METHOD FOR DETECTING AN OBJECT AND METHOD FOR TRAINIING A DETECTION NEURAL NETWORK

TECHNICAL FIELD

[0001] Various embodiments relate to methods for detecting an object, and methods for training a detection neural network.

BACKGROUND

[0002] Object detection is a key capability for advanced driver assistance systems and autonomous vehicles, as it is required in a multitude of applications such as collision avoidance and navigation. Object detection may be accurately performed on dense three-dimensional (3D) point clouds, such as point clouds generated by LiDAR sensors. LiDAR sensors are expensive and as such, there is a demand to generate 3D point clouds using more affordable sensors such as cameras. Pseudo-LiDAR techniques uses neural networks to convert two-dimensional images to 3D point clouds. While the pseudo-LiDAR technique is able to generate 3D point clouds using images captured by cameras, the depth estimation in the 3D point clouds may be inaccurate, especially for objects that are partially occluded or far-away in the images. The inaccuracies in the depth estimation, in turn, affects the accuracy of object detections.

[0003] In view of the above, there is a need for an improved method for detecting objects using automotive sensors, that can address at least some of the abovementioned problems.

SUMMARY

[0004] According to various embodiments, there is provided a computer-implemented method for detecting an object. The method includes extracting features from an image. The method further includes determining, for each pixel of the image, a depth estimation, based on the extracted features to generate a depth map. The method further includes determining, for each pixel of the image, an uncertainty value of a respective depth estimation, to generate a depth uncertainty map. The method further includes generating a point cloud representation of the image. Each data point in the point cloud includes a respective depth estimation and a corresponding uncertainty value. The method further includes detecting an object represented in the image based on the point cloud representation, using a detection neural network.

[0005] According to various embodiments, there is provided a computer-implemented method for training a detection neural network that includes the above-mentioned method for detecting an object. The method for training the detection neural network further includes generating detection data indicative of objects detected by the detection neural network, computing a loss function based on comparing the detection data to ground truth data indicative of objects represented in the image, and adjusting weights of the detection neural network based on the loss function.

[0006] According to various embodiments, there is provided a computer program. The computer program includes instructions, which, when the program is executed by a computer, cause the computer to carry out the above-mentioned method for detecting an object or the above-mentioned computer-implemented method for training a detection neural network.

[0007] According to various embodiments, there is provided a data carrier signal that carries the above-mentioned computer program.

[0008] Additional features for advantageous embodiments are provided in the dependent claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0009] In the drawings, like reference characters generally refer to the same parts throughout the different views. The drawings are not necessarily to scale, emphasis instead generally being placed upon illustrating the principles of the invention. In the following description, various embodiments are described with reference to the following drawings, in which:

[0010] FIG. 1 shows a simplified schematic diagram of an object detection method according to a first embodiment.

[0011] FIG. 2 shows a simplified schematic diagram of an object detection method according to a second embodiment.

[0012] FIG. 3 shows a simplified hardware block diagram of a vehicle according to various embodiments.

[0013] FIG. 4 shows a flow diagram of a computer-implemented method for detecting an object according to various embodiments.

[0014] FIG. 5 shows a flow diagram of a computer-implemented method for training a detection neural network according to various embodiments.

[0015] FIG. 6 shows a simplified block diagram of the computer program 600 according to various embodiments.

DESCRIPTION

[0016] Embodiments described below in context of the devices are analogously valid for the respective methods, and vice versa. Furthermore, it will be understood that the embodiments described below may be combined, for example, a part of one embodiment may be combined with a part of another embodiment.

[0017] It will be understood that any property described herein for a specific device may also hold for any device described herein. It will be understood that any property described herein for a specific method may also hold for any method described herein. Furthermore, it will be understood that for any device or method described herein, not necessarily all the components or steps described must be enclosed in the device or method, but only some (but not all) components or steps may be enclosed.

[0018] The term "coupled" (or "connected") herein may be understood as electrically coupled or as mechanically coupled, for example attached or fixed, or just in contact without any fixation, and it will be understood that both direct coupling or indirect coupling (in other words: coupling without direct contact) may be provided.

