Supplementary Material
Inverse Composition Discriminative Optimization for Point Cloud Registration

Jayakorn Vongkulbhisal[1,2], Beñat Irastorza Ugalde[2], Fernando De la Torre[2,3], João P. Costeira[1]
[1]ISR - IST, Universidade de Lisboa, Lisboa, Portugal
[2]Carnegie Mellon University, Pittsburgh, PA, USA
[3]Facebook Inc., Menlo Park, CA, USA
jvongkul@andrew.cmu.edu, birastor@andrew.cmu.edu, ftorre@cs.cmu.edu, jpc@isr.ist.utl.pt

Contents

This document is organized into 4 sections.

- Sec. 1: Elaboration of the derivation of $\mathbf{h}$
- Sec. 2: Additional Experiments
- Sec. 3: Analysis on the Effect of Parameters
- Sec. 4: Visualization of Results

1. Elaboration of the derivation of $\mathbf{h}$

Sec. 3.4 in the main paper describes the derivation of the feature function $\mathbf{h}$. In this section, we provide an elaboration of the derivation. Here, we refer to the equations from the main paper with an *m*, *e.g.*, (15m).

The goal of Sec. 3.4 is to transform the update $\Delta\mathbf{x}$ into a form that we can learn with (13m). The main idea is to set $\Delta\mathbf{x}$ as the negative gradient of $J(\tilde{\mathbf{x}})$ in (16m), then factorize it into two terms: $\mathbf{D}$ which contains the unknown, *i.e.*, the information about ψ; and $\mathbf{h}$ which contains all known information, *i.e.*, the information about the point clouds at the current estimate. We elaborate this step, *i.e.*, from (17m) to (23m), as follows.

First, let us recall (17m), where we look at the update due to a single term (i, j). Here, we refer to $\|\mathbf{g}_{ij}(\mathbf{0}_6; \mathbf{x})\|$ as $z_{ij} \in \mathbb{R}$:

$$\Delta\mathbf{x}_{ij} = -\mathbf{w}_{ij}\varphi(z_{ij}). \tag{1}$$

Notice that $\varphi(z_{ij})$ is a scalar. Since we know the values of z_{ij} and $\mathbf{w}_{ij}$ but not φ (as it depends on the unknown ψ), we need to separate them. This is done by expressing $\varphi(z_{ij})$ as the inner product between φ and Dirac delta function δ:

$$\Delta\mathbf{x}_{ij} = -\mathbf{w}_{ij}\int_V \varphi(v)\delta(v - z_{ij})dv, \tag{2}$$

which is (18m). Looking at a single element l of $\Delta\mathbf{x}_{ij}$, we arrive at (19m):

$$[\Delta\mathbf{x}_{ij}]_l = -\int_V \varphi(v)\left([\mathbf{w}_{ij}]_l\delta(v - z_{ij})\right)dv. \tag{3}$$

We now see that the unknown φ is separated from the known $[\mathbf{w}_{ij}]_l$ and z_{ij}. In addition, the integral (3) can be thought of as an inner product between $-\varphi(v)$ and $[\mathbf{w}_{ij}]_l\delta(v - z_{ij})$ over the space $V = \mathbb{R}$, which is similar to $[\mathbf{Dh}]_l$, the inner product between $\mathbf{h}$ and row l of $\mathbf{D}$. However, (3) is an inner product between functions over a continuous domain, unlike $[\mathbf{Dh}]_l$ which is an inner product between vectors. So in order to represent (3) as an inner product between vectors, we discretize φ and δ into vector forms: φ is discretized into a vector $\boldsymbol{\varphi}$; and δ to a discretized delta function, which is similar to a standard basis vector $\mathbf{e}_\beta$. However, we cannot discretize the whole space $V = \mathbb{R}$. Thus, we only discretize the range $[0, r] \subset V$ into q boxes (note that $z_{ij} = \|\mathbf{g}_{ij}(\mathbf{0}_6; \mathbf{x})\|$ can only be a non-negative number, thus we do not need to discretize the negative side of V). In order to compute the index of the vector that z_{ij} discretizes to, we define a discretization function $\gamma : \mathbb{R} \to \{0, 1, \dots, q\}$ as

