Journal of
Electronic Imaging

SPIEDigitalLibrary.org/jei

Affine registration for multidimensional point sets under the framework of Lie group

Lifen Ding
Yaxin Peng
Chaomin Shen
Zhiyu Hu

Affine registration for multidimensional point sets under the framework of Lie group

Lifen Ding
Yaxin Peng
Shanghai University
School of Science, Department of Mathematics
Shanghai 200444, China
E-mail: yaxin.peng@shu.edu.cn

Chaomin Shen
East China Normal University
Department of Computer Science
Shanghai 200241, China

Zhiyu Hu
Shanghai University
Institute of NanoMicroEnergy
Shanghai 200444, China

Abstract. *An affine registration algorithm for multidimensional point sets under the framework of Lie group is proposed. This algorithm studies the affine registration between two data sets, and puts the expectation maximization-iterative closest point (EM-ICP) algorithm into the framework of Lie group, since all affine transformations form a Lie transformation group. The registration is carried out via minimizing an energy functional depending on elements of the affine transformation Lie group. The key point for applying the idea of Lie group is that, during the minimization via iteration, we must guarantee the next iteration step of the transformation is still an element in the same group, starting from an element in a Lie group. Our solution is utilizing the element of Lie algebra to represent that of Lie group near the identity via the exponential map, i.e., we use the first canonical coordinate representation of Lie group. Several comparative experiments between the proposed Lie-EM-ICP algorithm and the Lie-ICP algorithm are performed, showing that the proposed algorithm is more accurate and robust, especially in the presence of outliers. This algorithm can also be generalized to other registration problems in general, provided that desired transformations are within certain Lie group.* © 2013 SPIE and IS&T [DOI: 10.1117/1.JEI.22.1.013022]

1 Introduction

Registration of data sets is a key and difficult issue in computer vision and image processing such as three-dimensional (3-D)-reconstruction, stereo matching, image fusion, and shape recognition.[1–3] The purpose of registration is to find a transformation that best aligns one data set to the other.

The iterative closest point (ICP) algorithm, proposed independently by Besel and Mckay,[4] Chen and Medioni,[5] and Zhang,[6] becomes one of the most popular frameworks for data set registration because of its simplicity, high-speed, and low computational complexity. In fact, it is shown in theory that the ICP algorithm is of polynomial time complexity.[7] With the extensive applications of the ICP algorithm, a large number of variants have been proposed to improve its accuracy, robustness, or expend its application range. Liu[8] developed an algorithm which penalizes the closest point sharing; Sharp et al.[9] used invariant features to the ICP algorithm, which decreased the probability of being trapped in a local minimum; Du and Ying[10–12] took scale factor with lower and upper bounds into account in the registration, which enormously expanded the application range of ICP, while later Du[13] also considered the affine registration problem. Zhu and his coauthors[14,15] introduced the bidirectional distance measurement into the ICP, and also presented a point-to-line metric with bounded scale.

On the other hand, Ying et al.[16,17] introduced Lie group to parameterize the ICP algorithm for $n-D$ data registration, forming a unified framework to analyze the registration algorithms.

For the detailed description of Lie group and corresponding concepts (e.g., Lie algebra), please refer to the Appendix at the end of this paper. Here, we mention that many transformations belong to certain Lie groups (see Table 1). This inspires us to use the view point of Lie group to handle the registration problem.

In fact, Lie group has been already utilized in computer vision and pattern recognition. For example, it was applied to estimate the camera motion,[18] model the rigid body motion,[19] and reconstruct the spatial and motion properties from images.[20,21] Its role in data set registration, however, is still very little. Hence, in this paper, we pay more attention to the usage of Lie group in registration; in particular, we will study the group of affine transformations and its application in registration.

Paper 12320 received Aug. 18, 2012; revised manuscript received Nov. 28, 2012; accepted for publication Jan. 16, 2013; published online Feb. 7, 2013.

0091-3286/2013/$25.00

Table 1 Fundamental transformations and the corresponding Lie groups.

Registration	Transformation	Symbol	Type of the elements of the Lie group
Nonrigid	Affine	$\mathrm{GL}(n)\ltimes T(n)$	(nonsingular matrix, $n-D$ vector)
	Linear	$\mathrm{GL}(n)$	Nonsingular matrix
	Shear	$\mathrm{SUT}(n)$	Upper triangular matrix whose diagonals are 1
	Stretch	$\mathrm{Diag}^{+}(n)$	Diagonal matrix whose entries are positive
Rigid	Rotation or reflection	$\mathrm{O}(n)$	Orthogonal matrix
	Rotation	$\mathrm{SO}(n)$	Orthogonal matrix whose determinant is $+1$
	Translation	$\mathrm{T}(n)$	$n-D$ vector

In real registration problems, noise or outliers must be taken into account. Unfortunately, most ICP-based algorithms become less powerful in the presence of noise. To overcome this shortcoming, many probabilistic methods were developed. Chui and Rangarajan proposed a soft assignment technique called thin-plate spline (TPS) robust point matching (RPM) algorithm[22–24]; Zheng and Doermann formulated point matching as an optimization problem to preserve local neighborhood structures during match.[25] Other probobilistic models were also proposed in Refs. 26, 27, and 28. In particular, the expectation maximization (EM)-ICP method introduced by Granger and Pennec[27] has become a well-known method for registration of data sets with noise and/or outliers. It also proposed a coarse-to-fine approach based on the deterministic annealing scheme to avoid local minima, and achieved an optimal accuracy.

In our Lie group approach, we adopt the idea of EM-ICP, thus we term our algorithm as Lie-EM-ICP. Our contributions in this paper are twofold. One is that we generalize the idea of Ying et al.[16,17] to the case that noise and/or outliers exist; more important, we present a unified Lie group framework to the EM-ICP algorithm.

The Lie-EM-ICP is carried out via minimizing an energy functional depending on elements of affine transformation Lie group. The key point for applying the framework of Lie group is that, starting from an element in a Lie group, we must guarantee the next iteration step of transformation is still an element in this group. Our solution is utilizing the element of Lie algebra to represent that of Lie group near the identity via the exponential map, i.e., we use the first canonical coordinate representation of Lie group.

The remainder of this paper is organized as follows. In Sec. 2, we briefly describe the general data set registration model, then reproduce the EM-ICP algorithm from an energy minimization perspective. In Sec. 3, we introduce our Lie-EM-ICP, with our focus on the M-step over a Lie group, which is a nonlinear optimization problem. Section 4 is devoted to several comparative and practical numerical experiments, and this paper is concluded in Sec. 5. We also describe the basic idea of Lie group and its corresponding concepts in the Appendix.

2 Registration and the EM-ICP Algorithm

2.1 *Registration Problem*

Let $S=\{s_i|i=1,\cdots,N_s\}$ be a scene point set in n-dimensional space $\mathbb{R}^n$, $M=\{m_j|j=1,\cdots,N_m\}$ be a model point set in $\mathbb{R}^n$, and $\mathcal{T}=\{T\}$ be a transformation set in $\mathbb{R}^n$. The aim of registration is to register point sets S and M by selecting a suitable element T in $\mathcal{T}$, such that the transformed point set $T\cdot S=\{T\cdot s_i|i=1,\cdots,N_s\}$ is "close" to the model point set M.

In order to mathematically define the "closeness" between two point sets, we need to define three distances step by step, i.e., (1) distance between two points; (2) distance between a point and a set; and (3) distance between two sets. The "closeness" can be measured using the distance between two sets.

1. Distance between two points $T\cdot s_i\in T\cdot S$ and $m_j\in M$.

   In this paper we adopt the Mahalanobis distance:
   $$d_1(T\cdot s_i,m_j)=\frac{1}{2\sigma^2}\|T\cdot s_i-m_j\|^2,$$
   where σ is a parameter.
2. Distance between a point $T\cdot s_i$ and a set M.

