
Luca Carlone
Lecture 25: Advanced topics: Dense 3D Reconstruction
Big Picture

Trajectory Planning:
- Path planning
- Trajectory optimization

Desired trajectory:

Controller:
- Control inputs

Robot:
- Robot's state

Sensors (e.g., cameras):
- Estimator (e.g., Visual Odometry)
- Sensors (e.g., cameras)

Map and current robot state:
- SLAM (e.g., pose graph optimization)
- Loop closure detection (e.g., place recognition)

Goal:
- >30Hz

Map and current robot state:
- ~1Hz
Big Picture

Trajectory Planning

Path planning → Trajectory optimization → desired trajectory → Controller → Robot

control inputs

Estimator (e.g., Visual Odometry)

Sensors (e.g., cameras)

Dense 3D reconstruction

SLAM (e.g., pose graph optimization)

Loop closure detection (e.g., place recognition)

map and current robot state

Goal

>30Hz

~1Hz

Today!
Today

- Dense Reconstruction
- 3D representations
- (Some) Multi-view Stereo
- Depth fusion
- Final thoughts

Figure 1 in R. A. Newcombe et al., "KinectFusion: Real-time dense surface mapping and tracking," 2011 10th IEEE International Symposium on Mixed and Augmented Reality, Basel, Switzerland, 2011, pp. 127-136, doi: 10.1109/ISMAR.2011.6092378. © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/fair-use/

Multi-View Stereo: A Tutorial

Yasutaka Furukawa
Washington University in St. Louis
furukawa@wustl.edu
Carlos Hernández
Google Inc.
carlos.hernandez@google.com

2015

2016

ElasticFusion: Dense SLAM Without A Pose Graph

Thomas Whelan*, Stefan Leutenegger*, Renato F. Salas-Moreno1, Ben Glocker1 and Andrew J. Davison*
*Dyson Robotics Laboratory at Imperial College, Department of Computing, Imperial College London, UK
1Department of Computing, Imperial College London, UK
{t.whelan, s.leutenegger, r. salas-moreno1, b.glocker, a.davison}@imperial.ac.uk

Figure 1 in H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 1366-1373, doi: 10.1109/IROS.2017.8020315. © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/fair-use/

Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning

Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto
Autonomous Systems Lab, ETH Zürich

Abstract—Micro Aerial Vehicles (MAVs) that operate in unstructured, unexplored environments require fast and flexible local planning, which can replan when new parts of the map are explored. Trajectory optimization methods fulfill these needs, but require obstacle distance information, which can be given by Euclidean Signed Distance Fields (ESDFs).

We propose a method to incrementally build ESDFs from Truncated Signed Distance Fields (TSDFs), a common implicit surface representation used in computer graphics and vision. TSDFs are fast to build and smooth out sensor noise over many observations, and are designed to produce surface meshes. Meshes allow human operators to get a better assessment of the robot’s environment, and set high-level mission goals.
Point Clouds

- Map representation
- 3D Topology
- Lightweight?
- Filters
- Noise/Outliers?
- Semantics?

- ✓
- ✓/
- ✓/
- ✓/
- ✓/

- No, if Dense
- No, if Sparse
### Point Clouds

<table>
<thead>
<tr>
<th></th>
<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>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No, if Dense</td>
<td>No, if Sparse</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The table above summarizes the characteristics of Point Clouds. The symbols ✓ and ✗ indicate whether the characteristic is present or absent, respectively. The symbol ✓/✗ indicates that the characteristic is present for Sparse Point Clouds and absent for Dense Point Clouds.
## Geometric Primitives

Point, lines, planes

![Raw data map](image1.png)

![Planes extracted](image2.png)

![Plane boundaries](image3.png)

### Table of Map Representations

<table>
<thead>
<tr>
<th></th>
<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>
<td>✓</td>
</tr>
<tr>
<td></td>
<td></td>
<td>No, if Dense</td>
<td>No, if Sparse</td>
<td>No, if Sparse</td>
<td></td>
</tr>
<tr>
<td>Geometric primitives</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓/✗</td>
<td>✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>No, if Sparse</td>
<td>No, if Sparse</td>
<td></td>
</tr>
</tbody>
</table>
# Object-based Maps

[Salas-Moreno et al, 2014]

Figure 3 in R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. J. Kelly and A. J. Davison, "SLAM++: Simultaneous Localisation and Mapping at the Level of Objects," 2013 IEEE Conference on Computer Vision and Pattern Recognition, Portland, OR, USA, 2013, pp. 1352-1359, doi: 10.1109/CVPR.2013.178 © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/

<table>
<thead>
<tr>
<th></th>
<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>
<td>✓</td>
</tr>
<tr>
<td>primitives &amp; objects</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓/✗</td>
<td>✗</td>
</tr>
</tbody>
</table>
# Volumetric Methods: Voxels/Octrees

![Image](image.png)

