Colored Point Cloud Registration Revisited

Jaesik Park Qian-Yi Zhou Vladlen Koltun
Intel Labs

Abstract

We present an algorithm for aligning two colored point clouds. The key idea is to optimize a joint photometric and geometric objective that locks the alignment along both the normal direction and the tangent plane. We extend a photometric objective for aligning RGB-D images to point clouds, by locally parameterizing the point cloud with a virtual camera. Experiments demonstrate that our algorithm is more accurate and more robust than prior point cloud registration algorithms, including those that utilize color information. We use the presented algorithms to enhance a state-of-the-art scene reconstruction system. The precision of the resulting system is demonstrated on real-world scenes with accurate ground-truth models.

1. Introduction

We are concerned with the following problem: given two roughly aligned three-dimensional point clouds, compute a tight alignment between them. This is a well-known problem in computer vision, computer graphics, and robotics. The problem is typically addressed with variants of the ICP algorithm [1, 3, 31]. The algorithm alternates between finding correspondences and optimizing an objective function that minimizes distances between corresponding points. A common failure mode of ICP is instability in the presence of smooth surfaces [14, 46]. The alignment can slip when geometric features do not sufficiently constrain the optimization.

This ambiguity can be alleviated if the points are associated with color. This is often the case. Modern depth cameras commonly produce pairs of depth and color images. Many industrial 3D scanners are also equipped with synchronized color cameras and provide software that associates color information with the 3D scans. Multi-view stereo pipelines reconstruct colored point clouds from image collections [8, 13, 39]. Considering color along with the geometry can increase the accuracy of point cloud registration.

The standard formulation for integrating color into geometric registration algorithms is to lift the alignment into a higher-dimensional space, parameterized by both position and color. Typically, correspondences are established in a four- or six-dimensional space rather than the physical three-dimensional space [21, 22, 27, 28]. This is an elegant approach, but it is liable to introducing erroneous correspondences between points that are distant but have similar color. These correspondences can pull away from the correct solution and prevent the method from establishing a maximally tight alignment.

In this work, we develop a different approach to aligning colored point clouds. Our approach establishes correspondences in the physical three-dimensional space, but defines a joint optimization objective that integrates both geometric and photometric terms. A key challenge is that color is only defined on discrete points in the three-dimensional space. To optimize a continuous joint objective, we need to define a continuous and differentiable photometric term, the gradient of which indicates how color varies as a function of position. This is challenging because unstructured point clouds do not provide a natural parameterization domain. We build on dense and direct formulations for RGB-D image alignment, which use the two-dimensional image plane as the parameterization domain [35, 25, 44, 40]. To define a photometric objective for point cloud alignment, we introduce a virtual image on the tangent plane of every point, which provides a local approximation to the implicit color variation. Using this construct, we generalize the photometric objectives used for RGB-D image alignment to unstructured point cloud alignment. The resulting photometric objective is integrated with a geometric objective defined using the same virtual image planes. This enables efficient joint photometric and geometric optimization for point cloud alignment. Our formulation unifies RGB-D image registration and colored point cloud registration. We show that our algorithm achieves tighter alignment than state-of-the-art registration algorithms, including those that use color information.

Our primary contribution is a new approach to colored point cloud registration. Beyond this, we make two supporting contributions. Since point cloud registration plays a central role in high-fidelity scene reconstruction, we have used the presented algorithms to enhance a state-of-the-art

2380-7504/17 $31.00 © 2017 IEEE
DOI 10.1109/ICCV.2017.25

scene reconstruction system [4]. To quantitatively evaluate reconstruction accuracy on real-world scenes, we have created a dataset of indoor scenes scanned with an industrial laser scanner. Experiments demonstrate that the enhanced pipeline produces significantly more accurate reconstructions.

2. Related Work

The ICP algorithm [1, 3, 31] has been a mainstay of geometric registration in both research and industry for many years. Its variants have been extensively studied [31, 33, 38]. Notably, point-to-plane ICP has been broadly adopted due to its fast convergence [3, 31]. ICP and other local refinement algorithms require a rough initial alignment as input. Such initial alignment can be obtained via global registration algorithms [18, 43, 47]. These global algorithms address a more difficult problem since they must establish correspondences with no initialization. While significant progress in global alignment has been made, the alignment produced by state-of-the-art global registration algorithms can often be improved by local refinement.

Most local registration algorithms that utilize color information lift the problem to a higher-dimensional space, which is used to establish correspondences [21, 22, 28, 27]. Godin et al. [15] use color to prune correspondences. Our approach is different in that we establish correspondences in the physical 3D space inhabited by the point clouds that are being registered, but optimize a joint photometric and geometric objective. A recent work [7] represents color information in a Gaussian mixture model. It is built upon a probabilistic registration algorithm [11] and is orders of magnitude slower than common ICP variants or our approach.

Many approaches to RGB-D image registration have been explored. Huhle et al. [20] and Henry et al. [17] combine image matching with geometric registration. Other approaches optimize a direct photometric objective defined densely over the images [35, 25]. Whelan et al. [40] introduce a joint optimization objective that combines the photometric objective and a point-to-plane ICP objective. We build on these works, specifically on the dense and direct formulations for RGB-D image registration. We review the photometric objective used for RGB-D image registration in Section 3 and then show that it can be generalized to unstructured point clouds. A key challenge that distinguishes point clouds from RGB-D images is the lack of a regular grid parameterization.

Dense reconstruction from RGB-D sequences has been extensively studied [29, 17, 24, 10, 4, 41]. Such reconstruction systems commonly have three key components: surface alignment (in the form of odometry and loop closure), global optimization, and surface extraction. We show that the colored point cloud registration approach presented in this paper can be used to increase the accuracy of the surface alignment step in a state-of-the-art reconstruction pipeline, significantly increasing the accuracy of the final reconstruction. To evaluate this quantitatively, we collect a dataset of RGB-D video sequences with dense ground-truth 3D models acquired with an industrial laser scanner. Many RGB-D datasets have been collected in prior work [37, 42, 19, 5, 12]. To our knowledge, none of them are accompanied by dense and accurate ground-truth 3D models of whole scenes. Synthetic datasets have been created for this purpose [16, 4]. We complement these efforts with real-world datasets.

3. RGB-D Image Alignment

In this section, we review the photometric objective for RGB-D image alignment [35, 25] and combine it with a geometric objective defined on the same image plane. This introduces notation and lays the groundwork for colored point cloud alignment, which will be presented in Section 4.