   To define this, we introduce a match number w_{ij} between $T\cdot s_i\in T\cdot S$ and $m_j\in M$, which describes the probability that $T\cdot s_i$ matches m_j when T is given. Thus, w_{ij} should satisfy $w_{ij}\geq 0$, and $\sum_{j=1}^{N_m}w_{ij}=1$.

   If w_{ij} is defined, the distance between a point $T\cdot s_i$ and a set M can be described as
   $$d_2^W(T\cdot s_i,M)=\sum_{j=1}^{N_m}w_{ij}\cdot d_1(T\cdot s_i,m_j)=\sum_{j=1}^{N_m}w_{ij}\frac{\|T\cdot s_i-m_j\|^2}{2\sigma^2}.$$
3. Distance of two sets $T\cdot S$ and M.

   Once the distance between a point and a set is defined, the distance between two sets can be described as
   $$d_3^W(T\cdot S,M)=\sum_{i=1}^{N_s}d_2^W(T\cdot s_i,M)=\sum_{i=1}^{N_s}\sum_{j=1}^{N_m}w_{ij}\frac{\|T\cdot s_i-m_j\|^2}{2\sigma^2}. \tag{1}$$

   If w_{ij} is known, our aim for registration is to find a suitable $T\in\mathcal{T}$, such that Eq. (1) reaches its minimum.

2.2 EM-ICP Algorithm

In what follows, we reproduce the EM-ICP algorithm from an energy minimization perspective, i.e., our approach is different from the original one. Later we show that the final iterative formulas between ours and the original one are the same.

Note that in Eq. (1) a suitable w_{ij} should also be determined. In what follows, we analyze the properties that w_{ij} should possess. If we view the event that sets $T \cdot S$ and M match each other as a random event, and w_{ij} is the probability that point $T \cdot s_i$ matches point m_j, then the corresponding entropy term is

$$\text{Entropy}(w_{ij}) = -\sum_{i,j} w_{ij} \log_2 w_{ij}.$$

Entropy indicates the randomness of the random event. By the second law of thermodynamics, all dynamic systems will evolute towards the direction where the entropy increases. Therefore, if we have no prior assumption on w_{ij}, the entropy of our system should be as large as possible. As a consequence, we wish the negative entropy as small as possible. This can be done by adding the negative entropy to the distance Eq. (1) during the minimization. Thus, we have the energy functional

$$E(T, W) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} + \beta \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \ln w_{ij}, \tag{2}$$

where β is a balancing parameter. Without loss of generality, $\log_2$ has been changed by $\ln$, and β can be set as 1, since β can be injected to the variance σ^2. Remember also that the minimization of Eq. (2) has constraints that

$$w_{ij} \geq 0, \quad \sum_{j=1}^{N_m} w_{ij} = 1, \quad \forall\, i = 1, \cdots, N_s. \tag{3}$$

To solve the minimization problem of Eq. (2) with constraints Eq. (3), the Lagrange multiplier method can be used to change the problem as an unconstrained one. Thus, the unconstrained energy functional becomes

$$E(T, W, \lambda) = \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} + \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \ln w_{ij} + \sum_{i=1}^{N_s} \lambda_i \left(\sum_{j=1}^{N_m} w_{ij} - 1 \right). \tag{4}$$

Therefore,

$$(T^*, W^*, \lambda^*) = \arg \min_{T, W, \lambda} E(T, W, \lambda) \tag{5}$$

is the desired transform T^*, match number $W^* = (w_{ij}^*)$ and $\lambda^* = (\lambda_i^*)$.

In order to obtain W^*, we differentiate Eq. (4) with respect to w_{ij}. Thus,

$$0 = \frac{\partial E}{\partial w_{ij}} = \frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} + \ln w_{ij} + 1 + \lambda_i,$$

then

$$w_{ij} = \exp[-(1 + \lambda_i)] \cdot \exp\left(-\frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} \right).$$

In order to obtain λ^*, we differentiate Eq. (4) with respect to λ_i. Thus,

$$0 = \frac{\partial E}{\partial \lambda_i} = \sum_{j=1}^{N_m} w_{ij} - 1.$$

Putting the expression of w_{ij} into the above equality, we have

$$\exp[-(1 + \lambda_i)] = 1 / \sum_{j=1}^{N_m} \exp\left(-\frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} \right).$$

Putting the above equality to the expression of w_{ij} again, we have

$$w_{ij} = \frac{\exp\left(-\frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} \right)}{\sum_{j=1}^{N_m} \exp\left(-\frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2} \right)}. \tag{6}$$

Obviously, Eq. (6) satisfies the non-negative and sum-to-1 conditions in Eq. (3).

Equation (6) is equivalent to the E-step in the usual EM-ICP algorithm, i.e., we can obtain the desired match matrix $W = (w_{ij})$ via Eq. (6), once T is determined.

Now we focus on determining T^*. It can be calculated from $\frac{\partial E}{\partial T} = 0$. When the matrix W is determined, T^* can be obtained by

$$\min_{T \in \mathcal{T}} \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \frac{\|T \cdot s_i - m_j\|^2}{2\sigma^2}, \tag{7}$$

i.e., the M-step in the EM-ICP algorithm.

Thus, from an initial value T^0 of T, w_{ij}^1 can be calculated via the E-step; then, having w_{ij}^1 fixed, T^1 can be calculated via the M-step. The iteration will go on by alternatively calculating $\{w_{ij}^k\}$ and $\{T^k\}$ via E-step and M-step, respectively, for $k = 1, 2, \cdots$, until the stopping criteria are met. Thus, we acquire the approximated desired transform T^* and corresponding desired match matrix W^*.

3 Lie-EM-ICP Algorithm

As mentioned in the introduction, Lie group is a suitable frame to address the registration problem. In this section, we show in detail how Lie group method is used in the M-step.

3.1 *Algorithm Statement*

Here we assume that the transformation set $\mathcal{T}$ is a Lie transformation group. For simplicity, we assume that G is an affine transformation group (or its subgroup) in $\mathbb{R}^n$. We denote G as $G = H \ltimes T(n)$, where $T(n)$ is the translation part, and H is a Lie subgroup without translation, i.e., H is a general linear group $GL(n)$ (or its subgroup). So each $g \in G$ can be expressed as $g = (h, t)$ with $h \in H$ and $t \in T(n)$, and the way of transformation is $g \circ x = h \cdot x + t$ for all $x \in \mathbb{R}^n$.

Now we change our notation of the previous section. We use $g \in G$, instead of $T \in \mathcal{T}$. The formulas in Sec. 2 are rewritten in the new notation below.

E-step: Eq. (6) turns to be

$$w_{ij} = \frac{\exp\left(-\frac{\|h \cdot s_i + t - m_j\|^2}{2\sigma^2}\right)}{\sum_{j=1}^{N_m} \exp\left(-\frac{\|h \cdot s_i + t - m_j\|^2}{2\sigma^2}\right)}. \tag{8}$$

M-step: Eq. (7) turns to be

$$\min_{g \in G} \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \frac{\|g \circ s_i - m_j\|^2}{2\sigma^2}$$
$$= \min_{h \in H, t \in T} \sum_{i=1}^{N_s} \sum_{j=1}^{N_m} w_{ij} \frac{\|h \cdot s_i + t - m_j\|^2}{2\sigma^2}. \tag{9}$$

Then, our aim is to find h, t by minimizing Eq. (9).

Below only the M-step is concerned, since the E-step is merely a simple computation using Eq. (8).

