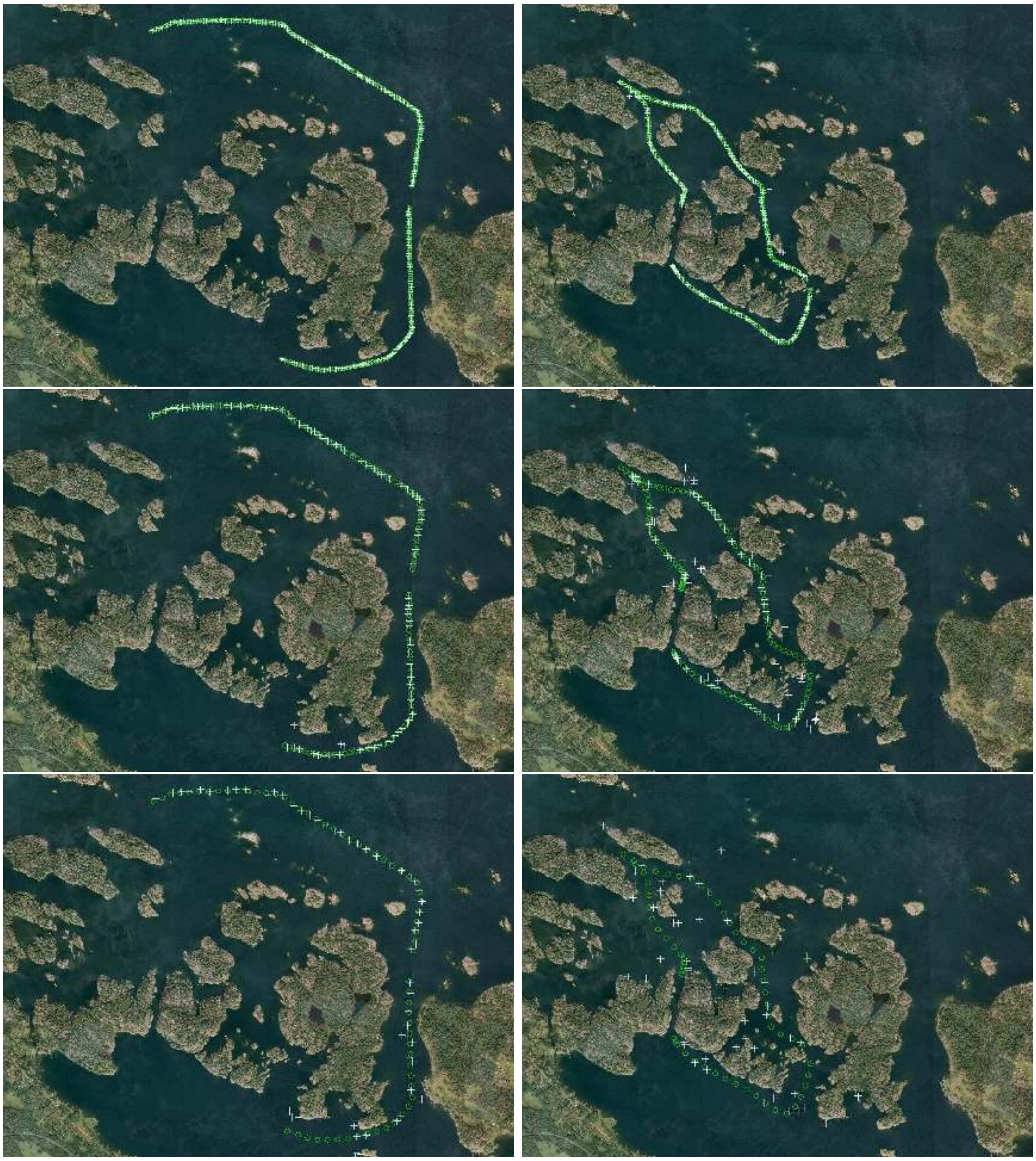

Figure 19: Position estimation results: GPS (green), proposed method (white). 8m grid (top), 16m grid (middle), 32m grid (bottom). Test day 2 (left) and test day 3 (right).