$$\gamma(y) = \begin{cases} \lceil y \rceil, & y \in [0, q] \\ 0 & \text{otherwise,} \end{cases} \tag{4}$$

where $\lceil\cdot\rceil$ rounds up a number. With γ, we can express the discretized (3) as (20m):

$$[\Delta\mathbf{x}_{ij}]_l \approx -\boldsymbol{\varphi}^\top[\mathbf{w}_{ij}]_l\mathbf{e}_{\gamma(\frac{q}{r}z_{ij})}, \tag{5}$$

where we define $\mathbf{e}_0 = \mathbf{0}_q$. Note that since any z_{ij} outside $(0, r]$ will have $\gamma(\frac{q}{r}z_{ij}) = 0$, which leads to $\mathbf{e}_{\gamma(\frac{q}{r}z_{ij})} = \mathbf{0}_q$, those z_{ij} will be disregarded from the computation of $\mathbf{h}$.

From here, we assemble $[\Delta\mathbf{x}_{ij}]_l$ over i, j, l back to $\Delta\mathbf{x}$:

$$\Delta\mathbf{x} = \sum_{i=1}^{N_M}\sum_{j=1}^{N_S}\Delta\mathbf{x}_{ij} \tag{6}$$

$$\approx \sum_{i=1}^{N_M}\sum_{j=1}^{N_S}\begin{bmatrix} -\boldsymbol{\varphi}^\top[\mathbf{w}_{ij}]_1\mathbf{e}_{\gamma(\frac{q}{r}z_{ij})} \\ \vdots \\ -\boldsymbol{\varphi}^\top[\mathbf{w}_{ij}]_6\mathbf{e}_{\gamma(\frac{q}{r}z_{ij})} \end{bmatrix} \tag{7}$$

$$= \sum_{i=1}^{N_M}\sum_{j=1}^{N_S} \begin{bmatrix} -\boldsymbol{\varphi}^\top & \cdots & \mathbf{0}_q^\top \\ \vdots & \ddots & \vdots \\ \mathbf{0}_q^\top & \cdots & -\boldsymbol{\varphi}^\top \end{bmatrix} \begin{bmatrix} [\mathbf{w}_{ij}]_1 \mathbf{e}_{\gamma(\frac{q}{r} z_{ij})} \\ \vdots \\ [\mathbf{w}_{ij}]_6 \mathbf{e}_{\gamma(\frac{q}{r} z_{ij})} \end{bmatrix} \tag{8}$$

$$= \left(-\mathbf{I}_6 \otimes \boldsymbol{\varphi}^\top\right) \left(\sum_{i=1}^{N_M}\sum_{j=1}^{N_S} \bigoplus_{l=1}^{6} [\mathbf{w}_{ij}]_l \mathbf{e}_{\gamma(\frac{q}{r} z_{ij})} \right) \tag{9}$$

$$= \mathbf{D}\mathbf{h}_{r,q}(\mathbf{x};\theta), \tag{10}$$

where $\otimes$ is the Kronecker product, and $\bigoplus$ is vector concatenation. This expression allows us to use $\mathbf{h}$ (derived from the known information about the point clouds) as a feature for learning the map $\mathbf{D}$ (derived from the unknown φ) such that $\mathbf{D}\mathbf{h}$ approximates the update step $\Delta\mathbf{x}$ (the negative gradient of the unknown cost J).

2. Additional Experiments

In this section, we provide an additional results on the ETH laser registration dataset [1]. In the main text, we provide the results for consecutive scans. Here, we consider registration for pairs of scans that are 2 steps apart (a total of 518 pairs). The cumulative plots for the registration error are shown in Fig. 1. We can see that ICDO can register more accurately that the baseline algorithms. In this case, the average computation times are 0.07s for ICP, 0.55s for IRLS, 2.08s for CPD, 18.93s for GMR, and 2.88s for ICDO.

