Supplementary Document
Discriminative Optimization: Theory and Applications to Point Cloud Registration

Jayakorn Vongkulbhisal[†,‡], Fernando De la Torre[‡], João P. Costeira[†]
[†]ISR - IST, Universidade de Lisboa, Lisboa, Portugal
[‡]Carnegie Mellon University, Pittsburgh, PA, USA
jvongkul@andrew.cmu.edu, ftorre@cs.cmu.edu, jpc@isr.ist.utl.pt

1. Experiments on 2D registration

We performed experiments on 2D registration using 4 shapes in Fig. 1 from [2, 4][1]. The shapes were normalized by removing the mean and scaling so that the largest dimension fitted $[-1, 1]$. We used the same baselines and performance metrics as in the main paper. Since MATLAB does not provide ICP for 2D case, we used the code of IRLS and set the cost function to least-squared error as ICP. In this 2D experiment, we set the error threshold for successful registration to 0.05 of the model's largest dimension.

(a) (b) (c) (d)

Figure 1. Point clouds for 2D registration experiment.

Training of DO: For each shape, we generated $N = 10000$ samples to train 30 maps as its SUM. Each sample is generated by adding the following perturbations: *(1) Rotation and translation:* We added a random rotation within 85 degrees and translation within $[-0.4, 0.4]^2$. *(2) Noise and outliers:* We added Gaussian noise of variance of 0.03 to each point, and added outliers of 0 to N_M[2] points in $[1.5, 1.5]^2$. *(3) Incomplete model:* We randomly sampled a point and removed from 0% to 60% of its closest points. For 2D case, we found that the shape in Fig. 1d had very different densities in different parts, which causes the denser area to dominate the values in $\mathbf{h}$. To alleviate this problem, in each training iteration, we preprocessed the features $\mathbf{h}^{(\mathbf{i})}$ by normalizing each element to lie in $[0, 1]$ before learning an update map. We trained a total of $K = 30$ maps, and set $\sigma^2 = 0.5$ and $\lambda = 2 \times 10^{-2}$. During test, we set the maximum number of iteration to 100. Training time for DO for all shapes took less than 45 seconds, except for the shape in Fig. 1d which took 147 seconds due to large number of points.

Test scenarios: We evaluated the performance of the alignment method by varying four types of perturbations: (1) the standard deviation of the noise ranging from 0 to 0.1, (2) the initial angle from 0 to 180 degrees, (3) the ratio of outliers from 0 to 2, and (4) the ratio of incomplete scene shape from 0 to 0.7. While we perturbed one parameter, the values of the other parameters were set as follows: noise SD = 0, initial angle uniformly sampled from 0 to 60 degrees, ratio of outliers = 0, and ratio of incompleteness = 0. For 2D case, we did not vary the number of scene points (as in 3D case) because each point cloud was already a sparse outline of its shape. The ratio of outlier is the fraction of the number of points N_M of each shape. All generated scenes included random initial translation within $[-0.4, 0.4]^2$. A total of 50 rounds were run for each variable setting for each shape.

Result: Fig. 2 shows the results for 2D registration. In terms of speed, DO performed faster than CPD and GMMReg, while being slower than ICP and IRLS. In terms of successful registration, ICP and IRLS had good success rates only when the perturbations and initial angles were small. GMMReg performed well in almost all cases, while CPD did not do well when there were a large number of outliers. DO, which learns the update steps from training data, obtained high success rates in almost all cases, but it does not do as well as CPD and GMMReg when the noise was extremely high. This is because the noise we generated was Gaussian noise, which is the noise model assumed by both CPD and GMMReg. This shows that when the problem is accurately modelled as an optimization problem, the optimum is generally the correct solution. GMMReg also outperformed DO when the outliers were high which may be due to its annealing steps. On the other hand, DO ob-

[1]Available from http://www.cs.cmu.edu/ ytsin/KCReg/KCReg.zip and http://cise.ufl.edu/ anand/students/chui/rpm/TPS-RPM.zip

[2]Recall that N_M is the number of points in the model point cloud.

