Multi-View Point Cloud Registration Based on Evolutionary Multitasking With Bi-Channel Knowledge Sharing Mechanism

Yue Wu, Member, IEEE, Yibo Liu, Maoguo Gong, Senior Member, IEEE, Peiran Gong, Hao Li, Member, IEEE, Zedong Tang, Qiguang Miao, and Wenping Ma, Senior Member, IEEE

Abstract—Multi-view point cloud registration is fundamental in 3D reconstruction. Since there are close connections between point clouds captured from different viewpoints, registration performance can be enhanced if these connections be harnessed properly. Therefore, this article models the registration problem as multi-task optimization, and proposes a novel bi-channel knowledge sharing mechanism for effective and efficient problem solving. The modeling of multi-view point cloud registration as multi-task optimization are twofold. By simultaneously considering the local accuracy of two point clouds as well as the global consistency posed by all the point clouds involved, a fitness function with an adaptive threshold is derived. Also a framework of the co-evolutionary search process is defined for the concurrent optimization of multiple fitness functions belonging to related tasks. To enhance solution quality and convergence speed, the proposed bi-channel knowledge sharing mechanism plays its role. The intra-task knowledge sharing introduces aiding tasks that are much simpler to solve, and useful information is shared across aiding tasks and the original tasks, accelerating the search process. The inter-task knowledge sharing explores commonalities buried among the original tasks, aiming to prevent tasks from getting stuck to local optima. Comprehensive experiments conducted on model object as well as scene point clouds show the efficacy of the proposed method.

Index Terms—Evolutionary multitasking (EMT), knowledge sharing, multi-task optimization (MTO), point cloud registration.

I. INTRODUCTION

The problem of point cloud registration is fundamental in many domains, such as 3D reconstruction [1], [2], computer vision [3], [4], robotics [5], [6], etc. Due to occlusion and restricted viewpoint of each scan while obtaining data, a set of point clouds are required to reconstruct the whole object model or scene, and different coordinate frames are attached to each of these point clouds. The objective of pairwise point cloud registration is to find a rigid transformation that brings a source point cloud to the same coordinate frame as the target one. Accordingly, the objective of multi-view point cloud registration is to simultaneously estimate optimal rigid transformations for each point cloud, in order to bring them to the same coordinate frame. Pairwise registration is not trivial because low overlap between two point clouds may be the case, and there is a large search space as well. The problem becomes even more challenging when it comes to multi-view registration, for that global consistency concerning all the point clouds involved is hard to be guaranteed. Fig. 1 shows an example of multi-view point cloud registration with three point clouds involved.

Given two 3D point clouds, the source point cloud \( P = \{p_i \in \mathbb{R}^3\}_{i=1}^{N_p}\) and the target one \( Q = \{q_i \in \mathbb{R}^3\}_{i=1}^{N_q}\), the goal of pairwise point cloud registration is to find the optimal rigid transformation \( T \in SE(3)\), which is composed of a rotation matrix \( R \in SO(3)\) and a translation vector \( t \in \mathbb{R}^3\), that brings \( P \) to the same coordinate where \( Q \) lives. Generally, overlap ratio between two point clouds may vary in a wide range due to captured from different viewpoints, resulting in a varied number of outliers in the registration. Moreover, a large amount of noise may also exist in real scanned data, increasing the difficulty of the problem. Therefore, the registration should be implemented in a robust manner in order to alleviate the negative effect caused by data contamination. The registration between two point clouds can be formulated as

\[
\min_{T \in SE(3)} \sum_{i=1}^{N_p} \varphi(q_i(T))
\]

\( SE(3) = \left\{ \begin{pmatrix} R & t \\ 0 & 1 \end{pmatrix} \mid R \in SO(3), t \in \mathbb{R}^3 \right\} \)

\( SO(3) = \{ R \in \mathbb{R}^{3 \times 3} \mid R^T R = I, \det(R) = 1 \} \)
where $N_p$ is the number of points in $P$, $\varphi(\cdot)$ is a robust loss function, and the error function $g_1(\cdot)$ between matched points of the two point clouds represents the distance metric used in the registration. In this work, Euclidean distance is considered, i.e.,

$$g_1(T) = \|Tp_i - q_j\|$$

with $q_j$ the closest point in $Q$ of transformed $p_i$:

$$q_j = \arg\min_{q_k \in Q} \|Tp_i - q_k\|$$

It has to be mentioned that homogeneous coordinate is used to denote a point. For example, the coordinate of point $p_i$ is actually $[x_i, y_i, z_i, 1]^T$, with the fourth dimension always be 1. The use of homogeneous coordinate is for convenience of mathematical notation. Function (1) can solve the registration between two point clouds, which considers only local accuracy. In real applications, the problem of multi-view point cloud registration is more common and more challenging, with the objective to simultaneously bring several point clouds acquired from different viewpoints to the same coordinate frame. A simple way to achieve multi-view registration is through concatenating pairwise registration results, which is known to accumulate errors along the process, resulting in a visible gap between the first point cloud and the last one. The error accumulation comes from the violation of global consistency caused by combining registration results with local accuracy maximizing. Therefore, it is a natural thought to concern both local accuracy and global consistency when solving pairwise registration, in order to strike a balance between local and global conflicts in the problem of multi-view registration. Consider the multi-view registration problem with three point clouds involved, and denote them $P^1, P^2, P^3$. Then generally there are three pairwise registration tasks to be solved, which treat $P^1, P^2, P^3$ and $P^2, P^3, P^1$ as source and target, respectively. Accordingly, transformation matrices $T_{12}, T_{23}, T_{31}$ are solutions to each of the three tasks, with $T_{ij}$ representing the transformation of $P^i$ to the coordinate of $P^j$. Given the settings above, the global consistency between $P^1, P^2, P^3$ should be satisfied to alleviate error accumulation:

$$T_{31} \ast T_{23} \ast T_{12} = I$$

where $I$ represents the $4 \times 4$ identity matrix. Suppose a concrete point in $P^1$, after being transformed by the combination of the transformation matrices $T_{12}, T_{23}, T_{31}$, should get back to the exact position where it starts.

Inspired by the ability of human brain to tackle with multiple related tasks simultaneously, evolutionary multi-task optimization (EMTO) has recently emerged as an increasingly popular research topic in the field of evolutionary computation. Compared with its single task counterparts, i.e., traditional evolutionary algorithms, which solve only one task in a single run, EMTO is able to conduct evolutionary search concurrently on multiple optimization problems possessing varied search spaces and unique function landscapes. The superior performance of EMTO benefits from the knowledge sharing mechanism incorporated, with which useful knowledge is transferred across tasks while the search process online. Compared with traditional EAs, both convergence speed and solution quality of EMTO methods have been shown to be enhanced on continuous optimization as well as discrete optimization problems [7, 8, 9, 10, 11]. Despite its satisfying applications in the literature, the potential of EMTO has not been investigated in point cloud registration problems.

Multi-view point cloud registration can be considered as multi-task optimization, because there are close connections between point clouds captured from different viewpoints, and useful knowledge can be shared across tasks if these connections can be properly harnessed. For example, as shown in Fig. 2, there are two undirected graphs, where each node represents a certain point cloud, and each edge connects two point clouds if there exists overlap area between the two. Loops can be found in the graphs, and red dashed edges in each graph show an example of a loop containing three point clouds. Within each loop, the loop closure constraint, i.e., the global consistent constraint, should be satisfied to reach a high quality registration. However, there is no such guarantee in existing EA based methods. Therefore, this work investigates to simultaneously solve the registration of point clouds that form a loop using EMTO techniques, by concurrently optimizing fitness functions which concern both the pairwise registration accuracy and the loop closure constraint.

Existing EMTO methods have developed implicit as well as explicit knowledge sharing mechanisms, for accelerating convergence speed and enhance solution quality. Generally, implicit knowledge sharing based methods employ one single population for solving multiple optimization tasks, and information is shared across tasks through crossover operations between individuals possessing different skill factors. Although there is performance gain, the optimization for solving each task may be overly influenced by genetic materials transferred from other tasks. Due to the randomness of knowledge transfer, benefits of positive knowledge transfer may be canceled out in the presence of negative knowledge transfer, leading to limited enhancement. In contrast to implicit EMTO methods, multiple populations employed in the explicit methods facilitate a better control over the direction and extent of knowledge transfer, and multiple search mechanisms are able to be incorporated to solve different
tasks. However, the explicit EMTO methods are designed to be problem dependent, so a direct transfer of existing explicit EMTO methods to point cloud registration is not expected to work well. Therefore, in this paper an explicit EMTO method designed for point cloud registration is proposed.

The proposed method is termed as MTPCR, indicating explicit evolutionary MultiTasking Point Cloud Registration. In MTPCR, each component task has a unique population, and knowledge sharing occurs in two ways. For the purpose of intra-task knowledge sharing, aiding tasks that are much simpler to solve are introduced, and accordingly the tasks being helped (the target tasks) are called original tasks. The intra-task knowledge sharing guarantees that useful information in the form of genetic materials, is transferred between aiding tasks and original tasks, boosting the convergence speed of the original tasks. The inter-task knowledge sharing is proposed to further prevent tasks from getting stuck to local optima, by leveraging the connections posed by loop closure constraint, as shown in (4). For this purpose, best individuals of the original tasks are selected as the knowledge source, and knowledge being transferred to a certain task is learned by the combination of best individuals from other tasks. Comprehensive experiments are conducted on point clouds from model objects and indoor scenes, validating the effectiveness of the proposed approach.

