

# An Estimation of Distribution Algorithm Based on Variational Bayesian for Point-Set Registration

Hualong Cao<sup>✉</sup>, Qiqi He, Haifeng Wang<sup>✉</sup>, Zenghui Xiong, Ni Zhang, and Yang Yang<sup>✉</sup>, Member, IEEE

**Abstract**—Point-set registration is widely used in computer vision and pattern recognition. However, it has become a challenging problem since the current registration algorithms suffer from the complexities of the point-set distributions. To solve this problem, we propose a robust registration algorithm based on the estimation of distribution algorithm (EDA) to optimize the complex distributions from a global search mechanism. We propose an EDA probability model based on the asymmetric generalized Gaussian mixture model, which describes the area in the solution space as comprehensively as possible and constructs a probability model of complex distribution points, especially for missing and outliers. We propose a transformation and a Gaussian evolution strategy in the selection mechanism of EDA to process the deformation, rotation, and denoising of selected dominant individuals. Considering the complexity of the model, we choose to optimize from the perspective of variational Bayesian, and introduce a prior probability distribution through local variation to reinforce the convergence of the algorithm in dealing with complex point sets. In addition, a local search mechanism based on the simulated annealing algorithm is added to realize the coarse-to-fine registration. Experimental results show that our method has the best robustness compared with the state-of-the-art registration algorithms.

**Index Terms**—Asymmetric generalized Gaussian mixture model (AGGMM), estimation of distribution algorithm (EDA), evolution strategy (ES), point-set registration (PSR), variational Bayesian.

## I. INTRODUCTION

POINT-SET registration (PSR) is the process of finding the correspondence between the model point set and the target point set and then optimally aligning these point sets. PSR has

Manuscript received 11 June 2021; revised 24 September 2021 and 3 December 2021; accepted 27 December 2021. Date of publication 30 December 2021; date of current version 3 October 2022. This work was supported in part by the National Natural Science Foundation of China under Grant 41971392 and Grant 41661080, and in part by the Yunnan Province Ten-Thousand Talents Program. (Hualong Cao, Qiqi He, and Haifeng Wang contributed equally to this work.) (Corresponding author: Yang Yang.)

Hualong Cao, Qiqi He, Haifeng Wang, and Zenghui Xiong are with the School of Information Science and Technology and the Laboratory of Pattern Recognition and Artificial Intelligence, Yunnan Normal University, Kunming 650500, China (e-mail: caohl315@163.com; delline\_he@163.com; whf200608@163.com; ynnu\_xzh@125.com).

Ni Zhang is with the Yunnan Chinese Language and Culture College and the Laboratory of Pattern Recognition and Artificial Intelligence, Yunnan Normal University, Kunming 650500, China (e-mail: 2249585739@qq.com).

Yang Yang is with the School of Information Science and Technology, the School of Physics and Electronic Information, and the Laboratory of Pattern Recognition and Artificial Intelligence, Yunnan Normal University, Kunming 650500, China (e-mail: yyang\_ynu@163.com).

Color versions of one or more figures in this article are available at <https://doi.org/10.1109/TEVC.2021.3139304>.

Digital Object Identifier 10.1109/TEVC.2021.3139304

an indispensable role in machine vision [1], [2], image processing [3], and pattern recognition [4], [5]. Although current algorithms have been proposed to improve the robustness of registration, they always face some challenges, such as poor registration robustness due to a large number of outliers or missing points, long time consumption, inconsistent dimensionality of the input point set, and other problems. Therefore, designing a highly robust registration algorithm has always been a very important and difficult problem.

The iterative closest point (ICP) is a well-known method in PSR. Researchers, such as Granger and Pennec [6], Fitzgibbon [7], and Chetverikov *et al.* [8] have proposed some improved algorithms in ICP, whereas Chui and Rangarajan [2] extended this method to nonrigid PSR by proposing the robust PSR algorithm (TPSRPM). In addition, Lian *et al.* [9] proposed a clustering correspondence projection algorithm using quadratic planning and Tsin and Kanade [10] proposed a density-based method around the same time ICP was developed. Jian and Vemuri [11] proposed the GMMREG method, which uses a closed form to express the  $L_2$  distance between two Gaussian mixtures to align the mixture density and achieve registration. The most famous algorithm in PSR is the coherent point drift (CPD) [12] algorithm proposed by Myronenko and Song. CPD is based on GMM and uses the EM algorithm to maximize the likelihood function of the parameters. The MaL2E [13] method proposed by Ma *et al.* applied the transformation estimation algorithm using the L2E estimator to establish a robust sparse and dense correspondence of nonrigid registration. The VBPSM [14] method proposed by Qu *et al.* uses a combination of regression and clustering strategies to solve the point-set matching problem under the Bayesian framework. Zhang *et al.* [15] proposed a global local correspondence and transformation estimation (GL-CATE) PSR method. He *et al.* [16] proposed an adaptive hierarchical probabilistic model under a variational inference framework for addressing the PSR problem.

Although PSR has received much research attention, the current method still faces the following problems, which motivate this research.

- 1) *Deformation, Rotation, and Noise Issue:* Most of the current algorithms [1], [2], [12], [17] used the neighborhood structure of the point set to establish the correspondence among the point sets to address certain issues, such as deformation, rotation, and noise. However, when these methods restore the transformation from the correspondence, robustness is ignored. In addition, these problems were solved by converting the underlying space between

nonparametric point features, but this solution is limited by the fact that the true correspondence is nonrigid.

- 2) *Outlier and Missing Issue*: The current methods used a single attribute and a decision maker to estimate assumed inliers. These problems are being faced by the current GMM-based methods, such as [11]–[15], and [17], which deviate the cluster centers of the inlier and subsequently lead to unstable registration accuracy. Some GMM-based methods added an extra uniform distribution to the outliers. However, when the outlier distribution is irregular, their effect remains very poor.
- 3) *Objective Optimization Issue*: Most of the current PSR algorithms were EM optimization to solve this issue, such as: [2], [12], [14], and [16]. Although the EM method promotes the development of MLE, due to the logarithmic form of the likelihood function and the inclusion of hidden variables, this method cannot easily obtain the derivative. In addition, the sensitivity of the EM method to the initial values of parameters may affect its convergence efficiency and easily lead to a local optimal state.

Taking into account the above issues of PSR, the best registration scheme can be derived through mathematical derivation. However, the current PSR models have become too diverse that some algorithms do not have the mathematical features required by traditional optimization methods based on mathematical derivation. We deliberate that evolutionary algorithms (EAs) [18]–[23] can be used to solve many complex optimization problems, and have the mathematical features required by traditional optimization methods based on mathematical derivation. However, traditional EA [24]–[26] are often oriented to the individual level, and our PSR needs to look at the problem from a global perspective, which leads to poor registration robustness. If we want to solve these problems, we need to look at the problems at the population level. If we analyze the distribution law of the entire point set, can it improve the robustness of PSR?

Based on this idea, we thought of the estimation of distribution algorithm (EDA) [27] in EA, which uses an explicit probability model and an implicit form of sampling to generate new individuals. The use of explicit probabilistic models in optimization allows EDA to feasibly solve the difficult optimization problems caused by the uncertainty of the PSR correspondence. The advantage of EDA is that these models reveal a lot of information in PSR. This information can in turn be used to design problem-specific neighborhood operators for local search to further improve robustness of PSR. Therefore, this article proposes to use a simple and effective EDA to solve the PSR problem. EDA offers two advantages. First, EDA is a stochastic EA based on statistical learning theory, thereby making this algorithm coherent with the PSR model based on coordinates. Second, EDA has good global search capabilities and has a promising application prospect [28]–[31] for multiobjective optimization [32], [33] similar to PSR.

For different optimization problems, various probability models need to be designed to describe the distribution of the solution space. An appropriate probability model will describe the relationship between variables. Therefore,

EDA can generate better individuals based on the structural information of the optimization problems of nonlinearity and variable coupling. Compared with other EAs, EDA is a population-based macro-evolutionary method that uses global information in the solution space and historical information in the evolution process, thereby increasing its global search capabilities and convergence speed. In addition, EDA is simple and easy to implement. EDA is particularly convenient in estimating the distribution of the solution space and sampling new individuals and may even be mixed with other algorithms to enhance its optimization performance. Obviously, the definition of the probability model is crucial. By using probabilistic models, EDAs can introduce *a priori* information for optimization to use variational Bayes (VB). The use of prior information for optimization has been examined in previous research [34]. Practitioners can incorporate two sources of deviation into EDA, namely: 1) prior knowledge and 2) information obtained from EDA that was previously run on similar issues. These two methods can also be combined with VB or other methods. Our algorithm treats the domain knowledge extracted from user preferences as prior knowledge and uses such knowledge it to enhance EDA.

In sum, we apply EDA to solve the PSR problem. The main contributions of our work are summarized as follows.

- 1) In order to solve *issue 1*, we propose a transformation based on Gaussian to select inlier on the target point on the selection operator and then perform Gaussian mixtures to deal with rigid and nonrigid transformations; we also propose an evolution strategy (ES) [35], [36] for denoising [37]–[39] and propose simulated annealing [40], [41] for the *a priori* treatment of variance to achieve a coarse-to-fine registration.
- 2) In order to solve *issue 2*, we build the asymmetric generalized Gaussian mixture model (AGGMM) to build a probabilistic model for evolution. This model adapts well to the probability distribution in various complex situations during the registration process, thereby improving the robustness of the registration. This model controls the mixing ratio of the corresponding points to deal with the missing points and visualizes outliers for their aggregation during registration and for reducing their impact.
- 3) In order to solve *issue 3*, we propose EDA based on VB, update each individual through VB, and check whether the termination condition is satisfied. Given that the EM algorithm can easily fall into the local optimum when processing large-scale data, we use the VB framework instead of the EM algorithm for the posterior approximation. Meanwhile, we propose a local variational strategy and use the Newton method to achieve a closed form update of the tail parameters to deal with those tricky items in the variational objective function.

The remainder of this article is organized as follows. In Section II, we briefly introduce the preliminary work, including the graphical description of the PSR and the technical route of our method. Section III is devoted to the VB proposed based on AGGMM within the EDA framework, including the determination of the selection operator, the construction of the

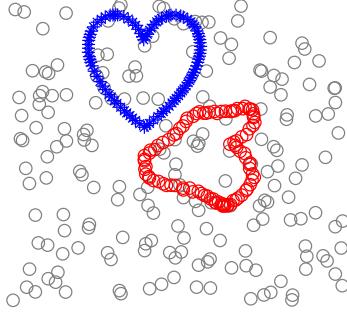


Fig. 1. Illustration of PSR. The blue “\*” represents the *model* point set, the red “○” point represents the *target* point set, and outliers are gray “○.”

model, the prior and update of parameters, sampling, and termination conditions. See Section IV for experimental results and analysis results. Section V concludes and discusses this article.

## II. PROBLEM EXPLANATION AND METHOD PROCESS

The goal of PSR is to find the point-to-point correspondence between two given sets of points. PSR is graphically illustrated in Fig. 1. “○” represents the *target* point set, and “+” represents the *model* point set. PSR aims to establish a one-to-one correspondence between the blue “\*” and red “○” amid the presence of outliers (gray “○”). Therefore, the PSR problem can be described as a given *model* point set  $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^K$  with  $K$  points and a *target* point set  $\mathcal{T} = \{\mathbf{t}_n\}_{n=1}^N$  with  $N$  points, both of which are  $D$ -dimensional.

To efficiently solve problems in PSR, we propose EDA, which initially performs a preregistration process and then transforms the registration problem into a probability optimization problem. Afterward, through the selection, establishment of the probability model, sampling, termination, variation, and other operations are performed to find the correspondence between the *model* point set and the *target* point set. The probability formula of PSR can be expressed as the marginal integral of mapping function  $\mathcal{F}$  relative to *model* and *target*

$$p(\mathcal{T}|\mathcal{M}) = \int p(\mathcal{T}, \mathcal{F}|\mathcal{M})p(\mathcal{F})d\mathcal{F}. \quad (1)$$

The mapping function  $\mathcal{F}$  plays a key role in transforming the *model* point into the *target* space. However, given certain issues such as population optimization, the number of potential variables to be processed may be too large to be directly calculated. Therefore, the deterministic approximation scheme VB is based on analytical approximations to the posterior distribution, hence making this method very suitable for addressing multiobjective optimization problems. The framework of VB deterministic approximation can be infinitely approximated toward the desired optimal solution with each evolutionary update.