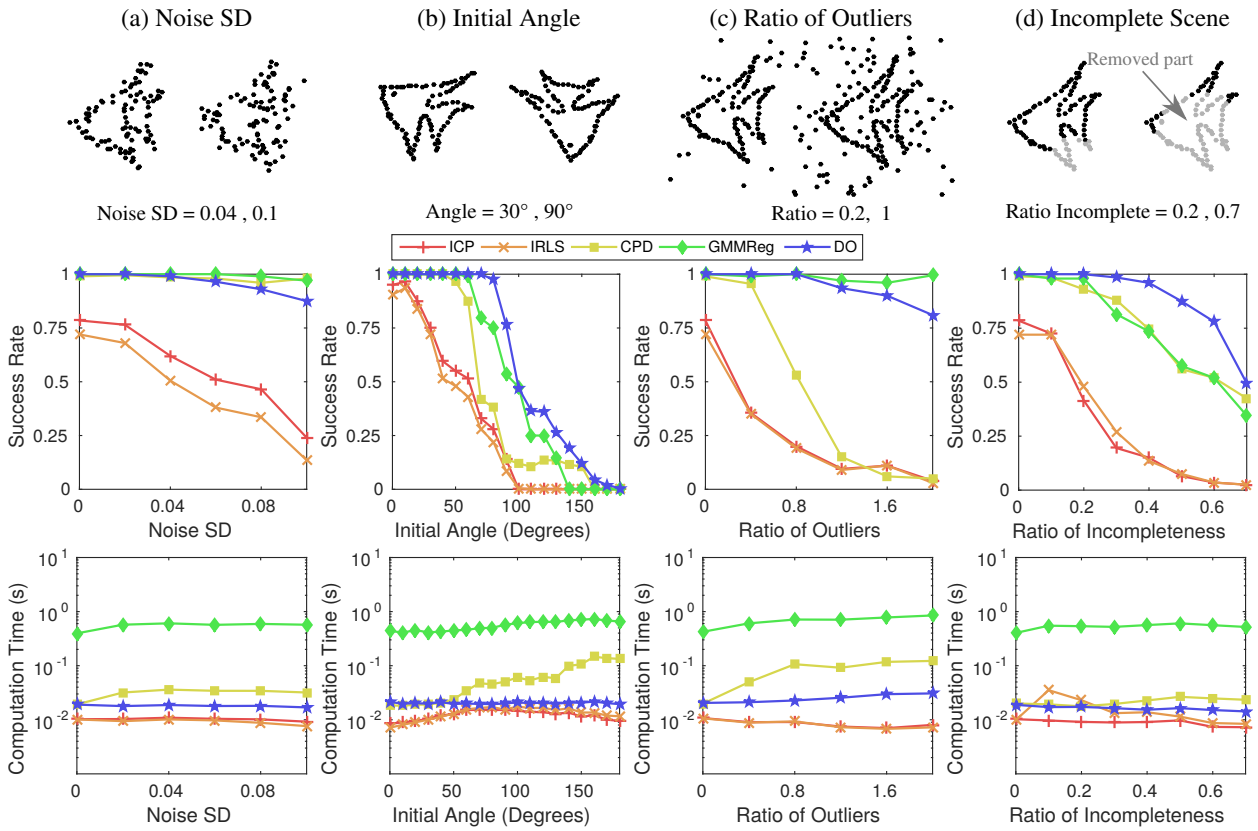

Figure 2. Results for 2D registration experiment. (Top) Examples of scene points with different perturbations. (Middle) Success rate. (Bottom) Computation time.

tained the best success rate in terms of initial angles and incomplete scenes. These perturbations cannot be easily modelled, and this result shows that it is beneficial to learn to obtain a good solution from training data as done by DO.

2. Proof of convergence of training data

In this section, we provide the proof of theorem 1 (re-stated below) from the main paper.

Theorem 1. (Strict decrease in training error under a sequence of update maps (SUM)) *Given a training set* $\{(\mathbf{x}_0^{(i)}, \mathbf{x}_*^{(i)}, \mathbf{h}^{(i)})\}_{i=1}^N$*, if there exists a linear map* $\hat{\mathbf{D}} \in \mathbb{R}^{p\times f}$ *such that, for each* i*,* $\hat{\mathbf{D}}\mathbf{h}^{(i)}$ *is strictly monotone at* $\mathbf{x}_*^{(i)}$*, and if* $\exists i : \mathbf{x}_k^{(i)} \neq \mathbf{x}_*^{(i)}$*, then the update rule:*

$$\mathbf{x}_{k+1}^{(i)} = \mathbf{x}_k^{(i)} - \mathbf{D}_{k+1}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)}), \tag{1}$$

with $\mathbf{D}_{k+1} \subset \mathbb{R}^{p\times f}$ *obtained from least-squared regression (Eq. (3) in the main paper), guarantees that the training error strictly decreases in each iteration:*

$$\sum_{i=1}^{N} \|\mathbf{x}_*^{(i)} - \mathbf{x}_{k+1}^{(i)}\|^2 < \sum_{i=1}^{N} \|\mathbf{x}_*^{(i)} - \mathbf{x}_k^{(i)}\|^2. \tag{2}$$

