General Discriminative Optimization for Point Set Registration

Yan Zhao[a,b], Wen Tang[a,*], Jun Feng[b], Taoruan Wan[c], Long Xi[a]

[a]*Department of Creative Technology, Bournemouth University, Poole, BH12 5BB, UK*
[b]*Department of Information Science and Technology, Northwest University, Xi'an, 710127,China*
[c]*School of Informatics, University of Bradford, Bradford,BD7 1DP,UK*

Abstract

Point set registration has been actively studied in computer vision and graphics. Optimization algorithms are at the core of solving registration problems. Traditional optimization approaches are mainly based on the gradient of objective functions. The derivation of objective functions makes it challenging to find optimal solutions for complex optimization models, especially for those applications where accuracy is critical. Learning-based optimization is a novel approach to address this problem, which learns the gradient direction from data sets. However, many learning-based optimization algorithms learn gradient directions via a single feature extracted from the data set, which will cause the updating direction to be vulnerable to perturbations around the data, thus falling into a bad stationary point. This paper proposes the General Discriminative Optimization (GDO) method that updates a gradient path automatically through the trade-off among contributions of different features on updating gradients. We illustrate the benefits of GDO with tasks of 3D point set registrations and show that GDO outperforms the state-of-the-art registration methods in terms of accuracy and robustness to perturbations.

Keywords: Point set registration, Supervised learning, Learning-based

*Corresponding author:
Email addresses: zhaoy@bournemouth.ac.uk (Yan Zhao), wtang@bournemouth.ac.uk (Wen Tang), fengjun@nwu.edu.cn (Jun Feng), t.wan@bradford.ac.uk (Taoruan Wan), lxi@bournemouth.ac.uk (Long Xi)

optimization

1. Introduction

Point set registration has been actively studied in computer vision and computer graphics. For shape reconstruction [1] [2], point set registration is used to find the overlaps of point sets reconstructed from images and align them in a global coordinate system. For face recognition [3], point set registration aligns the face descriptors extracted from a face with different facial expressions or different viewpoints. In medical image processing [4], point set registration is the fundamental step to fuse multiple images(e.g., computed tomography (CT), magnetic resonance imaging (MRI), and positron emission tomography (PET)). In intelligent vehicles [5], point set registration is an important step to align the images and extract feature points that will be further used for location and mapping. For shape retrieval [6], point set registration converts unstructured shapes into structured ones to rapidly retrieve 3D shapes that resemble a query object from a database.

The goal of point set registration is to find correspondences and to estimate the transformation between two or more point sets. Various rigid registration methods arise to solve the estimation of transformation parameters, i.e., a map defined as rotation and translation, which is essentially a mathematical optimization task. Gradient-based algorithms are widely used for solving the optimization problems in the applications of registration, such as gradient descent for multi-view reconstruction [7], Gauss-Newton for face alignment [8], Levenberg-Marquardt for surface fitting [9], Conjugate Gradient for surface reconstruction [10].

One of the gradient-based optimization algorithms is Newton's algorithm [11], which is an extremely powerful technique due to quadratic convergence. The computational cost for obtaining the second-order gradient information of the Hessian matrix makes Newton's method not feasible in many cases. Quasi-Newton methods are proposed to generate an estimation of the inverse Hession matrix, which leads to faster computation time. However, Quasi-Newton methods (such as Broyden–Fletcher–Goldfarb–Shanno algorithm (BFGS)) take a larger memory space to store the inverse Hessian approximation, thus could be detrimental for large, complicated tasks. Limited-memory BFGS (LBFGS), as the variant of the BFGS method, only stores a set of vectors and calculates a reduced rank approximation to the Hessian approximation, which needs much less memory to operate. However, the amount of storage required by LBFGS depends on the parameter setting that determines the number of BFGS corrections saved.

The high complexity and the large storage required for inverse Hessian approximation pose challenges to the applications of gradient-based optimization methods in computer vision and graphics while limiting the performances of the traditional registration methods that use gradient-based approaches for parameter estimation. In contrast, learning-based registration methods have a higher performance of registration with less time-consuming. The robustness and the high efficiency of the learning-based registration are due to that the approach integrates the traditional optimization (modeling and solution) as a learning-based optimization process, in which gradient directions are learned without calculating the Jacobian matrix or Hessian matrix. For example, Supervised Descent Method (SDM) [12] [13] and Discriminative Optimization (DO) method [14] learn to update directions from the single feature of the training data set and mimics gradient descent to estimate transformation parameters without registration modeling. However, if a single feature lacks robustness, it could make the learned direction susceptible to perturbations and likely to be trapped in stationary points rather than optimal solutions.

In this paper, we put forward a General Discriminative Optimization (GDO) method for point clouds registration. GDO, as the learning-based optimization method, overcomes the limitation of the learning-based registration methods. Our key insight is that, by balancing the contribution of different extracted features on the updating gradient, we can learn a sequence of gradient directions directly from training data sets while making the gradient path converge to the optimal point as closely as possible. We provide a framework for updating gradient directions via different features and show the proof of GDO's convergence. For 3D points registration, GDO learns a sequence of directions through the 3D coordination and density information of the point sets. The experimental results show that GDO outperforms the state-of-the-art registered algorithms in terms of robustness and accuracy on different data sets.

In the next section, we review related work on 3D registration and optimization. The following sections introduce our framework and theoretical analysis of GDO. Finally, we evaluate the registration performance of our algorithm.

2. Previous Work

2.1. 3D point sets registration

Point set registration has been an important problem in computer vision for the last few decades. The most commonly used method for registration is based on the iterative closest point (ICP) [15] algorithm, which finds the best transformation parameters of a group of three-dimensional points through rigid transformation and continuous iteration to minimize the difference between two point sets. Due to its conceptual simplicity, high usability, and good performance in practice, ICP and its variants are very popular and have been successfully applied in numerous real-world tasks. However, ICP is sensitive to outliers and needs initialization to be close to the optimal solution to avoid a bad local minimum. ICPMCC [16] combines ICP and the correntropy to improve the robustness of ICP in terms of the noises and outliers. GO-ICP [17] combines ICP with a branch-and-bound (BnB) scheme to search the optimal 3D motion space *SE(3)* efficiently. RICP [18], as a semantic-based method, avoids ICP trapping into local minimum due to the non-homogeneous point-set distribution or the poor initial pose through combining region selection, point matching, and noise treatment. Iteratively Reweighted Least Squares(IRLS) [19] uses various cost functions to provide robustness to outliers and avoid bad local minima. Fast global registration (FGR) [20] searches the correspondences between point sets through 3D feature descriptors and optimizes robust objectives based on those correspondences. Normal Distribution Transformation (NDT) [21] applies a statistical model to match 3D point sets. Coherent Point Drift (CPD) [22] achieves point sets registration based on a Gaussian mixture model, which moves the Gaussian mixture model centroids coherently as a group to preserve the topological structure of the point sets.

All the above registration methods cast point set registration as a traditional optimization problem, which can be divided into two stages: registration modeling and searching solutions. And the performances of the traditional registration approaches depend on the robustness of the registration models and searching methods. Learning-based registration methods, such as Supervised Descent Method (SDM) and Discriminative Optimization (DO), integrate the registration modeling and searching solutions as a learning-based optimization process, which leads to the robustness and high efficiency of registration.

2.2. Optimization algorithms

A general formula of the objective function of an optimization problem can be cast as follows:

$$\boldsymbol{x}_* = \min_{\boldsymbol{x} \in \mathbf{S}} f\left(\boldsymbol{x}\right). \tag{1}$$

$f : \mathbf{S} \rightarrow \mathbf{R}$ models the phenomena of interest and then finds the best solution $\boldsymbol{x}_*$ through a suitable search method. $\mathbf{S}$ is a set including all possible solutions for the objective function.

The least-square regression is the most popular form of objective functions, which is frequently employed in the majority of computer vision and computer graphics [23] [24] [25]. [26] [27] add various regularization terms to improve the robustness of least square regressions and reduce over-fitting. However, the addition of regularization terms increases the complexity of optimization models, which will further make it challenging to get the derivation of objective functions. Gradient descent and its variants, Newton's methods, and Quasi-Newton methods are commonly used to search optimal solutions of optimization models [28]. The learning rate of gradient descent is not optimal, the gradient information is not readily available [29], the Hessian matrix may not be positive definite, or the convergence rate is slower.