Main contributions of this paper are summarized as follows.

1) This paper solves multi-view point cloud registration in an evolutionary multitasking setting. The modeling of multi-view registration as multi-task optimization enables the concurrence of pairwise accuracy and global consistency, which are both essential in problem solving.

2) An explicit evolutionary multi-task optimization approach with a bi-channel knowledge sharing mechanism is developed. The proposal of intra-task sharing aims to boost convergence speed of each component task, and the idea of inter-task sharing explores commonalities between tasks to avoid local optima.

3) To strike a balance between local accuracy and global consistency, a robust fitness function is derived to evaluate individuals with respect to each task. Local accuracy is guaranteed with an adaptive threshold, and global consistency is considered by the incorporation of loop closure constraint.

The remainder of this paper is organized as follows. Section II begins with a literature review on point cloud registration as well as recent advancements in evolutionary multitasking, and then the motivation of this paper is introduced. Section III presents details of the proposed explicit evolutionary multitasking point cloud registration method. The design of experiments is introduced in Section IV, then Section V presents and discusses the experimental results. Finally, conclusions are drawn in Section VI.

II. PRELIMINARY

In this section, literature review of related works for point cloud registration and evolutionary multitasking methods is first presented, and then motivation of this paper is analyzed.

A. Related Work

1) Point Cloud Registration: A great number of point cloud registration methods have been developed in the literature. Among these methods, the Iterative Closest Point (ICP) algorithm [12] is the most popular one and inspired a variety of subsequent works [13], [14]. Popular though ICP is, its assumption, full overlap between two point clouds, can rarely be the case in real scanned data. The trimmed ICP [15] was then developed to solve the registration with partial overlap by estimating and updating overlap ratio between two point clouds. However, the ICP family are known to be local convergent, demanding fine initializations to provide satisfactory results. Descriptors based methods are another route to achieve registration, PFH [16], FPFH [17] and ISS [18] are examples of commonly used local descriptors. After potential matches between points from different point clouds are established using computed descriptors, RANSAC [19] based approach is usually adopted later to achieve coarse registration. Efficient though these methods are, they are also shown to be sensitive to noise, and performances degrade rapidly with the increase of noise levels. Gaussian Mixture Model (GMM) based methods [20], [21], [22] assume that points in different point clouds are drawn from one or several gaussian mixtures, and expectation maximization (EM) algorithm is then used to achieve registration. Although they are more robust than ICP based ones, there is a tendency for them to get stuck to local optima as well, thus initial transformations are also required. With the rapid development of deep learning techniques, some deep learning based registration methods emerged in recent years, though there is a lack of interpretability [23]. PointNetLK [24] is the first attempt in this domain, and some other work follows [25], [26], [27]. Despite large amounts of labeled data are required during training, these trained models are shown to have preferences over some categories, leading to problems in terms of generalization. Moreover, these models are trained mainly on point clouds with full or relatively high overlap, and the ground truth rotations as well as translations are small, leading to inferior performance compared with traditional methods when it comes to real scanned data.

There are also some prior works solving registration problems using evolutionary algorithms (EAs) [28], [29], [30], [31]. Due to the advantage brought by population based search, initial transformations are not required in EA based methods. Furthermore, there is no restriction that the fitness function be convex, facilitating accurate evaluation specialized for point cloud registration which may be somewhat complex. For example, [29] proposed a surface interpenetration measure (SIM) as the optimization objective between two point clouds, which is accurate but not computationally efficient. Existing EA based methods were designed to solve pairwise registration problems where only local accuracy need to be considered, while multi-view point cloud registration which also requires the incorporation of global consistency is more general in real applications. Therefore, this paper solves the problem of multi-view registration by striking a balance between local accuracy and global consistency.

2) Evolutionary Multitasking: Evolutionary multi-task optimization (EMTO) has become an increasingly attracting research topic in the field of evolutionary optimization and
has been applied in many scientific and critical engineering problems \cite{32, 33, 34, 35}. The proposal of EMTO aims to simultaneously solve several optimization tasks, which have been considered distinct ones traditionally. Without loss of generality, assume \( k \) optimization tasks to be solved, the goal of EMTO is to search optimal solutions for each of the \( k \) tasks, i.e., \( \{ x_1^j, x_2^j, \ldots, x_k^j \} = \arg \min \{ f_1(x_1), f_2(x_2), \ldots, f_k(x_k) \} \), where \( x_j^j \in X_j \) for any \( j = 1, 2, \ldots, k \), and \( X_j \) is a \( d_j \)-dimensional search space for the \( j \)th task. In the literature, multifactorial evolutionary algorithm (MFEA) was proposed by Gupta et al. \cite{7} as the early EMTO paradigm. MFEA uses a unified search space for all the tasks involved, with the dimension of the unified search space the same as the largest one among the \( k \) tasks. A homogenous population is shared by all the \( k \) tasks, and individuals belonging to each task are distinguished by their skill factors. Knowledge transfer happens implicitly through crossover operations between individuals possessing different skill factors as the whole population evolves.

Inspired by the idea of MFEA, many subsequent EMTO algorithms have been developed to handle complex optimization tasks. On the basis of MFEA, Gupta et al. \cite{11} further extended it to the multiobjective optimization domain. Bali et al. \cite{36} proposed a linearized domain adaptation MFEA which transforms the search space of a simple task to the search space of its constitutive complex task to alleviate negative knowledge transfer. Wen and Ting \cite{37} proposed to detect parting ways where knowledge sharing become less effective and extended MFEA to many task optimization. Zhou et al. \cite{38} presented to measure the similarity between tasks from different perspectives in MFEA. Ding et al. \cite{39} proposed a generalized MFEA which is comprised of a decision variable translation strategy as well as a decision variable shuffling strategy to facilitate knowledge sharing between tasks whose optimums lie in different locations in the search space. More recently, Chen et al. \cite{40} proposed a many-task evolutionary framework, which is able to solve dynamic many-task optimization where tasks arrive at different time. Zhou et al. \cite{41} incorporated different crossover operators in the evolutionary process, and proposed a MFEA variant with an adaptive knowledge transfer strategy. In the above works, a homogenous population is shared by all the tasks involved, and knowledge sharing is achieved through crossover operations between individuals belonging to different tasks in an implicit manner.

In addition to the above implicit EMTO methods, explicit EMTO is a newly emerging paradigm proposed by Feng et al. \cite{10}. Compared to implicit algorithms where knowledge transfer happens through crossovers between individuals belonging to different tasks, knowledge transfer happens in a more controlled manner in explicit methods. In \cite{10}, best solutions found by each task can be transferred explicitly across tasks by leveraging a denoising autoencoder. After this work, Li et al. \cite{42} proposed an explicit multipopulation evolutionary framework to exploit positive knowledge transfer as well as to prevent negative knowledge transfer. Wu et al. \cite{43} proposed multitasking genetic algorithm which relies on estimating and using the bias among tasks and used the method to fuzzy system optimization. More recently, Tang et al. \cite{44} introduced an explicit knowledge transfer strategy implemented in low-dimension space via a learnable alignment matrix. Wang et al. \cite{45} proposed a multi-task evolutionary algorithm based on anomaly detection, where anomaly detection is a model designed to learn relationship between individuals of different tasks. Feng et al. \cite{46} further investigated the explicit implementation of EMTO for combinatorial optimization problems, and especially capacitated vehicle routing problem is studied. Compared with implicit EMTO methods, explicit ones have the following advantages. First, due to the independent population of each task, task-specific solution encodings can be adopted in each task, which is essential to the process of problem solving \cite{47, 48}. Next, the simplicity of incorporating different evolutionary operators in each task facilitates evolutionary search, as different evolutionary solvers have varied biases \cite{41, 49}. Further, knowledge transfer can happen in a controlled manner, both solution selection schemes such as elite selection and the direction of knowledge transfer are more flexible.

B. Motivation

1) Motivation of Multitasking in Multi-View Point Cloud Registration: Existing EA based point cloud registration algorithms were designed for the registration between two point clouds, i.e., pairwise registration. Although the whole object model or scene can be reconstructed by combining pairwise registration results, errors will accumulate along the process, leading to a visible gap between the first and the last scan. The main reason of error accumulation is the violation of global consistency among all the point clouds involved. Therefore, in this paper, a fitness function concerning both local accuracy and global consistency is derived, with the aim of striking a balance between local and global conflicts. However, the optimization of such a fitness function requires the concurrence of solutions from related pairwise registration tasks, which cannot be implemented in the traditional single-task manner. Naturally, in this paper, multi-view point cloud registration is studied in a multitasking setting.