An RGB-D image is composed of a color image I and a depth image D registered to the same coordinate frame. For simplicity we use intensity images. Given a pair of RGB-D images (I_i, D_i) and (I_j, D_j) and an initial transformation $\mathbf{T}^0$ that roughly aligns (I_j, D_j) to (I_i, D_i), the goal is to find the optimal transformation that densely aligns the two RGB-D images.

A photometric objective E_I is formulated in terms of squared differences of intensities [35, 25]:

$$E_I(\mathbf{T}) = \sum_{\mathbf{x}} \big(I_i(\mathbf{x}') - I_j(\mathbf{x})\big)^2, \tag{1}$$

where $\mathbf{x} = (u, v)^\top$ is a pixel in (I_j, D_j) and $\mathbf{x}' = (u', v')^\top$ is the corresponding pixel in (I_i, D_i). The correspondence is built by converting the depth pixel $(\mathbf{x}, D_j(\mathbf{x}))$ to a 3D point in the camera space of (I_j, D_j), transforming it with $\mathbf{T}$, and projecting it onto the image plane of (I_i, D_i). Formally,

$$\mathbf{x}' = \mathbf{g}_{uv}\big(\mathbf{s}(\mathbf{h}(\mathbf{x}, D_j(\mathbf{x})), \mathbf{T})\big). \tag{2}$$

Here $\mathbf{h}$ is the conversion from a depth pixel to a 3D point in homogenous coordinates:

$$\mathbf{h}(u, v, d) = \Big(\frac{(u - c_x) \cdot d}{f_x}, \frac{(v - c_y) \cdot d}{f_y}, d, 1\Big)^\top, \tag{3}$$

where f_x and f_y are the focal lengths and (c_x, c_y) is the principal point. $\mathbf{s}$ is the following rigid transformation:

$$\mathbf{s}(\mathbf{h}, \mathbf{T}) = \mathbf{T}\mathbf{h}. \tag{4}$$

$\mathbf{g}$ is the inverse function of $\mathbf{h}$, which maps a 3D point to a depth pixel:

$$\mathbf{g}(s_x, s_y, s_z, 1) = \Big(\frac{s_x f_x}{s_z} + c_x, \frac{s_y f_y}{s_z} + c_y, s_z\Big)^\top. \tag{5}$$

The first two components of $\mathbf{g}$, denoted by $\mathbf{g}_{uv}$, form the corresponding pixel $\mathbf{x}'$ on the image plane of (I_i, D_i).

Similarly, we can define a geometric objective E_D that compares the depth of pixel $\mathbf{x}$ and $\mathbf{x}'$. We notice that direct comparison between depth values $D_i(\mathbf{x}')$ and $D_j(\mathbf{x})$ leads to incorrect results since the depth values are measured in different camera spaces. We therefore compare $D_i(\mathbf{x}')$ with the warped depth $\mathbf{g}_d$, which is the third component of $\mathbf{g}$ as defined in Equation 5:

$$E_D(\mathbf{T}) = \sum_{\mathbf{x}} \big(D_i(\mathbf{x}') - \mathbf{g}_d(\mathbf{s}(\mathbf{h}(\mathbf{x}, D_j(\mathbf{x})), \mathbf{T}))\big)^2. \quad (6)$$

It is important that both the photometric objective E_I and the geometric objective E_D are defined on the same parameterization domain. In the next section, we show that a change of parameterization domain enables generalization of these objectives to unstructured point clouds.

A joint photometric and geometric objective can be formulated by combining E_I and E_D:

$$E(\mathbf{T}) = (1-\sigma)E_I(\mathbf{T}) + \sigma E_D(\mathbf{T}), \quad (7)$$

where $\sigma \in [0, 1]$ is a constant weight that balances the two terms.

4. Colored Point Cloud Registration

In this section we generalize the joint optimization objective (7) to aligning colored point clouds.

4.1. Parameterization

Let $\mathbf{P}$ be a colored point cloud, and let $C(\mathbf{p})$ be a discrete function that retrieves the intensity of each point $\mathbf{p}$. In order to use color in optimization, we need to generalize $C(\mathbf{p})$ to a continuous function so that we can compute its gradient.

Conceptually, we introduce a virtual orthogonal camera for each point $\mathbf{p} \in \mathbf{P}$. It is configured to observe $\mathbf{p}$ along the normal $\mathbf{n}_\mathbf{p}$. The image plane of this virtual camera is the tangent plane at $\mathbf{p}$. It parameterizes a virtual image that can be represented as a continuous color function $C_\mathbf{p}(\mathbf{u})$, where $\mathbf{u}$ is a vector emanating from $\mathbf{p}$ along the tangent plane: $\mathbf{u} \cdot \mathbf{n}_\mathbf{p} = 0$. The function $C_\mathbf{p}(\mathbf{u})$ can be approximated by its first-order approximation:

$$C_\mathbf{p}(\mathbf{u}) \approx C(\mathbf{p}) + \mathbf{d}_\mathbf{p}{}^\top \mathbf{u}, \quad (8)$$

where $\mathbf{d}_\mathbf{p}$ is the gradient of $C_\mathbf{p}(\mathbf{u})$. The gradient is estimated by applying least-squares fitting to $\{C(\mathbf{p}') | \mathbf{p}' \in \mathcal{N}_\mathbf{p}\}$, where $\mathcal{N}_\mathbf{p}$ is the local neighborhood of $\mathbf{p}$.

Specifically, let $\mathbf{f}(\mathbf{s})$ be the function that projects a 3D point $\mathbf{s}$ to the tangent plane of $\mathbf{p}$:

$$\mathbf{f}(\mathbf{s}) = \mathbf{s} - \mathbf{n}_\mathbf{p}(\mathbf{s} - \mathbf{p})^\top \mathbf{n}_\mathbf{p}. \quad (9)$$

The least-squares fitting objective for computing $\mathbf{d}_\mathbf{p}$ is

$$\begin{aligned} L(\mathbf{d}_\mathbf{p}) &= \sum_{\mathbf{p}' \in \mathcal{N}_\mathbf{p}} \big(C_\mathbf{p}(\mathbf{f}(\mathbf{p}') - \mathbf{p}) - C(\mathbf{p}')\big)^2 \\ &\approx \sum_{\mathbf{p}' \in \mathcal{N}_\mathbf{p}} \big(C(\mathbf{p}) + \mathbf{d}_\mathbf{p}{}^\top (\mathbf{f}(\mathbf{p}') - \mathbf{p}) - C(\mathbf{p}')\big)^2, \end{aligned} \quad (10)$$

with the additional constraint $\mathbf{d}_\mathbf{p}^\top \mathbf{n}_\mathbf{p} = 0$. This is a linear least-squares problem and can be solved efficiently during preprocessing.