The distribution information of the candidate solutions in the search space is described by establishing a probability model. Statistical learning methods are then applied to establish a probability model that describes the distribution of solutions from the macro perspective of the group. Afterward,

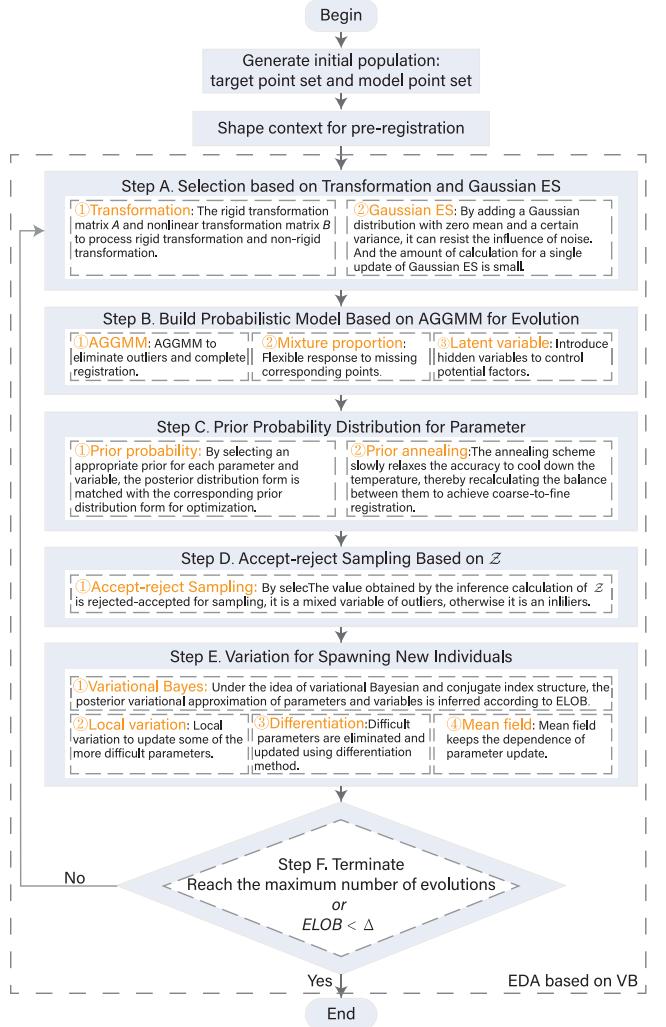


Fig. 2. Main contribution points of this article and a schematic diagram of our method. Our contributions are marked in orange.

the probability model is randomly sampled to generate a new population. The evolution of the population is realized repeatedly in this way. On the basis of this idea, we select the inlier from the point set in a way similar to Gaussian regression in the selection operator. We then apply the rejection–accept sampling method in the sampling phase. When the KL divergence is as small as the expected value, or after a certain algebraic evolution, the evolution is terminated, and the overall solution reaches optimality, thereby completing the PSR.

## III. METHOD

Given that our method is based on VB’s EDA, the standard EDA process is strictly followed. The process of our method is illustrated in Fig. 2, in which our contributions are marked in orange.

In general, if two point sets have similar shapes, then their corresponding points have similar neighborhood structures. Therefore, finding the correspondence between two point sets is equivalent to finding the point with the most similar feature descriptors in another point set. Inspired by this idea, we use

the shape context [42] to preprocess a corresponding relation of the point set.

### A. Selection Based on Transformation and Gaussian ES

We propose the transformation and Gaussian ES to generate a mixed model with  $K$  components related to the *model* point set in the space of the *target* point set. In other words, we select the dominant population ( $K$  inliers) through transformation and Gaussian ES.

1) *Transformation*: Given the rigid and nonrigid transformations in the point set, we propose linear and nonlinear matrices to explain the transformations in the PSR.

$\mathcal{F}$  in (1) is critical in establishing the correspondence between the *model* and the *target* point sets. Given that PSR has rigid and nonrigid registrations,  $\mathcal{F}$  in our method is divided into two parts, namely, the linear relationship that controls the rigid transformation and the nonlinear relationship that controls the nonrigid transformation. Therefore,  $\mathcal{F}$  can be rewritten as

$$\mathcal{F}_{m_k} = \mathbf{A}\mathcal{M} + \mathbf{B}\Phi(\mathcal{M}) \quad (2)$$

where  $\mathbf{A}$  is the linear matrix of  $D \times (D + 1)$  that controls the rigid transformation, and the column of  $D + 1$  is the transformation variable.  $\mathbf{B}$  and  $\Phi(\mathcal{M})$  are the weight matrices of the radial basis function (RBF) and the basis matrix with radial basis, which are used to represent nonlinear transformations. For simplicity and convenience, we place the bias vector into  $\mathbf{A}$ . Afterward, we can find the optimal rigid and nonrigid matching conversion parameters  $\mathbf{A}$  and  $\mathbf{B}$ .

In this case,  $\mathcal{F}_{m_k}$  can easily cope with the rigid and nonrigid transformations in PSR. We then define the transition variable  $\mathcal{X} = \{\mathbf{x}_k\}_{k=1}^K$  in the target space by transforming the model. The  $k$ th variable  $\mathbf{x}_k$  can be expressed as

$$\mathbf{x}_k = \mathbf{A}\tilde{m}_k + \mathbf{B}\Phi(\tilde{m}_k) \quad (3)$$

where  $\tilde{m}_k = [1; m_k]$ .

2) *Gaussian ES for Denoising*: Given the influence of noise in PSR, we assume this noise to be Gaussian noise. Therefore, we propose Gaussian ES for denoising.

Given that an image generally has different degrees of noise, certain amount of noise is observed when generating feature points. In this case, denoising is needed. Many studies show that the distribution of noise is mostly Gaussian. Therefore, this article assumes that each noise is Gaussian noise (our method can be directly applied to other noise models). In addition, given that ES is zero-order optimization, a Gaussian distribution with zero mean and a certain variance is added to help the ES effectively resist the influence of noise when the calculation amount of a single update is small. Following these ideas, we propose ES for denoising. We update (3) as follows, where the  $k$ th variable  $\mathbf{x}_k$  is:

$$\mathbf{x}_k = \mathbf{A}\tilde{m}_k + \mathbf{B}\Phi(\tilde{m}_k) + \delta_k. \quad (4)$$

$\delta_k$  denotes the Gaussian ES or the Gaussian noise with a zero-mean of  $D$ -dimension and anisotropic precision. We assume that each component of Gaussian ES is a Gaussian distribution with zero mean and anisotropic precision matrix, that  $\nu$

indicates the relationship among the mixed components of the noise point, and that  $\nu_{kj}$  is a 1-of- $J$  binary vector. On the basis of these points, we propose to mixing the  $D$ -dimensional Gaussian and  $J$ -component for noise modeling to solve heteroscedastic noise and further increase the robustness of our model [43].

We then assume that the mixed components of the noise follow a Gaussian distribution with mean  $\mu_j$  and an anisotropy accuracy  $\Lambda_j$ . The conditional distribution of  $\mathbf{x}_k$  can then be written as

$$p(\mathbf{x}_k | \Upsilon) = \prod_{k=1}^K \prod_{j=1}^J \mathcal{N} \left( \mathbf{x}_k | \mathbf{A}\tilde{m}_k + \mathbf{B}\Phi(\tilde{m}_k) + \mu_j, \Lambda_j^{-1} \right)^{\nu_{kj}} \quad (5)$$

where  $\Upsilon = \{\mathbf{A}, \mathbf{B}, \mu, \Lambda, \nu\}$ ,  $\mu = \{\mu_j\}_{j=1}^J$ ,  $\Lambda = \{\Lambda_j\}_{j=1}^J$ ,  $\nu = \{\nu_k, k = 1, \dots, K\}$  is an indicator variable used to identify the source of noise, and  $\nu_k$  is a 1-of- $J$  binary vector with elements  $\{\nu_{kj}, j = 1, \dots, J\}$ .

We then obtain  $K$  Gaussian mixtures from the target point, which means that each target point will produce a Gaussian mixture and that inliers are selected.

### B. Build Probabilistic Model Based on AGGMM for Evolution

The probabilistic model is the core of EDA, and EDA describes the spatial distribution of solutions and the overall evolutionary trend of the population through probabilistic models and their updates. Therefore, we propose a set of universal probability models as follows.

1) *AGGMM*: We use the AGGMM to perform PSR and eliminate outliers and missing correspondences.

The  $K$  Gaussian mixture is generated by the target point set after transformation, and each target point concentration point can obtain the corresponding probability distribution according to the posterior distribution. However, as shown in Fig. 1, PSR usually has many outliers. However, previous studies have used the Gaussian distribution to build models, complete the registration, and deal with outliers. However, given the symmetry of the Gaussian distribution, many components are needed to explain the asymmetry and long tail of skewed data, which would lead to complex calculations and other challenges. Therefore, many researchers model the distribution of flexibility and robustness similar to GMM, such as Lamma distribution, Beta distribution, Student- $t$  distribution [16], [44], etc. Following this idea, we use the AGGMM as the component density. We divide this model into two different mixtures and complete the registration and outlier elimination at the same time. We use the transitional blend and  $K$  component to find the corresponding relationship, and the outlier blend and  $K_0$  component to deal with the outliers. We then obtain the following formula:

$$p(t_n; \theta_1, \theta_2) = \sum_{k=1}^K \pi_k \mathcal{A}(t_n; \theta_1) \sum_{k=K+1}^{K+K_0} \pi_k \mathcal{A}(t_n; \theta_2) \quad (6)$$

where  $\pi_k$  is the mixing coefficient and  $h(s_n; \theta)$  is the AGG. The probability density function (PDF) of an AGGM can be

represented by

$$\mathcal{A}(t_n; \theta) = \begin{cases} \tau_a \cdot \exp\left[-\frac{\tau_b(-t_n + \eta)^w}{(\sigma_l)^w}\right], & t_n \leq \eta \\ \tau_a \cdot \exp\left[-\frac{\tau_b(t_n - \eta)^w}{(\sigma_r)^w}\right], & t_n > \eta \end{cases} \quad (7)$$

with

$$\tau_a = \frac{\Gamma\left(\frac{3}{w}\right)^{1/2}}{(\sigma_l + \sigma_r)\Gamma\left(\frac{1}{w}\right)^{3/2}}, \quad \tau_b = \frac{\Gamma\left(\frac{3}{w}\right)^{w/2}}{\Gamma\left(\frac{1}{w}\right)^{w/2}}$$

and  $\Gamma(\star)$  is the gamma function defined as  $\Gamma(x) = \int_0^{+\infty} e^{-t} t^{x-1} dt$ .  $\theta$  is a set of parameters  $\theta = \{\eta, \sigma_l, \sigma_r, w\}$ , where  $\sigma_l > 0$ ,  $\sigma_r > 0$ , and  $w > 0$ , and  $\eta$  is a virtual expectation of AGG, that is, the value of the abscissa corresponding to the intersection of the left and right halves are used to divide the equivalent halves.  $\sigma_l$  and  $\sigma_r$  denote the variances of the left and right halves, respectively. The tail parameter  $w$  controls the peak thick tail of the distribution, and its value determines the characteristics of the AGG distribution.

$\pi_k$  in (6) is used to deal with missing correspondence. PSR has many missing situations that can be regarded as the disappearing mixture ratio. Therefore, the mixture ratio of the missing points is equivalent to 0. Following this idea, we treat the parameterized mixing ratio as a random variable jointly controlled by Dirichlet in order to release the change of the mixing ratio. Therefore, for models that are likely to miss the corresponding relationship, the impact on reweighting is small.

2) *Latent Variable*: The negative impact of latent variables on the effect of registration is offset.

For each observation  $t_n$ , we have a corresponding latent variable  $z_n$  comprising a binary vector with elements  $z_{nk}$  for  $k = 1, \dots, K + K_0$ , and we define as  $\mathcal{Z} = \{z_n\}_{n=1}^{K+K_0}$ , where element  $z_n$  is 1 and all other elements are 0. In other words, all entries are equal to 0 except for that at the position indicating correspondence. Therefore, the distribution of  $\mathcal{Z}$  conditioned on  $\pi$  has the following polynomial density:

$$p(z_n|\pi) = \prod_{k=1}^{K+K_0} \pi_k^{z_{nk}}. \quad (8)$$

We can obtain the conditional distribution of the observed data vector based on the given latent variables and the previously defined component parameters as follows:

$$p(t_n|\mathcal{E}, \mathcal{H}) = \prod_{n=1}^N \prod_{k=1}^K \mathcal{A}(t_n|x_k, \sigma_{ulk}, \sigma_{urk}, u_k)^{z_{nk}} \times \prod_{k=K+1}^{K+K_0} \mathcal{A}(t_n|\mathcal{G}_k, \sigma_{vdk}, \sigma_{vrk}, v_k)^{z_{nk}} \quad (9)$$

where  $\mathcal{G}_k$  denotes the mean of one component in an outlier. The set of all random variables is  $\mathcal{E} = \{\mathcal{Z}, \mathcal{X}, \mathcal{G}, \pi, \sigma_{ul}, \sigma_{vl}, \sigma_{ur}, \sigma_{vr}\}$ , and  $\mathcal{H} = \{u, v\}$  denotes the parameter set.

