TRR
JOURNAL OF THE TRANSPORTATION RESEARCH BOARD

Research Article

Transportation Research Record
1–15

DOI: 10.1177/03611981231184187
journals.sagepub.com/home/trr
Sage

Spatio-Temporal Fusion of LiDAR and Camera Data for Omnidirectional Depth Perception

Linlin Zhang[1], Xiang Yu[1*], Yaw Adu-Gyamfi[1] and Carlos Sun[1]

Abstract

Object recognition and depth perception are two tightly coupled tasks that are indispensable for situational awareness. Most autonomous systems are able to perform these tasks by processing and integrating data streaming from a variety of sensors. The multiple hardware and sophisticated software architectures required to operate these systems makes them expensive to scale and operate. This paper implements a fast, monocular vision system that can be used for simultaneous object recognition and depth perception. We borrow from the architecture of a start-of-the-art object recognition system, YOLOv3, and extend its architecture by incorporating distances and modifying its loss functions and prediction vectors to enable it to multitask on both tasks. The vision system is trained on a large database acquired through the coupling of LiDAR measurements with complementary 360-degree camera to generate a high-fidelity labeled dataset. The performance of the multipurpose network is evaluated on a test dataset consisting of a total of 7,634 objects collected on a different road network. When compared with ground truth LiDAR data, the proposed network achieves a mean absolute percentage error rate of 11% on the passenger car within 10 m and a mean error rate of 7% or 9% on the truck within 10 m and beyond 10 m, respectively. It was also observed that adding a second task (depth perception) to the modeling network improved the accuracy of object detection by about 3%. The proposed multipurpose model can be used for the development of automated alert systems, traffic monitoring, and safety monitoring.

Keywords

data and data science, artificial intelligence, data analytics, deep learning, machine learning (artificial intelligence), neural networks

Recent advances in deep learning algorithms have enabled autonomous systems to recognize objects at unprecedented accuracies across multiple domains including healthcare, agriculture, transportation, and many more. In transportation, these systems are used to assist with everyday driving, traffic monitoring, infrastructure assessment, and developing alert systems to improve safety on roadways. Object recognition alone, however, is not sufficient; there is also a need to perceive the respective depths of each object to enable autonomous systems to understand their environment. The current study develops a multipurpose model that implements a framework for simultaneous object recognition and depth perception at online speeds.

The prevalent depth perception approaches normally utilize three types of sensors: monocular camera, stereo camera, and light detection and range (LiDAR). LiDAR point cloud data provides accurate, long-range, and short-range 3D information of its environment and is by far the most popular technique, especially among systems where high levels of precision are required. Although the costs associated with using LiDAR have significantly reduced over the years, they are still prohibitively expensive compared with other alternatives such as camera-based depth perception systems (*1*). Additionally, the accuracy of LiDAR-based object recognition algorithms is relatively low (especially for smaller objects); as a result most systems rely on cameras, which have more intuitive

[1]Department of Civil and Environmental Engineering, University of Missouri-Columbia, Columbia, MO

*Xiang Yu is also affiliated to Cook, Flatt & Strobel Engineers, P.A., Kansas City, MO

Corresponding Author:
Linlin Zhang, lz5f2@mail.missouri.edu

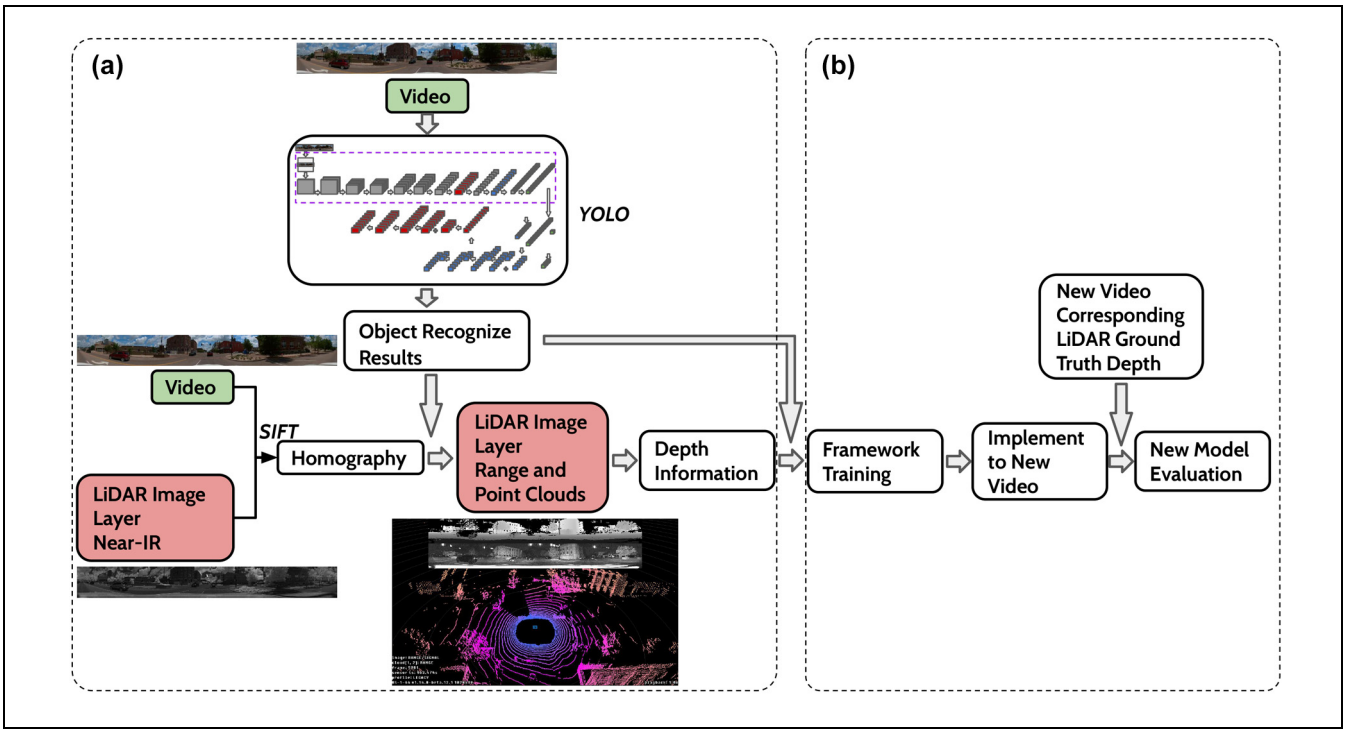


Figure 1. Proposed pipeline: (*a*) data fusion and synchronization to generate the training data with the application of You Only Look Once (YOLO) and scale invariant feature transform algorithm and (*b*) train a modified YOLOv3 architecture model for depth perception and model evaluation.

Note: IR = infrared; LiDAR = light detection and range.

information for object recognition. This adds to the complexity, latency, and cost of deploying such systems.

To overcome the limitations of LiDAR for object recognition and the lack of meaningful depth perception from cameras, many researchers have proposed approaches that enable the fusion of outputs from both systems. For example, Kumar et al. integrated camera and LiDAR data by first estimating the camera's intrinsic parameters (optical, digital, and geometric characteristics) via the well-known checkerboard approach (*2*). Thereafter, the extrinsic parameters of the LiDAR were estimated with a planar 3D marker. After the calibration processes, the LiDAR points were successfully projected onto a camera image with 2D bounding box proposals extracted. Although this approach is efficient, and most likely to achieve better results, it also introduces some new challenges, such as the need to calibrate two sensors with the camera intrinsic parameters and the camera and LiDAR extrinsic parameters each time they are set up (*2*). Some of the old issues, such as prices and not being intuitive enough to the public, still exist. Multiview video processing has also been proposed as a low-cost alternative to LiDAR because of its ability to generate comparable 3D point cloud data by analyzing video feeds in stereo or through triangulation. They are, however, easily corrupted at night and by inclement weather conditions such as snow and heavy rainfall. In addition, their ability to perceive depth at long-range distances is limited. The requirement for precise calibration of intrinsic and extrinsic camera parameters (orientation of cameras) are also required for stereo processing, which adds to the complexities of setup and maintenance.