Several works have proposed to use learning techniques to compute the gradient directions of objective functions. Specifically, this is done by learning a sequence of regressors to replace the gradient directions of objective functions. [30] [31] regard weak learner as a gradient to update the parameter vector. [32] applies cascaded regression into facial landmark tracking system. [33] and [13] all learn a sequence of regressor matrices to update the shape parameters at per iteration. The former learns a set of averaged Jacobian and Hessian matrices from data, and the latter learns a mapping from image features to problem parameters directly.

[34] [35] [14] explore a framework to learn search directions from the feature of data without cost functions. Although this approach avoids the computation of Jacobian and Hessian matrices, it also uses only a *single feature* to learn the update direction. The lack of other features of data increases the risk of perturbations around data on the update of gradient directions. The *cooperation* of multiple features is able to effectively preserve and utilize geometric details of point sets, which is mostly used in semantic segmentation [36].

In view of this, in this paper, our proposed GDO algorithm learns the gradient update directions by combining different features of the point sets, fully utilizing the detailed information of point sets to reduce the impact of perturbations on gradient directions and, as a result, increasing the accuracy of the parameter estimation for registration.

3. General Discriminative Optimization

3.1. Motivation of Discriminative Optimization

Discriminative Optimization (DO) updates gradient directions according to the feature of input data without calculating the Jacobian or Hessian matrix. More specifically, DO splits gradient information as the updating map $\boldsymbol{D} \in \mathbf{R}^{p\times f}$ and the feature $\mathbf{h} : \mathbf{R}^p \rightarrow \mathbf{R}^f$, and updates the map $\boldsymbol{D}$ through approaching the current estimated parameter vector $\mathbf{x}_t$ to ground truth $\mathbf{x}_*$.

$$\mathbf{x}_{t+1} = \mathbf{x}_t - \boldsymbol{D}_{t+1}\mathbf{h}\left(\mathbf{x}_t\right) \tag{2}$$

$$\boldsymbol{D}_{t+1} = \min_{\tilde{\boldsymbol{D}}} \frac{1}{N}\sum_{i=1}^{N}\left\|\mathbf{x}_*^i - \mathbf{x}_t^i + \tilde{\boldsymbol{D}}\mathbf{h}\left(\mathbf{x}_t^i\right)\right\|_2^2 + \frac{\lambda}{2}\left\|\tilde{\boldsymbol{D}}\right\|_F^2 . \tag{3}$$

Where $\|\cdot\|_F$ is the Frobenius norm, and λ is a hyperparameter.

Despite not calculating the Jacobian or Hessian matrix, DO still has several issues theoretically. One issue is that DO uses a single feature of data to gain a sequence of updating maps. The lack of other features of data increases the risk of perturbations around data on the update of gradient directions. In this case, GDO explores the *collaboration* of different features $\mathbf{H}_f$ to reduce the impact of perturbations on the gradient direction.

Another theoretical issue of DO is that the constraint for the convergence of DO requires each $\boldsymbol{D}\mathbf{h}\left(\mathbf{x}\right)$ to be strictly monotone at ground truth for all samples. Actually, not all features are able to fulfill this constraint. In other words, the convergence constraint limits the select of feature function $\mathbf{h}$. We provide a weaker constraints for the convergence of the learning-based optimization.

3.2. Method

The objective function used to derive the feature of GDO can be formulated as follows:

$$\min_{\mathbf{x}} \quad \Phi\left(\mathbf{x}\right) = \sum_{i=1}^{I}\gamma_i \frac{1}{J_i}\sum_{j_i=1}^{J_i}\varphi_i\left(\mathbf{g}_{j_i}\left(\mathbf{x}\right)\right). \tag{4}$$

Where I is the number of categories of the different penalty functions φ_i. J_i is the number of residual functions $\mathbf{g}_{j_i}$. γ_i is the weighting coefficient of penalty function φ_i.

$$\mathbf{h}_i = \frac{1}{J_i}\sum_{j_i=1}^{J_i}\left[\frac{\partial \mathbf{g}_{j_i}}{\partial \mathbf{x}}\right]_{k,l}\delta\left(\mathbf{v} - \mathbf{g}_{j_i}\right). \tag{5}$$

$$\gamma_i = \frac{Tr\left(\mathbf{Cov}\left(\mathbf{h}_i\right)\right)}{\sum_{i=1}^{I} Tr\left(\mathbf{Cov}\left(\mathbf{h}_i\right)\right)}. \tag{6}$$

Where we express $\mathbf{g}_{j_i}(\mathbf{x})$ as $\mathbf{g}_{j_i}$ and $\varphi_i\left(\mathbf{g}_{j_i}(\mathbf{x})\right)$ as φ_i to reduce notation clutter. $[\boldsymbol{Y}]_k$ is the k_{th} row of $\boldsymbol{Y}$,and $[\boldsymbol{y}]_k$ means the k_{th} element of $\boldsymbol{y}$. $\delta(\mathbf{x})$ is the Dirac function. $Tr\left(\mathbf{Cov}\left(\mathbf{h}_i\right)\right)$ is the trace of the covariance matrix of $\mathbf{h}_i$. If $I = 2$, the feature of GDO $\mathbf{H}_f$ can be represented as follows:

$$\mathbf{H}_f = \begin{bmatrix} \gamma_1 \mathbf{h}_1 \\ \gamma_2 \mathbf{h}_2 \end{bmatrix}. \tag{7}$$

The details of the derivation and the summary of notation have been provided in the supplementary material.

3.3. Relation to the original DO

GDO can be seen as the extension of DO. When $I = 1$, the coefficient γ_i is set to 1, which means that DO has a single feature. In this case, GDO and DO are equivalent. When $I \neq 1$, the sum of the coefficients is still equal to 1, and GDO achieves the *cooperation* of multiple features. It is worth noting that the way to combine the features is derived from the function Eq.4, and the coefficients are learned from the features.

4. GDO Framework

4.1. Learning for GDO

Assume that we are given a set of training data $\left\{\left(\mathbf{x}_0^i, \mathbf{x}_*^i, \mathbf{H}_f(\mathbf{x}_0^i)\right)\right\}_{i=1}^{N}$, including N problem instances, each instance has its ground truth parameter $\mathbf{x}_*^i$, the initial parameter $\mathbf{x}_0^i$, and the extracted feature $\mathbf{H}_f(\mathbf{x}_0^i)$. For simplicity, we denote $\mathbf{H}_f(\mathbf{x}_t^i)$ as $\mathbf{H}_{ft}^i$ to represent the feature of the i-th sample at the t-th iteration. GDO aims at learning a sequence of maps $\boldsymbol{D}_{t+1}$ by approaching $\mathbf{x}_t^i$ to $\mathbf{x}_*^i$.

$$\boldsymbol{D}_{t+1} = \min_{\tilde{\boldsymbol{D}}} \frac{1}{N} \sum_{i=1}^{N} \left\| \mathbf{x}_*^i - \mathbf{x}_t^i + \tilde{\boldsymbol{D}} \mathbf{H}_{ft}^i \right\|_2^2 + \frac{\lambda}{2} \left\| \tilde{\boldsymbol{D}} \right\|_F^2. \tag{8}$$

Where $\|\cdot\|_F$ is the Frobenius norm, and λ is a hyperparameter.

We can apply the initial training data $\left\{\left(\mathbf{x}_0^i, \mathbf{x}_*^i, \mathbf{H}_{f0}^i\right)\right\}_{i=1}^N$ to (8) to learn map $\boldsymbol{D}_1$ at first. Then, $\boldsymbol{D}_1$ will be applied to (2) to get the current estimation $\mathbf{x}_1$. At each step, a new parameter vector can be created by recursively applying the update rule in (2). The learning process is repeated until certain termination criteria are met, for example, until the error is not reduced too much or the maximum number of iterations T is reached. The pseudocode for training GDO is shown in Alg.1.