### C. Prior Probability Distribution for Parameter

1) *Prior Probability Distribution*: We transform the PSR problem into an optimization problem based on a probabilistic model. Combining (1) and (9), we can easily see that in this complex Bayesian model, each parameter follows a prior probability distribution, and latent variables and parameters are present. Therefore, one can easily assume a variational inference, which is a Bayesian learning method optimized by matching the posterior distribution form with the corresponding prior distribution form. Therefore, we initially select the appropriate priors for each parameter and variable. These priors are usually selected from the distribution of the exponential family. In our probability model, the priors of each variable are also selected from the distribution of the exponential family.

First, for  $A$  and  $B$  in (5), their priors are all Gaussian distributions of a certain variance with zero-mean, and the variances are all extracted from the Gamma distribution to represent the automatic relevance determination (ARD). We assume that these variances are  $\alpha$  and  $\beta$ , respectively, which follow the Gamma distribution with prior parameters  $\{a_0, b_0\}$  and  $\{c_0, d_0\}$ . Similarly, we introduce an independent Gaussian–Wishart prior distribution to control mean  $\mu_j$  and accuracy  $\Lambda_j$  of each Gaussian distribution,  $\mu_0, \varpi_0, \epsilon_0$ , and  $\varrho_0$  are the prior parameters of the Gaussian–Wishart distribution for ES mixtures, given that the mean and accuracy are both unknown. The distribution of  $v$  is conditional on a mixing proportion  $\gamma$ , which follows a Dirichlet distribution with prior parameter  $j_0$ .

Second, for the latent variable  $\mathcal{Z}$  in (9), (8) shows that  $\mathcal{Z}$  is a polynomial distribution with  $\pi$  as the prior parameter.  $\pi$  is the same as  $\gamma$  and follows the Dirichlet distribution  $\text{Dir}(\pi; \kappa_0)$ . The prior of  $\mathcal{G}$  follows an asymmetric Gaussian distribution controlled by parameters  $\xi_0, \zeta_0$ , and  $\varsigma_0$ . However,  $\sigma_{ul}, \sigma_{vl}, \sigma_{ur}, \sigma_{vr}, u$ , and  $v$  are all complicated, their priors cannot be calculated. Therefore, we simplify  $\sigma_{ul}, \sigma_{vl}, \sigma_{ur}$ , and  $\sigma_{vr}$  based on local variation and use the differentiation method to parameterize  $u$  and  $v$ . Equation (7) shows that the prior probabilities of  $\sigma_l$  and  $\sigma_r$  are the same. Therefore, we only show that one of them can be simplified by using local variation and prove the other naturally. Let us take  $\sigma_l$  as an example. From (7), we find that  $\sigma_l$  exists in two terms: one is the denominator part of the previous term at the bottom of the index and the other is on the index. After we perform logarithmic transformation, we find that the former is relatively simple, whereas the latter is very complicated. We also cannot easily find their corresponding prior probabilities. However, this process can be simplified by local variation, and the exponential distribution can be selected as the prior for the simplified complex term. Therefore, we only make partial changes to this complex term and keep the previous simple term to ensure approximate results. Given space limitations, refer to [45] for the specific derivation process. Only the results of the local variation will be presented in this article.

The complex term is

$$g(\sigma_l) = -\frac{\tau_b(-t_n + \eta)^w}{(\sigma_l)^w}. \quad (10)$$

The first-order Taylor expansion for (10) is  $h(\sigma_l, \sigma_l^*) = g(\sigma_l^*) + g'(\sigma_l^*)(\sigma_l - \sigma_l^*)$ . Then, let  $\varphi = g'(\sigma_l^*)$ , and the

following item about  $\sigma_l^*$  is defined as  $j(\varphi)$ . Therefore, we convert (10) into the following simple linear relationship  $h(\sigma_l, \varphi) = \varphi\sigma_l + j(\varphi)$ . Afterward, we obtain the best  $\varphi$  as follows according to the local variation:

$$\varphi = \frac{\tau_b(-t_n + \eta)^w}{\sigma_l^{w+1}} \cdot \left( \frac{w+1}{w} \right)^{-(w+1)} \cdot [w^{1/(w+1)} + w^{-[w/(w+1)]}]^{w+1}. \quad (11)$$

By transforming complex terms into simple linear relationships, we can understand that the prior of  $\lambda_0$  follows an exponential distribution and that the same is true for  $\sigma_{ul}$ ,  $\sigma_{vl}$ ,  $\sigma_{ur}$ , and  $\sigma_{vr}$ . For  $u$  and  $v$ , using the local variation for processing will only introduce complexity. Therefore, we treat it as a parameter.

We then determine *a priori* of each variable, and the results are as follows:

$$\begin{aligned} p(\mu, \Lambda) &= p(\mu|\Lambda)p(\Lambda) = p(\mu, \Lambda|\mu_0, \varpi_0, \epsilon_0, \varrho_0) \\ &= \prod_{j=1}^J \mathcal{N}(\mu|\mu_0, (\varpi_0 \Lambda)^{-1}) \mathcal{W}(\Lambda|\varrho_0, \epsilon_0) \\ \mathcal{Z}_n &\sim \text{Mult}(z_n|\pi) = \prod_{k=1}^{K+K_0} \pi_k^{z_{nk}} \\ \pi &\sim \text{Dir}(\pi; \kappa_0) = c(\kappa_0) \prod_{k=1}^{K+K_0} \pi_k^{\kappa_0-1} \\ \mathcal{G} &\sim (\mathcal{G}_k; \xi_0, \zeta_0, \varsigma_0) = \frac{\varsigma_0}{2\xi_0 \Gamma\left(\frac{1}{\xi_0}\right)} \exp\left(-\left|\frac{\mathcal{X}_k - \xi_0}{\zeta_0}\right|^{\xi_0}\right) \\ \sigma &\sim E(\lambda_0) = \lambda_0 \cdot \exp\{-\lambda_0 \sigma\} \\ \alpha &\sim \text{Gam}(\alpha|a_0, b_0), \beta \sim \text{Gam}(\beta|c_0, d_0) \\ v &\sim \text{Mult}(v|\gamma), \gamma \sim \text{Dir}(\gamma; j_0). \end{aligned} \quad (12)$$

We then select the optimal control methods for each variable, including conjugate priors, local variation, and parametrization. The directed graph in Fig. 3 is built according to our designed probability model.

**2) Prior Annealing:** The initialization of the prior parameters has a great influence on the posterior approximation of VB inference. The prior scale matrices  $\lambda_{ul}$  and  $\lambda_{ur}$  are either too large or too small, which will influence the effect of the registration. Larger matrices will also reduce the chances for the algorithm to be trapped in the local minimum, thereby leading to inaccurate alignment and greater uncertainty. In contrast, although smaller  $\lambda_{ul}$  and  $\lambda_{ur}$  can achieve fine matching, the local minimum can be easily captured.

Therefore, we use an annealing scheme to slowly relax  $\lambda_{ul}$  and  $\lambda_{ur}$  and recalculate the balance between them. We observe a series of stages (i.e.,  $\{20, 10, 5, 2, 1\}$ ) to lower the temperature and achieve coarse-to-fine registration.

#### D. Accept–Reject Sampling Based on $\mathcal{Z}$

We obtain  $K$  Gaussian mixture components in the target point set, and each target point corresponds to a certain component based on posterior probability. In addition, the proposed

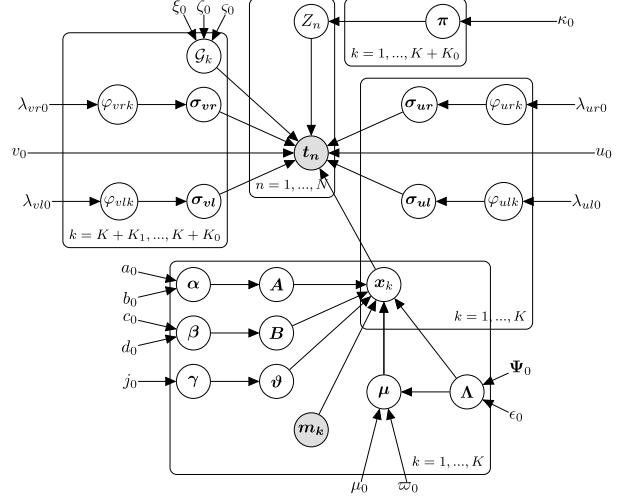


Fig. 3. Directed acyclic graph of the probability model for PSR. The white circle nodes are the latent random variables, the gray-shaded circles represent the observed variables, the noncircled nodes are constants, and the direction of the arrow represents the dependency.

mixture model with  $K_0$  components estimates the outliers to prevent them from affecting the matching accuracy [12], [46]. For each target point  $t_n$ , a corresponding indicator variable  $\mathcal{Z} = \{z_n\}_{n=1}^{K+K_0}$  can be found in the *model*. Among them,  $z_n$  represents a 1-of- $(K+K_0)$  binary vector. We compare the value obtained by the inference calculation of  $\mathcal{Z}$  with the expected threshold  $t_z$ . When  $\mathcal{Z} > t_z$ , it is a mixed variable of outlier; otherwise, it is inlier. This is accept–reject sampling of inlier.

#### E. Terminate

Our proposed probability model can determine its joint probability distribution after evolution and selection, but our goal is to find the corresponding posterior probability distribution  $p(\mathcal{T}|\mathcal{M})$ . Equation (1) shows that  $p(\mathcal{T}|\mathcal{M})$  can be expressed as the marginal probability of each random parameter and variable. Let  $\psi$  denote a collection of random parameters and variables such that  $\psi = \{\mathcal{X}, \mathcal{Z}, \mathcal{G}, \mathbf{A}, \mathbf{B}, \mathbf{v}, \mu, \Lambda, \alpha, \beta, \gamma, \pi, \sigma_{ul}, \sigma_{vl}, \sigma_{ur}, \sigma_{vr}\}$ . Therefore, according to Fig. 3, (1) can be updated to a series of conditional probabilities as follows:

$$\begin{aligned} p(\mathcal{T}|\mathcal{M}) &= \int p(\mathcal{T}|\mathcal{Z}, \mathcal{X}, \mathcal{G}, \sigma_{ul}, \sigma_{ur}, \sigma_{vl}, \sigma_{vr}, u_0, v_0) \\ &\quad \times p(\mathcal{X}|\mathbf{A}, \mathbf{B}, \boldsymbol{\vartheta}, \mu, \Lambda, \mathcal{M}) p(\mathcal{Z}|\pi) p(\pi|\kappa_0) p(\mathbf{A}|\alpha) \\ &\quad \times p(\alpha|a_0, b_0) p(\mathbf{B}|\beta) p(\beta|c_0, d_0) p(\boldsymbol{\vartheta}|\gamma) p(\gamma|j_0) \\ &\quad \times p(\mu, \Lambda|\mu_0, \varpi_0, \epsilon_0, \varrho_0) p(\mathcal{G}|\xi_0, \zeta_0, \varsigma_0) p(\sigma_{ul}|\varphi_{ul}) \\ &\quad \times p(\varphi_{ul}|\lambda_{ul}0) p(\sigma_{ur}|\varphi_{ur}) p(\varphi_{ur}|\lambda_{ur}0) p(\sigma_{vl}|\varphi_{vl}) \\ &\quad \times p(\varphi_{vl}|\lambda_{vl}0) p(\sigma_{vr}|\varphi_{vr}) p(\varphi_{vr}|\lambda_{vr}0) d\psi. \end{aligned} \quad (13)$$

In the framework of VB,  $\log p(\mathcal{T}|\mathcal{M})$  is the objective optimization function that can be decomposed into evidence lower bound (ELOB)  $\mathcal{L}(q(\Phi))$  and KL divergence  $KL(q(\Phi)||p(\Phi|\mathcal{T}, \mathcal{M}))$  [45]. After update iteration, ELOB gradually approximates the logarithmic marginal probability. Therefore, we set a minimum value, and if ELOB reaches this value, then the algorithm will be terminated. Meanwhile,

if ELOB still cannot reach this value after a maximum number of iterations  $o$ , then the algorithm will also be terminated.

