Fusion of Passive and Active Electro-Optical Sensor Data for Enhanced Scene Understanding in Challenging Conditions

Egil Bae

Norwegian Defence Research Establishment (FFI), P.O. Box 25, 2027 Kjeller, Norway

ABSTRACT

Passive electro-optical systems, such as visible and infrared light cameras, and active systems, such as ladar or LiDAR, can acquire detailed two- and three-dimensional images of a scene. This paper presents a sensor fusion framework that combines passive and active electro-optical sensor data to reveal subtle patterns. It creates fused data structures that merge 3D coordinates and light intensities, and customizes recent methods for classification of high-dimensional irregular data to segment the fused structures into different object classes. The framework can also be used to discriminate weak laser return pulses from noise by extracting point clusters in the higher-dimensional data under certain constraints. Methods for estimating and compensating for motion during acquisition of the data can be integrated in the framework to prevent misalignments. Experiments on IR images and 3D point clouds acquired by a ladar demonstrate scene segmentation, object recognition and motion estimation in various challenging conditions.

Keywords: sensor fusion, segmentation, scene analysis, object recognition, point clouds, ladar, infrared imaging

1. INTRODUCTION

Electro-optical systems can be divided into those that are passive and those that are active. Passive systems include visible and infrared light cameras, and various multi- and hyperspectral variants that record the scene in many different wavelengths at the same time. Sensors in the visible spectrum are mature and have reached widespread adoption, but have limited performance in low-light environments. Sensors in the infrared spectrum can detect self-emitted radiation of objects in the scene and thus have a more robust all-around performance, especially the longer wavelength variants. They still rely on some temperature contrasts in the scene to produce meaningful images, which may not always be present for arbitrary or intentional reasons. In general, the performance of infrared sensors in terms of angular and temporal resolution (framerate) is not competitive with visible light senors. This is due to a number of reasons, such as worse performing semi-conducting material suitable for infrared sensors, and that the diffraction limit is proportional to the wavelength of light.

Active systems, in contrast, emit their own light and detect the reflected light from objects in the scene. This makes them independent from both external light sources and temperature contrasts in the scene. It is of course necessary that the scene objects reflect some of the light towards the detector. This is generally the case for material with some diffuse component of reflection, excluding perfect mirrors. Lidar (light detection and ranging) and ladar (laser detection and ranging) are terms for active systems that are sometimes used interchangably. One of the benefits of a lidar/ladar is that they can produce a 3D image of the scene. The arrival time of a return pulse can be used to calculate the distance to the point of reflection, which together with the angular direction of emission can be used to calculate its 3D coordinate. Such a collection of 3D coordinates of different scene points is called a point cloud and is well-suited for various different tasks, such as segmenting the scene into different object classes, detecting and recognizing objects, estimating object velocities and gaining an understanding of the scene. Lidars and ladars also have some limitations. Their observation range will in most practical systems be lower than that of passive systems, since the light waves need to travel through atmosphere both to and from the scene. The 'pixel rate' of a ladar depends on the available laser power, and will in practical systems

Further author information:
E-mail: Egil.Bae@ffi.no, Telephone: +47 95 92 67 31

Electro-Optical and Infrared Systems: Technology and Applications XXI, edited by
Duncan L. Hickman, Helge Bürsing, Ove Steinvall, Philip J. Soan, Proc. of SPIE
Vol. 13200, 1320004 · © 2024 SPIE · 0277-786X · doi: 10.1117/12.3034028

be significantly lower than that of a passive system. Hence, a compromise must be made between the angular resolution, field of view and frame rate.

This paper proposes a sensor fusion framework that utilizes sensor data from both passive and active electro-optical systems together in a complementary fashion. Under poor conditions the data from 2D images or a 3D point clouds may be ambiguous and be difficult to interpret individually. This may for instance be the case from far distances where the resolution or signal-to-noise ratio of both sensor data are poor. Different variants of sensor fusion can be categorized by how 'early' there is a connection between the different sensor data in the processing chain. In the 'late' variants, the sensor data are processed independently by different algorithm, before their outcomes are mixed in some probability distribution. Mid level variants extract individual features from each data set and combine them through the processing chain. The earliest variant is called data level fusion, where the raw data from each sensor are combined in a fused data structure before being analyzed as a single entity. It has the greatest potential for discrimination of subtle patterns that may not be appearent in the individual sensor data.

Our framework falls within data level fusion. It first creates fused data structures, such as high-dimensional data clouds, composed of geometrical 3D coordinates and light intensities from 2D sensors. The fused data structures are then analyzed and segmented into different object classes by adopting and customizing recent segmentation methods designed for high-dimensional irregular data.[1–8] The methodology can also be integrated earlier in the data acquisition process. From long ranges or under poor atmospheric conditions it becomes increasingly difficult to distinguish return pulses of solid objects from noise in ladar data. However, by accepting a large amount of noise in the data set, the methods can segment points corresponding to weak return pulses from solid objects as point clusters in the higher-dimensional fused data clouds. The methodology requires proper alignment of the sensor data in space and time. This poses some challenges in case of motion between and during acquisition of the active and passive sensor data, especially for sensors with low frame rates such as point scanning ladars. We describe methods that estimate and compensate for motion of scene objects and self-motion as part the overall framework. We also propose some improvements to our previous framework[5,7,8] for segmentation of pure ladar data. Experiments are presented throughout the paper demonstrating noise removal, segmentation of outdoor scenes into different object classes and motion estimation in various challenging conditions. Some comparisons are also given against results on pure ladar data. We also give some examples of object recognition, although a detailed treatment of that subject is out of the scope of this paper and is the subject of a future work.

Although our general framework is suited for any kind of 2D image sensors and lidar/ladar, it is motived by challenges faced with mid and longwave IR sensors and point scanning ladars. Experimental results are presented on data from those sensors.

1.1 Related work

The topic of utilizing passive and active electro-optical sensordata together have been subject to a body of research. A comprehensive review is out of the scope of the paper. Much research has been devoted to automatic alignment and calibration of 2D images and 3D point clouds by detecting and relating features, such as edges and corners, in both sensor data e.g.[9,10] Methods for processing and segmenting images on weighted graphs were proposed in[11] and could be used for segmentation of intensity images defined on point clouds. Most current work on combination of 2D image sensors and ladar data operate on later variants of fusion, e.g.[9,10,12] This paper follows a line of search on segmentation and classification of high-dimensional data on graphs using variational methods.[1–8] This methodology has been used for segmentation of point clouds from pure ladar data in.[5,7,8] A closer review of related work will be given within each section.

1.2 Organization

The paper is divided into two main sections. Section 2 describes how different data structures can be constructed from the passive and active sensor data, while compensating for different types of motion during and between acquisition process. Section 3 describes an algorithmic framework for processing and analyzing the fused data in order to detect weak return pulses, segment the data into different object classes and recognize objects. Experiments are presented both within and at the end of each section.

2. FUSION OF 2D IMAGES AND LADAR POINT CLOUDS WITH MOTION COMPENSATION

This section describes different approaches for aligning and fusing together 2D images and ladar/lidar data in common data structures, while the next section is focused on methods for analyzing the fused data. We assume the 2D sensors and ladar are internally calibrated, but that the different sensor data may be acquired at slightly different times. We focus particularly on point scanning ladars that acquire one 3D point at a time sequentially, but the framework is suitable for any kind of ladar. Motion of scene objects of self-motion poses the most significant challenge in order to obtain a good alignment between the sensor data. We start by introducing some terminology and give a formal setup of the problem, then describe how to compensate for motion, before describing how to construct the fused data structures.

2.1 Preliminaries - notations and terminology

For ease of exposition, we assume the 2D images and ladar data are corrected for any roll of the observer. The 2D image sensors 'see' within a field of view

$$\mathrm{FOV} = [\theta_{\min}, \theta_{\max}] \times [\phi_{\min}, \phi_{\max}],$$

where $\theta_{\min}$ and $\theta_{\max}$ are the minimum and maximum observed elevation angles and $\phi_{\min}$, $\phi_{\max}$ are the minimum and maximum observed azimuth angles. The IR images are defined on discrete pixels within the field of view:

$$\Delta\phi_i = [\phi_{\min} + \delta_\phi \cdot (i-1), \phi_{\min} + \delta_\phi \cdot i], \quad i = 1, ..., N_{\mathrm{az}}, \tag{1}$$

$$\Delta\theta_j = [\theta_{\min} + \delta_\theta \cdot (j-1), \theta_{\min} + \delta_\theta \cdot j], \quad i = 1, ..., N_{\mathrm{el}}, \tag{2}$$

where δ_ϕ and δ_θ are the pixel size in radians in the azimuth and pitch directions and N_{az} and N_{el} are the number of pixels in the azimuth and pitch directions. The set of all 'pixels' thus consists of all solid angles

$$[\Delta\phi_i \times \Delta\theta_j]_{i=1,...,N_{\mathrm{az}}\ j=1,...,N_{\mathrm{el}}}.$$

Functions can be defined on the pixels, such as the image intensities I_1 and I_2 recorded by longwave and midwave IR sensors. In general, we denote the image intensity by the c-dimensional vector function $\mathbf{I} = (I_1, I_2, ..., I_c)$ having one element for each channel, for instance two channels for long and mid wave IR, three channels for a color image or multiple channels for hyper/multi-spectral images. The image $\mathbf{I}(\Delta\phi_i, \Delta\theta_j)$ takes the intensity value $\mathbf{I}$ over the entire square area of pixel $\Delta\phi_i \times \Delta\theta_j$. We let $\mathbf{I}(i, j)$ be a short form notation for the intensity in the pixel with index i, j. For ease of derivations we assume that the pixel size of each channel are equal. In case of varying pixels size, all channels could be interpolated down to the smallest pixel size.

