ETH zürich

ETH Library

Voxgraph: Globally Consistent, Volumetric Mapping Using Signed Distance Function Submaps

Journal Article

Author(s):
Reijgwart, Victor; Millane, Alexander; Oleynikova, Helen; Siegwart, Roland; Cadena, Cesar; Nieto, Juan

Publication date:
2020-01

Permanent link:
https://doi.org/10.3929/ethz-b-000385682

Rights / license:
In Copyright - Non-Commercial Use Permitted

Originally published in:
IEEE Robotics and Automation Letters 5(1), https://doi.org/10.1109/lra.2019.2953859

This page was generated automatically upon download from the ETH Zurich Research Collection.
For more information, please consult the Terms of use.

Voxgraph: Globally Consistent, Volumetric Mapping using Signed Distance Function Submaps

Victor Reijgwart*, Alexander Millane*, Helen Oleynikova, Roland Siegwart, Cesar Cadena, Juan Nieto

***Abstract*—Globally consistent dense maps are a key requirement for long-term robot navigation in complex environments. While previous works have addressed the challenges of dense mapping and global consistency, most require more computational resources than may be available on-board small robots. We propose a framework that creates globally consistent volumetric maps on a CPU and is lightweight enough to run on computationally constrained platforms. Our approach represents the environment as a collection of overlapping Signed Distance Function (SDF) submaps, and maintains global consistency by computing an optimal alignment of the submap collection. By exploiting the underlying SDF representation, we generate correspondence-free constraints between submap pairs that are computationally efficient enough to optimize the *global* problem each time a new submap is added. We deploy the proposed system on a hexacopter Micro Aerial Vehicle (MAV) with an Intel i7-8650U CPU in two realistic scenarios: mapping a large-scale area using a 3D LiDAR, and mapping an industrial space using an RGB-D camera. In the large-scale outdoor experiments, the system optimizes a 120x80 m map in less than 4 s and produces absolute trajectory RMSEs of less than 1 m over 400 m trajectories. Our complete system, called *voxgraph*, is available as open source[1].**

***Index Terms*—Mapping, SLAM, Aerial Systems: Perception and Autonomy**

I. Introduction

In order to navigate and interact with their environment, robotic platforms typically build an internal representation of the observed world. This process, Simultaneous Localization And Mapping (SLAM), has been a focus of robotics research over past decades [1]. Many successful systems convert input data to a set of features, and use these as the map representation. Recent systems have shown estimation of globally consistent, feature-based maps in real-time [2], [3]. While feature-based maps have proven indispensable for motion estimation on robotic platforms [4], [5], they are of limited use for tasks beyond localization due to the difficulties of extracting the shape and connectivity of surfaces from a sparse set of samples.

Manuscript received: September, 10, 2019; Revised November, 12, 2019; Accepted October, 28, 2019.

This paper was recommended for publication by Editor S. Behnke upon evaluation of the Associate Editor and Reviewers' comments. This work was supported by the National Center of Competence in Research (NCCR) Robotics through the Swiss National Science Foundation, as well as the Defense Advanced Research Projects Agency (DARPA) under Agreement No. HR00111820045, and Armasuisse.

* The authors contributed equally.

All authors are with the Autonomous Systems Lab, ETH Zürich, Switzerland victorr@ethz.ch; millanea@ethz.ch; helenol@ethz.ch; rsiegwart@ethz.ch; cesarc@ethz.ch; nietoj@ethz.ch

Digital Object Identifier (DOI): 10.1109/LRA.2019.2953859

[1]https://github.com/ethz-asl/voxgraph

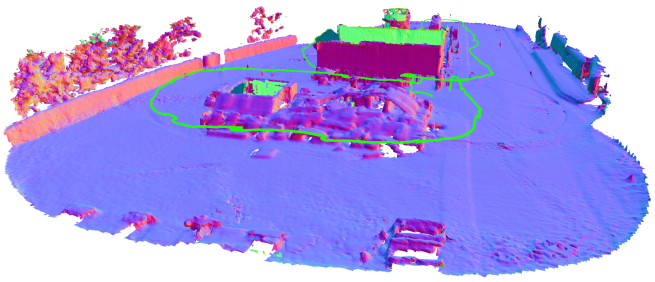

Fig. 1: A reconstruction resulting from a 400 m-long MAV flight through the search and rescue training site discussed in Sec. VIII-B. In the foregound a large pile of rubble from a collapsed structure is visible. The trajectory (green) also contains indoor-outdoor transitions through a building (visible behind rubble).

Several systems build more geometrically complete representations of the environment, producing maps that are useful for tasks such as obstacle avoidance and path planning. Implicit surfaces represented as Signed Distance Functions (SDFs) have emerged as an effective representation for this purpose. The move away from sparse features, however, makes producing a globally consistent map challenging, since global optimization of the map quickly becomes intractable as the amount of data increases. Many existing approaches, therefore, are limited to operation in small-scale environments where drift is limited, or store raw input data such that a globally accurate map can be calculated from scratch as further data becomes available.

This paper introduces a novel system for building globally-consistent volumetric maps on a CPU. Central to our approach is the representation of the world as a set of overlapping SDF submaps. In contrast to similar proposals, we do not compute the full sensor trajectory, but propose a map-centric approach instead. In particular, the proposed system estimates globally consistent submap poses based on the dense map itself, utilizing the underlying SDF representation to perform geometric alignment. We propose geometric constraints that are efficient enough to perform global optimization of the map frequently, and in real-time. Furthermore, we formulate the problem as a graph optimization and include odometry and loop closure constraints. In several experiments we show the efficacy of the resulting system for globally consistent large scale dense mapping and demonstrate its use on several MAV systems.

In summary the contributions of this paper are:

- A novel submap registration method based on the Euclidean Signed Distance Function (ESDF), points on the level-set, and constraint subsampling.
- Inclusion of global loop-closures from traditional meth-

ods alongside the SDF registration constraints, to maintain consistency in case of wide-baseline place revisiting.
- Application of an SDF submap-based mapping system to dynamic MAV flight in a large-scale environment.
- Release of an open-source implementation, as well as the challenging, novel dataset used for evaluation: a hexacopter equipped with visual, inertial, LiDAR and RTK GPS sensors flying through a artificial disaster site.

II. Related Work

SLAM has received considerable research attention in past decades leading to many successful approaches [1]. While sparse, feature-based representations have been widely adopted, denser representations of geometry are typically required to enable motion planning on robotic systems.