3.2 *Iterative Formula for M-Step*

If the element of Lie group is used as an argument in the functional, we may face the situation that the iterated element may not be an element in the Lie group, i.e., a small perturbation may cause the perturbed element falling outside this Lie group. For example, if an element from an orthonormal group with determinant 1 is iterated via a small perturbation, the result may no longer be an orthonormal matrix. Similarly, an element in a linear transformation group should be a nonsingular matrix; but after a small perturbation, it can possibly become a singular matrix. So the nonlinearity of the Lie group will cause difficulty for the variational computation Eq. (9). To prevent this, we introduce the Lie group parametric representation. The linearity of Lie algebra will guarantee that the element of next iteration step still lies in the Lie group.

3.2.1 *Computation of the translation part t*

Taking the partial derivative of Eq. (9) with respective to t, and letting the derivative be zero, we have

$$t = \frac{\sum_{i,j} w_{ij}(m_j - h \cdot s_i)}{\sum_{i,j} w_{ij}} = \frac{\sum_{i,j} w_{ij} m_j}{N_s} - h \cdot \frac{\sum_{i,j} w_{ij} s_i}{N_s}.$$

Therefore,

$$t = \mu_M - h \cdot \mu_S, \tag{10}$$

where

$$\mu_M = \frac{\sum_{i,j} w_{ij} m_j}{N_s}, \qquad \mu_S = \frac{\sum_{i,j} w_{ij} s_i}{N_s} \tag{11}$$

are the weighted centroids of M and S, respectively.

3.2.2 *Computation of the linear part h*

Once t is determined by Eq. (10), the minimization problem in Eq. (9) can be further simplified as follows:

$$h = \arg\min_{h \in H} \sum_{i,j} \|w_{ij} \cdot \tilde{m}_j - h \cdot \tilde{s}_i\|^2, \tag{12}$$

where

$$\tilde{m}_j = m_j - \mu_M, \qquad \tilde{s}_i = s_i - \mu_S.$$

Assume that the dimension of H is N. According to the first canonical coordinate representation of the Lie group, i.e., using the element of Lie algebra to represent the element of Lie group near the identity via the exponential map, each h^{k+1} can be locally calculated from h^k by

$$h^{k+1} = h^k \exp\left(\sum_{m=1}^{N} a_m^k e_m\right),$$

where $\{e_1, e_2, \ldots, e_N\}$ is a given basis of the Lie algebra $\mathfrak{h}$ of Lie group H, and $a_m^k \in \mathbb{R}$.

Then, solving the objective function in Eq. (12) becomes minimizing the following expression:

$$\varepsilon(a) = \sum_{i,j} w_{ij} \cdot \left\| \tilde{m}_j - h^k \exp\left(\sum_{m=1}^{N} a_m e_m\right) \tilde{s}_i \right\|. \tag{13}$$

for $a = (a_1, a_2, \cdots, a_N)^T$.

Then the key step of updating transformation h is turned into

$$a^* = \arg\min_{a \in \mathbb{R}^N} \varepsilon(a). \tag{14}$$

From the first-order optimality condition

$$\frac{\partial \varepsilon(a)}{\partial a_n} = 0, \qquad n = 1, \ldots, N,$$

we have that

$$0 = \frac{\partial}{\partial a_n} \sum_{i,j} w_{ij} \cdot \left\| \tilde{m}_j - h^k \exp\left(\sum_{m=1}^{N} a_m e_m\right) \tilde{s}_i \right\|^2$$
$$= \sum_{i,j} 2 w_{ij} \left\langle \frac{\partial}{\partial a_n} \left[h^k \exp\left(\sum_{m=1}^{N} a_m e_m\right) \tilde{s}_i - \tilde{m}_j \right], \right.$$
$$\left. h^k \exp\left(\sum_{m=1}^{N} a_m e_m\right) \tilde{s}_i - \tilde{m}_j \right\rangle. \tag{15}$$

Let $u = \sum_{m=1}^{N} a_m e_m$, we have the following approximation expressions up to the first order of u, if $\|u\| \ll 1$:

$$h\,\exp(u) \approx h + h \cdot u \frac{\partial}{\partial a_n} \exp(u) \approx e_n + \frac{1}{2}(e_n \cdot u + u \cdot e_n).$$

Therefore, Eq. (15) can be approximated by

$$\begin{aligned} 0 \approx & \sum_{i,j} w_{ij} \langle h^k \cdot e_n \cdot \tilde{s}_i, h^k \tilde{s}_i - \tilde{m}_j \rangle \\ & + \sum_{m=1}^{N} \left[\sum_{i,j} w_{ij} \langle h^k \cdot e_n \cdot \tilde{s}_i, h^k \cdot e_m \cdot \tilde{s}_i \rangle \right. \\ & \left. + \sum_{i,j} w_{ij} \langle \frac{1}{2} h^k \cdot (e_n \cdot e_m + e_m \cdot e_n) \cdot \tilde{s}_i, h^k \tilde{s}_i - \tilde{m}_j \rangle \right] a_m. \end{aligned}$$

Thus, we can get the approximate optimal solution $a^k = (a_1^k, a_2^k, \ldots, a_N^k)^T$ of Eq. (14) by solving the linear equations

$$Ca = -b, \tag{16}$$

where $C = (c_{nm})_{N \times N}$ and $b \in \mathbb{R}^N$ are given by

$$c_{nm} = \sum_{i,j} w_{ij} \mu_{mi}^T \mu_{ni} + \sum_{i,j} w_{ij} \gamma_{ij}^T \nu_{nmi}, \tag{17}$$

$$b_n = \sum_{i,j} w_{ij} \gamma_{ij}^T \mu_{ni}, \tag{18}$$

where

$$\begin{aligned} \mu_{mi} &= h^k \cdot e_m \cdot \tilde{s}_i, \\ \nu_{nmi} &= \frac{1}{2} h^k \cdot (e_n \cdot e_m + e_m \cdot e_n) \cdot \tilde{s}_i, \\ \gamma_{ij} &= h^k \cdot \tilde{s}_i - \tilde{m}_j. \end{aligned}$$

Note that from Eqs. (2) and (10), energy Eq. (2) becomes

$$E(W, h) = \sum_{i,j} w_{ij} \frac{\|h \cdot \tilde{s}_i - \tilde{m}_j\|^2}{2\sigma^2} + \sum_{i,j} w_{ij} \ln w_{ij}. \tag{19}$$

Combining the above analysis, we can derive an algorithm based on Lie group representation for registration as algorithm 1. In order to make the initial transformation sufficiently close to the desired solution, the principal component analysis and independent component analysis are applied in initializing the transformation, i.e., the h^0 in our algorithm. Then we iterate and obtain the final fine result.

According to Ref. 28, we also use the annealing scheme on the variance. Here σ_r^2 is the variance of real point noise, which can be estimated by the following formula from Eq. (13) of Ref. 28:

$$\sigma_r^2 = \frac{\sum_{ij} w_{ij} \|m_j - T^* s_i\|^2}{N_s \cdot \dim}.$$

This formula is obtained by the maximum likelihood estimation.[28] However, we choose the initial value σ^2 much bigger than σ_r^2 (we set $\sigma^2 = 10 \cdot \sigma_r^2$). The reason is that we want to find a global minimizer using the simulated annealing method. The annealing coefficient α is usually set as 0.9 to 0.95, and in our case we choose 0.9.

Algorithm 1 The Lie-EM-ICP registration algorithm.

Initialization: $h^0 \in H$, $w_{ij}^0 = 1/N_m$, σ_r^2 and σ^2 are given.

Compute $t^0 = \mu_M^0 - h^0 \cdot \mu_S^0$ using Eqs. (10) and (11).

Repeat: (step k, starting from $k = 0$):

E-Step: Compute w_{ij}^{k+1} using Eq. (8).

M-Step: Update $g^{k+1} = (h^{k+1}, t^{k+1})$, where h^{k+1} is updated using Eq. (12), by solving the linear system Eq. (16), and $t^{k+1} = \mu_M^{k+1} - h^{k+1} \cdot \mu_S^{k+1}$ using Eqs. (10) and (11).