#### F. Variation for Spawning New Individuals

An important idea in variational approximation is that under the conjugate index structure [47], the prior and posterior usually keep the same distribution form. Therefore, we can easily deduce the posterior of the parameters and variables based on variational approximation. However, given space constraints, this article does not explain in detail the process of deriving the posterior distribution. Readers may refer to [48] for a detailed discussion of this process.

*1) Rigid Transformation Matrix  $\mathbf{A}$ :* According to [49], the posterior over the transformation is factorized over the rows of  $\mathbf{A}$ , and the precision parameter  $\boldsymbol{\alpha}$  is placed over the column of  $\mathbf{A}$ . The update results for the approximate mean  $\hat{\mathbf{v}}_i$  and covariance  $\hat{\Sigma}_i$  over the  $i$ th row of the control rigid transformation matrix  $A$  are

$$\begin{aligned}\hat{\Sigma}_i^{-1} &= \text{diag}(\boldsymbol{\alpha}) + \sum_{j=1}^J \langle \Lambda_{ij} \rangle \sum_{k=1}^K (\tilde{\mathcal{M}} F_j \tilde{\mathcal{M}}^T) \\ \hat{\mathbf{v}}_i &= \hat{\Sigma}_i \left( \sum_{j=1}^J \langle \Lambda_{ij} \rangle \sum_{k=1}^K [\mathcal{X} - \mathbf{B} \Phi(\mathcal{M}) - \boldsymbol{\mu}_j] F_j \tilde{\mathbf{m}}_k^T \right)\end{aligned}\quad (14)$$

where  $F_j = \text{diag}([\vartheta_{1j}, \vartheta_{2j}, \dots, \vartheta_{kj}])$ , and a  $D(D+1)(D+1)$  covariance diagonal matrix is observed on the posterior distribution of  $\mathbf{A}$ . With regard to the variational posterior of its precision  $\boldsymbol{\alpha}$  parameters  $\{a_0, b_0\}$ , given that Section III-C has mentioned that its prior is the Gamma distribution, the variational posterior of  $\boldsymbol{\alpha}$  is still the Gamma distribution. The updated posterior parameters are

$$\begin{aligned}a_i &= a_0 + \frac{D}{2} \\ b_i &= b_0 + \frac{1}{2} \sum_{q=1}^D \langle A_{qi}^2 \rangle\end{aligned}\quad (15)$$

where  $a_0$  and  $b_0$  are the prior shape and inverse-of-scale parameters of the Gamma distribution, respectively.

*2) Weight Matrix  $\mathbf{B}$ :* Similarly, the mean  $\hat{\mathbf{x}}$  and covariance  $\hat{\Xi}$  over the  $i$ th row of  $B$  are expressed as

$$\begin{aligned}\hat{\Xi}_i^{-1} &= \text{diag}(\boldsymbol{\beta}) + \sum_{j=1}^J \langle \Lambda_{ij} \rangle \sum_{k=1}^K (\Phi(\mathcal{M}) F_j \Phi(\mathcal{M})^T) \\ \hat{\Xi}_i &= \hat{\Xi}_i \left( \sum_{j=1}^J \langle \Lambda_{ij} \rangle \sum_{k=1}^K [\mathcal{X} - \mathbf{A} \tilde{\mathcal{M}} - \boldsymbol{\mu}_j] F_j \Phi(\tilde{\mathbf{m}}_k)^T \right)\end{aligned}\quad (16)$$

where  $\hat{\Xi}$  is a diagonal matrix. The variational posterior of accuracy  $\boldsymbol{\beta}$  follows a Gamma distribution. The posterior parameter of  $\boldsymbol{\beta}$  is computed as

$$\begin{aligned}c_i &= c_0 + \frac{D}{2} \\ d_i &= d_0 + \frac{1}{2} \sum_{q=1}^D \langle B_{qi}^2 \rangle\end{aligned}\quad (17)$$

where  $c_0$  and  $d_0$  denote the prior shape and inverse-of-scale parameters of the Gamma distribution, respectively.

*3) Mean  $\boldsymbol{\mu}$  and Precision  $\boldsymbol{\Lambda}$  of Noise Mixtures:* According to (12), the variational posterior of  $\{\boldsymbol{\mu}, \boldsymbol{\Lambda}\}$  is a Gaussian-Wishart distribution, so the update of its posterior parameters is updated as

$$\begin{aligned}\epsilon_i &= \epsilon_0 + \sum_{k=1}^K \langle v_{ki} \rangle \\ \varpi_i &= \varpi_0 + \sum_{k=1}^K \langle v_{ki} \rangle \\ \boldsymbol{\mu}_i &= \frac{1}{\varpi_i} \left[ \varpi_0 \boldsymbol{\mu}_0 + \sum_{k=1}^K \langle v_{ki} \mathcal{I} \rangle \right] \\ \varrho_i^{-1} &= \varrho_0^{-1} + \frac{1}{\varpi_i} [\varpi_0 \boldsymbol{\mu}_0 \boldsymbol{\mu}_0^T + \langle \mathcal{I} F_j \mathcal{I} \rangle]\end{aligned}\quad (18)$$

where  $\mathcal{I} = \mathcal{X} - \mathbf{A} \tilde{\mathcal{M}} - \mathbf{B} \Phi(\mathcal{M})$ , which represents the error between the transformed variable  $\mathcal{X}$  and the model.

*4) Proportions  $\boldsymbol{\vartheta}$  of Noise Mixtures:* Let  $\langle \vartheta_{kj} \rangle$  denote the expectation of the  $j$ th mixture component

$$\begin{aligned}\langle \vartheta_{kj} \rangle &= \frac{1}{\mathfrak{C}_{kj}} \left\{ \langle \log \gamma_j \rangle + \frac{1}{2} \langle \log |\Lambda_j| \rangle \right. \\ &\quad \left. - \frac{1}{2} \langle (\boldsymbol{\varkappa}_k - \boldsymbol{\mu}_j)^T \boldsymbol{\Lambda}_j (\boldsymbol{\varkappa}_k - \boldsymbol{\mu}_j) \rangle \right\}\end{aligned}\quad (19)$$

where  $\mathfrak{C}_{kj}$  is a normalization factor, and  $\boldsymbol{\varkappa}_k = \mathcal{T}_{(m_k)} - \mathbf{A} \tilde{\mathbf{m}}_k - \mathbf{B} \Phi(\tilde{\mathbf{m}}_k)$  represents the error of the  $k$ th regression between the model and latent variable. The  $k$ th update of the parameter posterior  $j_k$  for  $\boldsymbol{\gamma}$  is

$$j_k = j_0 + \sum_{k=1}^K \langle \vartheta_{kj} \rangle.\quad (20)$$

*5) Latent Variable  $\mathcal{Z}$ :* The joint posterior of the latent variable can be obtained from the expectation of its complete conditional. Through marginalization and normalization, we can obtain the expectation of  $z_{nk}$  as

$$\langle z_{nk} \rangle = \frac{q(z_{nk} = 1)}{\sum_{k=1}^K q(z_{nk} = 1)}\quad (21)$$

where  $\langle z_{nk} \rangle$  indicates that  $z_{nk}$  expects  $\mathcal{Z}$ . The distribution of  $\mathcal{Z}$  conditioned on  $\pi$  has a polynomial density, and  $\pi$  follows  $\text{Dir}(\pi; \kappa_0)$ . Parameter  $\kappa$  is then updated as

$$\kappa_k = \kappa_0 + \sum_{k=1}^K \langle z_{nk} \rangle.\quad (22)$$

*6) Transition Variable  $\mathbf{x}_k$ :* For  $\mathbf{x}_k$ , the covariance  $\boldsymbol{\sigma}_{x_k}$  and mean  $\boldsymbol{\mu}_{x_k}$  are updated as

$$\begin{aligned}\hat{\boldsymbol{\sigma}}_{x_k}^{-1} &= \sum_{n=1}^N \langle z_{nk} \sigma_{ulk}^{u_k} \rangle + \sum_{j=1}^J \langle \vartheta_{kj} \boldsymbol{\Lambda}_j \rangle \\ \hat{\boldsymbol{\mu}}_{x_k} &= \hat{\boldsymbol{\sigma}}_{x_k} \left[ \sum_{n=1}^N \langle z_{nk} \sigma_{ulk}^{u_k} \rangle \mathbf{f}_n + \sum_{j=1}^J \langle \vartheta_{kj} \boldsymbol{\Lambda}_j \boldsymbol{\phi}_k \rangle \right]\end{aligned}\quad (23)$$

where  $\phi_k = A\tilde{m}_k + B\Phi(\tilde{m}_k) + \mu_j$ . Equation (23) is used when  $t_n \leq x_k$ , but when  $t_n > x_k$ ,  $\sigma_{ulk}$ , and  $u_k$  only need to be replaced with  $\sigma_{urk}$ , and  $v_k$ .

7) *Model Posterior Parameters After Transformation:* In Section III-C, we demonstrate that the priors of  $\sigma_{ul}$  and  $\sigma_{ur}$  are exponential distributions. Therefore, the posterior of  $\sigma_{ul}$  and  $\sigma_{ur}$  is also an exponential distribution, and the parameters  $\lambda_{ul}$  and  $\lambda_{ur}$  are updated as

$$\begin{aligned}\lambda_{ulk} &= \begin{cases} \lambda_{ul0} - \sum_{k=1}^K \langle \mathcal{Z}_{nk} \rangle E_{\mathcal{X}}[\varphi_{ulk}], & S_n \leq x_k \\ \lambda_{ul0}, & S_n > x_k \end{cases} \\ \lambda_{urk} &= \begin{cases} \lambda_{ur0}, & S_n \leq x_k \\ \lambda_{ur0} - \sum_{k=1}^K \langle \mathcal{Z}_{nk} \rangle E_{\mathcal{X}}[\varphi_{urk}], & S_n > x_k. \end{cases} \quad (24)\end{aligned}$$

8) *Mean  $\mathcal{G}$  and Precision  $\sigma_{vl}$  and  $\sigma_{vr}$  of Outlier Mixtures:* Like the precision  $\sigma_{ul}$  and  $\sigma_{ur}$  of transformed, the precision posterior of the outlier is also an exponential distribution. Each parameter is updated as

$$\begin{aligned}\lambda_{vdk} &= \begin{cases} \lambda_{vl0} - \sum_{k=K+1}^{K+K_0} \langle \mathcal{Z}_{nk} \rangle E_{\mathcal{G}}[\varphi_{vdk}], & S_n \leq \mathcal{G}_k \\ \lambda_{vl0}, & S_n > \mathcal{G}_k \end{cases} \\ \lambda_{vrk} &= \begin{cases} \lambda_{vr0}, & S_n \leq \mathcal{G}_k \\ \lambda_{vr0} - \sum_{k=K+1}^{K+K_0} \langle \mathcal{Z}_{nk} \rangle E_{\mathcal{G}}[\varphi_{vrk}], & S_n > \mathcal{G}_k. \end{cases} \quad (25)\end{aligned}$$

The mean posterior of  $\mathcal{G}$  is a generalized Gaussian. Equation (26) is used to update  $\xi_0$ , whereas  $\xi_0$  and  $\zeta_0$  need to be classified and updated according to different situations as follows:

$$\varsigma_k = \begin{cases} v_k, & \xi_0 \leq v_k \\ \xi_0, & \xi_0 > v_k. \end{cases} \quad (26)$$

When  $S_n \leq \mathcal{G}_k$

$$\begin{aligned}\xi_k^{\varsigma_k} &= \sum_{k=K+1}^{K+K_0} \langle \mathcal{Z}_{nk} \rangle \gamma_b \xi_0 S_n^{v_k} + \xi_k^{\xi_0} \langle \sigma_{vdk}^{v_k} \rangle \\ zeta_k^{\varsigma_k} &= \begin{cases} \frac{\langle \sigma_{vdk}^{v_k} \rangle \xi_0^{\xi_0}}{\sum_{k=K+1}^{K+K_0} \langle \mathcal{Z}_{nk} \rangle \gamma_b \xi_0}, & \xi_0 < v_k \\ \frac{\langle \sigma_{vdk}^{v_k} \rangle \xi_0^{\xi_0}}{\sum_{k=K+1}^{K+K_0} \langle \mathcal{Z}_{nk} \rangle \gamma_b \xi_0 + \langle \sigma_{vdk}^{v_k} \rangle}, & \xi_0 = v_k \\ \xi_0^{\xi_0}, & \xi_0 > v_k \end{cases} \quad (27)\end{aligned}$$

where

$$\gamma_b = \frac{\Gamma\left(\frac{3}{v_k}\right)^{v_k/2}}{\Gamma\left(\frac{1}{v_k}\right)^{v_k/2}}.$$

For  $\xi_k$  and  $\zeta_k$ , when  $S_n > \mathcal{G}_k$ ,  $\sigma_{vdk}$  in (27) need to be replaced with  $\sigma_{vrk}$ .