Similarly, we can assume that the virtual camera has a depth channel and define a continuous depth function $G_\mathbf{p}(\mathbf{u})$. Since its gradient at the origin is $\mathbf{0}$, the first-order approximation of $G_\mathbf{p}(\mathbf{u})$ is a constant function:

$$G_\mathbf{p}(\mathbf{u}) \approx (\mathbf{o}_\mathbf{p} - \mathbf{p})^\top \mathbf{n}_\mathbf{p}, \quad (11)$$

where $\mathbf{o}_\mathbf{p}$ is the origin of the virtual camera.

4.2. Objective

Let $\mathbf{P}$ and $\mathbf{Q}$ be two colored point clouds and let $\mathbf{T}^0$ be the coarse initial alignment. Our goal is to find the optimal transformation $\mathbf{T}$ that aligns $\mathbf{Q}$ to $\mathbf{P}$.

We formulate a joint optimization objective

$$E(\mathbf{T}) = (1-\sigma)E_C(\mathbf{T}) + \sigma E_G(\mathbf{T}), \quad (12)$$

where E_C and E_G are the photometric and geometric terms, respectively. $\sigma \in [0, 1]$ is a weight that balances the two terms.

The term E_C is defined by generalizing the photometric term E_I in Equation 1. The first change we make is to define residuals based on a correspondence set $\mathcal{K} = \{(\mathbf{p}, \mathbf{q})\}$ instead of the pixel set $\{\mathbf{x}\}$. Here $\mathcal{K}$ is created following the ICP algorithm: in each optimization iteration, $\mathcal{K}$ is recomputed as the set of correspondence pairs between $\mathbf{P}$ and $\mathbf{T}^k\mathbf{Q}$ that are within distance ε, where $\mathbf{T}^k$ is the current transformation.

To use the virtual camera introduced in Section 4.1, $\mathbf{q}$ is projected to a point $\mathbf{q}'$ on the tangent plane of $\mathbf{p}$:

$$\mathbf{q}' = \mathbf{f}(\mathbf{s}(\mathbf{q}, \mathbf{T})), \quad (13)$$

where $\mathbf{s}$ is the rigid transformation in Equation 4 and $\mathbf{f}$ is the projection function in Equation 9. Using the local color function $C_\mathbf{p}$ in (8) and the projected point $\mathbf{q}'$ in (13), E_C is defined as

$$E_C(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{K}} \big(C_\mathbf{p}(\mathbf{q}') - C(\mathbf{q})\big)^2. \quad (14)$$

Similarly, we generalize the geometric term E_D in Equation 6 to E_G:

$$E_G(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q}) \in \mathcal{K}} \big(G_\mathbf{p}(\mathbf{q}') - (\mathbf{o}_\mathbf{p} - \mathbf{s}(\mathbf{q}, \mathbf{T}))^\top \mathbf{n}_\mathbf{p}\big)^2. \quad (15)$$

Substituting $G_{\mathbf{p}}(\mathbf{q}')$ using (11), variable $\mathbf{o_p}$ is eliminated:

$$E_G(\mathbf{T}) = \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left((\mathbf{s}(\mathbf{q},\mathbf{T}) - \mathbf{p})^\top \mathbf{n_p}\right)^2. \quad (16)$$

This function is equivalent to the point-to-plane objective in the ICP algorithm [3, 31]. When only the geometric term is used ($\sigma = 1$), our algorithm reduces to point-to-plane ICP.

Putting everything together, the joint optimization objective (12) can be written as

$$\begin{aligned} E(\mathbf{T}) &= (1-\sigma) \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left(r_C^{(\mathbf{p},\mathbf{q})}(\mathbf{T})\right)^2 \\ &+ \sigma \sum_{(\mathbf{p},\mathbf{q})\in\mathcal{K}} \left(r_G^{(\mathbf{p},\mathbf{q})}(\mathbf{T})\right)^2, \end{aligned} \quad (17)$$

where $r_C^{(\mathbf{p},\mathbf{q})}$ and $r_G^{(\mathbf{p},\mathbf{q})}$ are the photometric and geometric residuals, respectively:

$$r_C^{(\mathbf{p},\mathbf{q})}(\mathbf{T}) = C_{\mathbf{p}}(\mathbf{f}(\mathbf{s}(\mathbf{q},\mathbf{T}))) - C(\mathbf{q}), \quad (18)$$

$$r_G^{(\mathbf{p},\mathbf{q})}(\mathbf{T}) = (\mathbf{s}(\mathbf{q},\mathbf{T}) - \mathbf{p})^\top \mathbf{n_p}. \quad (19)$$

4.3. Optimization

We minimize the nonlinear least-squares objective $E(\mathbf{T})$ using the Gauss-Newton method. In each iteration, we linearize $\mathbf{T}$ locally as a 6-vector $\xi = (\alpha, \beta, \gamma, a, b, c)$, which collates a rotational component ω and a translation $\mathbf{t}$. $\mathbf{T}$ is approximated by a linear function of ξ:

$$\mathbf{T} \approx \begin{pmatrix} 1 & -\gamma & \beta & a \\ \gamma & 1 & -\alpha & b \\ -\beta & \alpha & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \mathbf{T}^k, \quad (20)$$

where $\mathbf{T}^k$ is the transformation estimated in the last iteration. Following the Gauss-Newton method, we compute ξ by solving the linear system

$$\mathbf{J_r}^\top \mathbf{J_r} \xi = -\mathbf{J_r}^\top \mathbf{r}, \quad (21)$$

where $\mathbf{r}$ is the residual vector and $\mathbf{J_r}$ is its Jacobian, both evaluated at $\mathbf{T}^k$:

$$\mathbf{r} = \left[\sqrt{1-\sigma}\mathbf{r}_C; \sqrt{\sigma}\mathbf{r}_G\right], \quad (22)$$

$$\mathbf{r}_C = \left[r_C^{(\mathbf{p},\mathbf{q})}(\mathbf{T})|_{\mathbf{T}=\mathbf{T}^k}\right]_{(\mathbf{p},\mathbf{q})}, \quad (23)$$

$$\mathbf{r}_G = \left[r_G^{(\mathbf{p},\mathbf{q})}(\mathbf{T})|_{\mathbf{T}=\mathbf{T}^k}\right]_{(\mathbf{p},\mathbf{q})}, \quad (24)$$

$$\mathbf{J_r} = \left[\sqrt{1-\sigma}\mathbf{J}_{\mathbf{r}_C}; \sqrt{\sigma}\mathbf{J}_{\mathbf{r}_G}\right], \quad (25)$$

$$\mathbf{J}_{\mathbf{r}_C} = \left[\nabla r_C^{(\mathbf{p},\mathbf{q})}(\mathbf{T})|_{\mathbf{T}=\mathbf{T}^k}\right]_{(\mathbf{p},\mathbf{q})}, \quad (26)$$

$$\mathbf{J}_{\mathbf{r}_G} = \left[\nabla r_G^{(\mathbf{p},\mathbf{q})}(\mathbf{T})|_{\mathbf{T}=\mathbf{T}^k}\right]_{(\mathbf{p},\mathbf{q})}. \quad (27)$$