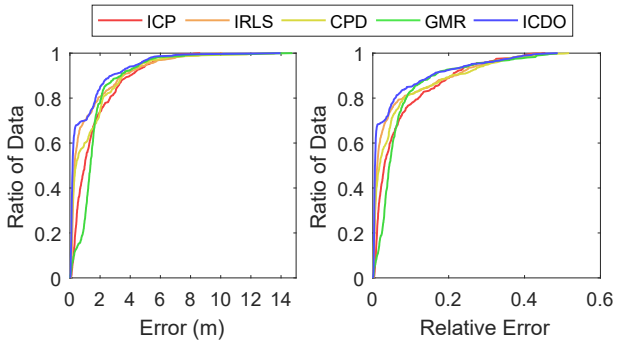

Figure 1. Results of ETH laser registration dataset (2 steps apart) in cumulative plots. (Left) Absolute registration error. (Right) Relative registration error.

3. Analysis on the Effect of Parameters

In this section, we look at the effect of the parameters of ICDO on the synthetic experiments and the ETH laser registration dataset [1]. We inspect 5 parameters: *(i)* regression parameter λ; *(ii)* number of boxes q; *(iii)* number of maps $\mathcal{T}$; *(iv)* discretization range r; and *(v)* rate for range reduction α. We follow the same training and test protocols as in the main paper, but we reduce the training samples to 2.5×10^4. For synthetic data, we only show the effect of the parameter on the success rate and computation time of different initial angles, where a total of 200 pairs were used to test each setting. For ETH dataset, we only show the absolute registration error. The experiments here were run on a server machine with AMD Opteron Processor 6378, 2400MHz, 128GB RAM.

Summary: We found that ICDO is not so sensitive to the regression parameter λ and number of discretization boxes q, while it can be rather sensitive to the discretization range r, the number of maps $\mathcal{T}$, and the rate for range reduction α. While synthetic data and ETH dataset prefer different sets of parameters, there exists a common set of parameters which perform well on both datasets (and also the Stanford's dragon) as reported in the main paper.

The detail of the results are provided in the following figures (explanations are provided in the caption):

- Regression parameter λ: Fig 2.
- Number of boxes q: Fig 3.
- Number of maps $\mathcal{T}$: Fig 4.
- Discretization range r: Fig 5.
- Rate of range reduction α: Fig 6.

4. Visualization of Results

In this section, we provide some visualization of the results from the main paper in Fig. 7 and 8. Column 1 shows the scenes in the initial configuration and the models. In column 2, we visualize the registration error between the scenes in the initial pose and the scenes in the ground truth pose. Column 3 shows the registered scenes with the models. Finally, column 4 visualizes the registration error between the registered scenes and the scene in the ground truth poses.

References

[1] F. Pomerleau, M. Liu, F. Colas, and R. Siegwart. Challenging data sets for point cloud registration algorithms. *The International Journal of Robotics Research*, 31(14):1705–1711, Dec. 2012. 2

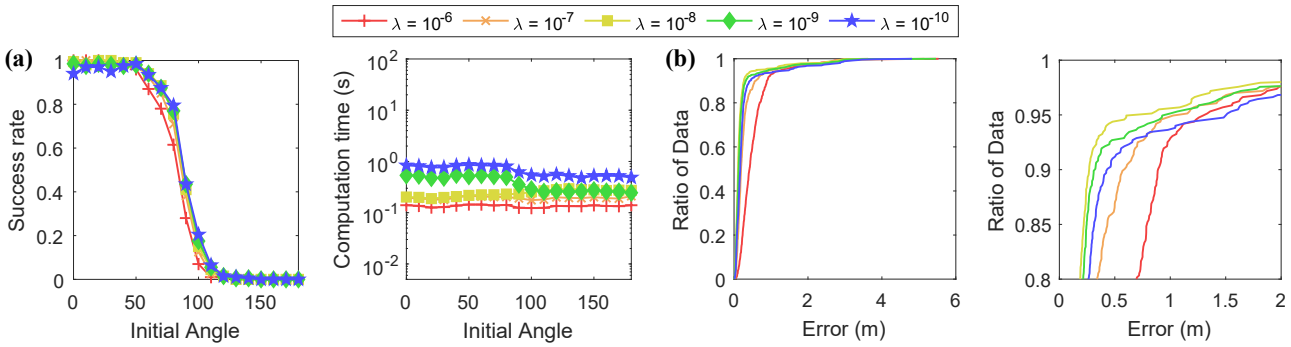

Figure 2. Effect of the regression parameter λ. (a) Results on synthetic data's initial angles in terms of success rate and time. (b) Results on ETH dataset (right subfigure shows a zoom of the left subfigure). We can see that λ has quite a small effect on the success rate and the error, except for $\lambda = 10^{-6}$ where the success rate in (a)-left is much lower, and the error in (b) is much higher than other λ's. In terms of computational time in (a)-right, we see that larger λ's require much less computational time than smaller λ's.