Table 3: Position estimation accuracy achieved with the proposed method with various grid sizes. Mean errors are given for points after the removal of outliers.

Grid size (m)	Test day	Mean error (m)	Std (m)	Median error (m)	Test day	Mean error (m)	Std (m)	Median error (m)
4	2	3.69	2.32	3.27	3	2.72	1.58	2.48
8	2	4.28	2.05	4.06	3	4.22	1.64	4.18
16	2	7.05	2.91	7.40	3	8.63	3.21	7.46
32	2	16.56	6.42	17.11	3	26.62	14.89	23.33

When increasing the grid size to 8m, the position estimate accuracy is only degraded marginally for the test images from day 2, and slightly more for the test images from day 3. For the latter images, there are a few outliers. These could easily be detected and removed if filtering the position estimates over time. Note that the outliers occur where there is a curve on the USV trajectory and the spectrum of the horizon line varies significantly over the image sequence, thus not creating a distinct peak in the MOSSE filter correlation response.

Increasing the grid size further to 16m, the position estimates for the test images from day 2 are degraded considerably but still judged to be useful for navigation if filtering the position estimates over time. For the test images from day 3, there are some more outliers (around 3% of the points) but the position estimates would be useful for navigation after filtering. The outliers could also be detected from a lower value on the maximum correlation score over the search area.

For a grid size of 32m, the position estimates for the test images from day 2 in rather open water are still useful but significantly degraded. For the test images captured closer to land, i.e. less than 100m to the shore, the grid size of 32m does not generate any useful results. The position estimates are very unreliable and filtering the position estimates over time does not help. The same observation applies to the test images from day 3. There are some decent position estimates but they are outnumbered by the erroneous estimates. The reason is that the distance to the shore is too small compared to the travelled distance while capturing the image sequence.

5.3.4 Proposed method in a scale pyramid

The results in the previous section show that a larger grid size could be used to first find a coarse position estimate, and then decrease the grid size to refine the position estimate in a scale pyramid. We start with a grid size of 16m as the first scale, since this was the coarsest scale that in general gave good position estimates. We use the position estimates obtained with the 16m grid as the center points of the search area at the next scale with an 8m grid. The new position estimates are then used as center points for the search areas at the finest scale with 4m grid. The position estimate results obtained at the 4m grid are shown in Figure 20 and they are shown in Table 4.

The accuracy of the final position estimates at the 4m grid are essentially the same irrespective if the method is applied directly at the 4m grid or in a scale pyramid over the 16m, 8m and 4m grids. The only difference is that with the scale pyramid, some outliers (around 3%) with very large position errors are generated. These outliers could be detected and suppressed either from the low maximum correlation score and/or by filtering the position estimates over time.

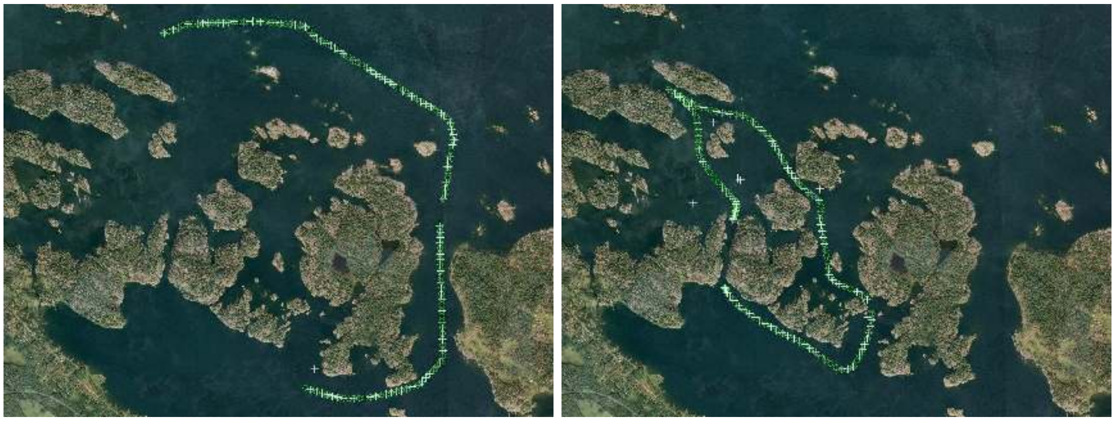

Figure 20: Position estimation results with scale pyramid from 16m grid to 4m grid; GPS (green), proposed method(white). Test day 2 (left) and test day 3 (right).