[0019] In this context, the device as described in this description may include a memory which is for example used in the processing carried out in the device. A memory used in the embodiments may be a volatile memory, for example a DRAM (Dynamic Random Access Memory) or a non-volatile memory, for example a PROM (Programmable Read Only Memory), an EPROM (Erasable PROM), EEPROM (Electrically Erasable PROM), or a flash memory, e.g., a floating gate memory, a charge trapping memory, an MRAM (Magnetoresistive Random Access Memory) or a PCRAM (Phase Change Random Access Memory).

[0020] In order that the invention may be readily understood and put into practical effect, various embodiments will now be described by way of examples and not limitations, and with reference to the figures.

[0021] According to various embodiments, a computer-implemented method for detecting an object may be provided. In various contexts, the method for detecting an object may also be interchangeably referred to, as an object detection method. The method may include generating a point cloud based on camera images, for example, using pseudo-LiDAR techniques. The method may include determining an uncertainty value, or a confidence level, of the depth estimations determined for the point cloud. The point cloud may be modified to include the

uncertainty values of the depth estimates as one of the dimensions of the point cloud, to result in a modified point cloud. The modified point cloud may be fed into a detection neural network that is configured to detect objects. The detection neural network may be trained using point clouds that include the depth uncertainty values as an added dimension, in addition to other spatial dimensions such as depth, horizontal position and vertical position. Existing state-of-the-art detection neural network may be used to detect objects in the modified point cloud. The method may improve the object detection accuracy over existing object detection using pseudo-LiDAR point clouds, while being more cost-effective over object detection using only LiDAR-generated point clouds.

[0022] FIG. 1 shows a simplified schematic diagram of an object detection method 100 according to various embodiments. The object detection method 100 may include extracting features from an input dataset 110 using a feature extractor. The features extracted from the input dataset 110 are also referred herein as extracted features 102. The input dataset 110 may include an image. The image may be generated by an imaging sensor such as a camera. The process of extracting features may retrieve from the image, information that is relevant for determining depth and for detection of objects. For example, the extracted features 102 may include information indicative of at least one of size, shape, position and color. The process of extracting the features may be carried out using an imaging processing algorithm, or may be performed using a machine learning model.

[0023] A depth map generator may generate a depth map 104 based on the extracted features 102. A respective depth map may be generated for each image in the input dataset 110. The depth map 104 may include a depth estimation for each data point, such as a pixel, of an image in the input dataset 110. In addition, the depth map generator may also determine an uncertainty value of each depth estimation, and output it as a depth uncertainty map 108. The depth uncertainty map 108 indicates a confidence level of each depth estimate in the depth map 104.

[0024] A point cloud generator may then generate a point cloud 106 based on the depth map 104 and the depth uncertainty map 108. The point cloud 106 may be generated further based on the input dataset 110. The point cloud generator may assign a depth estimation and its corresponding uncertainty value, to each data point of the image. The resulting point cloud 106 may include a three-dimensional representation of the image. In various contexts, the point cloud 106 may be referred to as a pseudo-LiDAR point cloud.

[0025] A detection neural network may generate an object detection 120 based on the point cloud 106. As the point cloud generator also receives information on the depth uncertainties, it

is able to assign appropriate weights in its consideration of the depth estimations to improve the detection accuracy.

[0026] ,

[0027] FIG. 2 shows a simplified schematic diagram of an object detection method 300 according to various embodiments. The object detection method 300 may include the object detection method 100. Similar to the object detection method 100, an input dataset 110 may be fed to a feature extractor configured to extract features from the input dataset 110. The input dataset 110 may include a plurality of images. The plurality of images may be generated by various different cameras. Next, at least one other neural network may receive the extracted features 102, and may generate multiple outputs. The outputs may include depth estimates in the form of depth map 104, depth uncertainty values in the form of depth uncertainty map 108, and instance segmentation 304. The instance segmentation 304 may indicate an outline or shape of a target object. A depth map filtering and masking module 302 may use the instance segmentation as a region of interest to mask the depth map 104. For example, if a pixel is not part of any object instance indicated by the instance segmentation output, the depth information at the pixel may be masked as invalid. Invalid depth information is not used to construct the point cloud, and not used by the object detector.

[0028] The depth map filtering and masking module 302 may also use bounding boxes, to mask the depth map 104, as an alternative or in addition, to the instance segmentation. The bounding boxes may be two-dimensional polygons for defining regions of interest. The bounding boxes may be sized and/or shaped based on the known dimensions of the target. The depth map filtering and masking module 302 may also filter the depth map based on the depth uncertainty map 108. For example, a predefined uncertainty threshold may be set, and the depth estimates in the depth map having a corresponding uncertainty value beyond the predefined uncertainty threshold may be removed, i.e. reset the depth value to a predefined value such as zero or infinity. This removes unreliable data points, which may distort the object detection results.