Proof. We assume that not all $\mathbf{x}_*^{(i)} = \mathbf{x}_k^{(i)}$, otherwise all $\mathbf{x}_*^{(i)}$ are already at their stationary points. Thus, there exists an i such that $(\mathbf{x}_k^{(i)} - \mathbf{x}_*^{(i)})^\top \hat{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)}) > 0$. We need to show that:

$$\sum_{i=1}^{N} \|\mathbf{x}_*^{(i)} - \mathbf{x}_{k+1}^{(i)}\|_2^2 < \sum_{i=1}^{N} \|\mathbf{x}_*^{(i)} - \mathbf{x}_k^{(i)}\|_2^2. \tag{3}$$

This can be shown by letting $\bar{\mathbf{D}} = \alpha\hat{\mathbf{D}}$ where:

$$\alpha = \frac{\beta}{\gamma}, \tag{4}$$

$$\beta = \sum_{i=1}^{N} (\mathbf{x}_k^{(i)} - \mathbf{x}_*^{(i)})^\top \hat{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)}), \tag{5}$$

$$\gamma = \sum_{i=1}^{N} \|\hat{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\|^2. \tag{6}$$

Since there exists an i such that $(\mathbf{x}_k^{(i)} - \mathbf{x}_*^{(i)})^\top \hat{\mathbf{D}}\mathbf{h}(\mathbf{x}_k^{(i)}) > 0$, both β and γ are both positive, and thus α is also positive. Next, we show that the training error decreases in each iteration as follows:

$$
\begin{aligned}
\sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_{k+1}^{(i)}\|^2 &= \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}+\mathbf{D}_{k+1}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\|^2 \\
&\leq \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}+\bar{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\|^2 \\
&= \sum_{i=1}^{N}\Big(\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2+\|\bar{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\|^2 \\
&\quad +2(\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)})^\top\bar{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\Big) \\
&= \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2 \\
&\quad +\underbrace{\sum_{i=1}^{N}\|\alpha\hat{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})\|^2}_{\alpha^2\gamma} \\
&\quad +2\alpha\underbrace{\sum_{i=1}^{N}(\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)})^\top\hat{\mathbf{D}}\mathbf{h}^{(i)}(\mathbf{x}_k^{(i)})}_{=-\beta} \\
&= \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2+\alpha^2\gamma-2\alpha\beta \\
&= \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2+\frac{\beta^2}{\gamma}-2\frac{\beta^2}{\gamma} \\
&= \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2-\underbrace{\frac{\beta^2}{\gamma}}_{>0} \\
&< \sum_{i=1}^{N}\|\mathbf{x}_*^{(i)}-\mathbf{x}_k^{(i)}\|^2.
\end{aligned}
$$

The second inequality is due to $\mathbf{D}_{k+1}$ being the optimal matrix that minimizes the squared error. Note that Thm. 1 does not guarantee that the error of each sample i reduces in each iteration, but guarantees the reduction in the average error. □

3. Exponential and logarithm maps

In this section, we provide explicit form of exponential maps and logarithm maps for $SE(3)$ and $SE(2)$. Interested readers can refer to the derivations in [1, 3].

3.1. 3D case : $SE(3)$

First, we define some notations. For $\mathbf{y}\in\mathbb{R}^3$, $[\cdot]_\times : \mathbb{R}^3\to\mathbb{R}^{3\times3}$ denotes the cross product matrix with its argument vector:

$$
[\mathbf{y}]_\times=\begin{bmatrix}0&-y_3&y_2\\y_3&0&-y_1\\-y_2&y_1&0\end{bmatrix}, \tag{7}
$$

and $[\cdot]_\triangle : \mathbb{R}^{3\times3}\to\mathbb{R}^3$ be its inverse operation, *i.e.*, $[[\mathbf{y}]_\times]_\triangle=\mathbf{y}$. We parametrize $\mathbf{x}$ as:

$$
\mathbf{x}=\begin{bmatrix}\mathbf{u}\\\omega\end{bmatrix}\in\mathbb{R}^6,\mathbf{u}\in\mathbb{R}^3,\omega\in\mathbb{R}^3. \tag{8}
$$

