Autonomous Reconstruction of Unknown Indoor Scenes Guided by Time-varying Tensor Fields

KAI XU, Shenzhen University and National University of Defense Technology
LINTAO ZHENG*, National University of Defense Technology
ZIHAO YAN, Shenzhen University
GUOHANG YAN, Shenzhen University
EUGENE ZHANG, Oregon State University
MATTHIAS NIESSNER, Technical University of Munich and Stanford University
OLIVER DEUSSEN, University of Konstanz and Shenzhen SIAT
DANIEL COHEN-OR, Shenzhen University and Tel-Aviv University
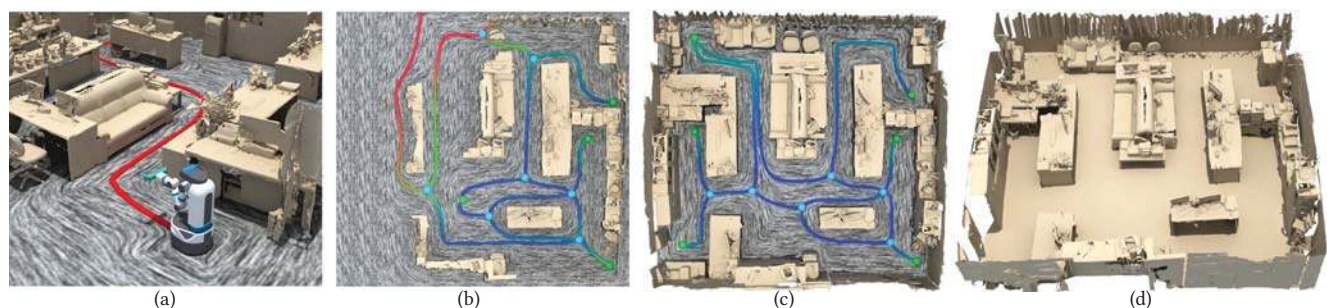HUI HUANG[†], Shenzhen University

Fig. 1. We introduce an algorithm for the autonomous reconstruction of indoor scenes, based on time-varying tensor fields (a). Given the partially scanned scene, a 2D tensor field is computed on-the-fly over the floor plane, constrained by the partial reconstruction (b). The robot is guided by the field with smooth paths, which are locally formed with field advection (red curve in (a)) and globally planned with the help of the field topology (see the curve networks in (b) and (c), with reconstruction uncertainty color-coded along the curves). The topological structure is well-defined for incomplete scenes (b), suited for guiding exploratory reconstruction of unknown scenes. The output is a full 3D reconstruction (d), at which the topology of the field reflects the scene layout (c).

Autonomous reconstruction of unknown scenes by a mobile robot inherently poses the question of balancing between exploration efficacy and reconstruction quality. We present a *navigation-by-reconstruction* approach to address this question, where moving paths of the robot are planned to account for both *global* efficiency for fast exploration and *local* smoothness to obtain high-quality scans. An RGB-D camera, attached to the robot arm, is dictated by the desired reconstruction quality as well as the movement of the robot itself. Our key idea is to harness a time-varying tensor field to guide robot movement, and then solve for 3D camera control under the constraint of the 2D robot moving path. The tensor field is updated in real time, conforming to the progressively reconstructed scene. We show that tensor fields are well suited for guiding autonomous scanning for two reasons: first, they contain sparse and controllable singularities that allow generating a *locally* smooth robot path, and second, their topological structure can be used for *globally* efficient path routing within a partially reconstructed scene. We have conducted numerous tests with a mobile robot, and demonstrate that our method leads to a smooth exploration and high-quality reconstruction of unknown indoor scenes.

*Kai Xu and Lintao Zheng are joint first authors.
†Corresponding author: Hui Huang (hhzhiyan@gmail.com)

Authors' addresses: Kai Xu, Shenzhen University, National University of Defense Technology; Lintao Zheng, National University of Defense Technology; Zihao Yan, Shenzhen University; Guohang Yan, Shenzhen University; Eugene Zhang, Oregon State University; Matthias Niessner, Technical University of Munich, Stanford University; Oliver Deussen, University of Konstanz, Shenzhen SIAT; Daniel Cohen-Or, Shenzhen University, Tel-Aviv University; Hui Huang, Shenzhen University.

CCS Concepts: • **Computing methodologies** → *Shape analysis*;

Additional Key Words and Phrases: Indoor scene reconstruction, autonomous reconstruction, tensor fields, path planning, camera trajectory optimization

1 INTRODUCTION

With the recent widespread access to commodity RGB-D cameras and the significant progress achieved in real-time reconstruction

(e.g., [Chen et al. 2013; Dai et al. 2016; Newcombe et al. 2011; Nießner et al. 2013; Whelan et al. 2015]), autonomous scene scanning and dense 3D reconstruction of indoor environments by mobile robots has drawn increasing attention from robotics and graphics communities [Charrow et al. 2015; Song et al. 2015; Xu et al. 2015]. Advances in robot and acquisition technologies facilitate the exploration and reconstruction of larger and more complex scenes, for a wide spectrum of applications ranging from virtual reality, games, movies, to autonomous service robots.

While the simultaneous exploration and mapping of unknown scenes is a long-standing problem, exploring an unknown scene for *complete scanning* and *quality reconstruction* poses new challenges. Conventional exploration of an unknown scene is driven by *global* spatial information about the environment. Consequently, it is designed for quick expansion of the robot's reach and acquisition of knowledge about the scene. In contrast, high-quality scanning for scene reconstruction hinges on *local* geometric information of visible surfaces. The robot (and its attached sensor) must move meticulously to ensure complete and stable capture, as well as sufficient overlap between adjacent scans, to reduce error in scanning and reconstruction. Achieving a balance between these two aspects (quick exploration of the environment and high-quality scene reconstruction) is the main challenge for autonomous scene reconstruction.

We present a *navigation-by-reconstruction* approach to address this challenge. In our approach, robot navigation is simultaneously constrained and guided by the progressive online reconstruction, accounting for both smoothness in local movement and efficiency in global exploration. Meanwhile, the control of the attached camera is dictated not only by the desired reconstruction quality, but also by the movement of the robot. Our goal is an efficient, as-complete-as-possible scene scanning with quality reconstruction using an economical exploration path. The main technical challenge is to achieve a well-synchronized planning for robot path and camera trajectory, thus addressing the balance between local reconstruction quality and global navigation efficacy.

Our key idea is to harness 2D directional fields to guide robot movement, and then solve for 3D camera control under the constraint of 2D robot moving paths. In 2D, we compute and update a geometry-aware tensor field [Zhang et al. 2007] constrained by the currently reconstructed scene. More specifically, the 3D scene geometry (i.e., the known surfaces) is projected to the floor plane. A set of 2D *tangential* constraints along the projected boundaries is extracted and used to compute/update the tensor field. The robot path is formed by particle advection over the tangential direction field, which inherently avoids obstacles. In 3D, we compute a smooth camera trajectory along the path to maximize the coverage of unknown or uncertain regions while satisfying the kinematic constraints between the robot's base and the camera.

During online scanning and reconstruction, the tensor field is updated in real time, conforming to the incrementally reconstructed scene (Figure 1). To ensure a smooth robot path when advecting particles over the time-varying field, we propose a space-time optimization of tensor fields via imposing both spatial smoothness and temporal coherence. There are several important advantages of tensor field guided navigation. First, tensor fields are orientation-free and thus contain much fewer singularities (degenerate points), as
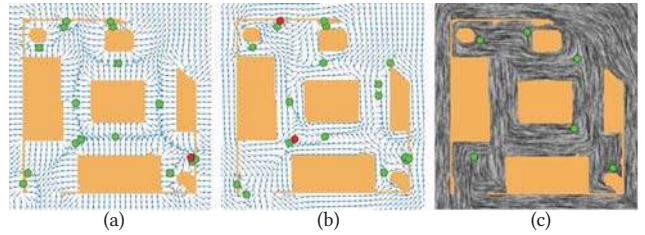

Fig. 2. A visual comparison of potential field (a), gradient field (b) and tensor field (c), computed under the same constraint of a partially observed scene. The potential field is generated with surface normals as constraint vectors, while the gradient field is constrained by surface tangents. The two vector fields suffer from crowded singularities (green and red dots), including sinks (red dots) which can trap the robot, due to orientation inconsistency. In contrast, the tensor field contains much fewer singularities and is sink-free.

compared to vector fields which are predominantly used in the literature (e.g., potential field [Khatib 1986] or gradient fields [Shade and Newman 2011]). Fewer singularities lead not only to smoother path advection, which is critical for quality reconstruction, but also to more efficient navigation due to fewer ambiguities. In addition, tensor fields are sink-free, avoiding the issue of local minima trapping. Figure 2 shows a visual comparison of these fields. Most importantly, the *topological skeleton* of a tensor field, comprised of all degenerate points and the separatrices connecting them, can be viewed as a routing graph. Using this global structure, it is possible to achieve global path planning for efficient scene scanning (Figure 1(b,c)).

In summary, our work makes the following contributions:

- Tensor field guided autonomous scanning of unknown indoor scenes, supporting both locally smooth path generation and globally efficient path routing.
- Temporally coherent tensor fields computed and updated with progressively acquired and online reconstructed scene.
- An efficient method for the path-constrained and scanning-quality-driven optimization of 3D camera trajectories.

The termination of our autonomous scanning for closed environments is clearly defined based on tensor field topology, i.e., no more unknown region can be scanned from any point on the topological skeleton. We have implemented our method on top of ROS [2014] and run it on a mobile robot Fetch [2016] holding an RGB-D camera. We demonstrate that our approach outperforms several state-of-the-art methods for a variety of indoor scenes, in terms of coverage rate, as well as reconstruction quality and efficiency.

2   RELATED WORK

Autonomous scene exploration and 3D mapping. Existing autonomous scanning systems either employ an articulated robotic arm to perform detailed scanning of a single object [Krainin et al. 2011; Kriegel et al. 2012; Wu et al. 2014], or drive a mobile robot equipped with a *fixed* camera for exploratory scene mapping [Charrow et al. 2015]. In contrast to these works, we employ an eye-in-hand setting on a mobile robot to achieve simultaneous exploration and scanning of complex scenes, which requires joint optimization of robot paths and camera trajectories.