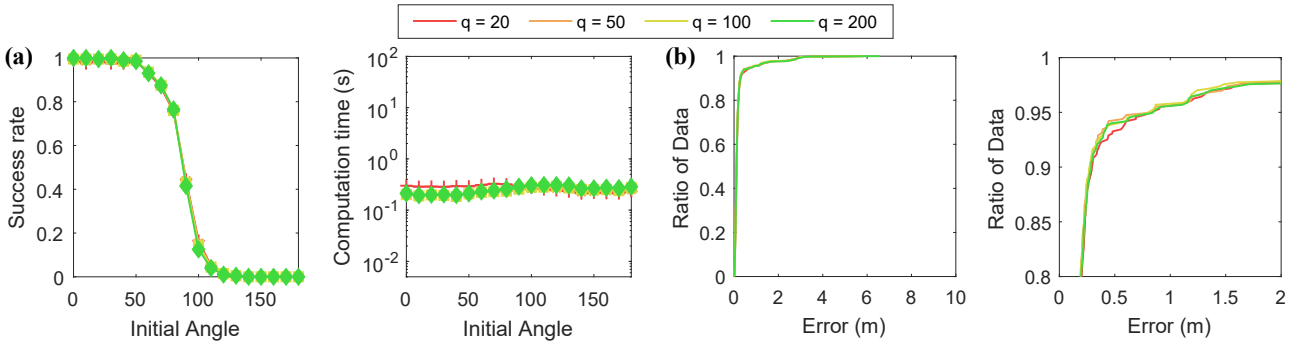

Figure 3. Effect of the number of boxes q. (a) Results on synthetic data's initial angles in terms of success rate and time. (b) Results on ETH dataset (right subfigure shows a zoom of the left subfigure). We can see that q has an almost insignificant effect on both the error and time. This illustrates that ICDO is rather insensitive to the number of boxes used in discretization.

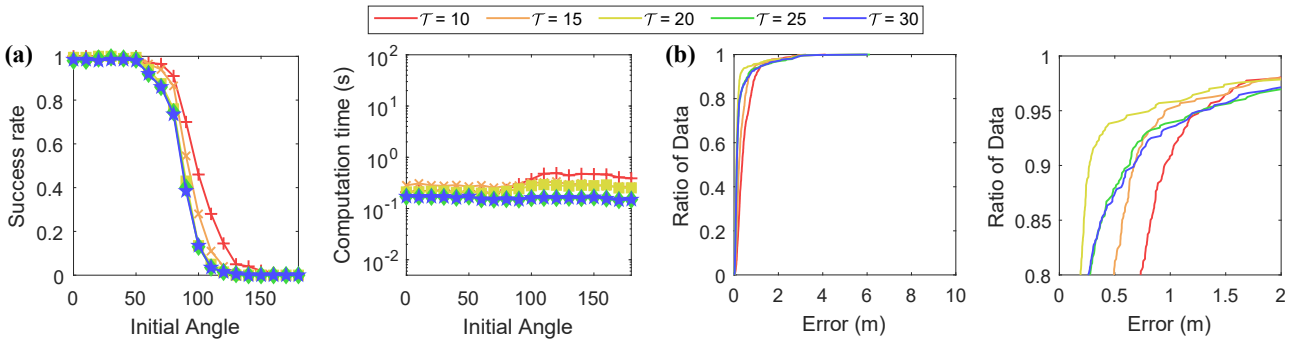

Figure 4. Effect of the number of maps $\mathcal{T}$. (a) Results on synthetic data's initial angles in terms of success rate and time. (b) Results on ETH dataset (right subfigure shows a zoom of the left subfigure). We can see in (a)-left that using a small $\mathcal{T}$ leads to more success registration. On the other hand, in (b), the best $\mathcal{T}$ should not be too low or too high. In terms of computation time in (a)-right, a small $\mathcal{T}$ requires more time than a large $\mathcal{T}$.