To address the above-mentioned limitations of current approaches for object recognition and depth perception, we propose the framework of a multipurpose model for fast, 360-degree monocular camera-based object detection, classification, and depth perception, as depicted in Figure 1. The 360-degree monocular camera is adopted because it can provide a comprehensive understanding of the surroundings with only one sensor, which will simplify and reduce the complexities of setup. To be able to obtain depth information for regions of interest (ROI), a novel pipeline for integrating 360-degree camera and LiDAR data is proposed. The integrated data is used to create a training dataset for building a multipurpose model for simultaneous object recognition and depth perception. The resulting model can simultaneously recognize objects and their corresponding depths from 360-degree camera alone without a need for LiDAR data capture. The pipeline adopted is fast, as it only relies on

camera data and does not require LiDAR data fusion during inferencing. In summary, the main contributions of the current study are as follows:

1) It develops a framework for spatial-temporal fusion of 360-degree video and LiDAR data. This framework enables us to associate each bounding box proposal in 2D video domain with actual distances or depths generated from mobile LiDAR data.
2) It defines a novel architecture that extends the output prediction vectors of Darknet-like backbone with information about depth. We also introduce a new loss function that enables the network to simultaneously generate bounding box proposals and corresponding depth of each object in a single shot.
3) It generates a large database of annotations for machine learning (ML) model development and comparative analysis.

The outline of the paper is organized as follows. First, an overview of the current trends and best practices in depth estimation from cameras and LiDAR is presented. In the next section, a brief background on the datasets used, and a discussion of the methodology proposed is highlighted. Then, we evaluate the performance of the proposed architecture for simultaneous object recognition and depth perception. A comparative analysis with benchmarked datasets from LiDAR data is also performed. Lessons learned, concluding remarks, recommendations, and additional research needs are presented in the final section of the paper.

Related Work

Existing methods for depth perception can be broadly grouped into two categories: relative depth estimation approaches which use monocular or stereo vision cameras to construct scene disparity maps which correspond to the relative distance of objects in a scene; and absolute distance estimators which use LiDAR to estimate the actual distance of objects. Whatever the approach, integration of camera data is crucial for simultaneous object recognition and depth perception. The following sections describe methods that have been developed to enable simultaneous automatic object detection and depth perception.

Camera-LiDAR Fused Object Detection and Depth Estimation

Vision systems which rely on a fusion of camera and LiDAR data are by far the most popular for precise object recognition and depth perception (*3*). There are two main ML-based approaches used for fusing camera and LiDAR data. The first class of techniques fuses the outputs of 2D detection frameworks such as Cascade region-based convolutional neural network (R-CNN), multi-scale CNN (MS-CNN), Recurrent Rolling Convolution (RRC), and 3D detection frameworks including PointRCNN, PointPillars, and PV-RCNN (*4–9*). For example, Pang et al. proposed an approach that fuses 2D and 3D object detection candidates as joint detection candidates before non-maximum suppression (*3*). Then, several 2D CNN layers and max-pooling technique are applied to the joint detection candidates to output the final 3D detection results. The authors tested their framework on KITTI datasets and results showed that the combination of the 2D detection framework with the 3D detection network achieved the highest average precision scores in 3D object detection (*10*).

Bai et al. found out that the fusion of camera-LiDAR using object proposals alone is susceptible to the quality of the images and the accuracy of the object detection models (*11*). This introduced the second class of algorithms for fusing LiDAR and camera data. The algorithms perform data fusion by using ML to manipulate the raw point cloud and image features. TransFusion is a two-layer ML-fusion framework proposed by Bai et al. to overcome the limitations of the first class of algorithms (*11*). It has a first layer that extracts a 3D bird's eye view (BEV) and 2D image features maps using a transformers network, and a second layer that uses a soft-association mechanism to determine which image features need to be fused with the 3D BEV feature map (*12*). Extensive experiments were conducted on the proposed network to demonstrate its robustness to poor-quality images. This framework, however, only focused on the improvement of 3D object detection. To obtain a depth map of the surrounding environment, proposed a deep fusion architecture which leverages ResNet-50 as an encoder and Residual Up-Projection blocks as decoder (*13*). The LiDAR and RGB images are fused into four channels as input to the ResNet-50 encoder. Then, a skip connection technique is adopted to add more details textures and features to the output depth map.

Stereo and Monocular Object Detection and Depth Estimation

LiDAR devices are prohibitively expensive; stereo and monocular depth estimation and object recognition approaches have been explored to provide a low-cost and fast alternative (*1*). Some of the most high-performing stereo-based depth estimators have been explored by Weng and Kitani, and Wang et al. who used a monocular depth estimator called DORN (*14–16*). Qian et al. have

also explored a stereo depth network (*1*, *17*). Their accuracy compared with laser-based methods has been explored in You et al. (*17*). The increasing accuracy of depth maps generated from stereo cameras has led to the emergence and development of the so-called "pseudo-LiDAR" point cloud data which has recently attracted the attention of many researchers (*15*). To obtain pseudo-LiDAR, depth map images must first be generated. The Weng and Kitani LiDAR can then be created by transforming estimated depths with the camera's extrinsic matrix which enables a depth map to be projected into a real LiDAR coordinate system to obtain the pseudo-LiDAR (*14*). Subsequent application of 3D object detection algorithms on pseudo-LiDAR, supplemented with image-based 2D object recognition algorithms such as Mask R-CNN, can be used to achieve similar accuracy to real LiDAR data in big objects detection (*18*). Weng and Kitani, and Wang et al., however, pointed out that a limitation of pseudo-LiDAR is in the detection of distant and small objects such as pedestrians and cyclists (*14*, *15*).

The current paper is inspired by recent monocular depth estimation approaches which are able to learn object distances directly from camera images (*19–21*). The main idea is to modify the architecture of 2D object recognition architectures by incorporating distances obtained from disparity images. These models are typically trained using the KITTI 3D object detection dataset which has associated depth information for each bounding box proposal (*10*). Alternatively, Beltrán et al. developed multiple regression models using width and height from YOLO detection bounding box as independent variables to predict depth information (*21*). Based on the adjusted R square, the linear regression model had the most accurate prediction results, with an estimation accuracy of more than 80%. All these monocular depth estimation approaches produced real-time processing capability ranging between 25 and 45 frames per second (*20*, *21*, *22*). Reported average prediction errors were 11% in Vajgl et al. including all eight classes, 80.4% average precision in Beltrán et al., and 71.68% prediction accuracy in Yu and Choi (*20–22*).

The vast majority of monocular depth estimation techniques are trained on unidirectional, front-facing images. Autonomous systems, however, require omnidirectional depth prediction to navigate complex scenes with precision. A limited number of research studies have attempted to address this challenge from a purely vision-based approach. Zhou et al., Zioulis et al., Zou et al., Su and Grauman, and Li et al., for example, focused on indoor depth estimation using omnidirectional images (*23–27*). The study leveraged spherical convolutional neural networks (CNNs) to learn rotational representations to estimate depth information while overcoming camera lens distortion problems (*28*). In Li et al., perspective patches are decomposed into multiple perspective patches, and geometric features are integrated to estimate depth from the perspective patches (*27*). Zou et al., on the other hand, proved that depth perception can be estimated using an omnidirectional image without decomposition (*25*). Rather than decomposing the perspective image multiple times, the authors applied a modified RoomNet directly to the omnidirectional image, which resulted in comparable depth estimation accuracy (*29*).

Building on the success of these studies, we developed a fast framework for monocular, omnidirectional depth perception and object recognition. We coupled LiDAR measurements with complementary 360-degree camera to generate a high-fidelity labeled dataset for training deep neural network architectures for simultaneous depth perception and object recognition. We borrow from the YOLOv3 object recognition framework and incorporate distance prediction vectors and loss functions to enable the model to learn and predict object depths from any direction.

Data Collection

In this study, an Ouster OS1-64 LiDAR sensor with 64 channels (360-degree horizontal field of view, 45-degree vertical field of view, 120 m detection range) was utilized for 3D spatial information collection, and an Insta360 One X2 360-degree monocular camera (360-degree horizontal field of view, 180-degree vertical field of view, $>$120 m detection range) was used for video streaming. With recent advances in LiDAR technology, the LiDAR sensor used in this study can output not only 360-degree field of view spatial information, but also four structured 2D fixed resolution image layers, such as near-infrared (IR), range, reflectivity, and signal, which are perfectly spatially correlated with the point clouds. The monocular camera sensor can record 360-degree video streaming.

Figure 2 shows how the 360-degree camera-LiDAR system was assembled and mounted on the top of a vehicle for real-world data collection. Because both sensors are collecting 360-degree data, the camera is mounted on the top of the LiDAR to avoid them blocking each other. To have more training dataset, the planned data collection route was along E Broadway, Downtown, Columbia, MO, U.S., because of the high volume of traffic and street parking facilities. The LiDAR device operates at a rotation rate of 10 Hz and outputs images with a default fixed resolution of 1024×64. For the 360-degree monocular camera, it can record videos with different resolutions and frame rates, but, in this study, videos were recorded with 1440×720 resolution and 30 frames per second.

Figure 2. The 360-degree camera-light detection and range (LiDAR) system was assembled and mounted on the top of a vehicle for real-world data collection.