For the purpose of 3D reconstruction, special consideration must be taken for view planning and camera movement, in order to enable

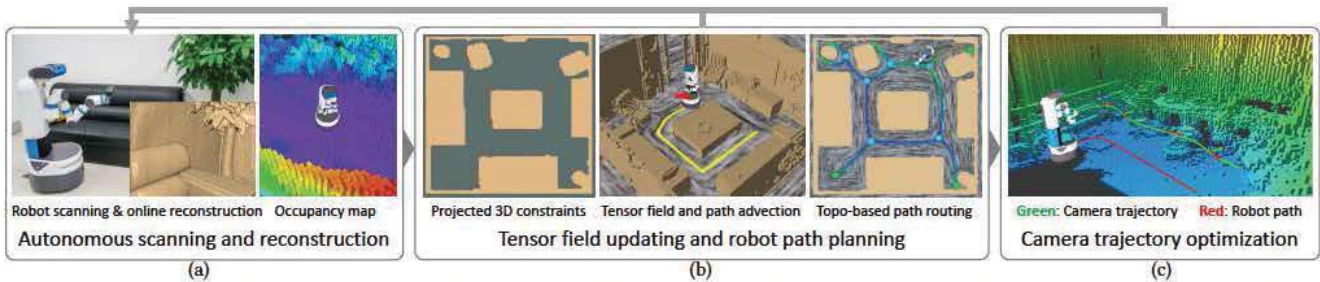| Robot scanning & online reconstruction	Occupancy map	Projected 3D constraints	Tensor field and path advection	Topo-based path routing	Green: Camera trajectory Red: Robot path
Autonomous scanning and reconstruction		Tensor field updating and robot path planning			Camera trajectory optimization
(a)		(b)			(c)

Fig. 3. An overview of our method and system. Our system runs an online scene reconstruction and employs an occupancy map for storing spatial occupancy information (a). The progressively reconstructed 3D scene geometry is projected onto the floor plane (b-left), to compute a geometry-aware time-varying tensor fields. Robot movement is locally directed by path advection over the fields (b-middle), and globally guided with path finding, based on the field topology (b-right). A smooth camera trajectory is computed along the path (c).

a robust camera localization and scan registration with Simultaneous Localization and Mapping (SLAM). Robot path planning is achieved via jointly minimizing the uncertainty of both scene mapping and camera localization [Thrun et al. 2002]. To enable robust SLAM, Ramanagopal and Ny [2016] propose a continuous view planning approach where adjacent frames are required to overlap sufficiently for the ease of frame-to-frame registration. Newcombe et al. [2011] present a technique for real-time reconstruction where sensor data is continuously received and fused into a 3D volume. Such fusion-based reconstructions, however, still suffer from drift issues due to ICP registration error, in particular in larger environments [Nießner et al. 2013]. Advanced methods, such as bundle adjustment [Agarwal et al. 2010], robust optimization [Choi et al. 2015], structure-based alignment [Zhang et al. 2014], or feature matching with deep learning [Zeng et al. 2016] are generally expensive for online use. This makes the optimization of smooth sensor trajectories especially important for autonomous online reconstruction.

Field-guided path planning. Using vector fields to guide robot navigation has been practiced for a long time [Borenstein and Koren 1989; Khatib 1986]. The most commonly employed is potential fields, which are generated by repulsive forces from known surfaces to avoid obstacles, or attractive forces to direct the robot towards a target position. Koren and Borenstein [1991] analyze the substantial shortcomings of such methods (e.g., easy trapping into a local minima or inducing unstable robot movement) and propose Vector Field Histograms (VFHs). A VFH is a polar histogram of obstacle vectors (viewing vectors from the robot to obstacles) that is locally constructed around the current robot position. Therefore, the VFH, together with its improved variants, is a statistical (instead of field-based) representation of a local environment. This representation is suitable for local obstacle avoidance, but not for global guidance.

Some works attempt to combine potential fields with a global guiding structure that is estimated from a *known* environment to attain global path planning [Ok et al. 2013]. Other types of fields such as gradient fields of harmonic scalar fields have also been studied [Shade and Newman 2011]. Information-theoretic approaches have been widely adopted for action selection during sensing, and achieve the state-of-the-art performance in autonomous exploration [Bai et al. 2016; Charrow et al. 2015]. Most of these works, however, are not designed for smooth path generation and thus the generated robot paths are not suitable for high quality

scanning. The work of Vallvé and Andrade-Cetto [2015] is the most related to ours. It adopts an information-theoretic method to estimate guidance constraints, which are used to compute a potential field for path planning. Our tensor-based direction field is computed with similar guiding constraints, but produces smoother robot paths.

View selection and trajectory optimization. The proper selection of view directions for a sensor is a fundamental problem for autonomous scanning. The most commonly adopted approach is discrete view selection, which is also referred to as Next Best View (NBV) problem. Many NBV algorithms have been developed for active scanning and/or recognition of single objects [Krainin et al. 2011; Wu et al. 2014; Xu et al. 2016] and scenes [Low and Lastra 2006]. Relatively few works, however, have studied the problem of continuous view planning or camera trajectory optimization, which would be a more practical approach for autonomous scene scanning. Reinforcement learning has been utilized to estimate camera trajectories for scene exploration [Kollar and Roy 2008], but not for quality 3D reconstruction. For scanning multiple objects, Fan et al. [2016] select the best views for each object and then optimize the entire scanning trajectory by solving a Traveling Salesman Problem, with a special scanning setup. It is, however, unclear how this method could be extended to a mobile robot setting which requires collision avoidance.

Directional fields and tensor fields. Directional fields, including those which are orientation-dependent or orientation-free, have many applications in computer graphics [Vaxman et al. 2016], ranging from surface parameterization and remeshing [Ray et al. 2009], nonphotorealistic rendering [Hertzmann and Zorin 2000] to street and urban modeling [Chen et al. 2008]. We are particularly interested in *symmetric tensor fields* (also known as 2-direction fields or line fields). A tensor field assigns each point in the problem domain an orientation-free tensor. A useful feature of such 2-direction fields is that they inherently minimize the number of singularities due to orientation ambiguities [Zhang et al. 2007]. To our knowledge, our work is the first that introduces tensor fields to guide autonomous scene scanning, and extends them to leverage temporal coherence for smooth path planning in a time-varying setting. The latter has only been done with vector fields [Chen et al. 2012]. Topological control of directional fields has been extensively studied for interactive field design [Chen et al. 2008; Zhang et al. 2006]. Compared to those works, we work in a different problem setting where the
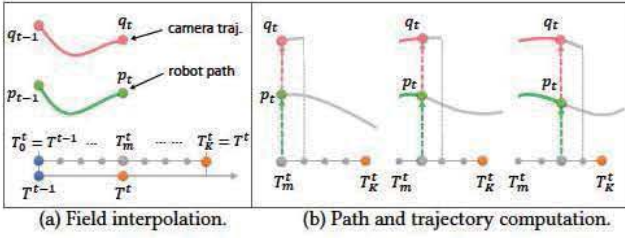
(a) Field interpolation. (b) Path and trajectory computation.

Fig. 4. Field interpolation and path / trajecotry computation over time. For a smooth transition from T^{t-1} to T^t, we interpolate them with K sub-steps (small grey dots in (a)), and start to evolve the field from the middle frame T_m^t. At each sub-step, a smooth robot path (lower grey curve in (b)) is generated by advecting a particle over the current field, along which a camera trajectory (upper grey curve) is computed based on the robot path.

geometric constraints vary constantly over time with progressive online reconstruction.

3 OVERVIEW

System overview. Our autonomous reconstruction system is composed of a mobile robot that explores an indoor room with an RGB-D camera mounted on top of it or held in its hand. We work with indoor scenes that can be regarded as a *flat layout* of walls and furniture, without staircases or sunken regions. We maintain and update two volumetric representations for the scene geometry, an occupancy grid (OG) for spatial exploration and a truncated signed distance field (TSDF) for surface reconstruction. In particular, we run Oc-toMap [Hornung et al. 2013] for storing complex spatial occupancy information, and use VoxelHashing [Nießner et al. 2013] for real-time reconstruction (Figure 3(a)). Both data structures are updated in parallel, and work in a common global reference frame.

Algorithm pipeline. Our algorithm interleaves three major steps over time: geometry-aware tensor field update, field guided path planning, and path-constrained camera trajectory computing. Figure 4 shows the computation and interaction of the various components over time. At each time step, a key-frame tensor field, denoted by T^t, is computed based on the up-to-date scene reconstruction. To ensure temporal coherence, however, one cannot switch to T^t directly. Instead, we conduct a smooth transition from the previous key-frame, T^{t-1}, to the current one, by interpolating them with K sub-steps, resulting in a temporally coherent field sequence $(T_0^t = T^{t-1}, \ldots, T_K^t = T^t)$. The field-evolution, as well as the path / trajectory computation, then starts from the middle frame T_m^t $(m = K/2)$, until $T_K^t = T^t$ is reached, at which the above process is restarted. Algorithm 1 summarizes the process.

Based on the smooth transition fields from T_m^t to T_K^t, we compute a smooth robot path (lower grey curves in Figure 4(b)) via particle advection, and optimize the camera trajectory (upper grey curves) constrained by this path. Note although the path advection can be done for an arbitrarily long time, the robot takes only the portion of a single sub-step (green part of the curve) and start a new advection for every new sub-step. Since the underlying fields transit smoothly, the paths across consecutive sub-steps are smoothly connected.

Tensor field update and guidance. Given the current reconstruction, we project the occupied 3D volume onto the floor plane and

---

Algorithm 1: Tensor field guided autonomous reconstruction

Input : Initial position of robot: p_0 (init.: $p_c \leftarrow p_0$).
Output : Reconstructed scene: S (init.: $S \leftarrow \varnothing$).

1 **repeat**
2    $T^t \leftarrow$ UpdateTensorField(S);
3    $(T_0^t, \ldots, T_K^t) \leftarrow$ GenSmoothFieldSeq(T^{t-1}, T^t);
4   **for** $s = \frac{K}{2}, \ldots, K-1$ **do**
5      $p(t)|_{t=s}^{s+1} \leftarrow$ PathAdvection(T_s^t, p_c);
6      $c(t)|_{t=s}^{s+1} \leftarrow$ TrajOptimization($p(t)|_{t=s}^{s+1}$);
7      $S \leftarrow$ ScanAndReconstruct($p(t)|_{t=s}^{s+1}, c(t)|_{t=s}^{s+1}$);
8      $p_c \leftarrow p_{s+1}$;
9    $T^{t-1} \leftarrow T^t$;
10 **until** *Stop condition is met*;

---

perform a point sampling over the projected cells lying on known surfaces. A tensor is then assigned to each sampled surface cell, with its major direction being tangential to the corresponding surface. A key-frame tensor field is computed using these 2D tensors as constraints (Figure 3(b)). Between adjacent key frames, we generate a smooth field sequence through a space-time optimization. To further increase the smoothness and simplify the topology, we introduce automatic topological control for the tensor fields. Specifically, we realize two physically meaningful control operations: cancelling of a pair of degenerate points and moving a single degenerate point. At any time step, the topological skeleton of the tensor field is used for global path routing (not reflected in Algorithm 1).

Camera trajectory optimization. Constrained by the actual robot path (reachable positions) and scanning guidance (look-at directions), a smooth camera trajectory is computed along the path (Figure 3(c)). This optimization is highly non-convex due to the estimation of base-arm kinematic constraints and scanning quality of a view. Therefore, we propose a discrete-continuous method, which first conducts a best view selection for each sample point along the path and then performs smooth curve fitting for view interpolation.

4 TENSOR FIELD GUIDED EXPLORATION

In this section, we first describe our method for computing time-varying tensor fields. We then introduce tensor field guided path planning, including both local path advection for quality reconstruction and global path finding for efficient scanning.