9) *Nonconjugate of  $u_k$  and  $v_k$ :* Given that the conjugate prior of  $u_k$  and  $v_k$  is very difficult to find, to improve the accuracy of the posterior effect, we merge the single-step update of the commonly used Newton method [50] into the VM step. Therefore, we obtain the following approximate solution to the element form of  $u$ .

$$u_k = u_0 - s \cdot \frac{\mathcal{L}'_{u_k}(q(\phi))}{\mathcal{L}''_{u_k}(q(\psi))} \quad (28)$$

### Algorithm 1: EDA Based on VB for PSR

---

**input** : Model point set  $\mathcal{M} = \{\mathbf{m}_k\}_{k=1}^K$  and Target point set  $\mathcal{T} = \{t_n\}_{n=1}^N$   
**initialise**:  $o, \xi_0, \zeta_0, \varsigma_0, \kappa_0, \lambda_{ul0}, \lambda_{ur0}, \lambda_{vl0}, \lambda_{vr0}, a_0, b_0, c_0, d_0, j_0, \mu_0, \varpi_0, \epsilon_0, \boldsymbol{\varrho}_0, \mathbf{A}, \mathbf{B}, \Delta$   
1 Shape context for pre-registration  
2 **for**  $i=0; i < o; i++$  **do**  
3     **Anneal**  $\lambda_{ul0}$  and  $\lambda_{ur0}$  **to achieve coarse-to-fine**  
4     infer the indicator variables  $\vartheta$  by Eq.19  
5     updata  $\mu, \varpi, \epsilon$ , and  $\boldsymbol{\varrho}$  of noise precision  $\tau$  by Eq.18  
6     updata  $j$  of the mixture proportions  $\boldsymbol{\gamma}$  by Eq.20  
7     infer the latent variable  $\mathcal{Z}$  by Eq.21  
8     update  $\lambda_{vl}, \lambda_{vr}, \xi_0, \zeta_0, \varsigma_0$  by Eq.25,Eq.26 and Eq.27  
9     updata  $\kappa$  of the mixing proportion  $\pi$  by Eq.22  
10    infer the rigid transformation matrix  $\mathbf{A}$  by Eq.14  
11    update  $a$  and  $b$  of the precision  $\boldsymbol{\alpha}$  by Eq.15  
12    infer the weight matrix  $\mathbf{B}$  by Eq.[16]  
13    update  $c$  and  $d$  of the precision  $\boldsymbol{\beta}$  by Eq.17  
14    updata  $\lambda_{ul}$  and  $\lambda_{ur}$  of the precision  $\sigma_{ul}$  and  $\sigma_{ur}$  of transition variable  $\mathcal{X}$  by Eq.24  
15    updata  $u_k$  and  $v_k$  by Eq.28 and Eq.29  
16    infer the transition variable  $\mathcal{X}$  by Eq.23  
17    evaluate the KL divergence  
18    **if**  $ELOB < \Delta$  **then**  
19       | **return**  $\mathcal{X}$   
20    **end**  
21 **end**  
**output** : transition variable  $\mathcal{X}$

---

where  $\mathcal{L}'_{u_k}(q(\psi))$  is the first-order reciprocal of  $\mathcal{L}(q(\psi))$  to  $u_k$ ,  $\mathcal{L}''_{u_k}(q(\psi))$  is the second-order reciprocal of  $\mathcal{L}(q(\psi))$  to  $u_k$ , and  $s$  is determined via backtracking line search.  $v$  is the same as  $u$

$$v_k = v_0 - s \cdot \frac{\mathcal{L}'_{v_k}(q(\psi))}{\mathcal{L}''_{v_k}(q(\psi))}. \quad (29)$$

We then obtain Algorithm 1.

## IV. EXPERIMENT

We demonstrate the robustness, effectiveness, efficiency, and practicability of our method in this section through comparisons with state-of-the-art algorithms. Given the particularity of PSR, there is no false positive (FP) in the absence of an outlier. Therefore, the precision is always 1. However, to ensure a fair comparison, we select recall, mean, and standard deviation (STD) as evaluation indicators that are computed using the metrics in [13] and [17]. All experiments are performed on a computer with Intel Core 4 2.50-GHz CPU and 16-GB RAM. We apply our method on JetBrains PyCharm 2018.3.5 with Python 3.6 and apply all the other methods on MATLAB R2017b.

### A. Parameter Settings

The initial values of the other parameters are set as follows:  
1)  $a_0, b_0, c_0$ , and  $d_0$  are set to 0.001; 2) the elements of  $j_0$  and

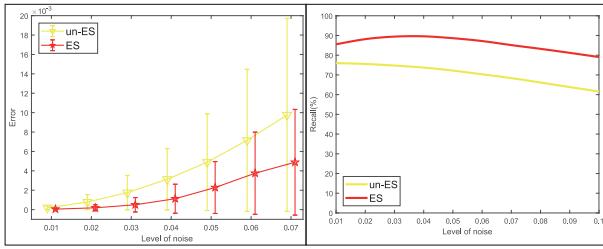


Fig. 4. Experimental error and recall under un-ES and under ES.

$\kappa_0$  are set to 1; 3)  $\mu_0$  is a zero vector,  $\varpi_0$  is set to 0.001,  $\epsilon_0$  is  $D + 1$ , and  $\varrho_0$  is set to  $I$ ; 4)  $\lambda_{v10}$  and  $\lambda_{v20}$  are set to 2; and 5)  $u$  and  $v$  are set to 2, and  $J$  is set to 3.

### B. Computational Complexity

According to (16), it can be seen that the time complexity of obtaining the weight matrix  $\mathbf{B}$  is  $O(KDK_n^2)$ . Because each row of  $\mathbf{B}$  can be solved separately with  $O(K_n)$ , time complexity and the ariational posterior of accuracy  $\beta$  need to be solved. The rigid transformation matrix  $\mathbf{A}$  still needs to be updated when dealing with rigidity and affine. Therefore, the time complexities when solving rigid and affine registration are both  $O(KDK_n^2)$ . For the nonrigid case, the time complexity of solving the posterior parameters of the transformed model is  $O(K(D + N))$ . Because there is an expectation  $\mathcal{Z}$ , and the memory requirement of the conversion variable  $\mathcal{X}$ , and the joint posterior of the latent variable can be obtained from its fully conditional expectation. The time complexity of the expected  $\mathcal{Z}$  solution can be seen as  $O(NKD^3)$  according to (21). Therefore, solving the registration problem in the nonrigid case is the time complexity  $O(K(D + N)D^2)$ .

### C. Verification of Algorithm Contributions

1) *Verifying the Effectiveness of Gaussian ES:* We first verify our proposed Gaussian ES for denoising. Fig. 4 shows the experimental error and recall under un-ES and ES. We achieve no ES by setting all parameters ( $\mu_0$ ,  $\varpi_0$ ,  $\epsilon_0$ , and  $\varrho_0$ ) to 0 and not updating them in the parameter update phase. The recall, mean, and STD in Fig. 4 are obtained by averaging the results of 100 progressive noise experiments on four datasets (105-point Chinese characters [51], 98-point fish, 302-point hands, and 96-point hearts [12]). Fig. 4 clearly shows that Gaussian ES has an excellent registration effect.

2) *Verifying the Effectiveness of AGGMM:* Fig. 5 shows the experimental error and recall of the probability model based on AGGMM and GMM. We simulate EDA based on the GMM-based probability model by selecting the VBPSM that builds the probability model based on GMM and uses VB to update the parameters. The 105-point Chinese characters and 98-point fish datasets are used in the experiment. The recall, mean, and STD in Fig. 5 are obtained by averaging the values of 100 progressive impact experiments. These experiments includes deformation (0.01–0.1), rotation (0.1–1), noise (0.01–0.1), missing (20–90), and outliers (0.1–1). Fig. 5 reveals an excellent registration effect when ES is used.

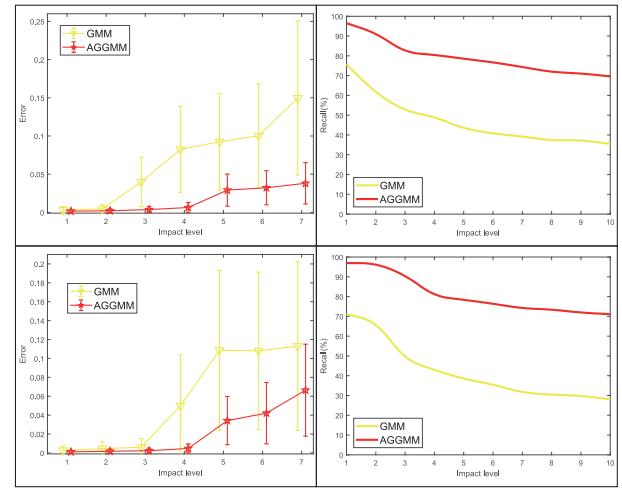


Fig. 5. Experimental error and recall of the probability model based on AGGMM and GMM. The first row presents the result of the experiment on Chinese characters dataset, and the second row presents the result of the experiment on the fish dataset.

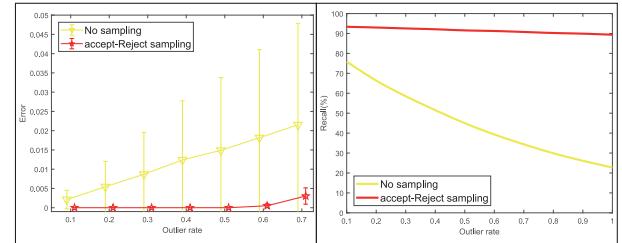


Fig. 6. Experimental error and recall of sampling and nonsampling.

3) *Verifying the Effectiveness of Accept–Reject Sampling:* We then test the effectiveness of accept–reject sampling in our method. Fig. 6 shows the experimental error and recall of sampling and nonsampling. Given that the inlier is selected when sampling, this part of the experiment only compares the effectiveness in the case of outliers. The recall, mean, and STD in Fig. 6 are obtained by averaging the values of 100 progressive outliers in each of our four datasets (105-point Chinese characters [51], 98-point fish, 302-point hands, and 96-point hearts [12]). The registration effect in the case of sampling is better than that in the case of no sampling.

### D. Validation of Algorithm Effectiveness Under Single Case

We tested the robustness of our method under deformation, rotation, missing, outlier, and noise conditions on typical datasets [12], [51]. For deformation, rotation, missing, and outliers, we present four sets of experimental results for the Chinese character and fish datasets [12], [51]. We also compare this method with state-of-the-art algorithms in terms of recall, mean, and STD, on the Chinese character, fish, hand, and heart datasets [12], [51]. The effect of some algorithms are graphically illustrated.