Compute the energy E^{k+1} using Eq. (19).

Annealing: Set $\sigma_{k+1}^2 = \max(\alpha \cdot \sigma_k^2, \sigma_r^2)$, where α is the annealing coefficient.

Until: $\sigma^2 = \sigma_r^2$ and $E^{k+1} \geq E^k$.

In the implementation, we use $\sigma_{k+1}^2 = \max(\alpha \cdot \sigma_k^2, \sigma_r^2)$ to update σ^2 in Eq. (19) for the $k + 1$'th iteration, where k is the iteration step. This will help that σ^2 gradually decreases to the real value. The truncation conditions are $\sigma^2 = \sigma_r^2$ and the energy Eq. (2) does not decrease.

Note that the proposed algorithm can be applied to other types of registration problems, provided that their registration transformations are elements of certain Lie groups. Therefore, the proposed algorithm provides a unified framework for registration problems.

4 Numerical Experiments

In this section, our proposed algorithm is compared with the Lie-ICP algorithm,[17] a well-known registration method. The comparison is divided into three parts: (1) registration for multidimensional data without outliers, (2) registration for multidimensional data with outliers, and (3) further evaluation of the Lie-EM-ICP. In each part, rigid and affine registrations for 3-D cases are performed, and it is easy to generalize the same method to higher-dimension. The algorithm is implemented in Matlab 2011b and tested on the Intel@Core™i5-2435M CPU@2.40 GHz with 2 GB RAM.

4.1 *Comparison for 3-D Data Registration Without Outliers*

In this subsection, the dragon and bunny data sets of Stanford 3-D scanning repository[29] are used for comparison. We use a subsampled dragon version of 1105 points, and a subsampled bunny version of 1889 points. The comparison results are shown in Figs. 1 and 2.

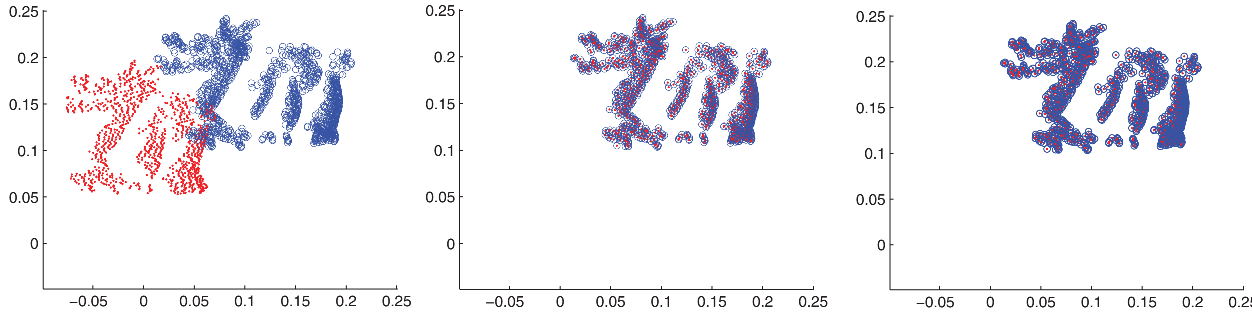
In each figure, the first column is the original configuration: one scene data set and one model data set. For the purpose of validating the registration result, the two data sets are constructed from one data set. To be more specific, given the scene data set S, the model data set M is obtained by conducting a transformation T_{gr} on the scene set. T_{gr} can be used as a groundtruthing for registration, since it is known in advance. In the dragon data set, the transformation includes a rotation with axis $(-0.0431 \quad 0.5016 \quad 0.0885)$, angle 29.29 deg, and a translation $(0.12 \quad 0.05 \quad 0.05)$. In the bunny data set, the transformation includes a rotation with axis $(-0.0534 \quad 0.4425 \quad -0.0602)$, angle 25.7728 deg, a scale factor $s = 1.25$, and a translation $(0.15 \quad 0.05 \quad 0.05)$; the second column is the registration result of our proposed algorithm, and the third column is the registration result of the Lie-ICP algorithm. Figure 3(a) and 3(b) shows a zoom-in of the registration results in Fig. 1(b) and 1(c), respectively.

Observations from Figs. 1 to 2 show that the proposed and the Lie-ICP algorithms have the same efficiency and accuracy for registration without outliers.

4.2 Comparison for 3-D Data Registration with Outliers

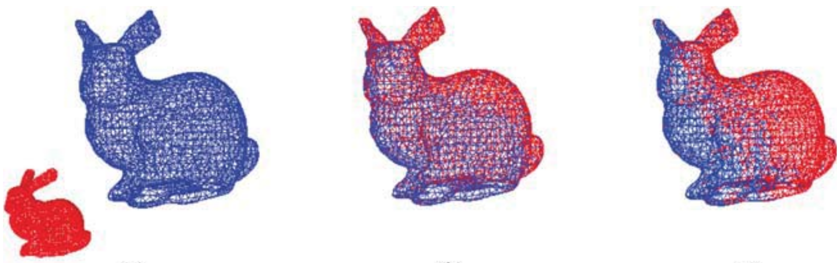
In this subsection, the scene and the model data sets are selected in the same way as Sec. 4.1. But some random outlier point are added into one or both data sets, as shown in the first column in Figs. 4 and 5 (dragon and bunny, respectively). In the dragon data set, 50 (i.e., $50/1105 \approx 5\%$) outliers are added randomly to both the scene and model data sets, respectively, from a normal distribution. In the bunny data set, 180 (i.e., $180/1889 \approx 10\%$) outliers are randomly added to the model data set from a normal distribution.

Both results are displayed and analyzed in Figs. 4 to 7, and corresponding parameters are given in Tables 2 and 3.



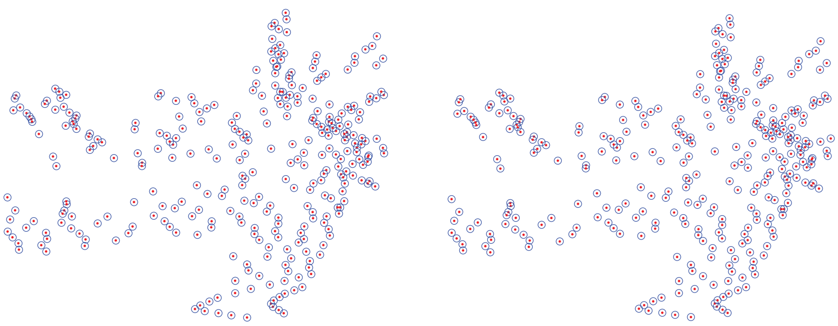
(a) (b) (c)

Fig. 1 Comparison of registration for the 3-D dragon data set without outliers. The scene set is shown in red, and the model set is shown in blue. (a) Initial configuration. (b) Result of our algorithm. (c) Result of the Lie-ICP algorithm.



(a) (b) (c)

Fig. 2 Comparison of registration for the 3-D bunny data set without outliers. The scene set is shown in red, and the model set is shown in blue. (a) Initial configuration. (b) Result of our algorithm. (c) Result of the Lie-ICP algorithm.



