Gaussian Fusion: Accurate 3D Reconstruction via Geometry-Guided Displacement Interpolation

Duo Chen[1,2], Zixin Tang[1], Zhenyu Xu[1], Yunan Zheng[1],Yiguang Liu[1]*
[1]Sichuan University, [2]Chongqing University of Education
duochen3@gmail.com liuyg@scu.edu.cn

Abstract

Reconstructing delicate geometric details with consumer RGB-D sensors is challenging due to sensor depth and poses uncertainties. To tackle this problem, we propose a unique geometry-guided fusion framework: 1) First, we characterize fusion correspondences with the geodesic curves derived from the mass transport problem, also known as the Monge-Kantorovich problem. Compared with the depth map back-projection methods, the geodesic curves reveal the geometric structures of the local surface. 2) Moving the points along the geodesic curves is the core of our fusion approach, guided by local geometric properties, i.e., *Gaussian curvature and mean curvature. Compared with the state-of-the-art methods, our novel geometry-guided displacement interpolation fully utilizes the meaningful geometric features of the local surface. It makes the reconstruction accuracy and completeness improved. Finally, a significant number of experimental results on real object data verify the superior performance of the proposed method. Our technique achieves the most delicate geometric details on thin objects for which the original depth map back-projection fusion scheme suffers from severe artifacts (See Fig.1).*

1. Introduction

With the increasing availability of active optical techniques, RGB-D sensors have become more practical and affordable. For instance, time of flight (ToF) based RGB-D sensors have been extensively integrated into smartphones and tablets. They are popular due to their high frame rate and excellent portability. Many seminal works leverage these features to reconstruct the indoor scenes in real-time [26, 15, 17, 35, 42]. However, the resolution and accuracy of depth maps are usually limited by cost and size. The pose estimation heavily relies on visual odometry, constrained by matching accuracy. In all, it is difficult to reconstruct the delicate geometric details through consumer RGB-D sensors.

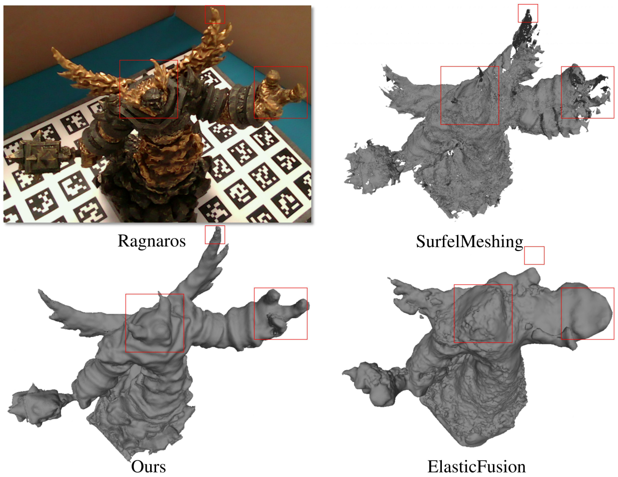

Figure 1. The state-of-the-art depth map back-projection fusion methods vs. our geometry-guided displacement interpolation approach on real object data (Ragnaros, see Tab.2 for reconstruction details). Our approach achieves the most delicate geometric details.

The fusion strategy directly decides reconstruction results. Moreover, most state-of-the-art methods employ a depth map back-projection fusion strategy, which differs by data representation. There are two popular representations to reconstruct one 3D model from observed RGB-D data [45]. One is to gather the spatial coordinates in a 3D voxel grid. *e.g.*, InfiniTAM [17], Kinect Fusion[26], and [27] average the truncated signed distance function (TSDF) [5] value if the projection on the depth map is verified. An alternative one to voxel-based representation is surfel/point-based model [30, 19, 42, 31]. Keller *et al.* [19, 42, 31] perform a weighted average between measurements and their back-projections to eliminate the poses and the measurements uncertainties. However, the depth map back-projection strategy only considers single-surfel/point information and ignores the meaningful geometric structure. Thus, it usually suffers from severe artifacts on reconstructing delicate geometric details. So that, we conclude three basic obversions: **1)** the projective correspondences are typically inaccurate due to sensor depth and poses uncertainties; **2)** those uncertainties are difficult to filter out by averaging; **3)** point to pixel projection and depth map back-projection fusion can not fully utilize the meaningful geometric structure of the local surface.

In this paper, we propose a unique fusion framework called Gaussian Fusion. In contrast to existing depth map back-projection approaches, the proposed method is based on geometry-guided displacement interpolation. To overcome the shortcomings of depth map back-projection techniques, we make the following assumptions: **1)** a sufficiently small local surface follows Gaussian distribution [2, 36]; **2)** the metrics between distributions can be characterized by Wasserstein distance; **3)** there are independent transference plans between local surfaces, which are going to be fused; **4)** the transference plans between fusion candidates should be optimal due to they are two observations of the same surface interfered by uncertainties. Thus, there are geodesic curves between measurements, as well as geodesic curves between Gaussian measures. In all, we pair measurements with the optimal transference plans by solving the mass transport problem, which is also known as the Monge-Kantorovich problem. We move the points along the geodesic curves and employ a novel advection to tackle the uncertainties.

This paper makes the following contributions:

- We introduce displacement interpolation into the 3D reconstruction area for the first time and propose a unique geometry-guided fusion framework.
- We present a flexible interpolation scheme based on geodesic curves, which reveal the geometric structures of the local surface.
- We propose a novel advection strategy guided by local geometric properties, *i.e.*, Gaussian curvature and mean curvature. A significant number of experimental results on real data verified the effectiveness of the proposed method.

2. Related work

This section discusses the most related work and highlights our novel non-back-projection fusion framework while briefly describing their characteristic.

Voxel-based methods. The main idea is based on TSDF fusion [5], which is widely used due to the highly efficient expression of reconstruction scenes. The seminal work Kinect Fusion [26] and the inspired works[27, 43, 44, 4, 6, 35, 16, 17, 20] employ a similar data fusion process, but the reconstruction capability is quite different. Zhou *et al.* [43, 44, 4, 6] aim to tackle accumulated pose error and integrate global pose optimization. Steinbrücker *et al.* [35, 16] enable multiple resolution reconstruction capability. Kähler *et al.* [17, 20] significantly improve the computational efficiency so that 3D reconstruction can be integrated on a mobile device. The first step of fusing the incoming depth map is to project the voxel back to the depth map with pre-estimated rigid body motion and a projective camera model. Then, it searches the nearest neighbor on the projected depth map and evaluates the projective distance. Usually, it computes the weighted average of individual TSDFs for each depth map in the volume. Nießner *et al.* [27] introduced voxel hashing as an essential computation improvement. They employ an efficient hashing strategy; it encodes the spatial coordinates with a hashing function. Unlike the previous implementation, we use voxel hashing to enable voxel-based association instead of the KD-tree-based search. Because the nearest neighbor search by KD-tree is still a bottleneck when the point cloud is dense.