Table 4: Position estimation accuracy achieved with the proposed method with 4m grid size, direct search and with scale pyramid search. Mean errors are given for points after the removal of outliers.

Grid size (m)	Test day	Mean error (m)	Std (m)	Median error (m)	Test day	Mean error (m)	Std (m)	Median error (m)
4	2	3.69	2.32	3.27	3	2.72	1.58	2.48
16-8-4	2	3.67	2.31	3.27	3	2.76	1.49	2.62

5.4 Computational speed

Our proposed method is implemented in Python and the PyTorch framework. Average runtimes of the main algorithmic steps are listed in Table 5. The listed runtimes are obtained with a standard desktop computer with an Intel Core i7-6700 @3.40 GHz CPU, and a GeForce GTX 1080 Ti GPU card. The inference of the two CNNs and the ray tracing to compute the geometric horizon are run on GPU. The remainder of the algorithms is run on the CPU.

The Ladybug camera frame rate is 10 fps and our proposed method uses 10 images captured during 1s to generate a position estimate. This means that the two CNNs can be run in real time in inference mode.

Table 5: The runtimes of the main algorithm steps in the proposed method and in the classical method.

Algorithm step	Average runtime	Algorithm step	Average runtime
HorizonFinder	15ms per frame	Canny + Hough	300ms per frame
HorizonSegmenter	60ms per frame	Canny + Distance func.	200ms per frame
MOSSE correlation	25ms per grid point	Spatial registration	1.2s per grid point

Assuming a search area of 24m×24m with a 4m grid size, i.e. 49 grid points, the runtime of the MOSSE correlation score is about 1.2s. This gives a total runtime around 2.0s, i.e. about a factor of two from a real-time implementation. This shows that our proposed method is real-time capable with relatively minor runtime optimization changes to the code. Runtimes for the spatial registration are achieved with a grid search over 17×17 pitch and roll angles and 21 yaw angles as described in section 3.

6 Conclusions

In this paper, we have investigated the problem of providing accurate position measurements for a USV operating in coastal areas or in the archipelago when GPS signals are denied or failing. We propose a position estimation method where the horizon line is extracted in a 360° panoramic image around the USV. We design two CNN architectures. The first CNN determines an approximate horizon line in the image and implicitly determines the camera orientation. The image is warped to compensate for the camera orientation. The second CNN then extracts the pixelwise horizon line and water line in the warped image. Consequently, the extracted horizon line is correlated with DEM data in the Fourier domain using a MOSSE correlation filter. Finally, we determine the location of the maximum correlation score over the search area to estimate the position of the USV. Our approach provides promising results by achieving position estimates with GPS-level accuracy in previously unvisited test areas.

We have shown that the proposed method with registration of the horizon line in the Fourier domain outperforms a state-of-the-art classical method with registration in the spatial domain, both in terms of position accuracy and computational speed. To reduce the search time, the proposed method can first be used at a coarser grid to generate a slightly less accurate position estimate, and then refine the position estimate using a finer grid for the next image sequence. Experiments show that the position estimate accuracy degrades gracefully with the available camera field of view. The degradation is scene dependent and is more severe over open water where there is less information in the horizon line to base the position estimate upon. The observation that horizon line registration is sufficient to provide very accurate position estimates suggests that our proposed method can be generalized to other domains and also be employed by

UGVs and UAVs operating on land.

Acknowledgments

This work was partially supported by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation.

This work was supported by the Swedish Foundation for Strategic Research (Smart Systems: RIT 15-0097).

