# Robust Simultaneous 3D Registration via Rank Minimization

Diego Thomas
National Institute of Informatics
Tokyo, Japan / JFLI, CNRS
diego_thomas@nii.ac.jp

Yasuyuki Matsushita
Microsoft Research Asia
Beijing, China
yasumat@microsoft.com

Akihiro Sugimoto
National Institute of Informatics
Tokyo, Japan
sugimoto@nii.ac.jp

## Abstract

*We present a robust and accurate 3D registration method for a dense sequence of depth images taken from unknown viewpoints. Our method simultaneously estimates multiple extrinsic parameters of the depth images to obtain a registered full 3D model of the scanned scene. By arranging the depth measurements in a matrix form, we formulate the problem as a simultaneous estimation of multiple extrinsics and a low-rank matrix, which corresponds to the aligned depth images as well as a sparse error matrix. Unlike previous approaches that use sequential or heuristic global registration approaches, our solution method uses an advanced convex optimization technique for obtaining a robust solution via rank minimization. To achieve accurate computation, we develop a depth projection method that has minimum sensitivity to sampling by reading projected depth values in the input depth images. We demonstrate the effectiveness of the proposed method through extensive experiments and compare it with previous standard techniques.*

## 1. Introduction

Automatic 3D registration from a set of depth images has a long history yet is still a challenging problem in computer vision. Early approaches to 3D registration have been developed for range data that are acquired from sparse viewpoints because the task of depth scanning has been expensive. Recently, a significant effort has been made to develop inexpensive consumer depth cameras that allow the acquisition of depth images at a video rate, *e.g.*, Microsoft Kinect [1]. The video-rate depth cameras are becoming a commodity tool for depth measurement with reasonable accuracy. Such a depth camera brings a new problem setting for 3D registration; registering a *dense* set of depth images taken from continuously varying viewpoints. Since most of the existing registration techniques are not designed for dense sets of depth images, it is desirable to have a new technique for robustly, efficiently, and simultaneously registering multiple depth images taken from dense viewpoints.

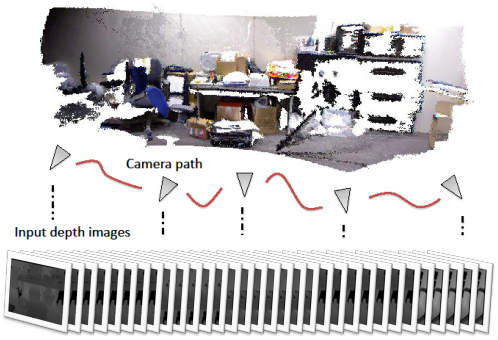



Figure 1: Illustration of the problem setting. A static scene is densely observed from a continuously varying viewpoint. From each viewpoint, a depth image is obtained. Our goal is to simultaneously register the observed depth images.

A wide class of 3D registration techniques focus on pairwise registration due to the heavy computational complexity of simultaneous registration, especially when the number of input depth images becomes large. With these techniques, however, the error of the independent pair-wise registration accumulates, which leads to significant global misalignment. Even though bundle adjustment or other heuristic global methods have been used for refining the registration result, fewer studies have been done on simultaneously registering multiple depth images.

In this work, we consider the situation depicted in Fig. 1. A static scene is densely scanned from an unknown continuous camera path, which gives a dense sequence of depth images (*e.g.*, a Kinect sensor recording VGA depth images at 30 fps). The camera intrinsics are assumed to be known and unchanged during the acquisition, while its extrinsics are unknown. The objective is then to align all the input depth images with each other simultaneously. Equivalently, we search for all the camera extrinsics that best align all depth images with a common cloud of points. The same setting is presented in KinectFusion [13], where video-rate 3D modeling results are shown. While their work focuses on efficient sequential registration for achieving real-time

scanning, we are interested in robust, and accurate simultaneous registration that is designed as an offline process.

To achieve the goal of simultaneous registration of a dense set of depth images, we develop a method based on a rank minimization strategy. We cast the problem of aligning a set of overlapping depth images as a problem of recovering a low-rank component from a high dimensional observation matrix. By stacking the depth images transformed to our reference coordinates using the extrinsics as a column vector in the high dimensional matrix, we formulate the problem as a simultaneous estimation of all the extrinsics and a low-rank matrix, which corresponds to the aligned depth images, as well as a sparse error matrix. The approach is motivated by previous work of Peng *et al*. [19], called RASL, which performs robust 2D image alignment from multiple images.

Estimating all the extrinsics that relate all depth images together cannot be simply achieved by applying RASL in a straightforward manner. Two major difficulties arise when simultaneously registering multiple depth images. Firstly, in contrast to 2D image alignment, we search for 2D to 3D transformations that align all depth images to a global point cloud. Accordingly, we have to formulate the relationship between the observation matrix we use and the 3D transformations that we want to estimate. Secondly, when registering depth images, a depth value itself changes with the camera pose while the 2D image case preserves the pixel intensity values. In other words, when the same scene point is observed, the depth value is dependent on the camera pose, while the intensity is not if a Lambertian scene is assumed. In this work, we explicitly formulate the problem of simultaneously registering multiple depth images and propose a solution method for it.