Surfel/point-based methods. Unlike voxel-based methods, the surfel/point-based methods store accumulated surfels[30]/points[24] to reconstruct the scanned scene. Surfel/point-based methods aim to represent local surface with point samples, encode additional information, like radius, confidence, timestamp. Several reconstruction algorithms [19, 37, 24, 42, 23, 11, 31] employ this kind of data representation. Stückler *et al.* [37] employ octrees to contain multiple resolution surfel maps. Keller *et al.* [19] introduced a widely used way to fuse each surfel or point. Points or surfels are projected into each incoming depth map through an estimated camera pose. They [19] calculate the exactly projected pixel on the depth map via a super-sampled index map. Schöps *et al.* [31] improved the efficiency by directly obtaining indices from the projection instead of loading them from an index map. Whelan *et al.* [42] label surfels as active and inactive; they only use the active surfels for depth map fusion. After the projection on the depth map is determined, the geometry consistency or photometric consistency is verified. Finally, a weighted average, typically Gaussian weight, is applied to handle minor uncertainties. After a certain number of steps, the unstable points [19], or conflicting surfels [31] are removed or replaced. Unlike the previous methods, there are not any projection or back-projection processes in our fusion framework. Therefore no image index maps are needed. Another improvement is that we employ hybrid representation, which combines voxels and points. Points in a predefined resolution voxel represent the local geometry, missing from surfel/point-based methods. By utilizing the geometric information, our approach can handle the uncertainties explicitly.

Displacement interpolation. McCann[25] first introduced the concept of displacement interpolation in the context of quadratic cost on Euclidean space. It gives a mass-preserving way to interpolate between probability measures, which is derived from the Monge-Kantorovich problem. A simple example for the Monge-Kantorovich prob-

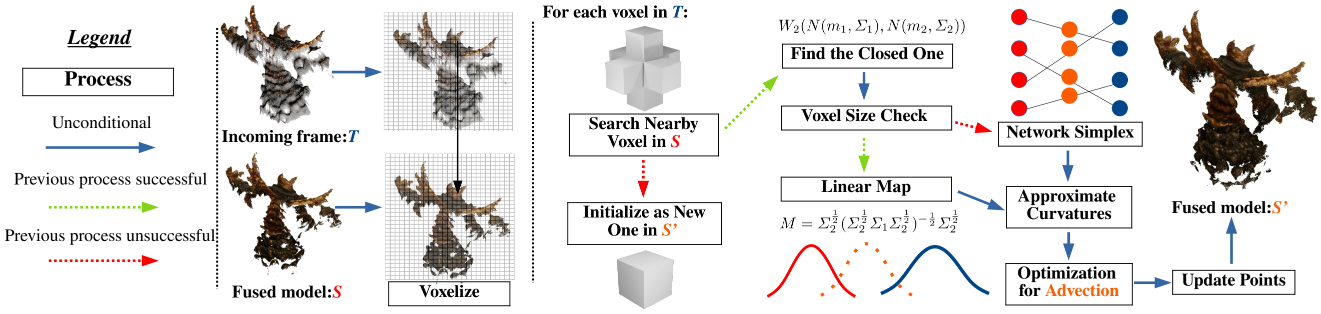
Legend
Process
Unconditional
Previous process successful
Previous process unsuccessful
Incoming frame:T
Fused model:S
Voxelize
For each voxel in T:
Search Nearby Voxel in S
Initialize as New One in S'
$W_2(N(m_1,\Sigma_1),N(m_2,\Sigma_2))$
Find the Closed One
Voxel Size Check
Linear Map
$M=\Sigma_2^{\frac{1}{2}}(\Sigma_2^{\frac{1}{2}}\Sigma_1\Sigma_2^{\frac{1}{2}})^{-\frac{1}{2}}\Sigma_2^{\frac{1}{2}}$
Network Simplex
Approximate Curvatures
Optimization for Advection
Update Points
Fused model:S'

Figure 2. Overview of our pipeline

lem is that there is a heap of cargos *f(x)* that needs to reshape to a target shape *g(y)* in another place with a cost function *c(x, y)*. The optimal transference plan π is obtained by minimizing the integral of the cost function,

$$\min_{\pi}\int c(x,y)d\pi(x,y) \tag{1}$$

To further elaborate displacement interpolation, we assume the cargo transport has some dynamics, namely, time-dependent transport. $c^{0,1}(x,y)$ is cost functions related with a Lagrangian action $\mathcal{A}(\gamma)$ on $\mathcal{X}\times\mathcal{X}$ as below[40],

$$c^{0,1}(x,y)=\inf\left\{\mathcal{A}^{0,1}(\gamma);\gamma_0=x,\gamma_1=y;\gamma\in\mathcal{C}([0,1];\mathcal{X})\right\} \tag{2}$$

where $\mathcal{C}$ is a class of continuous curves on $\mathcal{X}$, and γ is the curve characterized by a Lagrangian action $\mathcal{A}(\gamma)$, at the beginning of the curve $\gamma_0 = x$ $(t = 0)$, and the end of the curve is $\gamma_1 = y$ $(t = 1)$. Then, action $\mathcal{A}(\gamma)$ is defined as the integral of a Lagrangian over the curve:

$$\mathcal{A}(\gamma)=\int_{\gamma}L(\gamma_t,\dot{\gamma_t},t)dt \tag{3}$$

The infimum of the action functional $\mathcal{A}(\gamma)$ over all curves is the minimizing, constant-speed geodesic curve or simply geodesic. In the context of fusion, displacement interpolation is to move points along the geodesic curve. In this paper, we assume a sufficiently small local surface follows Gaussian distribution. On L^2-Wasserstein space, the geodesic curve between Gaussian measures is known to have a straightforward solution. However, the points could be non-uniform sampling and missing data. In these scenarios, it can not be directly applied to the fusion process. An alternative method based on the Hitchcock-Koopman formulation is employed because covariance can be incorrect when there are few points in a voxel.

3. Geometry-Guided Displacement Interpolation

Typically, the pipeline of 3D reconstruction via RGB-D sensors consists of depth map preprocessing, camera pose estimation, fusion, surface reconstruction. This paper mainly focuses on the fusion stage (Fig.2 provides a brief overview of our unique fusion pipeline) while giving a short description of other parts. Our algorithm's input is a set of RGB-D frames from a calibrated sensor (see Fig.3). Our output is an accurate dense point cloud. The surface reconstruction [18] can be quickly built on our output.

We employ a hybrid representation that stores points in a voxel hashing style, allowing for efficient voxel search. Our proposed fusion method leverages the local geometry through curvatures. It exploits that geometry structure information directly from the local surface without any photometric assumptions and can operate on both align to depth and align to RGB settings. It provides high flexibility and adaptiveness for various kinds of sensor settings.