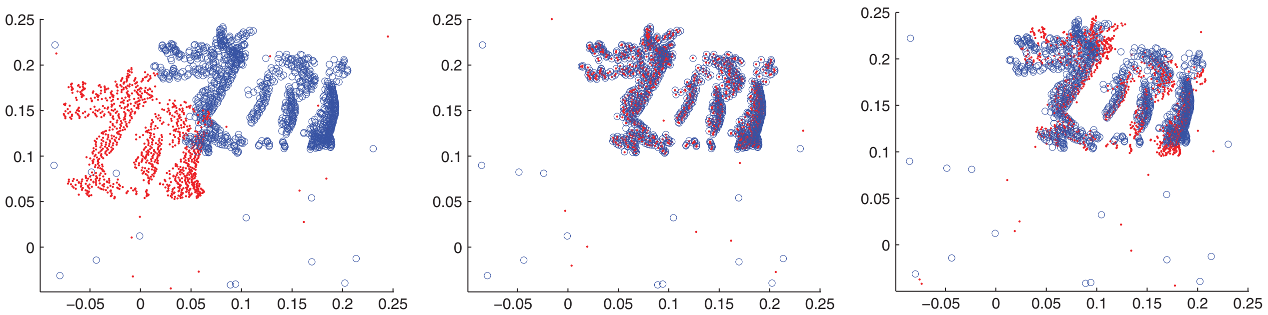
(a) (b)

Fig. 3 Zoom-in of the registration results in Fig. 1. (a) Our algorithm. (b) The Lie-ICP algorithm.

The second and the third columns in Figs. 4 and 5 are the registration results of the proposed algorithm and Lie-ICP, respectively, as before. Figure 6(a) and 6(b) gives a zoom-in of the registration result of the proposed algorithm and the Lie-ICP algorithm in Fig. 4, respectively. The root mean square (RMS) error is displayed in Fig. 7.

Figures 4 to 7 show that, in the presence of outliers, the proposed algorithm is still robust, while the Lie-ICP algorithm becomes less powerful. Tables 2 and 3 show that the difference of our algorithm between the estimated value and groundtruthing T_{gr} is smaller, compared to the Lie-ICP algorithm; the computation times of the two algorithms are comparable.

To test the robustness of the Lie-EM-ICP algorithm under various levels of outliers, three different amounts of random outliers are also added to the scene and model data sets. We generate outliers randomly from a normal distribution with 0.1-mean and 0.1-standard deviation. The ratios (number of outliers to that of original data) are from 0.02 to 0.14; in other words, the numbers of outliers points are from 30 to 150. The original data sets with three different amounts of outliers (with ratios of 0.02, 0.08, and 0.14) are shown in the first column of Fig. 8. The registration result of the proposed and the Lie-ICP algorithms are shown in the second and third columns of Fig. 8, respectively. The corresponding RMS errors with respect to difference number of outliers



(a) (b) (c)

Fig. 4 Comparison of registration for the 3-D dragon data set with outliers are added randomly to both the scene and model data sets, respectively. The scene set is shown in red, and the model set is shown in blue. (a) Initial configuration. (b) Result of our algorithm. (c) Result of the Lie-ICP algorithm.

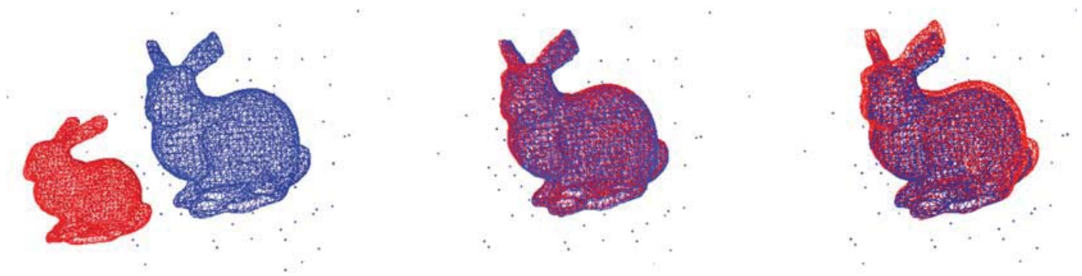

(a) (b) (c)

Fig. 5 Comparison of registration for the 3-D bunny data set with outliers. The scene set is shown in red, and the model set is shown in blue. (a) Initial configuration. (b) Result of our algorithm. (c) Result of the Lie-ICP algorithm.

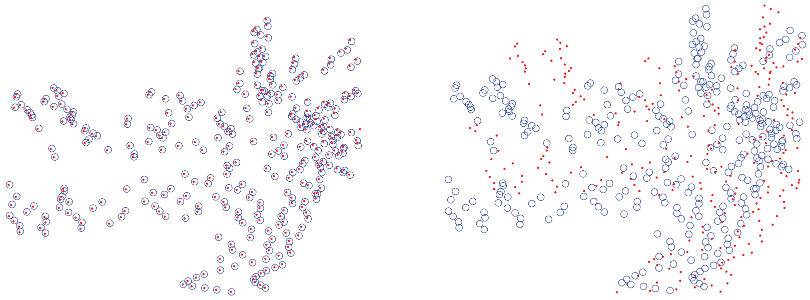
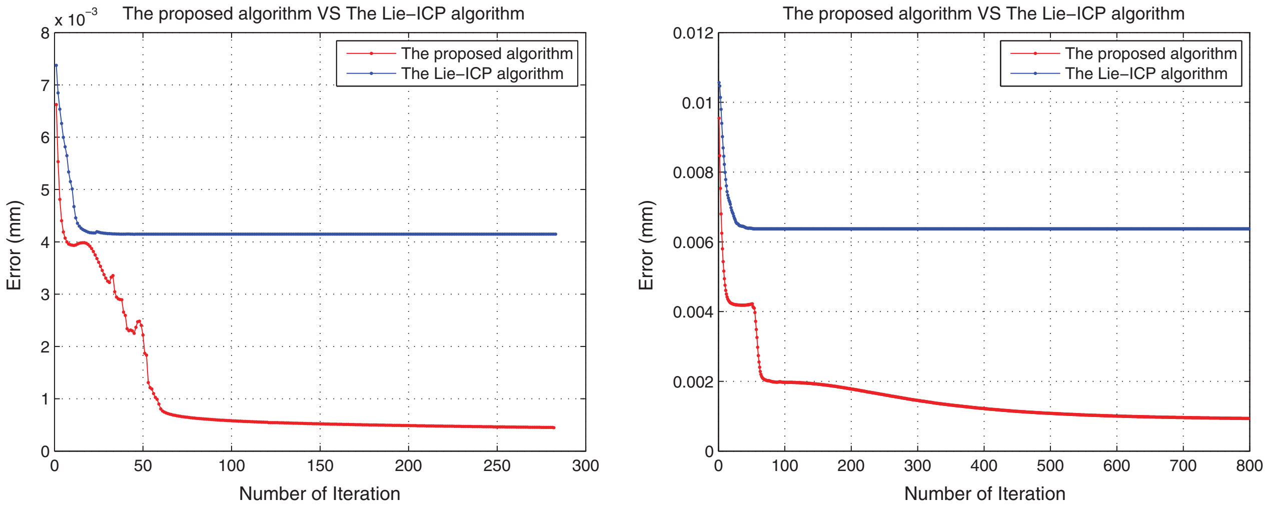

(a) (b)

Fig. 6 Zoom-in of the registration results in Fig. 4. (a) The proposed algorithm. (b) The Lie-ICP algorithm.



(a) (b)

Fig. 7 RMS errors of the registration. (a) The rigid dragon data registration in Figs. 4. (b) The affine bunny data registration in Fig. 5.

Table 2 Registration result (dragon) in Fig. 4.

	Groundtruthing	Initial transformation	The proposed algorithm	The Lie-ICP algorithm
Rotation axis	[−0.0431 0.5016 0.0885]	[0.1459 0.4027 0.0144]	[−0.0354 0.5051 0.0955]	[0.0721 0.4684 0.1216]
Rotation angle	29.29	24.5518	29.5202	28.0358
Translation	[0.12 0.05 0.05]	[0.0908 0.0509 0.0464]	[0.1192 0.0499 0.0511]	[0.1077 0.0509 0.0572]
Distance [Eq. (1)]	—	—	0.89	8.2
Time (s)	—	—	30.27	27

Table 3 Registration result (bunny) in Fig. 5.