[Oleynikova, ICRA’17](#)

© Oleynikova et al. All rights reserved. This content is excluded from our Creative Commons license. For more information, see [https://ocw.mit.edu/help/faq-fair-use/](https://ocw.mit.edu/help/faq-fair-use/)

<table>
<thead>
<tr>
<th></th>
<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>✗/✗ No, if Sparse</td>
<td>✓</td>
</tr>
<tr>
<td>primitives &amp; objects</td>
<td>✗</td>
<td>✓</td>
<td>✓</td>
<td>✓/✗ No, if Sparse</td>
<td>✗</td>
</tr>
<tr>
<td>Voxels</td>
<td>✓</td>
<td>✓/✗ No, if small voxel</td>
<td>✓</td>
<td>✓/✗ No, if large</td>
<td>✓</td>
</tr>
</tbody>
</table>
## Meshes

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Point Clouds</td>
<td>X</td>
<td>✓/✗</td>
<td>✓/✗</td>
<td>✓/✗</td>
<td>✓/✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No, if Sparse</td>
<td>No, if Sparse</td>
</tr>
<tr>
<td>primitives &amp; objects</td>
<td>X</td>
<td>✓</td>
<td>✓</td>
<td>✓/✗</td>
<td>✓/✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No, if Sparse</td>
<td></td>
</tr>
<tr>
<td>Voxels</td>
<td>✓</td>
<td>✓/✗</td>
<td>✓</td>
<td>✓/✗</td>
<td>✓/✗</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No, if large voxel</td>
<td></td>
</tr>
<tr>
<td>3D Mesh</td>
<td>✓</td>
<td>✓</td>
<td>✓/✗</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>
Today

- Dense Reconstruction
  - 3D representations
  - (Some) Multi-view Stereo
  - Depth fusion
- Final thoughts

Multi-View Stereo: A Tutorial

2015

Yasutaka Furukawa
Washington University in St. Louis
furukawa@wustl.edu

Carlos Hernández
Google Inc.
carlos.hernandez@gmail.com

ElasticFusion: Dense SLAM Without A Pose Graph

Thomas Whelan*, Stefan Leutenegger*, Renato F. Salas-Moreno1, Ben Groover1 and Andrew J. Davison*
*Dyson Robotics Laboratory at Imperial College, Department of Computing, Imperial College London, UK
1Department of Computing, Imperial College London, UK
{t.whelan.s.leutenegger.r.salas-moreno1.b.groover.a.davison}@imperial.ac.uk

Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning

Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto
Autonomous Systems Lab, ETH Zürich

Abstract—Micro Aerial Vehicles (MAVs) that operate in unstructured, unexplored environments require fast and flexible local planning, which can replan when new parts of the map are explored. Trajectory optimization methods fulfill these needs, but require obstacle distance information, which can be given by Euclidean Signed Distance Fields (ESDFs).

We propose a method to incrementally build ESDFs from Truncated Signed Distance Fields (TSDFs), a common implicit surface representation used in computer graphics and vision. TSDFs are fast to build and smooth out sensor noise over many observations, and are designed to produce surface meshes. Meshes allow human operators to get a better assessment of the robot’s environment, and set high-level mission goals.
Multi-view Stereo

From previous lectures: we know how to use SLAM to get a good estimate of the poses of the cameras.
Towards Internet-scale Multi-view Stereo

CVPR 2010

Yasutaka Furukawa¹  Brian Curless²
Steven M. Seitz¹,²  Richard Szeliski³

Google Inc.¹
University of Washington²
Microsoft Research³
Multi-view Stereo

The Visual Turing Test for Scene Reconstruction
Supplementary Video

Qi Shan†    Riley Adams†    Brian Curless†
Yasutaka Furukawa*    Steve Seitz**

†University of Washington    *Google

3DV 2013
Multi-view Stereo

Patch-based methods:

Estimate normal and center of patch to maximize photometric consistency:

\[ C_{ij}(p) = \rho(I_i(\Omega(\pi_i(p))), I_j(\Omega(\pi_j(p)))) \]

Example of matching score:

\[ 1 - \sum_{x,y} |W_1(x, y) - W_2(x, y)|^2 \]

[Furukawa and Ponce, “Accurate, Dense, and Robust Multi-View Stereopsis”, 2007]
Multi-view Stereo

**Enforcing regularity**: Markov Random Fields
Find depth $k_p$ of point “p” such that point is photo-consistent and depth changes smoothly.

$$E(\{k_p\}) = \sum_{p} \Phi(k_p) + \sum_{(p,q) \in \mathcal{N}} \Psi(k_p, k_q)$$

Unary potentials (similar to previous slides)
$$\Phi(k_p = d) = \min(\tau_u, 1 - C(p, d))$$

Pairwise potentials
$$\Psi(k_p = d_1, k_q = d_2) = \min(\tau_p, |d_1 - d_2|)$$

Depth is typically discretized before solving..
How Accurate is Multi-view Stereo?