3.1. Gaussian Voxel

Our proposed method uses a hybrid representation and frame to model fashion. We do not apply back-projection and build any image index map; the neighbors are frequently required. However, the KD-tree-based nearest neighbor search is a bottleneck when the point cloud is dense. To avoid costly nearest neighbor search, our hybrid representation stores points in a voxel hashing style similar to [22]. Different from [22], which only stores mean and covariance matrix in each voxel. We store points in a predefined resolution voxel with hashing function (Fig.2 **Voxelized**). Each voxel represents a local surface that follows Gaussian distribution [2, 36]. Thus, we call it Gaussian voxel. A fast voxel-to-voxel search can be applied between the incoming depth map and the fused model. Typically, it searches seven nearby voxels and chooses the one that has the closest Wasserstein distance (Fig.2 **Search Nearby Voxel, Find the Closest One**). If there are no voxels nearby, the incoming voxel is initialized as a new one in the fused model(Fig.2 **Initialized as New One**). Given a Gaussian measure μ_i to the point sets in the voxel, each point p_j comprises a world coordinate vector w_p (our fusion process primarily focus on updating the coordinate), a normal vector n_p, an RGB color vector c_p, an advection count f_p, and a lifetime l_p used to eliminate the outliers.

Each point should be advected a certain number of times in a given lifetime. Otherwise, it will be removed at the end of its lifetime. Our approach's key part is to find the proper advection between local surfaces along the geodesic curves. For Gaussian measures, an obvious solution has been introduced by [8, 12, 21, 28]. We will give a detailed discussion in the next section.

3.2. Linear Map on L^2-Wasserstein Space

The linear map can be easily derived from the covariance matrix (Fig.2 **Linear Map**). The algorithm steps are given at the end of this subsection. First, we introduce some fundamental ideas of L^2-Wasserstein space [38]. The central idea is that there are pairs of subsets on a separable, complete, metric space equipped with Borel $\delta-$algebra. The distance function between them is derived from the Monge-Kantorovich problem. We give brief definitions that follow [40, 38]. Without loss of generality, first, we give the definition based on Borel probability measures. Then, we give the linear map for Gaussian measures.

$\mathbf{L^2}$-Wasserstein space. Let $(\mathcal{X}, d)$ be a separable, complete, metric space. Given two Borel probability measures μ_0, $\mu_1 \in \mathcal{P}_2(\mathcal{X})$ with finite second moments on $\mathcal{X}$ satisfying [38]

$$\int_{\mathcal{X}} d(x,y)^2 \, d\mu(y) < \infty \tag{4}$$

Let π be the transference plan between μ_0 and μ_1 on $\mathcal{X} \times \mathcal{X}$ [40]. The marginals of π are μ_0 and μ_1, thus

$$\pi\left[\psi \times \mathcal{X}\right] = \mu_0\left[\psi\right], \pi\left[\mathcal{X} \times \psi\right] = \mu_1\left[\psi\right] \tag{5}$$

(5) holds for all Borel sets $\psi \in \mathcal{X}$. The L^2-Wasserstein distance function $W_2(\mu_0, \mu_1)$ between μ_0 and μ_1 in $\mathcal{P}_2(\mathcal{X})$ is defined by

$$W_2(\mu_0,\mu_1) = \left(\inf_{\pi \in \varPi(\mu_0,\mu_1)} \int_{\mathcal{X}\times\mathcal{X}} d(x,y)^2 \, d\pi(x,y)\right)^{\frac{1}{2}} \tag{6}$$

where $\varPi(\mu_0, \mu_1)$ is the collection of transference plans between μ_0 and μ_1, the infimum over $\varPi(\mu_0, \mu_1)$ is the L^2-Wasserstein distance, and the corresponding transference plan is optimal. The $W_2(\mu_0, \mu_1)$ is the distance function on $\mathcal{P}_2(\mathcal{X})$, and $(\mathcal{P}_2(\mathcal{X}), W_2)$ is called L^2-Wasserstein space over $\mathcal{X}$ [38].

For Euclidean space, the optimal transference plans are characterized by push-forward. There is a measurable map $M : \mathbb{R}^n \to \mathbb{R}^n$. Define the push-forward for a Borel probability measure μ_0 on $\mathbb{R}^n$ as follow

$$M_\sharp\mu_0[\psi] = \mu_1[\psi], \text{ if } \mu_1[\psi] = \mu_0[M^{-1}(\psi)] \tag{7}$$

(7) holds for all Borel sets $\psi \in \mathbb{R}^n$. The identity map on $\mathbb{R}^n$ is denoted by id.

$\mathbf{W_2}$ distance between Gaussian measures. In this paper, we assume the local surfaces follow Gaussian distribution. The L^2-Wasserstein distance between Gaussian measures was explicitly given by: I.Olkin *et al*. [28, 8, 12, 21] as follow

$$W_2(N(m_1,\Sigma_1), N(m_2,\Sigma_2)) = \\ |m_1 - m_2|^2 + tr\Sigma_1 + tr\Sigma_2 - 2tr\sqrt{\Sigma_2^{\frac{1}{2}}\Sigma_1\Sigma_2^{\frac{1}{2}}} \tag{8}$$

where $N(m, \Sigma)$ is Gaussian distribution, tr is the trace of a matrix, m is the mean, and $\Sigma \in Sym^+(n, \mathbb{R})$ is covariance matrix (in practical the covariance matrix should be symmetric positive definite when there are multiple points in a Gaussian voxel). Since Σ is a symmetric positive definite matrix, we define $\sqrt{\Sigma} = \Sigma^{\frac{1}{2}}$, and $\Sigma^{\frac{1}{2}} \cdot \Sigma^{\frac{1}{2}} = \Sigma$. McCann [25] shows that the displacement interpolation between any two Gaussian measures is also a Gaussian measure.

Displacement interpolation as geodesics. Based on [25] the linear map M between two centralized Gaussian distribution $N(0, \Sigma_1)$,$N(0, \Sigma_2)$ can be given by

$$M = \Sigma_2^{\frac{1}{2}}(\Sigma_2^{\frac{1}{2}}\Sigma_1\Sigma_2^{\frac{1}{2}})^{-\frac{1}{2}}\Sigma_2^{\frac{1}{2}}, \quad f(x) = Mx \tag{9}$$

$f(x) = Mx$ push-forward $N(0, \Sigma_1)$ to $N(0, \Sigma_2)$, the optimal transference plan between $N(0, \Sigma_1)$, $N(0, \Sigma_2)$ is $[id \times f]_\sharp N(0, \Sigma_1)$. Define advection $A(\alpha)$ by

$$A(\alpha) = [(1-\alpha)I + \alpha M] \tag{10}$$

where I is the identity matrix. The geodesic from $N(0, \Sigma_1)$ to $N(0, \Sigma_2)$ is $N(0, A(\alpha)\Sigma_1 A(\alpha))$ for $\alpha \in [0, 1]$. Although, we focus on Gaussian measures in this paper. Without loss of generality, we give the corollary of [40] *Theorem Displacement interpolation* which states displacement interpolation as geodesics on L^2-Wasserstein space as follow