[0029] The filtered depth map, also interchangeably referred herein as modified depth map, may be used to generate a point cloud 106. The depth uncertainty values from the depth uncertainty map 108 may also be added to the point cloud 106. The point cloud 106 may include depth estimations and corresponding uncertainty values. The depth uncertainty values may be added to the point cloud 106 as a dimension in addition to data representing three-dimensional spatial position.

[0030] Each image in the input dataset 110 may go through the same processes to result in a respective point cloud 106. The plurality of point clouds 106 may be fused into a combined point cloud 306. The point clouds 106 may first be transformed to a common coordinate system, before being fused together. For data points in the point clouds 106 with multiple depth estimate values from more than one point cloud 106, the depth estimate with the lowest uncertainty value may be chosen while the depth estimates with higher uncertainty values may be discarded.

[0031] Combining the data in the plurality of images in the input dataset 110 at the point cloud layer, instead of at earlier stages, provides the advantage that the method 300 can be a universal approach for different camera set ups. For example, the camera set up may be any one of mono camera, stereo camera, or multiple cameras forming a surround view. Further, the method 300 may be scalable to process more images, or data from more cameras.

[0032] The combined point cloud 306 having depth uncertainty as one of its dimensions, may be provided to the object detector. The object detector may include a neural network trained to learn the object detection end-to-end with multiple loss combined, including depth loss, instance segmentation loss, and object detection loss.

[0033] According to various embodiments, the method 300 may further include receiving LiDAR data 250 from a LiDAR sensor. The LiDAR data 250 may be converted into a LiDAR-generated point cloud. The LiDAR-generated point cloud may be fused together with the point cloud 106 that is generated based on the camera images, before the combined point cloud 306 is fed into the detection neural network. Camera images may be higher in data resolution than LiDAR data, while LiDAR data may have more accurate depth information. Therefore, generating the combined point cloud 306 using a hybrid LiDAR-camera inputs may result in a dense and accurate combined point cloud 306, that enables the detection neural network to detect objects with improved accuracy.

[0034] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the object detection method 100 or 300 may further include extracting features from a further image. The object detection method 100 or 300 may further include determining, for each pixel of the further image, an uncertainty value of a respective depth estimation, to generate a further depth uncertainty map. The object detection method 100 or 300 may further include generating a further point cloud representation of the further image, wherein each data point in the further point cloud representation comprises the respective depth estimation and the corresponding uncertainty

value, and detecting the object represented in the image and the further image, further based on the further point cloud representation, using the detection neural network.

[0035] In other words, the object detection method 100 or 300 may include processing multiple images to generate respective point cloud representations that include depth estimations and uncertainty value of the depth estimations, and detecting an object based on the multiple point cloud representations. The image and the further image may capture a common scene from different perspectives. Having multiple images may provide the advantage that any obscuration in one of the images may be compensated by another image so that the object detection accuracy may be improved. For example, a target object may be occluded in the field of view of a first camera that generates the image, but the target object may be visible in the field of view of a second camera that generates the further image. As a result, the depth estimation of the target object, as obtained from the further image, may be more accurate than the depth estimation of the target object as obtained from the image.

[0036] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the object detection method 100 or 300 may further include generating a combined point cloud representation based on the further point cloud representation and the point cloud representation, and detecting the object by inputting the combined point cloud representation into the detection neural network. The image and the further image may be taken from different viewpoints, and as such the combined point cloud representation has an enlarged field of view.

[0037] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, generating the combined point cloud representation may include transforming the further point cloud representation to a common coordinate system as the point cloud representation, resulting in a transformed further point cloud representation, and fusing the transformed further point cloud representation with the point cloud representation to result in the combined point cloud representation. The transformation places both point cloud representations in the same reference frame, so that their data points may be merged. This prevents inaccuracies resulting from, for example, a single object being detected at two different locations. The same network architecture may thus be used for different input set ups, for example, various arrangements of cameras and types of cameras, as the resulting point cloud representations are adjusted to a common reference and therefore may be merged together for object detection.