Algorithm 1 Training a sequence of update maps

Require: $\left\{\left(\mathbf{x}_0^i, \mathbf{x}_*^i, \mathbf{H}_{f0}^i\right)\right\}_{i=1}^N$, T, λ
Ensure: $\{\boldsymbol{D}_t\}_{t=1}^T$

```
for t = 0 to T − 1 do
    Compute D_{t+1} with (8)
    for i = 1 to N do
        Update x^i_{t+1} := x^i_t − D_{t+1} H^i_{ft}
    end for
end for
```

4.2. Convergence analysis of GDO

Theorem 4.1 (Convergence of GDO's training error). Given a training set $\left\{\left(\mathbf{x}_0^i, \mathbf{x}_*^i, \mathbf{H}_{f0}^i\right)\right\}_{i=1}^N$, if there exists a linear map $\hat{\boldsymbol{D}} \in \mathbf{R}^{p\times f}$ where $\hat{\boldsymbol{D}}\mathbf{H}_f$ meets the condition $\sum_{i=1}^N \left(\mathbf{x}_*^i - \mathbf{x}_t^i\right)^{\mathrm{T}} \hat{\boldsymbol{D}}\mathbf{H}_{ft}^i > 0$ at $\mathbf{x}_*^i$ for all i, and if there exists an i where $\mathbf{x}_t^i \neq \mathbf{x}_*^i$, then the update rule:

$$\mathbf{x}_{t+1}^i = \mathbf{x}_t^i - \boldsymbol{D}_{t+1}\mathbf{H}_{ft}^i. \tag{9}$$

$$\boldsymbol{D}_{t+1} = \min_{\tilde{\boldsymbol{D}}} \frac{1}{N}\sum_{i=1}^N \left\|\mathbf{x}_*^i - \mathbf{x}_t^i + \tilde{\boldsymbol{D}}\mathbf{H}_{ft}^i\right\|_2^2 + \frac{\lambda}{2}\left\|\tilde{\boldsymbol{D}}\right\|_F^2.$$

guarantees that the training error strictly decreases in each iteration:

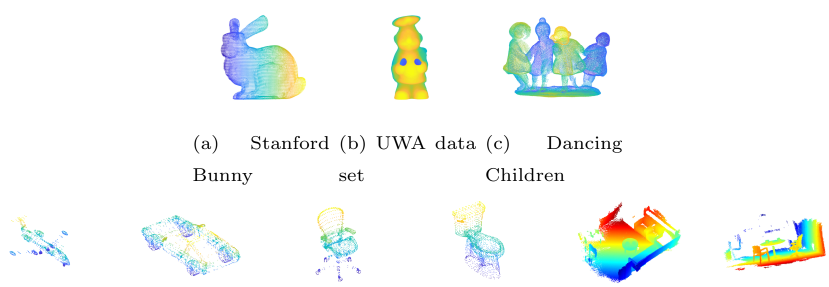
$$\sum_{i=1}^N \left\|\mathbf{x}_*^i - \mathbf{x}_{t+1}^i\right\|_2^2 < \sum_{i=1}^N \left\|\mathbf{x}_*^i - \mathbf{x}_t^i\right\|_2^2. \tag{10}$$

If $\hat{\boldsymbol{D}}\mathbf{H}_f$ is strongly monotone, and if there exist $H > 0$, $M > 0$ such that $\left\|\hat{\boldsymbol{D}}\mathbf{H}_f^i\right\|_2^2 \leq H + M \left\|\mathbf{x}_*^i - \mathbf{x}^i\right\|_2^2$ for all i, then the training error converges to zero.

The proof of Thm.4.1 is provided in Supplementary Material. Thm.4.1 says that for all instances, if $\hat{\boldsymbol{D}}\mathbf{H}_f$ meets the condition $\sum_{i=1}^{N} \left(\mathbf{x}_*^i - \mathbf{x}_t^i\right)^{\mathrm{T}} \hat{\boldsymbol{D}}\mathbf{H}_{ft}^i > 0$, then the average training error will decrease in each iteration; if $\hat{\boldsymbol{D}}\mathbf{H}_f$ is strongly monotone at $\mathbf{x}_*^i$, the average training error will converge to zero. Note that $\mathbf{H}_f$ can be not only a single function but also a combination of different functions of $\mathbf{x}^i$. DO also presents a similar convergence result for a update rule, but it requires $\hat{\boldsymbol{D}}\mathbf{H}_{ft}^i$to be strictly monotone at $\mathbf{x}_*^i$ for all i. Besides, different from the single feature $\mathbf{h}$ in DO, as the combination composed of several feature functions, $\mathbf{H}_f$ takes into account more features of data.

5. EXPERIMENTATION

This section describes how to apply GDO to 3D point set registration with various perturbations. We compare GDO with other classical registration methods on various data sets.



(a) Stanford Bunny (b) UWA data set (c) Dancing Children

(d) Airplane627 (e) Car201 (f) Chair964 (g) Toilet378 (h) Indoor Scene 01 (i) Indoor Scene 02

Figure 1: Experimental data sets

5.1. 3D Point set Registration

Let $\{\mathbf{M}, \mathbf{S}\}$ be two point sets in a finite-dimensional real vector space $\mathbf{R}^3$, which contains $\mathbf{N}_m$ and $\mathbf{N}_s$ points, respectively. Our goal is to find a rigid transformation $\boldsymbol{T}$ to be applied to scene set $\mathbf{S}$ such that the difference between $\mathbf{S}$ and model set $\mathbf{M}$ is minimized. The transformation matrix $\boldsymbol{T}$ is posed as the Lie algebra $\boldsymbol{x} \in \mathbf{R}^6$ in our optimization problem.

Feature for registration

The feature $\mathbf{H}_f$ for registration is combined by two different features: the coordinates-based feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^c$ and the density-based feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$.

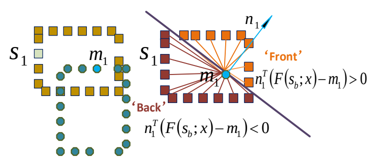


Figure 2: The positional relationship between scene points (square) $\mathbf{s}_1$ and model point (hexagon) $\mathbf{m}_1$.

We use the feature extraction method in [34] to extract the features $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^c$ and $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$, where $\mathbf{h}$ is devised to be a histogram indicating the weights of scene points on the 'front' and the 'back' sides of each model point. As shown in Fig.2.

$$\mathbf{S}_a^+ = \left\{\mathbf{s}_b : \mathbf{n}_a^{\mathrm{T}} \left(\mathbf{F}\left(\mathbf{s}_b;\boldsymbol{x}\right) - \mathbf{m}_a\right) > 0\right\} \tag{11}$$

$\mathbf{S}_a^+$ indicates the set of scene points on the 'front' of model point $\mathbf{m}_a$, and $\mathbf{S}_a^-$ contains the remaining scene points.; $\mathbf{n}_a \in \mathbf{R}^3$ is the normal vector of the model point $\mathbf{m}_a$; $\mathbf{F}(\mathbf{s}_b;\boldsymbol{x})$ is the function that applies rigid transformation with parameter $\boldsymbol{x}$ to scene point $\mathbf{s}_b$.

Then the feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^c$ can be calculated through the following formulas:

$$[\mathbf{h}(\boldsymbol{x};\mathbf{S})]_{a^+}^c = \frac{1}{z} \sum_{\mathbf{s}_b \in \mathbf{S}_a^+} \exp\left(\frac{-1}{\hat{\sigma}^2} \left\|\mathbf{F}(\mathbf{s}_b;\boldsymbol{x}) - \mathbf{m}_a\right\|^2\right). \tag{12}$$

$$[\mathbf{h}(\boldsymbol{x};\mathbf{S})]_{a^-}^c = \frac{1}{z} \sum_{\mathbf{s}_b \in \mathbf{S}_a^-} \exp\left(\frac{-1}{\hat{\sigma}^2} \left\|\mathbf{F}(\mathbf{s}_b;\boldsymbol{x}) - \mathbf{m}_a\right\|^2\right). \tag{13}$$

Where z normalizes $\mathbf{h}$ to sum to 1, and $\hat{\sigma}$ controls the width of the exp function. The design of the feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$ can be divided into two stages. The first stage is to calculate the probability of measuring each point of $\mathbf{S}$ in the boxes of $\mathbf{M}$, and the probability of each point of $\mathbf{M}$ in the boxes of $\mathbf{S}$, as shown in Fig.3. The second stage is to apply the calculated probability to (12),(13) to extract the density feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$.

