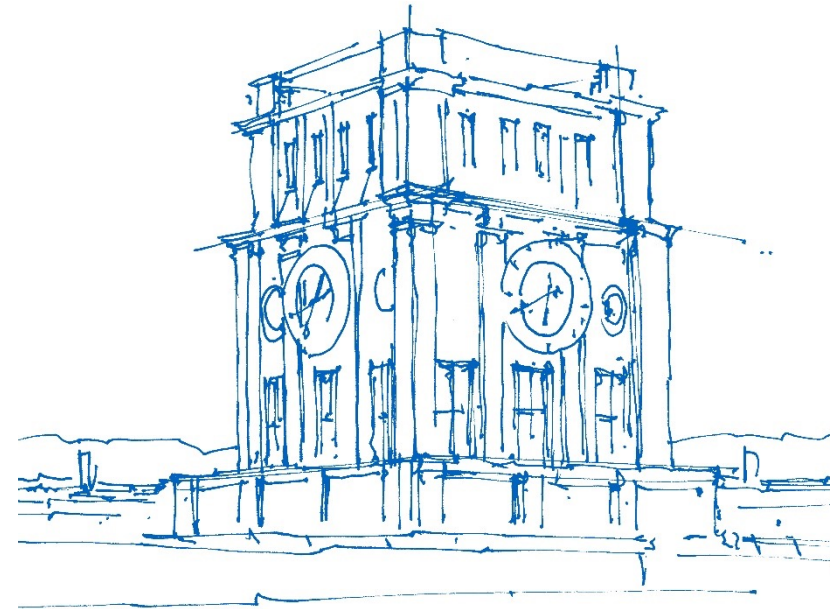


Real-Time Volumetric Mapping

- Wenjie Zheng
- Munich, 15. Jan 2024



Uhrenturm der TUM

AGENDA

1 INTRODUCTION

2 DEVELOPMENT OF REAL-TIME 3D MAPPING

3 METHODS DESCRIPTION

4 EXPERIMENTS AND RESULTS

5 COMMENTS AND FUTURE

2 INTRODUCTION

Development History of 3D Reconstruction



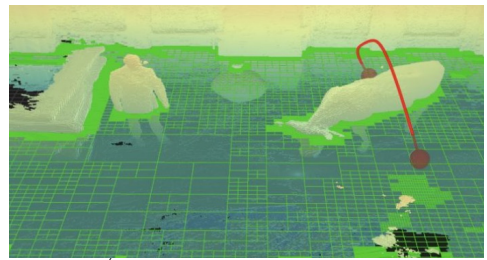
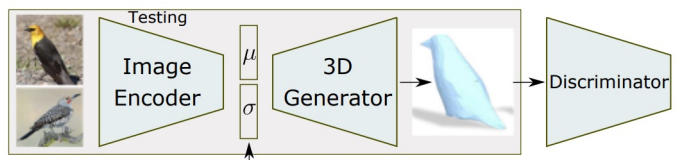
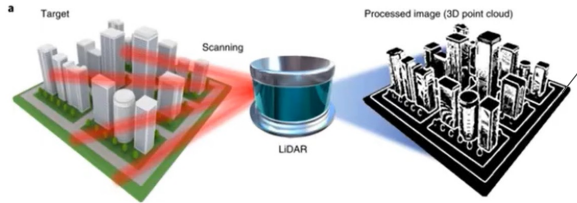
Early Computer Graphics Era • 1960s

Introduction of Lidar and 3D Scanners • 1990s

Sparse 3D Reconstruction(SLAM) • 2010s

Dense 3D Reconstruction and Deep Learning • 2010s

Real-Time Volumetric Mapping • 2010s



2 INTRODUCTION

Development History of 3D Reconstruction



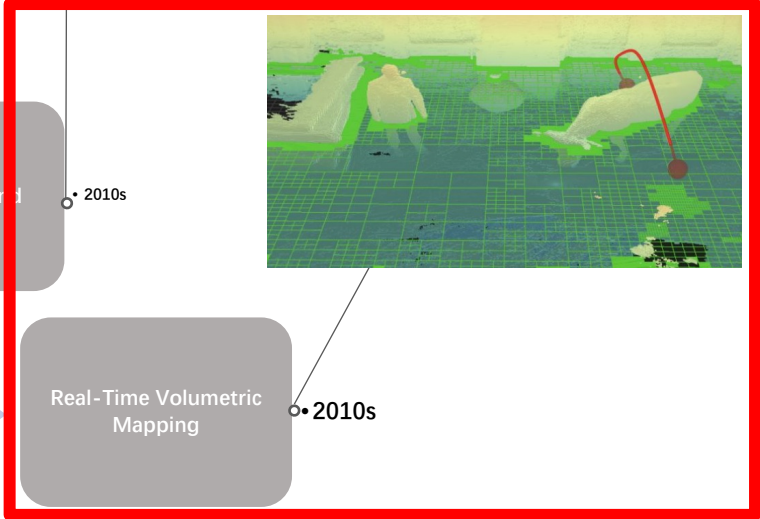
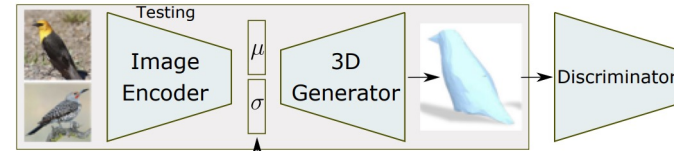
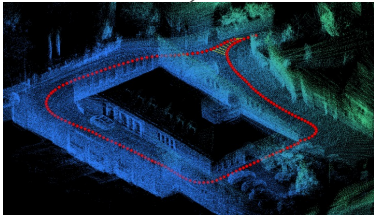
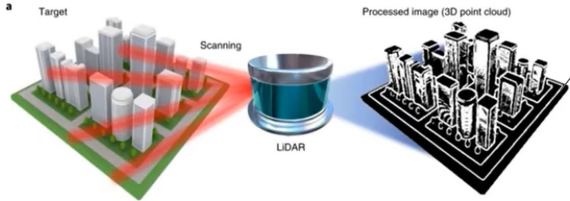
Early Computer Graphics Era • 1960s