[0038] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, generating the combined point

cloud representation may include fusing the further point cloud representation and the point cloud representation based on the depth uncertainty map and the further depth uncertainty map. For each data point in the combined point cloud representation, the depth estimation with the lower uncertainty value may be chosen over the depth estimation with the higher uncertainty value, so that the depth estimate in the combined point cloud representation is improved.

[0039] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the object detection method 100 or 300 may further include masking the depth map based on bounding boxes, to generate a modified depth map. The masking of the depth map may narrow down the number of objects that may be identified as the target. Using bounding boxes for masking the depth map may provide the advantage of fast inference speed.

[0040] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the object detection method 100 or 300 may include performing instance segmentation based on the extracted features, to generate instance segmentation masks, and masking the depth map based on the instance segmentation masks, to generate a modified depth map. Using instance segmentation masks to mask the depth map may result in more accurate object shape detection, as compared to, for example using bounding boxes to mask the depth map.

[0041] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, a single multi-tasking neural network performs the instance segmentation, the determining of the depth estimations and the determining of the uncertainty values. This provides the technical advantage of reducing the time and data required to train the neural network, as the neural network is simultaneously trained to perform multiple functions. Combining these tasks into a single neural network reduces computation and enables the object detection to be run in real-time.

[0042] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the multi-tasking neural network is trained with a sample input dataset that includes a plurality of training images, and with training data that includes ground truth spatial information of objects represented in the plurality of training images.

[0043] According to an embodiment which may be combined with the above-described embodiment or with any below described further embodiment, the neural network may include an encoder-decoder based network architecture.

[0044] The neural network may include an encoder that is configured to extract features from the input dataset 110, i.e. perform functions of the feature extractor. The machine learning model may include a plurality of decoders. Each decoder of the plurality of decoders may perform a specific task, for example, a respective decoder may generate each of depth map, depth map uncertainty values and instance segmentation. An example of the encoder-decoder architecture is disclosed in [1] "Multi-Task Learning Using Uncertainty to Weigh Losses for Scene Geometry and Semantics" by Kendall et. al. which is incorporated herein by reference. The encoder may include a backbone followed by a feature extension module. For example, the backbone may be ResNet101 and the feature extension module may be Atrous Spatial Pyramid Pooling (ASPP). The ASPP feature extension may increase contextual awareness of the encoder. In another example, [2] "Monocular 3D Object Detection and Box Fitting Trained End-to-End Using Intersection-over-Union Loss" by Jorgensen et. al. discloses another encoder where the backbone is ResNet34 or Dilated Residual Network (DRN) and the feature extension module is Feature Pyramid Network (FPN) that extends the feature map at different scales.

[0045] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the object detection method 100 or 300 may include filtering the depth estimations based on their respective uncertainty values, to generate a modified depth map. The modified depth map may be more accurate than the pre-filtered depth map, as the low quality depth estimations have been removed. Further, by reducing the less relevant data points, the inference speed of object detection may increase as the object detector processes less data.

[0046] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the point cloud representation of the image is generated based on at least the modified depth map. By doing so the resulting point cloud representations only include high quality depth estimations. This may improve the object detection accuracy.

[0047] According to an embodiment which may be combined with any above-described embodiment or with any below described further embodiment, the detection neural network is trained using a plurality of training point cloud representations generated based on a plurality of training images, wherein each data point in each training point cloud representation comprises a depth estimation and a corresponding uncertainty value of the depth estimation, and further using ground truth data that is indicative of objects represented in the plurality of training images.

[0048] FIG. 3 shows a simplified hardware block diagram of a vehicle 330 according to various embodiments. The vehicle 330 may be, for example, an autonomous mobile robot, a self-driving car, or a vehicle equipped with driver assistance capability. The vehicle 330 may include an object detection device 200. The object detection device 200 may include at least one processor 230. The object detection device 200 may include at least one memory 234. The at least one memory 234 may store a computer program product that includes instructions. These instructions may cause the processor 230 to perform the process steps of the object detection method 100. The at least one memory 234 may include a non-transitory computer-readable medium. The vehicle 330 may further include at least one sensor 232. The at least one sensor 232 may include a camera, and may further include a LiDAR sensor. The at least one processor 230, the at least one sensor 232 and the at least one memory 234 may be coupled to one another, for example, mechanically or electrically, via a coupling line 240.

[0049] According to various embodiments, the object detection device 200 may be part of a sensor device that may be installed onto the vehicle. In another example, the object detection device 200 may be part of a processing unit of a vehicle. The sensors onboard a vehicle may include cameras and LiDAR.