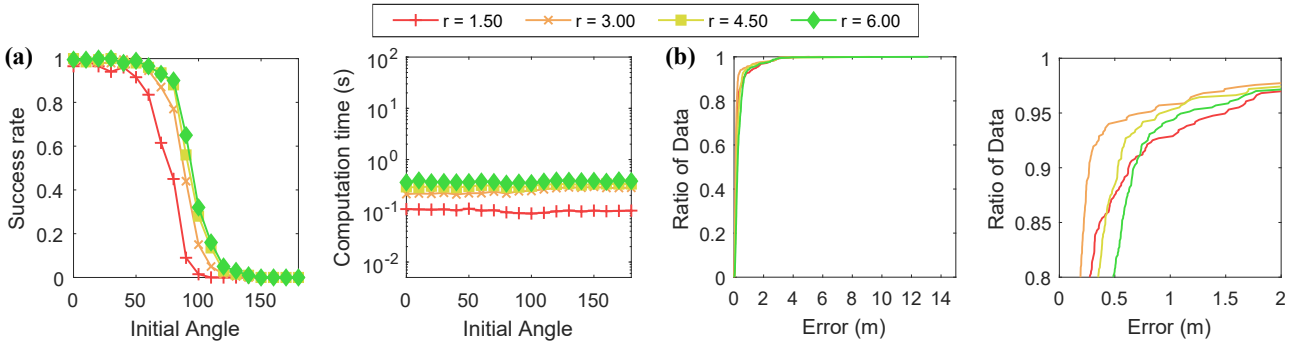

Figure 5. Effect of the discretization range r. (a) Results on synthetic data's initial angles in terms of success rate and time. (b) Results on ETH dataset (right subfigure shows a zoom of the left subfigure). We can see that in (a), a larger r increases success rates for large initial angles while also increases computation time. On the other hand, for ETH dataset in (b), we can see that the optimal r is 3, but other values also perform quite well.

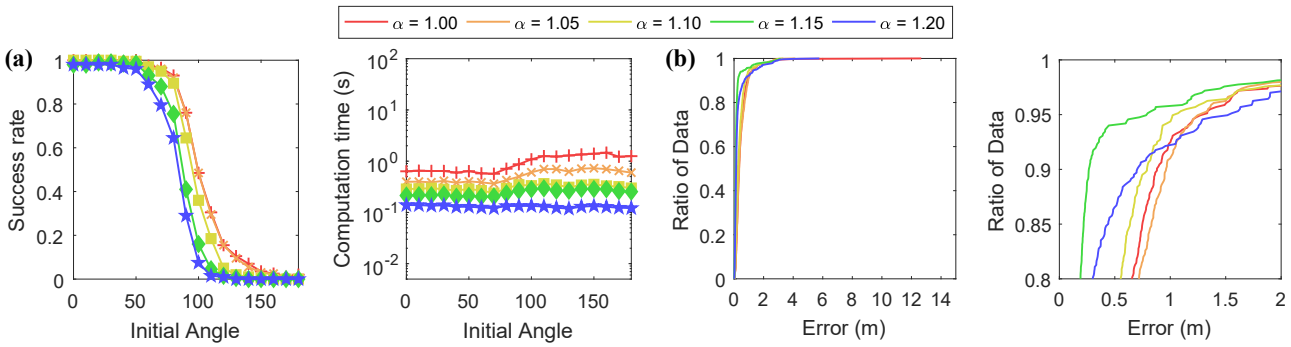

Figure 6. Effect of the rate of range reduction α. (a) Results on synthetic data's initial angles in terms of success rate and time. (b) Results on ETH dataset (right subfigure shows a zoom of the left subfigure). We can see that α has quite a strong effect on the result. For synthetic data in (a)-left, using a small α gives a significantly better success rate, while it also significantly increases computation time, as shown in (a)-right. For ETH dataset, setting α to 1.15 gives a much better result than others.