4.1 Time-varying tensor fields

Tensor field basics. A tensor field on a 2D domain $\mathbb{D} \subset \mathbb{R}^2$ is a smooth tensor-valued function $T : \mathbb{D} \rightarrow \mathbb{R}^{2 \times 2}$ which assigns to every point $p \in \mathbb{D}$ a second-order tensor:

$$T(p) = \begin{pmatrix} \tau_{11}(p) & \tau_{12}(p) \\ \tau_{21}(p) & \tau_{22}(p) \end{pmatrix}. \tag{1}$$

A tensor $[\tau_{ij}]$ is symmetric if and only if $\tau_{ij} = \tau_{ji}$ $(i \neq j)$. We focus only on symmetric tensors. A symmetric tensor T can be uniquely

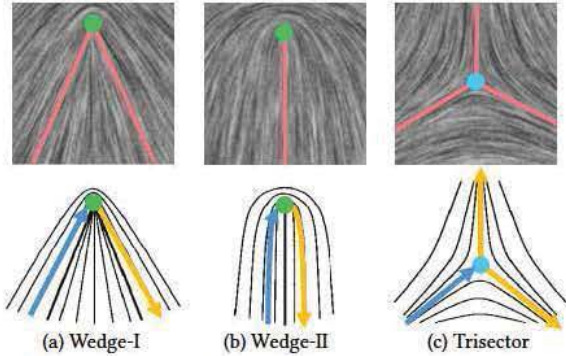**(a) Wedge-I**   **(b) Wedge-II**   **(c) Trisector**

Fig. 5. Two basic types of degenerate points, wedge (in two different forms) and trisector. The red lines in the top row are separatrix lines. The bottom row shows the discontinuous incoming (blue) and outgoing (yellow) paths passing through the degenerate points.
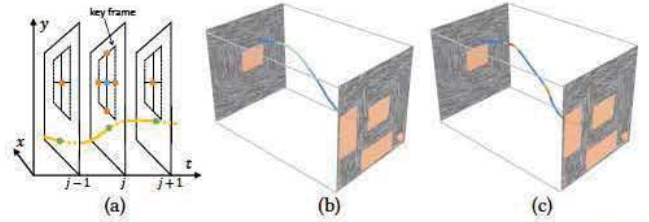

Fig. 6. (a): Illustration of time-varying field computation over space-time grid and pathline advection (yellow curve). (b-c): The pathlines generated with (b) and without (c) spatial-temporal coherence. The curvature is color-coded and plotted along the paths.

decomposed into an isotropic part S and an anisotropic part A:

$$T = S + A = \lambda \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} + \mu \begin{pmatrix} \cos 2\theta & \sin 2\theta \\ \sin 2\theta & -\cos 2\theta \end{pmatrix}, \quad (2)$$

where $\mu > 0$ and $\theta \in [0, 2\pi)$. The major and minor eigenvectors of T are perpendicular to each other. In our setting, the robot movement is directed by the *major eigenvectors* at a given point. A key feature of a symmetric tensor field is its orientation (sign) ambiguity everywhere, making it equivalent to a line field that does not distinguish between forward and backward. This avoids the problem of vector orientation in computing vector fields under geometric constraints [Xu et al. 2009].

A point p is *degenerate* if and only if $T(p) = 0$, and *regular* otherwise. Degenerate points are of great importance to our problem since they may cause discontinuous or ambiguous robot movement. A degenerate point of a tensor field is equivalent to a singularity in a vector field. However, a vector field usually contains sinks which can trap the robot and moreover, can cause oscillating movement around them due to numerical stability issue. In contrast, a tensor field contains only two types of degenerate points (Figure 5), namely *wedges* and *trisectors*, but not sinks, thus avoiding the local trapping issue. In addition, tensor fields allow flexible topological control via manipulating degenerate points [Zhang et al. 2007]; see Section 4.2. This justifies our choice of tensor field for robot guidance.

Key-frame tensor field generation. Given the scene geometry at the current time step, we compute a key-frame tensor field with the geometric constraints. To this end, we first project the grid cells corresponding to *known surfaces* in the OctoMap grid onto the floor plane. The surface cells which are higher than the robot height are not projected since they would not affect robot movement. On the floor plane, we perform farthest point sampling over the centers of the projected boundary cells, to select a set of 2D constraint points. The sampling distance d_s is heuristically set to 0.2m, measured by 2D Euclidean distance. We then define a *basis tensor field* for every constraint point, which is a regular tensor field whose *major eigenvector* aligns to the tangent of the 2D boundary at that point. The final tensor field is formed by combining the basis tensor fields of

all constraint points with the help of Gaussian radial basis function:

$$T(p) = \sum_i e^{-\|p - p_i\|^2 / \sigma^2} T_i(p), \quad (3)$$

where T_i is a basis field computed around constraint point p_i. The Gaussian band width σ can be used to control the range of influence of a basis field. Since geometric constraints are mainly used for local path guidance, they do not need to have large ranges of influence. Thus, we use a small value for the band width: $\sigma = 2.5d_s$. Due to the highly dense tangential constraints in our problem setting, we opt to compute a field for each constraint separately and then blend them with interpolation, rather than solving a densely constrained field in one shot which may lead to sub-optimal solution.

Spatial-temporal field interpolation. Given two adjacent key frames of fields, T^{t-1} and T^t, our next task is to compute K sub-step fields that vary smoothly over time. A straightforward solution is to conduct point-wise linear interpolation between source and target tensors. Such an independent point-wise interpolation, however, cannot guarantee intra-frame field smoothness. Instead we formulate a spatial-temporal interpolation over the scalar fields of τ_{11} and τ_{12} (see their definition in Equation (1)). Specifically, we minimize the bi-harmonic energy functional, using the scalar field at time $t-1$ and t as Dirichlet boundary conditions:

$$\|\Delta_{ST}\tau\|^2 + \alpha\|\tau(;0) - \tau(;t-1)\|^2 + \alpha\|\tau(;n) - \tau(;t)\|^2, \quad (4)$$

where Δ_{ST} is the spatial-temporal Laplacian defined on the space-time grid as shown in Figure 6(a). Let $\tau(v;t)$ denote the scalar value at grid vertex v at time t, and $\tau(;t)$ the scalar field at time t, where τ can be any τ_{ij} in Equation (1). $\alpha = 5 \times 10^3$ is the weight of boundary conditions. The discretization of the spatial-temporal Laplacian is:

$$\omega\tau(i;j) = \sum_{k \in N(i)} \omega_S \tau(k;j) + \omega_T \tau(i;j-1) + \omega_T \tau(i;j+1), \quad (5)$$

where $N(i)$ is the set of 1-ring neighbors of vertex i in each frame. ω_S is the intra-frame (spatial) Laplacian weight and ω_T the inter-frame (temporal) weight. $\omega = \sum_{k \in N(i)} \omega_S + 2\omega_T$ is the normalization weight. We set $\omega_S = 1$ and $\omega_T = 2$ by default.

4.2 Field-guided path planning

Path generation by particle advection. Having obtained the time-varying tensor fields $T(t)$, we compute the robot movement path as a pathline defined by a particle advected by the fields, starting from the current position. Since tensor fields do not distinguish between

forward and backward, we need to first disambiguate orientation by orienting the major eigenvectors of the tensors *within a local region* surrounding the current robot position. This is achieved by letting the major eigenvectors of the tensor field follow the front direction of the robot (the angle between these directions should be less than 90°), to minimize rotation of the robot movement. Note that this local orientation is trivial to determine, and much easier than globally orienting a vector field. After this orientation step, the tensor field $T(t)$ becomes a vector (velocity) field $V(t)$ in which particle advection can be expressed as a differential equation: $\frac{dp}{dt} = V(p;t)$. Its solution, given an initial value $(p_0;t_0)$, is

$$p(t) = p_0 + \int_0^t V(p(s);t_0 + t)\mathrm{d}s,$$

which defines the pathline starting from position p_0 and time t_0. As shown in Figure 4(b), we compute a pathline for a time period of three times sub-steps. The robot moves for only one sub-step along the path, to get to the next position and restart path advection.

Figure 6 demonstrates how the smoothness of pathlines is affected by the underlying time-varying tensor fields. In (b), we show a pathline generated by advecting over the time-varying fields between two key frames, computed with spatial-temporal coherence. As a comparison, we show in (c) the pathline formed by fields computed via point-wise temporal interpolation, with the same initial value. The color-coded curvature is plotted along the paths. In Figure 12, we show the impact of smooth robot paths on the quality of online reconstruction, justifying our choice of time-varying tensor fields.

Path splitting at degenerate points. Both types of degenerate points, wedges and trisectors, can introduce discontinuities to the generated path, due to the discontinuous tensor field at those points. Therefore, special treatment is needed at degenerate points to avoid potential problems caused by such discontinuities or ambiguities.

- *Stop and split at a wedge.* Around a wedge point, the robot path would make a sharp turn or even a U-turn (Figure 5(a,b)). To avoid shaking of the camera caused by such sudden turns, we detect the closest wedge point ahead on the robot path and reduce the movement speed of the robot until reaching the wedge. At the wedge, both robot path and camera trajectory are replanned, thus splitting the path and trajectory at the point. During the stop, the robot computes a smooth connecting trajectory, to smoothly transit between the two separate camera trajectories.
- *Stop and choose a branch at a trisector.* When the robot reaches a trisector point, it could follow either of the two branches (separatrices) ahead (Figure 5(c)). We resolve this ambiguity by choosing the better branch based on reconstruction uncertainty (see path routing below). Specifically, we choose the branch along which more unknown or uncertain regions could be explored. After determining the branch to pursue, the same path and trajectory splitting process is performed as for a wedge.

Path routing with field topology. Local path advection cannot guarantee an efficient and complete coverage of the entire scene. We need to exert global path planning, based on the *topological skeleton*

(a) Topology of partial scene.

(b) A minimum spanning tree.

(c) Topology of full scene.
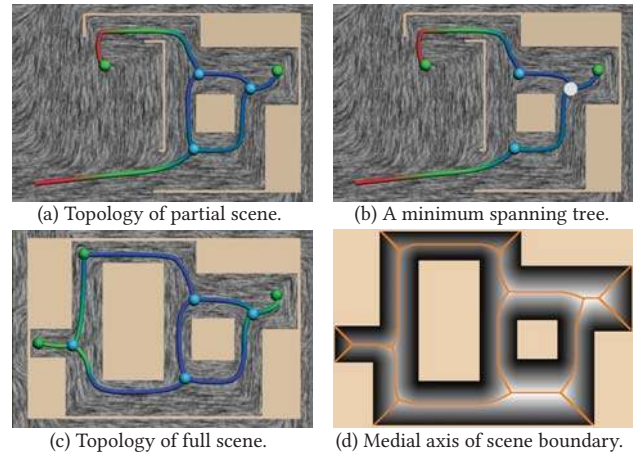

(d) Medial axis of scene boundary.

Fig. 7. The topological skeleton of tensor field can be computed for a partially scanned scene (a) and used for guiding the robot scanning. When the robot (white dot) arrives at a trisector, a minimum cost spanning tree is generated from the topological graph, to enable branch selection (b). When the reconstruction is complete, the field topology (c) conforms approximately to the medial axis of the full scene boundary (d).

of the current tensor field. Given any tensor field, its topological skeleton can be viewed as an undirected graph with all the degenerate points as graph nodes and the separatrices connecting them as edges (Figure 7(a,c)). Our global guidance takes effect when the robot reaches a trisector point, at which it needs to determine which branch to choose. We accomplish this by computing a minimum cost spanning tree from the topological graph, rooted at the trisector being visited. The cost of each edge on the graph is defined as:

$$c(e) = \left(\frac{1}{\ell(e)} \int_e I(e, p)\mathrm{d}p\right)^{-1}.$$

where $\ell(e)$ is the path length of edge e. $I(e, p)$ is the information gain of reconstruction at a point p on e. Given a point, the information gain is defined as the maximum reconstruction uncertainty (entropy) that can be reduced from among all views at that point. Given the current reconstruction, the information gain of a specific view point is measured based on how much unknown region can be observed from the view, as well as the scanning distance and slope angle of the view; see details in Appendix in supplemental material. Thus, the cost measures the reciprocal of the expected information gain along the path corresponding to edge e. Therefore, by choosing the branch leading to the minimal cost path, the robot favors shorter paths with higher information gain. Figure 8 gives an example. Without global path routing, the robot is easily trapped in the lower area and fails to scan the top area with missing data. Guided by the field topology with reconstruction entropy, the robot can immediately move to the remote area needing more scans.