A 3D point cloud is usually represented by coordinates in a 3D Euclidean coordinate system. However it is recorded in spherical coordinates similarly to a 2D image. Laser pulses are emitted in different azimuth ϕ and pitch θ directions, and the time-of-arrival of the return pulses are used to calculate the distance to the scene point intersecting the laser pulse. The triplet (d, θ, ϕ) indicates the spherical coordinates of the point and is naturally suited for fusion with a 2D image as discussed above. The point cloud of M points can be represented in spherical coordinates as

$$\{d^i, \theta^i, \phi^i\}_{i=1}^M, \tag{3}$$

where θ^i is the pitch angle ϕ^i azimuth angle and d^i is the distance to point i. The distance d^i therefore plays a similar role to the intensity in a 2D image. In contrast to a 2D image, the distance may not be unique, as each laser pulse may reflect from several points. If the threshold for detecting return pulses is set low, many 'false' signals may be detected caused, e.g., by atmospheric scattering or noise in the detector. The topic of distinguishing weak return pulses of real objects from noise will discussed later. Another distinction between a 2D image and a ladar point cloud is that the azimuth and pitch angles are not necessarily distributed on a regular grid. This is particularly the case for point scanning ladars, which may adjust the scanning pattern arbitrarily. However, since the 2D image may be recorded at a different time from a different point of view, we need to transform the spherical coordinates to a different coordinate system with origin in the same point as the IR image. This non-linear transformation can be computed by going via the Euclidean domain. The point

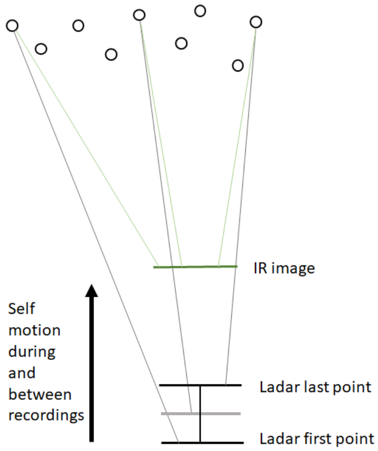


Figure 1: Illustration of fusion of IR image and ladar point cloud under self motion.

coordinates can be represented in a Euclidean coordinate system, with origin in the point of view of the observer, through the transformation:

$$\tilde{x}_1^i = d^i \sin(\theta^i)\cos(\phi^i), \tag{4}$$

$$\tilde{x}_2^i = d^i \sin(\theta^i)\sin(\phi^i), \tag{5}$$

$$\tilde{x}_3^i = d^i \cos(\theta^i). \tag{6}$$

The Euclidean representation will be utilized by the algorithms for data analysis and for visualization. Here $\tilde{\mathbf{x}}^i$ denotes the raw measured point coordinates. Since a point scanning ladar acquires one point at a time, motion will cause the shape of objects to appear deformed in the point cloud. The next section describes how to compensate for motion-based shape deformations and represent the point cloud in a coordinate system centered at the position of the observer during acquisition of the 2D image. These transformations are crucial for obtaining good alignment of the sensor data during fusion. We distinguish betweeen self-motion of the sensor platform and motion of objects in the scene.

2.2 Compensation for motion between and during acquisition of sensor data

We would like to relate a point cloud recorded from a potentially moving observer within the time interval $[T_0, T_1]$ with an IR image recorded from the same observer at time $T_2 > T_1$ as illustrated in Figure 1(a). Some of the scene objects may move during acquisition of the sensor data.

2.2.1 Self motion of sensor platform

Assume that the sensor platform has moved to position $\mathbf{T}^i$ at acquisition of point i. The point coordinates in the same point cloud observed from the origin of the coordinate system can be retrieved by the translation

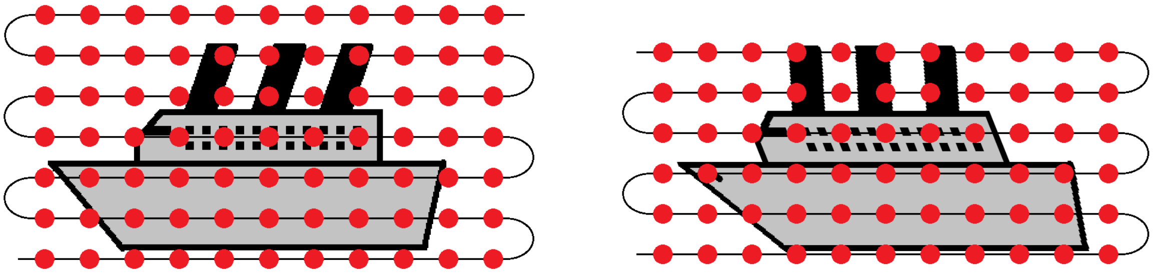
$$\mathbf{x}^i = \tilde{\mathbf{x}}^i + \mathbf{T}^i.$$

Assume that $\mathbf{T}^{2D}$ indicates the position of the observer when the 2D image is recorded. The point cloud can be represented with origin in $\mathbf{T}^{2D}$ by the transformation

$$\mathbf{x}^i = \tilde{\mathbf{x}}^i + \mathbf{T}^i - \mathbf{T}^{2D}.$$

2.2.2 Motion of scene objects

Moving scene objects between acquisition of the 2D image and ladar point cloud represent a greater challenge than self motion, since their velocities are unknown variables. Motion during acquisition also causes the shape of objects to appear deformed in 3D point clouds acquired by a point scanning ladar, since certains objects, such



(a) Scan of a static object (b) Scan of an object moving to the left

Figure 2: Illustration of a line scanning ladar and a typical scanning pattern. In case the scene object is moving during acquisition, its shape appears deformed in the resulting point cloud. The extent of shape deformation becomes more severe for wide view angles, slow scan speeds or fast-moving objects. For illustrative purposes the shape deformations in the figure are exaggerated. The images are reused from our previous work.[13]

as ships, may move several meters within the acquisition time. See Figure 2 for an illustration. The motion of ships may be well approximated by a constant velocity within the time window between acquisition of ladar and 2D images. Our previous article[13] described methods for estimating the velocity of moving objects from a single ladar recording, while also compensating for deformations in the 3D point clouds. Methods for segmenting objects from the point cloud will be described in the next section. Assume for now that the object has been segmented from the point cloud and its velocity $\mathbf{v}^*$ has been estimated. Let $\tilde{\mathbf{x}}^{i_1}, ..., \tilde{\mathbf{x}}^{i_n}$ be the measured 3D points of the object and let $t^{i_1}, ..., t^{i_n}$ be their acquisition times, and t^{2D} be the acquisition time of the 2D image. Here $i_1, i_2, ..., i_n$ are the indices of points in the segmented object. The point cloud can be manipulated to represent its appearance at the acquisition time of the 2D image by the transformation

$$\mathbf{x}^{i_j} = \tilde{\mathbf{x}}^{i_j} + (t^{2D} - t^{i_j})\mathbf{v}^*, \quad j = 1, ..., n. \tag{7}$$

The manipulated point cloud $\tilde{\mathbf{x}}^{i_1}, \tilde{\mathbf{x}}^{i_2}, ..., \tilde{\mathbf{x}}^{i_n}$ and 2D image can be now be fused using the same methodology as for static objects.

The above formulas for compensating for self-motion and moving scene objects can also be combined in the formula

$$\mathbf{x}^{i_j} = \tilde{\mathbf{x}}^{i_j} + \mathbf{T}^{i_j} - \mathbf{T}^{2D} + (t^{2D} - t^{i_j})\mathbf{v}^*.$$

2.3 Fusion of 2D images and 3D point clouds

The previous sections have established how to calculate the 3D point coordinates $\mathbf{x}_i$ of the scene in a coordinate system centered at the position of the sensor platform during acquisition of the 2D image, while compensating for displacements and deformations caused by motion of scene objects of self-motion during acquisition.

We would like to relate the pixels in the 2D image and 3D points in the point cloud. In order to do so, the point cloud from view point $\mathbf{T}^{2D}$ should be represented as a distance for different azimuth and pitch view angles. This can be retrieved through the transformation from Eucledian to spherical coordinates:

$$d(\mathbf{x}) = \sqrt{x_1^2 + x_2^2 + x_3^2}, \tag{8}$$

$$\theta(\mathbf{x}) = \arctan\left(\frac{x_2}{x_1}\right), \tag{9}$$

$$\phi(\mathbf{x}) = \arccos\left(\frac{x_3}{\sqrt{x_1^2 + x_2^2 + x_3^2}}\right). \tag{10}$$

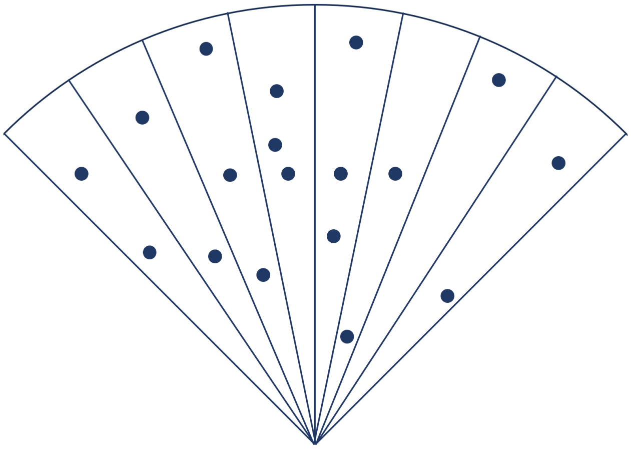


Figure 3: Illustration of a point cloud overlaid on the field of view of a 2D image. Multiple points may be observed within the field of view of each pixel, especially if the sensitivity for detection of return pulses is high. Depending on the sensors, the divergence of the laser beam may be larger or smaller than the field of view of the 2D pixels. In this example the 2D sensor has a lower angular resolution. The example will be continued in later illustrations.

The above transformations give the distance and angular coordinates of 3D points recorded by the active system, from the same view point as the 2D image was recorded from. They will be utilized in the next sections to fuse these sensor data in different ways. An illustration is shown in Figure 3, where a 3D point cloud is overlaid the field of view of a 2D image with 8 pixels. Each 3D point may be related to a single 2D pixel through its spherical coordinates. However, each 2D pixel could be related to many 3D points as several points may fall within the instantaneous field of view of each pixel.

2.3.1 Back-projection of IR images onto ladar point clouds

Note that from formulas (1)-(2), a point with azimuth angle $\phi(\mathbf{x}) \in [\phi_{\min}, \phi_{\max}]$ and elevation angle $\theta(\mathbf{x}) \in [\theta_{\min}, \theta_{\max}]$ falls within the pixel with index

$$i(\phi(\mathbf{x})) = \text{floor}\big(\frac{\phi(\mathbf{x}) - \phi_{\min}}{\delta_\phi}\big), \quad j(\theta(\mathbf{x})) = \text{floor}\big(\frac{\theta(\mathbf{x}) - \theta_{\min}}{\delta_\theta}\big).$$

The point $\mathbf{x}$ can be 'colored' by the image intensity at that pixel, i.e. we can define

$$\mathbf{I}^{\text{pc}}(\mathbf{x}) = \begin{cases} \mathbf{I}(i(\phi(\mathbf{x})), j(\theta(\mathbf{x})) & \text{if } \phi(\mathbf{x}) \in [\phi_{\min}, \phi_{\max}] \text{ and } \theta(\mathbf{x}) \in [\theta_{\min}, \theta_{\max}] \\ 0 & \text{else} \end{cases} \tag{11}$$