## 2. Related work

While successful attempts to simultaneously register multiple 2D images have been made, a large amount of 3D registration methods focus on pair-wise alignment. In general, pair-wise registration methods can be divided into two categories: (1) ones that use a sparse set of point correspondences, which we call sparse feature-based methods, and (2) ones that use a dense set of point correspondences, which we call dense correspondence methods.

Sparse feature-based methods, like SIFT [16] and its variants [4, 6, 25], are known to be fast and efficient, but with limited accuracy. In these techniques, sets of key-points are first detected, and a discriminative descriptor is attached to each key-point. Then, the detected key-points are matched across the range images to estimate the best transformation using various approaches, such as RANSAC [8], entropy maximization [15], or expectation maximization [12].

The Iterative Closest Point (ICP) method [5] is a conventional method that uses a dense set of point correspondences, where each point in one scan is matched with its closest point in the other scan to obtain dense correspondences. Various extensions have been developed [13, 24, 28, 30, 31] for improving the computational cost and accuracy. Different metrics such as point-to-point [5], or point-to-plane metrics [22] can be used to select the closest points, and various outlier rejection strategies, such as reciprocity or rigidity, are used to improve registration results. While the ICP-based methods are in general accurate, they also present some limitations. One of the major problems is that point matches are computed independently. As a result, the obtained cost function includes an accumulation of local errors, which often makes the function trapped into a local minima when the scene presents multiple symmetries.

For registering multiple depth images, there are pair-wise and simultaneous approaches. In general, pair-wise registration of multiple depth scans can be further divided into two categories: (1) methods that use a frame-to-frame approach [9, 28]; (2) methods that use a frame-to-global-model approach [13, 17, 27]. In the first category, Weise *et al*. [28] combine geometric and texture registration methods to align pairs of successive frames. Cui *et al*. [9] propose to combine registration and super-resolution methods. From an initial estimate of the alignment obtained using SLAM [10], super-resolution depth images are obtained [20, 21], which are then aligned using a non-rigid registration method with a mixture of Gaussians [14]. The final position of each camera can then be obtained by combining multiple pair-wise transformations. In the latter category, Izadi *et al*. [13] use a framework where live depth scans are registered to a global model of the 3D scene.

After aligning all input depth scans, various heuristic approaches are used to correct propagated errors. For example, Torsello *et al*. [26] develop an algorithm that uses projection of pair-wise alignments onto a reference frame and diffusion along a graph of adjacent nodes. Sharp *et al*. [23] propose to distribute the accumulated errors using an optimization strategy over the graph of neighboring views. When a loop in the input depth scans sequence is available, additional loop closure adjustment methods [29, 13] are used to correct the propagated errors, sometimes at the cost of global deformations in the final 3D model.

Extensions of the ICP algorithm [11, 18] have been proposed for simultaneous registration of multiple range images. However, handling multiple range images simultaneously dramatically increase the computational time. As pointed out in [11] it takes $O(nb_{im}^2 N\log(N))$ operations to find all point correspondences across $nb_{im}$ range images with $N$ points each. Such methods are thus unpractical when aligning a dense sequence of range images.

Our method is motivated by recent advances in the robust principal component analysis (RPCA) [7]. Based on

RPCA, Peng *et al*. [19] proposed a method called RASL, which performs robust simultaneous 2D alignment of multiple images. Motivated by these previous works, we develop a robust simultaneous 3D alignment method that does not require computing matches through all input images, but takes advantage of advanced convex optimization techniques. While the extension of the previous method to the 3D registration is not straightforward, we develop a solution method that effectively takes into account the 3D to 2D projections and handles issues that arise in 3D registration.

## 3. Proposed method

Our method takes a dense sequence of depth images recorded with unknown camera motion. We assume that the depth images of our input share a common overlapping region of the scene. We start with an initial guess of the camera extrinsics, and all depth measurements are projected to the reference camera coordinates. With this setting, our method finds extrinsics that align depth images in a simultaneous manner using a rank minimization strategy.

### 3.1. Notation

We represent the 6-DOF camera extrinsics estimated for the $k$-th depth image $d_k$ as a rigid transformation matrix:

$$T_k = \begin{bmatrix} R_k & \mathbf{t}_k \\ \mathbf{0}^\top & 1 \end{bmatrix} \in \mathbb{SE}_3$$

where the Euclidean group $\mathbb{SE}_3 := \{R, \mathbf{t} | R \in \mathbb{SO}_3, \mathbf{t} \in \mathbb{R}^3\}$. This maps the $k$-th local coordinates to the global coordinates. We will also use a single constant camera intrinsic matrix $K \in \mathbb{R}^{3\times3}$ that transforms points on the sensor plane to image pixels.

Let us denote by $\rho$ a function that performs projection onto the sensor plane of $\mathbf{w} = (x, y, z)^\top \in \mathbb{R}^3$ to obtain its projection on the image plane $\mathbf{w_p} = (u, v, 1)^\top \in \mathbb{R}^3$ by $\mathbf{w_p} = \rho(\mathbf{w})$. The function $\rho^{-1}$ performs the inverse projection: $\mathbf{w} = \rho^{-1}(\mathbf{w_p}, z)$. We will also use the function $H$ to denote a homogeneous operator $H(\mathbf{u}) := (\mathbf{u}^\top|1)^\top$, and the reverse dehomogenization operator $H^{-1}((\mathbf{u}^\top|1)^\top) := \mathbf{u}$. By a little bit of notation abuse, we will denote the $z$-component of a vector $\mathbf{w}$ as $z(\mathbf{w})$.