Introduction of Lidar and 3D Scanners (high quality 3D point cloud) • 1990s

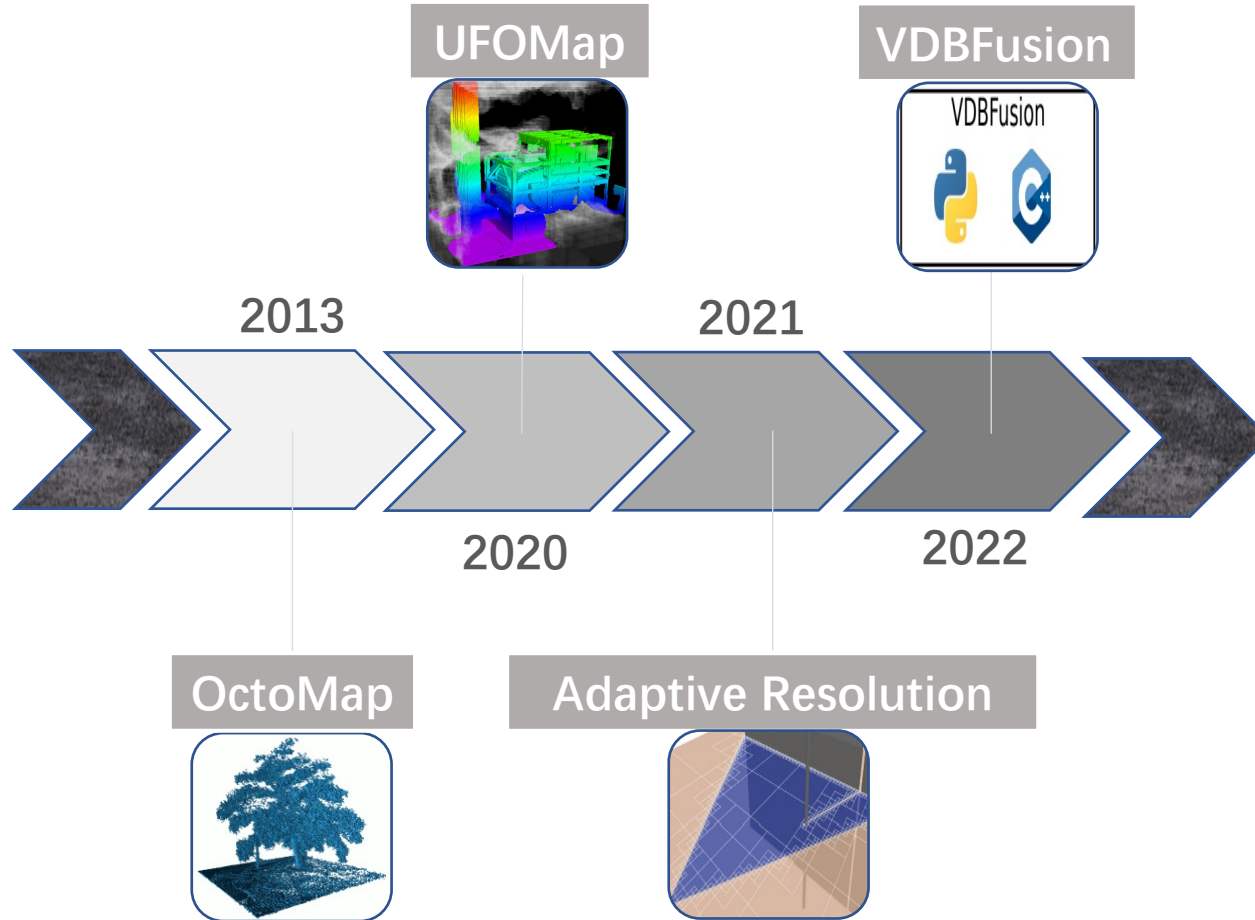
Sparse 3D Reconstruction(SLAM) • 2010s

Dense 3D Reconstruction and Deep Learning • 2010s

Real-Time Volumetric Mapping • 2010s

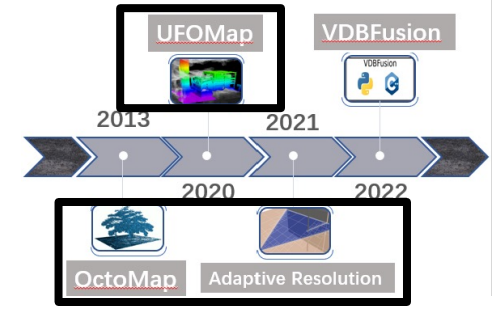


2 DEVELOPMENT REAL-TIME 3D MAPPING



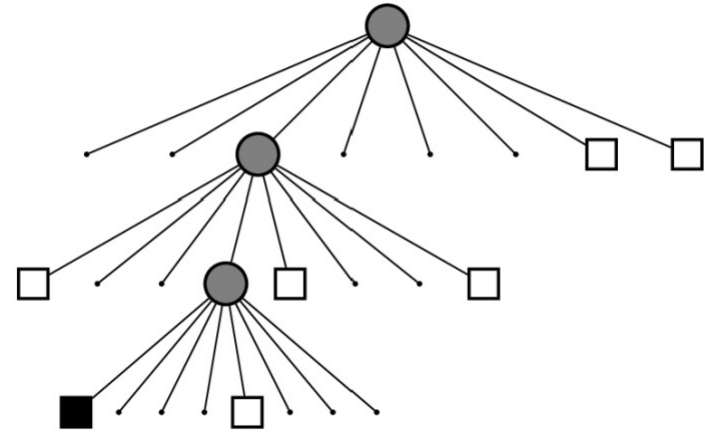
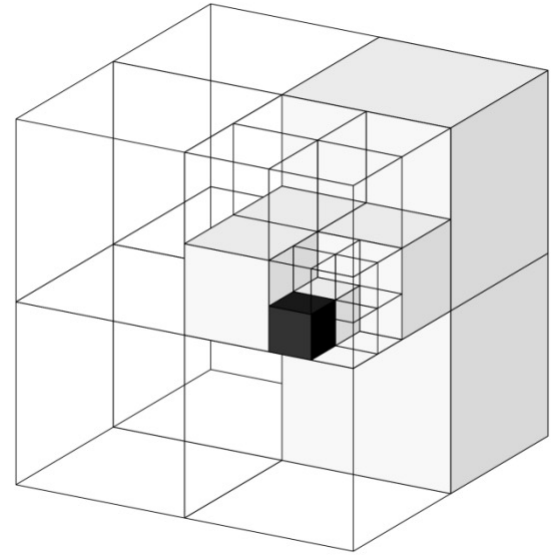
2 DEVELOPMENT REAL-TIME 3D MAPPING

Related Work



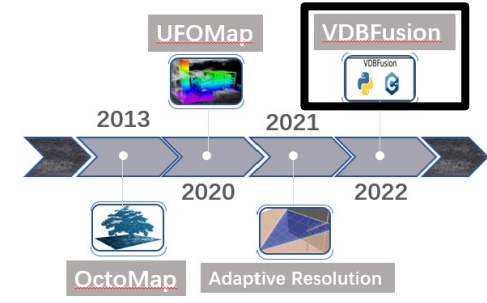
Octree

- Tree-based data structure
- Recursive subdivision of space
- Volumes allocated as needed
- Multi-resolution