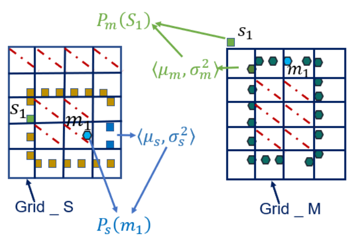


Figure 3: The first stage for designing the density feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$. The Grid_S represents the grids around the model $\mathbf{S}$. The Grid_M represents the grids around the model $\mathbf{M}$. The grid marked by the red dotted line represents the grid where is no point, which will be removed when calculating the mean μ and covariance σ^2.

The probability of measuring each point of $\mathbf{S}$ in the boxes of $\mathbf{M}$ can be calculated as follows, and the probability of measuring the points of $\mathbf{M}$ in the boxes of $\mathbf{S}$ can be calculated in a similar way.

1. The 3D space around the point set $\mathbf{M}$ is subdivided regularly into boxes with constant size (e.g. the Grid_S, Grid_M in Fig.3).
2. For each box, the following is done:
   - Collect all 3D points $\boldsymbol{m}_{i=1,2,\cdots,N_m}$ in $\mathbf{M}$ contained in this box. If there is no point in a box, the box will be removed (e.g. the grids marked by the red dotted line in Fig.3).
   - Calculate the mean
   $$\boldsymbol{\mu_m} = \frac{1}{N_m}\sum_{i=1}^{N_m}\boldsymbol{m}_i.$$
   - Calculate the covariance matrix
   $$\boldsymbol{\sigma_m^2} = \frac{1}{N_m}\sum_{i=1}^{N_m}\left(\boldsymbol{m}_i - \boldsymbol{\mu_m}\right)\left(\boldsymbol{m_i} - \boldsymbol{\mu_m}\right)^{\mathrm{T}}.$$

3. The probability of measuring each point $\boldsymbol{s}_j$ of $\mathbf{S}$ in this box is now modeled by the normal distribution $\mathcal{N}\left(\boldsymbol{\mu_m}, \boldsymbol{\sigma_m^2}\right)$.

$$\mathbf{P}_m\left(\boldsymbol{s}_j\right) \sim \exp\left(\frac{-(\boldsymbol{s_j}-\boldsymbol{\mu_m})^{\mathrm{T}}(\boldsymbol{s_j}-\boldsymbol{\mu_m})}{2\boldsymbol{\sigma}}\right).$$

$$[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d_{a^+} = \frac{1}{z}\sum_{\mathbf{s}_b\in\mathbf{S}_a^+}\exp\left(\frac{-1}{\hat{\sigma}^2}\left\|\mathbf{P}_m(\mathbf{F}(\mathbf{s}_b;\boldsymbol{x}))-\mathbf{P}_s(\mathbf{m}_a)\right\|^2\right). \tag{14}$$

$$[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d_{a^-} = \frac{1}{z}\sum_{\mathbf{s}_b\in\mathbf{S}_a^-}\exp\left(\frac{-1}{\hat{\sigma}^2}\left\|\mathbf{P}_m(\mathbf{F}(\mathbf{s}_b;\boldsymbol{x}))-\mathbf{P}_s(\mathbf{m}_a)\right\|^2\right). \tag{15}$$

The final feature $\mathbf{H}_f$ can be posed as:

$$\mathbf{H}_f = \begin{bmatrix}\gamma_1[\mathbf{h}\left(\boldsymbol{x};\mathbf{S}\right)]^c \\ \gamma_2[\mathbf{h}\left(\boldsymbol{x};\mathbf{S}\right)]^d\end{bmatrix}. \tag{16}$$

We get the coefficients γ_1, γ_2 using (6), which represent the contributions of features on updating gradient direction.

5.2. GDO Training Settings

The parameters in the GDO training process are the same as those in the code provided in the Github of DO [34] for the comparison experiments on the synthetic data sets. We normalize a given model shape $\mathbf{M}$ to $[-1,1]^3$ and uniformly sample from $\mathbf{M}$ with the replacement 400 to 700 points to generate a scene model. Then we apply the following perturbations to the scene model: *(i) Rotation and translation:* The rotation is within 60 degrees and the translations is in $[-0.3, 0.3]^3$, which represents ground truth $\mathbf{x}_*$; *(ii) Noise and Outliers:* Gaussian noise with the standard deviation 0.05 is added to the scene model. 0 to 300 points within $[-1.5, 1.5]^3$ are added as the sparse outliers. Besides, a Gaussian ball of 0 to 200 points with the standard deviation of 0.1 to 0.25 is used to simulate the structured outliers; *(iii) incomplete shape:* We remove 40% to 90% points from scene model to simulate occlusions, the detailed removing approach can be found in [34]. For all experiments, we generated 30000 training samples, set up iterations $T = 30$ and set λ as 2×10^{-4}, β^2 as 0.03, and the initial transformation $\mathbf{x}_0$ is $\mathbf{0}^6$. For the second feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$, we build the uniform grid in the range $[-2, 2]$ with 81 points in each dimension.

For the comparison experiments on Modelnet40 dataset, we design three modes for GDO training. (i) $mode_1$: The rotation is within 45 degrees and the translations is in $[-0.5, 0.5]^3$; (ii) $mode_2$: The rotation is within 90 degrees and the translations is in $[-0.5, 0.5]^3$; (iii) $mode_3$: The rotation is within 90 degrees, the translations is in $[-0.5, 0.5]^3$ and Gaussian noise with the standard deviation 0.05 is also applied. The first two modes aim to compare the registration of all methods in terms of varying degrees of rotation, named *single-class training.* The latter is to compare the performance of different methods on the registration with multiple perturbations, named *multi-class training.* We generated 30000 training samples for all modes, and the training sample will be normalized to $[-1, 1]^3$ without downsampling. The number of points of all samples is 5120.

5.3. Performance Metrics

Baselines. We compared GDO with the advanced learning-based approach DO [34], two point-based approaches (ICP and IRLS), two density-based approaches (CPD and NDT) and the feature-based approach (FGR).

We used the successful registration rate, average MSE and computation time as performance metrics.

Successful Registration Rate. A registration is successful when the mean ℓ_2 is less than 0.05 of the model's largest dimension.

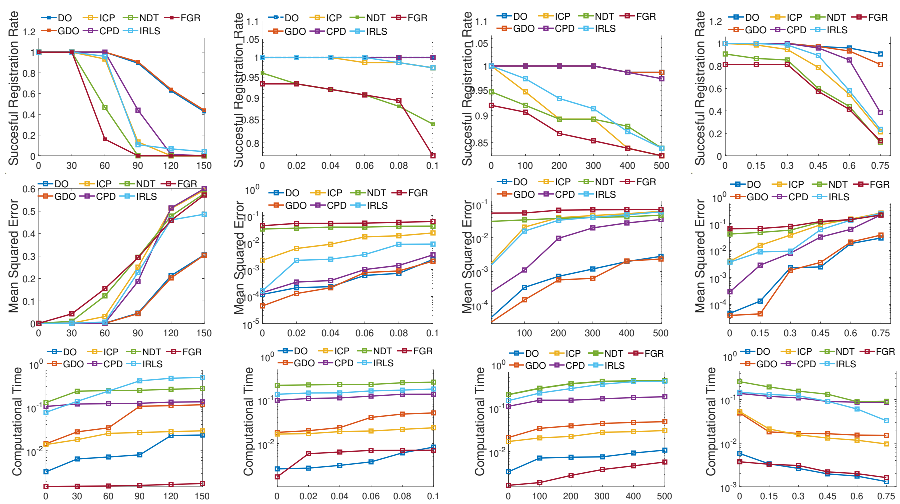
Average MSE. It is worth noting that the MSE is the mean ℓ_2 error between the model and scene sets, and the Average MSE is the average for MSE for all test sets.

In order to make the experimental results more clear, we use log_{10} MSE and log_{10} computing time to describe the accuracy and efficiency of the registration of all registration methods on ModelNet40 dataset.