	Groundtruthing	Initial transformation	The proposed algorithm	The Lie-ICP algorithm
Rotation axis	[−0.0534 0.4425 −0.0602]	[−0.0896 0.3636 −0.0870]	[−0.0675 0.4486 −0.0581]	[−0.0673 0.3750 −0.0742]
Rotation angle	25.7728	22.0271	26.2065	22.2408
Scale	1.25	1.3724	1.2507	1.3223
Translation	[0.15 0.05 0.05]	[0.1600 0.0406 0.0457]	[0.1500 0.0514 0.0491]	[0.1551 0.0470 0.0477]
Distance [Eq. (1)]	—	—	0.85	6.5
Time (s)	—	—	15.74	30.92

are displayed in Fig. 9. It is clear that the proposed algorithm outperforms the Lie-ICP algorithm, and is more robust in all examples.

4.3 *Further Evaluation of Lie-EM-ICP*

In this subsection, the Chui-Rangarajan synthesized data set is used to demonstrate the robustness of our algorithm with respect to initial transformation parameters under a large number of runs.

Given the pair of scene and model sets, we conduct our algorithm and the Lie-ICP algorithm 10 times. Each time, 10% random outliers are added to the data set. Therefore, the stochastic outliers will affect the result of coarse registration, i.e., our estimation initial transformation parameters.

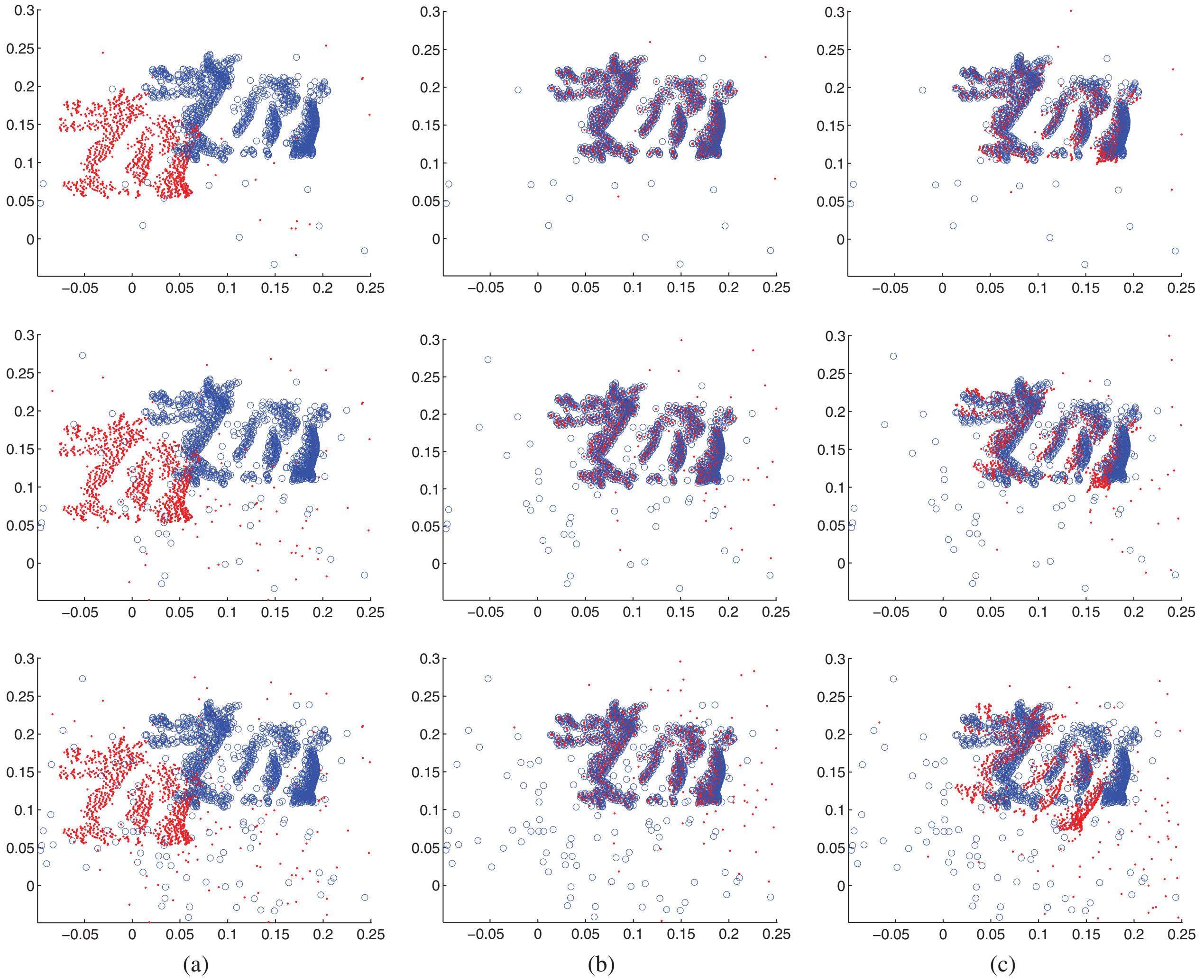


Fig. 8 Comparison of registration with respect to different amounts of outliers. (a) Original data sets with different amounts of outliers added (first row: 2%; second row: 8%; and bottom row: 14%). (b) Result of our algorithm. (c) Result of the Lie-ICP algorithm.

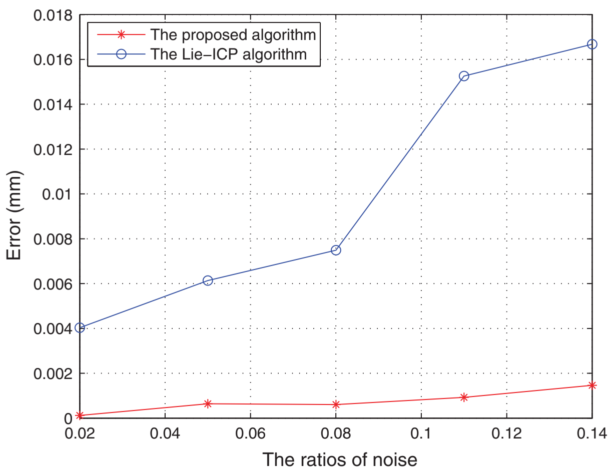


Fig. 9 Comparison of the RMS errors between our algorithm and the Lie-ICP algorithm with respect to different ratios of outliers in Fig. 8.

To avoid possible bias bought by a single example, ten registration examples made by different groundtruthing T_{gr} are displayed in Fig. 10. Boxplots of the RMS errors for the pair of databases are listed in Fig. 11, in which the left and right figures illustrate the results of the Lie-ICP algorithm and the Lie-EM-ICP algorithms, respectively. Figure 11(a) is the pair for the scale case and Fig. 11(b) is for the rotation case.

Three important observations can be obtained from Fig. 11. First, in each pair the mean RMS error of ten samples is lower in the Lie-EM-ICP algorithm than that in the Lie-ICP algorithm, which means that our algorithm is more accurate. Second, our algorithm shows lower median and variability than the Lie-ICP algorithm, which indicates that our algorithm outperforms the Lie-ICP in terms of robustness. Third, although the mean RMS errors of both algorithms increase when deformation is large, our increasing rate of the RMS error is lower. In general, the proposed algorithm appears to be more accurate and robust than the Lie-ICP.