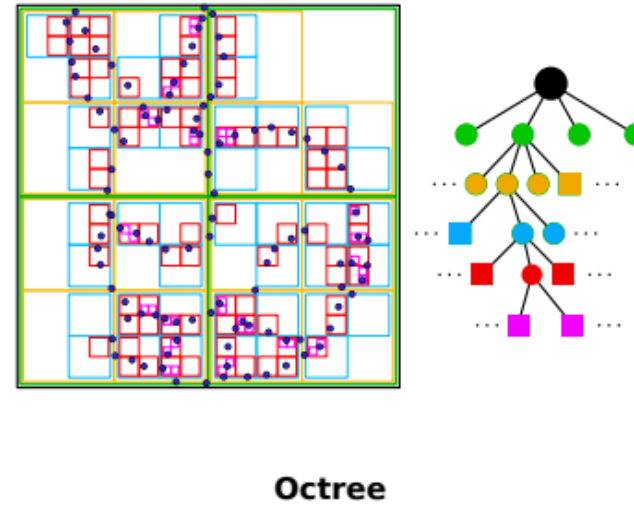
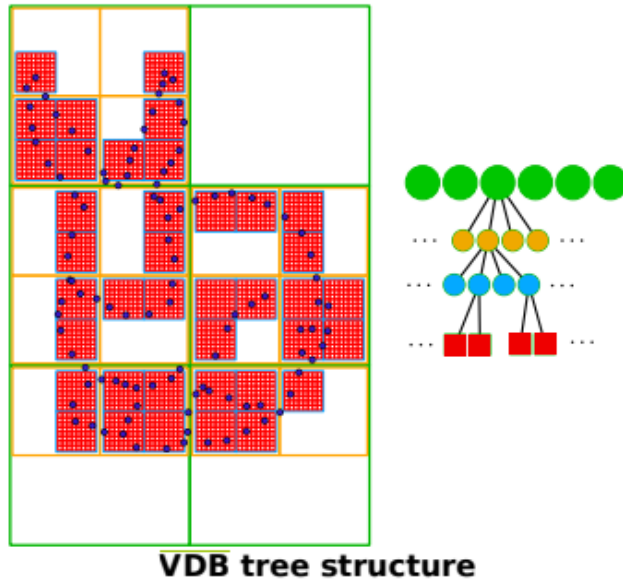
2 DEVELOPMENT REAL-TIME 3D MAPPING

Related Work

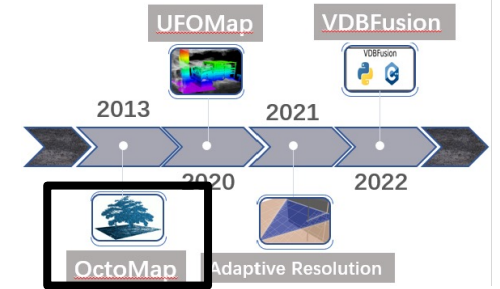


VDB Tree

- Fixed depth



- Fix Depth to divide

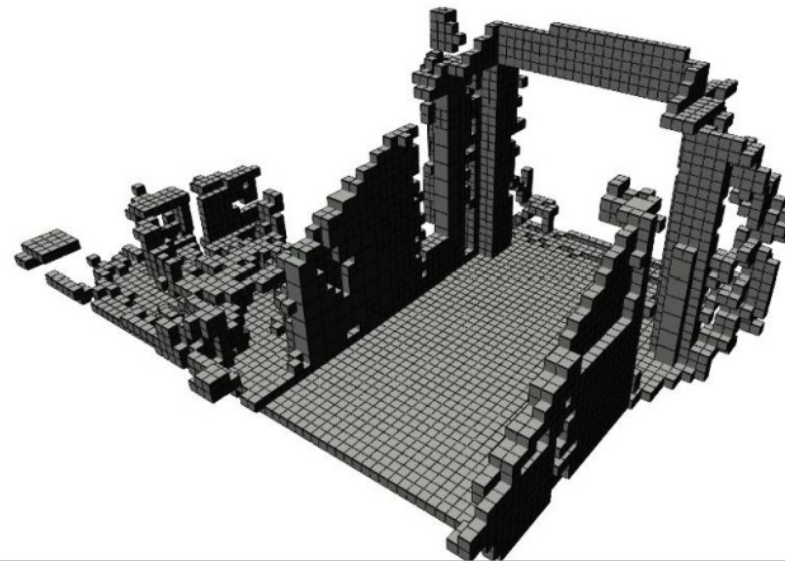


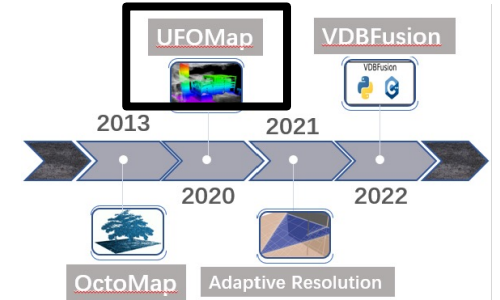
OctoMap

- Based on octree data structure
- Occupancy Probabilities for each node including free and occupied

Occupancy probability $P(n | z_{1:t}) = \left[1 + \frac{1 - P(n | z_t)}{P(n | z_t)} \frac{1 - P(n | z_{1:t-1})}{P(n | z_{1:t-1})} \frac{P(n)}{1 - P(n)} \right]^{-1}$

Occupancy Update $L(n | z_{1:t}) = L(n | z_{1:t-1}) + L(n | z_t)$ *log-odds notation*