5.4. Parameters settings

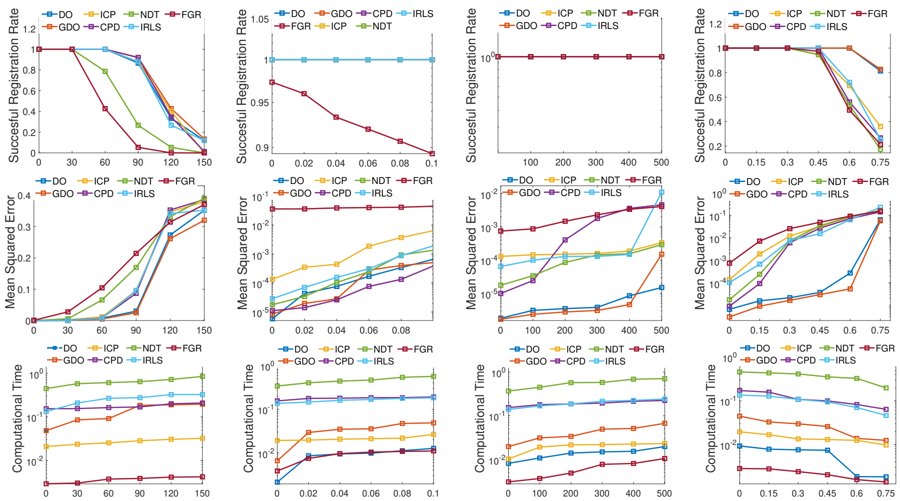
The maximum number of iterations of all registration methods were set to 30. For *DO* and *GDO*, we set λ as 2×10^{-4}, β^2 as 0.03. The value of the tolerance of absolute difference between current estimation and ground truth in iterations is 1e-4; For *ICP*, the tolerance of absolute difference in translation and rotation is 0.01 and 0.5 respectively; For *IRLS*, we used *Huber criterion function* as the regression function, the remaining parameters were set as the same as the setting of *ICP*. For *CPD*, the type of transformation is set to *rigid*, and the expected percentage of outliers with respect to a normal distribution is 0.1, the tolerance value is the same of that in DO. For *NDT*, the value of expected percentage of outliers is set to 0.55, and the tolerance value is set as the same of that in ICP ; For *FGR*, the value of the division factor used for graduated non-convexity is 1.4, the maximum correspondence distance is 0.025, the value of the similarity measure used for tuples of feature points is 0.95, the value of the maximum tuple numbers for trading off between speed and accuracy is set to 1000.

For *BCPD*, the expected percentage of outliers is set to 0.1, the parameter in Gaussian kernel is 2.0 and the expected length of displacement vector is 400. All deep-learning based registration networks are trained on an Nvidia Geforce 2080Ti GPU with 12G memory. For *PCRNet*, the kernel sizes are 64, 64, 64, 128, 1024, 1024, 512, 512, 256 and 7. The iteration for rotation and translation is set to 8. Adam optimizer with an initial learning rate of 0,1, 300 epochs and a batch size of 32 are used for the training process. For *PointnetLK*, the kernel sizes are 64, 64, 64, 128, 1024. The maximum iteration for rotation and translation is set to 30. Adam optimizer with an initial learning rate of 0.001, 250 epochs and a batch size of 10 are used for the training process. For *DCP*, the kernel sizes are 64, 64, 128, 256, 512, 1024, 256, 128, 64, 32 and 7. The iteration for rotation and translation is set to 1. Adam optimizer with an initial learning rate of 0.001, 250 epochs and a batch size of 32 are used for the training process. For *ICPMCC*, the error threshold is set to 10^{-7}, the iteration number is 30, and the number of nearest points for calculating normal vectors is set to 10.



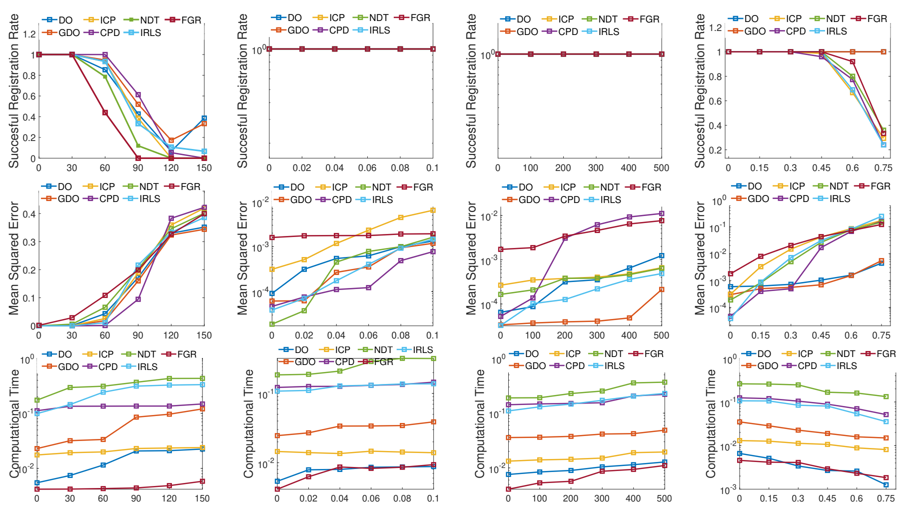
(a) Initial Angles (b) Noise SD (c) Outliers (d) Incompleteness

Figure 4: Results of 3D registration with Bunny model under different perturbations;(Top) Examples of scene points with different perturbations. (Second Row) Successful Registration Rate (SRR). (Third row) Average MSE (AMSE). (Bottom) Computation Time. In the presence of noise and outliers, the registration success rates of most algorithms are the same, which is 1, so the number of visualized dash lines is less than the number of algorithms. Learning-based registration algorithms (DO, GDO) can deal with point set registration with more accuracy than traditional registration algorithms (ICP, CPD, NDT, IRLS, and FGR). GDO is more time-consuming than DO, although its performance is slightly better than the performance of DO.



(a) Initial Angles (b) Noise SD (c) Outliers (d) Incompleteness

Figure 5: Results of 3D registration with Chef model under different perturbations



(a) Initial Angles (b) Noise SD (c) Outliers (d) Incompleteness

Figure 6: Results of 3D registration with Dancing Children model under different perturbations

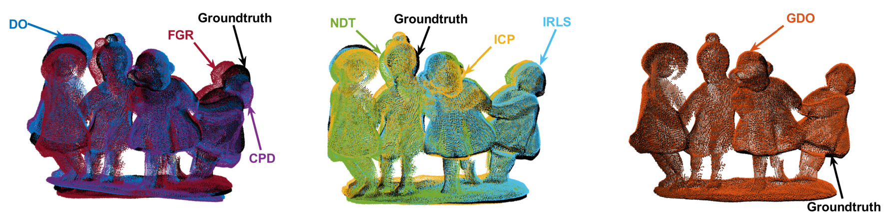
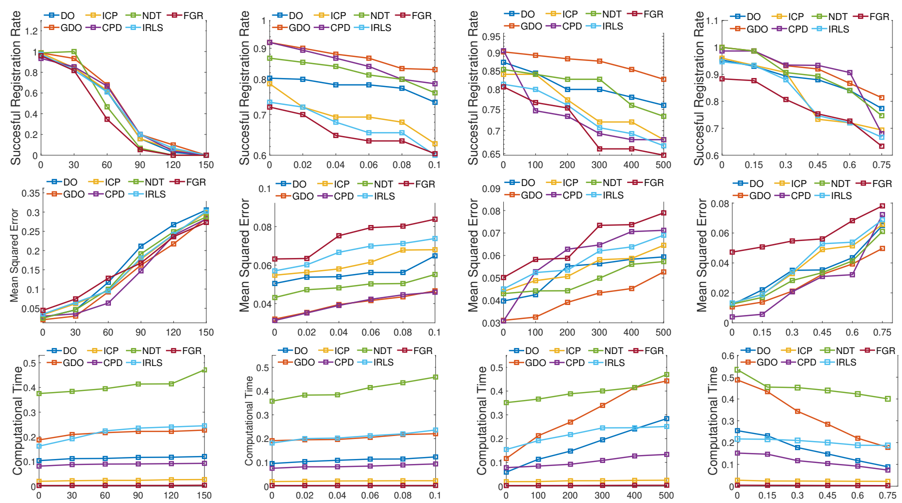