2) Motivation of the Proposed Explicit Evolutionary Multitasking Point Cloud Registration: Existing loss functions for pairwise registration are known to be hard to optimize due to the appearance of local optima, and optimization becomes more challenging when it comes to the above mentioned loss function which also concerns relationship between solutions of related tasks. Implicit EMTO methods maintain a single population for solving all the tasks involved, where knowledge sharing happens randomly through crossovers between individuals possessing different skill factors, thus the relation between tasks cannot be fully utilized. Explicit EMTO methods have shown their potential in multi-task optimization, so the proposed method is implemented explicitly. To be specific, aiding tasks that are much simpler to solve are constructed as the knowledge source of intra-task knowledge sharing, aiming to boost the convergence of the original tasks to be solved. Furthermore, the design of inter-task knowledge sharing makes full use of relationship between tasks posed by global consistent constraint, with the aim of reducing probability of getting stuck to local optima. The two knowledge sharing mechanisms combined together can enhance performance in terms of both convergence speed and solution quality, as shown in Section V.
III. METHODOLOGY

In this section, the proposed explicit evolutionary multitasking point cloud registration is presented in detail. To better understand the whole structure, basic concepts and the framework of the proposed method are first introduced. Then the bi-channel knowledge sharing mechanism is presented. Fitness function concerning both local accuracy and global consistency is finally derived.

A. Framework

As described earlier, it is difficult to optimize the function which takes into account both local accuracy and global consistency at the same time. The proposed method aims to optimize such a function effectively and efficiently by introducing a bi-channel knowledge sharing mechanism in multitasking setting. To better understand the co-evolutionary process of the proposed method, basic notations and the framework for multi-view point cloud registration are presented.

1) **Original task:** the optimization of aforementioned function which concerns local accuracy and global consistency at the same time is called the original task. The three original tasks to be solved simultaneously are denoted $J_{1-o}, J_{2-o}, J_{3-o}$. Details of the function used in original tasks are derived in Section III-C.

2) **Aiding task:** each aiding task solves registration between the same two point clouds as its corresponding original task, but with different fitness function which is much simpler to optimize. The three aiding tasks are denoted $J_{1-a}, J_{2-a}, J_{3-a}$. The function used in aiding tasks is set according to [29].

3) **Intra-task sharing:** each aiding task and its corresponding original task solve registration between the same two point clouds with different fitness functions, i.e., they cooperate to solve the same pairwise registration. Therefore, knowledge sharing happens within each task pair is termed intra-task knowledge sharing.

4) **Inter-task sharing:** $J_1, J_2, J_3$ are the three registration tasks which solve the registration between $P^1$ and $P^2, P^2$ and $P^3, P^3$ and $P^1$, respectively. Accordingly, knowledge sharing happens across these tasks is termed inter-task knowledge sharing.

Keep the above concepts in mind, the framework of the proposed method for point cloud registration is shown in Algorithm 1. Each task is assigned with an independent population, and the evolutionary process starts from generating $n$ individuals for each population belonging to an original task or an aiding task, according to the range of values representing the search space. After fitness evaluation, $topR \times n$ top individuals are recorded for knowledge learning and sharing. Both intra-task and inter-task knowledge sharing happens in a probabilistic manner along each generation. Populations are updated with an elitist selection strategy at the end of each generation. Since differential evolution and simulated binary crossover (SBX) are incorporated in the framework, readers can refer to [50] and [51] for more details.

With the introduction of aiding tasks, there is actually an increase of fitness evaluations in each generation. However, the additional fitness evaluations are worthwhile. As can be seen in Section V, the performances on the original tasks are very poor if intra-task knowledge sharing is not used. In other words, the original tasks cannot be solved without intra-task knowledge sharing. By contrast, with the help from aiding tasks, both solution quality and convergence speed of the original tasks are enhanced by a large margin. Therefore, the introduction of aiding tasks is essential for problem solving.

B. Knowledge Sharing

The process of knowledge sharing is to capture useful traits embedded in each component task, which can be transferred among tasks to enhance performance. As aforementioned, the final objective is to optimize $J_{1-o}, J_{2-o}, J_{3-o}$ simultaneously along the evolutionary search. For this purpose, a bi-channel
knowledge sharing mechanism is developed, which is comprised of intra-task and inter-task knowledge sharing.

1) Solution Selection: Knowledge is represented in the form of solutions, i.e., individuals, so the choice of solutions used for knowledge learning and sharing has an essential impact on the improvement of the performance of the proposed bi-channel knowledge sharing mechanism. Negative knowledge sharing is considered to deteriorate performance, thus should be avoided. In the multitasking point cloud registration setting, high-quality solutions carry more useful information than inferior ones. Therefore, a proportion of top solutions in terms of objective values are maintained for knowledge learning and sharing. As shown in Algorithm 1, the input \( topR \) specifies the ratio of top individuals to be maintained in each population.

2) Intra-Task Knowledge Sharing: The idea of intra-task knowledge sharing is inspired from the evolutionary multitasking community that genetic materials from easy tasks may aid the search process of hard ones through knowledge learning and transfer. So here the concept of aiding task comes into being, whose purpose is to help the optimization of the original task. The solution of the newly constructed aiding task should be near to the original task in the search space to facilitate positive knowledge sharing, and in this way useful knowledge, i.e., genetic materials, can be transferred from aiding task to the original task in a straightforward manner. For the registration of two point clouds, the aiding task that considers only local accuracy needs not to provide a precise registration result, but is expected to produce a coarse solution near the optimal transformation. So in this paper the loss function used in [29] for coarse registration is chosen as the function to be optimized in the aiding task. As both the aiding task and the original task solve the registration between the same two point clouds, in other words, they cooperate to solve the same pairwise registration, knowledge sharing occurs between aiding task and original task is consequently termed as intra-task knowledge sharing. As shown in line 7 of Algorithm 1, intra-task knowledge sharing happens in a probabilistic manner at each generation, controlled by parameter \( p-\text{intra} \). Algorithm 2 demonstrates the process of intra-task knowledge sharing. Generally genetic materials from aiding task can help speed up the convergence of original task, but in turn the aiding task may also be aided by original task. Therefore, intra-task knowledge sharing between aiding task and original task proceeds in both directions.

3) Inter-Task Knowledge Sharing: The proposal of intra-task knowledge sharing aims to speed up the convergence of the evolutionary process, but the aiding task itself may also get stuck to local optima sometimes, resulting in less helpful or even negative knowledge sharing. Therefore, the proposal of inter-task knowledge sharing is to reduce the probability of local optima, by leveraging useful traits buried in each task. Recall the global consistent constraint in (4), which actually connects the three registration tasks closely. If the connections posed by the constraint can be harnessed properly, knowledge can be learned and shared across tasks to further enhance performance. With this idea the inter-task knowledge sharing mechanism is proposed. Using basic operations of matrix multiplication, the relationships between transformation matrices of the three registration tasks can be derived:

\[
\begin{align*}
T'_{12} &= (T_{31} \ast T_{23})^{-1} \\
T'_{23} &= (T_{12} \ast T_{31})^{-1} \\
T'_{31} &= (T_{23} \ast T_{12})^{-1}
\end{align*}
\]  

(5)

where \( T'_{i,(i+1)\%3} \) is the supposed transformation matrix of task \( J_i-o \) combined by transformation matrices from the other two tasks. \( T_{i,(i+1)\%3} \) is the transformation matrix of task \( J_i-o \), which brings point cloud \( P^n \) to the same coordinate of \( P^{(i+1)\%3} \). (5) reflects the relationship between transformation matrices of the three registration tasks. By combining best solutions from the other two tasks, useful knowledge can be learned and transferred. Accordingly, knowledge sharing occurs in this way is termed as inter-task knowledge sharing. As shown in line 10 of Algorithm 1, inter-task knowledge sharing also happens in a probabilistic manner at each generation, controlled by parameter \( p-inter \). Algorithm 3 exhibits the process of inter-task knowledge sharing.

4) Comparison Between Intra-Task and Inter-Task Knowledge Sharing: There are two main differences between intra-task and inter-task knowledge sharing. On the one hand, intra-task knowledge sharing uses a proportion of top individuals as knowledge source, while inter-task knowledge sharing chooses the best individual of each original task for knowledge learning. The different selection mechanism comes from that inter-task knowledge sharing plays its role mainly in the later stage of the evolutionary search. The combination of best individuals from the other two original tasks suffers less uncertainty than the combination of two individuals randomly selected from top individuals maintained by the other two original tasks. On the other hand, knowledge in the form of individuals can be directly transferred between aiding task and the original task in intra-task knowledge sharing, because they share the same individual representation. By contrast, inter-task knowledge sharing needs an additional decoding and encoding process, for that the learning
of knowledge is through transformation matrix multiplication as shown in (5).