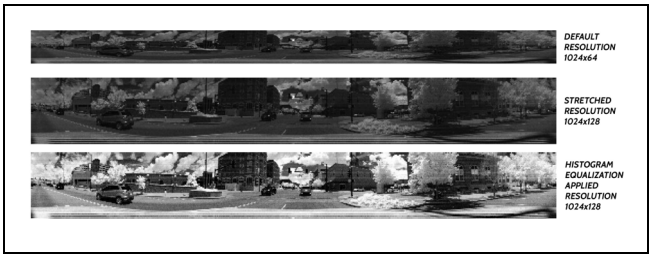


Figure 3. The light detection and range (LiDAR) output image layers: default resolution (1024 × 64) (*top*), resized (stretched) resolution (1024 × 128) (*middle*), and histogram equalization applied resolution (1024 × 128) (*bottom*). (After processing, the resolution becomes 1024 × 128 which matches the real-world view. The histogram equalization technique is applied to the stretched image layer near-infrared [IR] to increase the contrast.)

Methodology

The methodological framework developed to achieve high-fidelity omnidirectional depth prediction from monocular cameras involved five main steps: 1) LiDAR and video data are collected concurrently and data from both sensors are pre-processed to enhance their edges and features that are used for data fusion; 2) An image-matching algorithm—scale invariant feature transform (SIFT)—is used to find corresponding features between the camera and LiDAR data (*30*). This step results in a perspective transformation matrix which is used to align both datasets and subsequently project detected objects from the video onto the LiDAR domain to obtain depth information for each object; 3) YOLO is employed to detect objects in 360-degree videos, while the extracted depth information of each object from LiDAR is fused with the detection results; 4) A multipurpose network is built using a modified YOLOv3 architecture; 5) Fused object information from video and LiDAR depth information were used to build a training dataset for training the multipurpose network for simultaneous object recognition and depth perception. The following sections provide a detailed description of each step of the framework.

Data Pre-Processing

The default resolution of the images output from the LiDAR (1024 × 64) is resized by a factor of 2 to a new resolution of 1024 × 128 using bicubic interpolation of 4 × 4 neighborhood. Furthermore, to make the LiDAR image layers near-IR provide more information and improve visual effects, a computer vision technique called "histogram equalization" is applied to increase the contrast by spreading out the pixel intensity values (*31*). Finally, the LiDAR image layer near-IR, with histogram equalization applied and a resolution of 1024 × 128, is used for later data fusion and synchronization. Figure 3 shows, from top to bottom,

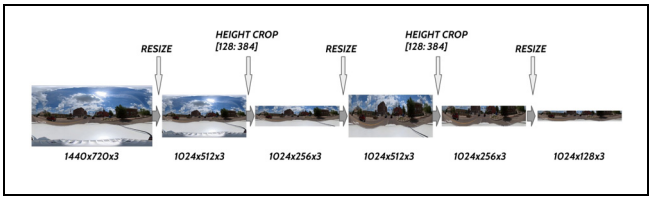


Figure 4. The pipeline for preprocessing the 360-degree video dataset.

the LiDAR image layer Near-IR with default resolution, resized resolution, and histogram equalization applied.

For video dataset preprocessing, since the vertical field of view of the 360-degree monocular camera is wider than that of LiDAR, the video dataset had to be cropped to a similar view to the LiDAR using a multi-scale approach, shown in Figure 4. It can be noticed that the vertical field of view of LiDAR is 45-degree and that of the 360-degree monocular camera is 180 degrees. The perfect quadruple relationship provides us a clue to crop the video dataset. The proposed multiscale approach resizes and crops the height of the original video image to achieve better matching with the LiDAR vertical field of view. The pre-processing of the video datasets is crucial for later fusion with the LiDAR datasets.

After preprocessing the LiDAR and video datasets, we obtain a high-contrast, real-world view of LiDAR image layers and a cropped and resized video view that is close to the LiDAR view. In this research, only two image layers of LiDAR datasets are used, near-IR and range. The image layer near-IR is used to align the LiDAR data with the 360-degree camera data, while the range is used for depth information extraction after the fusion. In Figure 5, the top image is the cropped and resized video image from 360-degree monocular camera; the following four in order are high-contrast resized near-IR, range, reflectivity, and signal.

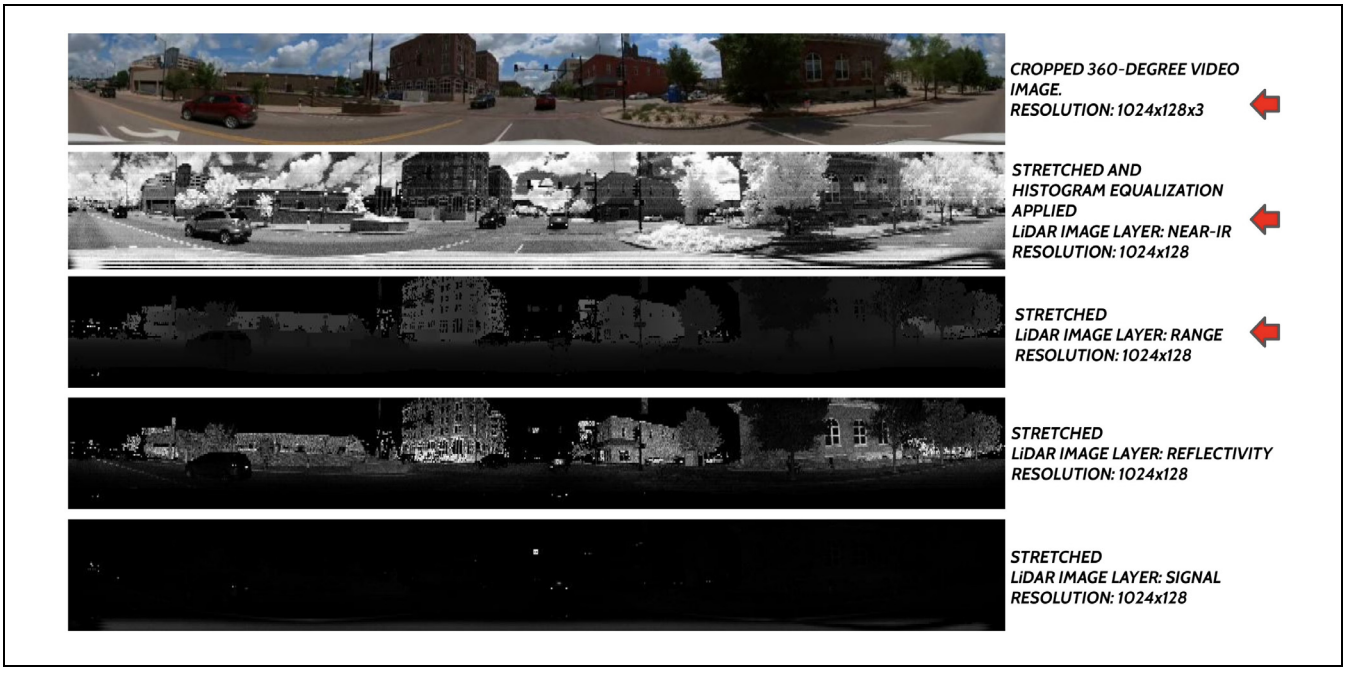


Figure 5. The top image is from 360-degree video datasets and is a resized and cropped image. The following four images are from light detection and range (LiDAR) datasets and are, in order: high-contrast stretched near-infrared (IR), stretched range, stretched reflectivity, and stretched signal from LiDAR sensor.

Data Fusion and Synchronization

Although we obtain similar field of views of the 360-degree camera and LiDAR datasets through preprocessing, there are still discrepancies because of the differences in sampling rates: 30 frames per second for the camera and 10 frames per second for the LiDAR. Spatial and temporal data alignment and synchronization techniques were, therefore, developed to address this discrepancy.

Temporal data alignment maps each LiDAR image to three camera images (since the camera sample rate is 3x higher than LiDAR). This step is straightforward, as each LiDAR and video image are timestamped (Unix timestamp) during data acquisition. Temporal data alignment is, however, not enough: since the camera is mounted on top of the LiDAR sensor, the relative position of objects on the road to the sensors constantly changes as the car moves on the roadway. Therefore, the result of temporal alignment (1:3 mapping) is passed through a spatial alignment procedure to correct for this discrepancy.

