Towards Real-Time Metric-Semantic SLAM

Antoni Rosinol*¹, Yun Chang¹, Marcus Abate¹, Daniel Wrafter¹, Siyi Hu¹, Ben Smith², Dan Griffith², Luca Carlone¹

¹MIT
²SPARK Lab, Lincoln Laboratory
Motivation

Real-Time Metric-Semantic SLAM, what is it?

• **Metric**: understanding the scene at the geometric level (landmarks, lines, planes, normals, surfaces …)

• **Semantic**: understanding the entities in the scene at a human level (objects such as tables, chairs, coffee mug…)

• **Real-Time**: we do not want to wait for hours, not even minutes.
Motivation

Fully autonomous systems should operate given high-level tasks, and figure out the necessary low-level tasks.
What does a robot need to accomplish high-level tasks?

3D Scene Understanding

3D Semantic segmentation
3D Geometry of the Scene
3D Localization

Metric-Semantic SLAM

Source: SLAMcore
Motivation

Plethora of applications:

• Search-and-Rescue: find stranded climbers on the mountain
• Human-level navigation: go to the kitchen and bring me coffee
• Exploration: find an exit to this building
• Inventory: count and retrieve all chairs in this venue
• Workplace Co-bots: give me the wrench, hold this object
• Agriculture robots: detect and remove weeds, pick and count apples
• Autonomous cars: bring me to work avoiding pedestrians, cars, …
State-of-the-art Human readable Map

Palais des Congrès de Montréal

Direct indoor access to hotels via:
- Fairmont The Queen Elizabeth
- InterContinental
- Hôtel Bonaventure Montréal
- Hyatt Regency Montréal
- Marriott Château Champlain
- Le St-James
- Le Westin Montréal
- W Montréal
- Gouverneur Hotel Place Dupuis

Antoni Rosinol
Real-Time Metric-Semantic SLAM

5/13/19 6
State-of-the-art Robot readable Map

Point Clouds…
Bridge the Gap between human vs robot maps

Requirements for the ideal Metric-Semantic 3D map:

• **Dense** 3D geometry with **topological information** (surfaces, normals, planes)

• **3D Semantic** information (walls, floor, objects)

• **Lightweight**
  • Low resolution when possible (planes: walls, floor, …)
  • Easy to compute, store and process
Point Clouds

• **Main benefits**: allow *accurate* and *fast* localization.
• **Main disadvantages**: *sparse*, lacks topology (normal, surfaces, …)

- Most classical representation for SLAM, yet unsuitable for tasks such as Obstacle-free navigation, Path Planning.
- Semantics can be encoded on 3D points [1], but relies on the point cloud being dense for meaningful segmentation.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Clouds</td>
<td>✗</td>
<td>✓/✗</td>
<td>✗</td>
<td>✓/✗</td>
</tr>
</tbody>
</table>

Point Clouds

How can we recover the topology of the scene from sparse samples?
3D Mesh

Encoding connectivity of the 3D landmarks in a 3D mesh?
3D Mesh

- **Main benefits**: adds *topological* properties, while being *efficient*, *multi-resolution*.
- **Main disadvantages**: sensitive to noise, outliers, conceptually difficult to build incrementally.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Clouds</td>
<td>X</td>
<td>✓ / X</td>
<td>X</td>
<td>✓ / X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No, if Dense</td>
<td>No, if Sparse</td>
<td></td>
</tr>
<tr>
<td>3D Mesh</td>
<td>✓</td>
<td>✓</td>
<td>X</td>
<td>✓</td>
</tr>
</tbody>
</table>
3D Mesh

- Ideally, one may achieve computer graphics levels of detail where needed, while keeping mesh coarse otherwise:

If it wasn’t for the noisy and outlier 3D points...
Volumetric Methods: Voxels/Octrees

- **Main benefits:** robust to noise/outliers, dense.
- **Main disadvantages:** costly to compute/store, fixed resolution, lacks geometric invariance (shifts of cost volume produce different results).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Clouds</td>
<td>X</td>
<td>✓ / X</td>
<td>✓</td>
<td>✓ / X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No, if Dense</td>
<td></td>
<td>No, if Sparse</td>
</tr>
<tr>
<td>3D Mesh</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Voxels</td>
<td>X</td>
<td>✓ / X</td>
<td>✓</td>
<td>✓ / X</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No, if small voxel</td>
<td></td>
<td>No, if large voxel</td>
</tr>
</tbody>
</table>
3D Meshes need regularization