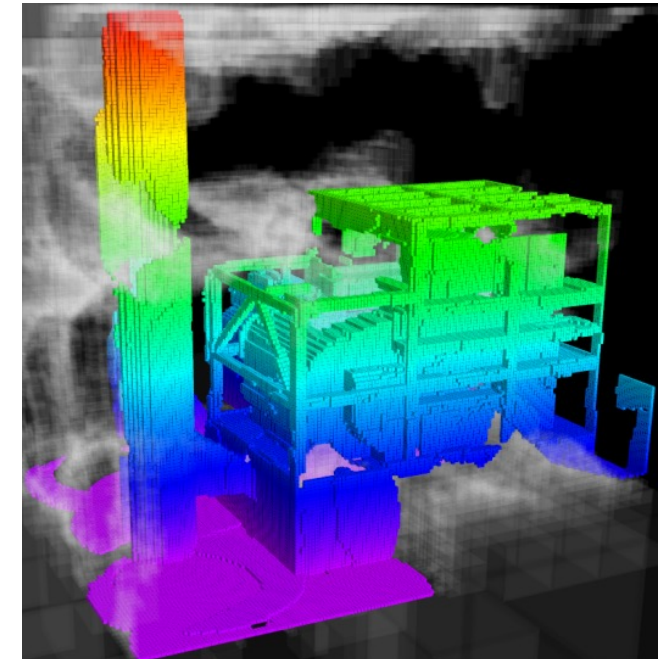


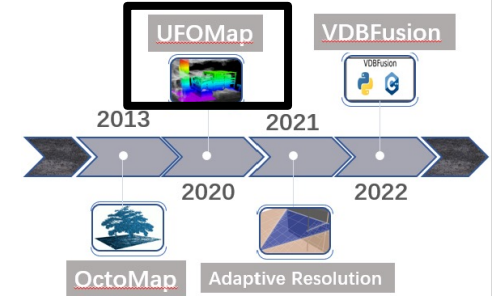
UFOMap

- Based on Octree
- ➔ Probabilistic representation of occupancy including Free, occupied and unknown space
- Nodes indicators
- 3 faster Methods for insertions of Octree

$$\text{state}(n) = \begin{cases} \text{unknown} & \text{if } t_f \leq \text{occ}(n) \leq t_o \\ \text{occupied} & \text{if } t_o < \text{occ}(n) \\ \text{free} & \text{if } t_f > \text{occ}(n) \end{cases}$$

where $\text{occ}(n)$ is the occupancy value of the node n .



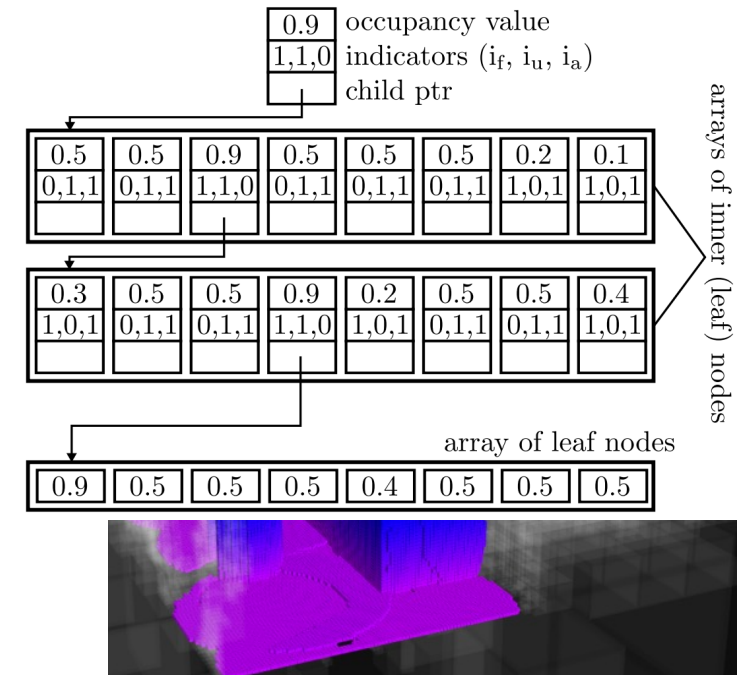


UFOMap

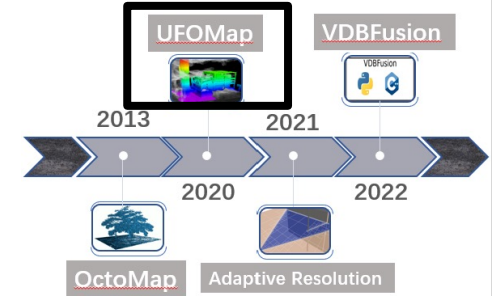
- Based on octrees
- Probabilistic representation of occupancy including Free, occupied and unknown space
- Nodes indicators
- 3 faster Methods for insertions of Octree



Indicator	Meaning
i_f	node contains free space
i_u	node contains unknown space
i_a	all of a node's children are the same



2 METHODS DESCRIPTION



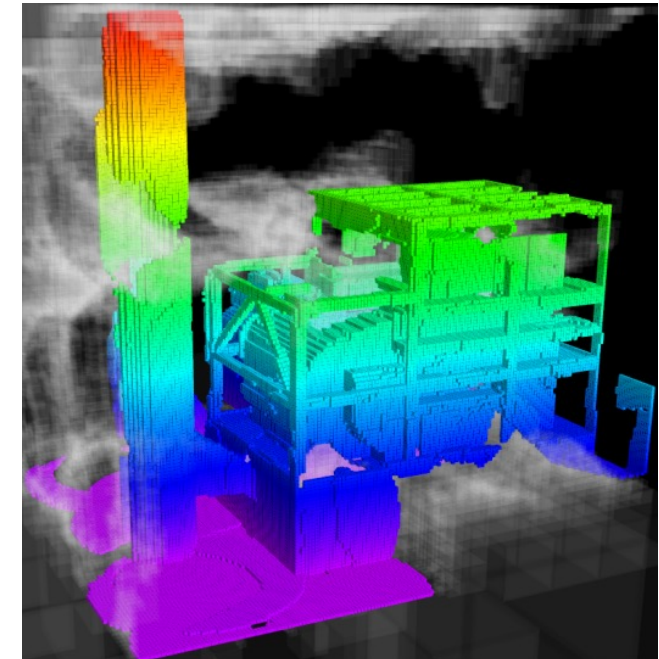
UFOMap

- Experiments and results
- ➔ • Memory Consumption and Node Count
- Insertion

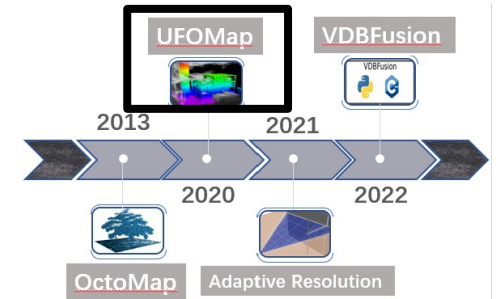
TABLE I

MEMORY CONSUMPTION AND NUMBER OF NODES COMPARISON BETWEEN UFOMAP AND OCTOMAP ON THE OCTOMAP 3D SCAN DATASET.