Individuals in each population are in the form of six dimensional vectors, \([r_x, r_y, r_z, t_x, t_y, t_z]\). Here \(r_x, r_y\), and \(r_z\) denote the rotation angles along \(x, y, z\) axis, and \(t_x, t_y, t_z\) denote the translations along each axis. The rotation matrix \(R\) can be constructed via the multiplication of rotation matrices along each axis:

\[
R = \begin{bmatrix}
\cos(r_z) & -\sin(r_z) & 0 \\
\sin(r_z) & \cos(r_z) & 0 \\
0 & 0 & 1
\end{bmatrix} \times \begin{bmatrix}
\cos(r_y) & 0 & \sin(r_y) \\
0 & 1 & 0 \\
-\sin(r_y) & 0 & \cos(r_y)
\end{bmatrix} \\
\times \begin{bmatrix}
1 & 0 & 0 \\
0 & \cos(r_x) & -\sin(r_x) \\
0 & \sin(r_x) & \cos(r_x)
\end{bmatrix}
\]

(6)

where the rotation is performed along the order of \(x, y, z\) axis with angles \(r_x, r_y\), and \(r_z\), respectively. After obtaining the \(3 \times 3\) rotation matrix, the \(4 \times 4\) transformation matrix can also be obtained by the combination of rotation matrix \(R\) and translation vector \([t_x, t_y, t_z]^T\):

\[
T = \begin{bmatrix}
t_x \\
t_y \\
t_z \\
0^T \\
1
\end{bmatrix}
\]

(7)

After the learning of knowledge in the form of matrix multiplication, inter-task knowledge sharing requires the learned knowledge to be encoded back to the form of six dimensional vector to facilitate knowledge transfer according to Algorithm 3. Given the learned transformation matrix \(T'\), the translation vector \([t'_x, t'_y, t'_z]^T\) can be directly accessed owing to the relationship between \(T\) and \(t_x, t_y, t_z\) as shown in (7). The three rotation angles can be calculated according to (8), where \(T'[i, j]\) is the element located at \(i\)th row and \(j\)th column of transformation matrix \(T'\). Note that the row and column indices \(i\) and \(j\) start from 1.

\[
\begin{align*}
\theta_x &= \arctan(T'[3, 2] / T'[3, 3]) \\
\theta_y &= \arctan(-T'[3, 1] / \sqrt{(T'[3, 2])^2 + (T'[3, 3])^2}) \\
\theta_z &= \arctan(T'[2, 1] / T'[1, 1])
\end{align*}
\]

(8)

C. Fitness Function

In this section, a fitness function considers only local accuracy with an adaptive threshold first kicks in as the basic part, and it is further combined with global consistent constraint to obtain the final form used in original tasks. To alleviate negative impact caused by noise and outliers, point cloud registration is treated as the problem of maximizing the consensus size in this paper, i.e., the minimization of maximum consensus loss is considered, as it is supposed to diminish the influence of outliers and tolerate noise in a given level [52]. With the Euclidean distance metric in the form of (2), \(\varphi(\cdot)\) in (1) can be defined as:

\[
\varphi(x) = 1 - I(x < \varepsilon)
\]

(9)

where \(I(\cdot)\) is the indicator function that returns 1 if its input is true and 0 otherwise. The parameter \(\varepsilon\) is a threshold defining the maximum distance allowed for paired points from different point clouds to be considered inliers. Note that the relationship between paired points is not bi-directional, for example, if \(q_j\) in \(Q\) is the nearest point of \(p_i\) in \(P\), \(p_i\) is not necessarily in turn the nearest point of \(q_j\). Take into account mappings from both directions, as well as the number of points in both point clouds, the optimization objective can be formulated as:

\[
\min_{T \in SE(3)} \frac{1}{N_p + N_q} \left( \sum_{i=1}^{N_p} \varphi(q_i(T)) + \sum_{j=1}^{N_q} \varphi(p_j(T^{-1})) \right)
\]

(10)

where \(N_p\) and \(N_q\) denote the number of points of point cloud \(P\) and \(Q\), respectively. \(T^{-1}\) is the reverse of transformation matrix \(T\), representing the transformation that brings \(Q\) to the same coordinate of \(P\). The parameter \(\varepsilon\) needs tuning for different data to obtain satisfying registration results, due to varied scale of different point clouds. This paper proposes to calculate the value of the threshold with regard to the data used for registration, and the calculation can be carried out by analyzing demo registration results shown in Fig. 3. There are two uniformly distributed point clouds in 2D case to be registered, colored by orange and blue. Fig. 3(a) and Fig. 3(b) show the best and worst registration results, respectively. For the best registration result, distances between paired points are all equal to 0, meaning that the two point clouds are perfectly registered. However, such perfect matching is not likely to happen when register partially overlapping point clouds contaminated by noise and outliers. Therefore, it is reasonable to investigate the worst registration result where the registration result can still be considered as a successful try, as shown in Fig. 3(b). Without changing correct correspondence relationships between paired points, black lines in Fig. 3(b) show the maximum shift allowed between matched point pairs, and the parameter \(\varepsilon\) is determined using the maximum shift. Mathematically, given the distance \(d\) between nearest neighbors in the original point clouds, the maximum allowable shift can be calculated by means of the Pythagorean theorem:

\[
\varepsilon = \frac{1}{2} \sqrt{d^2 + d^2} = \frac{\sqrt{2}}{2} d.
\]

(11)
Simple though it is, the way of calculating threshold \( \varepsilon \) is shown to work well in 3D, where \( d \) is averaged over all the distances of each point to its nearest neighbor in the same point cloud.

By far, (10) with the proposed adaptive threshold can be well used in pairwise registration problems, but not enough to handle accumulated errors when it comes to multi-view registration. As discussed in Section I, the main reason of error accumulation is caused by the violation of global consistency. Therefore, it is a natural idea to take into account both local accuracy (10) and global consistency (4) to further improve the fitness function. To achieve a better balance between local and global conflicts, controlling parameters are added in the combination. Then the final fitness function is reformulated in the form:

\[
\min_{T \in SE(3)} \frac{1}{N_p + N_q} \left( \sum_{i=1}^{N_p} \varphi(g_i(T)) + \sum_{j=1}^{N_q} \varphi(g_j(T^{-1})) \right) + \alpha \star \gamma(\|T_{31} \cdot T_{23} \cdot T_{12} - I\|_F)
\]

(12)

Parameter \( \alpha \) is used to tune the proportion of the two parts.

IV. EXPERIMENT DESIGN

This section presents the design of experiments, including comparison methods, registration problems for validation and parameter settings.

A. Comparison Methods

To illustrate the accuracy and robustness of the proposed fitness function, four registration methods are compared in terms of registration accuracy. To illustrate the efficacy of the proposed bi-channel knowledge sharing mechanism, two multi-task optimization methods as well as the proposed method in different parameter settings are compared in terms of solution quality and convergence trends.

The four registration methods compared in terms of registration accuracy are: the trimmed iterative closest point algorithm [15], the K-means clustering based approach [53], the expectation-maximization perspective for multi-view registration [22], and the coherent point drift with local surface geometry for point cloud registration [54], which are abbreviated TrICP, KMEANS, EMPMR, and LSG-CPD, respectively. The proposed MTPCR utilizes a bi-channel knowledge sharing mechanism, to illustrate the effectiveness of intra-task and inter-task knowledge sharing, counterparts of MTPCR with different settings are also compared.

The two multi-task optimization methods compared in terms of solution quality and convergence trends are the multifactorial evolutionary algorithm [7] and the generalized multifactorial evolutionary algorithm [39], which are abbreviated MFEA and GMFEA, respectively. The proposed MTPCR utilizes a bi-channel knowledge sharing mechanism, to illustrate the effectiveness of intra-task and inter-task knowledge sharing, counterparts of MTPCR with different settings are also compared. MTPCR-Intra denotes MTPCR with only intra-task knowledge sharing, MTPCR-Inter denotes MTPCR with only inter-task knowledge sharing, and MTPCR-NKS denotes the version with no knowledge sharing.

B. Validation Problems

Point clouds used in the experiments are from Stanford 3D Scanning Repository.\(^3\) Aim@Shape project\(^4\) and Augmented ICL-NUIM Dataset.\(^5\) Table I tabulates the properties of point clouds used in experiments. Each model dataset contains point clouds acquired from different viewpoints of one object model, and each scene dataset contains partial views of indoor scenes. In addition to different types of model or scene, the scale also differs greatly. The “Overlap Ratio” column provides overlap ratios between the three point clouds used in each problem, with the lowest ratio among the three in bold font. For simplicity of notation, each problem is assigned a tag, with a, b, c or d

\(^3\)[Online]. Available: http://graphics.stanford.edu/data/3Dscanrep/
\(^4\)[Online]. Available: http://visionair.ge.imati.cnr.it/
\(^5\)[Online]. Available: http://redwood-data.org/indoor/dataset.html
added to the data name, and the lowest overlap ratio goes down from a to d. Ground truth transformations to recover the whole object model or scene are available, providing convenience for fair comparison.

C. Parameter Setting

Parameters for the proposed MTPCR: the population size \( n \) is set 100 for each of the six populations attached to optimize each task. The maximal number of generation is 60. The probability of intra-task and inter-task knowledge sharing \( p_{\text{intra}} \) and \( p_{\text{inter}} \) are both set to be 0.5. The ratio \( \alpha \) of top individuals maintained for knowledge learning is 0.1. The scale factor used in differential evolution is 0.5. The parameter in simulated binary crossover goes from 2 to 10 linearly with the evolving process. Since there is no prior information about the point clouds to be registered, the range of angles for searching is set to \([−\pi, \pi]\) along each axis, and the range of shifts is \([-\text{shift}, \text{shift}]\) with the point clouds decentralized first, where \( \text{shift} \) is the maximum length of bounding box of the point clouds involved. The search space can cover all the possibility of solutions to be found.