Corollary 1 ***(Displacement interpolation as geodesics on L^2-Wasserstein space)***
Given two Borel probability measures $\mu_0, \mu_1 \in \mathcal{P}_2(\mathcal{X})$ on a complete, separable, metric, locally compact space $(\mathcal{X}, d)$, where $\mathcal{P}_2(\mathcal{X})$ is the probability measures with finite moment of order 2. $(\mathcal{P}_2(\mathcal{X}), W_2)$ is L^2-Wasserstein space paired with Wasserstein distance W_2. There is a continuous curve $(\mu_\alpha)_{0\leq\alpha\leq1} \in \mathcal{P}_2(\mathcal{X})$.
Given two equal properties:
(1) $(\mu_\alpha)_{0\leq\alpha\leq1}$ is the law of $(\gamma_\alpha)_{0\leq\alpha\leq1}$. (γ_0, γ_1) is an optimal coupling, and γ is a geodesic with constant speed.
(2) $(\mu_\alpha)_{0\leq\alpha\leq1}$ is a geodesic curve in the space $(\mathcal{P}_2(\mathcal{X}), W_2)$.

Corollary 1 is admitted by *Theorem Displacement interpolation* and its corollaries [40] where the $p > 1$. There is an important remark that for different $p > 1$, the geodesic curves in $\mathcal{P}_p(\mathcal{X})$ are also different. Namely, geodesic on L^2-Wasserstein space is not the same as L^p-Wasserstein space ($p > 2$). [40] also gives the uniqueness of displacement interpolation, the displacement interpolation is unique if the optimal transference plan is unique.

Linear map between Gaussian voxels. It is easy to compute the mean and covariance for each Gaussian voxel, assume $N(m_1, \Sigma_1)$ is from the fused model, $N(m_2, \Sigma_2)$

is from the incoming frame. For each voxel in the incoming frame, we find seven nearby voxels in the fused model. First, we compute the Wasserstein distance through (8) and choose the closest one; then, we shift $N(m_1, \Sigma_1)$ to $N(0, \Sigma_1)$ in the voxel from the fused model, the linear map can easily be calculated by (9); finally, we apply the advection from (10) on the fused model and transform $N(0, A(\alpha)\Sigma_1 A(\alpha))$ back to $N((1-\alpha)m_1 + \alpha m_2, A(\alpha)\Sigma_1 A(\alpha))$. Then, the coordinates of the points in the fused voxel are updated. Before introducing geometry-guided constraint to determine proper α in (10), we discuss this method's limitation and give a complementation method in the next section.

3.3. Fusion with Hitchcock-Koopman Formulation

The presented method in the previous section depends on the covariance matrix. The core idea is to interpolate one Gaussian distribution to another through the covariance matrix. However, the points in Gaussian voxels can be non-uniform sampling and missing data. Namely, the covariance matrix can not represent the actual distribution through a few points. It should be at least 20 points by experience [33]. Therefore, the map could be incorrect with a few points in one voxel. For this reason, we introduce a complementation method, which is based on the Hitchcock-Koopman formulation [9].

Simplified Hitchcock-Koopman formulation. For Gaussian voxel with few points, we follow the simplified Hitchcock-Koopman formulation [9]. The original one can handle the distribution of a product from several sources of supply to numerous targets. It means the supplies may vary from the demands. The transport goods can be divided into several parts. In our case, we prefer to point-to-point fusion. To perform the point-to-point fusion, the number of points will be equal after removing the redundant ones far from the center. Given a set of points from source (fused model) $S = \{s_i, i = 1...n\}$ and a set of points from the target (incoming depth map) $T = \{t_j, j = 1...n\}$, and a transport cost $d_{i,j}$ to move one point from S to T. Assume the number of points from S, and T are equal. Then the mass transport problem is simplified to an assignment problem. The computation is also reduced. The energy is given by

$$\begin{aligned} &\min_a \sum_i \sum_j d_{i,j} a_{i\to j} \\ &\text{such that} : a_{i\to j} = \begin{cases} 1 & \text{if assign i to j} \\ 0 & \text{if not assign i to j} \end{cases} \\ &\sum_i a_{i\to j} = 1, \sum_j a_{i\to j} = 1 \end{aligned} \tag{11}$$

Thus, the collection of $a_{i\to j}$ is our assignment plan [1], which is a special case of the minimum-cost flow problem (MCFP). There are two ways to solve (11): transportation-simplex-based methods and network-simplex-based methods. Bonneel *et al.* [3] analyze these two main categories. Their experiment shows that network simplex is more efficient than transportation simplex. So that we employ a network simplex algorithm [7] similar to [3](Fig.2 **Network Simplex**). The network simplex algorithm is efficient in solving the MCFP. It has a known complexity in $\mathcal{O}(n^3)$ [41]. The covariance-based linear map is more computation efficient when the number of points is getting larger. Our flexible interpolation scheme allows switching between two methods according to the number of points in a voxel (Fig.2 **Voxel Size Check**).

Advection between points. After the assignment plan is determined, the advection can be defined as

$$w_i = (1-\alpha)s_i + \alpha \sum_j a_{i\to j} t_j, \quad i, j = 1...n \tag{12}$$

where w_i is the updated coordinate, $\alpha \in [0, 1]$, $s_i \in \mathbb{R}^3$ represents coordinate of a point from the fused model S, and $t_j \in \mathbb{R}^3$ is from the incoming frame T. The next question is how to determine the proper α. To tackle sensor depth and poses uncertainties, we give a geometry constraint based on curvatures. The detail will be given in the next section.

3.4. Geometry-Guided Advection

To obtain a proper advection, namely, choose a suitable $\alpha \in [0, 1]$. We employ a geometry constraint based on local geometry. We compute some approximate results for differential geometry using the covariance techniques. Then, a geometry-guided advection based on these results is proposed.