Figure 7: Registration results of Dancing Children model with 500 outliers

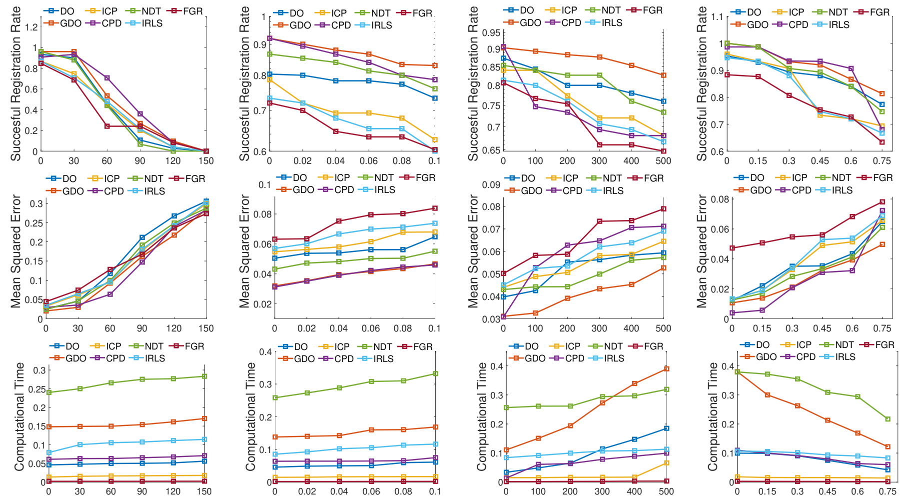


(a) Initial Angles (b) Noise SD (c) Outliers (d) Incompleteness

Figure 8: Results of 3D registration with Indoor Scene01 model under different perturbations

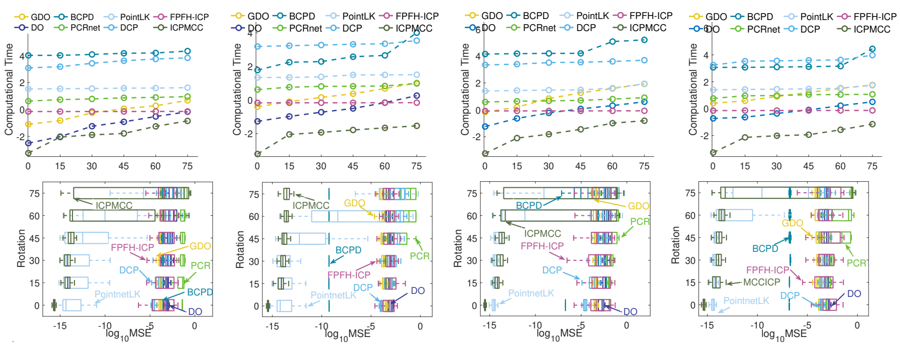
5.5. Registration Experiments

We have used the Stanford Bunny model[37], UWA dataset [38], Dancing Children, Indoor Scene[39] as the data sets for experiments Fig.1. Dancing Children are available at the AIM@SHAPE shape repository `http://visionair.ge.imati.cnr.it/ontologies/shapes/`. The model set $\mathbf{M}$ is generated by using the grid average downsample method in MATLAB to select 477 points from the original model.The performance of algorithms are evaluated by comparing the evaluation metrics in the case of various perturbations: *(1) rotation:* We compare the performance metrics when the initial angle is 0°, 30°, 60°, 90°, 120° and 150°[default=0°to60°]; *(2) noise:* The standard deviation of the noise



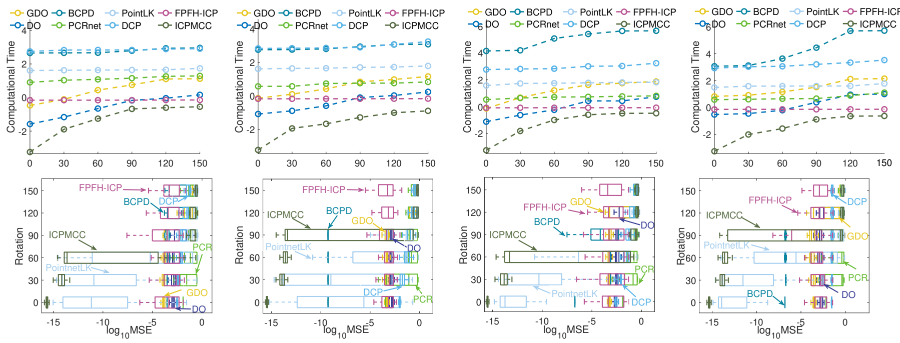
(a) Initial Angles (b) Noise SD (c) Outliers (d) Incompleteness

Figure 9: Results of 3D registration with Indoor Scene02 dataset under different perturbations



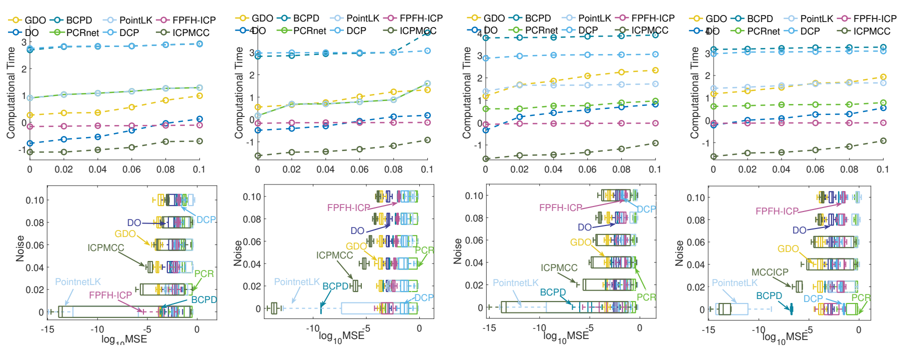
(a) Airplane627 (b) Car201 (c) Chair964 (d) Toilet378

Figure 10: The registration results on Modelnet40 with perturbation setting $mode_1$. (Top) The computational time for registration. (Bottom) the $log10MSE$ of all the comparison methods. GDO, DO, FPFH-ICP and ICPMCC cost less time to achieve the registration. Although BCPD and PointnetLK register point clouds with more accuracy, the computation time of both is higher, even the time required for BCPD is almost dozens of times the time required for other registration methods. By comparison, GDO represents better registration performance with less computational time. And FPFH-ICP has poor stability. ICPMCC is unable to handle the registration over 60^o.



(a) Airplane627 (b) Car201 (c) Chair964 (d) Toilet378

Figure 11: The registration results on Modelnet40 with perturbation setting $mode_2$. (Top) The computational time for registration. (Bottom) the $log10MSE$ of all the comparison methods. The accuracy of DO, GDO and BCPD has a sharp decrease after the registration over 90^o. However, GDO still performs better than other methods in terms of accuracy. The performance of the deep-learning methods (PCR, PointnetLK and DCP) is not good even on the registration with the rotation of 60^o. BCPD and DCP are still the most time-consuming.



(a) Airplane627 (b) Car201 (c) Chair964 (d) Toilet378

Figure 12: The registration results on Modelnet40 with perturbation setting $mode_3$. (Top) The computational time for registration. (Bottom) the $log10MSE$ of all the comparison methods. DO and GDO can keep the higher stability and accuracy on the registration with multiple perturbations, compared with other methods. The ability of deep-learning methods to handle the registration with multiple perturbations is poor than that of the traditional methods. The performance of FPFH-ICP is still stable, but the accuracy is not high.

is set to 0, 0.02, 0.04, 0.06, 0.08 and 0.1 [default=0]; *(3) outliers:* We set the number of outliers to 0, 100, 200, 300, 400 and 500 respectively [default=0]; *(4) incomplete ratio:* The ratio of incomplete scene shape is set to 0, 0.15, 0.3, 0.45, 0.6 and 0.75 [default=0]. The random translation of all generate scenes is within $[-0.3, 0.3]^3$. When one parameter is changed, the values of other parameters are fixed to default values. In addition, the scene points are sampled from the original model, not from $\mathbf{M}$. We will test 750 testing samples in each variable setting. It is noteworthy that the training samples are generated by adding various perturbations to the model $\mathbf{M}$ and assigning random parameters for the translation and rotation of the model $\mathbf{M}$. The testing samples are generated similarly, but the degree of perturbation and the parameters for transformation are different, and the down-sampled model is the original model instead of the model $\mathbf{M}$.