Many methods: volumetric stereo, space carving, Shape from silhouettes, carved visual hull
Today

- Dense Reconstruction
  - 3D representations
  - (Some) Multi-view Stereo
  - Depth fusion

- Final thoughts

Figure 1 in H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for on-board MAV planning," 2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Vancouver, BC, 2017, pp. 1366-1373, doi: 10.1109/IROS.2017.8202315. © IEEE. All rights reserved. This content is excluded from our Creative Commons license. For more information, see https://ocw.mit.edu/help/faq-fair-use/
Surfels

ElasticFusion: Dense SLAM Without A Pose Graph

Thomas Whelan, Stefan Leutenegger, Renato Salas-Moreno, Ben Glocker, Andrew Davison

Imperial College London

note: still based on RGB-D (contrarily to multi-view stereo)
A Gentle Start: 2D Occupancy Grid Maps

- discretize the environment into cells
- Each cell holds real number $[0,1]$, representing the probability of the cell being occupied

Map posterior

$$p(m \mid z_{1:t}, x_{1:t})$$

- Unknown Map
- Known sensor Depth and robot poses
A Gentle Start: 2D Occupancy Grid Maps

Bayes rule (omitting “x” for simplicity):

$$p(m_i | z_{1:t+1}) = \frac{p(z_{t+1} | m_i)p(m_i | z_{1:t})}{p(m_i)}$$

Log-odd representation is typically used to avoid numerical instabilities

Uninformative Prior

Probability of cell being occupied

Binary value (free/occupied)
Truncated Signed Distance Function (SDF)

- Store distance to nearest obstacle (with sign)
- Only update around obstacle itself
  (implicit surface model)

**Update rule:**

\[
\begin{align*}
    d(x, p, s) & = \| p - x \| \text{sign} \left( (p - x) \cdot (p - s) \right) \\
    w_{\text{const}}(x, p) & = 1 \\
    D_{i+1}(x, p) & = \frac{W_i(x)D_i(x) + w(x, p)d(x, p)}{W_i(x) + w(x, p)} \\
    W_{i+1}(x, p) & = \min(W_i(x) + w(x, p), W_{\text{max}})
\end{align*}
\]

Kinect Fusion (2011)

SIGGRAPH Talks 2011

KinectFusion: Real-Time Dynamic 3D Surface Reconstruction and Interaction

Shahram Izadi 1, Richard Newcombe 2, David Kim 1,3, Otmar Hilliges 1, David Molyneaux 1,4, Pushmeet Kohli 1, Jamie Shotton 1, Steve Hodges 1, Dustin Freeman 5, Andrew Davison 2, Andrew Fitzgibbon 1

1 Microsoft Research Cambridge 2 Imperial College London 3 Newcastle University 4 Lancaster University 5 University of Toronto
Kintinuous (2013)

GPU, bounded memory ...
VoxBlox (2017)
From Voxels to Meshes

Marching cubes

https://www.youtube.com/watch?v=B_xk71YopsA
Kimera (2020)

Kimera-VIO tracks sparse 3D landmarks for fast and accurate state estimation.
Metric-semantic 3D Reconstruction

Kimera-Semantics

Kimera-VIO

Kimera-Mesher

Kimera-RPGO
Today

- Dense Reconstruction
  - 3D representations
  - (Some) Multi-view Stereo
  - Depth fusion
- Final thoughts

ElasticFusion: Dense SLAM Without A Pose Graph

Thomas Whelan*, Stefan Leutenegger*, Renato F. Salas-Moreno1, Ben Glocker1 and Andrew J. Davison*

*Dyson Robotics Laboratory at Imperial College, Department of Computing, Imperial College London, UK
1Department of Computing, Imperial College London, UK

For more information, see https://ocw.mit.edu/help/fair-use/

Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning

Helen Oleynikova, Zachary Taylor, Marius Fehr, Roland Siegwart, and Juan Nieto

Autonomous Systems Lab, ETH Zürich

Abstract—Micro Aerial Vehicles (MAVs) that operate in unstructured, unexplored environments require fast and flexible local planning, which can only be achieved when new parts of the map are explored. Trajectory optimization methods fulfill these needs, but require obstacle distance information, which can be given by Euclidean Signed Distance Fields (ESDFs).

We propose a method to incrementally build ESDFs from Truncated Signed Distance Fields (TSDFs), a common implicit surface representation used in computer graphics and vision. TSDFs are fast to build and smooth out sensor noise over many observations, and are designed to produce surface meshes. These allow human operators to get a better assessment of the robot’s environment, and set high-level mission goals.
Robot Perception or Computer Vision?

Computer vision

.. “a day on a cluster with 500 compute cores”

Robotics

50-100ms latency, embedded, incremental

No longer a dichotomy for many vision applications!
Robot Perception or Computer Vision?

Unordered Vs Sequential