To evaluate the partial derivatives in Equations 26 and 27, we use (18) and (19) and apply the chain rule:

$$\nabla r_C^{(\mathbf{p},\mathbf{q})}(\mathbf{T}) = \frac{\partial}{\partial \xi_i}(C_{\mathbf{p}} \circ \mathbf{f} \circ \mathbf{s}) \quad (28)$$

$$= \nabla C_{\mathbf{p}}(\mathbf{f}) \mathbf{J}_{\mathbf{f}}(\mathbf{s}) \mathbf{J}_{\mathbf{s}}(\xi), \quad (29)$$

$$\nabla r_G^{(\mathbf{p},\mathbf{q})}(\mathbf{T}) = \mathbf{n_p}^\top \mathbf{J}_{\mathbf{s}}(\xi), \quad (30)$$

where $\nabla C_{\mathbf{p}} = \mathbf{d_p}$ is the precomputed gradient for each point $\mathbf{p} \in \mathbf{P}$, $\mathbf{J}_{\mathbf{f}}(\mathbf{s})$ is the Jacobian of $\mathbf{f}$ derived from (9), and $\mathbf{J}_{\mathbf{s}}$ is the Jacobian of $\mathbf{s}$ with respect to ξ, derived from (4) and (20).

In each iteration, we evaluate the residual $\mathbf{r}$ and the Jacobian $\mathbf{J_r}$ at $\mathbf{T}^k$, solve the linear system in (21), update $\mathbf{T}$ by applying the incremental transformation ξ to $\mathbf{T}^k$ using (20), and map the transformation into $SE(3)$. In the next iteration, we reparameterize $\mathbf{T}$ around $\mathbf{T}^{k+1}$ and repeat.

4.4. Coarse-to-fine processing

Objective (12) is non-convex and the optimization can get trapped in local minima. To alleviate this problem, we use a coarse-to-fine scheme. We build a point cloud pyramid by downsampling the input point cloud using a voxel grid with increasing voxel size. The downsampling algorithm approximates the points in each voxel with their centroid. Therefore, in terms of the optimization objective, a residual at a coarser level is the combination of several residuals at a finer level. The objective function at a coarser level is smoother and can guide the Gauss-Newton method to deeper minima. The optimization is performed at each level of the pyramid, from coarse to fine. The result of a coarse level initializes the optimization at the next level.

Algorithm 1 summarizes the presented algorithm for colored point cloud registration.

5. Scene Reconstruction

We have presented joint photometric and geometric optimization algorithms for aligning RGB-D images (Section 3) and colored point clouds (Section 4). The benefit of the joint objective is that it locks the alignment both along the tangent plane (via the photometric term) and along the normal direction (via the geometric term). Thus it is more robust and more accurate than using either objective alone. We now demonstrate the utility of these algorithms by using them to increase the accuracy and robustness of a state-of-the-art scene reconstruction system [4].

We build on the publicly available implementation of this system, replacing two key steps using the algorithms presented in this paper. The system takes an RGB-D sequence as input and proceeds through the following steps.

1. Build local geometric surfaces $\{\mathbf{P}_i\}$ (referred to as fragments) from short subsequences of the input RGB-D sequence;

Algorithm 1 Colored point cloud alignment

Input: Colored point cloud $\mathbf{P}$ and $\mathbf{Q}$, initial transformation $\mathbf{T}^0$
Output: Transformation $\mathbf{T}$ that aligns $\mathbf{Q}$ to $\mathbf{P}$

Build point cloud pyramids $\{\mathbf{P}^l\}$ and $\{\mathbf{Q}^l\}$
for $\mathbf{p} \in \mathbf{P}^l$ **do**
  Precompute $\mathbf{d_p}$ by minimizing (10)
  This defines function $C_\mathbf{p}$
$\mathbf{T} \leftarrow \mathbf{T}^0, L \leftarrow$ max_pyramid_level
for $l \in \{L, L-1, \cdots, 0\}$ **do** ▷ From coarsest to finest
  while not converged **do**
    $\mathbf{r} \leftarrow \mathbf{0}, \mathbf{J_r} \leftarrow \mathbf{0}$
    Compute the correspondence set $\mathcal{K}$
    for $(\mathbf{p}, \mathbf{q}) \in \mathcal{K}$ **do**
      Compute $r_C^{(\mathbf{p},\mathbf{q})}, r_G^{(\mathbf{p},\mathbf{q})}$ at $\mathbf{T}$ (Eq. 18,19)
      Compute $\nabla r_C^{(\mathbf{p},\mathbf{q})}, \nabla r_G^{(\mathbf{p},\mathbf{q})}$ at $\mathbf{T}$ (Eq. 29,30)
      Update $\mathbf{r}$ and $\mathbf{J_r}$ accordingly
    Solve linear system 21 to get ξ
    Update $\mathbf{T}$ using Equation 20, then map to $SE(3)$
Validate if $\mathbf{T}$ aligns $\mathbf{Q}$ to $\mathbf{P}$

2. Perform global registration and detect matchable fragment pairs $\{(\mathbf{P}_i, \mathbf{P}_j)\}$ by applying robust graph optimization over global registration results;
3. Tightly align matchable fragment pairs $\{(\mathbf{P}_i, \mathbf{P}_j)\}$ and build correspondence sets $\{\mathcal{K}_{i,j}\}$ between matchable fragments;
4. Optimize the fragment poses $\{\mathbf{T}_i\}$ and a camera calibration function $\mathcal{C}(\cdot)$ by minimizing an objective defined over the correspondences $\{\mathcal{K}_{i,j}\}$ [45];
5. Integrate RGB-D images to generate a mesh model for the scene.

We use the algorithms presented in Sections 3 and 4 to replace Steps 1 and 3.

