Reports of Research paper "PointFusion: Deep Sensor Fusion for 3D Bounding Box Estimation"

Abstract

The paper presents PointFusion, a novel method for 3D object detection that integrates image and 3D point cloud data. Unlike existing methods that rely on complex multi-stage pipelines or specific assumptions about the sensors and datasets, PointFusion is conceptually simple and application-agnostic. The image data and the raw point cloud data are independently processed by a Convolutional Neural Network (CNN) and a PointNet architecture, respectively. The resulting outputs are then combined by a novel fusion network, which predicts multiple 3D box hypotheses and their confidences using the input 3D points as spatial anchors. PointFusion is evaluated on two distinctive datasets: the KITTI dataset, featuring driving scenes captured with a Lidar-camera setup, and the SUN-RGBD dataset, capturing indoor environments with RGB-D cameras. Our model is the first to perform better or on par with the state-of-the-art on these diverse datasets without any dataset-specific model tuning.

Introduction

3D object detection is a fundamental problem in computer vision that has a significant impact on most autonomous robotic systems, including self-driving cars and drones. The primary objective is to recover the 6 Degrees of Freedom (DoF) pose and the 3D bounding box dimensions for each object of interest in a given scene. This involves accurately estimating an object's position, orientation, and size in three-dimensional space.

Despite recent advances in convolutional neural networks (CNNs) that have enabled accurate 2D detection in complex environments, achieving high accuracy in 3D object detection remains a challenging problem. Methods for estimating 3D bounding boxes from a single image, even with recent deep learning techniques, often result in relatively low accuracy, particularly in depth estimation at longer ranges. As a result, many current real-world systems either use stereo vision or enhance their sensor suite with Lidar and radar. The Lidar-radar mixed-sensor setup is particularly popular in self-driving cars and is typically handled by a multi-stage pipeline, which preprocesses each sensor modality separately and then performs a late fusion or decision-level fusion step using an expert-designed tracking system like a Kalman filter.

Inspired by the successes of deep learning in handling diverse raw sensory input, we propose an early fusion model for 3D box estimation, which directly learns to combine image and depth information optimally. Various combinations of cameras and 3D sensors are widely used in the field, and it is desirable to have a single algorithm that generalizes to as many different problem settings as possible. Many real-world robotic systems are equipped with multiple 3D sensors; for example, autonomous cars often have multiple Lidars and potentially also radars. Yet, current algorithms often assume a single RGB-D camera or a single Lidar sensor. Many existing algorithms also make strong domain-specific assumptions. For example, MV3D assumes that all objects can be segmented in a top-down 2D view of the point cloud, which works for the common self-driving case but does not generalize to indoor scenes where objects can be placed on top of each other. Furthermore, the top-down view approach tends to work well for objects such as cars but does not for other key object classes such as pedestrians or cyclists.

Unlike the above approaches, our architecture is designed to be domain-agnostic and agnostic to the placement, type, and number of 3D sensors. As such, it is generic and can be used for a variety of robotics applications. In designing such a generic model, we need to solve the challenge of combining the heterogeneous image and 3D point cloud data. Previous work addresses this challenge by directly transforming the point cloud to a convolution-friendly form. This includes either projecting the point cloud onto the image or voxelizing the point cloud. Both of these operations involve lossy data quantization and require special models to handle sparsity in the Lidar image or in voxel space.

Instead, our solution retains the inputs in their native representation and processes them using heterogeneous network architectures. Specifically for the point cloud, we use a variant of the recently proposed PointNet architecture, which allows us to process the raw points directly.

Related Work

The paper reviews previous work in 6-DoF object pose estimation, highlighting methods like keypoint matching and 3D box regression from images and depth data. Existing methods often rely on strong assumptions about object categories or sensor setups, which limits their scalability and generalizability. PointFusion addresses these limitations by combining image and point cloud data without such assumptions.