A notable feature of the topology-based path routing is that it works well also for partially reconstructed scenes (Figure 7(a)). This is due to two advantageous characteristics of tensor fields. First, topological graph is well defined for any tensor fields computed with any geometric constraint. Second, tensor fields minimize the number of singularities such that no extraneous singularity appears
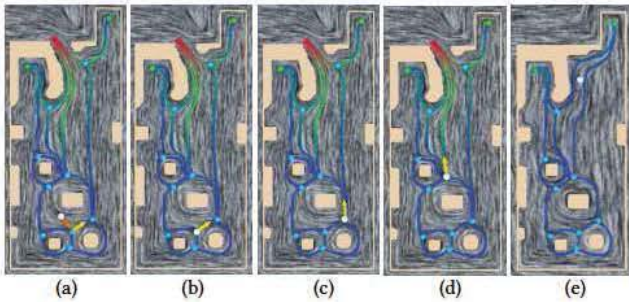
Fig. 8. The robot (white dot) is directed by the topological skeleton of the tensor field, and moves to the top area with missing data (see the separatrix lines with high reconstruction entropy in red color). In (a), the robot is first docked onto a trisector when passing through it, thereby launching branch selection for global path routing.

(a) Move a wedge towards inner corner.   (b) Move a wedge out of a dead end.

(c) Move a trisector away from obstacle.   (d) Move a trisector to avoid obstacle.

Fig. 9. Four cases for moving degenerate points.

(a) Not cancelable.   (b) Before cancelling.   (c) After cancelling.

Fig. 10. Cancelation of degenerate point pairs. The pair in (a) is not cancellable since they are detected to be topologically significant. The pair in (b) can be cancelled, with the cancellation result shown in (c).

in free space, leading to a simple and meaningful topological graph for the partial scene.

Terminating condition. Based on the topological structure of the tensor field and the definition of information gain, the termination of our autonomous scanning for closed environments can be defined as: if the expected information gain for all the accessible edges in the topological graph is below a threshold (we use 0.01 in our experiments), as in Figure 7(c), the robot stops exploring the scene. The accessibility of an edge is determined by examining the physical accessibility of sample points along that edge, based on a contact test of robot body and reconstructed scene geometry. At this time, the topological skeleton of the tensor field roughly reflects the topology (medial axis) of the closed boundary of the room (Figure 7(d)). Each point on the topological skeleton is locally well supported by the projected surfaces of the scene, in the spirit of the maximal inscribed disk definition of medial axis of 2D shapes [Blum 1967].

4.3 Automatic optimization of the guiding field

Another major benefit of tensor field is that it allows for manipulation of degenerate points, for improving field topology for either smoother path generation or simplified path routing. To achieve this, we adopt two topological operations which are originally proposed for interactive tensor field editing in [Zhang et al. 2007], i.e., moving a single degenerate point and cancelling a pair of degenerate points. The basic idea is to first convert the tensor field into a vector field: $V = \alpha(T) = \mu(\cos 2\theta, \sin 2\theta)^{\top}$, based on the definition in Equation (2). We then perform the corresponding editing operation on V to obtain V', using the method described in [Zhang et al. 2006], and then convert it back to a tensor field $T' = \alpha^{-1}(V')$. Rather than interactive editing, our problem setting demands automatic modification to be applicable to robot guidance. The automatic modifications are invoked after the computation of every key-frame. Note, however, they are optional and devised only to further improve the exploration efficiency of our method.

Movement of degenerate points. There are three cases in which we would like to move a degenerate point. *First*, for a wedge point, if the angle between its two separatrix lines is less than 90°, meaning that the wedge is formed due to a sharp corner, we move it inward to
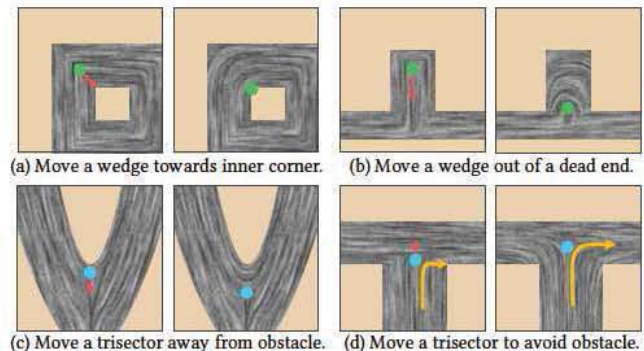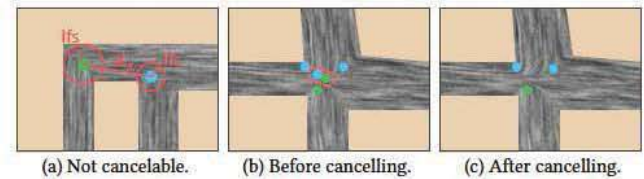
the corner. This "hides" the wedge into the corner, thus making the turn smoother; see Figure 9(a) for such an example. *Second*, a wedge point usually represents an end point of the topological skeleton of the tensor field. Given such a wedge point, if there is no more information that can be gained after a round-trip has been made around that point, the robot does not need to re-visit it with a deep U-turn. In this case, we again move that wedge point inward along its separatrix line for a distance so that the end path is eliminated; see Figure 9(b). The *third* case involves obstacle avoidance at a trisector. Since the robot would likely move towards a trisector along its separatrix line, we need to ensure that the robot does not collide into an obstacle when reaching trisector or after turning into a branch. This is done by moving the trisector away from obstacle until there is no collision detected either for the trisector point or the sample points along its separatrice (Figure 9(c,d)).

Cancellation of degenerate pairs. The pair cancellation operation allows the elimination of unwanted wedge-trisector (W-T) pairs, leading to a geometrically smoother and topologically simpler field. Unfortunately, not every W-T pair can be canceled since the wedge and/or trisector may indicate important topological features of the tensor field which might be destroyed. Therefore, the identification of cancelable degenerate pairs is critical for automatic cancellation. Edelsbrunner et al. [2002] propose a persistence-based identification process which is applicable only to scalar fields. In [Nieser et al. 2012], an intuition was made that a pair of degenerate points can be canceled only if they are close enough to each other, thus minimizing the alteration to field topology due to cancellation. Similarly, we detect cancelable pairs based on their shortest distance over the topological skeleton of the field. The remaining issue is how to set a proper threshold for this distance.

In defining the threshold, we consider the significance of a topological point on the medial axis of a shape (in parallel to a degenerate

point on topological skeleton in our case), measured by the *local feature size*, the distance from that point to the shape boundary [Dey 2006]. With the intention of preserving topologically significant degenerate pairs, we come to a heuristic condition for a W-T pair (p_S, p_T) to be cancelable:

$$d_s(p_W, y_T) < \text{lfs}(p_W) + \text{lfs}(p_T), \quad (6)$$

where $d_s(\cdot, \cdot)$ is the shortest distance between two points and $\text{lfs}(\cdot)$ is the local feature size of a point; see Figure 10(a). We found through experiments that this distance threshold is quite useful in practice, and we leave the rigorous verification and/or proof for future work. Once detected a W-T pair to be canceled, we find a small neighborhood surrounding them, within which we iteratively smooth the tensor field locally until each point inside the region becomes regular (Figure 10(c)). In practice, we perform Laplacian smoothing over the scalar fields of τ_{11} and τ_{12}.

5 CAMERA CONTROL FOR QUALITY SCANNING

Having obtained the robot path within the next time interval, we need to compute a camera trajectory to scan the scene along this path segment. Camera control has received extensive researched in the past decades. Our problem, however, is unique since the trajectory is not only driven by scanning efficiency and reconstruction quality but also constrained by the robot path.

5.1 Constrained optimization of the camera trajectory

Our goal is to compute a smooth 6DoF camera trajectory satisfying the following requirements. *First*, in order to scan the scene efficiently, camera views along the trajectory should obtain maximal coverage of unknown or uncertain regions, according to the current reconstruction. To ensure a proper reconstruction quality, however, coverage alone is not enough. A good view should look at its target regions from a close distance (within the valid scanning range), and point as orthogonally as possible to the target surfaces. *Second*, to facilitate frame-to-frame registration during online reconstruction, linear and angular speed of the camera should be as constant as possible, without exceeding a maximum threshold. *Third*, given the movement path of the robot base, every point on the trajectory must be reachable by the robot arm, according to the kinematic constraints between the base and the arm.

Summarizing the requirements above, we reach the following energy function for the camera trajectory:

$$E = \int_0^1 (-\omega_V V(q(t), \alpha(t)) + \omega_L q''(t) + \omega_A \alpha''(t)) dt, \quad (7)$$

$$\text{with} \quad q(t) \in \Psi(p(t)), q'(t) < v_m, \alpha'(t) < a_m, t \in [0, 1],$$

where $p : [0, 1] \to \mathbb{R}^3$ is a parametric curve representing the robot path, and $q : [0, 1] \to \mathbb{R}^3$ a parametric curve for the camera trajectory. For each point on the trajectory, the 6-DoF camera view includes a position $q(t)$ and a viewing angle $\alpha(t)$ (using the first view as the reference). $V(q, a)$ measures the quality of a camera view, based on the information gain of the view in reducing the uncertainty in scene reconstruction (see details in Appendix). $\Psi(p)$ is the reachable space (workspace) of the robot arm, when the robot stands at a point p (The specification of Fetch arm workspace is
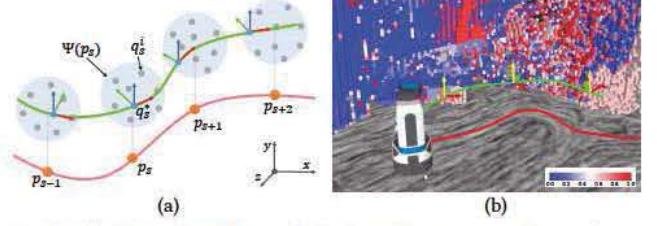

Fig. 11. (a): Illustration of our optimization of the camera trajectory (green curve) under the constraint of the robot path (red curve). (b): A real example of the trajectory optimization, where the reconstruction entropy is color-coded (blue is small and red is large) over the occupancy grid.

given in [Wise et al. 2016]). v_m and a_m are the maximum thresholds for linear and angular speed, respectively.

To solve Equation (7), our optimization samples the path p and minimizes the discretized energy:

$$
\begin{aligned}
\text{minimize} \quad E = &- \omega_V \sum_s V(q_s, a_s) \\
&+ \omega_L \sum_s (q_{s-1} - 2q_s + q_{s+1}) \\
&+ \omega_A \sum_s (a_{s-1} - 2a_s + a_{s+1}),
\end{aligned} \quad (8)
$$

$$\text{subject to} \quad q_s \in \Psi(p_s), q_{s+1} - q_{s-1} < 2v_m,$$
$$a_{s+1} - a_{s-1} < 2a_m, s = 1, \ldots, S - 1.$$

This objective function is highly non-convex due to the estimation of view quality and robot arm workspace. We therefore simplify the problem by further discretizing the feasible view space at a given base position into a discrete set of candidate views, and solve the problem with linear integer programming.