Approximate curvatures. For a Gaussian voxel, we first need to find the tangent plane $T_{\overline{p}}M$ on the center $\overline{p}$. Thus, we need to calculate the orthogonal frame $n_0, n_1, n_2 \in \mathbb{R}^3$, which are eigenvectors of the covariance matrix. Consider the normal n_0 is orthogonal to $T_{\overline{p}}M$, which is obtained as the eigenvector corresponding to the smallest eigenvalue. For brevity, we abbreviate n_0 as n. Define function as

$$\begin{aligned} L =& \langle r_{uu}, n\rangle = -\langle r_u, n_u\rangle \\ M =& \langle r_{uv}, n\rangle = -\langle r_u, n_v\rangle \\ N =& \langle r_{vv}, n\rangle = -\langle r_v, n_v\rangle \end{aligned} \tag{13}$$

where $r(u, v) = (x(u, v), y(u, v), z(u, v))$, (u, v) is the parameter of the local surface. $r_u, r_v, r_{uu}, r_{uv}, r_{vv}, n_u, n_v$ are partial derivatives. The normal curvature is defined as:

$$\begin{aligned} k_n = \frac{II}{I} &= \frac{-\langle dr, dn\rangle}{\langle dr, dr\rangle} \\ &= \frac{Ldudu + 2Mdudv + Ndvdv}{\langle dr, dr\rangle} \end{aligned} \tag{14}$$

where $dr = r_u du + r_v dv$, I is the first fundamental form, II is the second fundamental form. $n = r_u \wedge r_v / |r_u \wedge r_v|$ is the normal vector which is perpendicular to tangent plane $T_{\overline{p}}M$. The first fundamental form is the arc length of point p_i to

$\overline{p}$, which can be approximate with $\parallel p_i - \overline{p} \parallel_2^2$. The second fundamental form II can be represented by the algebraic distance δ between $T_{\overline{p}}M$ and arbitrary point p_i in Gaussian voxel,

$$\delta(p_i, T_{\overline{p}}M) = \langle n, \overrightarrow{p_i\overline{p}} \rangle \tag{15}$$

$$\begin{aligned} \overrightarrow{p_i\overline{p}} = \Delta r = & r(u_0 + \Delta u, v_0 + \Delta v) - r(u_0, v_0) \\ = & r_u(u_0, v_0)\Delta u + r_v(u_0, v_0)\Delta v \\ & + \frac{1}{2}(r_{uu}(u_0, v_0)\Delta u \Delta u \\ & + 2r_{uv}(u_0, v_0)\Delta u \Delta v + r_{vv}(u_0, v_0)\Delta v \Delta v) \\ & + o(\Delta u \Delta u + \Delta v \Delta v) \end{aligned} \tag{16}$$

since r_u, r_v are perpendicular to n and $o(\Delta u^2 + \Delta v^2)$ is high-order infinitesimal. (15) is reduced to

$$\delta(p_i, T_{\overline{p}}M) \approx \frac{1}{2}(L\Delta u \Delta u + 2M\Delta u \Delta v + N\Delta v \Delta v) \tag{17}$$

Then we have

$$II \approx \delta(p_i, T_{\overline{p}}M) \tag{18}$$

$$II \approx (p_i - \overline{p})^T \cdot n \tag{19}$$

Finally, the approximate solution of normal curvature is

$$k_n = \frac{II}{I} \approx \frac{(p_i - \overline{p})^T \cdot n}{\parallel p_i - \overline{p} \parallel_2^2} \tag{20}$$

Optimization for advection. The largest normal curvature and smallest normal curvature are called principal curvatures k_0, k_1. The Gaussian curvature is defined as $k_0 \cdot k_1$. Mean curvature is defined as $(k_0 + k_1)/2$. To tackle the sensor depth and poses uncertainties, we want the result of advection as smooth as possible, while the uncertainties usually lead to sharp artifacts (see Fig.1). The optimization of advection is given as follow

$$\arg\min_{\alpha}(\|k_0 \cdot k_1\|_2^2 + \|\frac{(k_0 + k_1)}{2}\|_2^2) \tag{21}$$

(21) can be easily computed by iteration.

Eigenvalue perturbation. This covariance matrix-based advection remains some limitations. The main concern is the eigenvalue perturbation [10, 39]. Noises that come with sensors can not be obliterated. Some additive noises in the Gaussian voxel are added to the coordinates(x, y, z). The covariance matrix $\tilde{C}$ with additive noises and without additive noises C are considered as positive definite real symmetric matrices (eigenvalue λ is always larger than zero in our case). We assume that perturbations are sufficiently small, *i.e.*, $\tilde{C} - C = \Delta C$ for some small ΔC, which results in a small perturbation in eigenvector n_i and λ_i, *i.e.*, $\tilde{n}_i - n_i = \Delta n_i$ and $\tilde{\lambda}_i - \lambda_i = \Delta \lambda_i$. The $T_{\overline{p}}M$ is spaned by n_1, n_2. n_0 is the normal vector perpendicular to the plane, the corresponding eigenvalues are $\lambda_0 \leq \lambda_1 \leq \lambda_2$. An ideal situation is $\lambda_0 < \lambda_1 = \lambda_2$. However, in the actual case, there are two situations as follow: 1) $\|\tilde{\lambda}_0 - \tilde{\lambda}_1\| < \epsilon$, $\|\tilde{\lambda}_1 - \tilde{\lambda}_2\| < \varepsilon$, where ε is sufficiently small and ϵ is much larger than ε; 2) $\|\tilde{\lambda}_0 - \tilde{\lambda}_1\| < \epsilon$, $\|\tilde{\lambda}_1 - \tilde{\lambda}_2\| < \varepsilon$, where both ε and ϵ are sufficiently small. Situation 1) is a double-edged sword. A small perturbation may not affect the computation of principal curvatures. However, situation 1) implies the local surface may be a flat plane. It is detrimental to choose the proper advection. It usually happens when there are only three points in one Gaussian voxel since three points always fit a plane. A larger voxel is required. But it may lead to situation 2), all eigenvalues are close to each other. If the voxel is too large, it fails to capture the property of local geometry. Then, it isn't easy to find a proper tangent plane. A small perturbation may lead to eigenvalue perversion, which means the normal vector can be disordered.

Adaptive voxel resolution. Adaptive voxel resolution is required to solve this dilemma. A reasonable choice is to compute the average point radius of the incoming depth map and the fused models, then choose the maximum one as the voxel resolution. It means the voxel resolution changes over time.

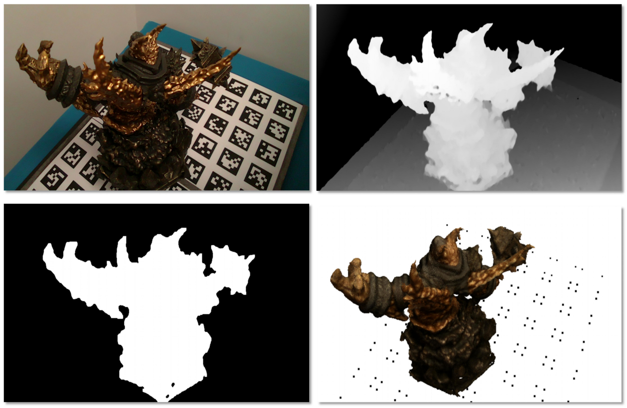

Figure 3. Illustration for handheld RGB-D real dataset, from left-top to bottom-right, RGB image, depth map, mask, our reconstruction. A markerboard [29] is used for mask and sensor (L515) pose estimation.