[0050] FIG. 4 shows a flow diagram of a computer-implemented method 400 for detecting an object according to various embodiments. The method 400 may include any one of the object detection methods 100 and 300. The method 400 may include processes 402, 404, 406 408 and 410. The process 402 may include extracting features from an image. The image may be part of the input dataset 110. The process 404 may include determining for each pixel of the image, a depth estimation, based on the extracted features 102 to generate a depth map 104. The depth map 104 may include respective depth estimations of pixels in the image. The process 406 may include determining, for each pixel of the image, an uncertainty value of a respective depth estimation, to generate a depth uncertainty map 108. The depth uncertainty map 108 may include respective uncertainty values of the depth estimation of pixels in the image. The process 408 may include generating a point cloud representation of the image. Each data point in the point cloud representation may include the respective depth estimation and the corresponding uncertainty value, as determined in the process 406. The process 410 may include detecting an object represented in the image based on the point cloud representation, using a detection neural network. The point cloud representation may include the point cloud 106. In another embodiment, the point cloud representation may include the combined point cloud 306.

[0051] FIG. 5 shows a flow diagram of a computer-implemented method 500 for training a detection neural network according to various embodiments. The method 500 may include

processes 502, 504 and 506. The process 502 may include generating detection data based on input training data. The process 502 may include the method 400, where the process 410 results in the detection data. The detection data may be indicative of objects detected by the detection neural network. The process 504 may include computing a loss function based on comparing the detection data to ground truth data indicative of objects represented in the image. The process 506 may include adjusting weights of the detection neural network based on the loss function. The ground truth data and the input training data may be part of a training dataset. Examples of suitable training dataset include nuScenes and KITTI 3D. The loss function may include localization loss, classification loss and direction loss. Examples of these loss functions are disclosed in [3] “PointPillars: Fast Encoders for Object Detection from Point Clouds” by Lang et. al. which is incorporated herein by reference, for example, in section 3.2 “Loss” of the document.

[0052] According to various embodiments, a data structure may be provided. The data structure may be generated by carrying out the steps of the method 500 for training a detection neural network. The data structure may include the detection neural network.

[0053] According to various embodiments, a computer program 600 may be provided. The computer program 600 may include instructions, which, when the computer program 600 is executed by a computer, cause the computer to carry out any one of the methods 100, 300, 400 or 500.

[0054] According to various embodiments, a data carrier signal may be provided. The data carrier signal may carry the computer program.

[0055] FIG. 6 shows a simplified block diagram of the computer program 600 according to various embodiments. The computer program 600 may include a feature extraction network 602, a point cloud generator 604 and a detection neural network 606. The computer program 600 may optionally further include at least one of a filtering and masking module 608 and a point cloud combiner 610. The feature extraction network 602 may be trained to output extracted features 102 based on images that it receives as input. The extracted features 102 may include depth estimates, depth uncertainty values and instance segmentations. The depth estimates may be arranged into depth maps 104. The depth uncertainty values may be arranged into depth uncertainty maps 108.

[0056] The feature extraction network 602 may include an encoder for encoding the input images, and at least one decoder. For example, the at least one decoder may include a depth decoder that generates depth estimates. The at least one decoder may include a further decoder that generates depth uncertainty values. The at least one decoder may include an instance

segmentation decoder that generates instance segmentation. The instance segmentation decoder may define which instance each pixel in the image belongs to, using an association to the instance's centroid. In an alternative embodiment, the depth decoder may determine both the depth estimates and the depth uncertainty values.

[0057] The losses from the plurality of decoders may be weighted and summed together to obtain a weighted combined loss. The feature extraction network 602 may adjust its outputs based on the weighted combined loss, to achieve multi-tasking learning.

[0058] An example of a multi-tasking feature extraction network 602 is disclosed in [1]. In [1], an example of the architecture may be found in Figure 1 and an example of the training of the neural network may be found in Appendix A of the document. An example of a suitable training dataset may include the KITTI depth evaluation dataset. The training method used may be supervised training.

[0059] In another example, the uncertainty values may be output by a Baynesian neural network. In a Baynesian neural network, the weights and biases of the network are treated as random variables, and their distribution is estimated during the training process. This allows for a probabilistic model interpretation and provides a measure of uncertainty in the predictions. An example of the Baynesian neural network is disclosed in [4] "Uncertainty Estimation for End-to-End Learned Dense Stereo Matching via Probabilistic Deep Learning" by Mehltretter which is incorporated herein by reference.