We also conduct comparative registration experiments on the ModelNet40 dataset [40] with traditional methods (BCPD [41], FPFH-ICP [42] and ICPMCC [16]) and other advanced deep-learning-based registration methods, such as PCR-Net [43], PointnetLK [44], and DCP [45] (as shown in $(d) \sim (g)$ of Fig.1). There are three kinds of comparison settings corresponding to the training modes in 5.2: for $mode_1$: the initial angle is 0°, 15°, 30°, 45°, 60° and 75°; for $mode_2$: the initial angle is 0°, 30°, 60°, 90°, 120° and 150°; for $mode_3$: the initial angle is 0°, 30°, 60°, 90°, 120° and 150° [default=0° to 90°] and the standard deviation of Gaussian noise is set to 0, 0.02, 0.04, 0.06, 0.08 and 0.1 [default=0]. It is worth noting that when we change one parameter, the values of other parameters are fixed to the default value. We will test 100 test samples in each variable setting. The registration results with the single-class training scheme are shown in Fig.10 and Fig.11. The registration results with the multi-class training scheme are shown in Fig.12.

Experimental Results

Registration results. Fig.4 and Fig.5 show the 3D registration results on Bunny model and Chef model with various perturbations. It can be seen that in the presence of arbitrary perturbation, learning-based registration algorithms (DO, GDO) can achieve more accurate registration results than the traditional registration methods (ICP, CPD, NDT, IRLS, FGR). Compared with DO, the performance of GDO is slightly better than DO. However, GDO is more time-consuming, the reason for which is that the second feature $[\mathbf{h}(\boldsymbol{x};\mathbf{S})]^d$ calculates the density probability of each point in point sets, which involves the search of the closest box. Also, the calculation way of the second feature determines that the running time of GDO and the size of the point set are positively correlated.

Fig.6 and Fig.7 show the registration results on Dancing Children model. The trend and distribution of the running time of all algorithms on the Dancing Children model are the same as that on Bunny or Chef models. GDO is more capable when dealing with the complex model than registering simple models (Bunny, Chef), which can be illustrated by the Mean Square Error criteria.

Fig.8 and Fig.9 show the results of 3D registration on Indoor Scenes. The performances of NDT and GDO are prominent when registering real scenes models.

While FGR and ICP required low computation time for all cases, they had low success rates when the perturbations were high. CPD performed well in all cases except when the number of outliers was high. The running time of IRLS was similar to that of CPD when dealing with the registration of simple models (Bunny, Chef); it did not perform well when the model was highly incomplete. NDT achieved more accurate registration of real scenes than other algorithms; it was the most time-consuming for all cases. For the learning-based algorithms, DO and GDO outperformed the baselines when registering simple models. When dealing with the complex model (Dancing Children) and real large scene models, GDO performed better than DO. This is because DO just considers one single feature that does not consider the internal topology or density distribution of points, which makes it lack robustness than GDO.

Fig.10 displays the performance of all methods on registration with $mode_1$. It can be seen that the accuracy of BCPD is higher than other methods, but BCPD takes almost dozens of times as long as other algorithms. DCP takes about the same time as BCPD, but its accuracy and stability are poor. The poor performance also occurs on the PCRnet method. By contrast, PointnetLK can keep higher stability and accuracy when dealing with the registration not over 60^o. Compared with the deep-learning methods, as the traditional learning-based method, DO and GDO can achieve the registration with higher accuracy and stability. FPFH-ICP also performs well. The stability of ICPMCC has a sharp decrease when ICPMCC registers the registration over 60^o.

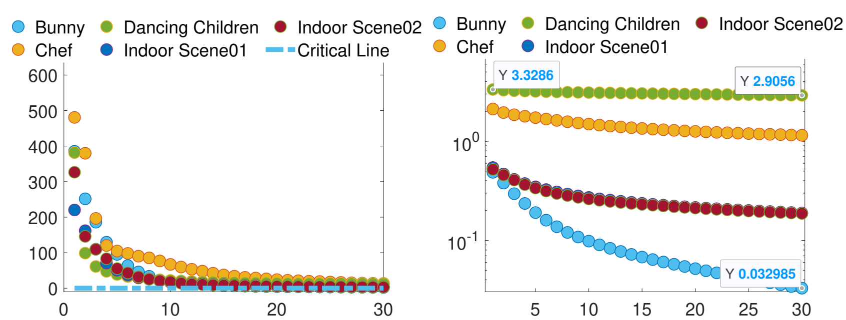
Fig.11 shows the registration results on the perturbation of larger rotations $mode_2$. The stability and accuracy of ICPMCC and PointnetLK are worse when ICPMCC and PointnetLK handle the registration over 60^o. The performance of DCP and PCR is unstable as ever. DO, GDO, and BCPD can keep the high accuracy and stability until they register points sets with large rotations (over 120^o). Nevertheless, the accuracy of GDO is higher than that of BCPD and DO when dealing with the registration over 120^o. FPFH-ICP still keeps its high stability and accuracy, and the performance of ICPMCC is poor once it is used to achieve the registration with larger rotations.

Fig.12 illustrates the registration results on Modelnet40 dataset with multiple perturbations $mode_3$. DO and GDO can keep the higher stability and accuracy on the registration with multiple perturbations, compared with other methods. The ability of deep-learning methodsto handle the registration with multiple perturbations is poor than that of the traditional methods. The performance of FPFH-ICP is still stable, but the accuracy is not high.

In summary, the learning-based methods (DO and GDO) have higher stability and robustness compared with deep-learning methods (PCRnet, PointnetLK, and DCP) and other traditional methods. FPFH-ICP performs well even on the registration with larger rotations, but the accuracy of FPFH-ICP is not better, which may be caused by the fewer iterations for FPFH to find correspondences. The ability to achieve more accurate and stable registration on larger rotations or multiple perturbations for the deep-learning methods and ICPMCC is limited.

The benefit of the FPFH-ICP is the ability to handle registration with larger rotations while maintaining higher stability. Comparing the registration results on $mode_2$ and $mode_3$, it can be seen that the only drawback of the traditional learning-based methods (DO and GDO) is the less ability to register point clouds over 120^o, which illustrates that the learning-based methods are more vulnerable on rotations, not noises. In addition, the features in GDO can be replaced by any features extracted by 3D feature descriptors such as Fast Point Feature Histograms (FPFH) descriptors, Signature of Histogram of Orientations (SHOT), and so on. The potential issue of the usage of various descriptors is whether it will increase the degree of over-fitting of the learning-based methods.

Verify Convergence. Fig.13 shows the Convergence Criteria and Training Error of our method on different data sets. We can find that the $\hat{\boldsymbol{D}}\mathbf{H}_f$ in our method meets the convergence condition $\sum_{i=1}^{N}\left(\boldsymbol{x}_*^i-\boldsymbol{x}_t^i\right)^{\mathrm{T}}\hat{\boldsymbol{D}}\mathbf{H}_f\left(\boldsymbol{x}_t^i\right)>0$ for all data sets, and the training error of our method decreases in each iteration.



(a) Convergence Criteria (b) Training Error

Figure 13: The Convergence Criteria and Training Error of our method on different data sets. (a) The value of $\sum_{i=1}^{N}\left(\boldsymbol{x}_*^i-\boldsymbol{x}_t^i\right)^{\mathrm{T}}\hat{\boldsymbol{D}}\mathbf{H}_f\left(\boldsymbol{x}_t^i\right)$on different data sets. (b) The training error of our method on different data sets.

6. Conclusion and Discussion

This paper proposes general discriminative optimization (GDO) method to solve the transformation parameter estimation in point set registration by learning update directions from different features of training samples. Specifically, GDO derivates an approach to achieve the *collaboration* of the different extracted features from point sets to reduce the effect of perturbations on updating directions. In this paper, GDO combines a coordinates-based feature and a density-based feature to update the gradient map to improve the accuracy and robustness of transformation estimation. We provided a theoretical result on the convergence of the registration method under mild conditions. We also illustrate GDO outperformed state-of-the-art registration approaches on different data sets. The major advantage of GDO over traditional registration methods and learning-based registration methods include robustness to outliers and other perturbations, which is more prominent when dealing with complex 3D models and real scene models registration. The limitation of GDO is that the training point cloud and the test point cloud are highly relevant, limiting its ability to train many point clouds and achieve multiple point clouds registration like the registration methods based on deep learning. In addition, the feature extraction approach of GDO takes longer as the number of points increases. Future works of interest are to design a feature function that is more robust to perturbations and more efficient, and to design a registration framework to enable GDO to achieve multiple point clouds registration. The strong theoretical foundation and good registration performance of GDO suggest its usefulness as a general-purpose registration technique.