Dataset	Memory usage			Number of nodes			
	UFOMap (MB)	OctoMap (MB)	Reduction (%)	UFOMap	OctoMap	UFOMap leaves (%)	OctoMap leaves (%)
FR-078 tidyup	7.42	21.49	65.47	1 642 113	1 369 165	80.51	85.01
FR-079 corridor	19.70	51.86	62.01	2 823 713	1 829 134	75.20	80.70
Freiburg campus	58.71	155.46	62.23	8 402 193	5 515 178	75.10	80.96
freiburg1_360	15.98	42.05	62.00	2 161 849	1 547 112	71.72	82.53
New College	29.41	75.40	60.99	4 157 217	2 633 701	74.37	80.27



2 METHOD DESCRIPTION



UFOMap

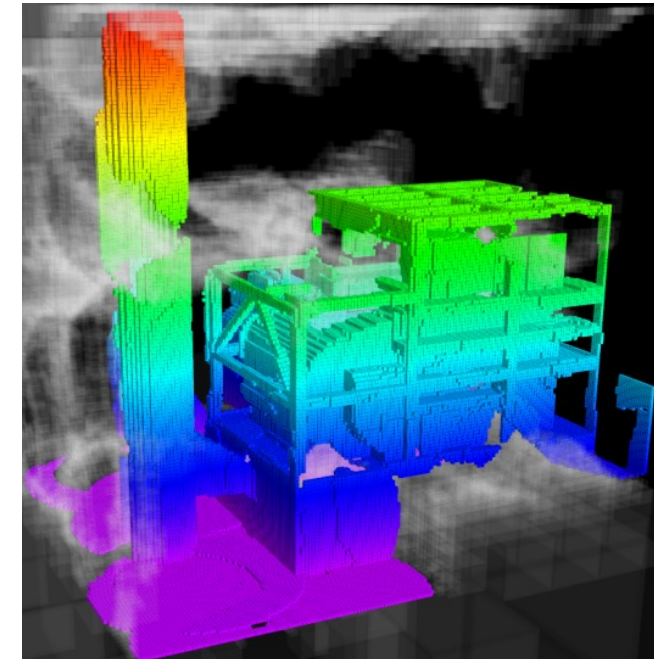
- Experiments and results
 - Memory Consumption and Node Count
 - Insertion

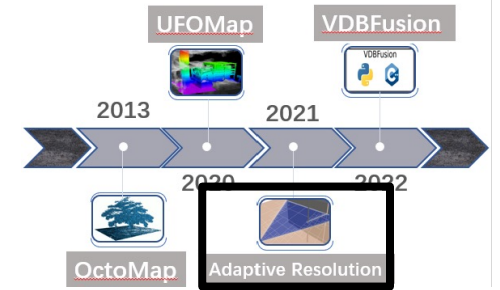


TABLE II
INSERTION TIMINGS COMPARISON BETWEEN UFOMAP AND OCTOMAP,
ON THE COW DATASET², WITH DIFFERENT VOXEL SIZES.

Method	Voxel size (cm)	Total (ms)	Ray tracing (ms)	Insertion (ms)
UFOMap	16	4.98±1.30	4.64±1.08	0.34±0.27
OctoMap		5.52±1.66	4.86±1.18	0.66±0.56
UFOMap	8	12.3±7.4	10.4±5.8	1.9±1.6
OctoMap		16.3±10.4	12.2±6.9	4.1±3.7
UFOMap*		8.1±3.4	6.7±2.3	1.4±1.1
UFOMap†		6.5±2.2	6.0±1.8	0.5±0.4
UFOMap	4	60.9±44.7	46.8±32.7	14.1±12.2
OctoMap		104.6±82.2	71.8±53.0	32.9±30.1
UFOMap*		21.1±12.0	14.3±6.7	6.8±5.4
UFOMap†		10.9±4.7	9.3±3.5	1.6±1.3
UFOMap	2	371±254	264±176	107±79
OctoMap		745±548	521±369	224±188
UFOMap*		74±44	42±22	32±22
UFOMap†		28±15	20±9	9±7

50%



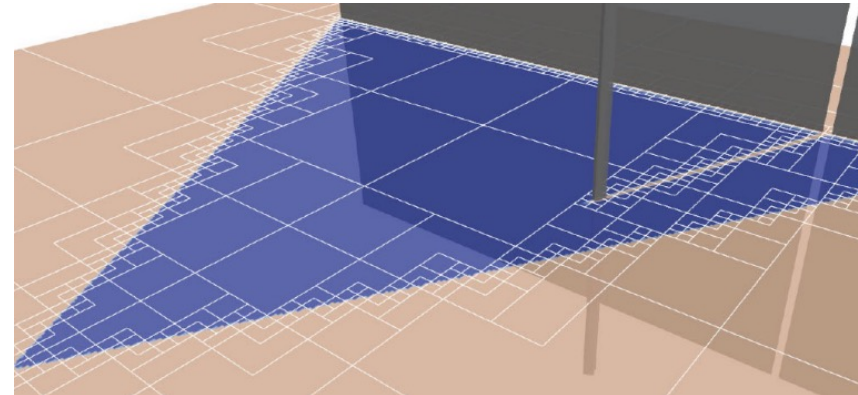


Adaptive-Resolution Mapping

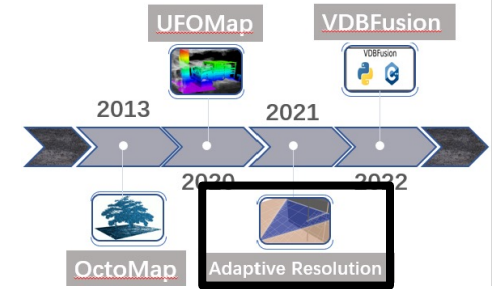
- Based on octrees
- Probabilistic representation of occupancy including Free, occupied and unknown space
- ➔ Adaptive-resolution volumetric mapping
 - Criterion
 - Decision process (Bounding box + Pooling images)