Data Fusion Spatially. The conventional approach for camera-LiDAR data alignment as shown in Kumar et al. can be complicated and time-consuming as it relies on calibration of intrinsic and extrinsic parameters of both devices (*2*). In this study, since both the camera and LiDAR are capable of outputting 360-degree images of the environment, a computer vision-based image matching algorithm, SIFT, is adopted to compute a perspective transformation matrix between the images from 360-degree video and 2D near-IR images from the LiDAR dataset (*30*). The SIFT algorithm is used because it is invariant to images rotation, affine distortion, illumination, and viewpoint changes (*32*, *33*). Additionally, Karami et al. observed that SIFT performs better than other image matching techniques such as speed-up robust feature (SURF), and oriented FAST, rotated BRIEF (ORB) in most scenarios (*32*, *34*, *35*). The perspective transformation matrix resulting from the application of SIFT captures translational and rotational differences between the 360-degree video and LiDAR datasets. However, the perspective transformation matrix has proven to be sensitive to the quality of the data that passes to it. Thus, RANSAC regressor is used to filter inliers and eliminate outliers used in the perspective transformation matrix calculation process. Figure 6 summarizes the perspective transformation matrix calculation algorithm.

While the vehicle is in motion, the 360-degree camera and LiDAR may shake slightly because of wind and uneven road conditions. This could cause significant deviations between the two datasets, and, therefore, a need to update the perspective transformation matrix. To account for this effect, we computed a perspective transformation matrix every 2 s, on rolling bases. Figure 7 shows the process used to update the perspective transformation matrix to account for camera shaking. There

Algorithm Homography Matrix Calculation

function Calculate_Homography ($\boldsymbol{Time\_table}$, $\boldsymbol{lidar\_frm_i}$, $\boldsymbol{camera\_frm_i}$, $\boldsymbol{if_i}$)
let $\boldsymbol{lidar\_frm_i}$ ← frame of lidar
let $\boldsymbol{camera\_frm_i}$ ← corresponding frame of camera
let $\boldsymbol{if_i}$ ← if calculate Homography
for $\boldsymbol{lidar\_frm_i}$, $\boldsymbol{camera\_frm_i}$, $\boldsymbol{if_i}$ in $\boldsymbol{Time\_table}$ **do**
  if $\boldsymbol{if_i = 1}$ **then**
    $\boldsymbol{lidar\_img_i}$ = read image ($\boldsymbol{lidar\_frm_i}$)
    $\boldsymbol{camera\_img_i}$ = read image ($\boldsymbol{camera\_frm_i}$)
    if mean ($\boldsymbol{lidar\_img_i}$) ≥ **30** and mean ($\boldsymbol{camera\_img_i}$) ≥ **30** **then**
      $\boldsymbol{M}$ = Calculate Homography using SIFT and RANSAC
    else
      $\boldsymbol{M}$ = None
  return $\boldsymbol{M}$

Figure 6. The perspective transformation matrix calculation algorithm.

Algorithm Homography Matrix Update

function Get_Homography_Matrix($\boldsymbol{Time\_table}$, $\boldsymbol{row_i}$, $\boldsymbol{lidar\_frm_i}$, $\boldsymbol{camera\_frm_i}$, $\boldsymbol{if_i}$)
let $\boldsymbol{lidar\_frm_i}$ ← frame of lidar
let $\boldsymbol{camera\_frm_i}$ ← corresponding frame of camera
let $\boldsymbol{if_i}$ ← if calculate Homography
let $\boldsymbol{M_{matrix}}$ ← [ ]
for $\boldsymbol{lidar\_frm_i}$, $\boldsymbol{camera\_frm_i}$, $\boldsymbol{if_i}$ in $\boldsymbol{Time\_table}$ **do**
  $\boldsymbol{lidar\_img_i}$ = read image ($\boldsymbol{lidar\_frm_i}$)
  $\boldsymbol{camera\_img_i}$ = read image ($\boldsymbol{camera\_frm_i}$)
  if mean ($\boldsymbol{lidar\_img_i}$) ≥ **30** and mean ($\boldsymbol{camera\_img_i}$) ≥ **30** **then**
    $\boldsymbol{M}$ = Calculate_Homography ($\boldsymbol{Time\_table}$, $\boldsymbol{row_i}$, $\boldsymbol{lidar_{frm_i}}$, $\boldsymbol{camera_{frm_i}}$, $\boldsymbol{if_i}$)
  else
    $\boldsymbol{M}$ = None
  if $\boldsymbol{M}$ = None **then**
    $\boldsymbol{i = i + 1}$
    $\boldsymbol{lidar\_img_i}$ = read image ($\boldsymbol{lidar\_frm_i}$)
    $\boldsymbol{camera\_img_i}$ = read image ($\boldsymbol{camera\_frm_i}$)
    $\boldsymbol{M}$ = Calculate_Homography ($\boldsymbol{Time\_table}$, $\boldsymbol{row_i}$, $\boldsymbol{lidar_{frm_i}}$, $\boldsymbol{camera_{frm_i}}$, $\boldsymbol{if_i}$)
    until $\boldsymbol{i = i + 4}$
    if $\boldsymbol{M}$ is not None **then**
      $\boldsymbol{M_{matrix}}$ ← append ($\boldsymbol{M}$)
  return $\boldsymbol{M_{matrix}}$

Figure 7. Obtain a list of perspective transformation matrix from corresponding video and light detection and range (LiDAR) datasets.

were cases, however, where a perspective transformation matrix could not be computed within 2 s because of insufficient matching feature points between the two datasets. In such cases, we extrapolated by using the perspective transformation matrix from the closest, successive frames. After the list of perspective transformation matrices have been obtained for corresponding video and LiDAR dataset, there are two ways to apply it on video and LiDAR datasets: 1) apply the updates every 2 s perspective transformation matrix and 2) combine the list of matrixes to a fixed perspective transformation matrix by calculating the median values for each element among the list of matrixes and apply for whole datasets. The results show that the fixed matrix outperforms the updated matrix. Therefore, the fixed perspective transformation matrix is adopted in this paper.

Object Detection and Depth Information Extraction

Accurate, fast, and comprehensive objection detection will play a crucial role in improving the accuracy of depth information extraction. In this paper, the popular unified, multiscale, anchor-based object detection algorithm, YOLO, is adopted because of its ability to process images quickly at higher accuracy rates, while producing generalizable representations (*36*). Originally introduced

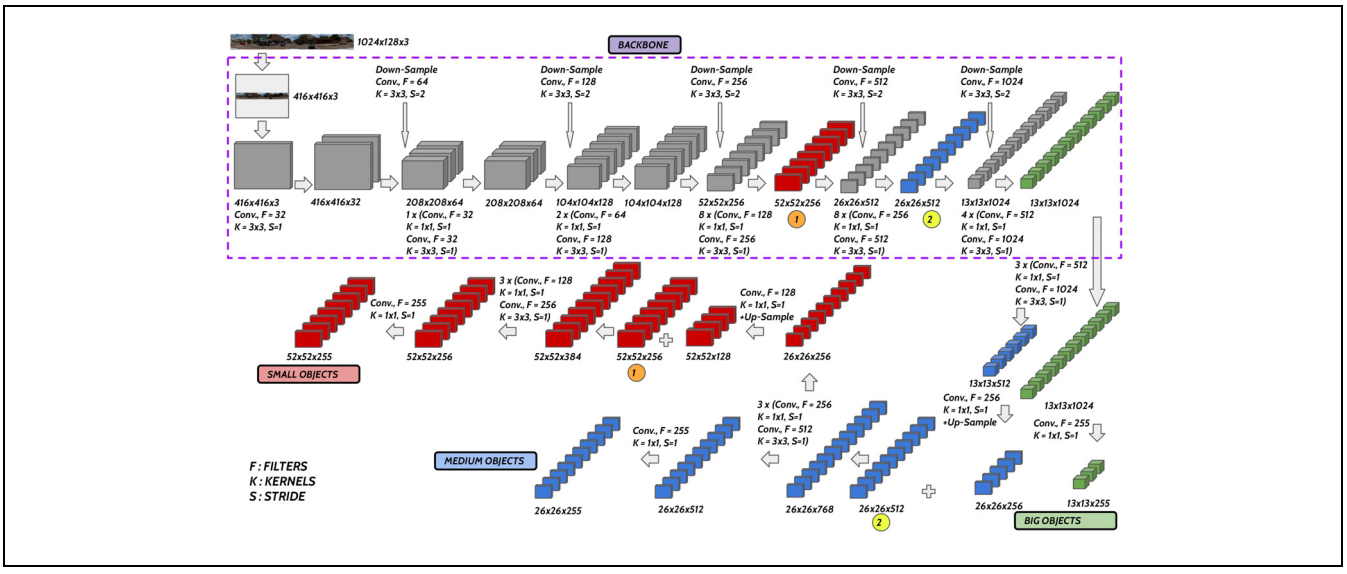


Figure 8. The default You Only Look Once Version 3 (YOLOv3) architecture.

by Redmon et al., YOLO and its variants have outperformed previous object detection algorithms such as R-CNN, fast R-CNN, and faster R-CNN (*36–42*).