This research is supported by CENIIT grant (18.14), and VR starting grant (2016-05543).

Appendix

Statistics for GPS measurement accuracy over time can be found in (William J. Hughes Technical Center, 2014). The histogram of the horizontal position error is well approximated with a Rayleigh distribution where 95% of the measurements have a position error smaller than 3.35m. This error is the average over 28 test sites and varies considerably between different sites. We make the assumption that the GPS measurements in the X direction (North-South) and Y direction (East-West), denoted $\mathrm{X_{GPS}}$ and $\mathrm{Y_{GPS}}$, are uncorrelated and from a normal distribution with zero mean and standard deviation σ_{GPS}.

The cumulative distribution function (cdf) and the mean value of a Rayleigh distribution are given by

$$\mathrm{cdf}(x, \sigma) = 1 - e^{x^2/(2\sigma^2)} \tag{7a}$$

$$\mathrm{Mean}(\sigma) = \sigma\sqrt{\frac{\pi}{2}}\ . \tag{7b}$$

Letting cdf $= 0.95$ and $x = 3.35$ for the GPS position measurements gives $\sigma_{\mathrm{GPS}} = 1.37$m.

We make the assumption that the position estimates from our proposed method are uncorrelated with the GPS measurements. Further, we assume that the position estimates in the X and Y directions, denoted $\mathrm{X_{MOSSE}}$ and $\mathrm{Y_{MOSSE}}$, are uncorrelated and from a normal distribution with zero mean and standard deviation σ_{MOSSE}.

We measure the position accuracy as the deviation between the position estimates from our proposed method and the GPS measurements, i.e. we measure the error

$$E = \sqrt{E_{\mathrm{X}}^2 + E_{\mathrm{Y}}^2}\ , \tag{8}$$

where

$$E_{\mathrm{X}} = \mathrm{X}_{\mathrm{MOSSE}} - \mathrm{X}_{\mathrm{GPS}} \tag{9a}$$

$$E_{\mathrm{Y}} = \mathrm{Y}_{\mathrm{MOSSE}} - \mathrm{Y}_{\mathrm{GPS}}\ . \tag{9b}$$

Now we make the assumption that our proposed method has GPS-level accuracy, i.e. the variance of the two position measurement methods are the same. Letting $\sigma_{\mathrm{MOSSE}} = \sigma_{\mathrm{GPS}}$, we obtain the variances

$$\mathrm{VAR}(E_{\mathrm{X}}) = \mathrm{VAR}(\mathrm{X}_{\mathrm{MOSSE}}) + \mathrm{VAR}(\mathrm{X}_{\mathrm{GPS}}) = 2\sigma_{\mathrm{GPS}}^2 \tag{10a}$$

$$\mathrm{VAR}(E_{\mathrm{Y}}) = \mathrm{VAR}(\mathrm{Y}_{\mathrm{MOSSE}}) + \mathrm{VAR}(\mathrm{Y}_{\mathrm{GPS}}) = 2\sigma_{\mathrm{GPS}}^2\ . \tag{10b}$$

If all our assumptions are valid, combining (7) and (10) will yield the expected mean value of the error E in (8) as

$$\mathrm{Mean}(E) = \sqrt{2\sigma_{\mathrm{GPS}}^2}\sqrt{\frac{\pi}{2}} = \sigma_{\mathrm{GPS}}\sqrt{\pi} \tag{11}$$

Inserting $\sigma_{\mathrm{GPS}} = 1.37$m gives Mean$(E) = 2.42$m. In the evaluation of our proposed method, we achieve an average deviation of 2.72m from the GPS measurements, which is close to the expected mean value given the assumptions. This result suggests that our proposed method provides position estimates with GPS-level accuracy.

References

Ahmad, T., Campr, P., Čadik, M., and Bebis, G. (2017). Comparison of semantic segmentation approaches for horizon/sky line detection. In *Neural Networks (IJCNN), 2017 International Joint Conference on*, pages 4436–4443. IEEE.

Airmar GH2183 specification (2019). `http://www.airmar.com/uploads/brochures/gh2183.pdf`.