Parameters for counterparts of MTPCR: The only difference between MTPCR and its different versions lie in the parameter \( p_{\text{intra}} \) and \( p_{\text{inter}} \). They are set 0.5 and 0 in MTPCR-Intra, 0 and 0.5 in MTPCR-Inter, both 0 in MTPCR-NKS.

Parameters for MFEA and GMFEA: there is only one population for optimizing all the tasks, so the population size of the two methods are both 600, ensuring the same number of new individuals evaluated in each generation. The maximal number of generation is 60. The probability allowing individuals possessing different skill factors to go through crossover is 0.5. The parameter in simulated binary crossover also goes from 2 to 10 linearly. There are two more parameters in GMFEA controlling the start generation to calculate directions and the interval to update directions, respectively. They are set 10 and 5.

Parameters for fitness function: the form of fitness function used in original tasks is given in (12). Threshold \( \varepsilon \) is set automatically according to point clouds in each registration problem using \( \varepsilon = \frac{\sqrt{2}}{3} \frac{d}{d} \), where \( d \) is averaged over the three mean distances of each point cloud, with each mean distance averaged over the distances of each point to its nearest neighbor. The \( \text{tanh} \) function is used as \( \gamma(\cdot) \), mapping the value of its input to the range \([-1, 1]\). Due to the non-negativity of Frobenius norm, the value is actually mapped to the range \([0, 1]\). \( \alpha \) is set 0.05 to keep a balance between local accuracy and global consistency. Fitness function used in aiding tasks is set according to [29]. Experiments are conducted with the help of MindSpore Lite tool.\(^6\)

V. RESULTS AND DISCUSSION

This section presents, analyzes and discusses the performance of the proposed method by comparing it with registration methods as well as multi-task optimization methods over the criteria of registration accuracy and robustness, solution quality and search efficiency, respectively.

\(^6\)MindSpore. https://www.mindspore.cn/
TABLE II
REGISTRATION ERRORS OF TRICP, KMEANS, EMPMR, LSG-CPD AND MTPCR ON CORRESPONDING PROBLEMS. ALL VALUES ARE MAGNIFIED ONE THOUSAND TIMES

| Problem | TRICP | KMEANS | EMPMR | LSG-CPD | MTPCR |
|---------|-------|--------|-------|---------|-------|
| Bunny-a | 203.24 ± 65.84 | 5.57 ± 3.34 | 219.93 ± 102.76 | 2.96 ± 2.46 | 6.23 ± 0.10 |
| Bunny-b | 300.08 ± 160.13 | 12.77 ± 4.68 | 180.74 ± 181.87 | 7.04 ± 2.59 | 8.99 ± 2.59 |
| Bunny-c | 251.12 ± 119.21 | 13.93 ± 5.64 | 160.14 ± 100.48 | 6.74 ± 4.93 | 12.38 ± 0.87 |
| Dragon-a | 132.56 ± 39.91 | 7.01 ± 1.83 | 127.28 ± 146.48 | 5.54 ± 4.61 | 2.75 ± 0.05 |
| Dragon-b | 194.13 ± 67.30 | 7.31 ± 4.35 | 191.18 ± 115.91 | 6.92 ± 5.89 | 11.06 ± 0.56 |
| Dragon-c | 114.90 ± 52.30 | 4.19 ± 1.83 | 137.84 ± 123.94 | 6.35 ± 3.48 | 11.44 ± 0.11 |
| Armadillo-a | 146.92 ± 77.38 | 6.34 ± 2.74 | 162.72 ± 100.55 | 5.79 ± 3.67 | 9.36 ± 0.13 |
| Armadillo-b | 146.18 ± 71.73 | 5.28 ± 3.18 | 165.02 ± 137.31 | 4.45 ± 2.34 | 6.62 ± 0.16 |
| Armadillo-c | 162.75 ± 76.46 | 5.77 ± 3.34 | 220.59 ± 141.23 | 6.73 ± 4.16 | 29.12 ± 0.19 |
| Bullfish-a | 179.94 ± 75.50 | 7.71 ± 4.04 | 198.58 ± 138.78 | 6.63 ± 4.12 | 1.88 ± 0.01 |
| Bullfish-b | 163.17 ± 80.64 | 6.59 ± 4.17 | 194.91 ± 139.50 | 6.74 ± 4.07 | 7.64 ± 0.20 |
| Bullfish-c | 182.69 ± 86.50 | 8.74 ± 4.65 | 247.73 ± 135.58 | 5.19 ± 6.95 | 42.42 ± 30.58 |

Fig. 5. Registration results of MTPCR on indoor scenes. First row: room. Second row: office. From left to right: registration problems with tag a, b, c, and d.

good on Bunny-b and Bunny-c, and exceeds other methods by a large margin. Situations are very similar on the three problems of Dragon, where EMPMR performs best on Dragon-a with MTPCR following very closely, and the performance of MTPCR exceeds all others on Dragon-a and Dragon-c and Dragon-c, demonstrating the proposed fitness function is less sensitive to outliers compared with other methods. Analyzing overlap ratios of the problems where MTPCR does not perform best is best to find that the lowest overlap ratios are relatively high, meaning that less outliers exist in the problem. Buste-c is an exception, on which LSG-CPD obtains abnormally low rotation error but high translation error. Though not perform best on several of the problems, the errors of MTPCR follows closely with the best results. By contrast, on the majority of problems where MTPCR provides best rotation error, performance of other methods vary greatly. Through results presented in Table II, the effectiveness of the proposed fitness function that considers both local accuracy and global consistency can be verified.

B. Robustness

In addition to accuracy comparison on the registration problems without noise, experiments are also conducted on representative problems with different gaussian noise levels to demonstrate the robust of the proposed fitness function. In order to obtain noisy data, the original point clouds are first normalized to the range $[-1, 1]$, and gaussian noise with zero mean and standard deviation $\sigma$ is added to the normalized point clouds before recovering to their original scale. In the experiments, the values 0.01, 0.02 and 0.03 are used as deviations to generate noisy point clouds. Table III tabulates the registration errors on Bunny and Buddha with random gaussian noise. Fig. 6 shows the visualization of registration results obtained by the proposed MTPCR on noisy point clouds, with different colors denote different point clouds.
TABLE III
REGISTRATION ERRORS OF TRICP, KMEANS, EMPMR, LSG-CPD AND MTPCR ON REPRESENTATIVE PROBLEMS WITH DIFFERENT GAUSSIAN NOISE LEVELS. ALL VALUES ARE MAGNIFIED ONE THOUSAND TIMES. LOWER VALUES DENOTE BETTER PERFORMANCE

| Problem | Noise | TrICP | KMEANS | EMPMR | LSG-CPD | MTPCR |
|---------|-------|-------|--------|-------|---------|-------|
| Bunny-a | 0.01  | 205.04 ± 64.79 | 133.41 ± 125.13 | 3.02 ± 2.81 | 23.87 ± 0.11 | 3.10 ± 0.72 | 9.38 ± 2.82 |
|         | 0.02  | 200.74 ± 63.25 | 99.68 ± 112.96 | 2.01 ± 1.6 | 27.90 ± 0.74 | 3.14 ± 0.96 | 33.66 ± 15.54 |
|         | 0.03  | 235.37 ± 66.66 | 65.69 ± 35.97 | 1.9 ± 1.07 | 47.91 ± 0.57 | 4.03 ± 0.7 | 67.61 ± 18.97 |
| Bunny-b | 0.01  | 236.39 ± 154.4 | 161.14 ± 152.5 | 5.54 ± 4.05 | 157.60 ± 217.80 | 5.35 ± 3.55 | 237.63 ± 191.97 |
|         | 0.02  | 286.40 ± 149.1 | 145.64 ± 170.8 | 4.65 ± 4.54 | 243.29 ± 186.33 | 6.77 ± 4.61 | 501.22 ± 208.87 |
|         | 0.03  | 276.00 ± 130.3 | 160.73 ± 163.88 | 7.25 ± 7.68 | 260.85 ± 166.48 | 10.59 ± 5.35 | 340.10 ± 213.08 |
| Bunny-c | 0.01  | 252.96 ± 133.8 | 143.87 ± 139.3 | 3.83 ± 4.31 | 106.16 ± 30.9 | 3.93 ± 0.71 | 197.74 ± 130.96 |
|         | 0.02  | 255.56 ± 114.1 | 174.71 ± 200.6 | 4.37 ± 4.79 | 202.18 ± 28.4 | 7.65 ± 1.03 | 258.99 ± 167.3 |
|         | 0.03  | 255.05 ± 102.5 | 192.90 ± 140.8 | 2.47 ± 1.5 | 159.59 ± 24.48 | 10.40 ± 2.22 | 413.50 ± 120.73 |
| Buddha-a| 0.01  | 200.37 ± 33.8 | 121.95 ± 64.8 | 7.03 ± 4.24 | 3.33 ± 0.97 | 0.34 ± 0.01 | 12.76 ± 19.06 |
|         | 0.02  | 204.01 ± 67.5 | 140.09 ± 246.9 | 6.72 ± 1.81 | 15.54 ± 0.96 | 1.05 ± 0.08 | 41.29 ± 20.07 |
|         | 0.03  | 202.49 ± 66.7 | 115.31 ± 104.2 | 5.13 ± 4.93 | 54.84 ± 0.79 | 3.10 ± 0.07 | 64.00 ± 112.3 |
| Buddha-b| 0.01  | 146.71 ± 60.0 | 186.50 ± 117.7 | 9.59 ± 6.41 | 39.84 ± 0.8 | 1.43 ± 0.07 | 18.05 ± 11.33 |
|         | 0.02  | 153.54 ± 82.37 | 187.99 ± 111.61 | 6.46 ± 5.78 | 53.65 ± 4.01 | 1.71 ± 0.53 | 25.37 ± 8.88 |
|         | 0.03  | 102.94 ± 97.95 | 161.12 ± 128.8 | 5.57 ± 4.24 | 57.16 ± 2.54 | 1.31 ± 0.3 | 36.99 ± 16.8 |
| Buddha-c| 0.01  | 183.72 ± 82.7 | 126.08 ± 27.6 | 6.44 ± 5.83 | 57.68 ± 21.6 | 10.89 ± 7.96 | 38.78 ± 22.9 |
|         | 0.02  | 187.43 ± 78.82 | 111.85 ± 106.05 | 9.99 ± 6.99 | 81.80 ± 21.78 | 14.47 ± 6.77 | 53.08 ± 49.01 |
|         | 0.03  | 183.56 ± 70.2 | 199.50 ± 103.5 | 12.43 ± 10.33 | 15.76 ± 13.76 | 15.69 ± 5.45 | 70.30 ± 35.51 |