Geometry-Based Methods

Geometry-based methods focus on estimating the 6-DoF object pose from a single image or an image sequence. These include keypoint matching between 2D images and their corresponding 3D CAD models or aligning 3D-reconstructed models with ground-truth models to recover the object poses. Gupta et al. propose to predict a semantic segmentation map as well as object pose hypotheses using a CNN and then align the hypotheses with known object CAD models using Iterative Closest Point (ICP). These types of methods rely on strong category shape priors or ground-truth object CAD models, making them difficult to scale to larger datasets. In contrast, our generic method estimates both the 6-DoF pose and spatial dimensions of an object without object category knowledge or CAD models.

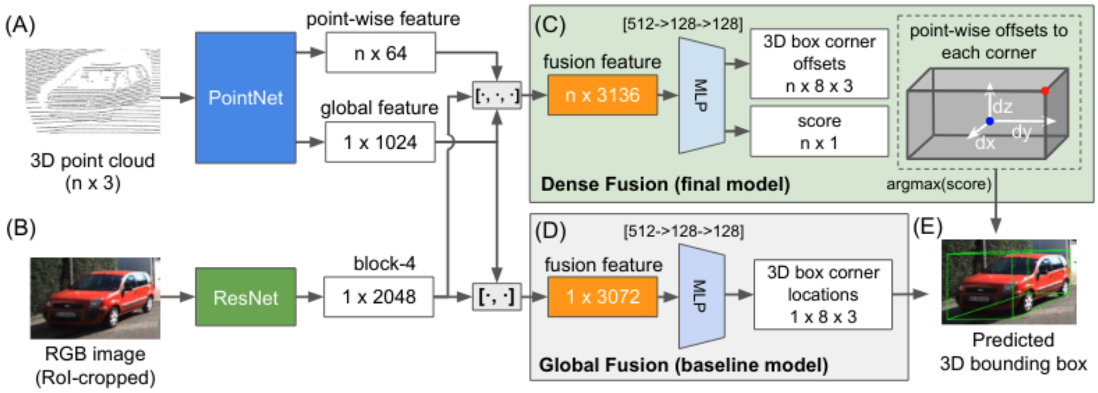
3D Box Regression from Images

Recent advances in deep models have dramatically improved 2D object detection, and some methods propose to extend the objectives with full 3D object poses. For instance, some approaches use R-CNN to propose 2D Regions of Interest (RoI) and another network to regress the object poses. Other methods combine a set of deep-learned 3D object parameters and geometric constraints from 2D RoIs to recover the full 3D box. Xiang et al. jointly learn a viewpoint-dependent detector and a pose estimator by clustering 3D voxel patterns learned from object models. Although these methods excel at estimating object orientations, localizing the objects in 3D from an image is often handled by imposing geometric constraints and remains a challenge due to the lack of direct depth measurements. One of the key contributions of our model is that it learns to effectively combine the complementary image and depth sensor information.

3D Box Regression from Depth Data

Newer studies have proposed to directly tackle the 3D object detection problem in discretized 3D spaces. Song et al. learn to classify 3D bounding box proposals generated by a 3D sliding window using synthetically-generated 3D features. A follow-up study uses a 3D variant of the Region Proposal Network to generate 3D proposals and uses a 3D ConvNet to process the voxelized point cloud. A similar approach by Li et al. focuses on detecting vehicles and processes the voxelized input with a 3D fully convolutional network. However, these methods are often prohibitively expensive because of the discretized volumetric representation. For example, some methods take around 20 seconds to process one frame. Other methods, such as VeloFCN, focus on a single Lidar setup and form a dense depth and intensity image, which is processed with a single 2D CNN. Unlike these methods, we adopt the recently proposed PointNet to process the raw point cloud. The setup can accommodate multiple depth sensors, and the time complexity scales linearly with the number of range measurements irrespective of the spatial extent of the 3D scene.