Better fragment construction. We create a fragment from every $k = 100$ RGB-D images. Within each subsequence, we test every pair of RGB-D images to see if they can be aligned. The initial alignment is estimated by building correspondences between ORB features in the color images [30], pruning with the 5-point RANSAC algorithm [36], and computing a transformation that aligns the corresponding depth pixels [9]. We then optimize objective (7) to obtain a tight alignment. The algorithm is detailed in the supplement. The alignment results are treated as edges in a pose graph. Robust pose graph optimization is performed to estimate the camera pose of each RGB-D image [4]. With a truncated signed distance volume and a color volume, RGB-D images are integrated into fragments in the form of colored point clouds [6, 29, 40]. This replaces Step 1.

Better fragment alignment. For tight alignment of fragment pairs, we use the colored point cloud alignment algorithm developed in Section 4. This provides more accurate fragment alignment. In particular, the new algorithm is considerably more robust to slippage along flat surfaces, as shown in Figure 1. This replaces Step 3.

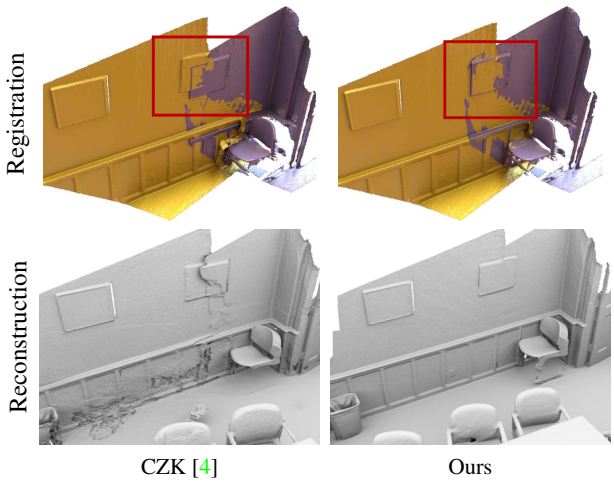

Figure 1. Left: failure of the ICP algorithm (top) leads to erroneous reconstruction (bottom). Right: our colored point cloud registration algorithm locks the alignment along the tangent plane as well as the normal direction (top), yielding an accurate scene model (bottom).

6. Dataset

To our knowledge, no publicly available RGB-D dataset provides dense ground-truth surface geometry across large-scale real-world scenes. To complement existing datasets, we have created ground-truth models of five complete indoor environments using a high-end laser scanner, and captured RGB-D video sequences of these scenes. This data enables quantitative evaluation of real-world scene reconstruction and will be made publicly available.

We scanned five scenes: Apartment, Bedroom, Boardroom, Lobby, and Loft. The size of each scene ranges from 21 to 86 square meters. Ground-truth data was collected using a FARO Focus 3D X330 HDR scanner. The scanner has an operating range of 0.6m to 330m. At a distance of 10 meters, its ranging accuracy is 0.1 millimeters. Each scene was scanned from multiple locations. The scans were merged using dedicated software provided by the manufacturer, which is used for range scan alignment in industrial applications.

In each scene, we captured a continuous RGB-D video sequence using an Asus Xtion Live camera. The lengths of the sequences range from 11 to 18 minutes. Each sequence thoroughly covers the respective scene. The RGB-D sequences can be used as input to scene reconstruction systems. The ground-truth models can be used to evaluate the accuracy of the results. The dataset is summarized in Table 1. The ground-truth models are visualized in the supplement.

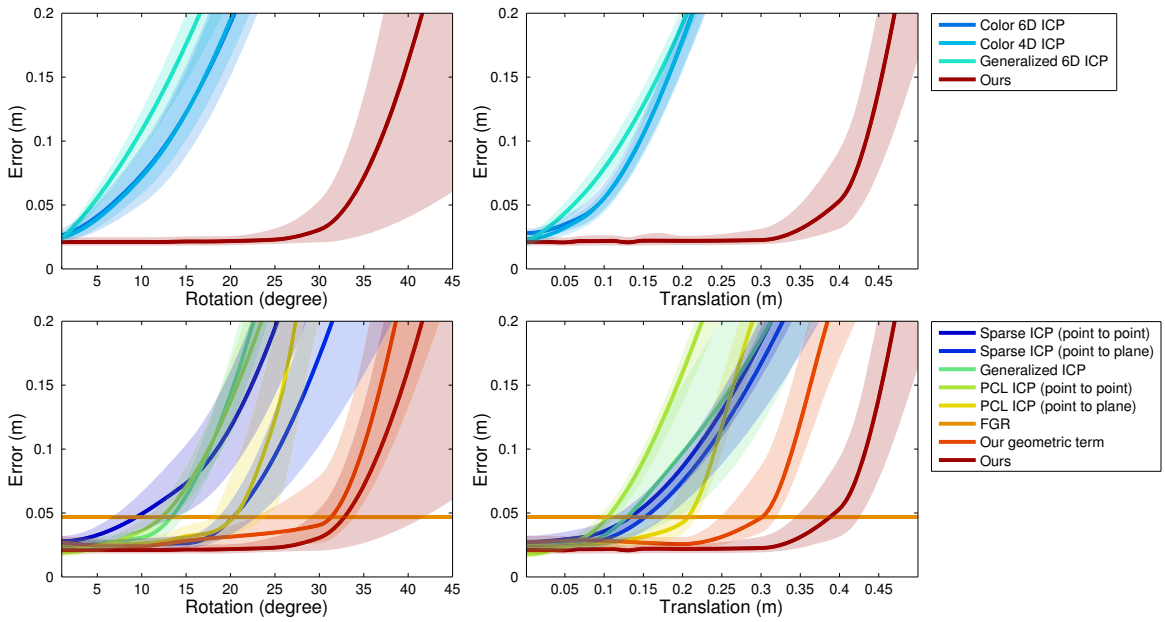

Figure 2. Evaluation of point cloud alignment on the TUM RGB-D dataset [37]. The presented algorithm is compared to prior algorithms that use color (top) and to algorithms that do not (bottom). The algorithms are initialized with transformations that are perturbed away from the true pose in the rotational component (left) and the translational component (right). The plot shows the median RMSE at convergence (bold curve) and the 40%-60% range of RMSE across trials (shaded region). Lower is better. Our algorithm outperforms all prior methods.

Name	Size (m^2)	# of laser pts.	# of RGBD frames
Apartment	69.17	18.7M	31.9K
Bedroom	21.01	10.9M	21.9K
Boardroom	60.90	17.4M	24.3K
Lobby	86.46	14.5M	20.0K
Loft	34.74	14.5M	25.3K

Table 1. Dataset statistics. Dense ground-truth surface models were acquired for five real-world scenes using an industrial laser scanner. The scenes were then scanned with an RGB-D camera.