5.2 Discrete-continuous trajectory optimization

Discrete view selection. For each path sample p_s, we sample C points within its reachable space $\{q_s^i \in \Psi(p_s)\}_{i=1}^C$. For each reachable point q_s^i, we compute the best viewing angle a_s^i based on the view quality, which leads us to a set of candidate views $\{(q_s^i, a_s^i)\}$ (Figure 11(a)). Thus, the optimization in (Equation 8) can be reduced to a 0-1 integer linear programming problem, by associating each candidate with a binary decision variable $x_s^i \in \{0, 1\}$, where $x_s^i = 1$ if the i-th candidate of p_s is selected and $x_s^i = 0$ otherwise:

$$
\begin{aligned}
\text{minimize} \quad E = &- \omega_V \sum_s \sum_i V(q_s^i, a_s^i) x_s^i \\
&+ \omega_L \sum_s \sum_{i,j,k} (q_{s-1}^i - 2q_s^j + q_{s+1}^k) x_{s-1}^i x_s^j x_{s+1}^k \\
&+ \omega_A \sum_s \sum_{i,j,k} (a_{s-1}^i - 2a_s^j + a_{s+1}^k) x_{s-1}^i x_s^j x_{s+1}^k,
\end{aligned} \quad (9)
$$

$$
\begin{aligned}
\text{subject to} \quad &(q_{s+1}^i - q_{s-1}^j) x_{s+1}^i x_{s-1}^j < 2v_m, \\
&(a_{s+1}^i - a_{s-1}^j) x_{s+1}^i x_{s-1}^j < 2a_m, \\
&x_s^i \in \{0, 1\}, \quad \sum_k x_s^k = 1, \\
&s = 1, \ldots, S - 1, \quad i, j = 1, \ldots, C.
\end{aligned}
$$

We stipulate that exactly one candidate is selected at each path sample by imposing $\sum_k x_s^k = 1$. The Gurobi Solver [2016] is employed to solve the optimization efficiently.

Continuous view interpolation. The next step is to connect the selected best views $\{(q_s^*, a_s^*)\}_{s=1}^S$ to form a smooth trajectory. We use cubic Hermite splines to interpolate the positions and tangents at all views. The tangent at a view point is computed as the cross product of the look-at vector and the y-axis. We then interpolate viewing angles along the trajectory based on rotation minimizing frames [Wang et al. 2008], which minimizes the total angular speed using two adjacent selected views as boundary conditions. Figure 11(b) shows an example of computed trajectory (green curve) along a robot path (red curve), given the reconstruction entropy.

6   IMPLEMENTATION

System setup. Our Fetch robot has a built-in computer with an Intel I5-4570S CPU (2.9GHz×4) and 16GB RAM, running ROS on a Linux system. On this computer, we run OctoMap, tensor field update, path generation, and camera trajectory optimization. The computed guiding instructions (moves and poses) are directly sent to ROS to drive the robot base and arm. The robot holds a Kinect (version 1) in its single arm. The online reconstruction runs on a mobile workstation with an Intel I7-6700HQ CPU (2.6GHz×4), 16GB RAM, and an Nvidia Quadro M3000M graphics card. The mobile workstation is carried by Fetch, and the Kinect sensor is powered by the battery of Fetch, making the whole system self-contained and cable-less. Note that we do not use the pre-installed depth camera mounted on the head of Fetch.

To demonstrate the generality of our method, we also implemented and tested our system on a Turtlebot, a mobile robot mounted with a fixed (forward-looking) RGB-D camera (Kinect v1). The Turtlebot carries a laptop computer with an Intel i7-6500U CPU (2.5GHz×4) and 8GB RAM, running all computations except online reconstruction, and driving the robot with ROS. The online reconstruction runs on the same mobile workstation as that used by Fetch, except that it is remotely connected by wireless LAN rather than carried on the robot. This, again, makes the system self-contained. Please refer to the accompanying video for the two systems at work.

Online reconstruction algorithm. We employ VoxelHashing [Nießner et al. 2013] for online surface reconstruction, which uses both RGB and depth information for frame-to-frame registration (camera pose estimation). For the Fetch setup, we utilized the (filtered) inertial measurement unit (IMU) data provided with the robot arm to initialize the registration, which leads to an improved reconstruction quality. Note that, however, the registration between frames that undergo large transformations is still difficult, even assisted by IMU data. This makes the optimization of robot path and camera trajectory still essential to guarantee a high-quality surface reconstruction, as demonstrated in Figure 12.

Parameters. When computing the tensor field, we employ a 2D spatial grid with a resolution of 0.05m. The time step in computing time-varying fields is chosen to be 4 seconds, with 10 sub-steps. Thus, the duration of each sub-step is 0.4 seconds. With a moving speed of 0.3m/s, the robot moves about 1.2m per time step. Figure 15

evaluates the impact of different choices for time step and sub-steps on obstacle avoidance and path smoothness. The maximum limits for linear and angular speed are set to v_m =0.5m/s and v_m =40 degree/s, respectively. During the discrete optimization of the camera trajectory, we sample 10 candidate views for each path sample.

Handling of collisions. Our tensor fields are formed with constraints of surface tangents, making the path advection inherently obstacle avoiding. There are, however, still two cases that require special treatment. First, if the initial position of the robot is close to a wall, the robot may keep close to walls, given that there is only tangential moves. This may cause collisions due to the non-neglectable robot volume. Therefore, once the robot is detected to be too close to a known wall (< 0.3m), it moves away from the wall along its normal direction, if that is allowed by the known space. Second, due to incomplete scanning, a path may stretch into an unknown region in which an obstacle could block the path. If such a case is detected, we add a look-at view pointing towards the unknown region, to the path samples near that region (< 2m), during the camera trajectory optimization. This ensures that the robot looks into that region before rushing into it.

Collision avoidance for the robot arm can be incorporated into the candidate view selection (Section 5.2), by removing those candidates which lies in occupied cells in OctoMap. Although this does not guarantee the entire camera trajectory is collision-free, due to the discrete view sampling and data incompleteness in OctoMap, we found the solution to be sufficient in our setting since the robot rarely approaches a part of the scene too close, unlike in object-targeted scanning.

Extraction and utility of field topology. To extract the topology of a tensor field, we first detect degenerate points, by examining the tensor index of every 2D grid point, and then compute for each degenerate point the directions (angles) of its separatrices [Delmarcelle and Hesselink 1994]. Starting from a degenerate point, we trace separatrices along its separatrix directions via particle advection, which stops at another degenerate point. The numerical instability in path tracing, however, may miss connections between degenerate points. To mitigate this, we first pair the degenerate points by tracing 2D shortest distance paths between them, and then force each two paired degenerate points to be connected by a separatrix, following the method used in [Palacios and Zhang 2007].

The trisector points in the topological skeleton serve as "gateways" in global path routing, at which the robot performs branch selection. The robot, however, does not necessarily move along a separatrix so it rarely hits a trisector exactly. To allow path routing, we need a mechanism to detect whether the robot is passing by, though not exactly hitting on, a trisector, and then "dock" it to that trisector. To this end, we examine whether the robot path first approaches and then leaves away a trisector, with the minimal distance to the trisector less than a given threshold (0.5m). Once a docking trisector is detected, the robot moves to that trisector when reaching a position of minimal distance, and then launches the branch selection. Figure 7(a) shows an example of this case.

(a) Potential field path, fixed camera.

(b) Tensor field path, fixed camera.

(c) Non-smooth camera trajectory.
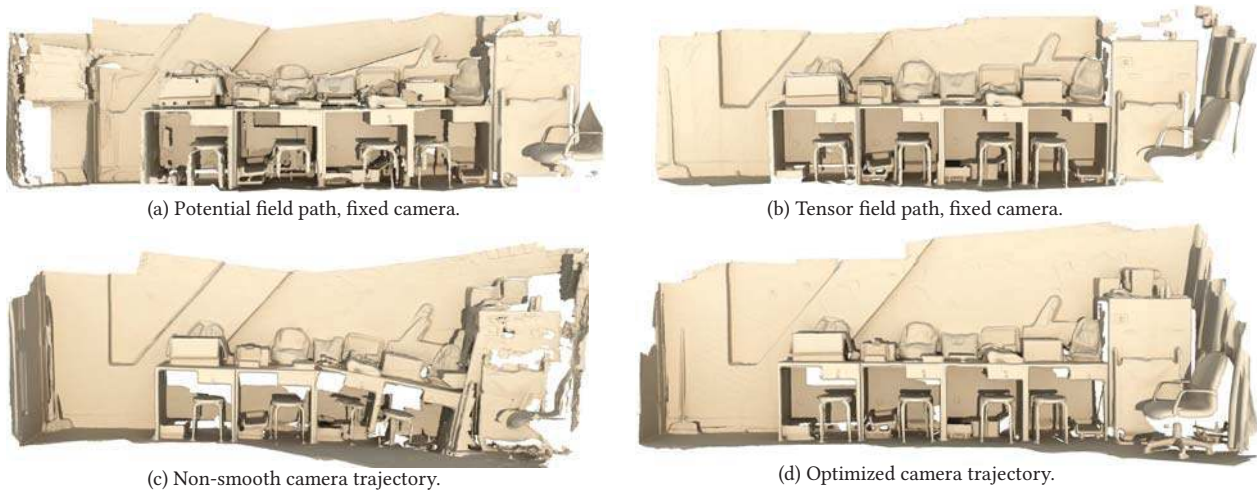

(d) Optimized camera trajectory.

Fig. 12. Comparing the reconstruction quality obtained by scanning the same scene corner along a path generated by tensor fields (a) v.s. that by potential fields (b), with a fixed camera. Given the same path (generated by tensor fields), we compare the results along camera trajectories with (c) and without (d) the smooth terms in the trajectory optimization.

7 RESULTS AND EVALUATION

Results of autonomous reconstruction. We have tested our system (Fetch robot) with seven real indoor scenes; see the results from Figure 1, 19 and 22. For all these scenes, our method achieved reasonable scene coverage with high-quality 3D reconstruction. Table 1 reports some timings and statistics on these scenes. Overall, our robot moves at an average speed of 0.3m/s and finishes rooms of tens of square meters in less than 5 minutes. Here we also report the total scanning time and travel distance, with and without topology-based global path routing. It can be observed that global planning saves significant scanning effort. Table 2 lists the average and maximum running time for the various algorithmic components in each time step. The major part of the computation time is spent on the updating of tensor fields.

Figure 12 demonstrates how smooth robot paths and camera trajectories affect the quality of online reconstruction. We visually compare the reconstruction of the same corner of a room, scanned with a fixed camera pointing to the wall, along a path segment generated by our tensor field and by potential field, respectively. Our method leads to a more accurate depth fusion, due to the smoother robot path and frame-to-frame transition. To demonstrate the effect of camera trajectory optimization, we compare the reconstruction along the same robot path (generated by our method), with and without the smoothness constraint on linear and angular speed in trajectory optimization. It can be seen that our method, with smooth robot path and camera trajectory, results in the best reconstruction quality with higher coverage. Please also see the quantitative evaluations and comparisons in the next two subsections.

7.1 Quantitative evaluation

We conducted a series of evaluations mainly concerning two aspects, reconstruction quality and scanning efficiency, with both synthetic and real scenes. Reconstruction quality is measured by the root-mean-square error (RMSE) of the ICP-based frame-to-model

Table 1. Timings and statistics for seven real scenes. The total scanning time and travel distance are compared for the cases with and without topology-based global path planning.

Scene	Area	w/ topo. planning		w/o topo. planning	
		Time	Travel	Time	Travel
Fig. 1	60 m^2	4.8 min.	58.3 m	11.8 min.	88.1 m
Fig. 19(a)	24 m^2	2.7 min.	19.7 m	4.5 min.	35.2 m
Fig. 19(b)	22 m^2	2.1 min.	13.3 m	3.4 min.	28.7 m
Fig. 22(a)	70 m^2	4.1 min.	67.2 m	6.7 min.	87.5 m
Fig. 22(b)	35 m^2	3.6 min.	40.5 m	10.2 min.	79.8 m
Fig. 22(c)	25 m^2	3.3 min.	28.2 m	9.6 min.	66.3 m
Fig. 22(d)	50 m^2	4.4 min.	52.1 m	9.7 min.	83.7 m

Table 2. Average and maximum running time (in sec.) of the four algorithmic components (tensor field update, path advection, trajectory optimization, and others including topology control etc.), within each time step.