1) *Transformation for Deformation and Rotation:* Columns 1–4 of Fig. 7 show the registration results obtained by our method in solving different degrees of deformation and rotation. Our method achieves an almost perfect alignment in both

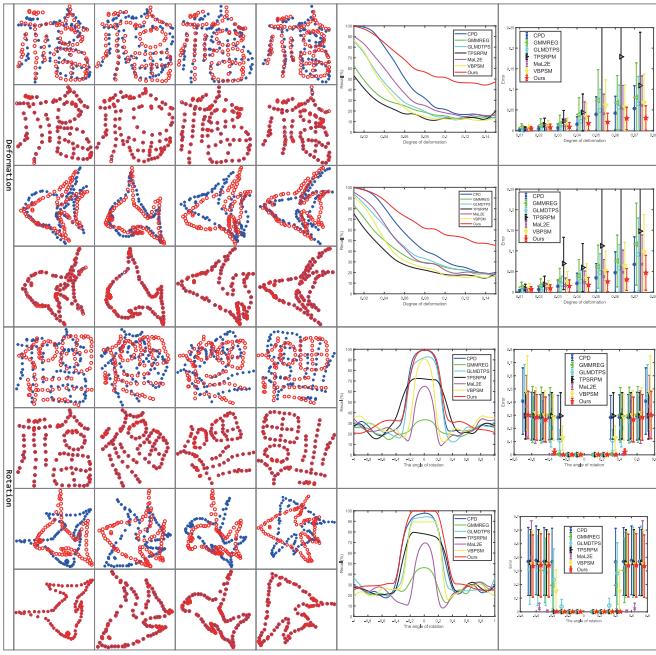


Fig. 7. Effect diagram of test deformation and rotation on a 2-D point set. Columns 1–4 present the result of our method for the 98-point fish and 105-point Chinese characters datasets, and every four rows present the corresponding deformation and rotation. Our goal is to align the *model* point set (blue “\*”) with the *target* point set (red “○”). For each set of experiments, the upper picture shows the *model* and *target* point sets, whereas the lower picture shows the registration result. Columns 5 and 6 present the results of the quantitative comparison between our method and state-of-the-art registration algorithms. The recall and error bars represent the recall, mean, and STD of registrations over 100 trials.

cases. Meanwhile, columns 5 and 6 compare the recall and error of our method with those of the other six methods. We compare the registration performance of each algorithm based on their recall, mean, and STD on all 100 samples under different deformation degrees. The quantitative comparison reveals that our method has obvious advantages over other methods, especially in the case of large deformation. For the rotation deformation test, when the rotation angle is greater than  $0.5\pi$ , all methods obtain similar recall, mean, and STD. However, the maximum rotation angle that our method can withstand is significantly higher than that withstood by other methods. This is mainly because the linear and nonlinear matrices in the selection operator explain the various transformations in PSR. In addition, our proposed AGGMM to build a probability model strengthens the evaluation of the correspondence between the point sets.

2) *Gaussian ES for Noise*: Fig. 8 shows the experimental results when dealing with noise. We compare the recall and error of our method with those of CPD [12], GMMREG [11], GLMDPTS [17], TPSRPM [2], MaL2E [13], and VBPSM [14] on the 105-point Chinese characters [51], 98-point fish, 302-point hands, and 96-point hearts datasets [12]. Columns 5 and 6 of Fig. 8 present the corresponding recall and error graphs for the case with gradually changing noise. Our method outperforms all the others on the four datasets, followed by VBPSM, owing to its also uses GMM to model noise for dealing with noise. Hence, validating the robustness of using Gaussian ES

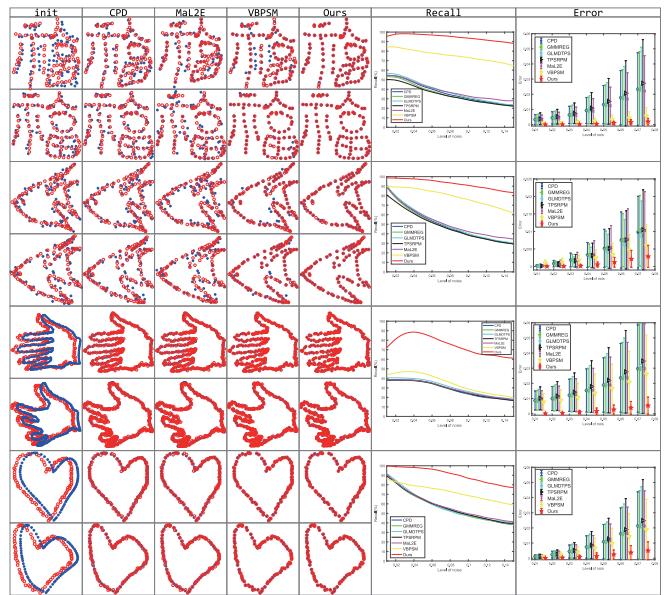


Fig. 8. Effect of testing noise on four 2-D point sets. Columns 1–5 present the results of some methods in noise processing on the four datasets, and each dataset shows 2 sets of effect pictures. Our goal is to align the *model* point set (blue “\*”) with the *target* point set (red “○”). For each set of experiments, the pictures in Column 1 show the *model* and *target* point sets, whereas the pictures in Columns 2–5 show the registration results of the corresponding method. Columns 6 and 7 present the results of the quantitative comparison between our method and state-of-the-art registration algorithms. The recall rate and error bars represent the recall rate, mean, and STD of registrations of more than 100 trials.

to deal with noise. Meanwhile, all the other methods demonstrate almost the same performance in dealing with noise, and our method can therefore be used to deal with highly complex situations.

3) *AGGMM for Missing Correspondences and Outliers*: Fig. 9 shows the experimental results when dealing with missing correspondences and outliers. We conduct experiments on the 105-point Chinese characters and 98-point fish datasets for these two cases, respectively. Columns 1–4 show the experimental results of the proposed method in this article. The fifth and sixth columns show the recall and error comparison diagrams of various methods (CPD [12], GMMREG [11], GLMDPTS [17], TPSRPM [2], MaL2E [13] and VBPSM [14]). As shown in Fig. 9, our method outperforms the rest when dealing with missing corresponding points and outliers. Although MaL2E [13] can also achieve good results, when the number of missing points and outliers is too high, the effect of MaL2E is significantly reduced. As mentioned earlier, the feature that our method can easily deal with missing and outliers mainly comes from the stable update of  $\pi_k$  under the VB framework and the flexibility of AGGMM. In this way, a better matching effect can still be achieved in the case of serious missing and outlier values.

4) *Coarse-to-Fine Registration Based on Annealing*: Fig. 10 shows our coarse-to-fine registration process from rough to precise. We show the initial position and final registration effect and then select the result at every 50 iterations. The figure also shows the annealing process.

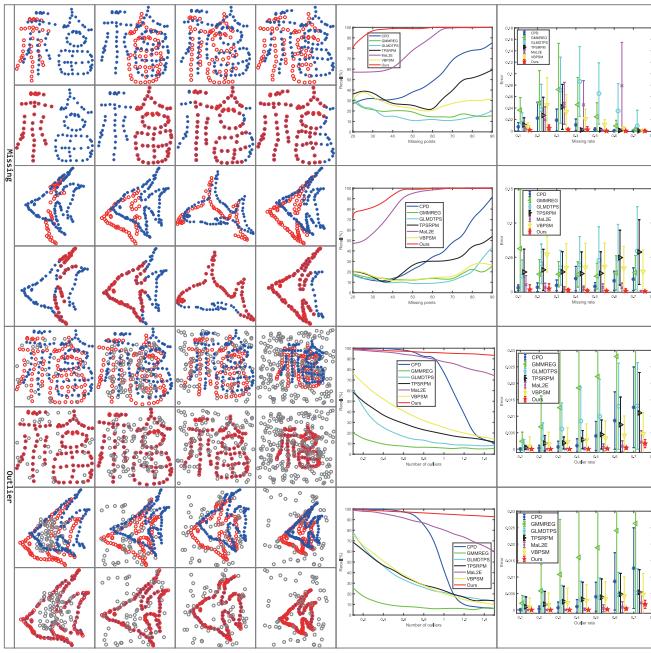


Fig. 9. Experimental results for the case with missing correspondence and outliers on a 2-D point set. Columns 1–4 show the experimental results of the proposed method. For each set of experiments, the picture in column 1 shows the *model* and *target* point sets, whereas columns 6 and 7 present the result of the quantitative comparisons between our method and state-of-the-art registration algorithms. The recall rate and error bars represent the recall rate of more than 100 trials, the registered mean, and STD.

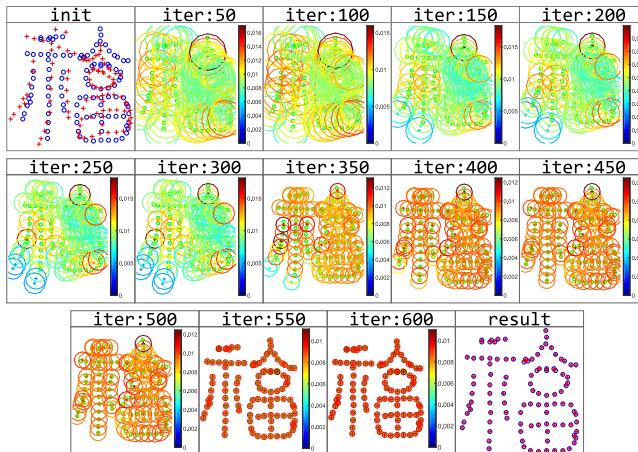


Fig. 10. Our method completes the evolution process of precise registration from coarse to fine. From left to right and top to bottom, the initial position is shown, the evolution process is selected every 50 iterations, and the final result is shown. The colored circles with “+” in the iterative process denote an approximate mixture of  $\mathcal{X}$  and  $\mathcal{G}$  with a confidence interval of 95, and  $q(\pi)$  is colored according to the mixture ratio.

Through the coarse-to-fine precise registration process, we establish a one-to-one correspondence between the point sets.

5) *Runtime:* Table I shows the running time of our method and the other algorithms under different datasets. The data presented in the table are computed as the average time for each algorithm to perform deformation, rotation, missing, outlier, and noise on the corresponding dataset after 100 experiments. The increase in the time spent by our algorithm stays relatively flat as the number of points increases. Our

TABLE I  
RUNTIME COMPARISON

| Data Set     | Fish         | Maple        | Beetle       | Cat          |
|--------------|--------------|--------------|--------------|--------------|
| Point Number | 98           | 215          | 487          | 1440         |
| CPD          | 0.265        | 0.392        | 1.400        | 15.417       |
| GMMRE        | 0.293        | 0.450        | 0.807        | <b>6.415</b> |
| GLMDTPS      | <b>0.055</b> | <b>0.187</b> | <b>0.772</b> | 18.497       |
| TPSRPM       | 0.937        | 3.221        | 30.001       | 482.036      |
| Ma-L2E       | 11.135       | 12.395       | 53.393       | 3154.128     |
| VBPSM        | 7.558        | 15.561       | 49.960       | 423.038      |
| Our          | 4.153        | 9.328        | 21.496       | 146.734      |

algorithm achieves a better running time compared with other iterative update optimization algorithms (TPSRPM, MaL2E, and VBPSM) yet slightly inferior to noniterative algorithms (e.g., CPD, GMMREG, and GLMDTPS). However, the running time of our algorithm can be shortened by using extra computer hardware. The superiority of our algorithm is also clearly reflected when large points are present. These benefit from our proposed AGGMM to build the core of EDA. Due to the flexibility and universality of AGGMM, the model becomes more comfortable as the number of points increases.

#### E. Validation of Algorithm Effectiveness Under Complex Cases

1) *Comparison of PSR Methods:* We then compare the performance of our proposed method with current PSR methods under multiple mixed situations. These algorithms include CPD [12], GMM [11], GLMDTPS [17], TPSRPM [2], Ma-L2E [13], and VBPSM [14]. We also use complex point sets for the experiment, including camel (234 points), face (317 points), horse (198 points), hand (302 points), elephant (295 points), line (60 points), maple (215 points), heart (96 points), and bird (146 points), all of which are available to the public [12], [51]. Fig. 11 shows the experimental results.

Fig. 11 shows that our method can maintain its robustness across a variety of complex situations, whereas the performance of the other methods in these situations is greatly reduced. The results of these methods grow much worse as the complexity of the situations increases. In contrast, our method can still complete the registration even when all five conditions exist. Mainly because our method is step by step when updating parameters. It can also be seen from the flow chart of our algorithm that the individual components that deal with the corresponding problem do not affect each other. Therefore, our method can achieve good robustness under multiple influencing factors.