Using the projection matrix obtained from the data alignment algorithm, object detection results from YOLO can be successfully projected onto and fused with the LiDAR depth data. Figure 7 shows the outcome of projection of a bounding box from video to LiDAR domain. Each vehicular object has been associated with a corresponding bounding box. Each pixel in the 2D near-IR image shown is perfectly spatially correlated with 3D point clouds, which significantly simplifies the depth information extraction process. As a result, we only need to know the position of the pixel of interest on the 2D near-IR image layer to find the corresponding 3D point in the point cloud. Each point in the point cloud contains location information $[x, y, z]$. Therefore, the distance from that point to the sensor can be calculated as $d = \sqrt{x^2 + y^2 + z^2}$.

Modified YOLOv3 Architecture for Depth Perception

Figure 8 shows the default architecture of the YOLOv3 algorithm. Darknet-53 is adopted as a backbone of YOLOv3. The backbone can also be taught as an encoder that serves as a feature extractor. This component of YOLOv3 can be replaced with any backbone, such as ResNext, ResNet-101, or ResNet-152. In comparison, although Darknet-53 has similar classification accuracy, it has faster computational speed and fewer layers (*40*). Instead of using max-pooling, stride-2 convolution is used to implement down sampling. The features extracted by the backbone are decoded on three succeeding scales into three output grids. This enables the architecture to detect objects at different scale—small, medium, and large—a concept that is adopted from feature pyramid networks (*43*). A grid is composed of cells responsible for detecting one of three objects (also called anchors) the center of which lies inside the cell. An object is detected if its intersection over union (IoU) with an anchor box is maximized.

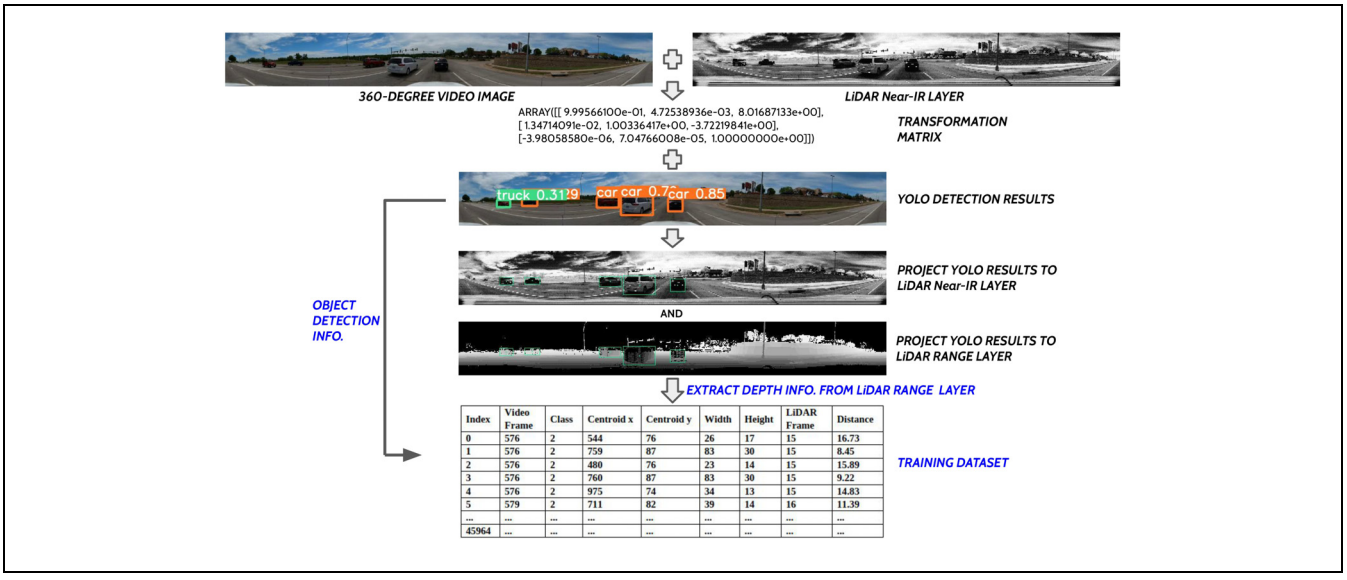
The default architecture of YOLOv3 cannot be used to estimate the distance of an object. To extend the YOLOv3 architecture to accommodate depth prediction, training datasets were reconfigured by incorporating information about object depths, the loss functions are updated to take the depth of the object into account, and prediction vectors were extended to produce the distance of an object for each cell in the image.

Loss Function. Object recognition models are designed to minimize four main losses: the bounding box center, width and height, prediction confidence, and classification loss. We introduce a new loss function to allow for depth loss minimization. For each scale of prediction, the loss function implemented in the current paper is designated as Equation 1:

$$l = \sum_{m=0}^{C^{w,h}} \sum_{n=0}^{B^a} \psi(m,n)[\lambda_1 \cdot L_1(m,n) + \lambda_2 \cdot L_2(m,n) + \lambda_3 \cdot L_3(m,n) + \lambda_4 \cdot L_4(m,n) + \lambda_5 \cdot L_5(m,n)] \quad (1)$$

where

λ_{1-5} = a penalizing factor for each respective part of the loss (in the current paper, we penalized bounding box and depth losses higher [by 5x] than classification and



Index	Video Frame	Class	Centroid x	Centroid y	Width	Height	LiDAR Frame	Distance
0	576	2	544	76	26	17	15	16.73
1	576	2	759	87	83	30	15	8.45
2	576	2	480	76	23	14	15	15.89
3	576	2	760	87	83	30	15	9.22
4	576	2	975	74	34	13	15	14.83
5	579	2	711	82	39	14	16	11.39
...	...	...	...	...	...	...	...	...
45964	...	...	...	...	...	...	...	...

Figure 9. The training data generation.
Note: IR = infrared; LiDAR = light detection and range.

confidence losses; the loss function iterates over $C^{w,h}$ grid cells and B^a anchors),
$\psi(m,n)$ = whether the m^{th} cell and the n^{th} anchor contains an object (the first four losses are defined according to Redmon and Farhadi) (*38*),
$L_1(m,n)$ = the mean square of center prediction computed for image cells with an object,
$L_2(m,n)$ = the mean width prediction computed for image cells with an object,
$L_1(m,n)$ = the mean height prediction computed for image cells with an object, and
$L_2(m,n)$ = a binary cross entropy loss for class predictions.
The output of each anchor box is designed as: $L_3(m,n)\, L_4(m,n)$ $b_x = \sigma(t_x); b_y = \sigma(t_y)$ – the sigmod of x and y coordinates, $b_w = P_w e^{tw}$; $b_h = P_h e^{th}$ – where P_w and P_h are anchor width and height. The depth loss introduced in this paper is defined as Equation 2: P_w

$$L_5(m,n) = \sum_{p=0}^{c} C_{m,n,p}\left(\bar{d}_{m,n,p} - d_{m,n,p}\right)^2 \qquad (2)$$

where
$C_{m,n,p}$ = the p^{th} class probability of the m^{th} cell and n^{th} anchor box.

Modified Prediction Vector. For object recognition only, the prediction vector for each cell is defined by: the objectness P_i—indicating whether a cell contains an object or not; bounding box coordinates $[x,y,w,h]$—center, width, and height of box; and the class prediction vector $[C_1, C_2, \ldots, C_n]$. The length of the prediction vector per cell is therefore defined as: (number of classes + 5) multiplied by the number of anchor boxes. We extend the prediction vector to incorporate depth information. The modified vector will therefore be $[P_i, x, y, w, h, d, C_1, C_2, \ldots, C_n]$. The length of the vector for each cell will be (number of classes + 6) multiplied by the number of anchor boxes used.

Training Data Generation

To obtain the training dataset, we processed four 360-degree video datasets and their corresponding LiDAR datasets. For the classes of object recognition, two classes were selected from 80 classes in the YOLO training dataset: car and truck. Based on the fusion and synchronization of datasets from the 360-degree monocular camera and LiDAR sensor, the YOLO object recognition bounding boxes were projected to the LiDAR datasets and the corresponding depth information was extracted from the LiDAR datasets directly.

Since the range of LiDAR distance detection is between 0 and 120 m, and LiDAR was mounted on the center of the top of a vehicle with a width 1.8 m, some outliers with a distance greater than 120 m or less than 1 m were removed. Finally, a training dataset containing 45,964 samples was generated. Figure 9 shows the process of generating the training dataset.

Evaluation of Predicted Depth Accuracy and Model Comparison

Evaluation of Predicted Depth Accuracy

The performance of the multipurpose network was evaluated on a test dataset collected on a different road

Figure 10. Multipurpose model object detection and depth estimation.