### 3.2. Problem formulation

Let us consider $n$ depth images that have a common overlapping region in the scene. We are interested in finding camera extrinsics that correspond to the depth measurements. A pixel $\mathbf{q} = (u, v)^\top$ has its corresponding 3D points $\mathbf{P}_k(u, v) = \rho^{-1}(K^{-1}H(\mathbf{q}), d_k(u, v))$ $(k = 1, \ldots, n)$ in the local coordinates, where $d_k$ denotes the $k$-th depth image. Each point $\mathbf{P}_k(u, v)$ is related to the corresponding visible scene point $\mathbf{X}_k(u, v) \in \mathbb{R}^3$ in the global coordinate system, by

$$\mathbf{P}_k(u, v) = H^{-1}(T_k^{-1}H(\mathbf{X}_k(u, v))).$$

The depth images $\{d_k\}_{k\in[1:n]}$ and the intrinsics matrix $K$ are given as input. We aim at estimating the extrinsics matrices $T_k$, or equivalently, estimating the aligned points $\mathbf{X}_k$.

For a reference viewpoint[1] *ref* with the extrinsics matrix $T_{ref} = I$, we can compute re-projected depth images $d_{(ref,k)}$ by projecting scene points $\mathbf{X}_k$ that are visible from the $k$-th view to the reference coordinates by

$$d_{(ref,k)}(H^{-1}(K\rho(\mathbf{P}_{(ref,k)}(u, v)))^\top) = z(\mathbf{P}_{(ref,k)}(u, v)),$$

where

$$\mathbf{P}_{(ref,k)}(u, v) = H^{-1}(T_{ref}^{-1}H(\mathbf{X}_k(u, v))) = \mathbf{X}_k(u, v).$$

When all points of $\mathbf{X}_k$ are well estimated, all the depth images $d_{(ref,k)}$ become well aligned up to occlusions, missing data, and data noise. By substituting $\mathbf{X}_k(u, v)$ with $H^{-1}(T_kH(\mathbf{P}_k(u, v)))$ we can then write:

$$\begin{aligned} d_{(ref,k)}(H^{-1}(K\rho(\mathbf{P}_{(ref,k)}(u, v)))^\top) = \\ z(H^{-1}(T_kH(\mathbf{P}_k(u, v))))). \end{aligned} \tag{1}$$

Therefore, $d_{(ref,k)}$ becomes a function of $T_k$. Now we use a compact representation $\tau_k$ for denoting the six extrinsic parameters of $T_k$, which are $\{R_x, R_y, R_z, T_x, T_y, T_z\}$. We denote $\tau = \tau_1, \ldots, \tau_n$, and $vec$ the vectorizing operator that only serializes pixels of the re-projected depth image that have a valid depth measurement in all the re-projected depth images. To compute this, we use a binary mask, which is computed as an intersection of valid entries of all the re-projected depth images. Therefore, the vectorizing operator depends on the current estimates of the extrinsics. We then denote $D(\tau) = [vec(d_{(ref,1)}, \tau), \ldots, vec(d_{(ref,n)}, \tau)]$ the matrix of all re-projected depth images in the vectorized form. Deriving a closed-form expression of the operator $vec$ is difficult; therefore, we use a procedural approach to compute this. As done in [19], the problem that we want to solve can then be re-written, with a trade-off parameter $\alpha$, as

$$\min_{A,E,\tau} (\text{rank}(A) + \alpha\|E\|_0) \;\; \text{s.t.} \;\; A + E = D(\tau),$$

where matrix $A$ represents the aligned depth images and matrix $E$ represents sparse errors or occlusions.

Since both rank minimization and $\ell_0$-norm minimization are NP-hard, in practice, we use[2] $\|\cdot\|_*$ instead of $\text{rank}(\cdot)$, and $\|\cdot\|_1$ for $\|\cdot\|_0$ as done by the Principle Component Pursuit method [7] because of the non-convexity of the original problem. In addition, to deal with the non-linearity of the constraint $A + E = D(\tau)$, we use a local linearization $D(\tau + \Delta\tau) = D(\tau) + J\Delta\tau$ with the Jacobian matrix $J$ of $D$ w.r.t. the transformations $\tau$, as done in [19]. This leads to

[1] The choice of the reference viewpoint is arbitrary. In our case, we choose the middle one $ref = \frac{n}{2}$.

[2] $\|A\|_* = \sum_{i=0}^{n-1} \sigma_i(A)$, where $\sigma_i(A)$ is the $i^{th}$ singular value of $A$.

the following convex optimization problem with unknowns $A$, $E$, and $\tau$:

$$\min_{A,E,\Delta\tau} \left( \|A\|_* + \frac{1}{\sqrt{m}} \|E\|_1 \right) \text{ s.t. } A+E = D(\tau)+J\Delta\tau, \quad (2)$$

where the weight $\alpha$ is set to $\frac{1}{\sqrt{m}}$, and $m$ is a number of lines in $D(\tau)$.

As shown in [19], the relaxation we used is the most appropriate, and the algorithm ensures convergence at a non-empty solution with a reasonable initialization.To efficiently solve Eq. (2), we use the adapted Augmented Lagrange Multiplier (ALM), as recommended by [19].