Arandjelovic, R., Gronat, P., Torii, A., Pajdla, T., and Sivic, J. (2016). Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307.

Baatz, G., Saurer, O., Köser, K., and Pollefeys, M. (2012). Large scale visual geo-localization of images in mountainous terrain. In *Computer Vision–ECCV 2012*, pages 517–530. Springer.

Baboud, L., Čadík, M., Eisemann, E., and Seidel, H.-P. (2011). Automatic photo-to-terrain alignment for the annotation of mountain pictures. In *CVPR 2011*, pages 41–48. IEEE.

Bolme, D. S., Beveridge, J. R., Draper, B. A., and Lui, Y. M. (2010). Visual object tracking using adaptive correlation filters. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 2544–2550. IEEE.

Brejcha, J. and Čadík, M. (2017). State-of-the-art in visual geo-localization. *Pattern Analysis and Applications*, 20(3):613–637.

Brejcha, J. and Cadík, M. (2018). Camera orientation estimation in natural scenes using semantic cues. In *2018 International Conference on 3D Vision (3DV)*, pages 208–217. IEEE.

Canny, J. (1986). A computational approach to edge detection. *PAMI*, 8:679–698.

Chiodini, S., Pertile, M., Debei, S., Bramante, L., Ferrentino, E., Villa, A. G., Musso, I., and Barrera, M. (2017). Mars rovers localization by matching local horizon to surface digital elevation models. In *Metrology for AeroSpace (MetroAeroSpace), 2017 IEEE International Workshop on*, pages 374–379. IEEE.

Chum, O., Perd'och, M., and Matas, J. (2009). Geometric min-hashing: Finding a (thick) needle in a haystack. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 17–24. IEEE.

Dumble, S. J. and Gibbens, P. W. (2015). Efficient terrain-aided visual horizon based attitude estimation and localization. *Journal of Intelligent & Robotic Systems*, 78(2):205–221.

Felsberg, M. and Hedborg, J. (2007). Real-time visual recognition of objects and scenes using p-channel matching. In *Scandinavian Conference on Image Analysis*, pages 908–917. Springer.

FLIR Ladybug3 specification (2019). `https://www.ptgrey.com/ladybug3-360-degree-firewire-spherical-camera-systems`.

Grelsson, B., Felsberg, M., and Isaksson, F. (2016). Highly accurate attitude estimation via horizon detection. *Journal of Field Robotics*, 33(7):967–993.

Grelsson, B., Robinson, A., Felsberg, M., and Khan, F. (2018). HorizonNet for visual terrain navigation. In *Accepted at IEEE Conference on Image Processing, Applications and Systems*.

Han, J., Park, J., Kim, J., and Son, N.-s. (2016). Gps-less coastal navigation using marine radar for usv operation. *IFAC-PapersOnLine*, 49(23):598–603.

He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

Hough, P. (1962). Method and means for recognizing complex patterns. *U.S. Patent 3069654*.

Irschara, A., Zach, C., Frahm, J.-M., and Bischof, H. (2009). From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2599–2606. IEEE.

Javidi, B. and Wang, J. (1994). Design of filters to detect a noisy target in nonoverlapping background noise. *JOSA A*, 11(10):2604–2612.

Kendall, A., Grimes, M., and Cipolla, R. (2015). Posenet: A convolutional network for real-time 6-dof camera relocalization. In *Computer Vision (ICCV), 2015 IEEE International Conference on*, pages 2938–2946. IEEE.

Kristan, M., Leonardis, A., Matas, J., Felsberg, M., Pflugfelder, R., Zajc, L. C., Vojir, T., Häger, G., Lukezic, A., Eldesokey, A., Fernandez, G., and et al. (2017). The Visual Object Tracking VOT2017 challenge results. In *2017 IEEE INTERNATIONAL CONFERENCE ON COMPUTER VISION WORKSHOPS (ICCVW 2017)*, IEEE International Conference on Computer Vision Workshops, pages 1949–1972. IEEE.