$$\max |l_k(w\mathbf{p}_i) - l_k(w\mathbf{p}_j)| < \varepsilon$$

for every $w\mathbf{p}_i, w\mathbf{p}_j$ in the node volume

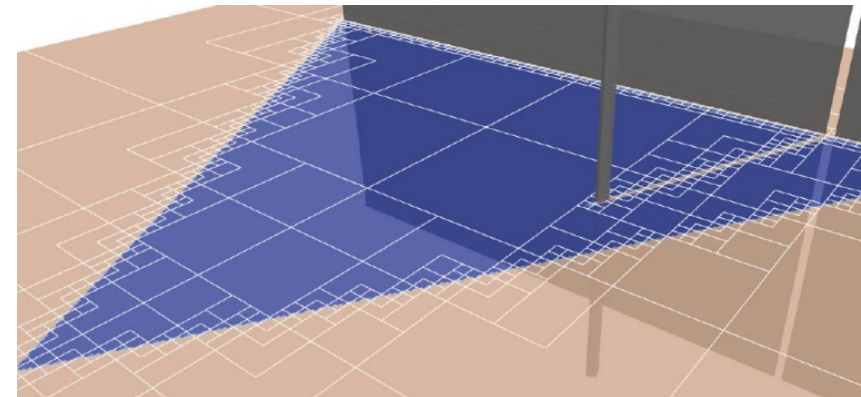
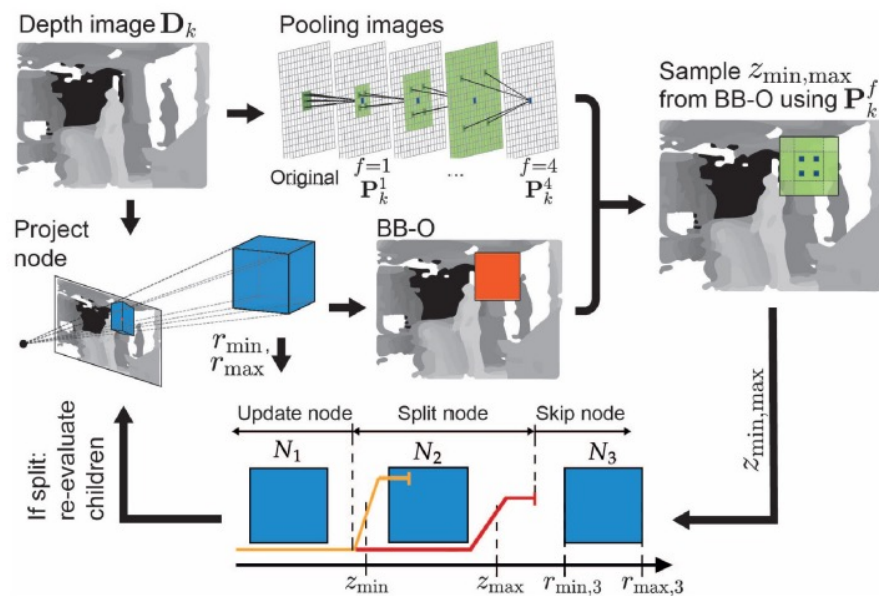


Quote: Multi-Resolution 3D Mapping With Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning

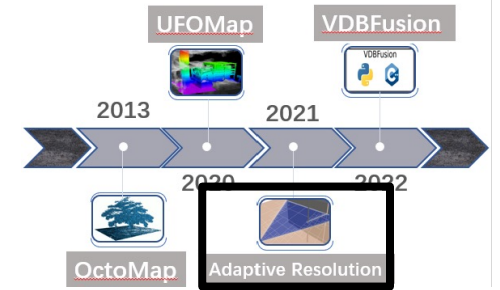


Adaptive-Resolution Mapping

- Based on octrees
- Probabilistic representation of occupancy including Free, occupied and unknown space
- Adaptive-resolution volumetric mapping
 - Criterion
 - Decision process (Bounding box + Pooling images)

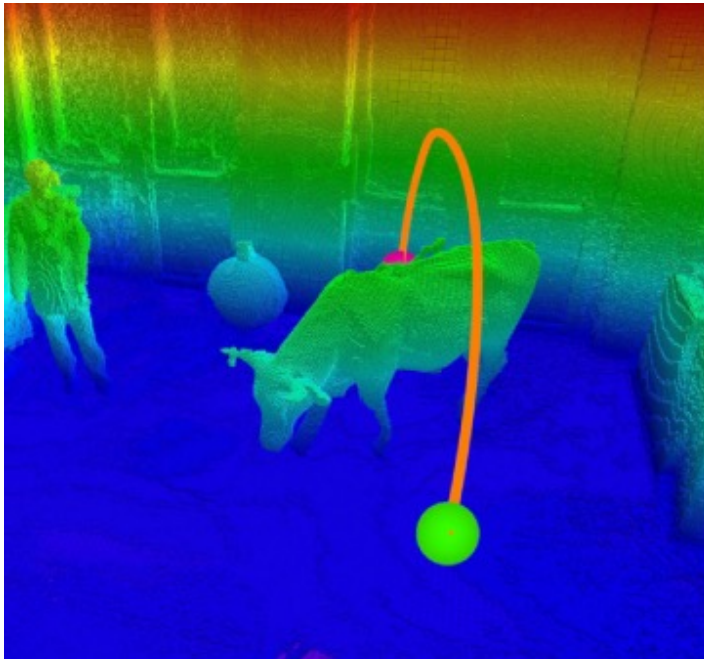


Quote: Multi-Resolution 3D Mapping With Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning

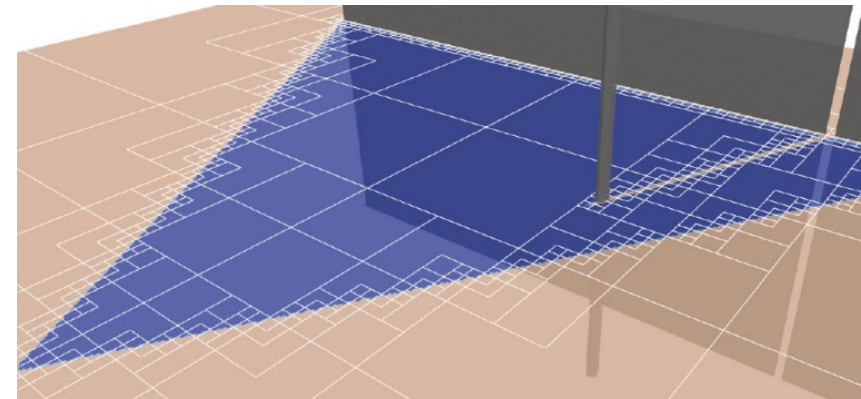


Adaptive-Resolution Mapping

- Experiments and results
 - Tracking performance



Dataset: Cow and Lady



Quote: Multi-Resolution 3D Mapping With Explicit Free Space Representation for Fast and Accurate Mobile Robot Motion Planning

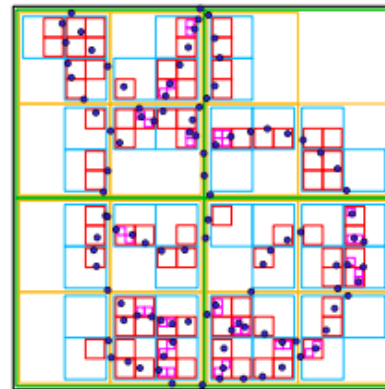
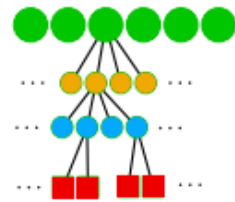
2 METHODS DESCRIPTION

VDBFusion

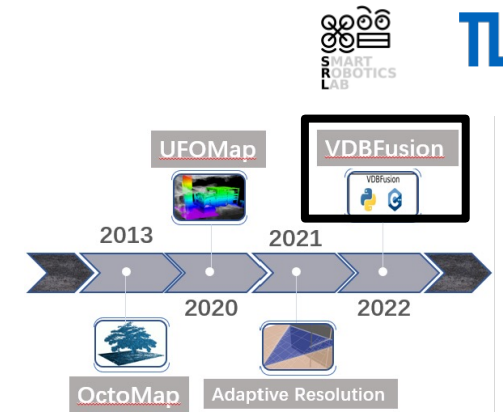
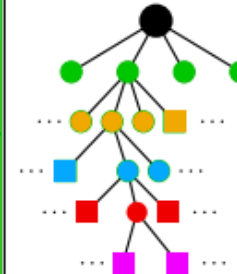
- Based on VDB structure
- TSDF-based mapping



VDB tree structure



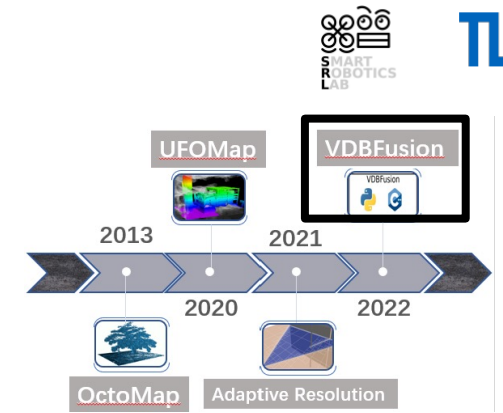
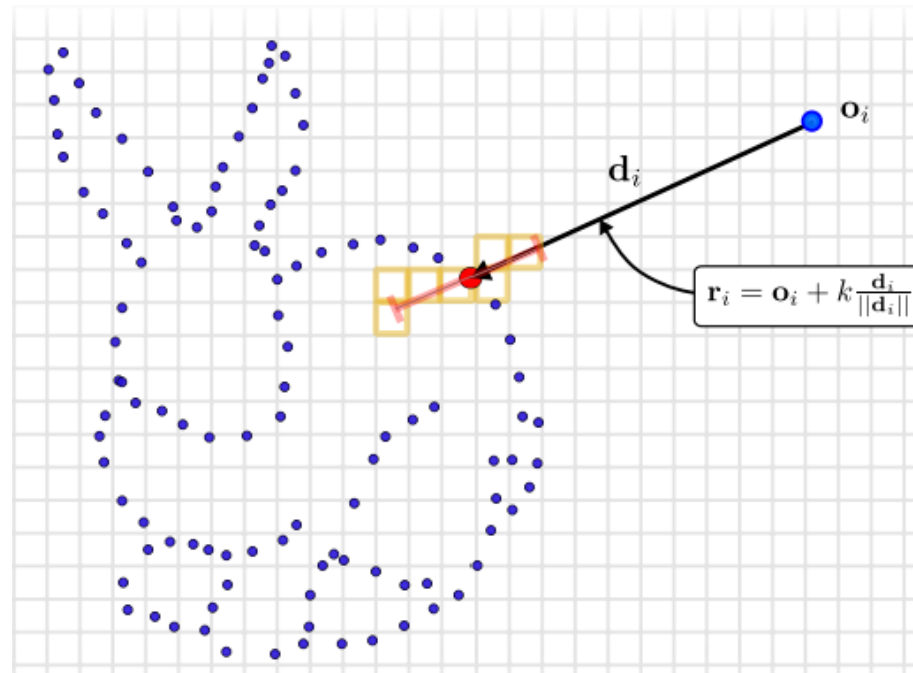
Octree



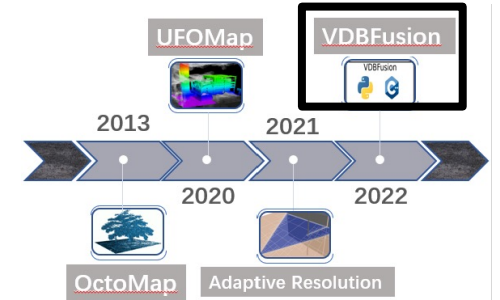
2 METHODS DESCRIPTION

VDBFusion

- Based on VDB tree structure
- TSDF-based mapping

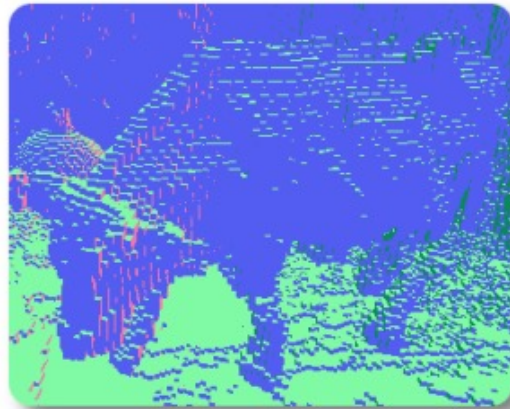


- Memory-efficient
- Fast to access
- Easy to use



VDBFusion

- Experiments and results
 - Tracking performance



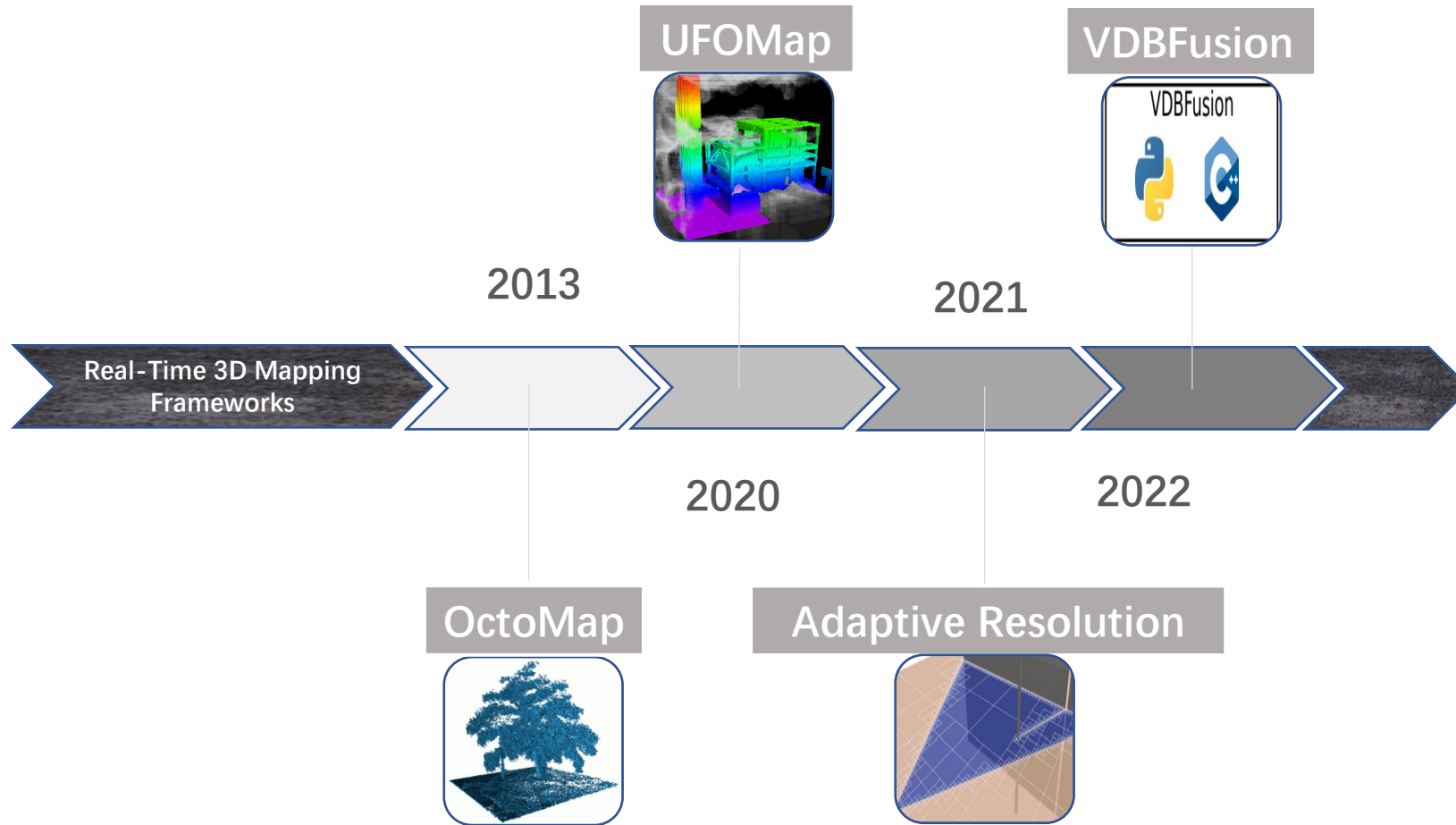
Octomap



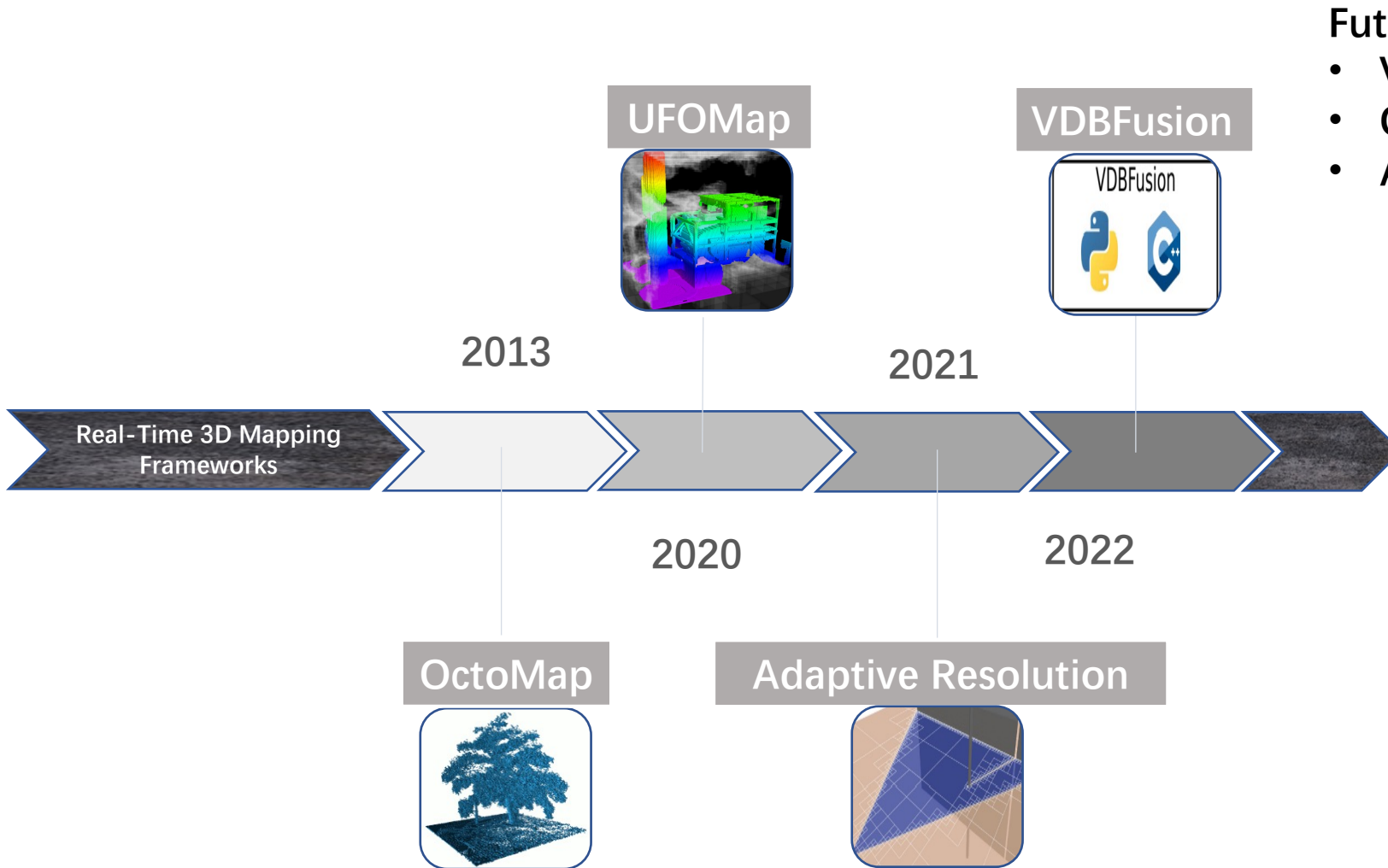
VDBFusion

Higher level of details

5 COMMENTS AND FUTURE



5 COMMENTS AND FUTURE



Future

- VDB structure
- Optimize the Algorithm VDB
- Adaptive resolution