Table 1. Measurement of Error Rate (ER)

			Abs (predicted-ground truth)/ground truth			
Class	Gt distance range	Sample size	ER min.	ER mean	ER max.	ER SD
Car	< =10 m	2,490	0.000	0.1071	0.8659	0.1106
	> 10 m	854	0.0005	0.2357	0.9533	0.1471
Truck	< = 10 m	7	0.0047	0.0608	0.1855	0.0715
	> 10 m	13	0.01730	0.0923	0.1928	0.0595

Note: Gt = ground truth; Abs = absolute value; min. = minimum; max. = maximum; SD = standard deviation.

network. The test data consisted of a total of 7,634 objects with their corresponding depth information. The mode of data capture and data fusion was similar to the process followed for generating the training data. Figure 10 shows example object detection and depth prediction results for a sequence of frames. Visual assessment of the figure shows consistency in depth prediction as the ego vehicle passes or is overtaken by vehicles.

The models' performance is evaluated quantitatively using two main criteria: the F-1 score, and the mean absolute percentage error (MAPE) defined by Equations 3 to 6. The F-1 score measures the accuracy of the predicted bounding boxes, whereas MAPE measures the accuracy of the depth predictions.

$$MAPE = \frac{1}{N}\sum_{t=1}^{n}\left|\frac{d_t - \overline{d_t}}{d_t}\right| \tag{3}$$

where
N = the number of samples,
d = the ground truth depth, and
$\bar{d}$ = the predicted depth.

MAPE is calculated for predictions with bounding box IoU greater than 0.5.

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \tag{4}$$

$$precision = \frac{True\ \ Positive}{True\ Positive + False\ Positive} \tag{5}$$

$$recal = \frac{True\ Positive}{True\ Positive + False\ Negative} \tag{6}$$

A true positive prediction must have an IoU of 0.5 or more with the ground truth, and their predicted labels must also match.

Table 1 shows the error rates for different classes of vehicle at different distances from the ego vehicle. In general, the error rates associated with distant objects are much higher than for objects closer to the ego vehicle. There were, however, instances where, for an object which was less than 10 m from the ego vehicle, the predicted depth was 87% higher. These outlier predictions were sometimes caused by matching errors: a distant

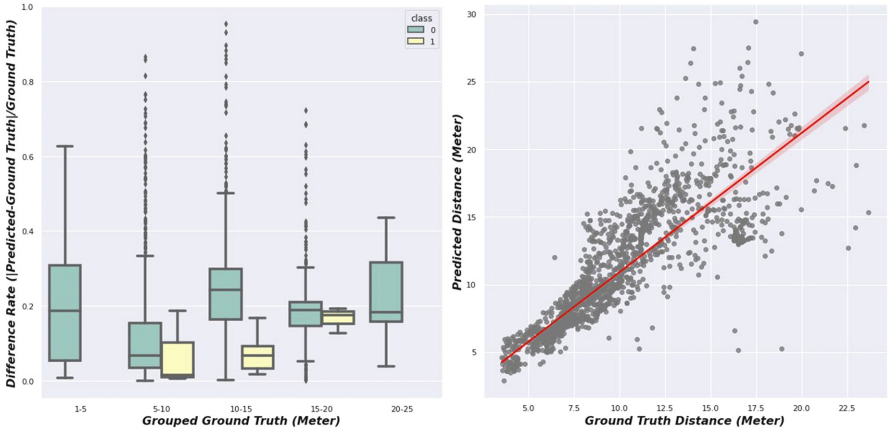


Figure 11. Box plot (*left*) of the grouped ground truth depth information and the error rate, and scatter plot (*right*) in which two classes are combined, since the truck sample size is small compared with the car sample size.
Note: class 0 = cars; class 2 = trucks; diff = difference; pred = predicted depth value; gt = ground truth depth value.

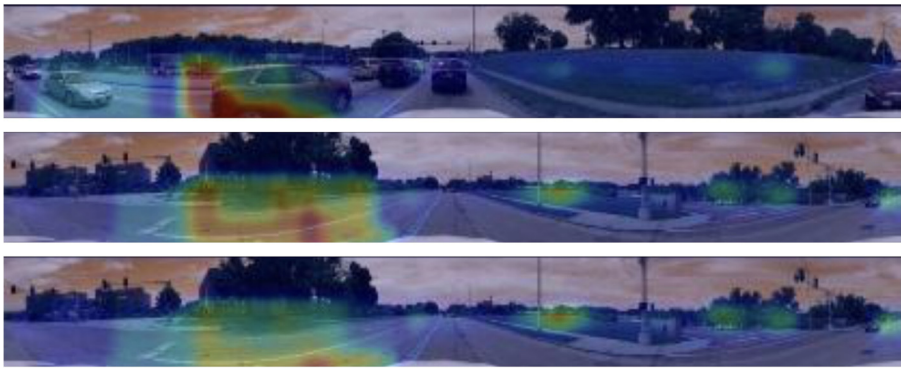


Figure 12. Three different scenarios of heat map of depth prediction errors as a percentage of ground truth depths from light detection and range (LiDAR).

object may be associated with a nearer object if their bounding box had a higher degree of overlap.

Figure 11 shows the box plot and scatter plot of the error rate, ground truth depth, and predicted depth. The x-axis in the box plot represents the grouped ground truth depth values which are extracted from the LiDAR dataset. The y-axis represents the error rate, which is calculated by the difference between the estimated depth values and their corresponding ground truth depth values divided by the corresponding ground truth depth values.

In the scatter plot, the x-axis represents the ground truth values and the y-axis represents the error rate values. It can be noticed that there is a moderate positive relationship between two variables, meaning the accuracy of depth prediction decreases as the object distance increases.

To further understand the models' robustness for omnidirectional depth prediction, we used heatmaps to visualize depth prediction error rates across the different directions of travel. From Figure 12, it is evident that prediction error rates were relatively higher for vehicles passing the ego vehicle than those ahead of or overtaken by the ego vehicle. Several factors could contribute to this: association with wrong ground truth labels, and sensor placement, to name a few.

With respect to object recognition accuracy, we compared the F-1 scores for a model built for a single task—object recognition to a multipurpose model (object recognition and depth perception). These results, shown in Table 2, are mixed. For car recognition, it appears that training a model to simultaneously recognize cars and predict depth improves the model's ability to recognize

Table 2. Influence of Multitasking on Overall Model Performance

Class	F1-score—detection only	F1-score—detection + depth
0: cars	0.629	0.712
1: trucks	0.830	0.812

objects. However, for trucks, the F-1 score dropped. The drop in accuracy could be because of low training samples of trucks in our training data. The data was collected on downtown streets and so we could not capture enough trucks in the training dataset.

With respect to model speed, real-time inferencing at batch size of 1, the model achieved 22 images per second using 416×416 image at half-precision (FP16) on NVIDIA GPU GTX 1080ti. This is about a 0.8x compared with the original YOLOv3 implementation.

Comparison with the State-of-the-Art

This section implements a state-of-the art monocular depth perception and compares its performance to the framework developed in the current study. Miangoleh et al.'s boosting monocular depth perception model is selected for comparative analysis because of its superior performance compared with other approaches described in Ranftl et al., Godard et al. (2017), and Godard et al. (2019) (*44–47*). The architecture of the boosting monocular depth model is such that a double-depth-estimation network is analyzed that combines two depth estimations of the same image at different resolutions adaptive to the image content to generate a result with high-frequency details while maintaining the structural consistency. It is observed that the low-resolution input to the network produces structurally consistent depth maps as they learn the overall global content in the image, while the high-resolution input captures the high-frequency details but loses the overall structure of the scene, generating low-frequency artifacts in the depth estimate. The proposed model, therefore, embeds the high-frequency depth details of the high-resolution patches into the structural consistent depth of the small-resolution input that provides a fixed range of depths for the full image.

The performance of the boosting depth prediction model is shown in Figure 13. Visual inspection of the depth images shows that the relative distances of objects from the ego vehicles are well captured by the model, as it can clearly distinguish between near and distant objects. To compare the predicted depths with the ground truths from LiDAR, we superimposed bounding box predictions on the depth images and used the average distance from a 10th percentile of pixels within the bounding box. The reason for not using an average or median of all pixels within the bounding box is that the bounding box area sometimes includes portions of the distant sky which can significantly increase the prediction error if not isolated. A root mean squared error and correlation coefficient against the ground truth data is used for quantitative evaluation of the boosting depth prediction model.