The main difficulty in depth image alignment arises when actually solving Eq. (2). Unlike the 2D image alignment case, the behavior of the function $D(\tau)$ becomes complex in 3D registration, and so is the problem of solving Eq. (2). This is (i) because the projection operator exhibited in Eq. (1) exists, (ii) because the depth of a point varies depending on the transformation $\tau$, and (iii) because the adjacency relationship between pixels in a depth image varies depending on the transformation $\tau$, due to occlusions. As a consequence, the Jacobian $J$ cannot be computed analytically but needs to be obtained procedurally. Therefore, for our problem, accurate and rapid computation of $J$ becomes fairly important. For this, we develop an efficient projection method to synthesize depth images from current estimates of $\tau$, which allows accurate and fast computation of the Jacobian $J$ using the finite difference method. We will describe this in the next section.

### 3.3. Projection of Depth Images

We develop an efficient projection method that has minimal sensitivity to the surface sampling. Our method takes depth images and extrinsics matrices as input and performs projection to a virtual camera image plane with respect to the reference depth image $d_{ref}$. The key idea is to compute depth values of pixels in the virtual camera image plane by interpolating depth values in the input depth images.

Let us assume a virtual camera $cam$. First, we generate the cloud of points $\mathbf{P}_{ref}$ from the reference depth image $d_{ref}$ using the intrinsics $K$. $\mathbf{P}_{ref}$ is then projected onto the virtual camera image plane to obtain the depth image $d^{cam}_{ref}$ ($d^{cam}_{ref}(H^{-1}(K_{cam}\rho(H^{-1}(T^{-1}_{cam}T_{ref}H(\mathbf{P}_{ref}(u,v)))))) = z(H^{-1}(T^{-1}_{cam}T_{ref}H(\mathbf{P}_{ref}(u,v)))))$. The cloud of points $\mathbf{P}^{cam}_{ref}$ is generated from $d^{cam}_{ref}$ using the intrinsics of $cam$.

For each depth image $d_k$, the projected image $d^{cam}_{(ref,k)}$ is computed as follows. For each pixel $(u,v)^\top$ of $d^{cam}_{ref}$ that has a valid depth, the pixel location $(u',v')$ of the point $\mathbf{P}^{cam}_{ref}(u,v)$ for the depth image $d_k$ is computed using the intrinsics and current extrinsics $K$ and $T_k$ ($(u',v') = H^{-1}(K\rho(H^{-1}(T^{-1}_k T_{cam}H(\mathbf{P}^{cam}_{ref}(u,v)))))^\top$). The corresponding depth $d_k(u',v')$ is estimated using bi-linear interpolation of the depth values in $d_k$. We finally compute the corresponding 3D point coordinates $\mathbf{p}' = H^{-1}(T_kH(\rho^{-1}(H^{-1}(K^{-1}(u',v',1)^\top), d_k(u',v'))))$ and transform it back to the local 3D coordinate system of $cam$ $\mathbf{p} = H^{-1}(T^{-1}_{cam}H(\mathbf{p}'))$ to obtain the depth of $d^{cam}_{(ref,k)}$ for the pixel $(u,v)^\top$ ($d^{cam}_{(ref,k)}(u,v) = z(\mathbf{p})$). The overall procedure is illustrated in Fig. 2. Note that if a pixel $(u,v)^\top$ of $d^{cam}_{ref}$ does not have a valid depth, then the pixel $(u,v)^\top$ of $d^{cam}_{(ref,k)}$ does not have a valid depth neither.

We use varying poses of virtual cameras for registering a set of depth images to avoid local minima that produce incoherent alignments in different viewpoints[3]. Typically, the virtual cameras are positioned at the front, left, and right of the reference camera. Note that the virtual camera's field of view needs to contain an overlapping area with all input depth images. The registration is then performed by iteratively aligning the depth images with respect to these virtual cameras.

The main advantage of computing the re-projected depth images in this way is that the accuracy of the projection is not limited by the sampling resolution of the depth images. On the other hand, one drawback of this approach is that points in the reference image that are not visible from other views will have wrong depth values after the projection. Nevertheless, this side effect is collectively handled by the error term $E$ in Eq. (2).

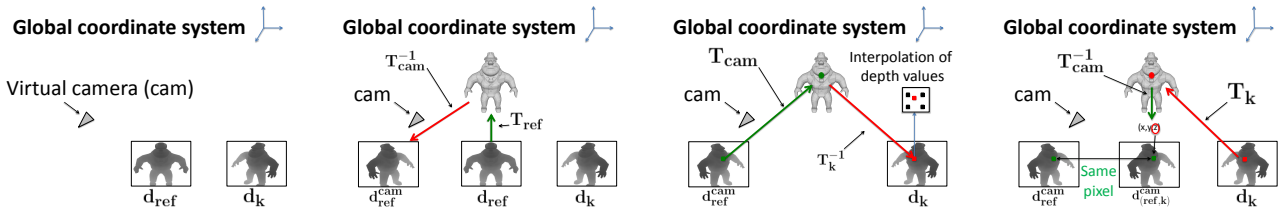
## 4. Experiments

To demonstrate the effectiveness of our proposed method, we evaluate our algorithm using both synthetic and real data. For comparison, we implemented the frame-to-global-model framework as proposed in [13]. For the extrinsics estimation step, we chose to use the GICP method as proposed in [22] in place of the linearized GICP as proposed in [13], because the GICP method is more accurate than its linearized version. We also compared our method with GICP used in the frame-to-frame framework, as proposed in [28]. We note that we used the GICP implementation provided by [2].