	Threshold Methods	1mm	2mm	3mm	4mm	5mm
Acc.	ElasticFusion[42]	28.40	44.86	55.56	63.56	69.79
	SurfelMeshing[31]	67.94	83.24	90.32	93.80	95.63
	GaussianFussion(ours)	**80.90**	**92.05**	**95.92**	**97.64**	**98.49**
Com.	ElasticFusion[42]	35.32	74.22	86.45	91.62	94.20
	SurfelMeshing[31]	68.35	**87.93**	**95.15**	**97.30**	**98.32**
	GaussianFussion(ours)	**71.67**	87.24	93.70	96.20	97.40

Table 1. Qualitative comparison on Skeleton between our method, ElasticFusion[42] and SurfelMeshing[31](Accuracy[%], Completeness[%]). Our method achieves the highest precision.

4. Experimental Results

In this section, we evaluate our fusion framework and the rivals, including [31, 42] on our real object dataset.

	Data	House_0	House_1	House_2	House_3	Ragnaros	Exia_0	Exia_1	Skeleton	Fighter	Squirtle	Roshan
	Methods	65f	98f	64f	300f	300f	199f	300f	195f	100f	300f	300f
points/ surfels	ElasticFusion[42]	0.05M	0.10M	0.08M	0.07M	0.13M	0.11M	0.08M	0.09M	0.07M	0.08M	0.07M
	SurfelMeshing[31]	0.21M	0.40M	0.30M	0.59M	1.16M	0.54M	0.80M	0.67M	0.35M	0.77M	0.27M
	GaussianFussion(ours)	**0.31M**	**0.64M**	**0.41M**	**1.20M**	**1.29M**	**0.88M**	**0.99M**	**1.02M**	**0.44M**	**0.82M**	**0.51M**

Table 2. Qualitative comparison between our method, ElasticFusion[42] and SurfelMeshing[31](M = million, f = frames).

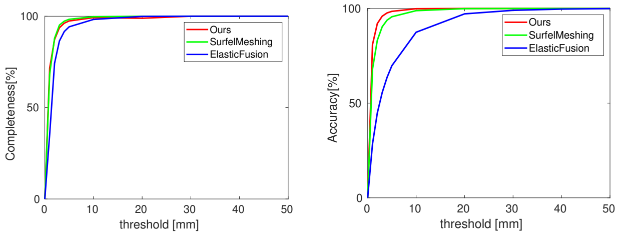

Figure 4. Completeness and accuracy plot for wide evaluation thresholds ($1mm$, $2mm$, $3mm$, $4mm$, $5mm$, $10mm$, $20mm$, $30mm$, $40mm$, $50mm$).

4.1. Reconstruction on Real Object

RGB-D sensor. The quality of the depth map is the essential factor for 3D reconstruction. The depth map must have enough resolution to sense the delicate geometric details. Since we focus on accurate 3D reconstruction with a consumer-level RGB-D sensor, the real object data is obtained by the latest low-cost device Intel RealSense LiDAR Camera L515 [13]. The average depth accuracy error at $1m$ is less than $5mm$, and the standard deviation is $2.5mm$ at VGA resolution [14]. To achieve a full depth map, L515 is equipped with an IR laser beam and utilizes a Micro-Electro-Mechanical System (MEMS) to scan the entire field of view (FOV). The onboard vision ASIC processes the original signal from the photodiode and outputs a depth map [14]. To our best knowledge, L515 is the most accurate off-shelf RGB-D sensor in terms of size and cost for the indoor scene. Thus, we do not evaluate our framework on the old model RGB-D sensors.

Data acquisition. We use handheld L515 to scan several objects indoor. The initial pose is computed through a markerboard [29] placed under the item. Then, the voxelized GICP [22] is applied to minimize the registration error. To accelerate the reconstruction process, we estimate a mask through the markerboard. The masked depth maps, RGB images, and poses are the input for the evaluated methods. The input is shown in Fig.3. There are two different align settings for L515: align depth maps to RGB frames and align RGB frames to depth maps. The former introduces extra depth estimation errors due to the interpolation of the depth map. RGB frame has a higher resolution (1280×720) than the depth map (1024×768) so that each depth pixel is extended to a 2×2 patch in the former setting [34]. Since the depth map's quality is more important for 3D reconstruction, we employ the latter setting. To make the input data consistent for all evaluated methods, we use the same depth maps input and back-project the depth maps into the point clouds before the fusion stage.

Qualitative results. Our method focuses on accurate real object reconstruction with a low-cost RGB-D sensor, excluding the synthetic datasets. Limited by the article space, we give results of eleven objects. See Table 2 and Fig.5 for an overall reconstruction comparison with the state-of-the-art methods. To visualize the results, both our technique and ElasticFusion[42] employ [18] to obtain a triangle mesh from the output point cloud, while SurfelMeshing [31] can generate high-quality mesh during reconstruction. With the same input depth maps, our method achieves the largest number of points. It indicates our technique achieves ultra-high geometric detail along with a complete global geometric shape. For further quantitative evaluation, we obtain the ground-truth model of object Skeleton through an industrial sensor of resolution $0.08mm$. We use voxelized ICP [22] to align the reconstructions to the ground-truth model. Then, we compute accuracy and completeness with a given threshold similar to [32]. Results at $1mm$, $2mm$, $3mm$, $4mm$, $5mm$ are shown in Tab.1. We also give a plot for wide thresholds, as shown in Fig.4. Our technique always achieves the highest accuracy score, Tab.1 and Fig.4 show that this fact does not depend on the threshold.

Global geometric shape. It is challenging to obtain sub-pixel reprojection error for all input frames. The depth map back-projection methods cannot explicitly handle the misalignment when the number of frames gets larger. As shown in Fig.1 (Ragnaros 300 frames), Fig.5 (House_3 300 frames, Exia_0 199 frames, Roshan 300 frames), Fig.7 (Squirtle 300 frames, Exia_1 300 frames). Thus, they can not achieve a complete global geometric shape like ours.

Local geometric detail. Delicate geometric details are hard to reconstruct correctly. The depth map back-projection methods require accurate camera poses. The weighted average between measurements and their back-projections ruin the local geometric details. As shown in Fig.5 Skeleton, the nasal septum is visible from our geometry-guided displacement interpolation, while others mess it up. The same results are shown in Fig.7 fingers of Squirtle, air vent of Exia_1; and Fig.6 aircraft landing gear.