PointFusion Model



MODEL ARCHITECTURE

Overview

PointFusion uses a two-stage setup where a 2D object detector first identifies regions of interest (RoIs) in the image, and then the PointFusion model estimates 3D bounding boxes for these regions using both image and point cloud data. The model consists of three main components:

1. **PointNet Variant**: Extracts features from the point cloud data.
2. **CNN**: Extracts appearance features from the image data.
3. **Fusion Network**: Combines the outputs from the PointNet and CNN to predict 3D bounding boxes.

Point Cloud Network

We process the input point clouds using a variant of the PointNet architecture by Qi et al. PointNet pioneered the use of a symmetric function (max-pooling) to achieve permutation invariance in the processing of unordered 3D point cloud sets. The model ingests raw point clouds and learns a spatial encoding of each point and also an aggregated global point cloud feature. These features are then used for classification and semantic segmentation. PointNet has many desirable properties: it processes the raw points directly without lossy operations like voxelization or projection, and it scales linearly with the number of input points. However, the original PointNet formulation cannot be used for 3D regression out of the box. Here we describe two important changes we made to PointNet:

- **No BatchNorm**: Batch normalization has become indispensable in modern neural architecture design as it effectively reduces the covariance shift in the input features. In the original PointNet implementation, all fully connected layers are followed by a batch normalization layer. However, we found that batch normalization hampers the 3D bounding box estimation performance. Batch normalization aims to eliminate the scale and bias in its input data, but for the task of 3D regression, the absolute numerical values of the point locations are helpful. Therefore, our PointNet variant has all batch normalization layers removed.
- **Input Normalization**: As described in the setup, the corresponding 3D point cloud of an image bounding box is obtained by finding all points in the scene that can be projected onto the box. However, the spatial location of the 3D points is highly correlated with the 2D box location, which introduces undesirable biases. PointNet applies a Spatial Transformer Network (STN) to canonicalize the input space. However, we found that the STN is not able to fully correct these biases. We instead use the known camera geometry to compute the canonical rotation matrix. This rotates the ray passing through the center of the 2D box to the z-axis of the camera frame.

Fusion Network

The fusion network takes as input an image feature extracted using a standard CNN and the corresponding point cloud feature produced by the PointNet sub-network. Its job is to combine these features and to output a 3D bounding box for the target object. Below we propose two fusion network formulations:

1. **Global Fusion Network**: The global fusion network processes the image and point cloud features and directly regresses the 3D locations of the eight corners of the

target bounding box. We experimented with a number of fusion functions and found that a concatenation of the two vectors, followed by applying a number of fully connected layers, results in optimal performance. A major drawback of the global fusion network is that the variance of the regression target is directly dependent on the particular scenario. For autonomous driving, the system may be expected to detect objects from 1m to over 100m. This variance places a burden on the network and results in sub-optimal performance.

2. **Dense Fusion Network**: To address the issues with the global fusion network, we propose the dense fusion network. Instead of directly regressing the absolute locations of the 3D box corners, for each input 3D point we predict the spatial offsets from that point to the corner locations of a nearby box. As a result, the network becomes agnostic to the spatial extent of a scene. The model architecture uses a variant of PointNet that outputs point-wise features. For each point, these are concatenated with the global PointNet feature and the image feature resulting in an n × 3136 input tensor. The dense fusion network processes this input using several layers and outputs a 3D bounding box prediction along with a score for each point. At test time, the prediction that has the highest score is selected to be the final prediction.

Scoring Functions

The dense fusion network uses a scoring function to select the best bounding box prediction. Two scoring functions are proposed:

1. **Supervised Scoring**: The supervised scoring loss trains the network to predict if a point is inside the target box. This focuses the network on learning to predict the spatial offsets from points that are inside the target bounding box. However, this formulation might not give the optimal result, as the point most confidently inside the box may not be the point with the best prediction.
2. **Unsupervised Scoring**: The goal of unsupervised scoring is to let the network learn directly which points are likely to give the best hypothesis, whether they are most confidently inside the object box or not. We need to train the network to assign high confidence to the point that is likely to produce a good prediction. This involves two competing loss terms: preferring high confidences for all points, and scoring corner prediction errors proportional to this confidence.