As shown in Figure 13, our panoramic, 360-degree images were sectioned into three parts—left, center, and right—to ensure that the monocular depth estimation model was tested on images that are similar to the ones

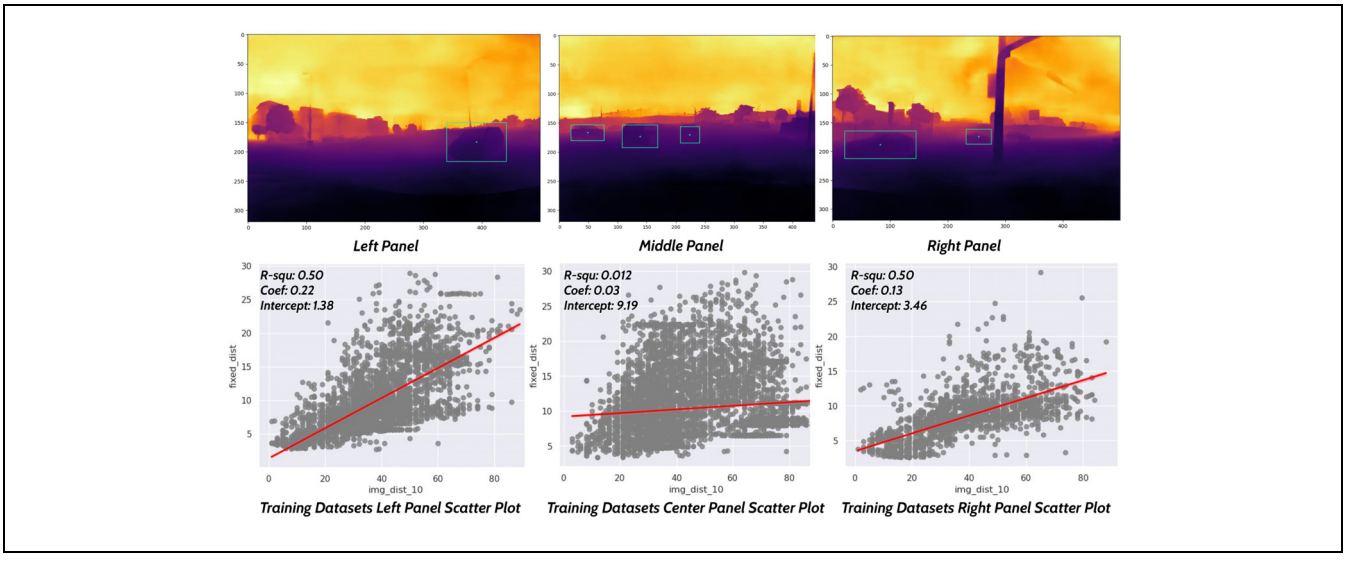


Figure 13. The first row is the monocular depth estimation results of left, middle, and right panels. The second row is scatter plots of the image intensities within the detected area versus ground truth depth: *x*-axis is the image intensity; *y*-axis is the ground truth.

used during training. The second row in Figure 13 shows the correlation between the predicted depths and the ground truths for the same training data as used in our model. The results show that the left panel Mean Square Error (MSE) value is 10.61, center panel MSE value is 12.76, right panel MSE value is 3.15, and the whole panoramic image MSE value is 10.70 for the ground truths within 30 m. For the ground truths within 10 m, the left panel MSE value is 4.37, center panel MSE value is 11.37, right panel MSE value is 1.89, and the whole panoramic image MSE value is 7.93.

Comparing with our model with MSE values 5.91 within 30 m and 1.53 within 10 m, the conclusion can be made that our model outperforms monocular depth estimation algorithms for the ROI quantitative depth estimation. The low performance of the monocular depth technique compared with our proposed framework could be for several reasons. First, small changes in relative distances are challenging for image-to-image-based depth estimation methods. Our proposed framework learns to predict absolute distance via regression on feature vectors, making it robust enough to perceive small changes in relative distance. Second, because monocular depth perception methods rely on low- and high-resolution inputs, they are highly sensitive to image resolution and scene illumination.

Concluding Remarks, Applications, and Open Challenges

This study developed an end-to-end framework for simultaneous object recognition and omnidirectional depth perception. Innovative data fusion pipelines were used to seamlessly align and integrate LiDAR and 360-degree camera information which were used to generate a training dataset for multipurpose learning. The state-of-the-art object recognition model, YOLOv3, was extended by incorporating distance measurements to enable it to detect vehicles and predict their respective distances from all directions.

LiDAR and camera information were fused spatially and temporally using SIFT with RANSAC regressors. The data fusion technique enabled us to uniquely project bounding box proposals in the video domain to LiDAR domain. This resulted in the generation of a large training dataset that was used to develop the multipurpose model. By modifying the training dataset, loss function, and prediction vectors, the study was able to train a network for object recognition and depth perception. Results show less than 10% depth prediction error for objects less than 10 m from the ego vehicle and about 20% error for longer distances. The accuracy of prediction for distant objects could be improved with more training data, as the majority of objects used in the current study fall within a range of less than 10 meters. Further investigation into the model's robustness showed that prediction errors for passing vehicles were slightly higher than for objects in any other direction.

A critical application of the proposed framework is toward the development of automated alert systems for truck-mounted attenuators (TMAs). TMA crashes have been on the rise, especially in mobile work zones. To date, alert systems used to warn distracted drivers in these construction work zones are operated manually, posing a risk to the operator and workers. The framework developed can be used to develop an alert system, which will process live feeds from a 360-degree camera, simultaneously recognize approaching vehicles and their distances, and automatically flag and alert accelerating vehicles.

There remain several open challenges to the methodology implemented in the current study. The first relates to the transformation matrix that is used to project objects in the video data to the LiDAR domain. Finding corresponding features between the LiDAR IR-image and the 360-degree camera can be challenging, especially when the scene does not contain objects with clearly defined edges, such as large buildings and vehicles. To overcome this, this paper searched for correspondence at different scales of image resolution after applying image contrast enhancement techniques. However, it is important to note that the resizing technique employed in this process may generate undetectable noises, even at critical positions. Although the resizing operation did not significantly affect the model's performance in this study, future investigations should prioritize addressing the noise generated by interpolation. To mitigate this issue, it is necessary to leverage techniques for generating high-definition images from low-resolution ones. Additionally, robust image-matching techniques are needed for calculating high-fidelity transformation matrices for data fusion.

Although the current study implemented class agnostic loss functions, the improvement in depth perception accuracy over a non-class agnostic approach was not significant. This could be a result of high class imbalance in our training dataset. However, the current study did not comprehensively study the impact of the loss function on the model outcomes. Another area that needs further investigation is the impact of data augmentation on the multipurpose model. In the current study, only contrast enhancement, image inversion, and flipping augmentation strategies were used. Image zooming, stretching, and any other techniques which changed the size of objects were not used as they had the tendency to confuse the model for depth perception. Further investigation into augmentation approaches that can improve both detection and depth perception accuracies need to be investigated. Building on the aforementioned challenges, an

additional aspect that demands attention is the evaluation of object-level depth. Exploring alternative methods to calculate the depth from pixel-level to object-level and assessing their effectiveness in reducing errors caused by background pixels would represent a promising avenue for future research in this domain.

Author Contributions

The authors confirm contribution to the paper as follows: study conception and design: Y. Adu-Gyamfi; data collection: L. Zhang, X. Yu; analysis and interpretation of results: X. Yu, Y. Adu-Gyamfi; draft manuscript preparation: L. Zhang, X. Yu, Y. Adu-Gyamfi, Sun. All authors reviewed the results and approved the final version of the manuscript.

Declaration of Conflicting Interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: This research is based on work supported by the National Science Foundation under grant no. 2045786.

ORCID iDs

Yaw Adu-Gyamfi https://orcid.org/0000-0002-1924-9792
Carlos Sun https://orcid.org/0000-0002-8857-9648