5. Conclusion

We introduce displacement interpolation into the 3D reconstruction area for the first time and present a novel fusion framework based on the proposed trick of geometry-guided advection. By integrating the displacement interpolation and the curvature constraint, our technique over-

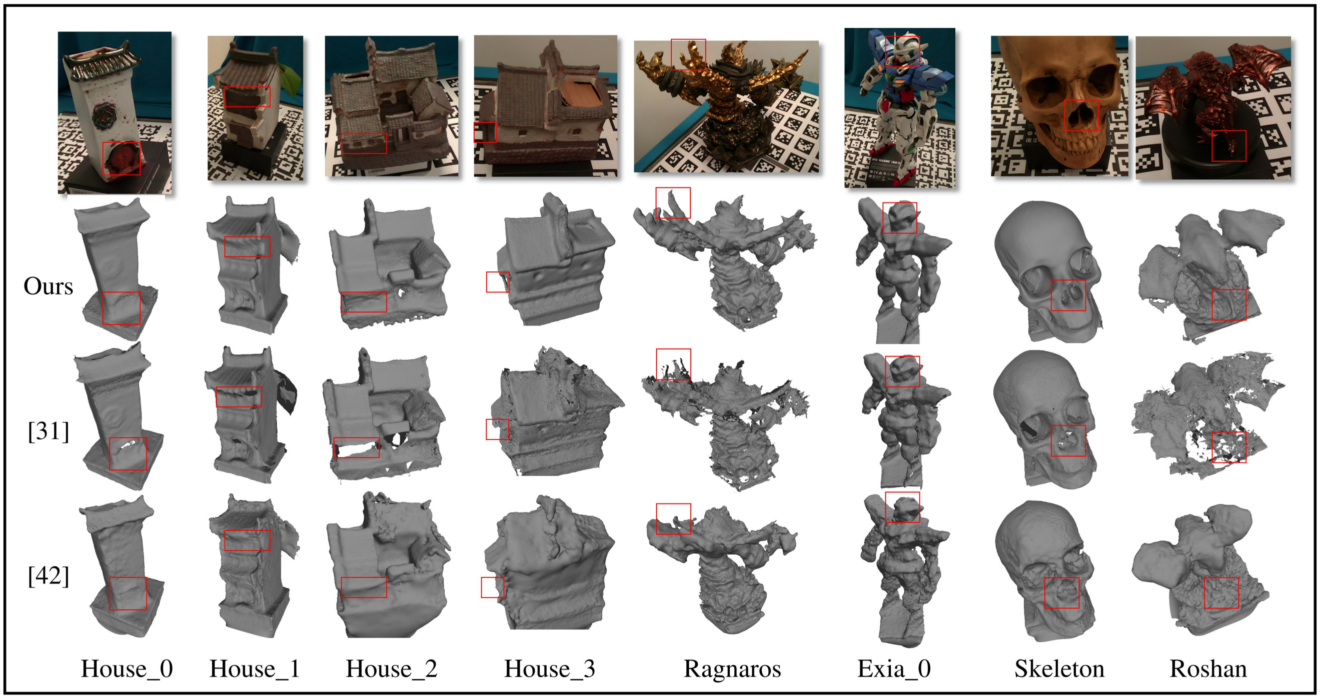

Figure 5. First row images are RGB images of the scanned objects, and the following rows are reconstruction from our technique, SurfelMeshing[31], and ElasticFusion[42] respectively. Our method achieves the complete global geometric shape and outperforms others in delicate geometric details. See Table 2 for more details.

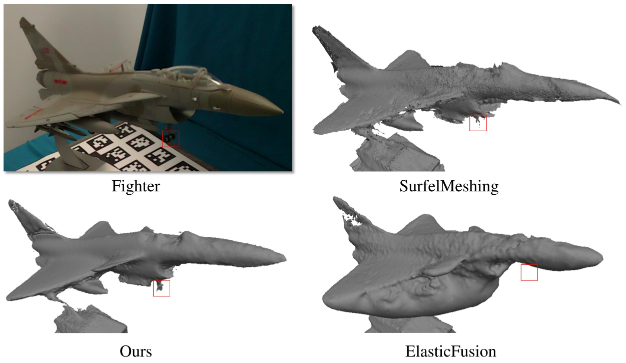

Figure 6. The aircraft landing gear on the left-top is small. Our method can get a reasonable reconstruction, while SurfelMeshing [31] and ElasticFusion [42] failed.

comes the shortcomings of the depth map back-projection fusion methods: 1) the geodesic curves resulted from the mass transport problem reveal the geometric structures of the local surface; 2) the motions along the geodesic curves are guided by local geometric properties, *i.e.*, Gaussian curvature and mean curvature. Our approach makes the best of the meaningful geometric properties through these two geometry-guided characteristics. Thus, the proposed method achieves the most delicate geometric details on thin objects for which the depth map back-projection fusion scheme failed. A significant number of experimental results show that our technique gets the best reconstruction quality on all items compared with the state-of-the-art methods. In all, the geometry-guided displacement interpolation makes our approach prominent.

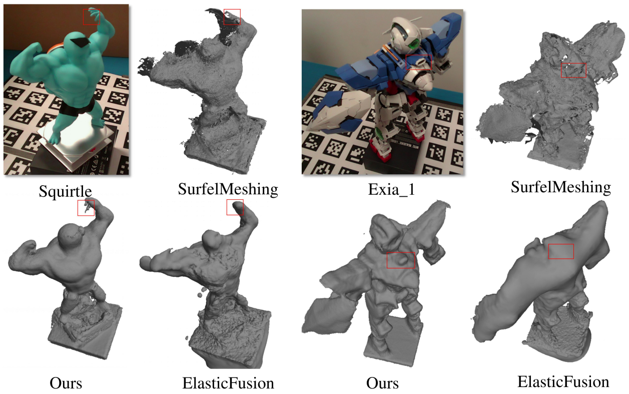

Figure 7. Our approach outperforms SurfelMeshing[31] and ElasticFusion[42] in delicate geometric details, like tiny fingers in Squirtle, air vent of Exia_1.

Acknowledgments. Thanks to editors and reviewers for the time and effort spent in handling this paper. This work is supported by NSFC under grants 61860206007 and U19A2071, as well as the funding from Sichuan University under grant 2020SCUNG205.

References

[1] Ravindra K Ahuja, Thomas L Magnanti, and James B Orlin. *Network Flows: Theory, Algorithms, and Applications*. Prentice hall, 1993.

[2] Peter Biber and Wolfgang Straßer. The normal distributions transform: A new approach to laser scan matching. In *Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)*, volume 3, pages 2743–2748. IEEE, 2003.

[3] Nicolas Bonneel, Michiel Van De Panne, Sylvain Paris, and Wolfgang Heidrich. Displacement interpolation using lagrangian mass transport. In *Proceedings of the 2011 SIGGRAPH Asia Conference*, pages 1–12, 2011.

[4] Sungjoon Choi, Qian-Yi Zhou, and Vladlen Koltun. Robust reconstruction of indoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5556–5565, 2015.

[5] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312, 1996.

[6] Angela Dai, Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Christian Theobalt. Bundlefusion: Real-time globally consistent 3d reconstruction using on-the-fly surface reintegration. *ACM Transactions on Graphics (ToG)*, 36(4):1, 2017.

[7] K Damian, B Comm, and M Garret. *The minimum cost flow problem and the network simplex method*. PhD thesis, Ph. D. Dissertation, Dissertation de Mastere, Université College Gublin, Irlande, 1991.

[8] DC Dowson and BV Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 12(3):450–455, 1982.

[9] Merrill M Flood et al. On the hitchcock distribution problem. *Pacific Journal of mathematics*, 3(2):369–386, 1953.