When the camera is taking drastically different positions during the scanning procedure, we use a sliding window with a fixed size through the input sequence of depth images. Namely, we define[4] $N_w$ as the size of the window (*i.e.* the maximum number of images that we register at once simultaneously) and initialize the process by simultaneously registering the first $N_w$ depth images. Then, the sequence is processed as follows: for each incoming depth image, the window is moved by one frame (namely, for the $i^{th}$ frame the window is composed of $\{d_k\}_{k\in[i-N_w+1,i]}$) and the current block of images are simultaneously registered. Finally,

[3] Such incoherences may be for example small shifts in the $z$ direction, which may be local minima when seen from the front, but clearly misaligned when seen from the left, or right side.

[4] We chose $N_w = 20$ in the experiments.




(a) Input. (b) Front projection of $d_{ref}$. (c) Projection of $d_{ref}^{cam}$ to $d_k$ and interpolation of depth values. (d) Affectation of back projected depth to $d_{(ref,k)}^{cam}$.

Figure 2: Illustration of our depth image projection operator.

for each input depth image, its final estimated extrinsic parameters are output as our result. Note that when a loop exists, standard refinement methods take several passes of the loop. In our method, such refinement is naturally performed on-the-fly by sliding the window, even when no loop exists.

The initial extrinsics were chosen as the identity matrix for the first window, and then we used previous estimates as the incoming frame's extrinsics. This initialization is reasonable as the viewpoint difference between successive frames is small when a video-rate camera is used.

In order to speed up the process, we created a pyramid of down-sampled images for each frame and applied our method with images from the highest tier. In the case where the registration fails because there are not enough points in the down sampled images, we used images from the next level of the pyramid. As expected, a gain in speed comes with loss in accuracy. Fig. 4 (a) shows the relationship between accuracy and time computation for different levels of the pyramid. We can also see that our method was stable when we used a pyramid of level 2 (i.e. $\frac{1}{4}$ times initial resolution). Depth image resolution became problematic when it reached $\frac{1}{16}$ times the input resolution. In this experiment, we used the synthetic data AL shown in Fig. 3, which is composed of 360 depth images. Note that we implemented our method on a 3.47 GHz PCU with 96.0 GB memory, in MATLAB and without any parallel computations.

### 4.1. Synthetic data

We used three synthetic data, AL, DRAGON, and TABLE, and created depth images by rendering from surrounding viewpoints. In all these experiments, we know the ground truth camera parameters. We added various levels of noise to the depth images to produce the final input. We first explain the evaluation metrics and then discuss the result.

**Registration error metric** We evaluate the registration error using the distance between the estimated position of points and the ground truth in the 3D world coordinates. The mean absolute error $\mathrm{MAE}(d_k)$ for the depth image $d_k$ is defined as

$$\mathrm{MAE}(d_k) = \frac{1}{Q}\sum_{q=1}^{Q}\left(\|\hat{T}_k\mathbf{P_k}(q) - T_k^*\mathbf{P_k}(q)\|_2\right),$$

where $\hat{T}_k$ is the estimated extrinsics[5], $T_k^*$ is the ground truth extrinsics, $\mathbf{P_k}$ is the cloud of points generated from the depth image $d_k$ and $Q$ is the number of points in $\mathbf{P_k}$.

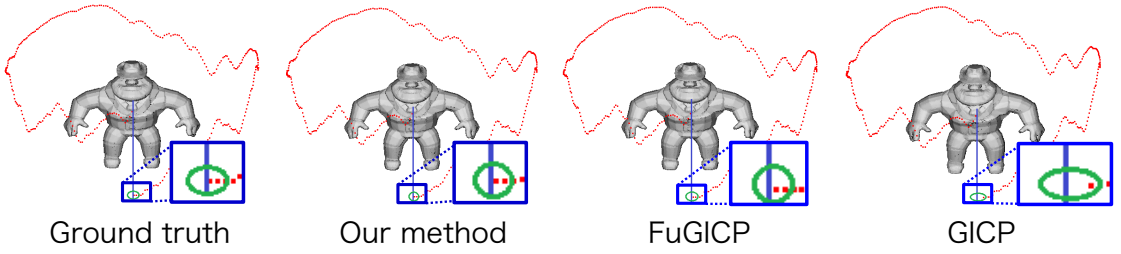
We use two error measures using $\mathrm{MAE}(d_k)$. One is the max error $e_{\mathrm{m}}$ of MAE, and the other is the average $e_{\mathrm{a}}$ of MAE defined respectively as

$$\begin{cases} e_{\mathrm{m}} &= \max_k\left(\mathrm{MAE}(d_k)\right), \\ e_{\mathrm{a}} &= \mathrm{mean}_k\left(\mathrm{MAE}(d_k)\right). \end{cases}$$