2) *Recall Analysis With Multivariate Influence Factors:* Fig. 12 shows the registration and recall results of four algorithms on the fish, Chinese characters, hands, hearts, camels, maple leaves, elephants, horses, and lines datasets. The influencing factors are set as follows: the degree of deformation is 0–0.1, the outlier ratio is 0–1, the rotation angle is 0°–180°, the missing ratio is 0–1, and the noise level is 0–0.1. The recall rate adjusted by two influencing factors at the same time shows the performance of the corresponding algorithm. The right side of Fig. 12 presents the value of the corresponding color. It can be seen from the color comparison of each image

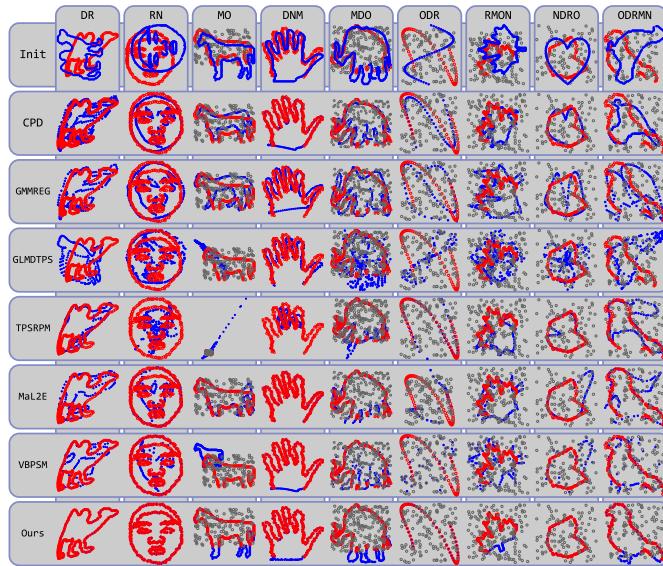


Fig. 11. Experimental results of various methods in complex situations. Init: the initial state of the *model* (blue “\*”) point set and the *target* (red “○”) point set including the outliers (gray “○”). These methods include DR ( $0.5 + 0.5\pi$ ), RN ( $0.5\pi + 0.3$ ), MO ( $0.35 + 2$ ), DNM ( $0.3 + 0.3 + 0.16$ ), MDO ( $0.33 + 0.3 + 2$ ), ORD ( $3 + 0.5\pi + 0.5$ ), RMON ( $0.35\pi + 0.25 + 2 + 0.3$ ), NDRO ( $0.3 + 0.5 + 0.35\pi + 2$ ), and ODRMN ( $2 + 0.3 + 0.35\pi + 0.33 + 0.3$ ).

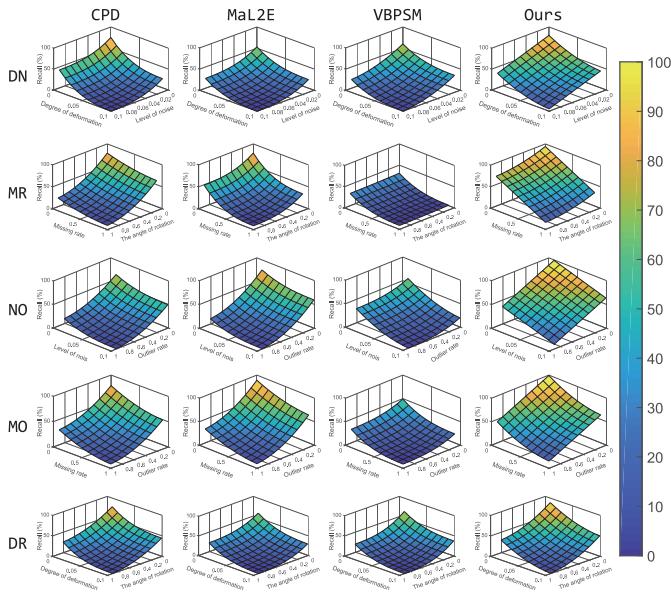


Fig. 12. Recall analysis with multivariate influencing factors. The registration recall results of various influencing factors for 100 experiments. The degree of deformation ranges from 0 to 0.1, the outlier ratio ranges from 0 to 1, the angle of rotation ranges from 0 to 1, the missing ratio ranges from 0 to 1, and the level of noise ranges from 0 to 0.1. The recall rate adjusted by two influencing factors at the same time shows the performance of the corresponding algorithm. We show the value of the corresponding color on the right side of the figure.

that our method is superior to all the other three methods in terms of registration accuracy. This experimental result once again shows that AGGMM and other components based on the EDA framework can provide more robust matching performance in the case of many outliers, large deformation, large angle rotation, and large missing ratio.

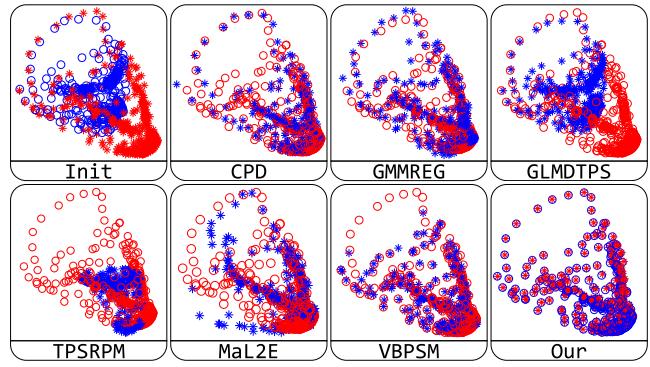


Fig. 13. Comparison of 3-D PSR experimental results for the initial state of the *model* (blue “\*”) and *target* (red “○”) point sets. The results of CPD, GMMREG, GLMDTPS, TPSRPM, MaL2E, and VBPSM with a deformation degree of 0.05 are presented.

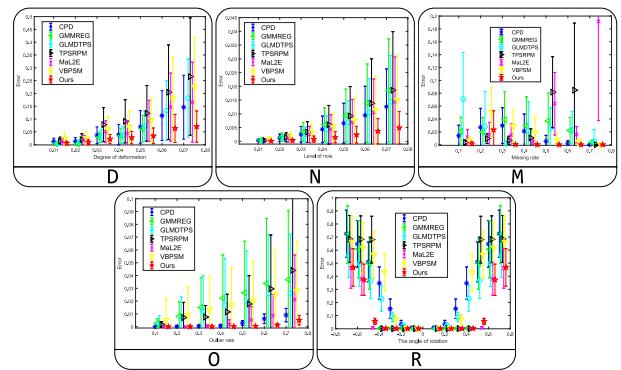


Fig. 14. Error comparison chart of the 3-D point set. We compare our method with state-of-the-art registration algorithms. The error is computed using the mean and STD obtained in 100 experiments on the 3-D “surface” point set in consideration of various influencing factors.

**3) 3-D PSR Experiment:** We then compare our method with six other algorithms on a 3-D point set [17]. Given space limitations, we only show here the experimental results of each method when the deformation degree is 0.03. The effect diagram in Fig. 13 attests to the superiority of our method over the other algorithms. We also perform experiments in other situations, but due to space limitations, the results are not presented here. Instead, we present an error comparison in Fig. 14. The chart data indicate that our method remains competent in 3-D PSR.

#### F. Verification of Algorithm Practicability

Given that image registration is the most widely used PSR, we test the performance of our method in image registration on the Oxford dataset. Fig. 15 compares the performance of our method in this dataset with other advanced algorithms. We use two classic images in the Oxford dataset for the comparison. In addition, in order to prove that our method has more effective practical cases in image registration, we also show the performance of our method across various fields of image registration, such as: low-altitude drone image registration, remote sensing image registration, hyperspectral image registration, and retinal image registration as shown in Fig. 16. The red rectangle represents the area with poor matching results.

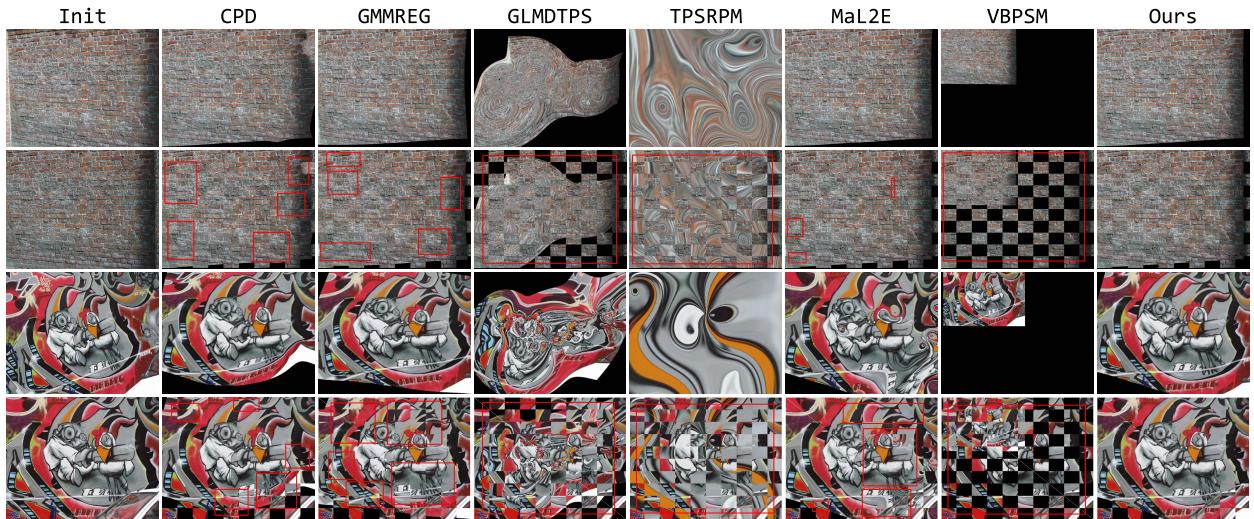


Fig. 15. Image registration of the Oxford dataset. The first column shows the registration and *model* images followed by the conversion image corresponding to each method and the  $10 \times 10$  pixel chessboard. The top two lines and the bottom two lines each represents one group. The red box marks those areas with poor matching results.

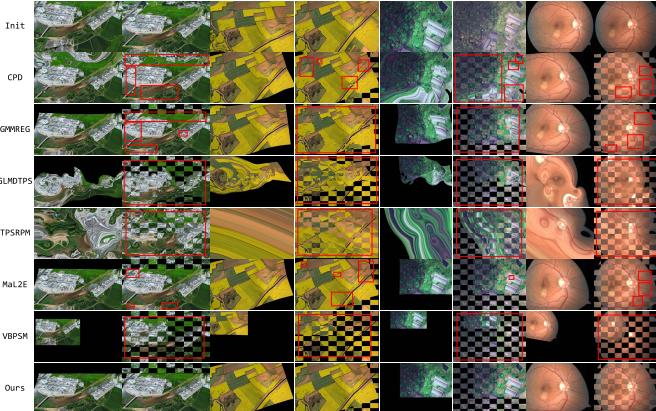


Fig. 16. Low-altitude drone images registration, remote sensing images registration, hyperspectral images registration, and retinal images registration. The red box marks those areas with poor matching results.

## V. CONCLUSION AND DISCUSSION

In this work, we have proposed a VB-based EDA for PSR. Overall, this work has the following three contributions: 1) considering the particularity of PSR, we have proposed a new selection operator, which is based on transformation and Gaussian ES to select dominant individuals and has solved the issues of deformation, rotation, and noise; 2) an EDA probability model based on AGGMM was proposed to describe the region in the solution space as comprehensively as possible to construct a probability model for irregular and regular point-set distributions, and it was proven to be robust in issues of missing and outliers; and 3) in the generation of new populations, we proposed a VB-based method to optimize the parameters of our EDA probability model to solve the objective optimization issue. A local mutation strategy has been proposed to solve the update problem of complex parameters in EDA to ensure the stable optimization of the new population; and a local search mechanism based on simulated annealing has been added to

enhance the optimization ability of the algorithm, avoid premature convergence, and realize the registration from coarse to fine.

It should be pointed out that although this article has established a probabilistic model framework for PSR, there are still several key issues that need to be studied in depth in future work: 1) the first is the optimization of annealing parameters. At present, these parameter models are all assumed to be fixed values. People have done some theoretical analysis work on this problem, such as *a priori* parameter convergence fixed-point iterative optimization method based on the second type of maximum likelihood estimation; 2) second, the outlier model based on AGGMM can also be improved by the mean outlier model, which can identify potential outliers and remove outliers. These methods combine multitask EAs [52] based on anomaly detection to optimize the objective function and reduce the influence of outliers; and 3) inlier can also be estimated by other descriptors based on multiple attributes, which can increase the difference of points. For example, the decomposition method of static and dynamic mechanisms of multiple reference points proposed by Nguyen *et al.* [53] and the 3-D point cloud autoencoder proposed by Rios *et al.* [54] can be used to describe feature points.

## ACKNOWLEDGMENT

Thanks to Qiqi He, Haifeng Wang, Zenghui Xiong, Ni Zhang, and Yang Yang for their theoretical guidance and experimental help, and the careful revision of this article. Thanks to the experimental platform provided by pattern recognition and artificial intelligence laboratory.

## REFERENCES

- [1] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.