Acknowledgment

This research did not receive any specific grant from funding agencies in the public, commercial, or not-for-profit sectors.

References

[1] Choi, SM, Kim, MH. Shape reconstruction from partially missing data in modal space. Computers & Graphics 2002;26(5):701–708.

[2] Ludwig, M, Berrier, S, Tetzlaff, M, Meyer, G. 3d shape and texture

morphing using 2d projection and reconstruction. Computers & Graphics 2015;51:146–156.

[3] Berretti, S, Werghi, N, Del Bimbo, A, Pala, P. Matching 3d face scans using interest points and local histogram descriptors. Computers & Graphics 2013;37(5):509–525.

[4] Zhu, X, Ding, M, Huang, T, Jin, X, Zhang, X. Pcanet-based structural representation for nonrigid multimodal medical image registration. Sensors 2018;18(5):1477.

[5] Li, L, Yang, M, Wang, C, Wang, B. Rigid point set registration based on cubature kalman filter and its application in intelligent vehicles. IEEE Transactions on Intelligent Transportation Systems 2018;19(6):1754–1765.

[6] Liu, Z, Xie, C, Bu, S, Wang, X, Han, J, Lin, H, et al. Indirect shape analysis for 3d shape retrieval. Computers & Graphics 2015;46:110–116.

[7] Goldlücke, B, Aubry, M, Kolev, K, Cremers, D. A super-resolution framework for high-accuracy multiview reconstruction. International Journal of Computer Vision 2014;106(2):172–191.

[8] Tzimiropoulos, G, Pantic, M. Gauss-newton deformable part models for face alignment in-the-wild. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2014, p. 1851–1858.

[9] Bo, P, Ling, R, Wang, W. A revisit to fitting parametric surfaces to point clouds. Computers & Graphics 2012;36(5):534–540.

[10] Paulsen, RR, Baerentzen, JA, Larsen, R. Markov random field surface reconstruction. IEEE Transactions on Visualization and Computer Graphics 2009;16(4):636–646.

[11] Fischer, A. A special newton-type optimization method. Optimization 1992;24(3-4):269–284.

[12] Xiong, X, De la Torre, F. Supervised descent method and its applications to face alignment. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2013, p. 532–539.

[13] Xiong, X, De la Torre, F. Global supervised descent method. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2015, p. 2664–2673.

[14] Vongkulbhisal, J, De la Torre, F, Costeira, JP. Discriminative optimization: Theory and applications to point cloud registration. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017,.

[15] Besl, PJ, McKay, ND. Method for registration of 3-d shapes. In: Sensor Fusion IV: Control Paradigms and Data Structures; vol. 1611. International Society for Optics and Photonics; 1992, p. 586–607.

[16] Du, S, Xu, G, Zhang, S, Zhang, X, Gao, Y, Chen, B. Robust rigid registration algorithm based on pointwise correspondence and correntropy. Pattern Recognition Letters 2020;132:91–98.

[17] Yang, J, Li, H, Jia, Y. Go-icp: Solving 3d registration efficiently and globally optimally. In: Proceedings of the 2013 IEEE International Conference on Computer Vision. 2013,.

[18] Wan, T, Du, S, Cui, W, Yao, R, Ge, Y, Li, C, et al. Rgb-d point cloud registration based on salient object detection. IEEE Transactions on Neural Networks and Learning Systems 2021;:1–13doi:10.1109/TNNLS.2021.3053274.

[19] Bergström, P, Edlund, O. Robust registration of point sets using iteratively reweighted least squares. Computational Optimization and Applications 2014;58(3):543–561.

[20] Zhou, QY, Park, J, Koltun, V. Fast global registration. In: European Conference on Computer Vision. Springer; 2016, p. 766–782.

[21] Biber, P, Straßer, W. The normal distributions transform: A new approach to laser scan matching. In: Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453); vol. 3. IEEE; 2003, p. 2743–2748.

[22] Myronenko, A, Song, X. Point set registration: Coherent point drift. IEEE Transactions on Pattern Analysis and Machine Intelligence 2010;32(12):2262–2275.

[23] Zhu, H, Sheng, B, Shao, Z, Hao, Y, Hou, X, Ma, L. Better initialization for regression-based face alignment. Computers & Graphics 2018;70:261–269.

[24] Liu, X, Zhang, J, Cao, J, Li, B, Liu, L. Quality point cloud normal estimation by guided least squares representation. Computers & Graphics 2015;51:106–116.

[25] Shen, XJ, Liu, SX, Bao, BK, Pan, CH, Zha, ZJ, Fan, J. A generalized least-squares approach regularized with graph embedding for dimensionality reduction. Pattern Recognition 2020;98:107023.

[26] Meer, P, Mintz, D, Rosenfeld, A, Kim, DY. Robust regression methods for computer vision: A review. International Journal of Computer Vision 1991;6(1):59–70.

[27] Ke, Q, Kanade, T. Robust l/sub 1/norm factorization in the presence of outliers and missing data by alternative convex programming. In: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05); vol. 1. IEEE; 2005, p. 739–746.

[28] Ruder, S. An overview of gradient descent optimization algorithms. arXiv preprint arXiv:160904747 2016;.

[29] Griewank, A, Walther, A. Evaluating derivatives: principles and techniques of algorithmic differentiation; vol. 105. SIAM; 2008.

[30] Mason, L, Baxter, J, Bartlett, PL, Frean, MR. Boosting algorithms as gradient descent. In: Advances in Neural Information Processing Systems. 2000, p. 512–518.

[31] Friedman, JH. Greedy function approximation: a gradient boosting machine. Annals of Statistics 2001;:1189–1232.

[32] Liu, Q, Yang, J, Deng, J, Zhang, K. Robust facial landmark tracking via cascade regression. Pattern Recognition 2017;66:53–62.

[33] Ge, Y, Peng, C, Hong, M, Huang, S, Yang, D. Joint local regressors learning for face alignment. Neurocomputing 2016;208:262–268.

[34] Vongkulbhisal, J, De la Torre, F, Costeira, JP. Discriminative optimization: Theory and applications to computer vision problems. arXiv preprint arXiv:170704318 2017;.

[35] Vongkulbhisal, J, De la Torre, F, Costeira, JP. Discriminative optimization: Theory and applications to computer vision. IEEE Transactions on Pattern Analysis and Machine Intelligence 2019;41(4):829–843.

[36] Hu, Q, Yang, B, Xie, L, Rosa, S, Guo, Y, Wang, Z, et al. Randla-net: Efficient semantic segmentation of large-scale point clouds. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2020, p. 11108–11117.

[37] Levoy, M, Gerth, J, Curless, B, Pull, K. The stanford 3d scanning repository. URL http://www-graphics stanford edu/data/3dscanrep 2005;5.

[38] Mian, AS, Bennamoun, M, Owens, RA. A novel representation and feature matching algorithm for automatic pairwise registration of range images. International Journal of Computer Vision 2006;66(1):19–40.

[39] Lai, K, Bo, L, Fox, D. Unsupervised feature learning for 3d scene labeling. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). IEEE; 2014, p. 3050–3057.

[40] Wu, Z, Song, S, Khosla, A, Yu, F, Zhang, L, Tang, X, et al. 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on Computer Vision and Pattern Recognition. 2015, p. 1912–1920.

[41] Hirose, O. A bayesian formulation of coherent point drift. IEEE transactions on pattern analysis and machine intelligence 2020;.

[42] Rusu, RB, Blodow, N, Beetz, M. Fast point feature histograms (fpfh) for 3d registration. In: 2009 IEEE International Conference on Robotics and Automation. IEEE; 2009, p. 3212–3217.

[43] Sarode, V, Li, X, Goforth, H, Aoki, Y, Srivatsan, RA, Lucey, S, et al. Pcrnet: Point cloud registration network using pointnet encoding. arXiv preprint arXiv:190807906 2019;.

[44] Aoki, Y, Goforth, H, Srivatsan, RA, Lucey, S. Pointnetlk: Robust & efficient point cloud registration using pointnet. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019, p. 7163–7172.

[45] Wang, Y, Solomon, JM. Deep closest point: Learning representations for point cloud registration. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. 2019, p. 3523–3532.