Advantages and Applications

Advantages of the PointFusion Model

1. **Accuracy and Robustness**: By combining the strengths of image data and point cloud data, PointFusion achieves higher accuracy in 3D object detection. The model can effectively handle variations in object appearance and depth, making it robust across different environments and sensor setups.
2. **Generality**: Unlike many existing methods that are tailored for specific sensor configurations or object categories, PointFusion is designed to be domain-agnostic. This means it can be applied to a wide range of applications without significant modifications.

3. **Efficiency**: The model processes raw point cloud data directly and uses an efficient network architecture that scales linearly with the number of input points. This ensures that PointFusion can operate in real-time, which is crucial for applications like autonomous driving.
4. **Flexibility**: PointFusion can be integrated with any state-of-the-art 2D object detector, allowing it to leverage advancements in 2D detection techniques. The modular design also makes it easy to adapt the model for new sensor types or configurations.

Potential Applications

1. **Autonomous Driving**: PointFusion can be used in self-driving cars to detect and track objects such as other vehicles, pedestrians, and cyclists. By providing accurate 3D bounding boxes, the model enables the vehicle to navigate safely and avoid collisions.
2. **Drones and UAVs**: For drones and unmanned aerial vehicles (UAVs), PointFusion can assist in obstacle detection and navigation in complex environments. This is particularly useful for tasks like surveillance, delivery, and search and rescue operations.
3. **Robotic Manipulation**: In industrial settings, PointFusion can be used for robotic arms and manipulators to accurately detect and grasp objects. This is important for automation tasks such as assembly, packaging, and material handling.
4. **Augmented Reality (AR) and Virtual Reality (VR)**: PointFusion can enhance AR and VR applications by providing precise 3D object detection. This allows for better interaction with virtual objects and improves the user experience in mixed reality environments.
5. **Security and Surveillance**: PointFusion can be deployed in security systems to detect and track intruders or suspicious objects in real-time. The model's accuracy and robustness make it suitable for monitoring large and complex areas.

Future Work and Enhancements

1. **End-to-End Integration**: One promising direction for future work is to integrate the 2D detector and PointFusion into a single end-to-end 3D detection system. This could further improve the efficiency and accuracy of the model by eliminating intermediate steps and allowing joint optimization.
2. **Temporal Integration**: Extending PointFusion with a temporal component to perform joint detection and tracking in video and point cloud streams is another valuable enhancement. This would enable the model to leverage temporal information for more accurate and consistent object tracking.
3. **Advanced Fusion Techniques**: Exploring more advanced fusion techniques, such as attention mechanisms or graph-based methods, could further improve the performance of PointFusion. These techniques can help the model better capture the relationships between different sensory inputs.
4. **Scalability and Deployment**: Ensuring that PointFusion can scale to larger datasets and be deployed on edge devices with limited computational resources is crucial for real-world applications. Optimizing the model for different hardware platforms and exploring model compression techniques will be important steps in this direction.

5. **Broader Applicability**: While PointFusion is already versatile, expanding its applicability to more domains, such as medical imaging or agricultural automation, could open up new possibilities. Customizing the model for specific use cases and integrating domain-specific knowledge will be key to achieving this.

Experiments

Datasets

PointFusion is evaluated on the KITTI and SUN-RGBD datasets, which feature outdoor driving scenarios and indoor environments, respectively. The KITTI dataset includes annotations for cars, pedestrians, and cyclists, while the SUN-RGBD dataset has over 700 object categories.

- **KITTI Dataset**: This dataset contains both 2D and 3D annotations of cars, pedestrians, and cyclists in urban driving scenarios. The sensor configuration includes a wide-angle camera and a Velodyne HDL-64E LiDAR. The official training set contains 7481 images. We split the dataset into training and validation sets, each containing around half of the entire set. We report model performance on the validation set for all three object categories.
- **SUN-RGBD Dataset**: This dataset focuses on indoor environments, with as many as 700 object categories labeled. The dataset is collected via different types of RGB-D cameras with varying resolutions. The training and testing sets contain 5285 and 5050 images, respectively. We report model performance on the testing set. Because SUN-RGBD does not have a direct mapping between the 2D and 3D object annotations, for each 3D object annotation, we project the 8 corners of the 3D box to the image plane and use the minimum enclosing 2D bounding box as training data for the 2D object detector and our models.