3D (local) mesh generation from noisy measurements requires regularization:

- Variational approaches [1]
- Surfel Meshing [2]
- Structural Regularities [3]:

Global methods such as Delaunay triangulation [4] or Poisson reconstruction [5] are too computationally expensive to run in real-time (for SLAM) on the dense point…

Source: [3]

[2] Thomas Schöps and Torsten Sattler and Marc Pollefeys "SurfelMeshing: Online Surfel-Based Mesh Reconstruction"
Our Approach:
Real-Time Multi-Frame Incremental 3D Mesh generation + Pose Estimation in a tightly coupled approach using Structural Regularities

APE translation error mapped onto trajectory [m]

RMSE ATE**: 7.7cm

RMSE ATE**: 5.7cm

Without Co-planarity*

Higher Fidelity 3D Mesh

*Co-planarity: 3D landmarks are constrained on the same plane.

With Co-planarity*

+25% Localization Accuracy

**RMSE ATE: Root Mean Squared Error for the Absolute Translation Error

https://www.mit.edu/~arosinol/research/struct3dmesh.html

Our Approach:
Real-Time Multi-Frame Incremental 3D Mesh generation + Pose Estimation in a tightly coupled approach using Structural Regularities

![Factor Graph](image)

**Fig. 2:** VIO factor graph combining Structureless ($\phi_{t_c}$), Projection ($\phi_{l_c}$) and Regularity ($\phi_R$) factors (SPR). The factor $\phi_R$ encodes relative constraints between a landmark $l_i$ and a plane $\pi_0$.

<table>
<thead>
<tr>
<th>Sequence</th>
<th>OKVIS</th>
<th>MSCKF</th>
<th>ROVIO</th>
<th>VINS-MONO</th>
<th>SVO-GTSAM</th>
<th>SPR</th>
</tr>
</thead>
<tbody>
<tr>
<td>MH.01</td>
<td>16</td>
<td>42</td>
<td>21</td>
<td>27</td>
<td>5</td>
<td>14</td>
</tr>
<tr>
<td>MH.02</td>
<td>22</td>
<td>45</td>
<td>25</td>
<td>12</td>
<td>3</td>
<td>13</td>
</tr>
<tr>
<td>MH.03</td>
<td>24</td>
<td>23</td>
<td>25</td>
<td>13</td>
<td>12</td>
<td>21</td>
</tr>
<tr>
<td>MH.04</td>
<td>34</td>
<td>37</td>
<td>49</td>
<td>23</td>
<td>13</td>
<td>22</td>
</tr>
<tr>
<td>MH.05</td>
<td>47</td>
<td>48</td>
<td>52</td>
<td>35</td>
<td>16</td>
<td>23</td>
</tr>
<tr>
<td>V1.01</td>
<td>9</td>
<td>34</td>
<td>10</td>
<td>7</td>
<td>7</td>
<td>6</td>
</tr>
<tr>
<td>V1.02</td>
<td>20</td>
<td>20</td>
<td>10</td>
<td>10</td>
<td>11</td>
<td>7</td>
</tr>
<tr>
<td>V1.03</td>
<td>24</td>
<td>67</td>
<td>14</td>
<td>13</td>
<td>×</td>
<td>17</td>
</tr>
<tr>
<td>V2.01</td>
<td>13</td>
<td>10</td>
<td>12</td>
<td>8</td>
<td>7</td>
<td>8</td>
</tr>
<tr>
<td>V2.02</td>
<td>16</td>
<td>16</td>
<td>14</td>
<td>8</td>
<td>×</td>
<td>10</td>
</tr>
<tr>
<td>V2.03</td>
<td>29</td>
<td>113</td>
<td>14</td>
<td>21</td>
<td>×</td>
<td>27</td>
</tr>
</tbody>
</table>

**FUSES: Fast Unconstrained SEmidefinite Solver**

- Fastest MRF solver: outperforms state of the art by 2-3x
- Near-optimal solution (typically 0.1% from opt.)
- Same approach can be applied to 3D mesh segmentation.
- Evaluation on Cityscapes dataset