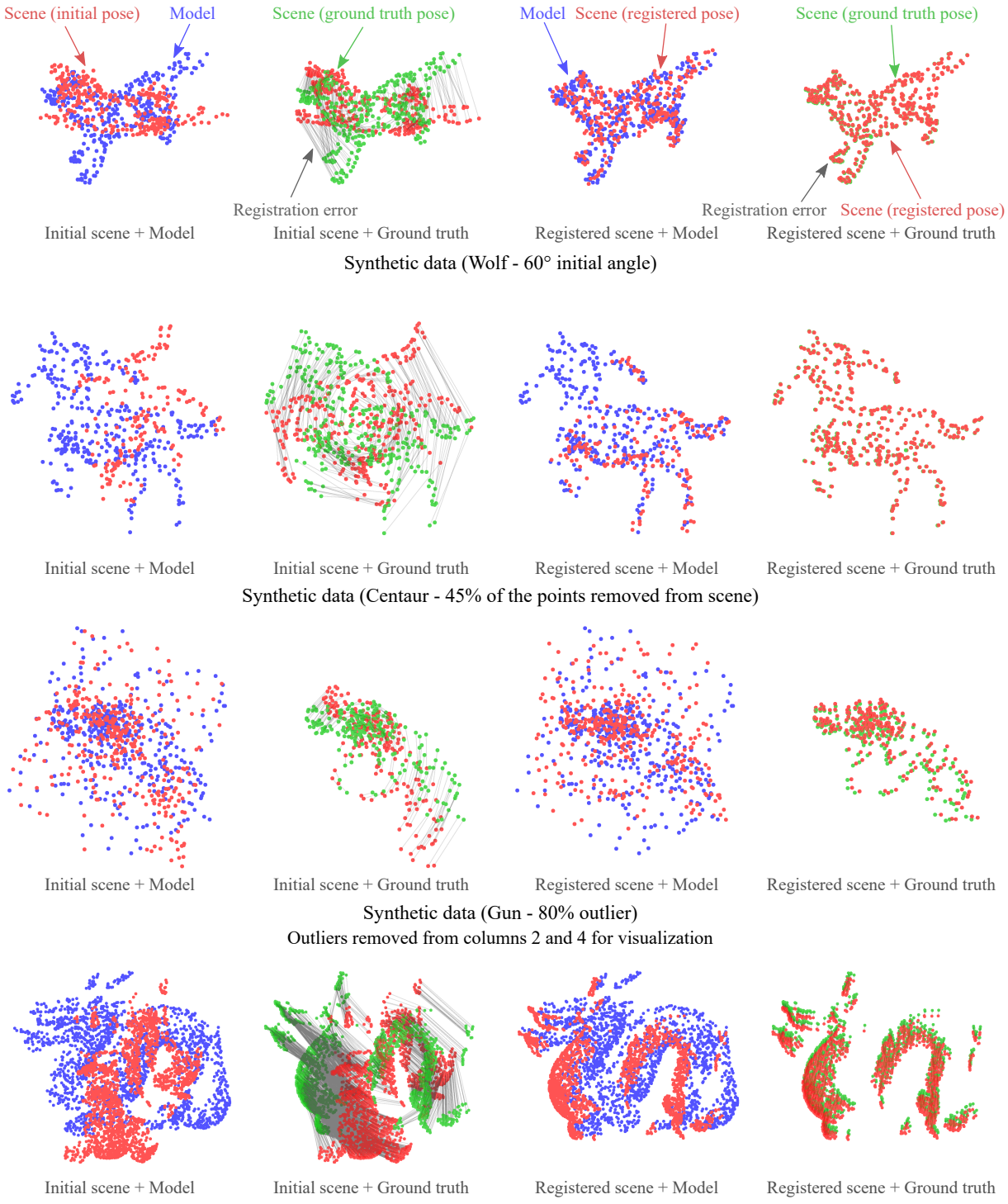

Figure 7. Visualization of results.
Column 1: Input to ICDO (Model (blue) and Scene in initial pose (red)).
Column 2: Initial error (Scene in ground truth pose (green) and Scene in initial pose (red). Gray lines are point-wise registration errors.)
Column 3: Registration result (Model (blue) and Scene in registered pose (red)).
Column 4: Final registration error (Scene in ground truth pose (green) and Scene in registered pose (red). Gray lines are point-wise registration errors.)