- [2] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Understand.*, vol. 89, nos. 2–3, pp. 114–141, 2003.
- [3] L. G. Brown, "A survey of image registration techniques," *ACM Comput. Surv.*, vol. 24, no. 4, pp. 325–376, Dec. 1992.
- [4] Z. Yang, Y. Yang, K. Yang, and Z.-Q. Wei, "Non-rigid image registration with dynamic Gaussian component density and space curvature preservation," *IEEE Trans. Image Process.*, vol. 28, pp. 2584–2598, 2018.
- [5] J. Ma, H. Zhou, J. Zhao, Y. Gao, J. Jiang, and J. Tian, "Robust feature matching for remote sensing image registration via locally linear transforming," *IEEE Trans. Geosci. Remote Sens.*, vol. 53, no. 12, pp. 6469–6481, Dec. 2015.
- [6] S. Granger and X. Pennec, "Multi-scale EM-ICP: A fast and robust approach for surface registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2002, pp. 418–432.
- [7] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," *Image Vis. Comput.*, vol. 21, nos. 13–14, pp. 1145–1153, 2003.
- [8] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust Euclidean alignment of 3D point sets: The trimmed iterative closest point algorithm," *Image Vis. Comput.*, vol. 23, no. 3, pp. 299–309, 2005.
- [9] W. Lian, L. Zhang, Y. Liang, and Q. Pan, "A quadratic programming based cluster correspondence projection algorithm for fast point matching," *Comput. Vis. Image Understand.*, vol. 114, no. 3, pp. 322–333, 2010.
- [10] Y. Tsin and T. Kanade, "A correlation-based approach to robust point set registration," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, 2004, pp. 558–569.
- [11] B. Jian and B. C. Vemuri, "Robust point set registration using Gaussian mixture models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1633–1645, Aug. 2011.
- [12] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 12, pp. 2262–2275, Dec. 2010.
- [13] J. Ma, W. Qiu, J. Zhao, Y. Ma, A. L. Yuille, and Z. Tu, "Robust  $l_2e$  estimation of transformation for non-rigid registration," *IEEE Trans. Signal Process.*, vol. 63, no. 5, pp. 1115–1129, Mar. 2015.
- [14] H.-B. Qu, J.-Q. Wang, B. Li, and M. Yu, "Probabilistic model for robust affine and non-rigid point set matching," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 2, pp. 371–384, Feb. 2017.
- [15] S. Zhang, Y. Yang, K. Yang, Y. Luo, and S.-H. Ong, "Point set registration with global-local correspondence and transformation estimation," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 2688–2696.
- [16] Q. He, J. Zhou, S. Xu, Y. Yang, R. Yu, and Y. Liu, "Adaptive hierarchical probabilistic model using structured variational inference for point set registration," *IEEE Trans. Fuzzy Syst.*, vol. 28, no. 11, pp. 2784–2798, Nov. 2020.
- [17] Y. Yang, S. H. Ong, and K. W. C. Foong, "A robust global and local mixture distance based non-rigid point set registration," *Pattern Recognit.*, vol. 48, no. 1, pp. 156–173, 2015.
- [18] Q. Yang *et al.*, "Adaptive multimodal continuous ant colony optimization," *IEEE Trans. Evol. Comput.*, vol. 21, no. 2, pp. 191–205, Apr. 2017.
- [19] Q. Yang, W.-N. Chen, J. D. Deng, Y. Li, T. Gu, and J. Zhang, "A level-based learning swarm optimizer for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 22, no. 4, pp. 578–594, Aug. 2018.
- [20] X. Wen *et al.*, "A maximal clique based multiobjective evolutionary algorithm for overlapping community detection," *IEEE Trans. Evol. Comput.*, vol. 21, no. 3, pp. 363–377, Jun. 2017.
- [21] G.-G. Wang and Y. Tan, "Improving metaheuristic algorithms with information feedback models," *IEEE Trans. Cybern.*, vol. 49, no. 2, pp. 542–555, Feb. 2019.
- [22] G.-G. Wang, L. Guo, A. H. Gandomi, G.-S. Hao, and H. Wang, "Chaotic krill herd algorithm," *Inf. Sci.*, vol. 274, pp. 17–34, Aug. 2014.
- [23] B. Zhu, Z. Xu, and M. Xia, "Dual hesitant fuzzy sets," *J. Appl. Math.*, vol. 28, no. 12, pp. 3265–3275, 2012.
- [24] R. Tinós, L. Zhao, F. Chicano, and D. Whitley, "NK hybrid genetic algorithm for clustering," *IEEE Trans. Evol. Comput.*, vol. 22, no. 5, pp. 748–761, Oct. 2018.
- [25] Y. Bi, B. Xue, and M. Zhang, "Dual-tree genetic programming for few-shot image classification," *IEEE Trans. Evol. Comput.*, early access, Jul. 28, 2021, doi: [10.1109/TEVC.2021.3100576](https://doi.org/10.1109/TEVC.2021.3100576).
- [26] A. Lensen, B. Xue, and M. Zhang, "Genetic programming for manifold learning: Preserving local topology," *IEEE Trans. Evol. Comput.*, early access, Aug. 23, 2021, doi: [10.1109/TEVC.2021.3106672](https://doi.org/10.1109/TEVC.2021.3106672).
- [27] E. Bengoetxea, P. Larrañaga, I. Bloch, and A. Perchant, "Estimation of distribution algorithms: A new evolutionary computation approach for graph matching problems," in *Energy Minimization Methods in Computer Vision and Pattern Recognition*, M. Figueiredo, J. Zerubia, and A. K. Jain, Eds. Heidelberg, Germany: Springer, 2001, pp. 454–469.
- [28] B. Doerr and M. S. Krejca, "Significance-based estimation-of-distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 6, pp. 1025–1034, Dec. 2020.
- [29] W. Shi, W.-N. Chen, Y. Lin, T. Gu, S. Kwong, and J. Zhang, "An adaptive estimation of distribution algorithm for multi-policy insurance investment planning," *IEEE Trans. Evol. Comput.*, vol. 23, no. 1, pp. 1–14, Feb. 2019.
- [30] B. Doerr and W. Zheng, "Sharp bounds for genetic drift in estimation of distribution algorithms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 6, pp. 1140–1149, Dec. 2020.
- [31] F. Wang, Y. Li, A. Zhou, and K. Tang, "An estimation of distribution algorithm for mixed-variable newsvendor problems," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 479–493, Jun. 2020.
- [32] V. A. Shim, K. C. Tan, and C. Y. Cheong, "An energy-based sampling technique for multi-objective restricted Boltzmann machine," *IEEE Trans. Evol. Comput.*, vol. 17, no. 6, pp. 767–785, Dec. 2013.
- [33] H. Karshenas, R. Santana, C. Bielza, and P. Larrañaga, "Multiobjective estimation of distribution algorithm based on joint modeling of objectives and variables," *IEEE Trans. Evol. Comput.*, vol. 18, no. 4, pp. 519–542, Aug. 2014.
- [34] M. Pelikan, M. W. Hauschild, and F. G. Lobo, *Estimation of Distribution Algorithms*. Heidelberg, Germany: Springer, 2015.
- [35] X. He, Z. Zheng, and Y. Zhou, "MMES: Mixture model-based evolution strategy for large-scale optimization," *IEEE Trans. Evol. Comput.*, vol. 25, no. 2, pp. 320–333, Apr. 2021.
- [36] P. Spettel and H.-G. Beyer, "Analysis of the  $(\mu/\mu_I, \lambda) - \sigma$  self-adaptation evolution strategy with repair by projection applied to a conically constrained problem," *IEEE Trans. Evol. Comput.*, vol. 24, no. 3, pp. 593–602, Jun. 2020.
- [37] D. V. Arnold and H.-G. Beyer, "A general noise model and its effects on evolution strategy performance," *IEEE Trans. Evol. Comput.*, vol. 10, no. 4, pp. 380–391, Aug. 2006.
- [38] H.-G. Beyer and B. Sendhoff, "Toward a steady-state analysis of an evolution strategy on a robust optimization problem with noise-induced multimodality," *IEEE Trans. Evol. Comput.*, vol. 21, no. 4, pp. 629–643, Aug. 2017.
- [39] D. V. Arnold and H.-G. Beyer, "Local performance of the  $(1+1)$ -es in a noisy environment," *IEEE Trans. Evol. Comput.*, vol. 6, no. 1, pp. 30–41, Jan. 2002.
- [40] F. J. Rodriguez, C. Garcia-Martinez, and M. Lozano, "Hybrid metaheuristics based on evolutionary algorithms and simulated annealing: Taxonomy, comparison, and synergy test," *IEEE Trans. Evol. Comput.*, vol. 16, no. 6, pp. 787–800, Dec. 2012.
- [41] S. Bandyopadhyay, S. Saha, U. Maulik, and K. Deb, "A simulated annealing-based multiobjective optimization algorithm: AMOSA," *IEEE Trans. Evol. Comput.*, vol. 12, no. 3, pp. 269–283, Jun. 2008.
- [42] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.
- [43] S. J. Roberts and W. D. Penny, "Variational bayes for generalized autoregressive models," *IEEE Trans. Signal Process.*, vol. 50, no. 9, pp. 2245–2257, Oct. 2002.
- [44] J. Zhou *et al.*, "Robust variational Bayesian point set registration," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 9904–9913.
- [45] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer, 2006.
- [46] R. Horaud, F. Forbes, M. Yguer, G. Dewaele, and J. Zhang, "Rigid and articulated point registration with expectation conditional maximization," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 3, pp. 587–602, Mar. 2011.
- [47] M. Beal and Z. Ghahramani, "The variational Bayesian EM algorithm for incomplete data: With application to scoring graphical model structures," in *Bayesian Statistics*, vol. 7. Oxford, U.K.: Oxford Univ. Press, 2003, pp. 453–464.
- [48] R. A. Choudrey, *Variational Methods for Bayesian Independent Component Analysis*. Oxford, U.K.: Univ. Oxford, 2002.
- [49] M. J. Beal, "Variational algorithms for approximate Bayesian inference," Ph.D. dissertation, Dept. Gatsby Comput. Neurosci. Unit, Univ. Coll. London, London, U.K., 2003.
- [50] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, MA, USA: Cambridge Univ. Press, 2004.

- [51] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Math. Methods Biomed. Image Anal. (MMBIA)*, 2000, pp. 190–197.
- [52] C. Wang, J. Liu, K. Wu, and Z. Wu, "Solving multi-task optimization problems with adaptive knowledge transfer via anomaly detection," *IEEE Trans. Evol. Comput.*, early access, Mar. 24, 2021, doi: [10.1109/TEVC.2021.3068157](https://doi.org/10.1109/TEVC.2021.3068157).
- [53] B. H. Nguyen, B. Xue, P. Andreae, H. Ishibuchi, and M. Zhang, "Multiple reference points-based decomposition for multiobjective feature selection in classification: Static and dynamic mechanisms," *IEEE Trans. Evol. Comput.*, vol. 24, no. 1, pp. 170–184, Feb. 2020.
- [54] T. Rios, B. van Stein, T. Bäck, B. Sendhoff, and S. Menzel, "Multi-task shape optimization using a 3D point cloud autoencoder as unified representation," *IEEE Trans. Evol. Comput.*, early access, Jun. 3, 2021, doi: [10.1109/TEVC.2021.3086308](https://doi.org/10.1109/TEVC.2021.3086308).



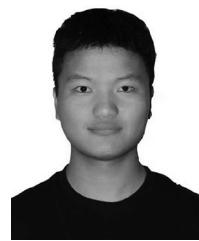
**Hualong Cao** received the B.S. degree from Suzhou University, Suzhou, China, in 2019. He is currently pursuing the M.S. degree with the School of Information Science and Technology, Yunnan Normal University, Kunming, China.

His current research interests are mainly image registration and pattern recognition.



**Haifeng Wang** is currently pursuing the B.S. degree with the School of Information Science and Technology, Yunnan Normal University, Kunming, China.

His research interest includes computer vision and remote sensing image processing.



**Zenghui Xiong** is currently pursuing the bachelor's degree with the School of Information Science and Technology, Yunnan Normal University, Kunming, China.

His research interests include natural language processing, image matching, and deep learning.



**Ni Zhang** received the B.S. degree from Yunnan Normal University, Kunming, China, in 2018, where she is currently pursuing the M.S. degree with Yunnan Chinese Language and Culture College.

Her current research interests are mainly in English translation and Thai written translation.



**Qiqi He** received the B.S. degree from Yunnan University, Kunming, China, in 2016. She is currently pursuing the M.S. degree with the School of Information Science and Technology, Yunnan Normal University, Kunming.

Her current research interests include point-set registration, image registration, and pattern recognition.



**Yang Yang** (Member, IEEE) received the master's degree from Waseda University, Tokyo, Japan, in 2007, and the Ph.D. degree from the National University of Singapore, Singapore, in 2013.

He is a Professor with the School of Information Science and Technology, Yunnan Normal University, Kunming, China. His research interest covers computer vision and pattern recognition, remote sensing, geography information system, and medical imaging.