Scene	Field		Path		Trajectory		Other	
	Avg.	Max.	Avg.	Max.	Avg.	Max.	Avg.	Max.
Synthetic	0.35	0.49	0.01	0.013	0.14	0.17	0.16	0.22
Fig. 1	0.55	1.02	0.01	0.022	0.23	0.33	0.26	0.38
Fig. 19(a)	0.39	0.75	0.01	0.013	0.15	0.19	0.16	0.27
Fig. 19(b)	0.35	0.77	0.01	0.014	0.12	0.16	0.16	0.23
Fig. 22(a)	0.49	0.95	0.01	0.019	0.21	0.29	0.23	0.34
Fig. 22(b)	0.42	0.81	0.01	0.017	0.17	0.25	0.19	0.31
Fig. 22(c)	0.37	0.75	0.01	0.015	0.13	0.21	0.15	0.23
Fig. 22(d)	0.47	0.87	0.01	0.019	0.19	0.26	0.21	0.34

registration during depth fusion [Choi et al. 2015]. For scanning efficiency, we evaluate how fast our method could achieve a full coverage of an unknown scene. Scene coverage rate is estimated against the ground-truth geometry of synthetic or real scenes (see below), based on the boundary voxels in the reconstructed occupancy map.

Synthetic evaluation is performed on Gazebo, a robot simulator provided by ROS. Gazebo supports simulation of robot dynamics and Kinect scanning. We collected 135 3D models of indoor scene
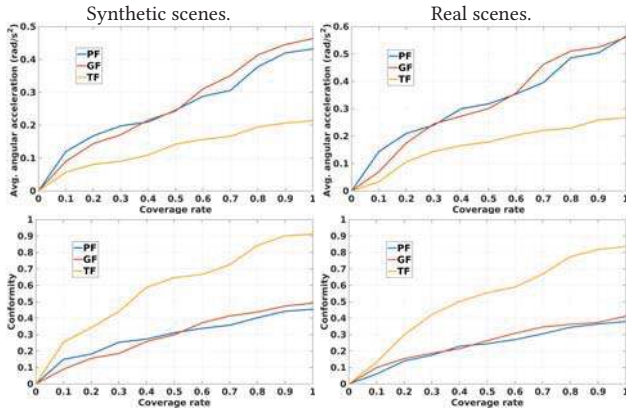
Fig. 13. Comparison of smoothness of pathlines generated by tensor fields (TF), potential fields (PF), and gradient fields (GF), measured by average (absolute) angular acceleration (top row), as well as the topological conformity of the three types of fields, measured against the medial axis of full scene boundaries (bottom row). The results were obtained and averaged over 135 synthetic (left column) and 7 real (right column) scenes.
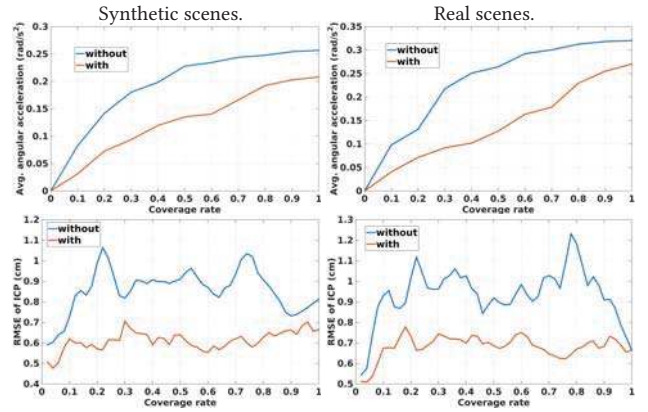

Fig. 14. Temporally coherent tensor fields lead to smoother paths (top row) and lower registration error in depth fusion (bottom row).
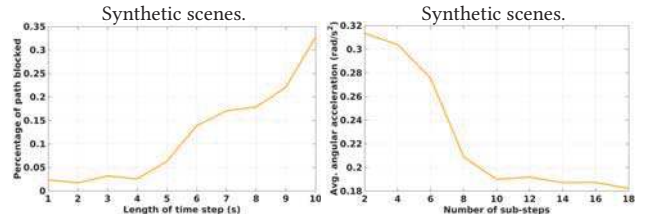

Fig. 15. The impact of different choices for time step and number of sub-steps on obstacle avoidance (percentage of advected paths being blocked) and path smoothness, respectively, both evaluated on the synthetic dataset.

and organized them into a ROS-friendly format, forming a *benchmark of autonomous virtual exploration and reconstruction* (see the supplemental material). The benchmark will be released, together with the source code of our method running on top of Gazebo.

For the seven real scenes, we build *pseudo-ground-truth reconstruction* by careful human scanning with a hand-held RGB-D camera. We used markers to facilitate offline refinement of scan registration. The markers were removed when performing robot scanning. Scene parts that are higher than the robot height (including the ceilings) are excluded from scanning in all tests, by setting the corresponding occupancy grid cells to be known (zero reconstruction entropy).

Alternative guiding fields. In order to verify the advantage of tensor fields (TF), over vector fields such as potential fields (PF) or gradient fields (GF), we conducted a number of comparisons. The main problem with vector fields is that they contain more singularities, which on a local scale sacrifices the smoothness of the generated paths. On a global scale, the topological structures induced by vector fields are usually too complex, making them not suitable for path routing. To compare with potential fields, we obtained a standard implementation based on the method in [Khatib 1986]. For gradient fields, we used the method by Shade and Newman [2011]. Default parameter settings were employed for the two methods. We conduct comparisons on the following two aspects:

- *Smoothness of generated paths.* We compare the smoothness of paths generated from the three kinds of fields during the autonomous exploration and scanning of both synthetic and real scenes. The initial positions of the robot in all three fields are the same. For all three methods, a fixed, forward-looking virtual camera is used. We plot in Figure 13(top row) the average (absolute) angular acceleration over the coverage rate of the scene. In general, paths become less smooth (with larger angular changes) as the scanning gets more complete, due to the increasing geometric constraints. Tensor fields generally

incur the least angular speed change of the generated paths, among all the three types.
- *Meaningfulness of topological structure.* For the purpose of global path routing, it is desirable that the topology of the guiding field reflects the floor structure of the scene being scanned. On both synthetic and real scenes, we compute the topological skeletons of the three types of fields, with different levels of scene completeness, and measure their conformity against the medial axis of the full scene boundary. The conformity between two skeletons is measured by the skeleton path similarity proposed in [Bai and Latecki 2008]. In general, the topological skeletons should approximate the scene topology better when scene geometry becomes more complete. Figure 13(bottom row) shows that the tensor fields always produce skeletons that best match the structural layout of the scenes.

Temporal coherence of tensor fields. We tested the smoothness of the pathlines generated by advection in tensor fields with and without temporal coherence. The plots in Figure 14(top row) show the average angular acceleration of the robot paths over coverage rate, for both cases. It can be observed that temporally coherent fields lead to much smoother robot paths. This in turn affects the reconstruction quality over coverage rate. As shown in the bottom row, the reconstruction error, measured by RMSE of ICP, is significantly lower when the guiding fields are temporally coherent.
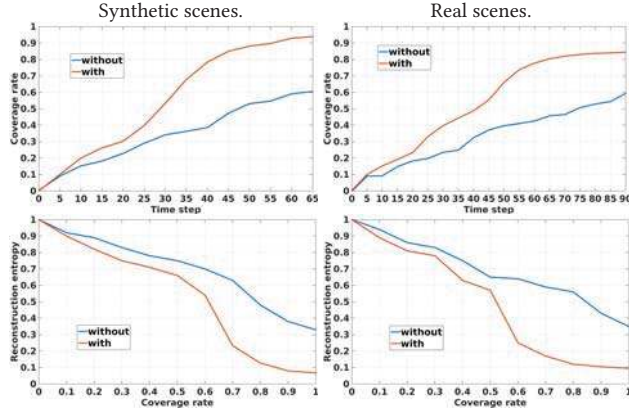
Fig. 16. Using global path planning, the scanning coverage is increased (top row) and the reconstruction entropy is reduced (bottom row) more quickly, especially in the later stages of scanning.
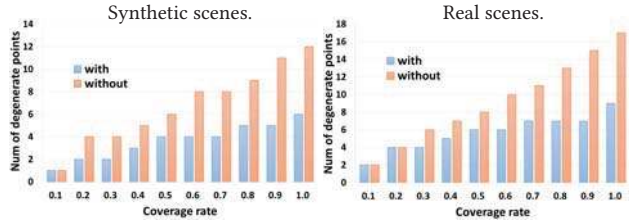

Fig. 17. Effect of topological control over field simplicity, measured by number of degenerate points.

Impact of time step during field updating. The length of time step suggests the updating frequency of the tensor fields, which is critical to obstacle avoidance. The lower the updating frequency, the more probable it is that the generated robot paths would be blocked by obstacles, because the guiding field may not reflect the so far acquired scene geometry. Meanwhile, a high updating frequency is computationally expensive. Within a time step, the number of sub-steps would affect the smoothness of tensor field interpolation: finer sub-steps produce temporally smoother fields, while demanding more computational time.

We evaluate the above effects using the synthetic dataset. To observe the impact of time step duration on obstacle avoidance, we plot in Figure 15(left) the percentage of the advected paths being blocked by obstacles, over different choices of time step. Given a robot moving speed of 0.3m/s, the maximum length of a time step which is collision-free is about 4s. In Figure 15(right), we show how the number of sub-steps affects the smoothness of the advected paths over time, with a fixed time step of 4s. The path smoothness, measured again by average angular acceleration, converges to the optimum when the number of sub-steps is 10.

Effect of global path planning. In Figure 16, we plot the coverage rate (top row) and reconstruction entropy (bottom row) during scene scanning, with and without global path planning. This experiment was conducted on the synthetic and real datasets. Coverage rate is estimated against the (pseudo) ground-truth reconstruction. As
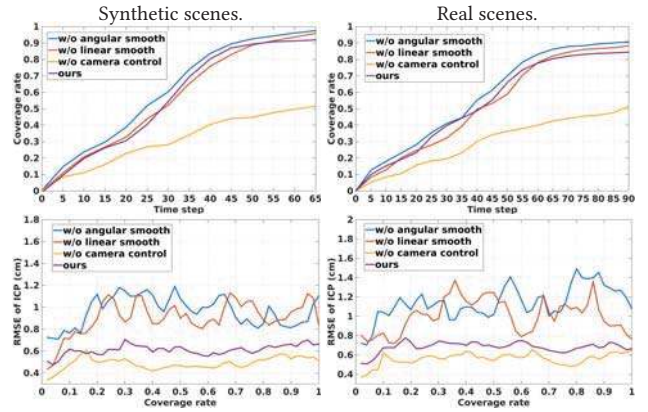

Fig. 18. Effect of optimizing the camera trajectory on coverage rate (top row) and reconstruction quality (bottom row).

shown by the plots in the top row, global planning results in a faster scene coverage. The bottom row plots the reconstruction entropy over coverage rate: when the scene coverage is low (in the early stages), the benefit of global planning is not prominent, due to the incomplete topological structure. When the scanning becomes more complete, however, global planning starts to take effect, leading to a faster convergence towards a complete and quality reconstruction. This verifies the benefits of our global path planning.