7. Results

7.1. Colored point cloud registration

We begin by evaluating the colored point cloud registration algorithm presented in Section 4. We compare the algorithm to three alternative algorithms for colored point cloud alignment. The first two are ICP variants that embed the point clouds in a higher-dimensional space: the algorithm of Men et al. [28] (referred to as `Color 4D ICP`) and the algorithm of Johnson et al. [21] (referred to as `Color 6D ICP`). The third is the algorithm of Korn et al. [27], as implemented in the Point Cloud Library [32] (referred to as `Generalized 6D ICP`).

The first evaluation was performed on four sequences from the TUM RGB-D dataset: fr1/desk, fr1/desk2, fr1/room, and fr3/office [37]. We split the RGB-D sequences into segments and construct colored fragments using volumetric integration [6] with the ground truth camera poses provided in the dataset. This gives us colored point clouds with known relative poses. We tested registration algorithms on pairs of point clouds that overlap by at least 30%. To evaluate the accuracy of the different algorithms as a function of the initial pose, we initialized them in two regimes. In the first, the rotational component of the initial transformation was perturbed away from the true pose. In the second, the translational component was perturbed. The results are shown in Figure 2 (top). Our algorithm is more accurate when the initialization is near the true pose, and is much more robust to poor initialization.

For completeness, we also evaluate against registration algorithms that do not use color. The results are shown in Figure 2 (bottom). `PCL ICP` is the Point Cloud Library implementation of the ICP algorithm [32]. `Sparse ICP` is the algorithm of Bouaziz et al. [2]. We tested these algorithms with both point-to-point and point-to-plane distance measures [31]. `Generalized ICP` is a Point Cloud Library implementation of the algorithm of Segal et al. [34]. `FGR` is the state-of-the-art global registration algorithm of Zhou et al. [47]. `Our geometric term` refers to our results using only the geometric term ($\sigma = 1$). `Ours` refers to our results using the full optimization objective. The difference between `Ours` and `Our geometric term` shows the benefit of using color information. The optimal value of σ is found by grid search, as detailed in the supplement.

Our second evaluation was conducted on the Cathedral scene from the multimodal IMPART dataset [26]. The dataset provides seven colored LiDAR scans of a large out-

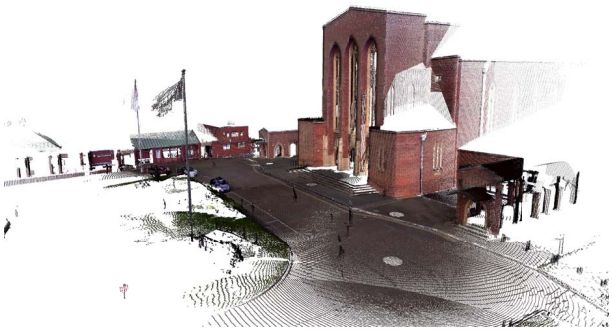

Figure 3. Seven colored LiDAR scans of the Cathedral scene [26] are aligned using our algorithm.

door scene, captured by a FARO laser scanner. The density of the points is not uniform. The results on this dataset are analogous to Figure 2 and are provided in the supplement. If the initial perturbation is more than 35 degrees in rotation or 4 meters in translation, the other methods begin to fail (RMSE>0.25m). In contrast, our algorithm aligns the scans tightly even when the initial perturbation is 40 degrees in rotation and 6 meters in translation.

The running time of the different algorithms is reported in Table 2. Runtime was measured on an Intel Core i7-5960X CPU with 8 parallelized threads. Our algorithm is faster than all other local registration algorithms. We hypothesize that the optimization converges faster due to the coarse-to-fine scheme and the photometric term.

Color 4D ICP [28]	3.64
Color 6D ICP [21]	3.66
Generalized 6D ICP [27]	16.11
Generalized ICP [34]	3.54
PCL ICP (point to point) [32]	2.43
PCL ICP (point to plane) [32]	1.77
Sparse ICP (point to point) [2]	8.96
Sparse ICP (point to plane) [2]	9.41
FGR* [47]	0.37
Ours	0.70

Table 2. Average running time (seconds). *FGR is a global registration algorithm that operates on fixed correspondences.

7.2. Scene reconstruction

We now evaluate the enhanced scene reconstruction system described in Section 5. Our first baseline is the system of Choi et al. [4] without our enhancements (referred to as `CZK`). Our second baseline is the `ElasticFusion` system of Whelan et al. [41], a state-of-the-art real-time pipeline. Note that neither our system nor `CZK` operate in real time, so `ElasticFusion` is at a disadvantage.

We begin with an evaluation on the existing SceneNN dataset [19]. This dataset does not provide ground-truth models, so our evaluation here is qualitative. We randomly sample two sequences from the dataset and reconstruct

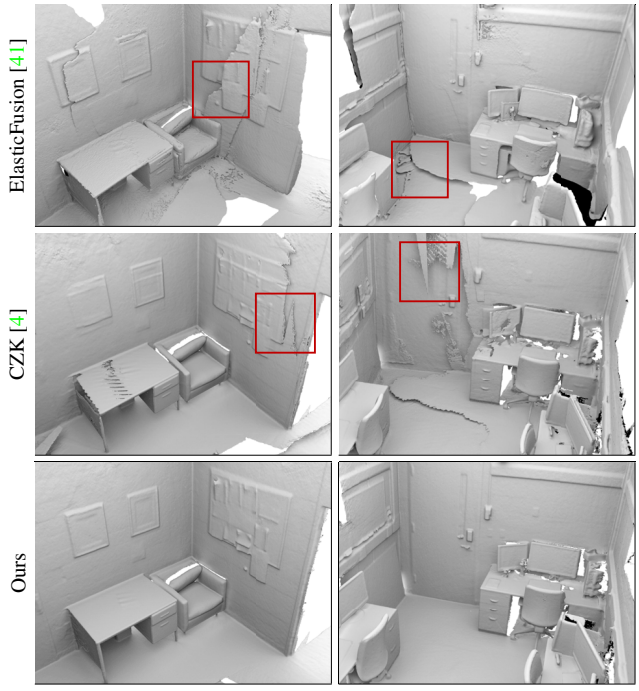

Figure 4. Reconstruction of two *randomly sampled* scenes from the SceneNN dataset [19]. Prior systems suffer from inaccurate surface alignment and produce broken geometry. Our system produces much cleaner results.

them with the three pipelines. The results are shown in Figure 4. For the purpose of visualization, Poisson surface reconstruction is applied to the output of `ElasticFusion` to create a mesh [23]. Our system produces the best qualitative results on both randomly sampled scenes.