3.1.1 Exponential map

The exponential map:

$$
\exp\left(\begin{bmatrix}[\omega]_\times&\mathbf{u}\\\mathbf{0}_3^\top&1\end{bmatrix}\right)=\begin{bmatrix}\mathbf{R}&\mathbf{t}\\\mathbf{0}_3^\top&1\end{bmatrix}, \tag{9}
$$

can be computed in the following steps:

$$
\theta=\|\omega\|, \tag{10}
$$

$$
\mathbf{V}=\mathbf{I}_3+\left(\frac{1-\cos\theta}{\theta^2}\right)[\omega]_\times+\left(\frac{\theta-\sin\theta}{\theta^3}\right)[\omega]_\times^2, \tag{11}
$$

$$
\mathbf{R}=\mathbf{I}_3+\left(\frac{\sin\theta}{\theta}\right)[\omega]_\times+\left(\frac{1-\cos\theta}{\theta^2}\right)[\omega]_\times^2, \tag{12}
$$

$$
\mathbf{t}=\mathbf{V}\mathbf{u}. \tag{13}
$$

3.1.2 Logarithm map

The logarithm map:

$$
\log\left(\begin{bmatrix}\mathbf{R}&\mathbf{t}\\\mathbf{0}_3^\top&1\end{bmatrix}\right)=\begin{bmatrix}[\omega]_\times&\mathbf{u}\\\mathbf{0}_3^\top&1\end{bmatrix}, \tag{14}
$$

can be compute in the following steps:

$$
\theta=\arccos\left(\frac{trace(\mathbf{R})-1}{2}\right), \tag{15}
$$

$$
\ln(\mathbf{R})=\frac{\theta}{2\sin\theta}(\mathbf{R}-\mathbf{R}^\top), \tag{16}
$$

$$
\omega=[\ln(\mathbf{R})]_\triangle, \tag{17}
$$

$$
\mathbf{V}^{-1}=\mathbf{I}_3-\frac{1}{2}[\omega]_\times+\frac{1}{\theta^2}\left(1-\frac{\theta\sin\theta}{2(1-\cos\theta)}\right)[\omega]_\times^2, \tag{18}
$$

$$
\mathbf{u}=\mathbf{V}^{-1}\mathbf{t}. \tag{19}
$$

3.2. 2D case : $SE(2)$

We parametrize $\mathbf{x}\in\mathbb{R}^3$ as:

$$
\mathbf{x}=\begin{bmatrix}\mathbf{u}\\\theta\end{bmatrix},\mathbf{u}\in\mathbb{R}^2,\theta\in\mathbb{R}. \tag{20}
$$

3.2.1 Exponential map

The exponential map:

$$\exp\left(\begin{bmatrix} 0 & -\theta & u_1 \\ \theta & 0 & u_2 \\ 0 & 0 & 1 \end{bmatrix}\right) = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_2^\top & 1 \end{bmatrix} \tag{21}$$

can be computed in the following steps:

$$\mathbf{R} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}, \tag{22}$$

$$\mathbf{V} = \begin{bmatrix} \frac{\sin\theta}{\theta} & -\frac{1-\cos\theta}{\theta} \\ \frac{1-\cos\theta}{\theta} & \frac{\sin\theta}{\theta} \end{bmatrix}, \tag{23}$$

$$\mathbf{t} = \mathbf{V}\mathbf{u}. \tag{24}$$

3.2.2 Logarithm map

The logarithm map:

$$\log\left(\begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0}_2^\top & 1 \end{bmatrix}\right) = \begin{bmatrix} 0 & -\theta & u_1 \\ \theta & 0 & u_2 \\ 0 & 0 & 1 \end{bmatrix}, \tag{25}$$

can be computed in the following steps:

$$\theta = \texttt{atan2}(R_{21}, R_{11}) \tag{26}$$

$$A = \frac{\sin\theta}{\theta}, \tag{27}$$

$$B = \frac{1-\cos\theta}{\theta}, \tag{28}$$

$$\mathbf{V}^{-1} = \frac{1}{A^2+B^2}\begin{bmatrix} A & B \\ -B & A \end{bmatrix}, \tag{29}$$

$$\mathbf{u} = \mathbf{V}^{-1}\mathbf{t} \tag{30}$$

References

[1] E. Eade. Lie groups for 2d and 3d transformations. http://www.ethaneade.org/lie.pdf, Nov 2013. Accessed: 2016-06-30. 3

[2] S. Gold, A. Rangarajan, C.-P. Lu, P. Suguna, and E. Mjolsness. New algorithms for 2D and 3D point matching: pose estimation and correspondence. *Pattern Recognition*, 38(8):1019–1031, 1998. 1

[3] Y. Ma, S. Soatto, J. Kosecka, and S. S. Sastry. *An Invitation to 3-D Vision*. Springer-Verlag New York, 2004. 3

[4] Y. Tsin and T. Kanade. A correlation-based approach to robust point set registration. In *ECCV*, 2004. 1