Topological control of tensor fields. We investigate the effect of the two automatic topological control mechanisms on field simplicity, both on the synthetic and real datasets. Figure 17 shows the numbers of degenerate points of tensor fields over coverage rate, with and without topological control. The results show that our topological control schemes lead to topologically simpler fields.

Optimization of the camera trajectory. In Figure 12, we show that optimized camera trajectories lead to higher reconstruction quality. Figure 18 verifies this effect quantitatively. In the top row, we show the effect of different terms in our optimization on reconstruction quality, plotted over the number of time steps. Here we compare our full method to a version without a camera control (with fixed, forward-looking camera), and that without linear or angular speed smoothing. In the bottom row, we show a comparison of scene coverage rate of the four methods above. These comparisons were conducted both on the synthetic and real datasets. The results show that our full optimization balances well between coverage efficiency and reconstruction quality.

Termination criteria. Our autonomous scene scanning terminates if no information can be gained any more from any point of the topological skeleton of the tensor field. The validity of this termination criteria can be verified by the relation between the total reconstruction entropy for the entire topological skeleton and the scene coverage rate, as shown in Figure 16 (brown curves in the bottom row). The results suggest that the convergence of the total entropy reduction (information gain) largely conforms to the maximal scene coverage, for both synthetic and real scenes. This verifies the effectiveness of our method and its termination criteria.
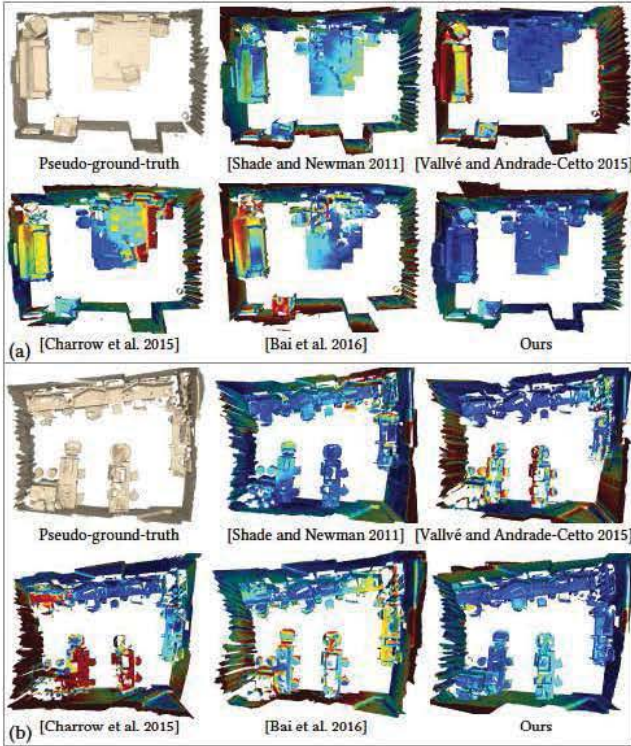
Fig. 19. Visual comparisons of reconstruction quality, depicted as color-coded RMSE plot on the reconstructed surfaces, obtained by our method and the four alternatives (red indicates large error and blue means small). Our method leads to the least amount of error.
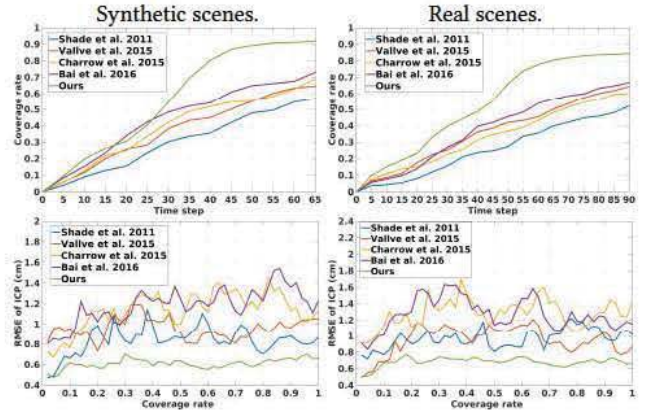

Fig. 20. Comparison of coverage rate and reconstruction quality obtained by our method and four alternative methods.

Table 3. Comparison of total scanning time and travel distance between our method and four alternatives.

Method	Synthetic scenes		Real scenes	
	Time	Travel	Time	Travel
[Shade and Newman 2011]	9.6 min.	127 m	15.2 min.	149 m
[Vallvé and Andrade-Cetto 2015]	7.3 min.	82 m	9.4 min.	97 m
[Charrow et al. 2015]	8.5 min.	104 m	13.7 min.	123 m
[Bai et al. 2016]	6.1 min.	65 m	8.8 min.	99 m
Ours	3.0 min.	46 m	4.1 min.	52 m

7.2 Comparison

We also provide a quantitative comparison between our method and four state-of-the-art methods: [Shade and Newman 2011], [Vallvé and Andrade-Cetto 2015], [Charrow et al. 2015] and [Bai et al. 2016]. The first method adopts gradient fields of harmonic guiding fields, while the second one employs potential fields. The latter three methods are all information-theory based, among which only the first is field-guided. Since the source code of the first three methods are unavailable, we use our re-implementation for the comparison, with the default parameter settings provided by the original works.

Under the same robot setting, i.e., a mobile robot mounted with a fixed and forward-looking camera, we run the four methods on the synthetic and real datasets. We evaluate their performance on coverage rate, reconstruction quality, path smoothness, as well as the reduction of reconstruction entropy. The former two results are shown in Figure 20, while the latter two can be found in the supplemental material.

In terms of the efficiency of exploration and scanning, information-theory based methods generally perform better, as reflected by the fast growth of the coverage rate of the latter three methods. Our method is inherently driven by the information gain of reconstruction during global path routing, thus achieving a comparable performance with those methods. This is also reflected in Table 3, which

compares the total scanning time and travel distance of the five methods, averaged over the tested synthetic and real scenes.

Meanwhile, our method results in a high reconstruction quality due to its special design choices made for quality scanning, i.e., smooth robot paths and camera trajectories. In Figure 19, we provide a visual comparison of reconstruction results obtained by our method and the four alternatives, on two real scenes. For each method, we show the reconstructed surfaces shaded with the color-coded RMSE, with respect to the corresponding pseudo-ground-truth reconstruction.

8 DISCUSSION AND CONCLUSIONS

We have presented a method for autonomous scanning of unknown indoor scenes, guided by time-varying and geometry-aware tensor fields. Our method attains the following key features: (i) It achieves both locally smooth path generation and globally efficient path finding, through exploiting the particle advection and topology-based guidance of tensor fields. (ii) It computes time-varying tensor fields that adapt to the time-evolving scene geometry and enhance spatial-temporal smoothness. (iii) In contrast to fixed-camera settings used by most robot navigation systems, our method pursues independent planning of robot and camera movement, thus achieving flexible exploration and scanning of indoor scenes.

Limitations. Our current solution has several limitations:

(1) Our method does not jointly optimize robot path and camera trajectory. The path planning accounts for neither the

scanning scope of the sensor nor the *inverse kinematic* constraints of the robot arm. A joint optimization with mutually-informed path and camera planning could potentially achieve a more efficient scanning and higher quality reconstruction. Such a joint planning, however, could also be extremely difficult and costly to optimize.

(2) Our navigation-by-reconstruction strategy computes smooth movement for both robot and camera. Nevertheless, one drawback of our strategy is that the robot can only perform pass-through scanning along with the robot movement rather than a detailed scanning towards a specific region or object while standing at a point. Therefore, our method cannot obtain a complete scanning coverage with a single pass navigation. These are two complementary scanning strategies, which would be useful for different scenarios.

In a scenario with high efficiency requirement, the smoothness and fluency of robot path and camera trajectory are vital for online reconstruction, since non-smooth camera movements can lead to unacceptable reconstructions due to the accumulation of drift; see Figure 12. Moreover, our solution addresses the local trapping issue, thanks to our topology-based guidance. Based on the results, we believe that our method achieves an essential and valid balance between exploration efficiency and reconstruction quality.

(3) The geometry-aware tensor fields can either be over- or under-constrained by the scene geometry, rendering them fail to guide the robot move as expected. Figure 21 shows two examples for the two cases above, respectively. The first demonstrates how an advected path passes through a narrow passage between two diagonally displaced blocks. Although the path always succeeds to pass through, no matter how narrow the passage is, it becomes too close to the wall after crossing the gap. The second shows how a path would fail to enter a "door" in a wall which is too thin, due to insufficient perpendicular constraint.

(4) In the initial stage of scanning, the reconstructed geometry may not be quite useful for computing an informative guiding field. This may cause collision if the robot moves too fast. A heuristic solution to this is to start with a small time step and gradually increase it as the scanning proceeds.

(5) When the time step is large, field interpolation could introduce new singularities. Although this is quite rare, topological smoothing [Zhang et al. 2007] could be invoked if needed. In any case, the main singularities with meaningful topology would persist from time step to time step.

(6) From a high-level perspective, our approach is purely geometric. We do not learn or infer high-level structure or semantics of the scene during scanning. Therefore, it does not offer higher level guidance to the robot such as anticipation of accessibility or safety, other than local avoidance of obstacles and greedy maximization of information gain.

Future work. We see several interesting directions to pursue and explore in the future. *First*, several technical components of our method, such as global guidance by field topology, are currently studied and realized as a proof-of-concept. We hope our work will
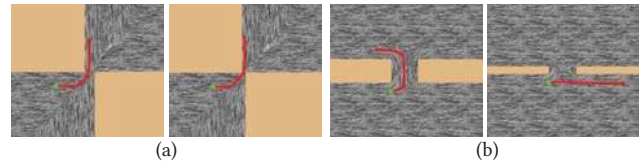

Fig. 21. Path advection in tensor fields which are either over- or under-constrained. (a): Although the field can always direct the particle through the narrow passage, the path becomes too close to the wall after crossing the gap. (b): A path would fail to enter a "door" in a wall which is too thin.

inspire more in-depth and theoretical future study on those problems. *Second*, we would like to investigate the use of 3D tensor fields for a joint planning of robot and camera movement in 3D space. The main difficulty here is that the smoothness and degenerate behavior of 3D tensor fields may not be easy to control. Moreover, an efficient GPU implementation may be needed to accommodate the highly intensive computation. *Third*, it is worth considering online analysis and learning within our framework of tensor field guidance, to achieve a more intelligent exploration and scanning in highly complex environments. *Fourth*, our navigation method could be extended to non-planar ground surfaces, such as terrains, which can be approximately regarded as 2D manifolds on which tensor fields are well defined [Zhang et al. 2007]. *Lastly*, an interesting direction would be applying 3D tensor fields for a more versatile robot guidance, with applications ranging from drone exploration and scanning to robot-hand grasping and manipulation.

ACKNOWLEDGEMENTS

REFERENCES

Sameer Agarwal, Noah Snavely, Steven M Seitz, and Richard Szeliski. 2010. Bundle adjustment in the large. In *European conference on computer vision*. Springer, 29–42.

Shi Bai, Jinkun Wang, Fanfei Chen, and Brendan Englot. 2016. Information-theoretic exploration with Bayesian optimization. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*. IEEE, 1816–1822.

Xiang Bai and Longin Jan Latecki. 2008. Path similarity skeleton graph matching. *IEEE T-PAMI* 30, 7 (2008), 1282–1292.

Harry Blum. 1967. A Transformation for Extracting New Descriptors of Shape. In *Models for the Perception of Speech and Visual Form*, Weiant Wathen-Dunn (Ed.). MIT Press, 362–380.

Johann Borenstein and Yoram Koren. 1989. Real-time obstacle avoidance for fast mobile robots. *IEEE Trans. Sys., Man, and Cyber.* 19, 5 (1989), 1179–1187.