We now perform a quantitative evaluation on the dataset presented in Section 6. Let the precision of a reconstructed model be the percentage of reconstructed points that have a ground-truth point within distance τ. Let the recall of a reconstructed model be the percentage of ground-truth points that have a reconstructed point within distance τ. We use $\tau = 20$ millimeters. Our primary measure is the F-score, the harmonic mean of precision and recall:

$$F = \frac{2 \cdot \text{precision} \cdot \text{recall}}{\text{precision} + \text{recall}}.$$

The F-score achieved by each system on each of the five scenes is reported in Table 3. Our system achieves an average F-score of 59.69%, versus 46.49% achieved by the `CZK` baseline. The reconstructions produced by our system are visualized in Figure 5.

8. Conclusion

We revisited the problem of colored point cloud registration and presented an algorithm that optimizes a joint

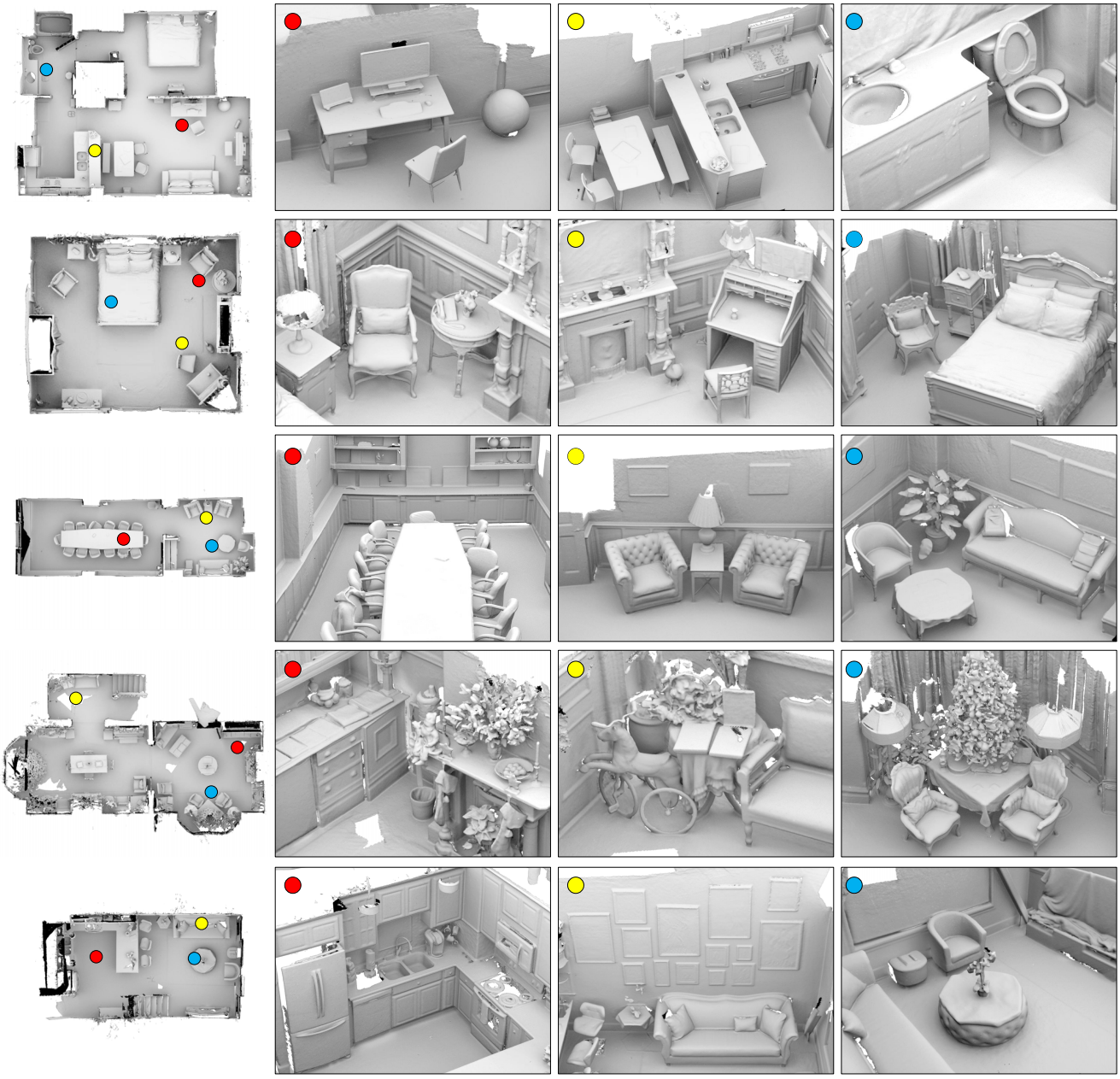

Figure 5. Scenes from the presented dataset, reconstructed using the presented system.

photometric and geometric objective. Our formulation unifies RGB-D image alignment and colored point cloud registration. Our approach outperforms prior registration algorithms. As an application, we used the presented approach to significantly improve the accuracy of a state-of-the-art scene reconstruction system. To quantitatively validate the results on real-world data, we created a dataset of five indoor scenes with accurate ground-truth models. Our dataset and reference implementations will be made publicly available.

	EF [41]	CZK [4]	Ours
Apartment	7.36	55.63	**61.68**
Bedroom	13.21	46.17	**75.25**
Boardroom	16.41	49.41	**50.43**
Lobby	7.35	35.37	**48.02**
Loft	30.60	45.88	**63.05**
Mean	14.99	46.49	**59.69**

Table 3. Results on the presented dataset. F-score in percentage points.