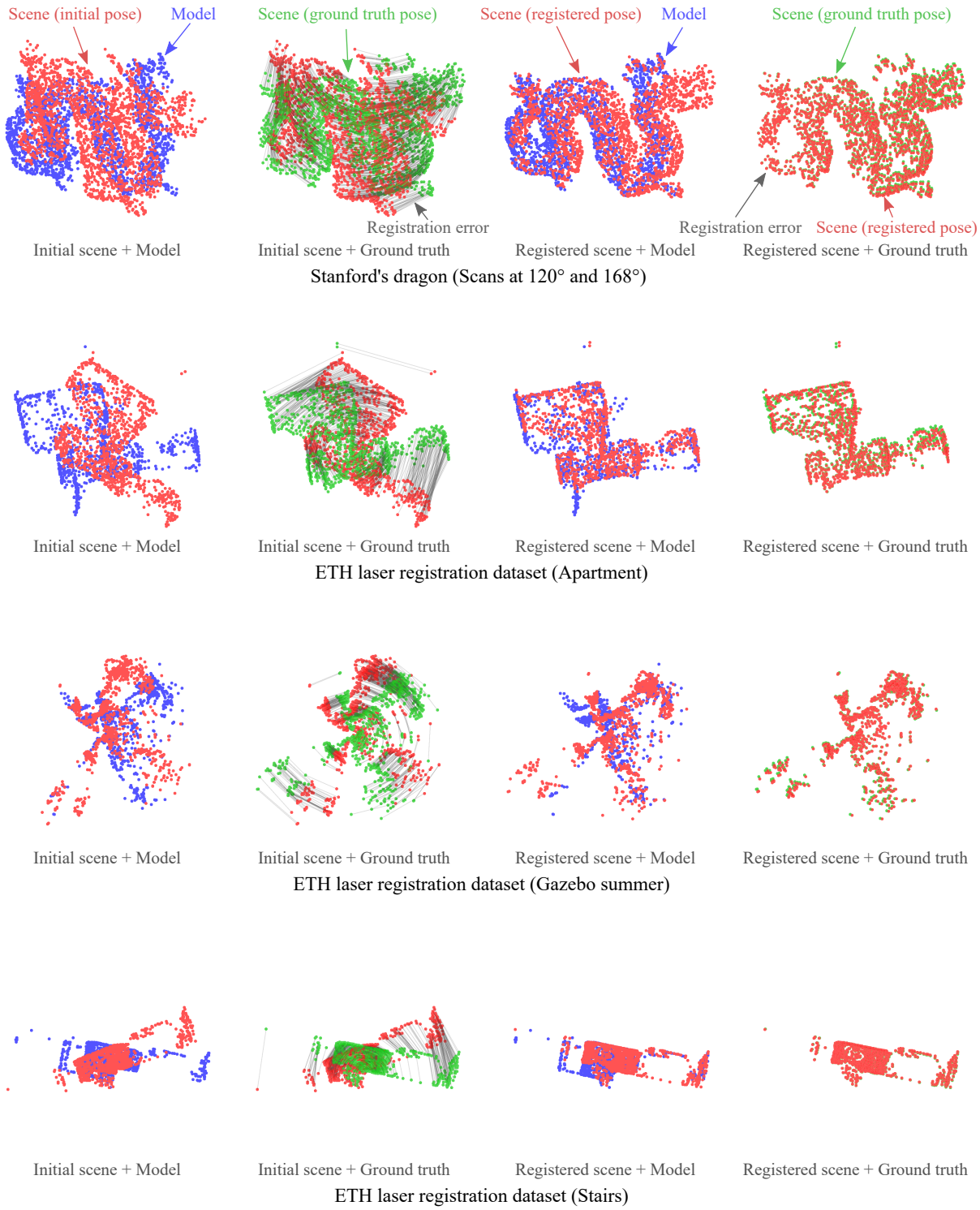

Figure 8. Visualization of results.
Column 1: Input to ICDO (Model (blue) and Scene in initial pose (red)).
Column 2: Initial error (Scene in ground truth pose (green) and Scene in initial pose (red). Gray lines are point-wise registration errors.)
Column 3: Registration result (Model (blue) and Scene in registered pose (red)).
Column 4: Final registration error (Scene in ground truth pose (green) and Scene in registered pose (red). Gray lines are point-wise registration errors.)