Benjamin Charrow, Gregory Kahn, Sachin Patil, Sikang Liu, Ken Goldberg, Pieter Abbeel, Nathan Michael, and Vijay Kumar. 2015. Information-theoretic planning with trajectory optimization for dense 3D mapping. In *Proceedings of Robotics: Science and Systems*.

Guoning Chen, Gregory Esch, Peter Wonka, Pascal Müller, and Eugene Zhang. 2008. Interactive procedural street modeling. In *ACM Trans. on Graph. (SIGGRAPH)*, Vol. 27. 103.

Guoning Chen, Vivek Kwatra, Li-Yi Wei, Charles D Hansen, and Eugene Zhang. 2012. Design of 2d time-varying vector fields. *IEEE Trans. Vis. & Computer Graphics* 18, 10 (2012), 1717–1730.

Jiawen Chen, Dennis Bautembach, and Shahram Izadi. 2013. Scalable real-time volumetric surface reconstruction. *ACM Trans. on Graph. (SIGGRAPH)* 32, 4 (2013), 113.
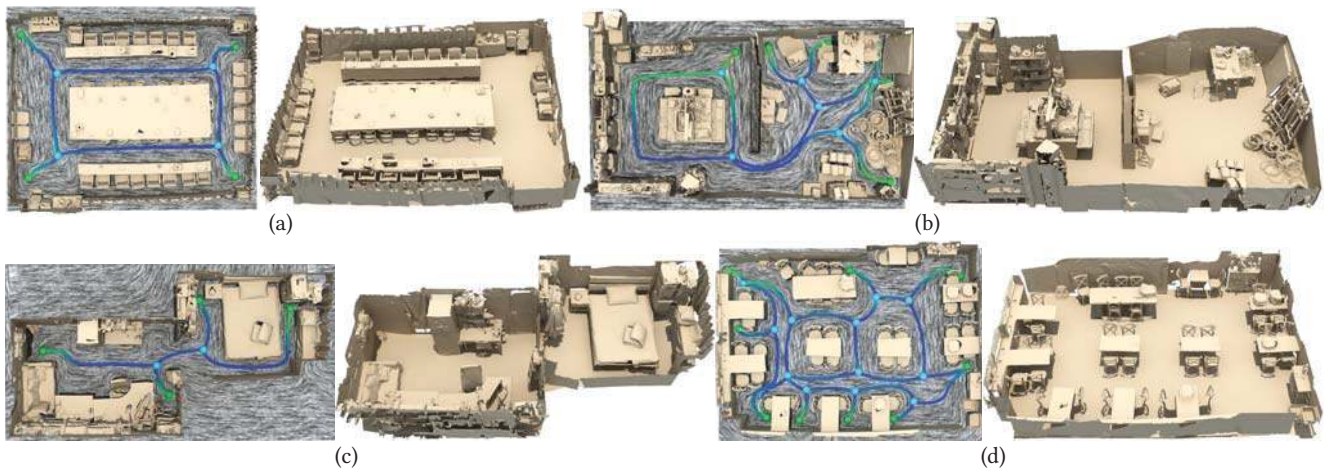
Fig. 22. Four real scenes scanned and reconstructed by our autonomous system. For each scene, we show the final field topology (left) and the reconstruction result (right). The scene in (c) is not closed, due to inaccessible narrow doors; the scanning was terminated by human.

Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. 2015. Robust reconstruction of indoor scenes. In *Proc. CVPR.* 5556–5565.

Angela Dai, Matthias Nießner, Michael Zollöfer, Shahram Izadi, and Christian Theobalt. 2016. BundleFusion: Real-time Globally Consistent 3D Reconstruction using On-the-fly Surface Re-integration. *arXiv preprint arXiv:1604.01093* (2016).

Thierry Delmarcelle and Lambertus Hesselink. 1994. The topology of symmetric, second-order tensor fields. In *Proc. IEEE Visualization.* 140–147.

Tamal K Dey. 2006. *Curve and surface reconstruction: algorithms with mathematical analysis.* Vol. 23. Cambridge University Press.

Herbert Edelsbrunner, David Letscher, and Afra Zomorodian. 2002. Topological persistence and simplification. *Discrete and Computational Geometry* 28, 4 (2002), 511–533.

Xinyi Fan, Linguang Zhang, Benedict Brown, and Szymon Rusinkiewicz. 2016. Automated View and Path Planning for Scalable Multi-Object 3D Scanning. *ACM Trans. on Graph. (SIGGRAPH Asia)* 35, 6 (2016), 239.

Fetch. 2016. Fetch Robotics. http://fetchrobotics.com/research. (2016).

Gurobi. 2016. Gurobi Solver. http://www.gurobi.com/LP. (2016).

Aaron Hertzmann and Denis Zorin. 2000. Illustrating smooth surfaces. In *Proc. of SIGGRAPH.* 517–526.

Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. 2013. OctoMap: An Efficient Probabilistic 3D Mapping Framework Based on Octrees. *Autonomous Robots* 34, 3 (2013), 189–206.

Oussama Khatib. 1986. Real-time obstacle avoidance for manipulators and mobile robots. *Int. J. Robotics Research* 5, 1 (1986), 90–98.

Thomas Kollar and Nicholas Roy. 2008. Trajectory optimization using reinforcement learning for map exploration. *Int. J. Robotics Research* 27, 2 (2008), 175–196.

Yoram Koren and Johann Borenstein. 1991. Potential field methods and their inherent limitations for mobile robot navigation. In *Proc. ICRA.* 1398–1404.

Michael Krainin, Brian Curless, and Dieter Fox. 2011. Autonomous Generation of Complete 3D Object Models Using Next Best View Manipulation Planning. In *Proc. ICRA.* 5031–5037.

Simon Kriegel, Christian Rink, Tim Bodenmüller, Alexander Narr, Michael Suppa, and Gerd Hirzinger. 2012. Next-best-scan planning for autonomous 3D modeling. In *Proc. IROS.* 2850–2856.

Kok-Lim Low and Anselmo Lastra. 2006. An adaptive hierarchical next-best-view algorithm for 3d reconstruction of indoor scenes. In *Proceedings of 14th Pacific Conference on Computer Graphics and Applications (Pacific Graphics 2006).*

Richard A Newcombe, Andrew J Davison, Shahram Izadi, Pushmeet Kohli, Otmar Hilliges, Jamie Shotton, David Molyneaux, Steve Hodges, David Kim, and Andrew Fitzgibbon. 2011. KinectFusion: Real-time dense surface mapping and tracking. In *Proc. IEEE Int. Symp. on Mixed and Augmented Reality.* 127–136.

Matthias Nieser, Jonathan Palacios, Konrad Polthier, and Eugene Zhang. 2012. Hexagonal global parameterization of arbitrary surfaces. *IEEE Transactions on Visualization and Computer Graphics* 18, 6 (2012), 865–878.

M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger. 2013. Real-time 3D Reconstruction at Scale using Voxel Hashing. *ACM Trans. on Graph. (SIGGRAPH Asia)* 32, 6 (2013), 169.

Kyel Ok, Sameer Ansari, Billy Gallagher, William Sica, Frank Dellaert, and Mike Stilman. 2013. Path planning with uncertainty: Voronoi uncertainty fields. In *Proc. ICRA.*

4596–4601.

Jonathan Palacios and Eugene Zhang. 2007. Rotational symmetry field design on surfaces. *ACM Trans. on Graph. (SIGGRAPH)* 26, 3 (2007), 55.

Manikandasriram Srinivasan Ramanagopal and Jerome Le Ny. 2016. Motion Planning Strategies for Autonomously Mapping 3D Structures. *arXiv preprint arXiv:1602.06667* (2016).

Nicolas Ray, Bruno Vallet, Laurent Alonso, and Bruno Levy. 2009. Geometry-aware direction field processing. *ACM Trans. on Graph.* 29, 1 (2009), 1.

ROS. 2014. ROS Wiki. http://wiki.ros.org/. (2014).

Robbie Shade and Paul Newman. 2011. Choosing where to go: Complete 3D exploration with stereo. In *Proc. ICRA.* 2806–2811.

S. Song, L. Zhang, and J. Xiao. 2015. Robot In a Room: Toward Perfect Object Recognition in Closed Environments. *arXiv preprint arXiv:1507.02703* (2015).

Sebastian Thrun and others. 2002. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium* 1 (2002), 1–35.

Joan Vallvé and Juan Andrade-Cetto. 2015. Potential information fields for mobile robot exploration. *Robotics and Autonomous Systems* 69 (2015), 68–79.

Amir Vaxman, Marcel Campen, Olga Diamanti, Daniele Panozzo, David Bommes, Klaus Hildebrandt, and Mirela Ben-Chen. 2016. Directional Field Synthesis, Design, and Processing. *Computer Graphics Forum* 35, 2 (2016), 545–572.

Wenping Wang, Bert Jüttler, Dayue Zheng, and Yang Liu. 2008. Computation of rotation minimizing frames. *ACM Trans. on Graph.* 27, 1 (2008), 2.

Thomas Whelan, Stefan Leutenegger, Renato F Salas-Moreno, Ben Glocker, and Andrew J Davison. 2015. ElasticFusion: Dense SLAM without a pose graph. In *Proc. Robotics: Science and Systems.*

Melonee Wise, Michael Ferguson, Derek King, Eric Diehr, and David Dymesich. 2016. Fetch and freight: Standard platforms for service robot applications. In *Workshop on Autonomous Mobile Service Robots.*

Shihao Wu, Wei Sun, Pinxin Long, Hui Huang, Daniel Cohen-Or, Minglun Gong, Oliver Deussen, and Baoquan Chen. 2014. Quality-driven poisson-guided autoscanning. *ACM Trans. on Graph. (SIGGRAPH Asia)* 33, 6 (2014), 203.

Kai Xu, Hui Huang, Yifei Shi, Hao Li, Pinxin Long, Jiannong Caichen, Wei Sun, and Baoquan Chen. 2015. Autoscanning for Coupled Scene Reconstruction and Proactive Object Analysis. *ACM Trans. on Graph.* 34, 6 (2015), 177.

Kai Xu, Yifei Shi, Lintao Zheng, Junyu Zhang, Min Liu, Hui Huang, Hao Su, Daniel Cohen-Or, and Baoquan Chen. 2016. 3D Attention-Driven Depth Acquisition for Object Identification. *ACM Transactions on Graphics (TOG)* 35, 6 (2016), 238.

Kai Xu, Hao Zhang, Daniel Cohen-Or, and Yueshan Xiong. 2009. Dynamic harmonic fields for surface processing. *Computers & Graphics* 33, 3 (2009), 391–398.

Andy Zeng, Shuran Song, Matthias Nießner, Matthew Fisher, Jianxiong Xiao, and Thomas Funkhouser. 2016. 3DMatch: Learning Local Geometric Descriptors from RGB-D Reconstructions. *arXiv preprint arXiv:1603.08182* (2016).

Eugene Zhang, James Hays, and Greg Turk. 2007. Interactive tensor field design and visualization on surfaces. *IEEE Trans. Vis. & Computer Graphics* 13, 1 (2007), 94–107.

Eugene Zhang, Konstantin Mischaikow, and Greg Turk. 2006. Vector field design on surfaces. *ACM Trans. on Graph.* 25, 4 (2006), 1294–1326.

Yizhong Zhang, Weiwei Xu, Yiying Tong, and Kun Zhou. 2014. Online Structure Analysis for Real-time Indoor Scene Reconstruction. *ACM Trans. on Graph.* 34, 5 (2014), 159.