[0060] In another example disclosed in [2], the uncertainty values are determined based on the weights of each task net. Each task net is a decoder configured to perform a respective task, such as depth estimation and instance segmentation.

[0061] The point cloud generator 604 may perform a 2D-to-3D conversion from pixel to camera coordinates. The point cloud generator 604 may generate a pseudo-LiDAR point cloud based on camera images. The point cloud generator 604 may derive a 3D location of each pixel in a camera's coordinate system, and back project all the pixels into 3D coordinates to form the point cloud. The point cloud generator 604 may further convert the 3D coordinates to global coordinates, or the coordinates of another point cloud, so that the plurality of point clouds may be fused by the point cloud combiner 610.

[0062] An example of the point cloud generator 604 is disclosed in [5] "Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving" by Yan Wang et. al. which is incorporated herein by reference, for example, in paragraph "Pseudo-LiDAR generation" of the document.

[0063] The filtering and masking module 608 may be configured to mask or filter the depth map generated by the feature extraction network 602, using at least one of the instance segmentation and the depth uncertainty values that are also generated by the feature extraction network 602. The output of the filtering and masking module 608 is a smaller dataset that may be fed to the point cloud generator 604. By removing data points that either have high uncertainty values thereby indicating inaccuracy of the datapoints, or by masking irrelevant objects using the instance segmentation output, the point cloud generator 604 may be able to generate point clouds faster.

[0064] The point cloud combiner 610 may be configured to combine more than one point clouds generated by the point cloud generator 604. The point cloud combiner 610 may merge data points at the same location from different point clouds to a single data point, based on the depth uncertainty values of the respective data points. The point cloud combiner 610 may retain the data point with the lower uncertainty value and discard the data point with the higher uncertainty value. The uncertainty value is inversely proportional to a confidence level of the depth estimate. In other words, a low uncertainty value indicates that the depth estimate is likely to be accurate. For example, a first point cloud includes a first data point (x,y, z, uncertainty1) and a second point cloud includes a second data point (x, y, z, uncertainty2) at the same location. The combined point cloud will fuse the first data point and the second data point to (x, y, z, min(uncertainty1, uncertainty2)).

[0065] In a global coordinate system, a grid may be predefined with reference points at every unit distance at each of x, y and z directions. Data points in the point cloud where distance to the reference point is less than a predefined threshold may be fused to (xref, yref, zref, min(uncertainty)) where xref refers to reference point in the x direction, yref refers to reference point in y direction and zref refers to reference point in the z direction. Sometimes camera calibration may be inaccurate. For example, one object point can be projected differently by different cameras to different points in the global coordinate system. By fusing the data points to the reference points, such duplicate projections may be fused to a single data point.

[0066] The detection neural network 606 may receive a point cloud and may generate a 3D object detection output. The 3D object detection output may include information such as at least one of object class, dimension, location and orientation. The detection neural network 606 may include for example, PointPillars. An example of the detection neural network 606 is disclosed in [3].

[0067] The detection neural network 606 may be trained end-to-end, together with the feature extraction network 602. The training dataset may include camera images and camera

parameters including camera intrinsic and extrinsic. The camera parameters may be used for coordinate system transformation. The camera intrinsic may be used for back projecting a pixel in the image to a point in the camera coordinate system. In other words, the image pixel may be back-projected to a point in the point cloud 106 at least partially based on the camera intrinsic. The camera extrinsic may be used for transforming the points in the point cloud 106 from the camera coordinate system to a global coordinate system. For example, the point clouds 106 may be transformed based on the camera extrinsic to the global coordinate system, before they are combined to result in the combined point cloud 306.

[0068] . An example of a suitable training dataset is the KITTI object detection 3D dataset available at https://www.cvlibs.net/datasets/kitti/eval_object.php?obj_benchmark=3d.

[0069] The neural networks in the computer program 600 may be trained according to the following steps.

[0070] First, the feature extraction network 602 may be trained to determine depth, instance segmentation and depth uncertainty. Next, the feature extraction network 602 and the detection neural network 606 may be trained end-to-end.

[0071] As an example of how the feature extraction network 602 and the detection neural network 606 may be trained, a training dataset may be provided to the feature extraction network 602 and the following standard steps for training a general neural network may be followed.