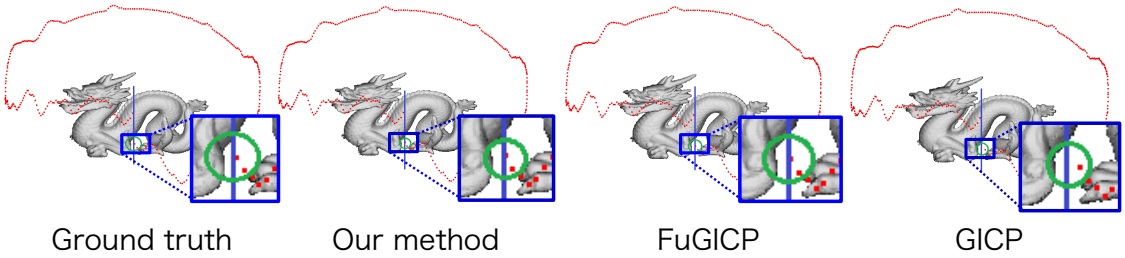
**Evaluation** For each synthetic scene, we evaluate our proposed method with uniform noise added in the depth images. Each depth image is perturbed with random noise in the interval $[-\alpha, \alpha]$, where $\alpha$ ranges between 0.0 and 10.0 [mm] with 1.0 [mm] interval. The typical noise level in the depth images acquired using a Kinect camera is about 3.0 [mm]. The depth images are generated from the point clouds. With depth images of VGA resolution, the resolution of the range images (i.e. the average distance between two neighboring points) was about 10.0 [mm]. For each data, we rendered 360 depth images as input. Namely, we rotated the camera around the object by 1.0 degree interval from 0 to 360 degrees. For AL and DRAGON we randomly perturbed the camera path to simulate a hand-held camera capturing.

Figure 3 shows qualitative registration results using the three synthetic scenes with our method (without using a pyramid), with GICP in the frame-to-global-model framework (called FuGICP) and with the frame-to-frame GICP (called GICP) in the case of $\alpha = 0.0$. These are all rendered as point clouds. Table 1 summarizes the quantitative results. While GICP in the frame-to-frame framework failed without accurate initialization, our method always obtained

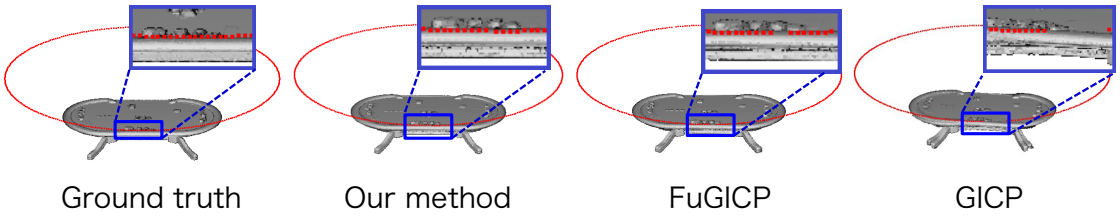
[5]Note that all extrinsics are transformed so that the extrinsics corresponding to the first depth image become the identity matrix.




(a) Dataset AL




(b) Dataset DRAGON




(c) Dataset TABLE

Figure 3: Registration results obtained with the three synthetic data, with zoom around the estimated position of the last camera. For data AL and DRAGON, the blue vertical line passing through the green circle is a marker for better visualization of the error. For the data TABLE, the marker is the first camera position.

the most accurate registration results, even when compared with FuGICP. In particular, Table 1 shows that the registration error obtained with our method is always below 10.0 [mm]. This means that the maximum deviation for all points in all depth images compared to their ground truth positions is below the resolution of the range data. The quantitative results thus validate the accuracy of our proposed registration method. From Table 1, we can also see that the maximum errors $e_m$ are close to the mean errors $e_n$. This means that there are no huge errors throughout the sequences of depth images, and thus the accuracy of our method is less affected by changes in viewpoint.

Note that our method is less affected to changes in shape of the object compared with FuGICP. This is because we use a global evaluation metric accounting for multiple depth images in contrast to the pair-wise local evaluation metric used in GICP.

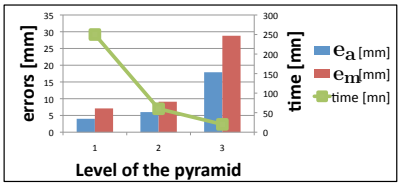
Figure 4 shows the results obtained with our method and FuGICP for the three synthetic scenes and for various noise levels added to the depth values. For time reasons, we chose to apply our method with a pyramid of level 2 (which explains why the errors for $\alpha = 0$ are slightly degraded compared with Table 1). For each noise level, all methods were run 10 times under the same conditions except for the noise distributions, whose registration errors $e_{\mathrm{m}}$ and $e_{\mathrm{a}}$ are shown in the plots. From Figure 4, we can see that our method achieves robustness against data noise while FuGICP is degraded as noise increases. In particular, adding noise in the data TABLE had dramatic effects for FuGICP.