Markov Random Field (MRF): assign a discrete label to each node given

\[
\min_{x_i \in \mathcal{L}} \sum_{i \in \mathcal{V}} E_i(x_i) + \sum_{(i,j) \in \mathcal{B}} E_{ij}(x_i, x_j)
\]

**PERFORMANCE ON THE CITYSCAPES DATASET (1000 SUPERPIXELS)**

<table>
<thead>
<tr>
<th>Method</th>
<th>Suboptimality</th>
<th>Accuracy (IoU)</th>
<th>Runtime (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Optimal Labels (%)</td>
<td>Relax Gap (%)</td>
<td>Round Gap (%)</td>
</tr>
<tr>
<td>FUSES</td>
<td>99.17</td>
<td>2.584</td>
<td>0.047</td>
</tr>
<tr>
<td>DARS</td>
<td>99.68</td>
<td>0.210</td>
<td>0.010</td>
</tr>
<tr>
<td>(\alpha)-exp</td>
<td>99.93</td>
<td>-</td>
<td>8.08e-4</td>
</tr>
<tr>
<td>LBP</td>
<td>99.99</td>
<td>-</td>
<td>1.74e-4</td>
</tr>
<tr>
<td>TRW-S</td>
<td>100.00</td>
<td>-</td>
<td>-8.31e-6</td>
</tr>
<tr>
<td>Bonnet (SP)</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>


Open-Source C++ code: [https://github.com/MIT-SPARK/FUSES](https://github.com/MIT-SPARK/FUSES)
Current Datasets: lack of at least one sensor modality

- Most datasets lack one of the following requirements:
  - Stereo Images
  - IMU data (synchronized with images)
  - 2D Semantic annotations
- Only KITTI satisfies requirements, but… just 200 labeled images, and poor IMU data synchronization.
- What about synthetic data simulators:
  Unfortunately, few simulators support modelling IMU data + Semantics:
  - Gazebo: but it does not provide photorealistic images…
  - FlightGoggles: IMU and photorealistic images, missing ground-truth semantic annotations.
  - AirSim: lacks comprehensive ROS support

Introducing our own
Photorealistic + Physics Simulator…
Photorealistic Physics Simulator:
Joint work with MIT Lincoln Labs: Benjamin Smith, Dan Griffith
Photorealistic Physics Simulator:
Joint work with MIT Lincoln Labs: Benjamin Smith, Dan Griffith
Photorealistic Physics Simulator:
Joint work with MIT Lincoln Labs: Benjamin Smith, Dan Griffith

- 2D Semantic Segmentation
- 3D Dense Stereo Reconstruction
- Global Semantic 3D Mesh

Diagram:
- Calib stereo → Photorealistic Physics Simulator → Stereo Images → CRF Optimization → 3D Semantic Mesh
- GT Poses → Stereo Images → Dense Stereo Reconstruction
- GT Semantics → Dense Stereo Reconstruction
- Semantic 3D Pointcloud → TSDF
- Marching Cubes

Joint work with MIT Lincoln Labs: Benjamin Smith, Dan Griffith
Kitti Results

Results in Kitti, with ground-truth poses but 2D semantic labels estimated using real-time ESPNetv2 [1]:

[1] Sachin Mehta and Mohammad Rastegari and Linda G. Shapiro and Hannaneh Hajishirzi “ESPNetv2: A Light-weight, Power Efficient, and General Purpose Convolutional Neural Network”
Future Work: solving 2D Semantic Segmentation failures

State-of-the-art 2D semantic segmentation techniques fail in a number of scenarios.
Future Work: solving 2D Semantic Segmentation failures
Future Work: solving dense 3D reconstruction failures

- Traditional SLAM fails in a number of cases as well:
  - Low-texture
  - Specularities, reflections
  - Low parallax

“Google employs a small army of human operators to manually check and correct the maps” [Wired]
Real-Time Metric-Semantic 3D Mesh SLAM: the ultimate perception pipeline?

- Might hold the key to perfecting:
  - 3D semantic segmentation
  - 3D geometry estimation
  - 3D localization
- On top of that Real-Time? Truly Disruptive…