References

[1] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *PAMI*, 1992. 1, 2

[2] S. Bouaziz, A. Tagliasacchi, and M. Pauly. Sparse iterative closest point. In *Symposium on Geometry Processing*, 2013. 6, 7

[3] Y. Chen and G. G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3), 1992. 1, 2, 4

[4] S. Choi, Q.-Y. Zhou, and V. Koltun. Robust reconstruction of indoor scenes. In *CVPR*, 2015. 2, 4, 5, 7, 8

[5] S. Choi, Q.-Y. Zhou, S. Miller, and V. Koltun. A large dataset of object scans. *arXiv:1602.02481*, 2016. 2

[6] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996. 5, 6

[7] M. Danelljan, G. Meneghetti, F. Shahbaz Khan, and M. Felsberg. A probabilistic framework for color-based point set registration. In *CVPR*, 2016. 2

[8] A. Delaunoy and M. Pollefeys. Photometric bundle adjustment for dense multi-view 3D modeling. In *CVPR*, 2014. 1

[9] D. Eggert, A. Lorusso, and R. Fisher. Estimating 3-D rigid body transformations: A comparison of four major algorithms. *Machine Vision and Applications*, 9, 1997. 5

[10] F. Endres, J. Hess, J. Sturm, D. Cremers, and W. Burgard. 3-D mapping with an RGB-D camera. *IEEE Transactions on Robotics*, 30(1), 2014. 2

[11] G. D. Evangelidis, D. Kounades-Bastian, R. Horaud, and E. Z. Psarakis. A generative model for the joint registration of multiple point sets. In *ECCV*, 2014. 2

[12] M. Firman. RGBD datasets: Past, present and future. In *CVPR Workshops*, 2016. 2

[13] Y. Furukawa and C. Hernández. Multi-view stereo: A tutorial. *Foundations and Trends in Computer Graphics and Vision*, 9(1-2), 2015. 1

[14] N. Gelfand, S. Rusinkiewicz, L. Ikemoto, and M. Levoy. Geometrically stable sampling for the ICP algorithm. In *3-D Digital Imaging and Modeling*, 2003. 1

[15] G. Godin, D. Laurendeau, and R. Bergevin. A method for the registration of attributed range images. In *3-D Digital Imaging and Modeling*, 2001. 2

[16] A. Handa, T. Whelan, J. McDonald, and A. J. Davison. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In *ICRA*, 2014. 2

[17] P. Henry, M. Krainin, E. Herbst, X. Ren, and D. Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *International Journal of Robotics Research*, 31(5), 2012. 2

[18] D. Holz, A. E. Ichim, F. Tombari, R. B. Rusu, and S. Behnke. Registration with the point cloud library: A modular framework for aligning in 3-D. *IEEE Robotics and Automation Magazine*, 22(4), 2015. 2

[19] B. Hua, Q. Pham, D. T. Nguyen, M. Tran, L. Yu, and S. Yeung. SceneNN: A scene meshes dataset with annotations. In *3DV*, 2016. 2, 7

[20] B. Huhle, M. Magnusson, W. Straßer, and A. J. Lilienthal. Registration of colored 3D point clouds with a kernel-based extension to the normal distributions transform. In *ICRA*, 2008. 2

[21] A. E. Johnson and S. B. Kang. Registration and integration of textured 3D data. *Image and Vision Computing*, 1999. 1, 2, 6, 7

[22] J. H. Joung, K. H. An, J. W. Kang, M. J. Chung, and W. Yu. 3D environment reconstruction using modified color ICP algorithm by fusion of a camera and a 3D laser range finder. In *IROS*, 2009. 1, 2

[23] M. M. Kazhdan and H. Hoppe. Screened Poisson surface reconstruction. *ACM Transactions on Graphics*, 32(3), 2013. 7

[24] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *IROS*, 2013. 2

[25] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *ICRA*, 2013. 1, 2

[26] H. Kim and A. Hilton. Influence of colour and feature geometry on multi-modal 3D point clouds data registration. In *3DV*, 2014. 6, 7

[27] M. Korn, M. Holzkothen, and J. Pauli. Color supported Generalized-ICP. In *VISAPP*, 2014. 1, 2, 6, 7

[28] H. Men, B. Gebre, and K. Pochiraju. Color point cloud registration with 4D ICP algorithm. In *ICRA*, 2011. 1, 2, 6, 7

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. In *ISMAR*, 2011. 2, 5

[30] E. Rublee, V. Rabaud, K. Konolige, and G. R. Bradski. ORB: An efficient alternative to SIFT or SURF. In *ICCV*, 2011. 5

[31] S. Rusinkiewicz and M. Levoy. Efficient variants of the ICP algorithm. In *3-D Digital Imaging and Modeling*, 2001. 1, 2, 4, 6

[32] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *ICRA*, 2011. 6, 7

[33] J. Salvi, C. Matabosch, D. Fofi, and J. Forest. A review of recent range image registration methods with accuracy evaluation. *Image and Vision Computing*, 25(5), 2007. 2

[34] A. V. Segal, D. Haehnel, and S. Thrun. Generalized-ICP. In *RSS*, 2009. 6, 7

[35] F. Steinbrücker, J. Sturm, and D. Cremers. Real-time visual odometry from dense RGB-D images. In *ICCV Workshops*, 2011. 1, 2

[36] H. Stewénius, C. Engels, and D. Nistér. Recent developments on direct relative orientation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60, 2006. 5

[37] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *IROS*, 2012. 2, 6

[38] G. K. L. Tam, Z. Cheng, Y. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7), 2013. 2

[39] M. Waechter, N. Moehrle, and M. Goesele. Let there be color! Large-scale texturing of 3D reconstructions. In *ECCV*, 2014. 1

[40] T. Whelan, M. Kaess, H. Johannsson, M. F. Fallon, J. J. Leonard, and J. McDonald. Real-time large-scale dense RGB-D SLAM with volumetric fusion. *International Journal of Robotics Research*, 34(4-5), 2015. 1, 2, 5

[41] T. Whelan, R. F. Salas-Moreno, B. Glocker, A. J. Davison, and S. Leutenegger. ElasticFusion: Real-time dense SLAM and light source estimation. *International Journal of Robotics Research*, 35(14), 2016. 2, 7, 8

[42] J. Xiao, A. Owens, and A. Torralba. SUN3D: A database of big spaces reconstructed using SfM and object labels. In *ICCV*, 2013. 2

[43] J. Yang, H. Li, D. Campbell, and Y. Jia. Go-ICP: A globally optimal solution to 3D ICP point-set registration. *PAMI*, 38(11), 2016. 2

[44] Q.-Y. Zhou and V. Koltun. Color map optimization for 3D reconstruction with consumer depth cameras. In *SIGGRAPH*, 2014. 1

[45] Q.-Y. Zhou and V. Koltun. Simultaneous localization and calibration: Self-calibration of consumer depth cameras. In *CVPR*, 2014. 5

[46] Q.-Y. Zhou and V. Koltun. Depth camera tracking with contour cues. In *CVPR*, 2015. 1

[47] Q.-Y. Zhou, J. Park, and V. Koltun. Fast global registration. In *ECCV*, 2016. 2, 6, 7