Detecting the Unexpected: The Path to Road Obstacles Prevention in Autonomous Driving

Shmoolik Mangan, PhD
Algorithms Development Manager, VAYAVISION

AutonomousTech TLV Israel 2018
VAYAVISION’s approach

Raw-Data, Sensor-Level Fusion
followed by unique Obstacle Detection Algorithms
To provide Top quality Perception/Cognition
Traditional Object-Level Fusion

- Each sensor has a separate perception engine
- Fusion at the object level

<table>
<thead>
<tr>
<th>Type</th>
<th>Resolution</th>
<th>Strength</th>
<th>Weakness</th>
</tr>
</thead>
<tbody>
<tr>
<td>Camera</td>
<td>Very high</td>
<td>High density Color</td>
<td>No distance</td>
</tr>
<tr>
<td>LiDAR</td>
<td>Medium</td>
<td>Distance</td>
<td>Sparse</td>
</tr>
<tr>
<td>RADAR</td>
<td>Low</td>
<td>Speed Distance</td>
<td>Very sparse Moving only</td>
</tr>
</tbody>
</table>
VayaVision’s Sensor-Level Raw-Data Fusion

- Fusion of raw data, (not objects)
  - Joint probabilities preserved at pixel level

- Processing is done after fusion
  - Detection based on joint look and location

- Sensors complement each other, creating better results:
  - Better detection rates
  - Lower false detection rate
  - Lower latency
Sensor-Level Raw-Data Fusion Principle: Lidar+Camera

- **Input:**
  - HD Camera image
  - Low density/resolution LIDAR point cloud

- **Algorithm principles:**
  1. Apply calibrations (Intrinsic & Extrinsic)
  2. Match RGB color and 3D pointcloud to create **RGBd** point-cloud
  3. RGBd points are accumulated over time
  4. 3D Point-cloud is Up-Sampled to the resolution of the HD image

- **Output:**
  - **HD-RGBd 3D model:**
    - Every camera pixel have 3D position
    - Each 3D point in the cloud has color
    - The “world model” is given at camera density
VAYAVISION’s Development Vehicle
Rev.1: HD Camera, Velodyne 64, i7 PC, One GPU

Complete perception runs in Real-Time: Fusion, Detection, Classification, Tracking
VayaVision`s Complete Perception: Road Free-Space and Object Detection

Complete perception runs in Real-Time: Fusion, Detection, Classification, Tracking
Using single GPU

Setup: HD Camera, Velodyne 64, One Nvidia GPU, i7 PC
Obstacle Prevention Problem: How to Detect the Unexpected

Why is it important?
- Unidentified obstacles create unexpected conditions
- Unexpected situations cause accidents

What is an obstacle?
- Oxford dictionary: A thing that blocks one's way or prevents progress
Identifying Cars and Pedestrians: Easy with DNN

- Free Nvidia library can do it
- YOLO, fast and free DNN can do it,
- Android mobile app can do it
  
- Every tech student can download a GIT and do it

Can one train a DNN to detect every \textit{thing}?
Detecting Unexpected Obstacle:

The inherent flaw of guided machine learning systems (aka DNN):

- How can one expect the unexpected?

- If it never appeared in your training data, what are the chances of identifying an obstacle?

“Nobody expects the Spanish inquisition...” Monty Python
Trained DNN Cannot Deal with the Unfamiliar, Unexpected

Thin construction-board on a Forklift:

- Bounding-Box based DNN vehicle detection: failed on this unfamiliar object
- Semantic segmentation free-space DNN: failed on this unfamiliar object

➢ It’s impractical to train a DNN to detect everything!