Metrics

The performance is measured using the 3D object detection average precision (AP3D) metric. A predicted 3D box is considered a true positive if its 3D intersection-over-union ratio (3D IoU) with a ground truth box exceeds a threshold. We compute a per-class precision-recall curve and use the area under the curve as the AP measure. For the KITTI dataset, the 3D IoU thresholds are 0.7, 0.5, and 0.5 for Car, Cyclist, and Pedestrian respectively. Following previous work, we use a 3D IoU threshold of 0.25 for all classes in SUN-RGBD.

Implementation Details

- **Architecture**: We use a ResNet-50 pretrained on ImageNet for processing the input image crop. The output feature vector is produced by the final residual block (block-4) and averaged across the feature map locations. We use the original implementation of PointNet with all batch normalization layers removed. For the 2D object detector, we use an off-the-shelf Faster-RCNN implementation pretrained on MS-COCO and fine-tuned on the datasets used in the experiments.
- **Training and Evaluation**: During training, we randomly resize and shift the ground truth 2D bounding boxes by 10% along their x and y dimensions. These boxes are

used as the input crops for our models. At evaluation time, we use the output of the trained 2D detector. For each input 2D box, we crop and resize the image to 224 × 224 and randomly sample a maximum of 400 input 3D points in both training and evaluation. At evaluation time, we apply PointFusion to the top 300 2D detector boxes for each image. The 3D detection score is computed by multiplying the 2D detection score and the predicted 3D bounding box scores.

Results

PointFusion is compared to state-of-the-art methods and several baseline models. It achieves competitive results on both datasets, demonstrating its general applicability and effectiveness. The dense fusion network outperforms the global fusion network, and the unsupervised scoring function performs better than the supervised one.

- **Evaluation on KITTI**: The comprehensive comparison of models trained and evaluated only with the car category on the KITTI validation set includes all baselines and state-of-the-art methods. Among our variants, the final model achieves the best performance, while the homogeneous CNN architecture has the worst performance, underscoring the effectiveness of our heterogeneous model design. The final model also outperforms the state-of-the-art method MV3D on the easy category and has similar performance on the moderate category. Our model learns a generic 3D representation that can be shared across categories, and fusion of Lidar and image information always yields significant gains over Lidar-only architectures.
- **Evaluation on SUN-RGBD**: The final model outperforms the rgb-d baseline by 6% mAP, showing that the CNN performs well when given dense depth information. Our model compares favorably to state-of-the-art approaches, achieving comparable or better results while being much faster.

Qualitative Analysis

Qualitative results showcase the model's ability to accurately predict 3D bounding boxes for various objects, even in complex scenes. The fusion of image and point cloud data helps in accurately estimating object dimensions and orientations. The fusion model is better at estimating the dimension and orientation of objects than the Lidar-only model. The model correctly detects objects in complex scenarios, although it occasionally fails in extremely cluttered scenes.

Conclusion and Future Work

PointFusion is a generic 3D object detection model that effectively combines image and point cloud data. It achieves state-of-the-art results on diverse datasets without requiring dataset-specific tuning. Future work includes integrating the 2D detector and PointFusion into a single end-to-end model and extending the approach to joint detection and tracking in video and point cloud streams. Additionally, exploring more advanced fusion techniques and incorporating temporal information could further enhance the model's performance and applicability in real-world scenarios.

References

The paper cites various influential works in the field, including advancements in 2D and 3D object detection, pose estimation, and methods for processing point cloud data. These references provide a comprehensive background and context for the development of the PointFusion model.

Output will be like