[0072] Before training the detection neural network 606, the weights of the neural network may be randomly initialized to numbers between 0.01 and 0.1, while the biases are randomly initialize to numbers between 0.1 and 0.9.

[0073] Subsequently, the first observations of the dataset are loaded into the input layer of the neural network and the output value(s) is generated by forward-propagation of the input values of the input layers. Afterwards the following loss function is used to calculate loss with the output value(s):

[0074] Subsequently, the first observations of the dataset may be loaded into the input layer of the neural network and the output value(s) is generated by forward-propagation of the input values of the input layers. Afterwards the following loss functions may be used to calculate loss with the output value(s):

[0075] Mean Square Error (MSE): $MSE = \frac{1}{n}\sum(y - \hat{y})^2$, where n represents the number of Cross entropy output. In other words, $y - \hat{y}$ represents the difference between actual and

predicted output. The MSE loss function may be applied to determine loss for regression output, such as object location, dimension and orientation.

[0076] Focal loss (FL): $FL(p_t) = -\alpha_t(1-p_t)^{\gamma} \log p_t$, where p_t represents estimated probability of class t, α_t represents weighting factor, and γ represents a tunable focusing parameters. An example of the definition of focal loss is disclosed in [6] “Focal Loss for Dense Object Detection” by Lin et. al. Another example of the definition of focal loss is disclosed in [3]. The focal loss function may be applied to determine object classification loss.

[0077] The weights and biases may subsequently be updated by an AdamOptimizer with a learning rate of 0.001. Other parameters of the AdamOptimizer may be set to default values. For example:

beta_1 = 0.9

beta_2 = 0.999

eps = 1e-08

weight_decay = 0

[0078] The steps described above may be repeated with the next set of observations until all the observations are used for training. This represents the first training epoch. This may be repeated until 10 epochs are completed.

[0079] While embodiments of the invention have been particularly shown and described with reference to specific embodiments, it should be understood by those skilled in the art that various changes in form and detail may be made therein without departing from the spirit and scope of the invention as defined by the appended claims. The scope of the invention is thus indicated by the appended claims and all changes which come within the meaning and range of equivalency of the claims are therefore intended to be embraced. It will be appreciated that common numerals, used in the relevant drawings, refer to components that serve a similar or the same purpose.

[0080] It will be appreciated to a person skilled in the art that the terminology used herein is for the purpose of describing various embodiments only and is not intended to be limiting of the present invention. As used herein, the singular forms "a", "an" and "the" are intended to include the plural forms as well, unless the context clearly indicates otherwise. It will be further understood that the terms "comprises" and/or "comprising," when used in this specification, specify the presence of stated features, integers, steps, operations, elements, and/or

components, but do not preclude the presence or addition of one or more other features, integers, steps, operations, elements, components, and/or groups thereof.

[0081] It is understood that the specific order or hierarchy of blocks in the processes / flowcharts disclosed is an illustration of exemplary approaches. Based upon design preferences, it is understood that the specific order or hierarchy of blocks in the processes / flowcharts may be rearranged. Further, some blocks may be combined or omitted. The accompanying method claims present elements of the various blocks in a sample order, and are not meant to be limited to the specific order or hierarchy presented.

[0082] The previous description is provided to enable any person skilled in the art to practice the various aspects described herein. Various modifications to these aspects will be readily apparent to those skilled in the art, and the generic principles defined herein may be applied to other aspects. Thus, the claims are not intended to be limited to the aspects shown herein, but is to be accorded the full scope consistent with the language claims, wherein reference to an element in the singular is not intended to mean “one and only one” unless specifically so stated, but rather “one or more.” The word “exemplary” is used herein to mean “serving as an example, instance, or illustration.” Any aspect described herein as “exemplary” is not necessarily to be construed as preferred or advantageous over other aspects. Unless specifically stated otherwise, the term “some” refers to one or more. Combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” include any combination of A, B, and/or C, and may include multiples of A, multiples of B, or multiples of C. Specifically, combinations such as “at least one of A, B, or C,” “one or more of A, B, or C,” “at least one of A, B, and C,” “one or more of A, B, and C,” and “A, B, C, or any combination thereof” may be A only, B only, C only, A and B, A and C, B and C, or A and B and C, where any such combinations may contain one or more member or members of A, B, or C. All structural and functional equivalents to the elements of the various aspects described throughout this disclosure that are known or later come to be known to those of ordinary skill in the art are expressly incorporated herein by reference and are intended to be encompassed by the claims.