References

1. Qian, R., D. Garg, Y. Wang, Y. You, S. Belongie, B. Hariharan, M. Campbell, K. Q. Weinberger, and W.-L. Chao. End-to-End Pseudo-LiDAR for Image-Based 3D Object Detection. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, 2020.
2. Kumar, G. A., J. H. Lee, J. Hwang, J. Park, S. H. Youn, and S. Kwon. LiDAR and Camera Fusion Approach for Object Distance Estimation in Self-Driving Vehicles. *Symmetry*, Vol. 12, No. 2, 2020, p. 324. https://doi.org/10.3390/sym12020324.
3. Pang, S., D. Morris, and H. Radha. CLOCs: Camera-LiDAR Object Candidates Fusion for 3D Object Detection. *Proc., IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, NV, 2020.
4. Cai, Z., and N. Vasconcelos. Cascade R-CNN: High Quality Object Detection and Instance Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 43, No. 5, 2021, pp. 1483–1498. https://doi.org/10.1109/TPAMI.2019.2956516.
5. Cai, Z., Q. Fan, R. S. Feris, and N. Vasconcelos. A Unified Multi-Scale Deep Convolutional Neural Network for Fast Object Detection. In *Computer Vision – ECCV 2016* (B. Leibe, J. Matas, N. Sebe, and M. Welling, eds.), Lecture Notes in Computer Science, Springer, Cham, Switzerland, Vol. 9908, 2016, pp. 354–370.
6. Ren, J., X. Chen, J. Liu, W. Sun, J. Pang, Q. Yan, Y. W. Tai, and L. Xu. Accurate Single Stage Detector Using Recurrent Rolling Convolution. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 21–26, 2017.
7. Shi, S., X. Wang, and H. Li. PointRCNN: 3D Object Proposal Generation and Detection from Point Cloud. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, June 13–19, 2019.
8. Lang, A. H., S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom. Pointpillars: Fast Encoders for Object Detection from Point Clouds. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, June 15–20, 2019.
9. Shi, S., C. Guo, L. Jiang, Z. Wang, J. Shi, X. Wang, and H. Li. PV-RCNN: Point-Voxel Feature Set Abstraction for 3D Object Detection. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Seattle, WA, 2020.
10. Geiger, A., P. Lenz, C. Stiller, and R. Urtasun. Vision Meets Robotics: The KITTI Dataset. *The International Journal of Robotics Research*, Vol. 32, No. 11, 2013, pp. 1231–1237. https://doi.org/10.1177/0278364913491297.
11. Bai, X., Z. Hu, X. Zhu, Q. Huang, Y. Chen, H. Fu, and C.-L. Tai. TransFusion: Robust LiDAR-Camera Fusion for 3D Object Detection with Transformers. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, 2022.
12. Vaswani, A., G. Brain, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. *Attention is All You Need*. No. 2017-December, 2017.
13. IEEE, and ITSS. The 2019 IEEE Intelligent Transportation Systems Conference - ITSC, Auckland, New Zealand, October 27–30, 2019.
14. Weng, X., and K. Kitani. Monocular 3D Object Detection with Pseudo-LiDAR Point Cloud. *Proc., IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, Seoul, Korea (South), 2019.
15. Wang, Y., W.-L. Chao, D. Garg, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-LiDAR from Visual Depth Estimation: Bridging the Gap in 3D Object Detection for Autonomous Driving. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, CA, 2019.
16. Fu, H., M. Gong, C. Wang, K. Batmanghelich, and D. Tao. Deep Ordinal Regression Network for Monocular Depth Estimation. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018.
17. You, Y., Y. Wang, W.-L. Chao, D. Garg, G. Pleiss, B. Hariharan, M. Campbell, and K. Q. Weinberger. Pseudo-LiDAR++: Accurate Depth for 3D Object Detection in Autonomous Driving. *arXiv Preprint arXiv:1906.06310*, 2019.
18. He, K., G. Gkioxari, P. Dollár, and R. Girshick. Mask R-CNN. *Proc., IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, 2017.
19. Davanthapuram, S., X. Yu, and J. Saniie. Visually Impaired Indoor Navigation Using YOLO Based Object

Recognition, Monocular Depth Estimation and Binaural Sounds. *Proc., IEEE International Conference on Electro Information Technology (EIT)*, Mt. Pleasant, MI, May 14–15, 2021, IEEE, New York, pp. 173–177.
20. Vajgl, M., P. Hurtik, and T. Nejezchleba. Dist-YOLO: Fast Object Detection with Distance Estimation. *Applied Sciences (Switzerland)*, Vol. 12, No. 3, 2022, p. 1354. https://doi.org/10.3390/app12031354.
21. Beltrán, J., C. Guindel, F. M. Moreno, D. Cruzado, F. García, and A. de La Escalera. BirdNet: A 3D Object Detection Framework from LiDAR Information. *Proc., 21st International Conference on Intelligent Transportation Systems (ITSC)*, Maui, HI, November 4–7, 2018.
22. Yu, J., and H. Choi. Yolo Mde: Object Detection with Monocular Depth Estimation. *Electronics (Switzerland)*, Vol. 11, No. 1, 2022, 76. https://doi.org/10.3390/electronics11010076.
23. Zhou, K., K. Yang, and K. Wang. Panoramic Depth Estimation via Supervised and Unsupervised Learning in Indoor Scenes. *Applied Optics*, Vol. 60, No. 26, 2021, pp. 8188–8197. https://doi.org/10.1364/ao.432534.
24. Zioulis, N., A. Karakottas, D. Zarpalas, and P. Daras. OmniDepth: Dense Depth Estimation for Indoors Spherical Panoramas. In *Computer Vision – ECCV 2018* (V. Ferrari, M. Hebert, C. Sminchisescu, and Y. Weiss, eds.), Lecture Notes in Computer Science, Springer, Cham, Switzerland, Vol. 11210, 2018, pp. 453–471.
25. Zou, C., A. Colburn, Q. Shan, and D. Hoiem. LayoutNet: Reconstructing the 3D Room Layout from a Single RGB Image. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 2018.
26. Su, Y. C., and K. Grauman. Learning Spherical Convolution for Fast Features from 360° Imagery. *Proc., Advances in Neural Information Processing Systems*, Long Beach, CA, Vol. 30, December 4–9, 2017.
27. Li, Y., Y. Guo, Z. Yan, X. Huang, Y. Duan, and L. Ren. OmniFusion: 360 Monocular Depth Estimation via Geometry-Aware Fusion. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, LA, 2022.
28. Cohen, T. S., M. Geiger, J. Köhler, and M. Welling. Spherical CNNs. *arXiv Preprint arXiv:1801.10130*, 2018.
29. Lee, C. Y., V. Badrinarayanan, T. Malisiewicz, and A. Rabinovich. RoomNet: End-to-End Room Layout Estimation. *Proc., IEEE International Conference on Computer Vision (ICCV)*, Venice, Italy, October 22–29, 2017.
30. Lowe, D. G. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.
31. Zhu, Y., and C. Huang. An Adaptive Histogram Equalization Algorithm on the Image Gray Level Mapping. *Physics Procedia*, Vol. 25, 2012, pp. 601–608. https://doi.org/10.1016/j.phpro.2012.03.132.
32. Karami, E., S. Prasad, and M. Shehata. Image Matching Using SIFT, SURF, BRIEF and ORB: Performance Comparison for Distorted Images. *arXiv Preprint arXiv:1710.02726*, 2017.
33. Wang, X., and W. Fu. Optimized SIFT Image Matching Algorithm. *Proc., IEEE International Conference on Automation and Logistics*, Qingdao, China, 2008.
34. Bay, H., A. Ess, T. Tuytelaars, and L. van Gool. Speeded-Up Robust Features (SURF). *Computer Vision and Image Understanding*, Vol. 110, No. 3, 2008, pp. 346–359. https://doi.org/10.1016/j.cviu.2007.09.014.
35. Rublee, E., V. Rabaud, K. Konolige, and G. Bradski. ORB: An Efficient Alternative to SIFT or SURF. *Proc., International Conference on Computer Vision*, Barcelona, Spain, 2011.
36. Redmon, J., S. Divvala, R. Girshick, and A. Farhadi. You Only Look Once: Unified, Real-Time Object Detection. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, 2016.
37. Redmon, J., and A. Farhadi. YOLO9000: Better, Faster, Stronger. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, 2017.
38. Redmon, J., and A. Farhadi. YOLOv3: An Incremental Improvement. *arXiv Preprint arXiv:1804.02767*, 2018.
39. Bochkovskiy, A., C.-Y. Wang, and H.-Y. M. Liao. YOLOv4: Optimal Speed and Accuracy of Object Detection. *arXiv Preprint arXiv:2004.10934*, 2020.
40. Girshick, R., J. Donahue, T. Darrell, and J. Malik. Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Columbus, OH, 2014.
41. Girshick, R. Fast R-CNN. *Proc., IEEE International Conference on Computer Vision (ICCV)*, Santiago, Chile, 2015.
42. Ren, S., K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 39, No. 6, 2017, pp. 1137–1149.
43. Lin, T. Y., P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie. Feature Pyramid Networks for Object Detection. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 21–26, 2017.
44. Miangoleh, S. H. M., S. Dille, L. Mai, S. Paris, and Y. Aksoy. Boosting Monocular Depth Estimation Models to High-Resolution via Content-Adaptive Multi-Resolution Merging. *Proc., IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Nashville, TN, 2021.
45. Ranftl, R., V. Vineet, Q. Chen, and V. Koltun. Dense Monocular Depth Estimation in Complex Dynamic Scenes. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Las Vegas, NV, June 27–30, 2016.
46. Godard, C., O. Mac Aodha, and G. J. Brostow. Unsupervised Monocular Depth Estimation with Left-Right Consistency. *Proc., IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Honolulu, HI, July 21–26, 2017.
47. Godard, C., O. Mac Aodha, M. Firman, and G. Brostow. Digging into Self-Supervised Monocular Depth Estimation. *Proc., IEEE/CVF International Conference on Computer Vision (ICCV)*, Seoul, Republic of Korea, October 27–November 2, 2019.