• “NOBODY expects the Spanish Inquisition! Our chief weapon is surprise...”
The Real Driving Challenge is: *Detecting the Unexpected*

- Detection begins with **sensing**
- *To avoid it - you have to sense it*

**Image sensors alone** – insufficient for detecting
- Based on image look, without depth – impractical to say it’s an obstacle
- Too many “patches” in the image
  - Too many FA
  - Cant avoid mis-detection
  - Cant deal with every arbitrary shape and look
The Real Driving Challenge is: Detecting the Unexpected

- Detection begins with sensing
- To avoid it - you have to sense it

- Lidar sensor alone – insufficient resolution
  - Sparse - too few data points
    - Small obstacles “fall between the lines”
    - One frame you see it, the other you don’t

Lidar points overlaid on image
At distance: 35m

Lidar used: Velodyne 64
Detecting the Unexpected - Enabled by Raw-Data Sensor fusion

- Detection begins with sensing
- To avoid it - you have to sense it

- RGBd from raw data fusion does the job:
  - Over-time accumulating RGBd points for detection
  - Obstacle height above road is determined at image resolution
  - Free road is accurately determined

Lidar points on image

Obstacle probability (Red) by RGBd

Near Distance: 15m
Far Distance: 35m
Free-Space in Front-View and Birds-Eye View (at 35m)

Top view Transform

Free-space Probability (Blue)

Bird’s eye view
Case 2: Detecting Door Opening 40m away
Case 3: Trash Bin on the Road
65m away
Case 4: Peacocks on the Road at Twilight
Case 5: Small Obstacle on a Dirty Road
Small Obstacle Detections Study
Variable Shape, Size and Color

Setup: HD Camera, Velodyne 64, One Nvidia GPU, i7 PC
## Small Obstacles Detection Distances

While driving at 40KM/h

<table>
<thead>
<tr>
<th>Object</th>
<th>Height [cm]</th>
<th>Detection distance [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>Laptop box</td>
<td>10</td>
<td>33.7</td>
</tr>
<tr>
<td>Keyboard box</td>
<td>16</td>
<td>49.4</td>
</tr>
<tr>
<td>Display box</td>
<td>17</td>
<td>38</td>
</tr>
<tr>
<td>Basketball</td>
<td>21</td>
<td>49</td>
</tr>
<tr>
<td>Small basket</td>
<td>22</td>
<td>46.4</td>
</tr>
<tr>
<td>Large basket</td>
<td>32</td>
<td>48</td>
</tr>
</tbody>
</table>
Same Ride, same Parameters – Practically Zero FAs on the Road
Complete Perception scheme: with Unidentified Obstacle Detection

Complete perception runs in Real-Time: Fusion, Detection, Classification, Tracking

Setup: HD Camera, Velodyne 64, One Nvidia GPU, i7 PC
Complete Perception scheme: with Unidentified Obstacle Detection

Complete perception runs in Real-Time: Fusion, Detection, Classification, Tracking

Setup: HD Camera, Velodyne 64, One Nvidia GPU, i7 PC
Summary

- Low-level Raw-Data sensor fusion through up-sampling is superior
  - Uses local correlations to enhance resolution of Lidar to produce High-density RGBd image

- Pixel-level detection based on up-sampled HD-3D physical model is superior for unfamiliar objects
  - Depth sensing resolution and accuracy are enhanced through up-sampling
  - Provides utmost fidelity in obstacle detection, regardless of size shape look or familiarity
NOBODY expects the Spanish Inquisition!
NOBODY expects the Small Obstacle!
Their chief weapon is surprise...

Monty Python

Thank you
Up-Sampled Model allows Bridging the Lidar Sparsity Gaps

Zoomed LiDAR on image:

<table>
<thead>
<tr>
<th>LiDAR reflections from road [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>58.6</td>
</tr>
<tr>
<td>49.5</td>
</tr>
<tr>
<td>42.9</td>
</tr>
<tr>
<td>37.7</td>
</tr>
<tr>
<td>33.1</td>
</tr>
<tr>
<td>30.5</td>
</tr>
</tbody>
</table>

Zoomed image:

Lidar used: Velodyne 64
So, if Dense-3D Cloud does the work, why do we need RGB Camera Image?

- **Alternative solution:**

- **Cost effective alternative:**