Best results of each problem is highlighted in bold font.

Fig. 6. Registration results of MTPCR on Bunny and Buddha with different gaussian noise levels. From top to bottom: gaussian noise levels with standard deviation 0.01, 0.02, and 0.03. From left to right: Bunny-a, Bunny-b, Bunny-c, Buddha-a, Buddha-b, and Buddha-c.

Compared to their noise free counterparts as shown in Table II, the accuracy of all the methods tend to decrease with the increase of noise levels, except for the KMEANS method. This may be ascribed to the clustering process used in this approach, where a fixed number of points near each cluster center are selected to estimate a new center, thus it is less sensitive to noise. The performance of TrICP seems similar whether there is noise or not, meaning that it is also less sensitive to noise. However, none of the best results are obtained by TrICP or KMEANS. On the problem Bunny-a with 0.01 gaussian noise, LSG-CPD achieves best rotation error, as in the noise free case. But with the increase of noise levels, the performance of MTPCR exceeds LSG-CPD on Bunny-a. On Buddha-a with 0.01 and 0.02 gaussian noise, EMPMR achieves best rotation error as well as in the noise free case, while it has also been exceeded when noise arises to 0.03. For all the problems listed in Table III with varied overlap ratios and different gaussian noise levels, MTPCR outperforms other methods on 15 out of 18, demonstrating the superiority in registering noisy point clouds.

C. Solution Quality

To evaluate the solution quality of the proposed approach, Table IV tabulates all the results obtained by MTPCR, MFEA, GMFEA, MTPCR-Intra, MTPCR-Inter, and MTPCR-NKS over 20 independent runs. In the table, solutions of the original tasks are presented, and the column “Task No.” represents each of the three original tasks for each problem. The column “B.Cost” and “Avg.Cost” show the best fitness values found in 20 independent runs and fitness values averaged over 20 independent runs, respectively. Superior solutions of each task are highlighted in bold font, and the last row “No. of Best” counts the number of superior solutions produced by each of these methods.
It can be observed from Table IV that in terms of B.Cost, MTPCR can always obtain solutions with the best quality on all the 69 tasks. MTPCR-Intra provides comparative best solutions on 57 tasks out of the 69 tasks, while MTPCR-Inter finds best solutions on 38 tasks, which is a little bit more than a half of the tasks. With no knowledge sharing incorporated, MTPCR-NKS can only finds best solutions on 17 of the tasks. By contrast, MFEA and GMFEA achieves best quality solutions on none of the tasks, although their solutions are near to the best on some tasks. For example, on problem "Office-a," "Office-b" and "Office-c," best solutions of MFEA and GMFEA are close to high quality solutions highlighted in bold font. The inability of MFEA and GMFEA for finding high quality solutions mainly come from their inability of local search, in that they can get close to high quality solutions but never reach the exact position where high quality solutions are. Due to the randomness of crossovers between individuals in the homogenous population of MFEA and GMFEA, the optimization process of one task may be overly influenced by genetic materials from other tasks, as registration between different point clouds has unique landscapes. Since independent populations are maintained in MTPCR-NKS, population attached to each component task guarantees the search process towards the direction of decreased function values. Although it suffers the risk of getting stuck to local optima, it has the chance of finding global optimum, and this confirms the effectiveness of conducting unique population for each task.
Fig. 7. Convergence curves of MTPCR versus MFEA, GMFEA and MTPCR-Intra, MTPCR-Inter, MTPCR-NKS on representative registration tasks. y-axis: loss averaged over 20 independent runs; x-axis: Generation. The left column of each subfigure represents aiding tasks, and the right column of each subfigure represents original tasks. (a) Convergence curves of Bunny-a. (b) Convergence curves of Dragon-a.

Qualities of solutions found by MTPCR-Inter are better than that of MTPCR-NKS, and the improvement benefits from inter-task knowledge sharing. However, without the help of intra-task knowledge sharing, the potential of inter-task sharing can not be fully explored, resulting in a limited improvement. As can be seen in Table IV, MTPCR-Intra is able to achieve high quality solutions on most of the tasks, validating the boon of constructing aiding tasks as knowledge source. With the combination of intra-task learning and inter-task learning, performance of MTPCR is further enhanced.

On the other hand, in terms of Avg.Cost which reflects the stableness of the optimization methods over 20 independent runs, it can be seen from Table IV that MTPCR takes the lead without doubt. The only four tasks that MTPCR-Intra exceeds MTPCR are “Armadillo-b J2-o,” “Room-a J1-o,” “Room-b J2-o” and “Office-b J2-0,” with MTPCR follows closely on two out of the four. Solution quality of MFEA, GMFEA, MTPCR-Inter and MTPCR-NKS are far from satisfactory. The main reason that MFEA and GMFEA perform poorly is similar as before, in that knowledge sharing happens through crossovers between individuals possessing different skill factors introduced much randomness in evolutionary search, and negative knowledge transfer will cancel out the benefits brought by positive transfer. This again confirms the boon of conducting explicit multitasking methods. MTPCR-Intra achieves the second best performance, while MTPCR-Inter performs poorly, this once more confirms the usefulness of knowledge transferred from aiding tasks. However, only intra-task knowledge sharing cannot avoid local optima, thus the average performance of MTPCR-Intra is not good enough. Although there is no significant improvement of MTPCR-Inter compared with MTPCR-NKS, inter-task knowledge sharing is very helpful to avoid local optima when combined with intra-task knowledge sharing. Therefore, the performance of MTPCR with bi-channel knowledge sharing benefits from both intra-task sharing and inter-task sharing greatly. In summary, neither intra-task knowledge sharing nor inter-task knowledge sharing is sufficient for the solving of point cloud registration, solution qualities in Table IV verify the effectiveness of proposed bi-channel knowledge sharing implemented in an explicit manner.

D. Search Efficiency

To assess the efficiency of the proposed method with bi-channel knowledge sharing mechanism, Figs. 7, 8, and 9 present the convergence curves of MTPCR, MFEA, GMFEA and MTPCR-Intra, MTPCR-Inter, MTPCR-NKS on representative registration problems averaged over 20 independent runs. In these figures, y-axis shows the actual cost without scaling, and x-axis is the number of generation. Note that there are six subfigures for each registration problem: the three in the left column correspond to the aiding tasks which are much simpler to optimize, and the three in the right column correspond to the original tasks which are hard to solve. Keeping these in mind, the curves can be analyzed as follows.

It can be observed from Fig. 7 that in terms of convergence curves of the aiding tasks, there is little difference between MTPCR, MTPCR-Intra, MTPCR-Inter and MTPCR-NKS in the first few generations, before they converge to different function values. The similar performances in the beginning can be
Fig. 8. Convergence curves of MTPCR versus MFEA, GMFEA and MTPCR-Intra, MTPCR-Inter, MTPCR-NKS on representative registration tasks. y-axis: loss averaged over 20 independent runs; x-axis: Generation. The left column of each subfigure represents aiding tasks, and the right column of each subfigure represents original tasks. (a) Convergence curves of Bunny-b. (b) Convergence curves of Buddha-b.