Lee, J.-T., Kim, H.-U., Lee, C., and Kim, C.-S. (2017). Semantic line detection and its applications. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3229–3237.

Li, Y., Snavely, N., Huttenlocher, D., and Fua, P. (2012). Worldwide pose estimation using 3d point clouds. In *European conference on computer vision*, pages 15–29. Springer.

Maas, A. L., Hannun, A. Y., and Ng, A. Y. (2013). Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, volume 30.

Melo, J. and Matos, A. (2017). Survey on advances on terrain based navigation for autonomous underwater vehicles. *Ocean Engineering*, 139:250–264.

Meneghetti, G., Danelljan, M., Felsberg, M., and Nordberg, K. (2015). Image alignment for panorama stitching in sparsely structured environments. In *Scandinavian Conference on Image Analysis*, pages 428–439. Springer.

Nugraha, B. T., Su, S.-F., et al. (2017). Towards self-driving car using convolutional neural network and road lane detector. In *Automation, Cognitive Science, Optics, Micro Electro-Mechanical System, and Information Technology (ICACOMIT), 2017 2nd International Conference on*, pages 65–69. IEEE.

Piasco, N., Sidibé, D., Demonceaux, C., and Gouet-Brunet, V. (2018). A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109.

Porzi, L., Rota Bulò, S., and Ricci, E. (2016). A deeply-supervised deconvolutional network for horizon line detection. In *Proceedings of the 2016 ACM on Multimedia Conference*, pages 137–141. ACM.

Radovic, M., Adarkwa, O., and Wang, Q. (2017). Object recognition in aerial images using convolutional neural networks. *Journal of Imaging*, 3(2):21.

Ramalingam, S., Bouaziz, S., Sturm, P., and Brand, M. (2009). Geolocalization using skylines from omni-images. In *2009 IEEE 12th International Conference on Computer Vision Workshops, ICCV Workshops*, pages 23–30. IEEE.

Sattler, T., Leibe, B., and Kobbelt, L. (2011). Fast image-based localization using direct 2d-to-3d matching. In *Computer Vision (ICCV), 2011 IEEE International Conference on*, pages 667–674. IEEE.

Sattler, T., Torii, A., Sivic, J., Pollefeys, M., Taira, H., Okutomi, M., and Pajdla, T. (2017). Are large-scale 3d models really necessary for accurate visual localization? In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6175–6184. IEEE.

SRTM (2019). `http://www2.jpl.nasa.gov/srtm/`.

Tzeng, E., Zhai, A., Clements, M., Townshend, R., and Zakhor, A. (2013). User-driven geolocation of untagged desert imagery using digital elevation models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 237–244.

U-blox EVK-8 specification (2019). `https://www.u-blox.com/en/product/evk-8evk-m8`.

Vaman, D. (2012). Trn history, trends and the unused potential. In *Digital Avionics Systems Conference (DASC), 2012 IEEE/AIAA 31st*, pages 1A3–1. IEEE.

Verbickas, R. and Whitehead, A. (2014). Sky and ground detection using convolutional neural networks. In *Proceedings of the International Conference on Machine Vision and Machine Learning*, volume 64.

Weyand, T., Kostrikov, I., and Philbin, J. (2016). Planet-photo geolocation with convolutional neural networks. In *European Conference on Computer Vision*, pages 37–55. Springer.

William J. Hughes Technical Center (July 31, 2014). Global Positioning System (GPS) Standard Positioning Service (SPS) Performance Analysis Report. *Report No. 86*.

Woo, J., Son, K., Li, T., Kim, G. S., and Kweon, I.-S. (2007). Vision-based uav navigation in mountain area. In *MVA*, pages 236–239.

Wu, Y., Tang, F., and Li, H. (2018). Image-based camera localization: an overview. *Visual Computing for Industry, Biomedicine, and Art*, 1(1):8.

Xu, Q., Zhang, C., and Zhang, L. (2017). Deep convolutional neural network based unmanned surface vehicle maneuvering. In *Chinese Automation Congress (CAC), 2017*, pages 878–881. IEEE.