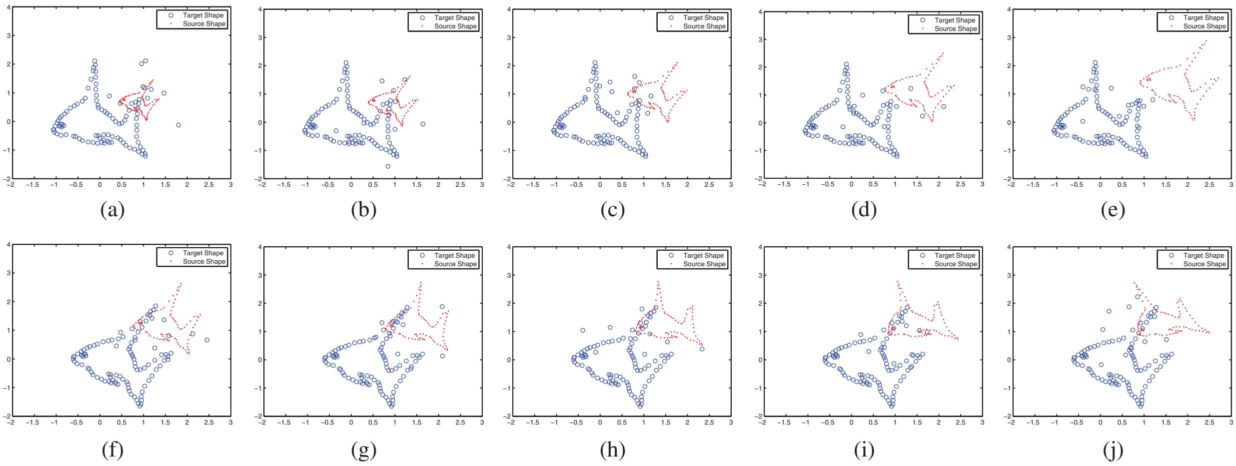


Fig. 10 Different transformation examples for the Chui-Rangarajan synthesized data set with 10% outliers. The scene set is shown in red, and the model set is shown in blue. Top row: fishes under stretch, in which the scales are from 0.4 to 0.8. Bottom row: fishes under rotation, in which the degrees of rotation are from 10 to 50.

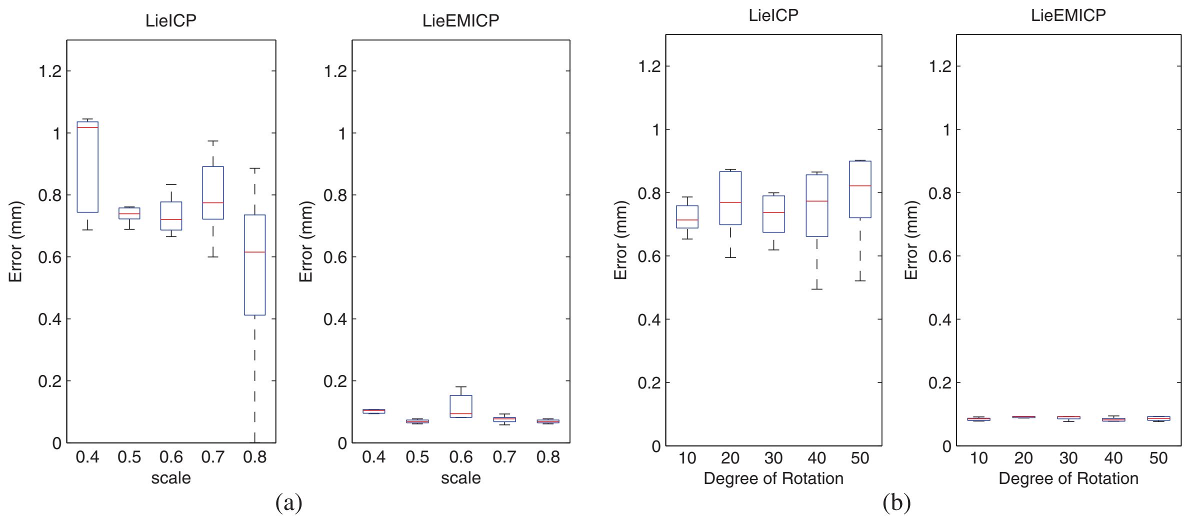


Fig. 11 RMS errors of the registration for examples in Fig. 10. (a) Comparison of the RMS errors between our algorithm and the Lie-ICP algorithm under stretch. (b) Comparison of the RMS errors between our algorithm and the Lie-ICP algorithm under rotation.

5 Conclusion

In this paper, the affine registration of multidimensional data sets is studied by combining the Lie group parametric representation with the EM-ICP algorithm. This unified registration algorithm is named as the Lie-EM-ICP algorithm.

This paper has three contributions. First, we have reproduced the EM-ICP algorithm in a lucid way, from the energy minimization perspective. Other two contributions belong to the Lie-EM-ICP algorithm, i.e., (1) we utilize the Lie group parametric representation to facilitate the variational computation, thanks to the linearity of the Lie algebra; and (2) we improve the robustness to outliers by combining the EM-ICP algorithm.

The Lie-EM-ICP algorithm is validated by numerical experiments, showing that the proposed algorithm is more accurate and robust than the Lie-ICP algorithm in the presence of outliers. Moreover, the Lie-EM-ICP is also robust when the number of outliers increases, while its registration speed is comparable to that of the Lie-ICP algorithm.

The proposed Lie group framework can also be generalized to other Lie transformation group registrations, not restricted to the affine registration. In this paper, we have discussed the affine registration, therefore we use the basis $\{e_1, e_2, \ldots, e_N\}$ for the Lie algebra $gl(n, R)$ of Lie group $GL(n, R)$. One generalization is, for example, if some subgroups of affine registration which only contain stretch and shear are considered, we only need to study the corresponding subalgebras of Lie algebra $gl(n, R)$, since they belong to the part of affine group.

Appendix

Lie group is a group (as well as a differential manifold) that the group composition and inverse operations are smooth.

Lie algebra is a vector space on which a Lie bracket is defined. For a Lie group G, the tangent vector space T_eG of G at the identity element e can be endowed with the structure of Lie algebra, so it is called the Lie algebra of G and often denoted by $\mathfrak{g}$. The Lie algebra of Lie group plays an important role in Lie group theory in that it encodes many properties of Lie group. There is a local diffeomorphism exp, named the exponential mapping, from a neighborhood N_0 of the zero element 0 of $\mathfrak{g}$ to some neighborhood N of the unit element e of G. Particularly, if G is of N dimension, then every element of G can be expressed as the form $\exp(\sum_{i=1}^{N} a_i e_i)$, where $a_i \in \mathbb{R}$ and $\{e_1, e_2, \ldots, e_N\}$ is a given basis of $\mathfrak{g}$. The vector $(a_1, a_2, \ldots, a_N)^T \in \mathbb{R}^N$ is called the first canonical coordinate with respect to basis $\{e_1, e_2, \ldots, e_N\}$. The exponential mapping is a local diffeomorphism. Lie groups occur naturally in the study of groups of matrices, where the group operation is just matrix multiplication. Two important Lie groups of matrices for data set registrations are the general linear group and the special orthogonal group. Affine transformation on $\mathbb{R}^n$ is an invertible transformation that can be characterized with a nonsingular matrix transformation plus a translation vector. Concretely, to each affine transformation A, there corresponds a pair (B, T), where the matrix $B \in GL(n, R)$ and the vector $T \in R^n$, such that $Ax = Bx + T$ for all $x \in \mathbb{R}^n$. The set of all affine transformations constitute a Lie group, denoted by Aff(n), with the composition operation $(B_1, T_1) \cdot (B_2, T_2) = (B_1B_2, B_1T_2 + T_1)$. We know that the set of all rotation transforms corresponds to $\mathrm{SO}(N)$ whose elements are orthonormal matrices with determinant $+1$.

We would like to express the above idea with more detail here. We use the rotation as an example. For the energy functional whose argument is the element of Lie group (i.e., orthonormal matrix in this case), if we do the variation directly on an orthonormal matrix, the result of an orthonormal element plus a small perturbation (even if it is still an orthonormal matrix) may no longer be an orthonormal matrix. We thus have to find an optimal solution in the Lie group, which is the image of the exponential map of a suitable element in the Lie algebra $\mathfrak{g}$ defined below.