Fig. 9. Convergence curves of MTPCR versus MFEA, GMFEA and MTPCR-Intra, MTPCR-Inter, MTPCR-NKS on representative registration tasks. y-axis: loss averaged over 20 independent runs; x-axis: Generation. The left column of each subfigure represents aiding tasks, and the right column of each subfigure represents original tasks. (a) Convergence curves of Bunny-c. (b) Convergence curves of Buddha-c.
ascribed to the same parameter settings, while different performances in the later stage are directed by the knowledge sharing mechanism used in the four methods. MFEA and GMFEA, although finds acceptable solutions at the end of their search process, is much slower while evolving. The fast convergence speed of our method may be ascribed to the use of independent population for solving each task, while MFEA and GMFEA both maintain a homogenous population and individuals for each task are distinguished by their skill factors. It is hard to tell whether MFEA or GMFEA performs better than the other from these curves, for they seem to entwine with each other along the evolutionary search. This somewhat indicates the decision variable translation strategy proposed in GMFEA is not that helpful for point cloud registration problems. Different knowledge sharing strategies do play an important role in influencing the evolutionary search process, but it is not obvious in optimizing the aiding tasks. Another point to be noticed in the figures is that MFEA and GMFEA starts from a relatively lower initial loss compared to our proposed method. The reason of this comes from that both MFEA and GMFEA maintain a single population along the evolutionary search, and hence their population size is six times greater than the population of each task in MTPCR to guarantee the same number of fitness evaluations in each generation. Therefore, the initial value of each task in MFEA and GMFEA is picked from 6n individuals, while the initial value of MTPCR is picked from n individuals only.

Performances of these methods diverge greatly when it comes to the optimization of the original tasks. In the right column of Fig. 7(a) and Fig. 7(b), it can be seen that MTPCR consistently provides the best solutions and exceeds other methods by a large margin. MTPCR-Intra is always at the second place, with MFEA and GMFEA follows. MTPCR-Inter is only a little better than MTPCR-NKS, while both of them provide results far from satisfactory. Unlike smooth convergence curves of the aiding tasks, the cliffhanger results demonstrate the difficulty of optimizing the original tasks. Although MTPCR-NKS performs fairly well in aiding tasks, it shows inability in optimizing the original tasks due to the lack of knowledge sharing. The limited improvement of MTPCR-Inter comes from the increase of population diversity brought by inter-task knowledge sharing, but the designed inter-task knowledge sharing does not work out the way it is supposed to. The main reason is that qualities of solutions from the other two tasks to derive inter-task knowledge are also very poor, leading to some useless or even negative knowledge transfer. Best function values found by MFEA and GMFEA continue to go down after a plateau in the beginning, and it seems that high-quality solutions might be obtained if enough generations are available. However, much efficient methods are required due to limited computational resources. Both the consistent decrease and slow convergence of the two methods result from implicit knowledge transfer they use. On the one hand, implicit knowledge transfer through crossovers among tasks introduces more gene diversity, thus avoiding one task to be trapped in local optima to some extent. On the other hand, there is no guarantee that useful knowledge transfer prevails in the search process. In comparison, the performance of MTPCR-Intra, in which intra-task knowledge sharing is adopted, is much more attracting. Though not the best when optimizing aiding tasks, the results of MTPCR-Intra are satisfying. The best solutions of aiding tasks and their corresponding original tasks are near in the search space, so the direct transfer of solutions help much to solve the original tasks. The potential of MTPCR-Intra can be further explored when combined with MTPCR-Inter, and that is why the MTPCR can always perform the best. Solutions themselves for inter-task knowledge learning are poor when using MTPCR-Inter alone, but things become different when intra-task knowledge sharing involves. The incorporation of intra-task sharing guarantees a fast convergence of the original tasks, but do not ensure the global optimum for two reasons. One is that the optiums of aiding tasks and original tasks are not exactly the same, and the other is that the optimization of aiding tasks itself may also get stuck to local optima. With relatively good solutions in the later stage, positive knowledge sharing prevail in inter-task sharing, which continues to help the search process.

E. Parameter Sensitivity

1) Influence of Parameter $\alpha$ Used in (12): In the final form of fitness function, parameter $\alpha$ is used to tune the proportion of local accuracy and global consistency. The purpose of this is to balance pairwise registration result and the relationship among point clouds in multi-view registration. Therefore, $\alpha$ has a great impact on the performance of multi-view point cloud registration. Fig. 10 plots the convergence curves of representative problems under different values of parameter $\alpha$ over 20 independent runs. It can be seen that when $\alpha$ equals 0.5, the convergence characteristics are quite unsatisfying, meaning that the global consistent constraint meddles to much in the evolutionary search. With the decrease of $\alpha$, convergence curves tend to be smoother. However, value 0.2 and 0.1 still seem to overemphasize the importance of global consistency, especially in the early stage. When $\alpha$ comes to 0.05 and below, there is no big difference whatever the value is. Table V tabulates the average registration errors under different values of $\alpha$ over 20 independend runs, and the results show that the best rotation errors are obtained when
TABLE V

AVERAGE REGISTRATION ERRORS UNDER DIFFERENT VALUES OF PARAMETER α USED IN EQUATION (12) OVER 20 INDEPENDENT RUNS. ALL VALUES ARE MAGNIFIED ONE THOUSAND TIMES

| α value | Bunny-a Err_R | Bunny-a Err_\(p\) | Bunny-b Err_R | Bunny-b Err_\(p\) | Bunny-c Err_R | Bunny-c Err_\(p\) |
|---------|---------------|-------------------|---------------|-------------------|---------------|-------------------|
| 0.5     | 141.06        | 9.78              | 1657.15       | 99.23             | 1622.86       | 71.65             |
| 0.2     | 6.92          | 0.27              | 260.17        | 15.14             | 303.77        | 16.59             |
| 0.1     | 7.22          | 0.37              | 105.88        | 4.09              | 508.85        | 21.11             |
| 0.05    | 6.71          | 0.29              | **10.48**     | 0.67              | **179.35**    | 9.39              |
| 0.02    | 8.19          | 0.32              | 11.18         | 0.57              | 188.29        | 9.83              |
| 0.01    | 7.72          | 0.32              | 105.25        | 5.63              | 314.65        | 7.81              |


Fig. 11. Convergence curves of Bunny-c under different values of intra-task sharing probability \(p_{-intra}\) and inter-task sharing probability \(p_{-inter}\). y-axis: loss averaged over 20 independent runs; x-axis: Generation. (a) Convergence curves of aiding tasks. (b) Convergence curves of original tasks.

α equals 0.05, meaning a better balance compared to other α values. According to the above analysis, a good compromise between local accuracy and global consistency can be achieved when the value of α is around 0.05. Therefore, α equals 0.05 is used in the experiments of this paper.

2) Influence of Knowledge Sharing Probabilities: The intra-task and inter-task knowledge sharing probabilities \(p_{-intra}\) and \(p_{-inter}\) are two key parameters of the proposed MTPCR. Therefore, it is reasonable to analyze how they affect the process of evolutionary search. For convenience, the two parameters are set identically in each run. Fig. 11 shows the convergence curves of Bunny-c under different knowledge sharing probabilities. The value 0.0 means no knowledge sharing at all, value 1.0 means knowledge sharing happens in every generation, and other values in between. It can be seen that in terms of aiding tasks, there is no big difference whatever the sharing probability is in the first 20 generations, while convergence curves diverge in the later stage, meaning that different sharing probabilities do affect the evolutionary search. When it comes to the original tasks, the influence of varied probabilities tends to be much more apparent. When \(p_{-intra}\) and \(p_{-inter}\) are both 0.0, the search process ends up stucking to local optima due to the lack of information exchange. When they are both 0.2, situation turns better, but not good enough due to the insufficient information exchange. However, the search performance does not monotonously increase with the increase of sharing probability. When the probabilities are 1.0 and 0.8, convergence curves end up with similar loss values as that of 0.2, although they are much smoother. This can be ascribed to the excessive knowledge sharing which may meddle too much in the search process, disturbing the local search of one population. By contrast, values in the middle can make a better balance to make use of transferred knowledge and local search of one population. Therefore, sharing probability around 0.5 is preferred, and this paper uses 0.5 in the experiments.

3) Influence of Sample Ratios: Equation (11) provides a way to calculate the threshold \(\varepsilon\) automatically according to different point clouds to be registered, instead of manually set with the change of point clouds. Therefore, it is reasonable to investigate its practicality to different sample ratios. Fig. 12 depicts the variation of rotation error and translation error with different point cloud sample ratios. The sample ratios from left to right are 0.01, 0.02, 0.03, 0.04, 0.05, 0.1, 0.2, 0.3, 0.4, 0.5 in the figure. It can be seen that when the sample ratio is greater than or equal to 0.1, there is little difference in both rotation error and translation error concerning different sample ratios. Rotation error and translation error grow a little when the sample ratio goes down to 0.05, before they grow rapidly with the decrease of sample ratios below 0.05. The results show that the way of calculating threshold \(\varepsilon\) is not sensitive to sample ratios in a wide range, validating its practicality.