Dense SLAM systems have employed a variety of map representations, the choice of which determines many properties of the resulting system. Several successful approaches propose to represent the map as a collection of well-localized camera frames and corresponding depth maps [6]. In contrast, Whelan et. al. [7] suggests attaching a deformable mesh to a pose-graph representing the past trajectory of a camera. The same authors, in more recent work [8], dispose of trajectory estimation altogether and represent the map as a deformable collection of surfels. These systems provide a richer understanding of the occupied space, yet they do not explicitly distinguish free space from unknown space, which limits their applicability for autonomous navigation. Conversely, volumetric maps store information about both free and occupied space. Occupancy grid mapping, in particular its efficient implementation in 3D [9], has seen widespread adoption on robotic platforms [4].

An alternative volumetric representation is based on implicit representations of surface geometry through SDFs, introduced in [10]. The efficacy of these representations in fusing high-rate, noisy depth data from consumer-grade depth cameras, shown first by Newcombe et. al [11], has led to a rapid rise in their popularity. Furthermore, the advantages of this representation have led to their recent adoption on robotic platforms [12], [13], [14], where the distance field has shown additional utility for optimization-based motion planning [15], [16]. However, in order to perform fusion of high-rate data into a volumetric representation, most approaches discard raw observational data. This information loss is problematic for maintenance of a globally-consistent map. In particular, the lack of a systematic approach for correcting the map to reflect new information relating past poses, loop-closure being a prototypical example, means that SDF maps are susceptible to corruption at a global scale.

Researchers have suggested a number of approaches to mitigate this problem. BundleFusion [17] proposes to maintain a globally-consistent Truncated Signed Distance Function (TSDF)-based reconstruction by storing raw input data, globally optimizing the sensor trajectory with each arriving frame, and frequently re-integrating stored sensor data into the global SDF. This approach produces state-of-the-art results in terms of reconstruction quality, but requires significant computational resources. Several recent works [18], [19] have proposed efficiency improvements to the original scheme, principally by utilizing keyframes and computing the global reconstruction as a function of these keyframes only, as well as speed-ups to the original global optimization scheme. In contrast to our proposal however, these works suggest to maintain a single monolithic map.

Another approach is to represent the reconstructed environment as a collection of submaps. Corrections to the global map are performed by adjusting the relative submap positions. This approach avoids storage and re-integration of raw sensor data. One distinguishing factor among submap-based systems is the policy for submap generation. Several works [20], [21] suggest *partitioning* the environment into submaps with the goal of minimizing subvolume overlap. Sensor tracking proceeds by performing frame-to-model alignment on all submaps within the sensor viewing frustum. Global consistency is maintained by optimizing a graph linking submaps and past poses through observation data. The advantage of spatial *partitioning* is that the size of the map remains bounded given a scene of bounded size. However, sensor tracking in several submaps simultaneously remains an expensive operation, particularly on systems lacking a GPU.

Another approach is to generate submaps continuously, resulting in a steady stream of new submaps, each potentially overlapping with existing submaps to an arbitrary degree. The advantage of this approach is that the sensor pose, at the time of frame integration, need only to be tracked with respect to the current submap. Several works follow this approach [22], [14], [23] employing variations on the frame-to-submap based graph optimization scheme to maintain global consistency of the resulting submap collection. In a different approach, Fioraio et. al. [24] propose a map-centric system in which the past trajectory is not considered during optimization of the dense map. Instead, submaps are constrained directly to one another through geometric alignment. We extend this work by proposing to use correspondence-free alignment based on the ESDF, and propose a weighted subsampling scheme to achieve real-time performance on computationally limited platforms. Furthermore, we include loop closure constraints to maintain consistency in the presence of wide baseline loops and show the suitability of our approach for MAV based mapping, using both RGB-D and LiDAR setups.

III. Preliminaries

We parameterize the transformation between coordinate frames A and B as $T_{BA} \in \mathrm{SE}(3)$, where $T_{BA} : \mathbb{R}^3 \to \mathbb{R}^3$ maps points in frame A to frame B such that $\boldsymbol{p}_B = T_{BA}\boldsymbol{p}_A$. Transformations are concatenated as $T_{AC} = T_{AB}T_{BC}$. We express the logarithmic map as $\log : \mathrm{SE}(3) \to \mathrm{se}(3)$, which maps rotational elements to the corresponding *lie algebra* around the identity element.

We assume that the odometry front-end provides estimates in which the direction of gravity is fully observable. This allows us to align several frames in our framework with the gravity vector. Therefore several transformations live in $\mathbb{R}^3 \times \mathrm{SO}(2)$ rather than $\mathrm{SE}(3)$. We will note where this is the case.

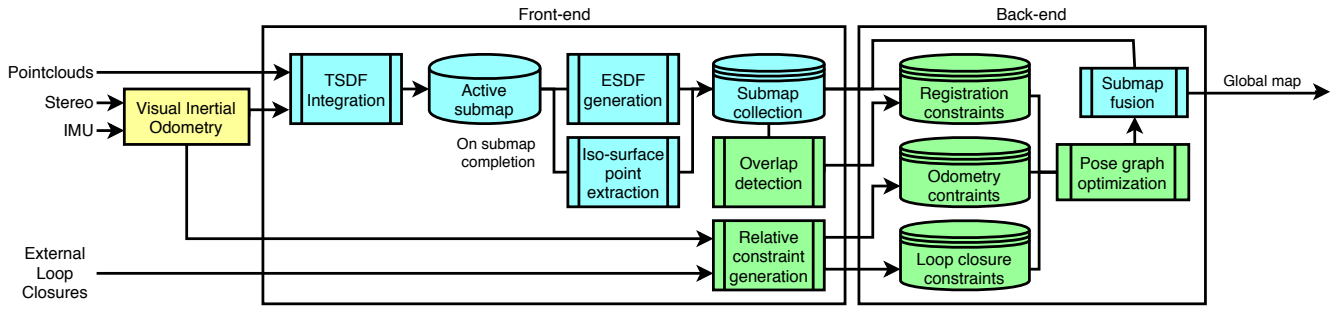

Fig. 2: Overview of the proposed system, which generates consistent volumetric maps based on pointclouds, odometry input and optional loop closures. It is split into a front-end and a back-end, which are discussed in Sec. VI and Sec. VII, respectively.