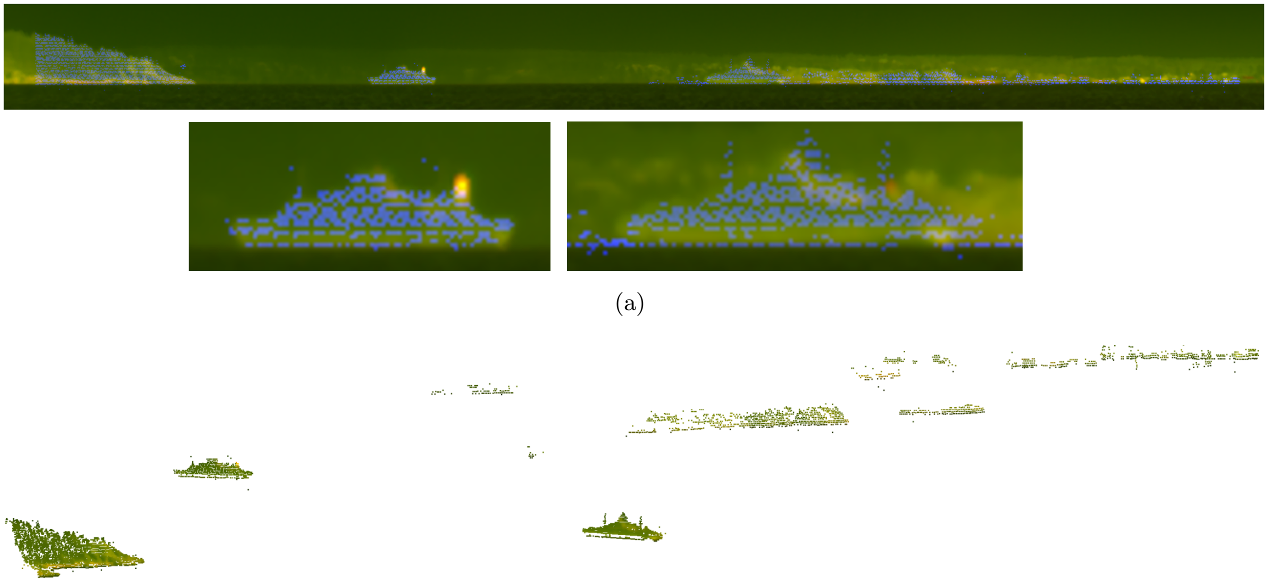
taking the intensity values of the 2D image within the field of view and zero outside the field of view. By this sceme, every 3D point that falls within the field of view of a pixel is colored by the intentities of that pixel. In case of ambiguity, where several points in the point cloud share the same azimuth and pitch angles ϕ, θ, the image intensities could alternatively be projected onto only the frontmost point or the backmost point within each pixel. Different variants of back-projection will be discussed further in the next section.

Instead of defining the intensity as a function over the set of points, another way of fusing the data structures is by creating a higher-dimensional point cloud where the image intensities are regarded as separate dimensions. Define $\mathbf{I}^i = \mathbf{I}^{\text{pc}}(\mathbf{x}^i)$ to be the image intensities at the 3D point $\mathbf{x}^i$. Assuming the image has c color channels, the vector $\mathbf{I}^i = (I_1^i, ..., I_c^i)$ is therefore a c-dimensional vector. A higher-dimensional point can be constructing by concatenating the geometrical point coordinates and intensity values

$$(x_1^i, x_2^i, x_3^i, I_1^i, ..., I_c^i).$$

The higher-dimensional point cloud is the set of all such points for $i = 1, ..., M$

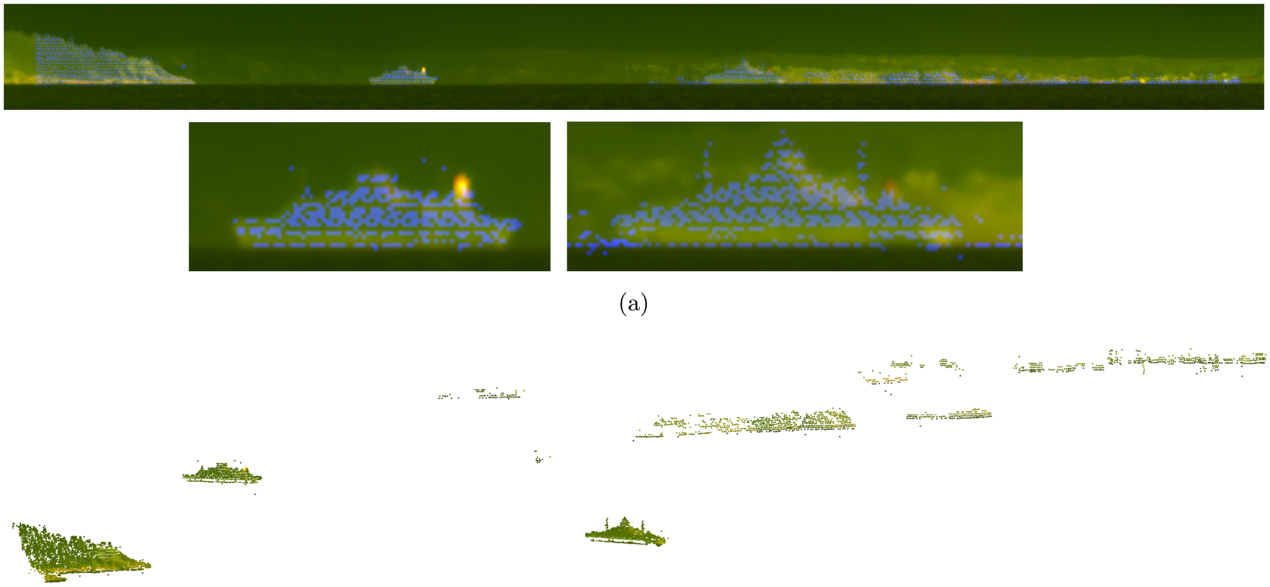
$$\{(x_1^i, x_2^i, x_3^i, I_1^i, ..., I_c^i)\}_{i=1}^M.$$



(a)

(b)

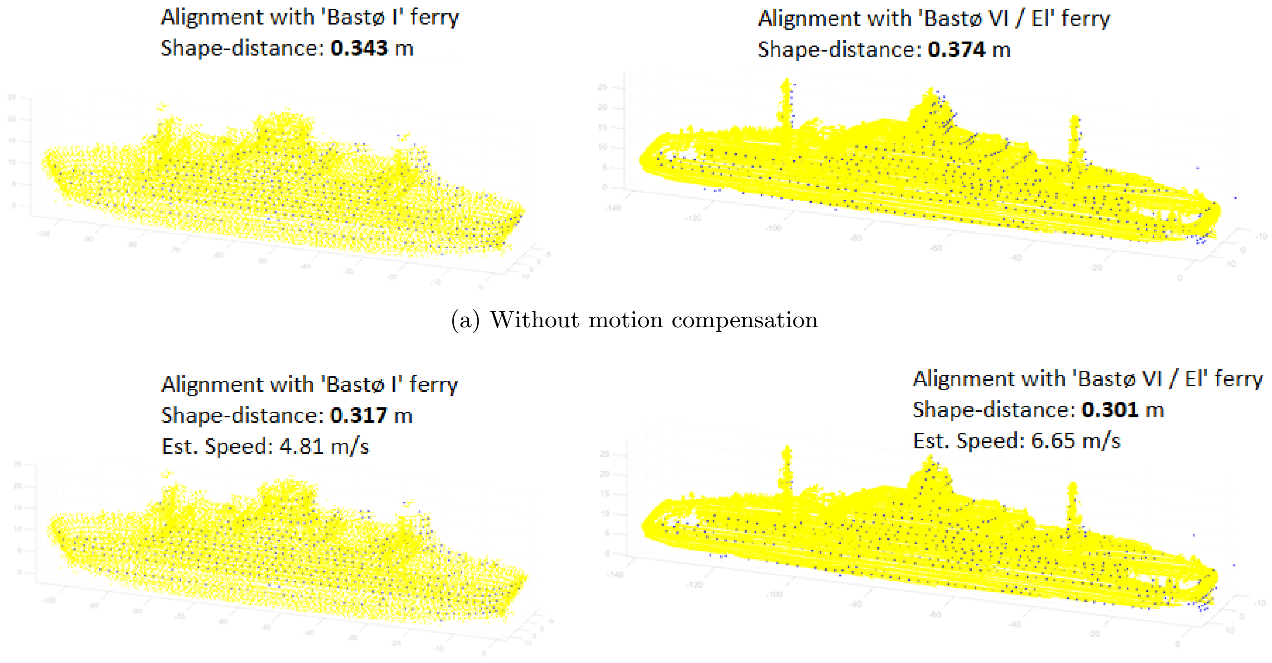
Figure 4: Fusion of 3D point cloud from ladar with long wave (green channel) and mid wave (red channel) IR images. (a) Forward projection of point cloud down to the image plane of an IR image. The point cloud distance function is visualized in the blue channel. (b) Back-projection of IR images onto a 3D point cloud. In this example there is no compensation for movement of objects between acquisition of the IR and ladar images.



(a)

(b)

Figure 5: Same example as Figure 4, but with compensation for motion of objects during and between acquisition of ladar and IR data. (a) Forward projection of point cloud down to the image plane of an IR image. The point cloud distance function is visualized in the blue channel. (b) Back-projection of an IR images onto 3D point cloud.


Alignment with 'Bastø I' ferry
Shape-distance: **0.343** m
Alignment with 'Bastø VI / El' ferry
Shape-distance: **0.374** m
Alignment with 'Bastø I' ferry
Shape-distance: **0.317** m
Est. Speed: 4.81 m/s
Alignment with 'Bastø VI / El' ferry
Shape-distance: **0.301** m
Est. Speed: 6.65 m/s

(a) Without motion compensation

(b) With motion compensation and estimation

Figure 6: The vessels marked in blue have been segmented from the point cloud in Figure 4 by methods that will be described in the next section. Blue indicates observed points without motion compensation, yellow indicates points on 3D models of the Bastø ferries. (a) No compensation for motion-based deformation. (b) Comparative matches while estimating and compensating for motion induced shape deformations of the vessels using the method described in.[13] The estimated velocities are used to predict the positions of the vessels at the acquisition time of the IR image for improved fusion as shown in Figure 5.

2.3.2 Forward-projection of ladar point clouds onto IR images

Another way of fusing the data is to represent the point cloud as a distance function over the same pixel grid as the 2D image. This can be achieved by regarding the point cloud as a distance $D^P(i,j)$ for each solid angle $\Delta\phi_i, \Delta\theta_j$. This approach naturally leads to some loss of information, as at most one distance can be encoded in each pixel. It must therefore be decided if $D^P(i,j)$ should indicate the distance to e.g. the farthest or closest point within solid angle $\Delta\phi_i, \Delta\theta_j$. Furthermore, the conversion to a regular grid leads to loss of information of point scanning ladars that scan irregularly. In order to visualize results, the distance function representing the ladar point cloud is encoded in the blue channel of a colour image, where darker shades of blue indicate closer points. Note that small relative distances may be difficult for the eye to observe from the shading. The visualization serves mostly to indicate where 3D points are recorded relative to the IR images.