VI. CONCLUSION

In this paper, an explicit evolutionary multitasking point cloud registration framework, MTPCR, is proposed. In particular, by simultaneously considering local accuracy and global consistency, a fitness function is derived, and the problem of multi-view point cloud registration is modeled as multi-task optimization. To concurrently solve the optimization tasks effectively and efficiently, a bi-channel knowledge sharing mechanism is developed. The proposed intra-task knowledge sharing guarantees a fast convergence speed by knowledge transfer between aiding tasks and original tasks, and the inter-task knowledge sharing helps to avoid local optima by exploring commonalities among
tasks. The two knowledge sharing mechanisms combined together are shown to be of great importance for problem solving. Comprehensive experiments are conducted on model point clouds as well as scene point clouds with varied overlap ratios and different noise levels, confirming the utility of both the fitness function and the proposed bi-channel knowledge sharing mechanism.

REFERENCES

[1] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner, “Scannet: Richly-annotated 3D reconstructions of indoor scenes,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 5828–5839.
[2] A. Zeng, S. Song, M. Nießner, M. Fisher, J. Xiao, and T. Funkhouser, “3Dmatch: Learning local geometric descriptors from RGB-D reconstructions,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2017, pp. 199–208.
[3] J. Yang, H. Li, D. Campbell, and Y. Jia, “Go-ICP: A globally optimal solution to 3D ICP point-set registration,” IEEE Trans. Pattern Anal. Mach. Intell., vol. 38, no. 11, pp. 2214–2224, Nov. 2016.
[4] H. Lei, G. Jiang, and L. Quan, “Fast descriptors and correspondence propagation for robust global point cloud registration,” IEEE Trans. Image Process., vol. 26, no. 8, pp. 3614–3623, Aug. 2017.
[5] F. Yu, J. Xiao, and T. Funkhouser, “Semi-automatic alignment of Lidar data at city scale,” in Proc. IEEE Conf. Comput. Vis. Pattern Recognit., 2015, pp. 1722–1731.
[6] A. P. Pavlov, G. W. Ovchinnikov, D. Y. Derbyshev, D. Tseterouk, and I. V. Oseledets, “AA-ICP: Iterative closest point with Anderson acceleration,” in Proc. IEEE Int. Conf. Robot. Automat., 2018, pp. 3407–3412.
[7] A. Gupta, Y.-S. Ong, and L. Feng, “Multiobjective evolution: Toward evolutionary multi-tasking,” IEEE Trans. Evol. Comput., vol. 20, no. 3, pp. 483–507, Jun. 2016.
[8] Y. Ong and A. Gupta, “Evolutionary multitasking: A computer science view of cognitive multitasking,” Cogn. Comput., vol. 8, pp. 125–142, 2016.
[9] L. Zhou, L. Feng, J. Zhong, Y.-S. Ong, Z. Zhu, and E. Sha, “Evolutionary multitasking in combinatorial search spaces: A case study in capacitated vehicle routing problem,” in Proc. IEEE Symp. Ser. Comput. Intell., 2016, pp. 1–8.
[10] L. Feng et al., “Evolutionary multitasking via explicit autoencoding,” IEEE Trans. Cybern., vol. 49, no. 9, pp. 3457–3470, Sep. 2019.
[11] A. Gupta, Y.-S. Ong, L. Feng, and K. C. Tan, “Multiobjective multi-facial optimization in evolutionary multitasking,” IEEE Trans. Cybern., vol. 47, no. 7, pp. 1652–1665, Jul. 2017.
[12] P. 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.
[13] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in Proc. Int. Conf. 3-D Digit. Imag. Model., 2001, pp. 145–152.
[14] F. Pomerleau, F. Colas, and R. Y. Siegwart, “A review of point cloud registration algorithms for mobile robotics,” Foundations Trends Robot., vol. 4, no. 3–4, 2015.
[15] D. Chetverikov, D. Stepanov, and P. Krsek, “Robust Euclidean alignment of overlapping point sets via genetic algorithm with growth operator,” IET Image Process., vol. 8, no. 10, pp. 582–590, 2014.
[16] L. Yan, J. Tan, H. Liu, H. Xie, and C. Chen, “Automatic registration of TLS-TLS and TLS-MLS point clouds using a genetic algorithm,” Sensors, vol. 17, no. 9, 2017, Art. no. 1979.
[17] J. Zhu, D. Meng, Z. Li, S. Du, and Z. Yuan, “Robust registration of partially overlapping point sets via genetic algorithm with growth operator,” IET Image Process., vol. 3, no. 3, pp. 1201–1208, 2006.
[18] L. Silva, O. R. P. Bellon, and K. L. Boyer, “Multiview range image registration using the surface interpenetration measure,” Image Vis. Comput., vol. 25, no. 4, pp. 585–596, 2007.
[19] A. Gupta, Y.-S. Ong, and L. Feng, “Insights on transfer optimization: Because experience is the best teacher,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 2, no. 1, pp. 51–62, Feb. 2018.
[20] M. Gong, Z. Tang, H. Li, and J. Zhang, “Evolutionary multitalking with dynamic resource allocating strategy,” IEEE Trans. Evol. Comput., vol. 23, no. 5, pp. 854–869, Oct. 2019.
[21] J. Wang, B. Cen, S. Gao, Z. Zhang, and Y. Zhou, “Cooperative evolutionary framework with focused search for many-objective optimization,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 4, no. 3, pp. 398–412, Jun. 2020.
[22] X. Zheng, A. K. Qin, M. Gong, and D. Zhou, “Self-regulated evolutionary multitasking optimization,” IEEE Trans. Evol. Comput., vol. 24, no. 1, pp. 16–28, Feb. 2020.
[23] K. K. Bai, A. Gupta, L. Feng, Y. S. Ong, and T. P. Siew, “Linearized domain adaptation in evolutionary multitasking,” in Proc. IEEE Congr. Evol. Comput., 2017, pp. 1295–1302.
[24] Y.-W. Wen and C.-K. Ting, “Parting ways and reallocating resources in evolutionary multitasking,” in Proc. IEEE Congr. Evol. Comput., 2017, pp. 2404–2411.
[25] L. Zhou, L. Feng, J. Zhong, Z. Zhu, B. Da, and Z. Wu, “A study of similarity measure between tasks for multi-facial evolutionary optimization,” in Proc. Genet. Evol. Computation Conf. Companion, 2018, pp. 229–230.
[26] J. Ding, C. Yang, Y. Jin, and T. Chai, “Generalized multitasking for evolutionary optimization of expensive problems,” IEEE Trans. Evol. Comput., vol. 23, no. 1, pp. 44–58, Feb. 2019.
[27] Y. Chen, J. Zhong, L. Feng, and J. Zhang, “An adaptive archive-based evolutionary framework for many-task optimization,” IEEE Trans. Emerg. Topics Comput. Intell., vol. 4, no. 3, pp. 369–384, Jun. 2020.
[28] L. Zhou et al., “Toward adaptive knowledge transfer in multifactorial evolutionary computation,” IEEE Trans. Cybern., vol. 51, no. 5, pp. 2563–2576, May 2021.
[29] G. Li, Q. Lin, and W. Gao, “Multi-facial optimization via explicit multi-tipopulation evolutionary framework,” Inf. Sci., vol. 512, pp. 1555–1570, 2020.
[30] D. Wu and X. Tan, “Multitasking genetic algorithm (MTGA) for fuzzy system optimization,” IEEE Trans. Fuzzy Syst., vol. 28, no. 6, pp. 1050–1061, Jun. 2020.
[31] Z. Tang, M. Gong, Y. Wu, W. Liu, and Y. Xie, “Regularized evolutionary multitasking optimization: Learning to intertask transfer in aligned subspaces,” IEEE Trans. Evol. Comput., vol. 25, no. 2, pp. 262–276, Apr. 2021.
[32] 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., vol. 26, no. 2, pp. 315–338, Apr. 2021.
[33] L. Feng et al., “Explicit evolutionary multitasking for combinatorial optimization: A case study on capacitated vehicle routing problem,” IEEE Trans. Cybern., vol. 51, no. 6, pp. 3143–3156, Jun. 2021.
Maoguo Gong (Senior Member, IEEE) received the B.Eng. degree from Nanjing Tech University, Nanjing, China, in 2021. He is currently working toward the master’s degree with the School of Computer Science and Technology, Xidian University, Xi’an, China. His research interests include evolutionary computation and 3-D registration.

Hao Li (Member, IEEE) received the B.S. degree in electronic engineering and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi’an, China, in 2013 and 2018, respectively. He is currently an Associate Professor with the School of Electronic Engineering, Xidian University. His research interests include computational intelligence and machine learning.

Qiguang Miao received the M.Eng. and Doctor degrees in computer science from Xidian University, Shaanxi, China. He is currently a Professor with the School of Computer Science and Technology, Xidian University. His research interests include intelligent image processing and multiscale geometric representations for images.

Wenping Ma (Senior Member, IEEE) received the B.S. degree in computer science and technology and the Ph.D. degree in pattern recognition and intelligent systems from Xidian University, Xi’an, China, in 2003 and 2008, respectively. She is currently a Professor with the School of Artificial Intelligence, Xidian University. Her research interests include natural computing and intelligent image processing. Dr. Ma is a Member of the Chinese Institute of Electronics.