IV. Problem Statement

Given a sequence of pointclouds produced by a sensor (frame C) moving through the world (frame W, gravity aligned) we aim to build a consistent, SDF-based reconstruction. At time step i we have an estimate of the sensor frame's position from odometry, with respect to a local frame O, parameterized as a rigid transformation $T_{OC^i} \in \mathrm{SE}(3)$. If T_{OC^i} was precisely known at the time of pointcloud capture, existing techniques [11], [25] could be used to build a monolithic TSDF map in O. For practical odometry sources, however, T_{OC^i} accumulates error over time. Following reasoning in our previous work [26], we build a series of submaps $\{\mathcal{S}_i\}_{i=1}^N$ with attached frames $\{S^i\}_{i=1}^N$ from contiguous sequences of sensor data. Submaps are defined by their pose in the world $T_{WS^i} \in \mathbb{R}(3) \times \mathrm{SO}(2)$ and an SDF. The SDF of a submap $\mathcal{S}_i$ is composed of distance and weight functions $\Phi_{\mathcal{S}_i} : \mathbb{R}^3 \to \mathbb{R}$ and $\omega_{\mathcal{S}_i} : \mathbb{R}^3 \to \mathbb{R}$, which respectively map points in $\mathbb{R}^3$ to d, a signed distance to the nearest observed surface, and w, a weighting/confidence measure. Under this setup, the reconstruction problem is reduced to determining the optimal submap poses $\{T_{WS^i}\}_{i=0}^N$.

V. System

Voxgraph is divided into a front-end and a back-end (see Fig. 2). The front-end (Sec. VI) converts incoming sensor measurements into submaps, as well as constraints to be passed to the back-end. The back-end (Sec. VI) maintains this set of constraints and estimates the most likely submap collection alignment.

VI. Front end

A. Submap Creation

Contiguous sequences of input pointclouds are combined into submaps by ray casting into a spatially hashed voxel grid (see our previous work [13], [26] for details). The voxel grid origin is co-located and aligned with its attached frame S, which is parameterized by its pose with respect to the world frame T_{WS}. We create new submaps at a fixed frequency (N time steps) following from the assumption that errors in odometry estimates accumulate slowly and smoothly, and therefore submaps that are created over sufficiently short timescales remain internally consistent. During construction we also store the sensor trajectory through the submap as a collection of submap-relative poses,

$$\mathcal{T}_{\mathcal{S}_i} = \{T_{S^iC^j}, \ldots, T_{S^iC^{j+N}}\}. \tag{1}$$

Finally, following submap completion, we compute additional information used by the back-end for submap registration. In particular, the submap's ESDF $\Phi_{\mathcal{S}}$ is computed from its TSDF as described in [13]. This process propagates Euclidian distances outside the truncation band used by the TSDF, allowing distance lookups further from surfaces. Furthermore, a collection of isosurface points $\mathcal{U}_{\mathcal{S}}$ are calculated through the marching cubes algorithm [27].

B. Constraint Generation

The front-end signals the back-end to add constraints to the underlying optimization problem. We implement three kinds of constraints: odometry, registration and loop closure constraints.

1) Odometry: We penalize successive submaps' deviation from their odometry-estimated relative pose. In particular for submaps $\mathcal{S}_i$ and $\mathcal{S}_{i+1}$ joined by the sensor frames $\{C_l\}_{l=k}^{k+N}$, we estimate the submaps' relative transformation through concatenation of odometry estimates as

$$\hat{T}_{S^iS^{i+1}} = T_{C^kC^{k+1}} T_{C^{k+1}C^{k+2}} \ldots T_{C^{k+N-1}C^{k+N}}. \tag{2}$$

This estimate is passed to the back-end to constrain the relevant submap frames (see Sec. VII-A).

2) Loop Closure: Our system accepts loop closures from external sources. Note that the system is agnostic to the source of loop closures; we tested, for example, DBoW2 [28] in Sec. VIII-B2. Input loop closures take the form of an estimated transformation $\hat{T}_{C^lC^k}$ linking the sensor frame C at time instances l and k (which are not contained within the same submap).

Loop closures relate the sensor position at arbitrary time instances, and therefore generally do not relate submap frames *directly*. To accommodate this we determine the two submaps, $\mathcal{S}_i$ and $\mathcal{S}_j$ which contained these time steps, and look up the submap-relative poses $T_{S^iC^l}$ and $T_{S^jC^k}$ in (1). These estimates are passed to the back-end (see Sec. VII-B).

3) Registration: Central to this paper's approach to maintaining consistency is geometric alignment of all overlapping submaps. The front-end detects pairs of overlapping submaps using Axis Aligned Bounding Boxes (AABBs) (see [29]). In

particular an Oriented Bounding Box (OBB) in the submap frame S is computed by finding the maximal dimensions of allocated voxel *blocks* [13] in each of the axes of S. This operation is performed once per submap following pointcloud integration. Then, before global optimization, we compute for each submap an AABB which bounds the OBB and is aligned with the world frame W using the current estimate of the submap pose T_{WS^i}. Registration constraints are created for all overlapping submap pairs, and then sent to the back-end (see Sec. VII-C).

VII. Back end

The back-end aligns the submap collection by minimizing the total error of all pose graph constraints. We solve the non-linear least squares minimization

$$\begin{aligned}\arg\min_{\mathcal{X}} \sum_{(i,j)\in\mathcal{R}} ||e^{i,j}_{\text{reg}}(T_{WS^i}, T_{WS^j})||^2_{\sigma_r} + \\ \sum_{(i,j)\in\mathcal{O}} ||\boldsymbol{e}^{i,j}_{\text{odom}}(T_{WS^i}, T_{WS^j})||^2_{\Sigma_O} + \\ \sum_{(i,j)\in\mathcal{L}} ||\boldsymbol{e}^{i,j}_{\text{loop}}(T_{WS^i}, T_{WS^j})||^2_{\Sigma_L}\end{aligned} \tag{3}$$

where

$$\mathcal{X} = \{T_{WS^1}, T_{WS^2}, \ldots, T_{WS^N}\} \tag{4}$$

are the submap poses, $T_{WS^i} \in \mathbb{R}^3 \times SO(2)$, and $\mathcal{O}$, $\mathcal{R}$ and $\mathcal{L}$ are sets containing submap index pairs joined by odometry, registration and loop-closure constraints respectively. The squared Mahalanobis distance for the odometry and loop closure constraints is written as $||\boldsymbol{e}||^2_{\Sigma_\mathcal{O}}$ and $||\boldsymbol{e}||^2_{\Sigma_\mathcal{L}}$, where Σ_O and Σ_L represent their covariance matrices. For the registration constraints $||e||^2_{\sigma_\mathcal{R}}$ corresponds to the squared weighted total distance, with scalar weight $\sigma_\mathcal{R}$. Note that the optimization variables T_{WS^i} live on a non-linear manifold. We therefore solve for optimal *increments* in the corresponding lie-algebra (see for example [30]). For clarity, however, we treat the optimization as directly over manifold elements in the following sections.

A. Odometry Residuals

Odometry residuals are implemented as cost-terms on the relative pose between submap base-frames, such that for general submaps $\mathcal{S}_i$ and $\mathcal{S}_j$ we have,

$$\boldsymbol{e}^{i,j}_{\text{odom}}(T_{WS^i}, T_{WS^j}) = \log(\hat{T}^{-1}_{S^iS^j} T^{-1}_{WS^i} T_{WS^j}). \tag{5}$$

In the less general case of *odometry* constraints $j = i+1$ and $\hat{T}_{S^iS^{i+1}}$ is the odometry-estimated relative pose calculated in the front-end.

B. Loop Closure Residuals

Loop closure residuals constrain the relative pose of points *within* two submaps which are not co-located with submap frames themselves. For $\mathcal{S}_i$ and $\mathcal{S}_j$ and sensor frames C^l and C^k we have,

$$\begin{aligned}\boldsymbol{e}^{i,j}_{\text{loop}}(T_{WS^i}, T_{WS^j}) = \\ \log(\hat{T}_{C^lC^k}(T_{WS^j}T_{S^jC^k})^{-1} T_{WS^i} T_{S^iC^l})\end{aligned}$$

where $T_{S^iC^l}$ and $T_{S^jC^k}$ are the poses of the sensor at time steps l and k with respect to their submap frames (see Sec. VI-B2).

C. Registration Constraints

In this section we describe our method for generating efficient pairwise registration constraints for submaps produced by the front-end. We adopt the reasoning behind the Iterative Closest Point (ICP) algorithm (that two submaps can be aligned by minimizing the Euclidean distance between points on their surfaces) and adapt it to utilize the underlying SDF representation to perform registration efficiently.

Submaps to be aligned by the back-end have a set of zero-level iso-surface points $\mathcal{U}_\mathcal{S}$, as well as an ESDF $\Phi_\mathcal{S}$, extracted in the front-end (Sec. VI). One approach to generating pairwise constraints would be to determine point-to-point correspondences between the iso-surface points of submap pairs, and minimize distance between these pairs. In this commonly used approach, the correspondence search can be the most expensive stage.

We propose to use correspondence-free alignment. In particular, for a point on the iso-surface of $\mathcal{S}_i$ we are able to determine the distance to the iso-surface of $\mathcal{S}_j$ by reading the ESDF value at that point. The registration constraint between two submaps then expresses the total squared distance difference evaluated for all reading points $\boldsymbol{p}^m_{\mathcal{S}_i}$ of submap $\mathcal{S}_i$,

$$e^{i,j}_{\text{reg}}(T_{WS^i}, T_{WS^j}) = \sum_{m=0}^{N_{\mathcal{S}_i}} r_{\mathcal{S}_i\mathcal{S}_j}(\boldsymbol{p}^m_{\mathcal{S}_i}, T_{S^jS^i})^2,$$

where $N_{\mathcal{S}_i}$ is the total number of iso-surface points $\mathcal{U}_{\mathcal{S}_i}$ of submap $\mathcal{S}_i$. The residual $r_{\mathcal{S}_i\mathcal{S}_j}$ is given by

$$\begin{aligned}r_{\mathcal{S}_i\mathcal{S}_j}(\boldsymbol{p}^m_{\mathcal{S}_i}, T_{S^jS^i}) &= \Phi_{\mathcal{S}_i}(\boldsymbol{p}^m_{\mathcal{S}_i}) - \Phi_{\mathcal{S}_j}(T_{S^jS^i}\ \boldsymbol{p}^i_{\mathcal{S}_i}) \\ &= -\Phi_{\mathcal{S}_j}(T_{S^jS^i}\ \boldsymbol{p}^m_{\mathcal{S}_i})\end{aligned}$$

where $T_{S^jS^i}$ is a function of optimization variables in $\mathcal{X}$ through

$$T_{S^jS^i} = T^{-1}_{WS^j} T_{WS^i}. \tag{6}$$

Note that $\Phi_{\mathcal{S}_i}(\boldsymbol{p}^m_{\mathcal{S}_i}) = 0$ for all points $\boldsymbol{p}^m_{\mathcal{S}_i}$, since these points lie on the zero-level surface of $\mathcal{S}_i$ by construction. Figure 3 shows a simplified example of a registration constraint.

In addition to the loss function itself, the solver also uses its first derivative. This Jacobian is obtained by differentiating the residual over the x, y, z and *yaw* coordinates of the pose of the submaps $\mathcal{S}_i$ and $\mathcal{S}_j$. For brevity, we represent the combination of these coordinates with $\boldsymbol{q}$ and state the general form of the solution, namely

$$\begin{aligned}\frac{\partial r_{\mathcal{S}_i\mathcal{S}_j}(\boldsymbol{p}^m_{\mathcal{S}_i})}{\partial \boldsymbol{q}} &= -\frac{\partial\left[\Phi_{\mathcal{S}_j}(T_{S^jS^i}\ \boldsymbol{p}^m_{\mathcal{S}_i})\ \right]}{\partial \boldsymbol{q}} \\ &= -\left.\frac{\partial \Phi_{\mathcal{S}_j}(\boldsymbol{s})}{\partial \boldsymbol{s}}\right|_{\boldsymbol{s}=T_{S^jS^i}\ \boldsymbol{p}^m_{\mathcal{S}_i}} \frac{\partial\left(T_{S^jS^i}\ \boldsymbol{p}^m_{\mathcal{S}_i}\right)}{\partial \boldsymbol{q}}.\end{aligned}$$

The first term corresponds to the derivative of the ESDF's trilinear interpolation function [31], given by

$$\Phi_{\mathcal{S}_j}(\boldsymbol{s}) = \boldsymbol{g}^T \boldsymbol{B}^T \boldsymbol{h}(\boldsymbol{s})$$

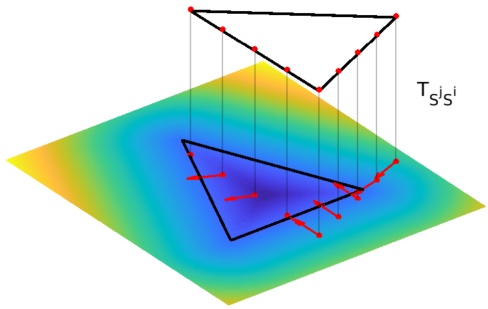

Fig. 3: Correspondence-free alignment for a simplified example. The figure shows a two dimension triangle and its corresponding ESDF (colored plane). The second triangle is transformed with respect to the original (and offset in z for clarity). Red iso-surface points are transformed into the ESDF using $T_{S^jS^i}$ and generate a cost equal to its value. The gradients of the evaluated cost are shown as red arrows.

where vector $\boldsymbol{g}$ holds the distances of the 8 voxels surrounding point $\boldsymbol{p}$, matrix $\boldsymbol{B}$ represents a constant 8 by 8 binary matrix and vector function $\boldsymbol{h}$ corresponds to

$$\boldsymbol{h} = [1 \;\; \Delta x \;\; \Delta y \;\; \Delta z \;\; \Delta x\Delta y \;\; \Delta y\Delta z \;\; \Delta z\Delta x \;\; \Delta x\Delta y\Delta z]^T$$

where Δx, Δy and Δz are the relative distances from $\boldsymbol{s}$ to its lower left voxel neighbour $\boldsymbol{v}$. For example, $\Delta x = (\boldsymbol{s}_x - \boldsymbol{v}_x)/r$ with r corresponding to the voxel width.

The first term, $\partial\Phi_{\mathcal{S}_j}(\boldsymbol{s})/\partial\boldsymbol{s}$, can thus conveniently be computed as a byproduct of the interpolation operations used to get the distance itself. The second term, $\partial\left(T_{S^jS^i}\;\boldsymbol{p}^m_{\mathcal{S}_i}\right)/\partial\boldsymbol{q}$, describes how point $\boldsymbol{p}^m_{\mathcal{S}_i}$ moves through submap $\mathcal{S}_j$ as the relative transformation $T_{S^jS^i}$ gets updated, and is well known.

D. Registration constraint sub-sampling

Given the typically large number of points on iso-surfaces within each submap, joint minimization of the registration error for all surface points of all pairs of overlapping submaps is computationally expensive, even for modestly sized maps. In this section, we describe our proposal for increasing the computation efficiency of pairwise registration constraints through sub-sampling.

We propose to approximate the registration constraints by using only a random subset of their residuals within each solver iteration. The resulting nonlinear optimization shares some similarities with Stochastic Gradient Descent (SGD), which has seen successful application to SLAM problems in the past [32]. For each iso-surface point $\boldsymbol{p}^m_{\mathcal{S}_i}$ in $\mathcal{U}_{\mathcal{S}_i}$ we determine a weight $\omega_{\mathcal{S}_i}(\boldsymbol{p}^m_{\mathcal{S}_i})$, by interpolating the weights of the voxels that surround it. The voxel weights provide a confidence measure on the distance estimated by each voxel. In a given solver iteration k, the registration error for a submap pair $\mathcal{S}_i, \mathcal{S}_j$ can be approximated by evaluating the registration error over the iso-surface point sub-set $\mathcal{V}_k$. We obtain the subset $\mathcal{V}_k$ by drawing $N_{\mathcal{V}_k}$ samples from $\mathcal{U}_{\mathcal{S}_i}$ with replacement, where the probability of drawing each point $v^m_{\mathcal{S}_i}$ is proportional to its weight $w = \omega_{\mathcal{S}_i}(v^m_{\mathcal{S}_i})$ (see Algo. 1).

Algorithm 1: Approximates the registration cost between two submaps using weighted sampling.

Input: Submap $\mathcal{S}_i$'s iso-surface point set $\mathcal{U}_{\mathcal{S}_i}$,
their weights $\omega_{\mathcal{S}_i}$ and the number of points $N_{\mathcal{S}_i}$
Submap $\mathcal{S}_j$'s SDF $\Phi_{\mathcal{S}_j}$
Sampling ratio α
Relative transformation $T_{S^jS^i}$
Output: Approximation of $e^{i,j}_{\text{reg}}(T_{WS^i}, T_{WS^j})$

```
function REGISTRATIONERRORTERM
    e^ij ← 0
    for m ← 1 to αN_{S_i} do
        draw v^m from U_{S_i} with probability ∝ ω_{S_i}(v^m)
        e^ij ← e^ij + Φ_{S_j}(T_{S^jS^i} v^m)^2
    end for
    e^ij ← (1/α) e^ij
    return e^ij
end function
```

VIII. Experiments

In this section we aim to validate our claim that the proposed system efficiently generates globally consistent SDF maps. To evaluate reconstruction performance we present simulation experiments (Sec. VIII-A) where access to ground truth geometry is readily available. In Sec. VIII-B we present field experiments conducted on an MAV flying through a large outdoor scene and evaluate trajectory accuracy as a proxy for reconstruction performance. Lastly we show qualitative results from an RGB-D equipped MAV mapping an industrial space.

A. Simulation Experiments

In this section we present a quantitative analysis of the proposed constraint sub-sampling strategy. We simulated a LiDAR-equipped MAV flying a 242 s trajectory around a multi-story building using Rotors simulator[2]. Drift is simulated by perturbing ground truth odometry with additive, biased Gaussian noise, which produces position and yaw errors which accumulate over time. The final error in the simulated odometry output, used as input to *voxgraph*, is 9.88 m.

We measure system performance in terms of reconstruction error. In particular, we fuse all submaps into a single map in the Global frame G. We then compute a ground truth SDF Ω_{GT} from the simulation environment using an open-source tool developed for this purpose[3]. As the ground-plane is a large, geometrically simple object that is well reconstructed by all systems, it tends to dominate the average error, pushing it towards a low constant value. We therefore exclude it from the evaluation to highlight trends more effectively. We calculate the Root Mean Squared Error (RMSE) error,

$$\mathcal{E}_p = \left(\sum_{\boldsymbol{p}_G \in \mathcal{O}} \left[\Phi_G(\boldsymbol{p}_G) - \Phi_{GT}(\boldsymbol{p}_G)\right]^2\right)^{1/2}, \qquad (7)$$

where $\Phi_G(\boldsymbol{p}_G)$ and $\Phi_{GT}(\boldsymbol{p}_G)$ are the SDFs of the estimated and ground truth global maps evaluated at $\boldsymbol{p}_G$. The set of all observed voxels in fused map is denoted $\mathcal{O}$.

We perform 120 experiments in which we vary the sub-sampling ratio between 100% and 0.01% and compare the

[2]https://github.com/ethz-asl/rotors_simulator
[3]https://github.com/ethz-asl/voxblox_ground_truth

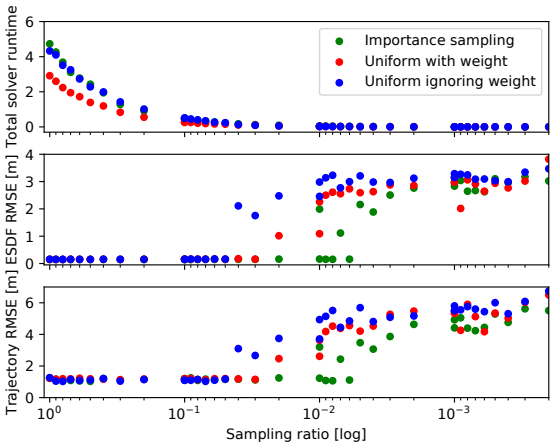

Fig. 4: Solver runtime (normalized over the dataset duration), RMSE and position errors for various constraint sub-sampling strategies and ratios resulting from a simulation study (Sec. VIII-A). The figure shows that, in this experiment, there is no observable performance degradation until constraints are sub-sampled at below 5%.

weighted sampling strategy introduced in Sec. VII-D to two alternative methods. In the first, we uniformly subsample the points $\mathcal{U}_S$ but use their weights in their cost residuals. In the second alternative strategy, we perform uniform subsampling and discard the weights altogether. Figure 4 shows the normalized runtime, reconstruction, and position RMSE for these experiments. As one might expect, pose graph optimization run-time decreases linearly with the sampling ratio. Between sub-sampling ratios of 100% and $\sim 5\%$ there is no-observable change in the reconstruction or position errors. This validates that constraint sub-sampling can be performed to increase computational efficiency without incurring significant costs in terms of performance. Furthermore, it shows that weighted sampling yields more reliable results when high downsampling ratios are used. For the remainder of this paper we therefore use weighted sub-sampling and a sampling ratio of 5%, which is chosen as a conservative trade-off between accuracy and run-time.

B. Field Experiments

We present results collected on-board a hexacopter MAV (shown in Fig. 5). All calculations are performed using an Intel NUC Core i7-8650U processor, which is carried by the platform. To demonstrate the flexibility of the proposed system we show results for both LiDAR and RGB-D-based sensor suites. LiDAR data is produced by an Ouster OS1 LiDAR. RGB-D data is produced by an Intel RealSense D415 Depth Camera. In all experiments a time-synchronized camera-imu setup is used with the visual-inertial odometry pipeline of Blösch et. al. [33] to provide odometry to the proposed system.

1) Lidar-based Mapping: In this section the MAV flew through a disaster area designed for training search and rescue personnel at Wangen an der Aare, Switzerland. Figure 1 shows an example reconstruction produced by *voxgraph* on one of these datasets. We perform a quantitative evaluation of the proposed system performance. As gathering ground truth geometry of such a large space is difficult, we attached an RTK-GNSS system to collect accurate positioning information during the MAV's flight. We use this as ground truth

Fig. 5: A hexacopter MAV flying through a search and rescue training area (Sec. VIII-B1). The platform was equipped with a 64 beam lidar (visible), monocular camera, time-synchronized IMU, and RTK-GNSS (used for trajectory evaluation only).

against which we evaluate estimated trajectories. Note that the proposed system is map-centeric, in the sense the we do not compute an optimal trajectory. To obtain a trajectory for evaluation, we take the history of submap-relative odometry measurements (1) and project them into the world frame using the optimal submap poses from (3).

We compare the proposed system against state-of-the-art frameworks for trajectory estimation. In particular, we compare against LOAM [34] (LiDAR) and VINS-MONO [5] (visual-inertial). We also compare against ROVIO [33] (visual-inertial odometry), which provides odometry estimates to *voxgraph*. This is included to show the performance of the proposed system *without* submap registration.

We evaluate the proposed and comparison frameworks using Absolute Trajectory Error (ATE) [35]. Prior to evaluation, we globally align the visual-inertial systems over 4 Degrees of Freedom (DOF) and LOAM over 6 DOF (necessary because the estimated trajecory is not gravity aligned). The estimates produced by the proposed system, as well as the ground truth trajectories, are shown in Fig. 7. We compute the RMSE between corresponding samples on the estimated and ground truth trajectories. Both the proposed and comparison systems have random elements. We therefore run each trajectory 10 times per dataset and report average figures for both the RMSE position error, as well as the CPU usage. Results of these evaluation are shown in Table I. The results show that the proposed system substantially outperforms the odometry estimated trajectory, as well as the comparison systems, on all datasets. *Voxgraph* achieves $0.83\,\mathrm{m}$, $0.59\,\mathrm{m}$, $0.94\,\mathrm{m}$, $0.52\,\mathrm{m}$ RMSE drift on the four trajectories respectively.

The proposed system runs on < 3 CPU threads on the processor carried by the MAV. Note that 100% in these terms corresponds to one (hyper) thread being fully utilized; the maximum computational load of the Intel NUC Core i7-8650U is therefore 800%. While the proposed system uses more CPU than the frameworks we compared to, ours is the only one to compute a volumetric map, usable for planning and collision-checking, as part of the CPU usage. If we discount the time for computing the T- and ESDF, the actual optimization time is comparable or even less than the comparison systems.

CPU usage for the proposed system is broken down into its major subroutines in Fig. 6, namely pointcloud integration,

	RMSE (m)				CPU (%)			
System	t0	t1	t2	t3	t0	t1	t2	t3
ROVIO	4.55	1.30	3.52	2.25	47	47	48	47
Voxgraph	**0.83**	**0.59**	**0.94**	**0.52**	265	230	305	286
Vins-Mono	5.51	-	1.11	-	159	-	157	-
Loam	2.64	1.29	6.43	3.55	131	129	125	127

TABLE I: RMSE for the field experiments conducted in Sec VIII-B. The proposed method *voxgraph* is compared against two state of the art systems for trajectory estimation, Vins-Mono [5] and LOAM [34]. The performance for ROVIO [33], which provides the proposed system with odometry estimates, is also reported. This result can be interpreted as the front-end-only performance of the proposed system. Note the results denoted at — indicate that the estimator diverged.

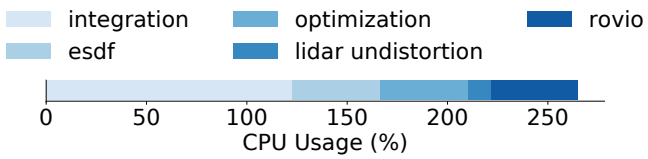

Fig. 6: A breakdown of *voxgraph* CPU usage during a typical MAV flight from Sec. VIII-B. The global optimization scheme suggested in this proposal consumes 44% of a single CPU core.

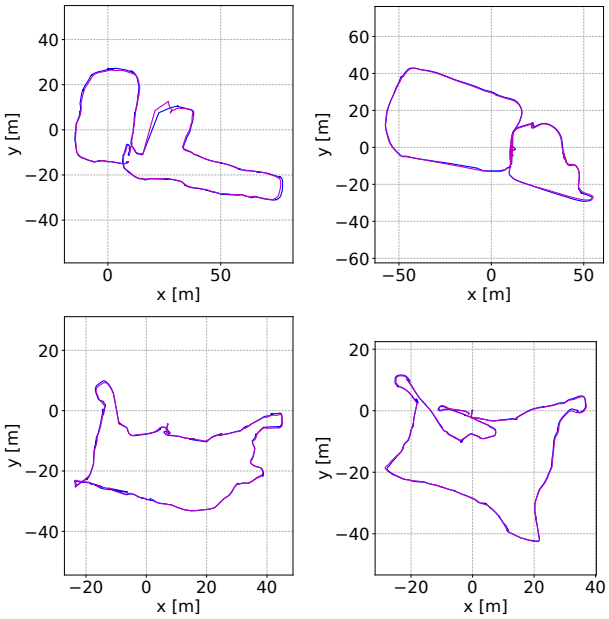

Fig. 7: The trajectories from the field experiments in Sec. VIII-B produced by an MAV flying through a large outdoor space used to train search and rescue personnel. The plots show the *voxgraph* estimated trajectory (blue), and RTK-GNSS measurements used to evaluate the system (pink).

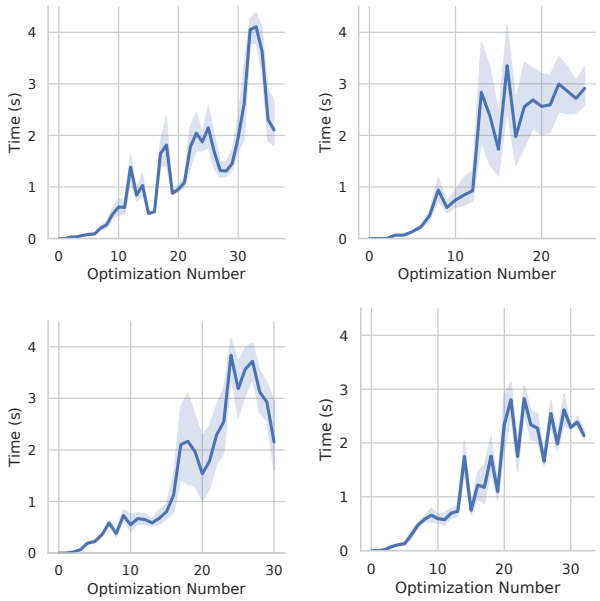

Fig. 8: The global optimization times from the field experiments in Sec. VIII-B produced by an MAV flying through a large outdoor space used to train search and rescue personnel. Each data point is generated by a single global optimization, during one of 10 mapping trials of each of 4 trajectories.

ESDF computation, pose-graph optimization, LiDAR undistortion, and ROVIO odometry. Global optimizations use 44% of a single CPU core. Figure 8 shows global optimization run-times for each of the 4 trajectories, (again 10 trials per trajectory are used to generate the results). The maximum runtime for global optimization is $\sim$4 s.

2) RGB-D-based Mapping: In order to validate how loop-closure constraints can be used in the proposed system, we applied our method to an indoor dataset [26]. The dataset is collected by an MAV, carrying an RGB-D camera and vi-sensor [36], flying through an underground industrial site at ETH Zürich. The dataset contains significant odometry drift and multiple wide-baseline loops which cannot be corrected using submap registration alone. We therefore utilize appearance-based place recognition [28] to generate loop-closure constraints from vi-sensor images. The dataset is processed on a desktop CPU with 5 cm voxels.

Figure 9 shows the results of mapping using the visual-inertial odometry system [33] alone, and *voxgraph* with and without loop closures. In the first two configurations the generated map shows significant distortion, resulting from the MAV revisiting a location with an inconsistent pose estimate. Inclusion of submap-relative loop-closure constraints (Sec. VII-B), generates corrects gross map distortions.

IX. Conclusion

In this paper we presented *voxgraph*, a novel framework for SDF-based reconstruction, aimed at producing globally consistent maps. The system represents the observed scene as a collection of overlapping SDF submaps. The system is a map-centric approach in which global consistency is maintained through geometric alignment of submaps. For each submap we compute a refined distance field, the ESDF, as well as a set of zero-level iso-surface points in the front-end. This information is used to generate efficient, correspondence-free registration constraints between submap pairs in the back-end. We propose to include only a random sub-sample of residual terms during optimization to dramatically increase computational efficiency. The efficiency of our registration constraints allow us to perform global optimization regularly, for instance after the completion of each additional submap.

The result of the proposed ideas is that the system creates consistent, large-scale maps in real-time on a lightweight computer. We show in a simulation study that constraint sub-sampling has the desired effect of boosting computational

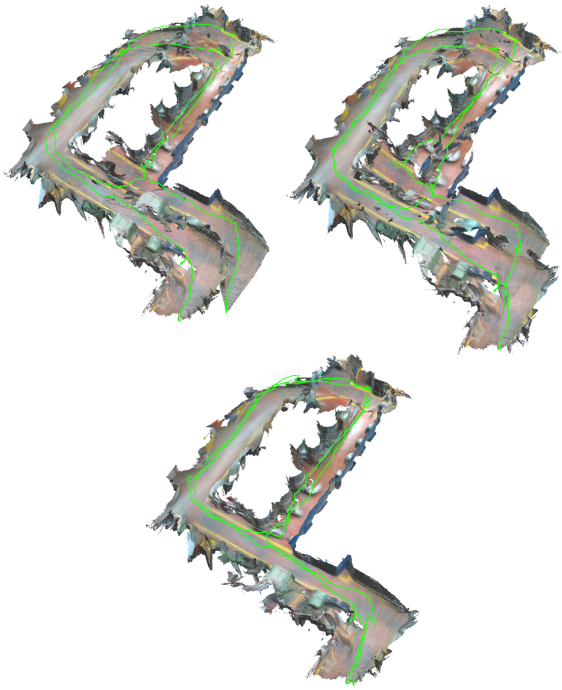

Fig. 9: Reconstruction of an industrial environment using data collected from two flights of an MAV using the proposed approach and RGB-D camera data. Consistency is maintained in the underlying representation through alignment of SDF subvolumes, even though the input odometry drifts over time. The top-left figure is the reconstruction using visual-inertial odometry-only, the top right is with registration constraints, and the bottom figure is using registration *and* loop-closure constraints.

efficiency, without dramatically impacting reconstruction accuracy. In field experiments, we show that despite not explicitly estimating an optimal trajectory, and spending the majority of our computation on generating a global volumetric map, the proposed system outperforms a state-of-the-art visual SLAM system and a top-performing LiDAR system, even when evaluated on trajectory error alone. Furthermore, *Voxgraph* is released as open-source.