2.4 Example results and notes on visualization

Example results in a maritime scene are shown in Figure 4 and 5. Here long and mid wave IR images are fused with a 3D point cloud obtained by scanning with a point scanning ladar. The long wave IR image is encoded in the green color channel and the midwave IR image in the red channel. Fusion by forward projection are shown in (a) and back-ward projection in (b). The two vessels move during acqusition of the sensor data. Results without any motion compensation are shown in Figure 4. Comperative results with motion compensation are shown in Figure 5. Here the motion of the vessels have been estimated by the algorithm described in our previous work,[13] by optimizing the shape distance between the observed objects and 3D model of the ferries over the deformation field in addition to the 3D pose of the objects. Results of the optimal aligments are shown in Figure 6 without (a) and with (b) motion compensation. Here blue indicates the observed object and yellow indicates the 3D models. The shape distance is used as one of several components for recognizing objects. However, due to the good conditions in this example the vessels can be recognized as the ferries Bastø I and VI respectively from the shape distance alone. Further examples of fusion will be shown in the next section.

3. ANALYZING FUSED 2D IMAGES AND 3D LADAR POINT CLOUDS

This section describes and demonstrates algorithms for analyzing the fused data in order to segment and classify scene objects. We will also briefly touch on object recognition altough a detailed treatment of that topic falls out of the scope of this paper.

3.1 Segmentation of N-dimensional point clouds

Our previous work[5,7,8] has shown that a graph-based framework is well-suited for segmentation and analysis of 3D point clouds. The point clouds could be segmented into different object classes based on geometry of the 3D points coordinates. In this section we show that the graph-based framework can be naturally extended to fused 2D and 3D data of the type described in Section 2.3.1. We also describe several improvements to our previous framework[5,7,8] for pure 3D point clouds, including more fine segmentation and incorporation of invariances to the distance to the scene.

Recall that the fused 2D image and ladar data structure can be represented as N-dimensional (ND) point clouds. For example, a 3D point cloud and 2D image with c channels could be represented as a c+3 dimentional point cloud $\{(x_1^i, x_2^i, x_3^i, I_1^i, ..., I_c^i)\}_{i=1}^M$. We let the bold letters $\mathbf{p}$ and $\mathbf{q}$ be vectors containing the coordinates of the points, e.g. $\mathbf{p} = (x_1, x_2, x_3, I_1, ..., I_c)$ or $\mathbf{p} = (x_1, x_2, x_3)$. In general, the point clouds will be constructed of artificially higher dimensions, where properties of local patches of points are encoded in each dimension. Details will be given in the next subsection.

A graph is a set of vertices together with a pairwise interconnection between the set of vertices. The set of all vertices is denoted by V. In our application, each vertex is associated with a data point $\mathbf{p}$ and that vertex will also be referred to as $\mathbf{p}$. A pair of vertices that are connected is called an edge in the graph. The set of all edges is denoted by E. Two vertices $\mathbf{p}$ and $\mathbf{q}$ are connected if and only if their vertex pair $(\mathbf{p}, \mathbf{q})$ is contained in the set of edges E. In this work, we construct edges between each node and its nearest neighbors in some metric space that will be specified later.

Once the graph has been constructed, it is possible to define functions over either the vertices or the edges. A weight function $w(\mathbf{p}, \mathbf{q})$ is defined on the edges $(\mathbf{p}, \mathbf{q}) \in E$ and can be used to measure the similarity between the points $\mathbf{p}$ and $\mathbf{q}$. A high value of $w(\mathbf{p}, \mathbf{q})$ indicates that $\mathbf{p}$ and $\mathbf{q}$ are similar and a low value indicates that they are dissimilar. A popular choice of the weight function for general data classification problems is the Gaussian

$$w(\mathbf{p}, \mathbf{q}) = e^{-\frac{d(\mathbf{p},\mathbf{q})^2}{\sigma^2}}, \tag{12}$$

where $d(\mathbf{p}, \mathbf{q})$ is the distance between $\mathbf{p}$ and $\mathbf{q}$ in some metric space and σ^2 is the variance. A simple examle is the weighted Euclidean distance

$$d(\mathbf{p}, \mathbf{q}) = \sqrt{\sum_k w_k^d (p_k - q_k)}, \tag{13}$$

where the weights are used for normalization, e.g. by $w_k^d = \frac{1}{\sigma_k^2}$ where σ_k^2 is the variation of data within dimension k.

Weights can also be defined on the nodes $\mathbf{D}(\mathbf{p})$, where $\mathbf{D}$ is a vector function with one component for each possible class. In supervised case it will be used to measure how well each point fits with characteristics of each class individually.

The problem of segmenting a point cloud into n regions is mathematically the problem of finding a partition of the set of points V into subsets $V_1, ..., V_n$ that do not overlap with each other. Point cloud segmentation can be formulated as an energy minimization problem by defining an energy functional that assigns an energy value to every possible partition $V_1, ..., V_n$ of V, and where the desired partition is encoded as the one with lowest energy. The minimization problem can be formulated in general as

$$\min_{\{V_i\}_{i=1}^n} \sum_{i=1}^n \sum_{\substack{(\mathbf{p},\mathbf{q})\in E:\\ \mathbf{p}\in V_i,\, \mathbf{q}\notin V_i}} w(\mathbf{p},\mathbf{q}) + \sum_{i=1}^n \sum_{\mathbf{p}\in V_i} D_i(\mathbf{p}) \tag{14}$$

$$\text{such that } \cup_{i=1}^n V_i \,=\, V\,, \quad V_k \cap V_l \,=\, \emptyset\,, \forall k \neq l\,. \tag{15}$$

The first term of (14) is used to encourage the merging of similar points to the same region. It sums together the weights on all edges whose endpoints belong to two different regions. In order to make this term small, the weights on these edges should be small, indicating that it is favourable to separate the two points in the point pair into two different regions. The last term of (14) is used to encourage region homogeneity. It measures how well each point fits with each region individually. The constraints impose that there should be no vacuum or overlap between the classes.

The problem (14) may also be expressed more intuitively as the dual maximization problem

$$\max_{\{V_i\}_{i=1}^n} \sum_{i=1}^n \sum_{\substack{(\mathbf{p},\mathbf{q})\in E:\\ \mathbf{p},\mathbf{q}\in V_i}} w(\mathbf{p},\mathbf{q}) - \sum_{i=1}^n \sum_{\mathbf{p}\in V_i} D_i(\mathbf{p}) \tag{16}$$

$$\text{such that } \cup_{i=1}^n V_i \,=\, V\,, \quad V_k \cap V_l \,=\, \emptyset\,, \forall k \neq l\,. \tag{17}$$

The interpretation of the first term of (16) is to maximize the similarity of points $\mathbf{p}$ and $\mathbf{q}$ within the same class, while the second term encourages points within each class to adhere to predefined characteristics of that class, as before.

In addition, various balancing contraints or penalty terms acting on the volume of the classes and the relationship between them can be imposed. The general framework can be used for unsupervised, semi-supervised or supervised classification and segmentation of various types of data.[1–6] In the unsupervised case, the second term involving $\mathbf{D}(\mathbf{p})$ is ignored so that points are clustered together based based on their similarity to each other. The classes are determined based on meaningful point clusters in the higher dimensional space, and the the number of classes may also be an unknown variable as in maximization of network modularity.[6] In the semi-supervised case, the class membership of some of the data points are known in advance and imposed as hard constraints in the optimization problem. Both labeled and unlabeled data are utilized together for improved performance when solving the minimization problem.

This work is focused on supervised segmentation, where some characteristics of each class is encoded in the node weights. We assume the number of classes is known and also incorporate some domain knowledge into the model. Volume contraints will not be utilized in this work. However, the following novel constraints related to volume will to be used in connection with detection of weak return pulses

$$|V_i \cap \Omega_k| \leq S_k, \quad k = 1, ..., K. \tag{18}$$

Here $\Omega_k \subset V$ is some subset of the points V, and $|V_i \cap B_k|$ is the number of points in the intersection of the segmented region V_i and B_k. The constraints say that the number of points within this intersection should be less than or equal to S_k. In the special case that $B_k = V$, then this constraint simply says that the volume of region V_i should be less than or equal to S_k. The points in B_k will typically consist of all 3D points that fall within the instantaneous field of view of the pixel in the active system corresponding to point k. Elaborations will be given in the next subsection.

A robust and efficient algorithmic framework for solving the minimization problems was given in our work[5] using variational calculus on graphs. It can be adapted with a few modification to incorporate constraints of the form (18). However, a description of this framework is more mathematically technical and falls outside the scope of this paper.

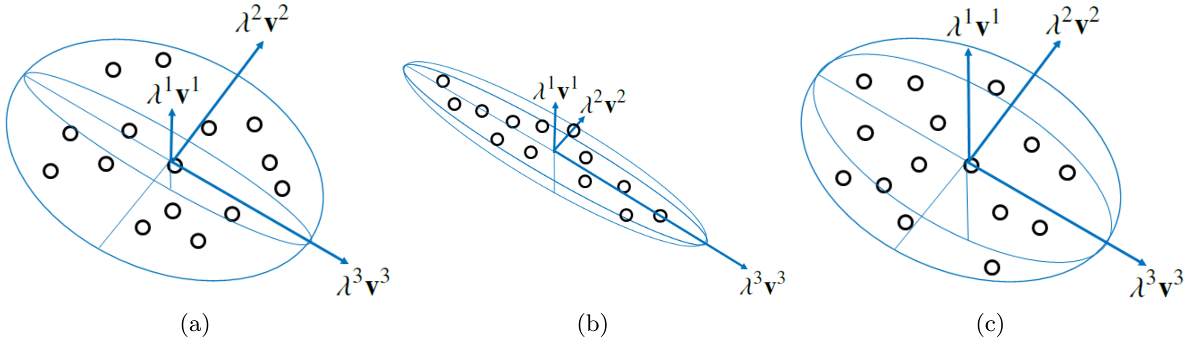


Figure 7: Illustration in three dimensions of some different distributions of point in the point patch $\mathcal{P}^i$. (a) Eigenvalues $\lambda^1(\mathbf{p}) \ll \lambda^2(\mathbf{p})$ indicating that the points located in a plane. The first eigenvector $\mathbf{v}^1$ indicates the normal vector of the plane. (b) $\lambda^2(\mathbf{p}) \ll \lambda^3(\mathbf{p})$ indicating that the points distributed along a thin structure. The third eigenvector $\mathbf{v}^3$ indicates the direction of the structure. (c) $\lambda^1(\mathbf{p}) \approx \lambda^2(\mathbf{p})$ and $\lambda^2(\mathbf{p}) \approx \lambda^3(\mathbf{p})$ indicating there is not a significant difference between the three eigenvalues). This suggest a random, irregular structure of the point coordinates.

3.2 Constructing the graph

Weights on the edges measure similarity between data points through the Formula (12). We are interested in measuring how similar local patches of points are to each other. For each point $\mathbf{p} = (x_1^i, x_2^i, x_3^i)$ or $\mathbf{p} = (x_1^i, x_2^i, x_3^i, I_1^i, ..., I_c^i)$ the local patch is defined as

$$\mathcal{P}^i = \{\mathbf{q} \in V \text{ such that } (\mathbf{p}^i, \mathbf{q}) \in N_k(\mathbf{q})\} \cup \mathbf{p}^i, \tag{19}$$

where $N_k(\mathbf{q})$ consisting of the point $\mathbf{p}^i$ and all its k nearest neighbors in the metric space with distance function (13). In practice we use the 20 nearest neighbors.

