Matching Misaligned Two-Resolution Metrology Data

Yaping Wang, Erick Moreno-Centeno, and Yu Ding, *Senior Member, IEEE*

***Abstract*— Multiresolution metrology devices coexist in today's manufacturing environment, producing coordinate measurements complementing each other. Typically, the high-resolution (HR) device produces a scarce but accurate data set, whereas the low-resolution (LR) one produces a dense but less accurate data set. Research has shown that combining the two data sets of different resolutions makes better predictions of the geometric features of a manufactured part. A challenge, however, is how to effectively match each HR data point to an LR counterpart that measures approximately the same physical location. A solution to this matching problem appears a prerequisite to a good final prediction. We solved this problem by formulating it as a quadratic integer program, aiming at minimizing the maximum interpoint distance difference among all potential correspondences. Due to the combinatorial nature of the optimization model, solving it to optimality is computationally prohibitive even for a small problem size. We therefore propose a two-stage matching framework capable of solving real-life-sized problems within a reasonable amount of time. This two-stage framework consists of downsampling the full-size problem, solving the downsampled problem to optimality, extending the solution of the downsampled problem to the full-size problem, and refining the solution using iterative local search. Numerical experiments show that the proposed approach outperforms two popular point set registration alternatives, the iterative closest point and coherent point drift methods, using different performance metrics. The numerical results also show that our approach scales much better as the instance size increases, and is robust to the changes in initial misalignment between the two data sets.**

***Note to Practitioners*—The central message of this paper is that aligning multiresolution data sets is important, but solving it turns out to be a nasty problem. If one throws it into an existing off-the-shelf optimization solution package, one is unlikely to be able to get any results at all for real-life-sized problems on the current computational hardware in any practical time horizon. If one uses a heuristic approach, the downside is that the solution outcomes are not robust and could lead to considerable deterioration in the solution quality when using the combined data sets. The proposed matching framework provides a competitive robust solution to this problem and can serve as a good offline tool to aid the geometric quality control process of manufactured parts.**

***Index Terms*— Coherent point drift (CPD), coordinate measuring machine, correspondences, iterative closest point (ICP), quadratic integer programming, rigid point set registration (RPSR), two-stage matching framework (TSMF).**

I. Introduction

TO ENSURE the dimensional quality of manufactured products, metrology equipment is needed to take coordinate measurements. Two lines of metrology devices coexist today: one is the contact coordinate measuring machine (CCMM) [1] with a mechanical touch probe and the other is the optical coordinate measuring machine (OCMM) [2] equipped with a laser scanning sensory system. