SM3D: Real-Time Simultaneous Mapping and 3D Detection

Group Member: Guanqing Li, Rui Yang, Runfa Li, Shuangquan Sun

Mentors: Prof. Trivedi, Akshay Rangesh

Presentation Date: 06/02/2020
**Input:** RGB Images: $I_t, I_{t-1}, I_{t-2}$
Depth Images: $D_{t-1}$
GPS Location: $P_{t-1}, P_{t-2}$

**Output:** Mapping: $Map_0$ to $Map_t$
Depth Images: $D_t$
3D Bounding Box: $Box_t$

- **Structure from Motion Module - Unsupervised**
- **3D Detection Module – Supervised**
• Expected goal
  ➢ Real-Time simultaneous mapping & 3D detection

• Novelty
  ➢ First time combining Structure from Motion (SFM) with 3D detection using Pseudo Lidar point cloud

• Significance
  ➢ A real-time system with diverse information (Mapping, 3D bounding box, depth image, Pseudo Lidar point cloud) which leaves slots for many further modification and improvement
CURRENT PROGRESS: SFM

- KITTI sequence09: 0-700 frame
- Blue line: estimated trajectory
- Red line: GT trajectory
CURRENT PROGRESS: SFM

- Complete KITTI sequence 09: 1590 frames
- Blue line: estimated trajectory
- Red line: GT trajectory

Original SFM Learner [1]

Advanced SFM Learner (this work)
<table>
<thead>
<tr>
<th>KITTI 09</th>
<th>ATE</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.0100</td>
<td>0.0016</td>
</tr>
<tr>
<td>Advanced</td>
<td>0.0091</td>
<td>0.0017</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>KITTI 10</th>
<th>ATE</th>
<th>RE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Original</td>
<td>0.0085</td>
<td>0.0016</td>
</tr>
<tr>
<td>Advanced</td>
<td>0.0081</td>
<td>0.0018</td>
</tr>
</tbody>
</table>

- ATE: Absolute trajectory error
- RE: Rotation Error
- Advanced VS Original

\[
\text{RE}_{\text{adv}} \approx \text{RE}_{\text{ori}}
\]

\[
\text{ATE}_{\text{adv}} < \text{ATE}_{\text{ori}}
\]
CURRENT PROGRESS: PSEUDO-LIDAR POINT CLOUD

- Image
- Ground Truth Depth Map
- Predicted Depth Map (Need improvement)
- Pseudo-LiDAR Point Cloud
CURRENT PROGRESS: 3D DETECTION

```python
subprocess.check_call(['cmake', ext.sourcedir] + cmake_args, cwd=self.build_temp, env=env)

build_make_file = 'build/temp-linux-x86_64-3.7/src/spconv/CMakeFiles/spconv.dir/build.make'
link_file = 'build/temp-linux-x86_64-3.7/src/spconv/CMakeFiles/spconv.dir/link.txt'
for file in [build_make_file, link_file]:
    with open(file) as f:
        newText = f.read().replace('/usr/local/cuda/lib64/libcuda.so', '/usr/lib64/nvidia/libcuda.so')
        with open(file, 'w') as f:
            f.write(newText)
subprocess.check_call(['cmake', '--build', '.'] + build_args, cwd=self.build_temp)
```
CURRENT PROGRESS: POINT STACKING ALGORITHM

Image:

\[ I_{t-1} \rightarrow I_{2I} \rightarrow I_t \]

Velodyne:

\[ V_{t-1} \rightarrow V_{2V} \rightarrow V_t \]

- \( V2I \): Velodyne → Reference → Rectified → Camera → Image
  (computationally expensive)

- \( V2V = I2I \)
  (If no relative motion between Lidar & Camera)
Stack Number = 3

- Keep the total point number same after "Stack Number" by removing the earliest points.

\[ N = (N - n_T) + n_T \]
CURRENT PROGRESS: POINT STACKING RESULT

Pseudo-LiDAR Point Cloud before Augmentation

Pseudo-LiDAR Point Cloud after Augmentation
(Stack number = 2)
CURRENT PROGRESS: REMAINING ISSUES

- Improve the quality of predicted depth maps
  - Choose appropriate hyperparameters to avoid overfitting of disparity loss
  - Now are using GT depth for following step
  - Quality of the depth image decide the performance of Lidar & 3D detection

- Train Pointpillar, include 3D detection into the architecture while maintain efficiency.
  - Mapping alone 68 ms/frame on week 6,
    182 ms/frame this week (same code, same environment, server busy)
INDIVIDUAL WORKLOAD

- Depth map and projection matrix: Runfa
- Depth to pseudo Lidar point clouds: All
- 3D detection on pseudo lidar point clouds
  - PointPillars (high speed): All
  - Debugging and documentation: Shuangquan
- Stacking Pseudo-LiDAR Point Clouds: Runfa
MILESTONES & TIMELINE

• Week 10:
  ➢ 3D detection

• Week 11:
  ➢ Report Writing
  ➢ Add supervision to 3D detection, end-to-end multi-task learning (flexible)


[3] Alex H. Lang, Sourabh Vora, Holger Caesar, Lubing Zhou, Jiong Yang, and Oscar Beijbom. Pointpillars: Fast encoders for object detection from point clouds. In CVPR, 2019. 1, 2, 6, 7, 8, 14, 15, 16
THANKS
QUESTIONS?