The Lie algebra $\mathfrak{g}$ of a Lie matrix group G is a first order approximation of Lie group near the identity of the Lie group. Lie algebra is a linear space, instead of a manifold for the case of Lie group. For example, let $B(t)$ be a 1-parameter subgroup of $\mathrm{SO}(N)$, then $B(t) \cdot B(t)^T = I$. Since the tangent vector of this 1-parameter subgroup at I is the element of Lie algebra of $\mathrm{SO}(N)$, i.e., $\frac{dB(t)}{dt}|_t = A \in \mathrm{SO}(N)$. Differentiating $B(t) \cdot B(t)^T = I$ with respect to t and let $t = 0$, we get $A + A^T = 0$. That means the element in Lie algebra of $\mathrm{SO}(N)$ is anti-symmetric. Near the identity of the Lie matrix group, for $A \in g$, there exists an exponential map $\exp: \mathfrak{g} \to G$,

$$\exp: A \to \exp(A) = e^A = 1 + A + \frac{1}{2!}A^2 + \frac{1}{3!}A^3 + \cdots .$$

The exponential map establishes a 1-1 correspondence between the elements of Lie group and Lie algebra near the identity.

If we express the element in Lie group by the element of Lie algebra using the exponential map, the Lie algebra of the orthonormal Lie group is the set of anti-symmetric matrices. We know that an anti-symmetric matrix plus a small anti-symmetric matrix is still an anti-symmetric matrix. Thus, we can first find the minimizer within the Lie algebra via some iteration method, then the minimizer in Lie group can be obtained by the exponential map which maps the minimizer in Lie algebra to that in Lie group.

Interested readers please refer to Refs. 30–32.

Acknowledgments

The authors would like to thank the editor and the anonymous reviewers for their helpful comments and suggestions. This work is supported by the 973 Programme (2011CB707104), the National Natural Science Foundation of China (11101260, 61005002, 10971066, and 61273298), the Shanghai Committee of Science and Technology Funds (10520710400), and the Discipline Project at the corresponding level of Shanghai (A.13-0101-12-005).

References

1. I. Carlbom, D. Terzopoulos, and K. Harris, "Computer-assisted registration, segmentation, and 3D reconstruction from images of neuronal tissue sections," *IEEE Trans. Med. Imag.* **13**(2), 351–362 (1994).
2. H. Chen and B. Bhanu, "Human ear recognition in 3D, "*IEEE Trans. Pattern Anal. Mach. Intell.* **29**(4), 718–737 (2007).
3. V. Radhika et al., "Robust stereo image matching for spaceborne imagery," *IEEE Trans. Geosci. Remote Sens.* **45**(9), 2993–3000 (2007).
4. P. Besl and N. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.* **14**(2), 239–256 (1992).
5. Y. Chen and G. Medioni, "Object modeling by registration of multiple range images," *Image Vis. Comput.* **10**(3), 145–155 (1992).
6. Z. Zhang, "Iterative point matching for registration of free-form curves and surfaces," *Int. J. Comput. Vision* **13**(2), 119–152 (1994).
7. D. Spielman and S.-H. Teng, "Smoothed analysis: an attempt to explain the behavior of algorithms in practice," *Commun. ACM* **52**(10), 76–84 (2009).
8. Y. H. Liu, "Penalizing closest point sharing for automatic free form shape registration," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(5), 1058–1064 (2011).
9. G. C. Sharp, S. W. Lee, and D. K. Wehe, "ICP registration using invariant features," *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(1), 90–102 (2002).
10. S. H. Ying et al., "A scale stretch method based on ICP for 3D data registration," *IEEE Trans. Autom. Sci. Eng.* **6**(3), 559–565 (2009).
11. S. Du et al., "An extension of the ICP algorithm considering scale factor," in *Proc. of the International Conference on Image Processing, (ICIP 2007)*, pp. 193–196, IEEE, San Antonio, Texas (2007).
12. S. Y. Du et al., "Scaling iterative closest point algorithm for registration of m-D point sets," *J. Visual Commun. Image Represent.* **21**(5–6), 442–452 (2010).
13. S. Y. Du et al., "Affine iterative closest point algorithm for point set registration," *Pattern Recogn. Lett.* **31**(9), 791–799 (2010).
14. J. H. Zhu et al., "Point-to-line metric based iterative closest point with bounded scale," in *Proc. of the 4th IEEE Conf. on Industrial Electronics and Applications (ICIEA)*, pp. 3032–3037, IEEE, Xi'an, China (2009).
15. J. H. Zhu et al., "Robust scaling iterative closest point algorithm with bidirectional distance measurement," *Electron. Lett.* **46**(24), 1604–1605 (2010).
16. S. H. Ying, Y. X. Peng, and Z. J. Wen, "Iwasawa decomposition: a new approach to 2D affine registration problem," *Pattern Anal. Appl.* **14**(2), 127–137 (2011).
17. S. H. Ying et al., "Lie group framework of iterative closest point algorithm for $n-D$ data registration," *Int. J. Pattern Recogn. Artif. Intell.* **23**(6), 1201–1220 (2009).
18. Y. Ma, J. Košecká, and S. Sastry, "Linear differential algorithm for motion recovery: a geometric approach," *Int. J. Comput. Vision* **36**(1), 71–89 (2000).
19. U. Grenander, M. Miller, and A. Srivastava, "Hilbert-Schmidt lower bounds for estimators on matrix lie groups for ATR," *IEEE Trans. Pattern Anal. Mach. Intell.* **20**(8), 790–802 (1998).
20. S. Krishnan et al., "Global registration of multiple 3D point sets via optimization-on-a-manifold," in *Proc. Third Eurographics Symposium on Geometry Processing*, Vienna, Austria (2005).
21. Y. Ma et al., "Euclidean reconstruction and reprojection up to subgroups," in *Proc. Seventh IEEE International Conf. on Computer Vision*, Vol. **2**, pp. 773–780, IEEE, Kerkyra, Greece (1999).

22. H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vision Image Understand.* **89**(2–3), 114–141 (2003).
23. A. Rangarajan, H. Chui, and J. S. Duncan, "Rigid point feature registration using mutual information," *Med. Image Anal.* **3**(4), 425–440 (1999).
24. A. Rangarajan et al., "A robust point-matching algorithm for autoradiograph alignment," *Med. Image Anal.* **1**(4), 379–398 (1997).
25. Y. Zheng and D. Doermann, "Robust point matching for nonrigid shapes by preserving local neighborhood structures," *IEEE Trans. Pattern Anal. Mach. Intell.* **28**(4), 643–649 (2006).
26. B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.* **33**(8), 1633–1645 (2011).
27. S. Granger and X. Pennec, "Multi-scale EM-ICP: a fast and robust approach for surface registration," in *Proc. of European Conf. on Computer Vision*, pp. 418–432, Springer, Denmark (2002).
28. S. Granger, X. Pennec, and A. Roche, "Rigid point-surface registration using an EM variant of ICP for computer guided oral implantology," in *Proc. of International Conf. on Medical Image Computing and Computer Assisted Intervention*, Vol. **2208**, pp. 752–761, Springer, Utrecht, Netherlands (2001).
29. "The Stanford 3D scanning repository," 6 September 2011, http://graphics.stanford.edu/data/3Dscanrep/ (28 January 2013).
30. F. Werner, *Foundations of Differentiable Manifolds and Lie Groups, Graduate Texts* in Mathematics 94, Springer Verlag, New York (1971).
31. S. Helgason, Differential Geometry, Lie Groups and Symmetric Space, Academic Press, New York (1978).
32. R. Murray, Z. X. Li, and S. Sastry, *A Mathematical Introduction to Robotic Manipulation*, CRC Press, Boca Raton, FL (1994).

Biographies and photographs for all authors are not available.