20th IEEE Intelligent Transportation Systems Conference (ITSC)

DepthCN: Vehicle Detection Using 3D-LIDAR and ConvNet

Alireza Asvadi, Luis Garrote, Cristiano Premebida, Paulo Peixoto, and Urbano Nunes

Institute of Systems and Robotics,
Department of Electrical Engineering and Computers,
University of Coimbra

Yokohama, JAPAN
October 2017
The proposed 3D-LIDAR-based vehicle detection algorithm (DepthCN)

**Offline Stage: DepthCN Optimization**
- HG Optimization
- ConvNet Training using Augmented DM Data

**Online Stage: Vehicle Detection**
- HG Using 3D-LIDAR Data
- Vehicle Classification using DM and ConvNet
Online Vehicle Detection

HG: Vehicle Hypothesis Generation (HG Using 3D-LIDAR Data)
- Grid-based Ground Removal
- Obstacle Segmentation for HG
- 3D–2D Projection

HV: Vehicle Hypothesis Verification
Candidate Extraction
Vehicle Classification using DM and ConvNet
- DM Generation
- Imbalanced Data and Data Augmentation
- ConvNet for Hypothesis Verification (HV)
HG: Vehicle Hypothesis Generation

3D-LIDAR PCD → Ground Removal → Obstacle Segmentation → 3D – 2D Projection → Vehicle Proposal BBs

ν, δ, η, ε, η"
HG: Vehicle Hypothesis Generation

- Grid-based Ground Removal
  Ground points are eliminated by rejecting cells containing points with low variance in Z dimension

- Obstacle Segmentation for HG
  Applying DBSCAN on the top-view X-Y values of the remaining points
  The segmented obstacles are then projected onto the DM

- 3D – 2D Projection
HV: Vehicle Hypothesis Verification

3D-LIDAR PCD → DM Gen. → Candidate Extraction → ConvNet Classification → Detected Vehicles

Candidate Extraction

Vehicle Classification using DM and ConvNet

- 66x112x1
- 32@5x5x1 S@1 P@2
- 64@5x5x32 S@1 P@2
- 64
- 2

- a vehicle candidate
- Convolution
- Max Pooling
- Convolution
- Max Pooling
- FC
- 0.46
- Dropout
- Softmax

Positive (Car) Negative
HV: Vehicle Hypothesis Verification

**Candidate Extraction**
A vehicle proposal BB in the DM is extracted as the vehicle candidate (candidate extraction), resized to 66×112 and inputted to ConvNet for classification.

**Vehicle Classification using DM and ConvNet**

- **DM Generation**
  Projecting sparse 3D-LIDAR’s point cloud on camera coordinate system, performing interpolation and encoding.

- **Imbalanced Data and Data Augmentation**
  A set of augmentation operations like scaling, flipping, jittering, cropping, rotation, brightness (depth) and aspect-ratio augmentation, and shifting each line with different small random biases is performed to aggregate and balance the dataset.

- **ConvNet for Hypothesis Verification (HV)**
  The ConvNet input size is set as 66×112×1 where 66 and 112. The ConvNet employed in DepthCN is composed by 2 Convolutional layers, 3 Rectified Linear Units (ReLUs), 2 Pooling layers, 2 Fully Connected (FC) layers, a Softmax layer, and a Dropout layer for regularization.
Detected Vehicles

The generated hypotheses and the detection results are shown as red and dashed-green BBs, respectively. The bottom figures show the result in the PCD, where the detected vehicles’ clusters are shown in different colors, and the remaining LIDAR points are shown in green. Notice that the depicted color-images are just to make visualization and understanding easier.
Offline DepthCN Optimization

- HG Optimization
- ConvNet Training using Augmented DM Data

ConvNet was trained on the augmented KITTI 3D-LIDAR-based DMs.

ConvNet Training with DMs
- Stochastic Gradient Descent (SGD)
- momentum of 0.9
- mini-batch size of 128
- max epochs of 40
- L2 regularization

The 32 convolutional filters learned in the first layer of ConvNet.
HG Optimization

The ground removal and clustering were optimized jointly, using exhaustive search, by maximizing the overlap of generated hypotheses with ground-truth BBs.

- In the grid-based ground removal, the parameters are grid cell size and variance threshold.
- The minimum number of points and the distance metric are related to DBSCAN.

<table>
<thead>
<tr>
<th>Parameters used in DepthCN</th>
</tr>
</thead>
<tbody>
<tr>
<td>( v )</td>
</tr>
<tr>
<td>0.5</td>
</tr>
</tbody>
</table>

Experimental Setup And Evaluation

KITTI Object Detection Dataset

- 80 % as training set (5,985 frames)
- 20 % as validation set (1,496 frames)
- On the ‘Car’ label

System Setup

- Hexa-core 3.5 GHz processor
- GTX 1080 GPU
- 64 GB RAM

Computational Analysis

- The runtime of DepthCN (un-optimized implementation) for processing a point cloud is about 2.3 seconds.
- Under MATLAB R2017a
Qualitative Results

Performance Analysis of Classification

The Convnet’s Vehicle Recognition Accuracy With (W) and Without (Wo) Applying Data Augmentation (Da).

<table>
<thead>
<tr>
<th>Dataset</th>
<th>WO-DA</th>
<th>W-DA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Train set</td>
<td>92.83%</td>
<td>96.02%</td>
</tr>
<tr>
<td>Validation set</td>
<td>86.69%</td>
<td>91.93%</td>
</tr>
</tbody>
</table>

Vehicle Detection Evaluation

Depthcn Vehicle Detection Evaluation On Kitti Test Set

mBoW uses hierarchical segmentation with bag-of-word classifiers whereas DepthCN uses DBSCAN with a ConvNet classifier.

DepthCN surpasses by about 1.5 percentage points the mBoW in Easy difficulty level.

<table>
<thead>
<tr>
<th>Benchmark</th>
<th>Easy</th>
<th>Moderate</th>
<th>Hard</th>
</tr>
</thead>
<tbody>
<tr>
<td>DepthCN</td>
<td>37.59%</td>
<td>23.21%</td>
<td>18.01%</td>
</tr>
<tr>
<td>mBoW [3]</td>
<td>36.02%</td>
<td>23.76%</td>
<td>18.44%</td>
</tr>
</tbody>
</table>
Qualitative Results:
This work has been financially supported by the European Union, Compete 2020 and Portugal 2020 programs under grant UID/EEA/00048/2013; and by “AUTOCITS: Regulation Study for Interoperability in the Adoption of Autonomous Driving in European Urban Nodes” - Action number 2015-EU-TM-0243-S, co-financed by the European Union (INEA-CEF). This research is being developed at the Institute of Systems and Robotics - University of Coimbra (ISR-UC).
Thank you for your attention

Questions?