[10] Valerie Fraysse, Michel Gueury, Frank Nicoud, and Vincent Toumazou. Spectral portraits for matrix pencils. 1996.

[11] Wei Gao and Russ Tedrake. Surfelwarp: Efficient non-volumetric single view dynamic reconstruction. *arXiv preprint arXiv:1904.13073*, 2019.

[12] Clark R Givens, Rae Michael Shortt, et al. A class of wasserstein metrics for probability distributions. *The Michigan Mathematical Journal*, 31(2):231–240, 1984.

[13] Intel. Intel realsense™ lidar camera l515. [EB/OL]. `https://www.intelrealsense.com/lidar-camera-l515/` Accessed January 27, 2021.

[14] Intel. Lidar camera l515 datasheet. [EB/OL]. `https://dev.intelrealsense.com/docs/lidar-camera-l515-datasheet` Accessed January 27, 2021.

[15] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568, 2011.

[16] Olaf Kähler, Victor Prisacariu, Julien Valentin, and David Murray. Hierarchical voxel block hashing for efficient integration of depth images. *IEEE Robotics and Automation Letters*, 1(1):192–197, 2015.

[17] Olaf Kähler, Victor Adrian Prisacariu, Carl Yuheng Ren, Xin Sun, Philip Torr, and David Murray. Very high frame rate volumetric integration of depth images on mobile devices. *IEEE transactions on visualization and computer graphics*, 21(11):1241–1250, 2015.

[18] Michael Kazhdan and Hugues Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics (ToG)*, 32(3):1–13, 2013.

[19] Maik Keller, Damien Lefloch, Martin Lambers, Shahram Izadi, Tim Weyrich, and Andreas Kolb. Real-time 3d reconstruction in dynamic scenes using point-based fusion. In *2013 International Conference on 3D Vision-3DV 2013*, pages 1–8. IEEE, 2013.

[20] Matthew Klingensmith, Ivan Dryanovski, Siddhartha S Srinivasa, and Jizhong Xiao. Chisel: Real time large scale 3d reconstruction onboard a mobile device using spatially hashed signed distance fields. In *Robotics: science and systems*, volume 4, 2015.

[21] Martin Knott and Cyril S Smith. On the optimal mapping of distributions. *Journal of Optimization Theory and Applications*, 43(1):39–49, 1984.

[22] Kenji Koide, Masashi Yokozuka, Shuji Oishi, and Atsuhiko Banno. Voxelized gicp for fast and accurate 3d point cloud registration. *EasyChair Preprint*, (2703), 2020.

[23] Damien Lefloch, Markus Kluge, Hamed Sarbolandi, Tim Weyrich, and Andreas Kolb. Comprehensive use of curvature for robust and accurate online surface reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2349–2365, 2017.

[24] Damien Lefloch, Tim Weyrich, and Andreas Kolb. Anisotropic point-based fusion. In *2015 18th International Conference on Information Fusion (Fusion)*, pages 2121–2128. IEEE, 2015.

[25] Robert J McCann. A convexity principle for interacting gases. *Advances in mathematics*, 128(1):153–179, 1997.

[26] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *2011 10th IEEE international symposium on mixed and augmented reality*, pages 127–136. IEEE, 2011.

[27] Matthias Nießner, Michael Zollhöfer, Shahram Izadi, and Marc Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics (ToG)*, 32(6):1–11, 2013.

[28] Ingram Olkin and Friedrich Pukelsheim. The distance between two random vectors with given dispersion matrices. *Linear Algebra and its Applications*, 48:257–263, 1982.

[29] Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE International Conference on Robotics and Automation*, pages 3400–3407. IEEE, 2011.

[30] Hanspeter Pfister, Matthias Zwicker, Jeroen Van Baar, and Markus Gross. Surfels: Surface elements as rendering primitives. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, pages 335–342, 2000.

[31] Thomas Schöps, Torsten Sattler, and Marc Pollefeys. Surfelmeshing: Online surfel-based mesh reconstruction. *IEEE transactions on pattern analysis and machine intelligence*, 42(10):2494–2507, 2019.

[32] Thomas Schöps, Johannes L Schonberger, Silvano Galliani, Torsten Sattler, Konrad Schindler, Marc Pollefeys, and Andreas Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3260–3269, 2017.

[33] Aleksandr Segal, Dirk Haehnel, and Sebastian Thrun. Generalized-icp. In *Robotics: science and systems*, volume 2, page 435. Seattle, WA, 2009.

[34] Guoping Wen Sergey Dorodnicov, Anders Grunnet-Jepsen. Projection, texture-mapping and occlusion with intel® realsense™ depth cameras. [EB/OL]. `https://dev.intelrealsense.com/docs/` Accessed January 27, 2021.

[35] Frank Steinbrücker, Jürgen Sturm, and Daniel Cremers. Volumetric 3d mapping in real-time on a cpu. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2021–2028. IEEE, 2014.

[36] Todor Stoyanov, Martin Magnusson, Henrik Andreasson, and Achim J Lilienthal. Fast and accurate scan registration through minimization of the distance between compact 3d ndt representations. *The International Journal of Robotics Research*, 31(12):1377–1393, 2012.

[37] Jörg Stückler and Sven Behnke. Multi-resolution surfel maps for efficient dense 3d modeling and tracking. *Journal of Visual Communication and Image Representation*, 25(1):137–147, 2014.

[38] Asuka Takatsu et al. Wasserstein geometry of gaussian measures. *Osaka Journal of Mathematics*, 48(4):1005–1026, 2011.

[39] Françoise Tisseur and Nicholas J Higham. Structured pseudospectra for polynomial eigenvalue problems, with applications. *SIAM Journal on Matrix Analysis and Applications*, 23(1):187–208, 2001.

[40] Cédric Villani. *Optimal transport: old and new*, volume 338. Springer Science & Business Media, 2008.

[41] Gary R Waissi. Network flows: Theory, algorithms, and applications, 1994.

[42] Thomas Whelan, Renato F Salas-Moreno, Ben Glocker, Andrew J Davison, and Stefan Leutenegger. Elasticfusion: Real-time dense slam and light source estimation. *The International Journal of Robotics Research*, 35(14):1697–1716, 2016.

[43] Qian-Yi Zhou and Vladlen Koltun. Dense scene reconstruction with points of interest. *ACM Transactions on Graphics (ToG)*, 32(4):1–8, 2013.

[44] Qian-Yi Zhou, Stephen Miller, and Vladlen Koltun. Elastic fragments for dense scene reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 473–480, 2013.

[45] Michael Zollhöfer, Patrick Stotko, Andreas Görlitz, Christian Theobalt, Matthias Nießner, Reinhard Klein, and Andreas Kolb. State of the art on 3d reconstruction with rgb-d cameras. In *Computer graphics forum*, volume 37, pages 625–652. Wiley Online Library, 2018.