References

[1] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J. J. Leonard, "Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age," *T-RO*, vol. 32, no. 6, pp. 1309–1332, 2016.

[2] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos, "Orb-slam: a versatile and accurate monocular slam system," *T-RO*, vol. 31, no. 5, pp. 1147–1163, 2015.

[3] T. Schneider, M. T. Dymczyk, M. Fehr, K. Egger, S. Lynen, I. Gilitschenski, and R. Siegwart, "maplab: An open framework for research in visual-inertial mapping and localization," *RAL*, 2018.

[4] M. Burri, H. Oleynikova, M. W. Achtelik, and R. Siegwart, "Real-time visual-inertial mapping, re-localization and planning onboard mavs in unknown environments," in *IROS*. IEEE, 2015, pp. 1872–1878.

[5] T. Qin, P. Li, and S. Shen, "Vins-mono: A robust and versatile monocular visual-inertial state estimator," *T-RO*, vol. 34, no. 4, pp. 1004–1020, 2018.

[6] J. Engel, T. Schöps, and D. Cremers, "Lsd-slam: Large-scale direct monocular slam," in *ECCV*. Springer, 2014, pp. 834–849.

[7] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," in *AAAI 2012*, 2012.

[8] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," in *RSS*. Robotics: Science and Systems, 2015.

[9] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.

[10] B. Curless and M. Levoy, "A volumetric method for building complex models from range images," in *SIGGRAPH*. ACM Press, 1996, pp. 303–312.

[11] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison *et al.*, "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *UIST*. ACM, 2011, pp. 559–568.

[12] Y. Lin, F. Gao, T. Qin, W. Gao, T. Liu, W. Wu, Z. Yang, and S. Shen, "Autonomous aerial navigation using monocular visual-inertial fusion," *JFR*, vol. 35, no. 1, pp. 23–51, 2018.

[13] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3d euclidean signed distance fields for on-board mav planning," in *IROS*. IEEE, 2017, pp. 1366–1373.

[14] R. Wagner, U. Frese, and B. Bäuml, "Graph slam with signed distance function maps on a humanoid robot," in *IROS*. IEEE, 2014, pp. 2691–2698.

[15] H. Oleynikova, Z. Taylor, R. Siegwart, and J. Nieto, "Safe local exploration for replanning in cluttered unknown environments for microaerial vehicles," *RAL*, vol. 3, no. 3, pp. 1474–1481, 2018.

[16] N. Ratliff, M. Zucker, J. A. Bagnell, and S. Srinivasa, "Chomp: Gradient optimization techniques for efficient motion planning," in *ICRA*. figshare, 2009.

[17] A. Dai, M. Nießner, M. Zollhöfer, S. Izadi, and C. Theobalt, "Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration," *ACM ToG*, vol. 36, no. 3, p. 24, 2017.

[18] R. Maier, R. Schaller, and D. Cremers, "Efficient online surface correction for real-time large-scale 3d reconstruction," *BMVC*, 2017.

[19] H. Lei and L. Fang, "Flashfusion: Real-time globally consistent dense 3d reconstruction using cpu computing," in *Robotics: Science and Systems*. RSS, 2018.

[20] P. Henry, D. Fox, A. Bhowmik, and R. Mongia, "Patch volumes: Segmentation-based consistent mapping with rgb-d cameras," in *3DV*. IEEE, 2013, pp. 398–405.

[21] O. Kähler, V. A. Prisacariu, and D. W. Murray, "Real-time large-scale dense 3d reconstruction with loop closure," in *ECCV*. Springer, 2016, pp. 500–516.

[22] S. Choi, Q.-Y. Zhou, and V. Koltun, "Robust reconstruction of indoor scenes," in *CVPR*, 2015, pp. 5556–5565.

[23] W. Hess, D. Kohler, H. Rapp, and D. Andor, "Real-time loop closure in 2d lidar slam," in *ICRA*. IEEE, 2016, pp. 1271–1278.

[24] N. Fioraio, J. Taylor, A. Fitzgibbon, L. Di Stefano, and S. Izadi, "Large-scale and drift-free surface reconstruction using online subvolume registration," in *CVPR*, 2015, pp. 4475–4483.

[25] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3d reconstruction at scale using voxel hashing," *ACM ToG*, vol. 32, no. 6, p. 169, 2013.

[26] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, "C-blox: A scalable and consistent tsdf-based dense mapping approach," in *IROS*, 2018.

[27] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *SIGGRAPH*, 1987.

[28] D. Gálvez-López and J. D. Tardós, "Bags of binary words for fast place recognition in image sequences," *R-RO*, vol. 28, no. 5, pp. 1188–1197, October 2012.

[29] Christer Ericson, *Real-time collision detection*, ser. Morgan Kaufmann series in interactive 3D technology. Amsterdam; Boston: Elsevier, 2005.

[30] F. Dellaert, M. Kaess *et al.*, "Factor graphs for robot perception," *Foundations and Trends® in Robotics*, vol. 6, no. 1-2, pp. 1–139, 2017.

[31] H. R. Kang, *Computational color technology*. Bellingham, Wash: SPIE Press, 2006.

[32] E. Olson, J. Leonard, and S. Teller, "Fast iterative alignment of pose graphs with poor initial estimates," in *ICRA*, 2006.

[33] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *IJRR*, vol. 36, no. 10, pp. 1053–1072, 2017.

[34] J. Zhang and S. Singh, "Low-drift and real-time lidar odometry and mapping," *Autonomous Robots*, vol. 41, no. 2, pp. 401–416, 2017.

[35] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *IROS*. IEEE, 2018, pp. 7244–7251.

[36] J. Nikolic, J. Rehder, M. Burri, P. Gohl, S. Leutenegger, P. T. Furgale, and R. Siegwart, "A synchronized visual-inertial sensor system with fpga pre-processing for accurate real-time slam," in *ICRA*. IEEE, 2014, pp. 431–437.