CLAIMS

1. A computer-implemented method (100, 300) for detecting an object, the method (100, 300) comprising:

extracting features from an image (402);

determining, for each pixel of the image, a depth estimation, based on the extracted features to generate a depth map (404);

determining, for each pixel of the image, an uncertainty value of a respective depth estimation, to generate a depth uncertainty map (406);

generating a point cloud representation of the image, wherein each data point in the point cloud representation comprises the respective depth estimation and the corresponding uncertainty value (408); and

detecting an object represented in the image based on the point cloud representation, using a detection neural network (410).

2. The method (100, 300) of any preceding claim, further comprising:

extracting features from a further image;

determining, for each pixel of the further image, an uncertainty value of a respective depth estimation, to generate a further depth uncertainty map;

generating a further point cloud representation of the further image, wherein each data point in the further point cloud representation comprises the respective depth estimation and the corresponding uncertainty value; and

detecting the object represented in the image and the further image, further based on the further point cloud representation, using the detection neural network.

3. The method (100, 300) of claim 2, further comprising:

generating a combined point cloud representation based on the further point cloud representation and the point cloud representation; and

detecting the object by inputting the combined point cloud representation into the detection neural network.

4. The method (100, 300) of claim 3, wherein generating the combined point cloud representation comprises

transforming the further point cloud representation to a common coordinate system as the point cloud representation, resulting in a transformed further point cloud representation, and

fusing the transformed further point cloud representation with the point cloud representation to result in the combined point cloud representation.

5. The method (100, 300) of any one of claims 3 to 4, wherein generating the combined point cloud representation comprises fusing the further point cloud representation and the point cloud representation based on the depth uncertainty map and the further depth uncertainty map.

6. The method (100, 300) of any preceding claim, further comprising:
masking the depth map based on bounding boxes, to generate a modified depth map.

7. The method of any preceding claim, further comprising:
performing instance segmentation based on the extracted features, to generate instance segmentation masks; and
masking the depth map based on the instance segmentation masks, to generate a modified depth map.

8. The method (100, 300) of claim 7, wherein a single multi-tasking neural network performs the instance segmentation, the determining of the depth estimations and the determining of the uncertainty values.

9. The method (100, 300) of claim 8, wherein the multi-tasking neural network is trained with a sample input dataset comprising a plurality of training images, and with training data comprising ground truth spatial information of objects represented in the plurality of training images.

10. The method (100, 300) of any preceding claim, further comprising:
filtering the depth estimations based on their respective uncertainty values, to generate a modified depth map.

11. The method (100, 300) of any one of claims 7 to 10, wherein the point cloud representation of the image is generated based on at least the modified depth map.

12. The method (100, 300) of any preceding claim, wherein the detection neural network is trained using a plurality of training point cloud representations generated based on a plurality of training images, wherein each data point in each training point cloud representation comprises a depth estimation and a corresponding uncertainty value of the depth estimation, and further using ground truth data that is indicative of objects represented in the plurality of training images.

13. A vehicle comprising an object detection device (200), wherein the object detection device (200) is configured to carry out the method (100, 300) of any preceding claim.

14. A computer-implemented method (500) for training a detection neural network (606), the method (500) comprising:
generating detection data indicative of objects detected by the detection neural network, wherein generating the detection data comprises
extracting features from an image (402);
determining, for each pixel of the image, a depth estimation, based on the extracted features to generate a depth map (404);
determining, for each pixel of the image, an uncertainty value of a respective depth estimation, to generate a depth uncertainty map (406);
generating a point cloud representation of the image, wherein each data point in the point cloud representation comprises the respective depth estimation and the corresponding uncertainty value (408); and
detecting an object represented in the image based on the point cloud representation, using the detection neural network (410), resulting in detection data;
computing a loss function based on comparing the detection data to ground truth data indicative of objects represented in the image; and
adjusting weights of the detection neural network based on the loss function.

15. A computer program comprising instructions, which, when the computer program is executed by a computer, cause the computer to carry out the method (100, 300, 500) of any preceding claims.

16. Data structure generated by carrying out the steps of the method according to claim 14.

17. A data carrier signal carrying the computer program of claim 15 or the data structure of claim 16.