Principle component analysis (PCA) is performed on points in $\mathcal{P}^i$ to reduce the dimensionality of the problem. We calculate eigenvalues and eigenvectors of the correlation matrix corresponding to points in $\mathcal{P}^i$. Define for notational convenience $\mathbf{q}^0 = \mathbf{p}$. Define the normalized vectors $\bar{\mathbf{q}}^i = \mathbf{q}^i - \text{mean}(\mathbf{q}^0, \mathbf{q}^1, ..., \mathbf{q}^k)$, for $i = 0, 1, ..., k$ and construct the matrix

$$\mathbf{Y} = [\bar{\mathbf{q}}^0\bar{\mathbf{q}}^1\bar{\mathbf{q}}^2...\bar{\mathbf{q}}^k]. \tag{20}$$

The eigenvalues and eigenvectors of the correlation matrix $\mathbf{Y}\mathbf{Y}^T$ indicate how the points in the neighborhood of $\mathbf{p}$ are distributed in relation to each other. The eigenvectors $\mathbf{v}^1(\mathbf{p}), \mathbf{v}^2(\mathbf{p}), ..., \mathbf{v}^m(\mathbf{p})$ indicate principle directions of variations and the corresponding eigenvalues $0 \leq \lambda^1(\mathbf{p}) \leq \lambda^2(\mathbf{p}) \leq ... \leq \lambda^m(\mathbf{p})$ indicate the variation in each direction, arranged from smallest to largest. See Figure 7 for some illustrations in three dimensions, where the eigenvectors and eigenvalues have some nice geometrical interpretations.

PCA capture local properties of the points, but information about their interspace distance tends to get lost. The point density is defined as another property of the patch:

$$\rho_i = \frac{1}{\sum_{\mathbf{q}\in\mathcal{P}^i} d(\mathbf{q}, \mathbf{p}^i)},$$

where d is a weighted Euclidean distance (13), where the weights depend on the scaling of the input images.

For a fused 3D point cloud and 2D image of c channels the dimension is $m = c + 3$. Concatenating pointwise coordinates and properties of the point patches all together results in the higher dimensional synthetic data cloud

$$\mathbf{p} = (x_1, ..., x_3, I_1, ..., I_c, \rho, \lambda^1, ..., \lambda^m, v_1^1, ..., v_m^1, v_1^2, ..., v_m^2, ..., v_1^m, ..., v_m^m).$$

The dimension of the synthetic point cloud is $N = m^2 + 2m + 1$. A particular component of $\mathbf{p}$ will be refered to by the name of that component, for instance $x_3(\mathbf{p})$ indicates the height coordinate and $\lambda^1(\mathbf{p})$ indicates the

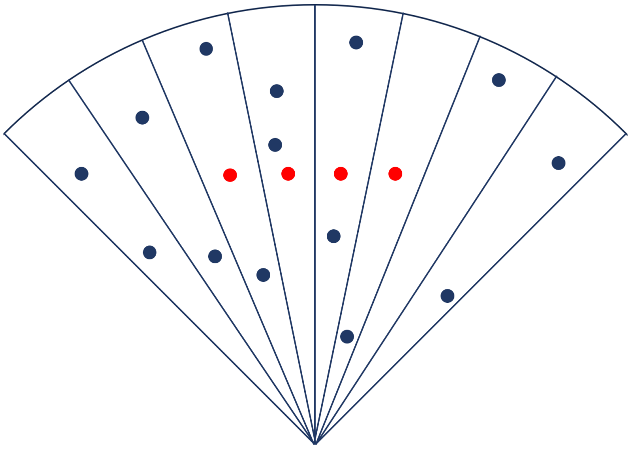


Figure 8: Continuation of Figure 3. The point cloud is segmented into dense objects (red) and noise (blue), and contraints are imposed so that the dense objects can contain at most one 3D point from the field of view of each pixel.

first eigenvalue derived from the point patch etc. With slight abuse of notation we will also refer to the spherical coordinates in 3D space corresponding to point $\mathbf{p}$ as $\phi(\mathbf{p})$, $\theta(\mathbf{p})$ and $d(\mathbf{p})$.

The edge weights $w(\mathbf{p}, \mathbf{q})$ can now be calculated from Formula (12) using the distance (13). Alternatively, the distance function could be a mixture of Euclidean distances between the first $2m$ dimensions and cosine distances between eigenvectors encoded in the remaining dimensions.

In this work we would like to encode a priori knowledge of the object classes into the model. This can be achieved through the node weights D_i. Each D_i is formulated as a weighted sum of n_f features $F^1, F^2, ..., F^{n_f}$:

$$D_i(\mathbf{p}) = w_i^1 F^1(\mathbf{p}) + w_i^2 F^2(\mathbf{p}) + ... + w_i^{n_f} F^{n_f}(\mathbf{p}), \tag{21}$$

where the weight w_i^j indicates how important feature F^j is within class i. The features F^j can take both positive and negative values, where negative values indicate favourable characteristics of a particular class, while positive values are disfavourable. Particular choices of features will be discussed for each application.

3.3 Detection of weak laser return pulses

A 3D point is detected if there a power spike from the ladar detector exceeding a certain threshold. If the scene is sufficiently reflective at the laser wavelength, and there is not too much atmosphere between the ladar and the scene, then the threshold can be set to a high level. This makes it possible to acquire a 3D image of the scene while excluding noise. Under less fortunate conditions, the laser return pulses from the scene are weaker, and may not be individually distinguishable for noise sources, such as reflectance from air particles in the atmosphere or background noise in the detector. This may occur if there is a very far distance between the ladar and the scene, if the atmopheric conditions are poor, or if the surfaces of the scene objects reflect poorly. By setting the threshold to a low level, more scene points may get detected, but at the expense of more noisy points. Our previous work showed how firm objects can be segmented from the rest of the scene by utilizing point density in the energy minimization framework.[7] A similar concept can be extended to fused 3D point clouds in IR images, and can be utilized to detect even weaker return pulses. The basic idea is to allow a large amount of noise in the data set and then extract solid objects as point clusters in the higher-dimensional fused point cloud.

We would like to define features F^1 and F^2 in the node weights such that F^1 is smaller than F^2 on firm objects and vice versa on noise such as reflections from the atmosphere. This can be encoded by defining:

$$F^1(\mathbf{p}) = \rho_t - \rho(\mathbf{p}) \quad F^2(\mathbf{p}) = \rho(\mathbf{p}) - \rho_t, \tag{22}$$

where ρ_t indicates the smallest expected density on firm objects.

Additionally, we assume that the firm objects are non-transparent, so that each laser pulse should only reflect from one scene point. This can be incorporated by requiring that the segmented region of firm objects should contain at most one 3D point from the field of view of each 2D pixel. The point cloud can be represented in spherical coordinates through the transformation (8)-(10). We let $(d(\mathbf{p}), \theta(\mathbf{p}), \phi(\mathbf{p}))$ represent the spherical coordinates of the 3D point $(\mathbf{p})$. Recall that the pixel $(i.j)$ of the 2D image see within the solid angle

$$[\Delta\phi_i \times \Delta\theta_j]_{i=1,\ldots,N_{\mathrm{az}},\, j=1,\ldots,N_{\mathrm{el}}}.$$

The ladar may have a different angular resolution than the 2D image. We let the instantanous field of view of of the ladar when recording point k be denoted

$$[\Delta\phi_k \times \Delta\theta_k]_{k=1,\ldots,M}.$$

We let the set of all points whose azimuth and pitch angle fall within the solid angle $[\Delta\phi_k \times \Delta\theta_k]$ be denoted as

$$B_k = \{\mathbf{p} \text{ s.t. } \phi(\mathbf{p}) \in \Delta\phi_k \text{ and } \theta(\mathbf{p}) \in \Delta\theta_k\}, \quad k = 1, ..., M. \tag{23}$$

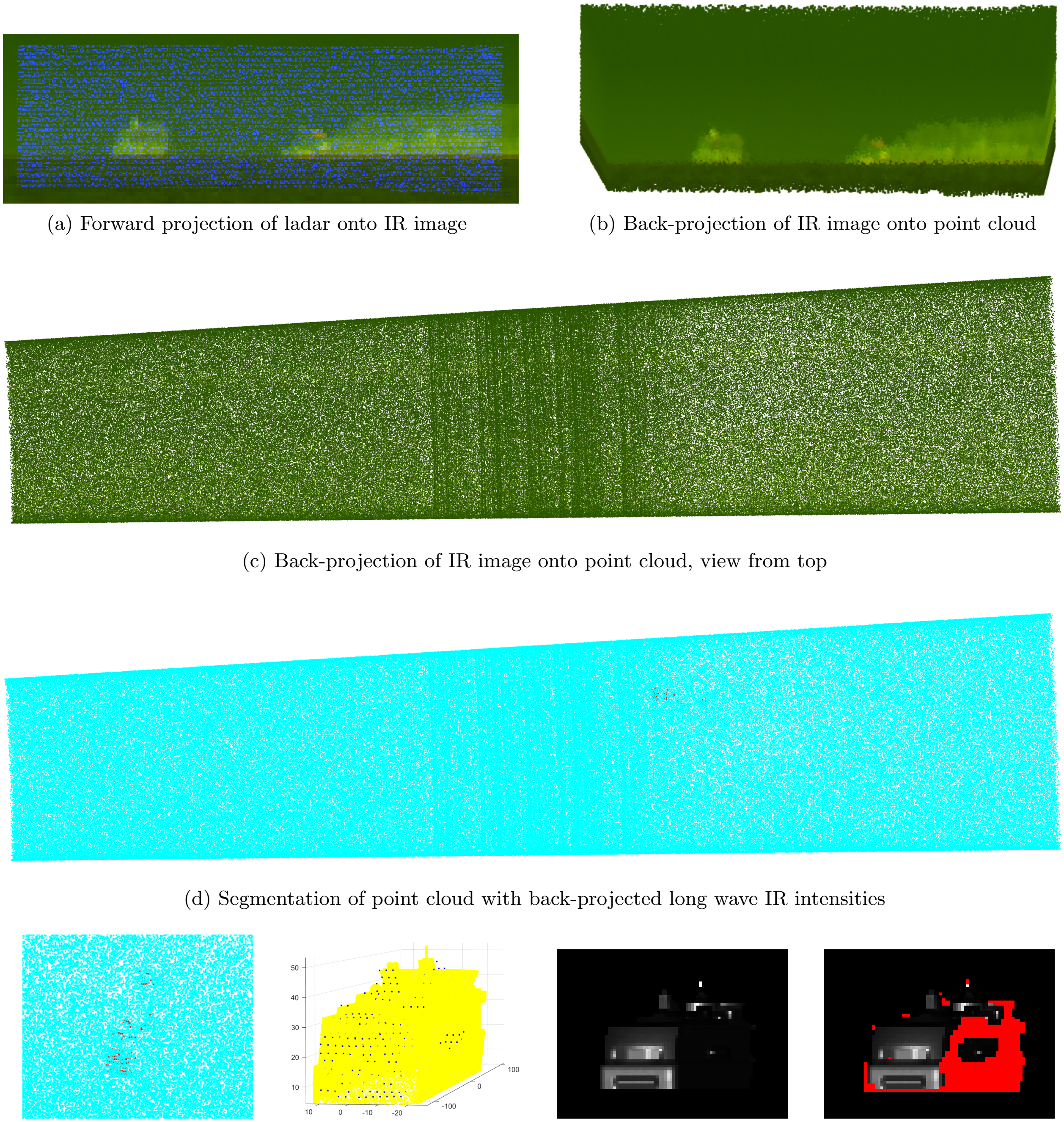
By imposing (18) with the above definition of B_k and $S_k = 1$ for all $k = 1, ..., M$ ensures that at most one 3D point is is contained in the segmented object within each solid angle. An illustration is shown in Figure 8, where points marked in red indicate the segmented object. Here the angular resolution of the ladar and 2D image are equal.

3.3.1 Example results

An example is shown in Figure 9 where the cruise ship 'Color Magic' is observed from a very far distance. Due to the thick atmosphere at sea levels, the laser pulses decay significantly while traveling to and from the vessel. The framework can be used for detecting weak return pulses from the vessel by setting the threshold for detecting incoming laser pulses to a very low level. The resulting point cloud is fused with the IR sensor data using backprojection and visualized in (b) and (c) from two different view points. Segmentation results are shown in (d) and (e) where light blue indicates the noise class and red indicates the class of solid objects. Note that the terrain in the foreground falls outside the range interval used in the visualization. Subfigures (f) - (g) visualize some of the features that can be used for object recognition. Subfigure (f) shows alignment with a 3D model of Color Magic, while estimating and correcting for motion induced deformations. The segmented object is marked by blue points and the 3D model is marked in yellow. The estimated speed in 11.46 m/s (22.3 knots) in the forward direction. The cruise speed of the ship is 22 knots according to Wikipedia. Subfigure (g) shows the expected relative strength of return pulses over the projected area of 'Color Magic' down to the image plane, using its hypothesized position and orientation in the world from the alignment in (g). The relative strength is calculated based on the incidence angle of the laser beams with the surface of 'Color Magic'. Subfigure (h) shows in red the projected area in the image plane where no 3D points are detected, overlaid over the relative strength image. Even though a relatively large part of the projected area remains undetected, this falls in line with expectations due to the slant laser beam incidence angle over those areas and thus does not significantly penalize recognition confidence. Some lower parts of the vessel remain unobserved since it is partially located beneath the horizon. Further features can also be derived from the IR images to strengthen the confidence. A more detailed treatment of object recognition using fused data will be the subject of in a future work.

3.4 Segmentation of outdoor scenes

We would like to segment outdoor scenes into common object classes such as vegetation, buildings, the ground surface, vehicles, roads, flag poles and power cables. Some domain knowledge is built into the model by defining features characteristic of each class. Previous work has derived features from local neighborhood of 3D coordinates in pure ladar data.[5,7,14,15] For instance, buildings and other human-made structures tend to be constructed of smooth surfaces, separated by sharp discontinuities,.[5,7,15] The smooth surfaces are appearent both geometrically and thermally in IR images. In contrast vegetation, such as trees and bushes, tend to have a more disorderly appearance, both geometrically and thermally. Smooth structures in the ND space of geometric coordinates and image intensities can be characterized by a small first eigenvalue relative to the second eigenvalue. This



(a) Forward projection of ladar onto IR image

(b) Back-projection of IR image onto point cloud

(c) Back-projection of IR image onto point cloud, view from top

(d) Segmentation of point cloud with back-projected long wave IR intensities

(e) Close-up view of segmentation

(f) Optimal alignment between the segmented object (blue) and the the cruise ship 'Color Magic' (yellow)

(g) Expected relative strength of return pulses from 'Color Magic' based on incidence angle of laser beams with the estimated surface orientation.

(h) Parts of the estimated projected object area (red) where no return pulses are detected.

Figure 9: Segmentation of vessel in fused data from ladar and IR sensors. By accepting large amounts of noise in the data set, weak return pulses are also detected. The vessel is segmented from noise as a point cluster in the higher-dimensional space of fused data while satisfying uniqueness constraints. The segmented vessel can be recognized as 'Color Magic' based on features illustrated in (f)-(h).

property can be encoded in the node weights as characteristics of the classes. In practice, we found that the best performance is achieved by only measuring smoothness and other characteristics in terms of the 3D geometries. The combination of geometry and intensities tend to be more useful on the edge weights as described in previous sections. In the following, we let $\lambda^1(\mathbf{p}), \lambda^2(\mathbf{p}), \lambda^3(\mathbf{p})$ and $\mathbf{v}^1(\mathbf{p}), \mathbf{v}^2(\mathbf{p}), \mathbf{v}^3(\mathbf{p})$ be eigenvalues and eigenvectors of the correlation martrix of 3D point patches. In particular, we have found the following features related to the eigenvalues and eigenvectors to be useful

$$F^1(\mathbf{p}) = \gamma^1\lambda^1(\mathbf{p}) - \lambda^2(\mathbf{p}), \qquad F^2(\mathbf{p}) = \lambda^2(\mathbf{p}) - \gamma^1\lambda^1(\mathbf{p}), \tag{24}$$

$$F^3(\mathbf{p}) = \gamma^2\lambda^2(\mathbf{p}) - \lambda^3(\mathbf{p}), \qquad F^4(\mathbf{p}) = \lambda^3(\mathbf{p}) - \gamma^2\lambda^2(\mathbf{p}), \tag{25}$$

$$F^5(\mathbf{p}) = -\left|\mathbf{v}^1(\mathbf{p}) \cdot \mathbf{n}\right|, \qquad F^6(\mathbf{p}) = \left|\mathbf{v}^1(\mathbf{p}) \cdot \mathbf{n}\right|, \tag{26}$$

$$F^7(\mathbf{p}) = -\left|\mathbf{v}^3(\mathbf{p}) \cdot \ell\right|, \qquad F^8(\mathbf{p}) = \left|\mathbf{v}^3(\mathbf{p}) \cdot \ell\right|. \tag{27}$$

Here γ^1 and γ^2 are two parameters greater than 1. If γ^1 times $\lambda^1(\mathbf{p})$ is smaller than $\lambda^2(\mathbf{p})$ for a sufficiently large value of γ^1, this indicates that $\lambda^1(\mathbf{p}) \ll \lambda^2(\mathbf{p})$ and consequently a high degree of planarity (see Figure 7 (a) for an illustration). If $\gamma^1\lambda^1(\mathbf{p}) < \lambda^2(\mathbf{p})$, then $F^1(\mathbf{p})$ becomes negative. Therefore $F^1(\mathbf{p})$ is a feature than can distinguish vegetation from regular structures. The feature $F^2(\mathbf{p})$ is just the negative of $F^1(\mathbf{p})$ and takes small values for regular structures, such as buildings, vehicles and the ground surface. Vehicles can be segmented from sufficiently short ranges and be distinguished from other regular structures by their dimensions and vicinity proximity to points in the ground class.

Similarly, if γ^2 times $\lambda^2(\mathbf{p})$ is smaller than $\lambda^3(\mathbf{p})$ for a sufficiently large value of γ^2, then $\lambda^2(\mathbf{p}) \ll \lambda^3(\mathbf{p})$ indicating a high degree of elongation (see Figure 7 (b) for an illustration). The feature $F^3(\mathbf{p})$ is negative if $\gamma^2\lambda^2(\mathbf{p}) < \lambda^3(\mathbf{p})$ and is therefore suitable for distinguishing thin structures in the scene. The feature $F^4(\mathbf{p})$ takes small and negative values for structures that are not thin. The orientation of the structure is given by the third eigenvector $\mathbf{v}^3$.

Features F^5 and F^6 measure the absolute cosine distance between the first eigenvector and the normal vector $\mathbf{n}$. F^5 takes a negative value if these vectors are parallel and gradually increases the less parallel they are to each other. The first eigenvector approximates the normal vector of the surface the points are sampled from. This feature takes a small value (desired) for objects normal vectors parallel to $\mathbf{n}$. $F^6 = -F^5$ and takes small values if the plane is oriented perpendicular to $\mathbf{n}$. For example, choosing $\mathbf{n} = (\mathbf{0}, \mathbf{0}, \mathbf{1})$ as the upward pointing vector, F^5 is smallest for horizontally oriented surfaces, while F^6 is smallest for vertically oriented surfaces. These features are used among others to distinguishing the ground surface from other objects.

Features F^7 and F^8 measure the absolute cosine distance between the third eigenvector and the vector ℓ, which is assumed to be of unit length. They can be used for measuring the direction of elongated structures. F^7 is smallest if the elongated structure is oriented parallel to ℓ and F^8 is smallest if the structure is oriented perpendicular to ℓ. The features can for instance be used for distinguising vertically oriented structures, like flag polos, from horisontally oriented structures, like power cables.

In general, the surface normal vector is insufficient for classifying the ground surface reliably in more challenging conditions, such as very hilly terrain. Additional features will be derived from point patches using some different neighborhood systems

$$\mathcal{N}_{top}(\mathbf{p}) = \{\mathbf{q} \in V \text{ s.t.} \sqrt{(x_1(\mathbf{p}) - x_1(\mathbf{q}))^2 + (x_2(\mathbf{p}) - x_2(\mathbf{q}))^2} \leq R_{top}\},$$

$$\mathcal{N}_{img}(\mathbf{p}) = \{\mathbf{q} \in V \text{ s.t.} \sqrt{(\theta(\mathbf{q}) - \theta(\mathbf{p}))^2 + (\phi(\mathbf{q}) - \phi(\mathbf{p}))^2} \leq R_{img} \text{ and } \theta(\mathbf{q}) \geq \theta(\mathbf{p})\}.$$

Here $\mathcal{N}_{top}(\mathbf{p})$ is all points within in a vertically oriented cylinder in 3D space surrounding point $(\mathbf{p})$. $\mathcal{N}_{img}(\mathbf{p})$ is all points within a upper half ball in the image plane surrounding point $\mathbf{p}$.

$$F^9(\mathbf{p}) = x_3(\mathbf{p}) - h^*(\mathbf{p}), \qquad F^{10}(\mathbf{p}) = h^*(\mathbf{p}) - x_3(\mathbf{p}), \tag{28}$$

$$F^{11}(\mathbf{p}) = d(\mathbf{p}) - d^*(\mathbf{p}), \qquad F^{12}(\mathbf{p}) = d^*(\mathbf{p}) - d(\mathbf{p}). \tag{29}$$

Features $F^9(\mathbf{p})$ and $F^{10}(\mathbf{p})$ measure the relative difference between the height cordinate $x_3(\mathbf{p})$ and a local estimate of the height of the ground surface $h^*(\mathbf{p})$. The height estimate is calculated for each point $\mathbf{p}$ as the smallest height among all points in its 2D neighborhood $\mathcal{N}_{top}(\mathbf{p})$, i.e.: $h^*(\mathbf{p}) = \min_{\mathbf{q}\in\mathcal{N}_{top}(\mathbf{p})} x^3(\mathbf{q}) + \xi_h$, where $\xi_h > 0$ is an error tolerance. $F^9(\mathbf{p})$ is expected to take smaller values for the ground than raised objects, and vice versa for $F^{10}(\mathbf{p})$.

Another characteristic of the ground surface is that it is expected to be located farther behind than raised objects, within the upper local neighborhood $\mathcal{N}_{img}(\mathbf{p})$ of points in the image plane. Features $F^{11}(\mathbf{p})$ and $F^{12}(\mathbf{p})$ measure the relative difference between the distance to the points $d(\mathbf{p})$ and a local estimate of the distance to the ground surface $d^*(\mathbf{p})$. The distance estimate is calculated as the largest distance in the neighborhood $\mathcal{N}_{img}(\mathbf{p})$, i.e. $d^*(\mathbf{p}) = \max_{\mathbf{q}\in\mathcal{N}_{img}(\mathbf{p})} d(\mathbf{q}) - \xi_d$, where $\xi_d > 0$ is an error tolerance. $F^{11}(\mathbf{p})$ is expected to take smaller values for raised objects, especially around their edges, than for the ground. Vice versa, $F^{12}(\mathbf{p})$ is expected to take smaller values for the ground than raised objects. These features are especially useful for hilly terrain and slant view angles.

Some object classes, like roads, are very difficult to distinguish from the ground surface purely based on the point geometries, especially from long ranges where the accuracy of point coordinates decreases. The combination of an IR sensor and ladar makes it possible to segment the roads from the rest of the scene. Roads can be characterized by the same features as the ground surface, but their points show up as distinct clusters in the higher-dimensional space due to their locally similar intensities and other properties like elongation. Some of the characteristics can also be endoded in the node weights. For a 1-channel image, like a LWIR image, the fused data points are 4-dimensional. Elongated structures have a large fourth eigenvalue compared to the third, which can be expressed by the features

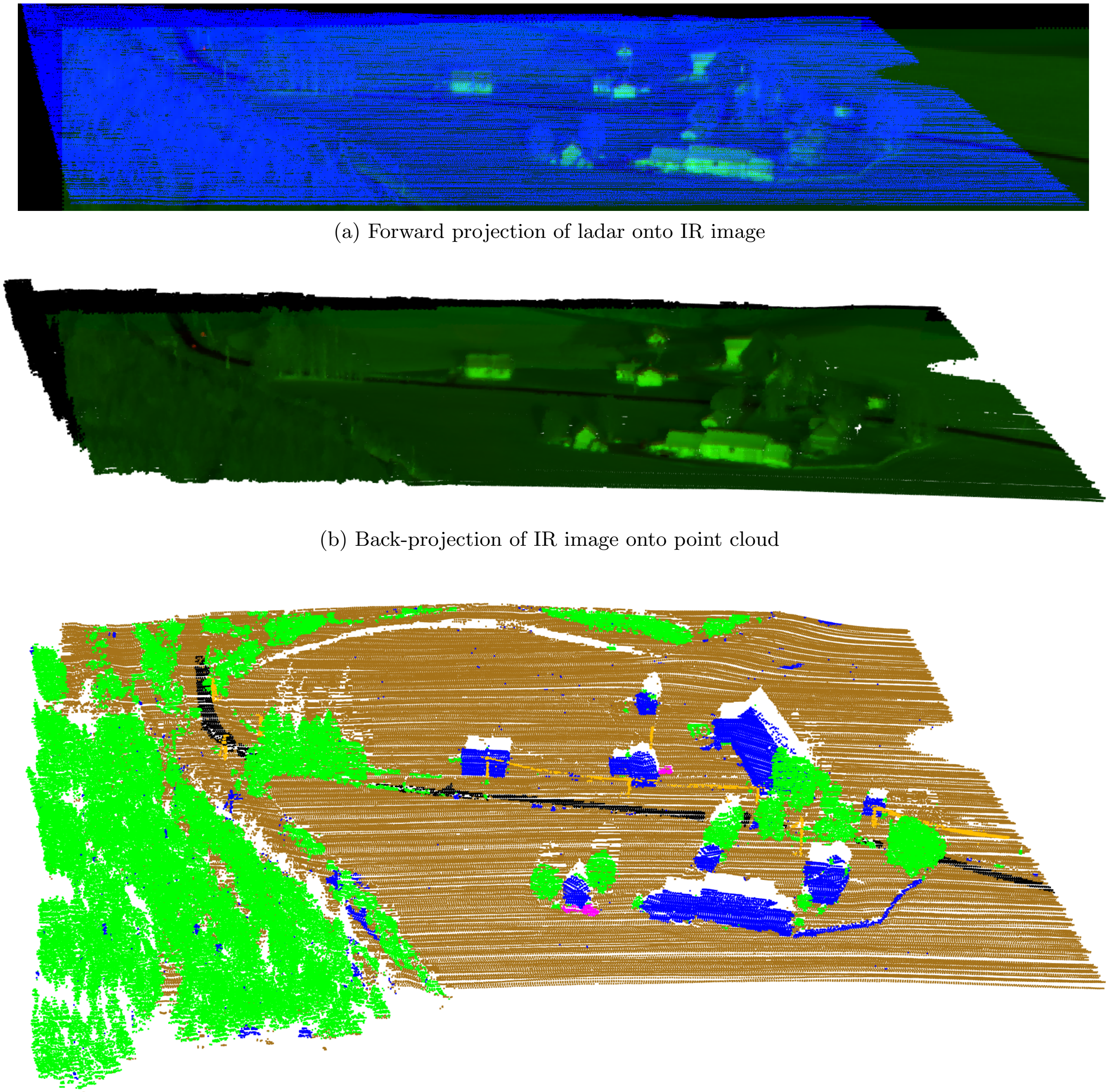
$$F^{13}(\mathbf{p}) = \gamma^3\lambda^3(\mathbf{p}) - \lambda^4(\mathbf{p}), \quad F^{14}(\mathbf{p}) = \lambda^4(\mathbf{p}) - \gamma^3\lambda^3(\mathbf{p}), \tag{30}$$

where $\gamma^3 > 1$. Here $F^{13}(\mathbf{p})$ is expected to be smallest at roads while $F^{14}(\mathbf{p})$ is expected smallest at other parts of the ground surface.

3.4.1 Example results

Example results using IR images and 3D point clouds from a ladar are shown in Figure 10 - 12, where outdoor scenes are segmented into different object classes. Visualizations of the scenes and of the fused data structures using forward and backward projection are shown in (a) and (b). In Figure 11 and 12, we also show comparisons against results on pure ladar data in subfigures (c), where a red X is depicted above some misclassifications. The points are colored by their assigned class membership, where green indicate vegetation, blue indicate buildings, brown indicate the ground surface, purple indicate vehicles, orange indicate thin strucutures, and black indicate roads. In addition there is a class of noise points that is omitted from visualization.

In Figure 10 the roads and vehicles are not easily discrernable in the pure ladar data, but get segmented with additional data from the passive IR sensor. On the other hand, the power lines, masts and flag pole are not visible in the IR images, but get detected and segmented based on their thin structures in the ladar data. The scenes in Figure 11 and 12 are challenging to segment for a number of reasons. The hilly terrain with very sharp cliffs at certain places makes it more difficult to distinguish the ground surface from other objects. The low altitude observation angle causes a lot of occlusions. The far observation range leads to low spatial resolutions and poor signals. The class memberships of some parts of the scenes are ambiguous when observed through each sensor data separately. On pure ladar data, the scene can be segmented into ground, buildings, vegetation and thin structures with reasonable accuracy using the proposed improvements to our previous methods.[5,7] There are some misclassification of parts of the buildings in the middle of Figure 11 (c) and the bottom of Figure 12(c) in addition to some trees and parts of the ground. On fused IR and ladar data the accuracy of segmentation increases, avoiding the aforementioned misclassifications, while also making it possible to segment roads into a separate class.

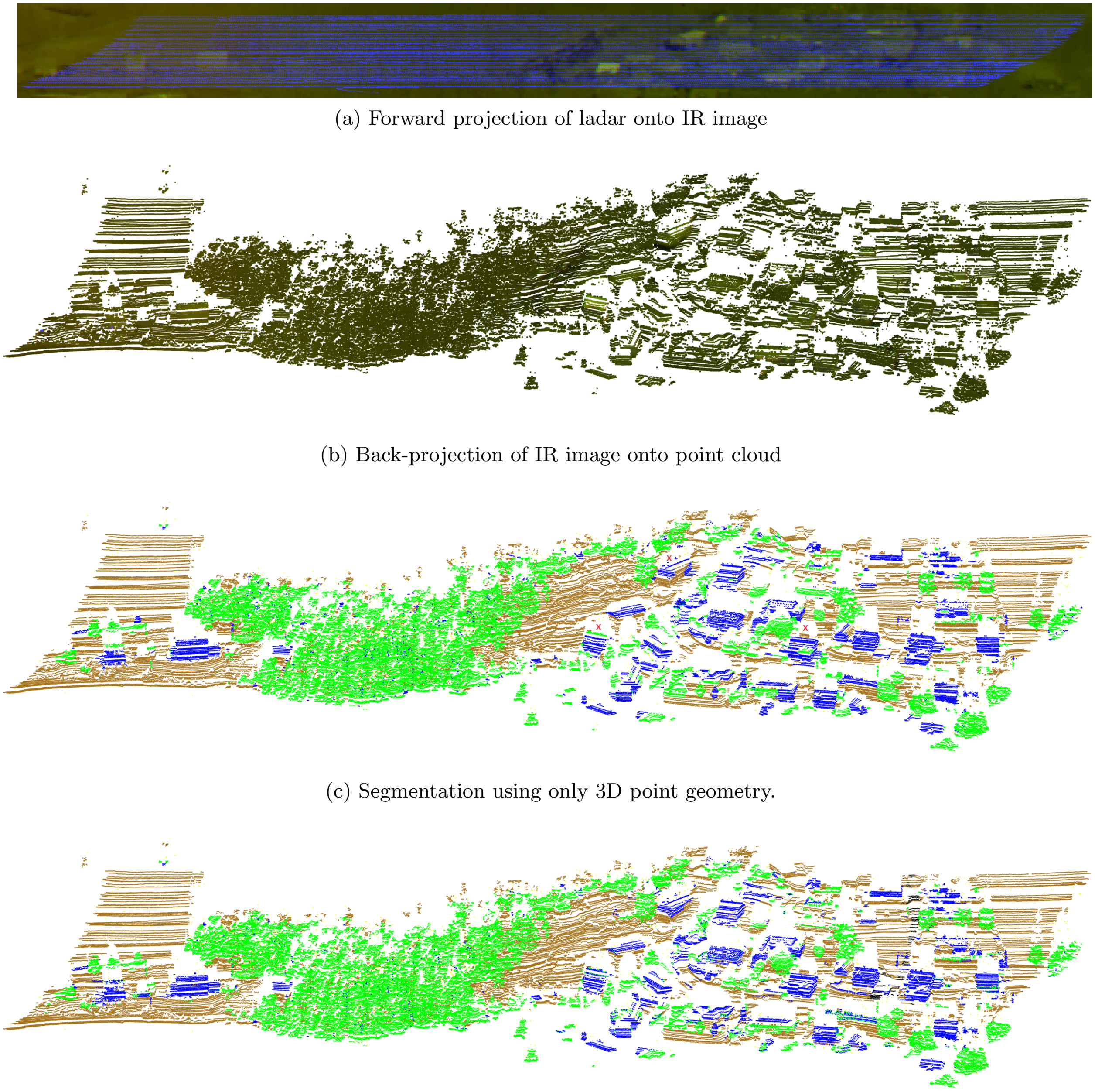


(a) Forward projection of ladar onto IR image

(b) Back-projection of IR image onto point cloud

(c) Segmentation using fused IR and 3D point geometry

Figure 10: Segmentation of a scene into different object classes based on fused IR and ladar data. The fused data is visualized in (a) and (b). Segmentation results are shown in (c). The scene is segmented into vegetation (green), buildings (blue), thin structures (orange), vehicles (purple), ground (brown), roads (black) and noise (removed from visualization).



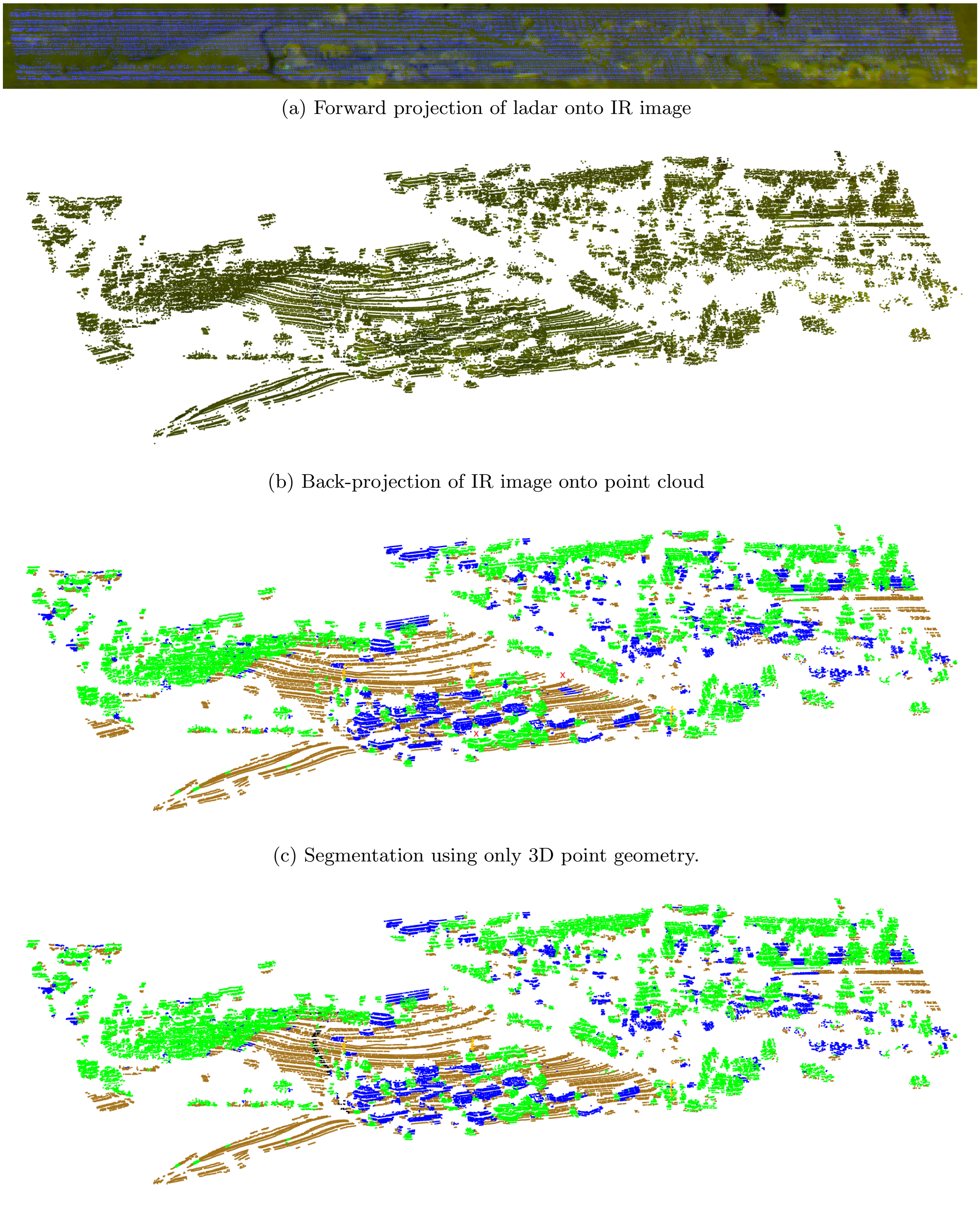
(a) Forward projection of ladar onto IR image

(b) Back-projection of IR image onto point cloud

(c) Segmentation using only 3D point geometry.

(d) Segmentation using fused IR and 3D point geometry

Figure 11: Segmentation of a complex scene into different object classes based on fused IR and 3D data (d) compared with pure ladar data (c). Visualization of fused data using forward and backward projection are shown in (a) and (b). The scene is segmented into vegetation (green), buildings (blue), thin structures (orange) ground (brown), roads (black) and noise (removed from visualization). The terrain is very hilly, with almost 90 degree cliffs at certain places. Combined with a low altitude observation angle and far observation distance this makes the scene challenging. A red X is depicted over some misclassifications on the pure ladar data in (c). Fused data improves the segmentation performance (d), particularly of some of the buildings, and makes it possible to distinguish roads from the rest of the ground surface.



(a) Forward projection of ladar onto IR image

(b) Back-projection of IR image onto point cloud

(c) Segmentation using only 3D point geometry.

(d) Segmentation using fused IR and 3D point geometry

Figure 12: Segmentation of a complex scene into vegetation (green), buildings (blue), vehicles (purple), thin structures (orange) ground (brown) and roads (black), and noise (removed from visualization). The segmentation performance on fused data (d) is generally improved compared to pure ladar data (c), such as preventing some misclassifications (a red X is depicted above some of them in (c)), and making it possible to distinguish roads from the rest of the ground surface.

4. CONCLUSIONS

This paper introduced a sensor fusion framework that combined passive and active electro-optical data to improve scene segmentation and object recognition in challenging environments. It created fused data structures that merged 3D coordinates and intensities from 2D images and then adopted and specialized methods for classification of high-dimensional data to segment the fused data structures. The general framework could be used for different applications: Weak laser return pulses could be separated from noise by extracting point clusters in the high dimensional data under certain constraints, exemplified on a maritime vessel observed from very long range through thick atmosphere; Outdoor scenes observed from slant view angles and long ranges could be segmented into different object classes with greater accuracy and class granularity that our previous work that utilized pure ladar data. Methods that corrected for motion induced distortions and misalignments between the sensor data were also described and utilized for improved object recognition. A detailed treatment of object recognition using fused data is the subject of a future work.

REFERENCES

1. A. L. Bertozzi and A. Flenner, “Diffuse interface models on graphs for classification of high dimensional data,” *SIAM journal on Multiscale Model. Simul.* **10**(3), pp. 1090–1118, 2012.
2. H. Hu, T. Laurent, M. A. Porter, and A. L. Bertozzi, “A method based on total variation for network modularity optimization using the MBO scheme,” *SIAM Journal of Applied Mathematics* **73**(6), pp. 2224–2246, 2013.
3. E. Merkurjev, E. Bae, A. L. Bertozzi, and X.-C. Tai, “Global binary optimization on graphs for classification of high-dimensional data,” *Journal of Mathematical Imaging and Vision* **52**(3), pp. 414–435, 2015.
4. C. Garcia-Cardona, E. Merkurjev, A. Bertozzi, A. Flenner, and A. Percus, “Multiclass data segmentation using diffuse interface methods on graphs,” *IEEE Transactions on Pattern Analysis and Machine Intelligence* **36**(8), pp. 1600–1613, 2014.
5. E. Bae and E. Merkurjev, “Convex variational methods on graphs for multiclass segmentation of high-dimensional data and point clouds,” *Journal of Mathematical Imaging and Vision* **58**(3), pp. 468–493, 2017.
6. Z. Boyd, E. Bae, X.-. Tai, and A. Bertozzi, “Simplified energy landscape for modularity using total variation,” *SIAM Journal on Applied Mathematics* **78**(5), pp. 2439–2464, 2018.
7. E. Bae, “Automatic scene understanding and object identification in point clouds,” *Proc. SPIE* **11160**, Electro-Optical Remote Sensing XIII, 111600M, 2019.
8. E. Bae, “Automatic object recognition within point clouds in clustered or scattered scenes,” *Proc. SPIE* **11538**, Electro-Optical Remote Sensing XIV, 1153804, 2020.
9. P. An, T. Ma, K. Yu, B. Fang, J. Zhang, W. Fu, and J. Ma, “Geometric calibration for lidar-camera system fusing 3D-2D and 3D-3D point correspondences,” *Optical Express* **28**, pp. 2122–2141, Jan 2020.
10. J. D. Choi and M. Y. Kims, “A sensor fusion system with thermal infrared camera and lidar for autonomous vehicles and deep learning based object detections,” *ICT Express* **9**(2), pp. 222–227, 2023.
11. F. Lozes, A. Elmoataz, and O. Lezoray, “Partial difference operators on weighted graphs for image processing on surfaces and point clouds,” *IEEE Transactions on Image Processing* **23**, pp. 3896–3909, 2014.
12. G. H. Lee, J. D. Choi, J. H. Lee, and M. Y. Kim, “Object detection using vision and lidar sensor fusion for multi-channel v2x system,” in *2020 International Conference on Artificial Intelligence in Information and Communication (ICAIIC)*, pp. 1–5, 2020.
13. E. Bae, “Velocity estimation and recognition of moving objects from a single ladar image,” *Proc. SPIE* **11866**, Electro-Optical and Infrared Systems: Technology and Applications XVIII and Electro-Optical Remote Sensing XV, 118660W, 2021.
14. D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Y. Ng, “Discriminative learning of markov random fields for segmentation of 3D scan data,” in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Diego, CA, USA*, pp. 169–176, 2005.
15. H. C. Palm, T. Haavardsholm., H. Ajer, and C. V. Jensen, “Extraction and classification of vehicles in ladar imagery,” *Proc. SPIE* **8731**, Laser Radar Technology and Applications XVIII, 873102, 2013.