### 4.2. Real data

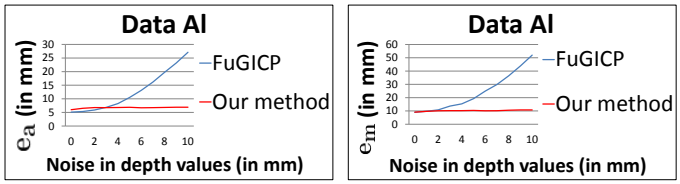
We also perform experiments using real data recorded by Microsoft Kinect. This sensor can record 30 depth images per second with a resolution up to $640 \times 480$. We used the RGBDemo software [3] to capture the depth and color images. Because we had to save the live data, the frame

Table 1: Registration errors $e_{\mathrm{m}}$, $e_{\mathrm{a}}$, and $e_{\mathrm{d}}$ for three synthetic scenes [mm] in the case of $\sigma^2 = 0$.

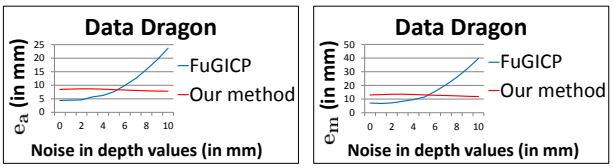
| | AL | | | DRAGON | | | TABLE | | |
|---|---|---|---|---|---|---|---|---|---|
| | Ours | FuGICP | GICP | Ours | FuGICP | GICP | Ours | FuGICP | GICP |
| Max error $e_{\mathrm{m}}$ | **7.1** | 9.0 | 39.4 | **3.7** | 7.1 | 14.4 | **6.9** | 19.8 | 97.4 |
| Average error $e_{\mathrm{a}}$ | **4.0** | 5.1 | 19.6 | **2.5** | 4.4 | 9.9 | **4.4** | 8.2 | 49.9 |



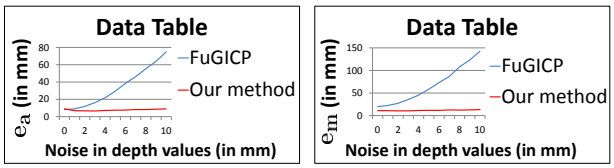

(a) Time versus acuracy.




(d) Dataset AL




(g) Dataset DRAGON



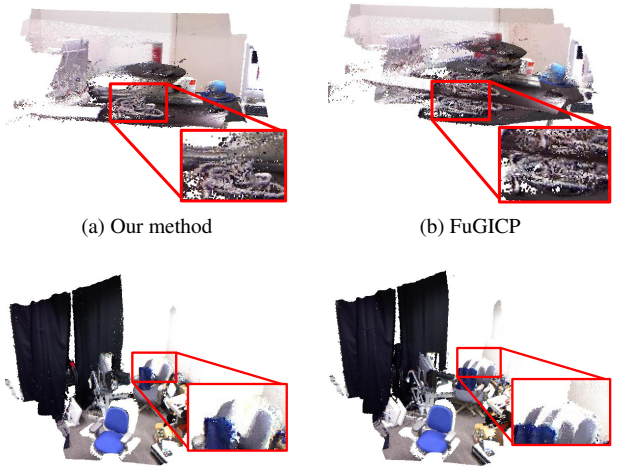

(j) Dataset TABLE

Figure 4: Plots of average registration errors. Max error $e_{\mathrm{m}}$, average error $e_{\mathrm{a}}$ w.r.t. different noise levels using our method and FuGICP are plotted.

rate dropped down to 10 images per second with some lags in the sequence. For comparison, we used FuGICP, which obtained better results than GICP with the synthetic data.

Figure 5 shows the results using two different real scenes with our method and FuGICP. The first scene consists of 300 images and the second scene consists of 200 images. While the FuGICP method breaks down in the first and second examples, our method can accurately register the depth images. The main reason for this is that even though GICP uses dense point correspondences between pairs of depth data, each correspondence is obtained independently. As a consequence, parts of the scene that are weakly supported by the global structure of the scene (feature-less parts, *e.g.*, walls or tables) tend to affect the registration result. In contrast, our method uses a global measure to ascertain the accuracy of the alignment (*i.e.*, the rank of the stacked matrix). Accordingly, local patches that lack discriminative geometric features have little impact on the registration result. In addition, missing points are dealt with during the projection process while occlusions and sparse depth measurement errors are modeled in the optimization problem, which leads to a robust registration method.

Note that in these experiments, the global camera motion amplitude was small and that it was the most advantageous situation for FuGICP. By doing so, the volumetric model



(a) Our method (b) FuGICP

(c) Our method (d) FuGICP

Figure 5: The results obtained with real data.

used in FuGICP could be restricted to a small part of the 3D world, which allowed fine discretization of the 3D scene. We chose the most challenging conditions to evaluate the gain in accuracy of our method compared with FuGICP.

## 5. Conclusion

We introduced a robust simultaneous 3D registration method for dense sets of depth images, based on a rank minimization strategy. By arranging the depth measurements in a matrix form, we formulated the problem as a simultaneous estimation of all the extrinsics and a low-rank matrix, which corresponds to the aligned depth images, as well as a sparse error matrix that models corruptions, such as occlusions. To solve the matrix decomposition problem, we used an advanced convex optimization technique that robustly finds a solution that is unaffected by sparse errors. We developed an efficient projection method that has minimal sensitivity to the surface sampling to achieve efficient optimization. Our extensive experiments using synthetic and real data demonstrated the robustness and accuracy of our proposed method for simultaneous registration of multiple depth images.

Since the rank of all aligned depth images is 1 in theory, enforcing this condition and minimizing the $L_1$ norm of the residuals may also be effective. Investigation in this direction is left for future work.

## References


[1] Microsof kinect: http://www.xbox.com/en-US/kinect. 1

[2] ICP PCL code: http://pointclouds.org/. 4

[3] RGBDemo: http://nicolas.burrus.name/index.php. 6

[4] H. Bay, T. Tuytelaars, and L. V. Gool. Surf: Speeded up robust features. *Proc. of ECCV'06*, pages 404–417, 2006. 2

[5] P. J. Besl and N. D. McKay. A method for registration of 3-D shapes. *IEEE Trans. on PAMI*, 14(2):239–256, 1992. 2

[6] N. Brusco, M. Andretto, A. Giorgi, and G. M. Cortelazzo. 3d registration by textured spin-images. *Proc. of 3DIM'05*, pages 262–269, 2005. 2

[7] R. Candes, X. Li, Y. Ma, and J. Wright. Robust principal component analysis? *Proc. of CoRR'09*, 2009. 2, 3

[8] S. Choi, T. Kim, and Z. Yu. Performance evaluation of ransac family. *Proc. of BMVC'09*, 2009. 2

[9] Y. Cui, S. Schuon, D. Chan, S. Thrun, and C. Theobalt. 3d shape scanning with a time-of-flight camera. *Proc. of CVPR'10*, 2010. 2

[10] A. Davison, I. Reid, N. Molton, and O. Stasse. Monoslam: Real-time single camera slam. *IEEE Trans. on PAMI*, pages 1052–1067, 2007. 2

[11] D. W. Eggerta, A. W. Fitzgibbon, and R. B. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad model. 1996. 2

[12] J. Herman, D. Smeets, D. Vandermeulen, and P. Suetens. Robust point set registration using em-icp with information-theoretically optimal outlier handling. *Proc. of CVPR'11*, pages 2465–2472, 2011. 2

[13] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, and A. Fitzgibbon. Kinectfusion: Real-time 3d reconstruction and interaction using a moving depth camera. *Proc. of ACM Symposium on User Interface Software and Technology*, 2011. 1, 2, 4

[14] B. Jian and B. C. Vemuri. A robust algorithm for point set registration using mixture of gaussian. *Proc. of ICCV'05*, 2:1246–1251, 2005. 2

[15] Y. Liu. Automatic range image registration in the markov chain. *IEEE Trans. on PAMI*, 32(1):12–29, 2010. 2

[16] D. G. Lowe. Object recognition from local scale-invariant features. *Proc. of ICCV'99*, 2:1150–1157, 1999. 2

[17] P. J. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. *Proc. of SMA'97*, 1997. 2

[18] K. Nishino and K. Ikeuchi. Robust simultaneous registration of multiple range images. *Proc. of ACCV'02*, pages 454–461, 2002. 2

[19] Y. Peng, A. Ganesh, J. Wright, W. Xu, and Y. Ma. Rasl: Robust alignment by sparse and low-rank decomposition for linearly correlated images. *IEEE Trans. on PAMI*, 2011. 2, 3, 4

[20] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. High-quality scanning using time-of-flight depth superresolution. *Proc. of CVPRW'08*, pages 1–7, 2008. 2

[21] S. Schuon, C. Theobalt, J. Davis, and S. Thrun. Lidarboost: Depth superresolution for tof 3d shape scanning. *Proc. of CVPR'09*, 2009. 2

[22] A. Segal, D. Haehnel, and S. Thrun. Generalized-icp. *Robotics: Science and Systems*, 2009. 2, 4

[23] G. C. Sharp, S. W. Lee, and D. K. Wehe. Multiview registration of 3d scenes by minimizing error between coordinate frames. *IEEE Trans. on PAMI*, 26(8):1037–1050, 2004. 2

[24] D. Thomas and A. Sugimoto. Robustly registering range images using local distribution of albedo. *Computer Vision and Image Understanding*, 28(4):649–667, 2011. 2

[25] E. Tola, V. Lepetit, and P. Fua. Daisy: an efficient dense descriptor applied to wide baseline stereo. *IEEE Trans. on PAMI*, 2009. 2

[26] A. Torsello, E. Rodola, and A. Albarelli. Multiview registration via graph diffusion of dual quaternions. *Proc. of CVPR'11*, pages 2441–2448, 2011. 2

[27] Y. Watanabe, T. Komuro, and M. Ishikawa. High-resolution shape reconstruction from multiple range images based on simultaneous estimation of surface and motion. *Proc. of ICCV'09)*, pages 1787–1794, 2009. 2

[28] T. Weise, B. Leibe, and L. V. Gool. Accurate and robust registration for in-hand modeling. *Proc. of CVPR'08*, pages 1–8, 2008. 2, 4

[29] T. Weise, T. Wismer, B. Leibe, and L. V. Gool. In-hand scanning with online loop closure. *Proc. of 3DIM'09*, pages 1630–1637, 2009. 2

[30] Z. Xie, S. Xu, and X. Li. A high-accuracy method for fine registration of overlapping point of clouds. *Image and Vision Computing*, 28(4):563–570, 2010. 2

[31] H. Zhang, O. Hall-Holt, and A. Kaufman. Range image registration via probability fields. *Proc. of CGI'04*, pages 546–552, 2004. 2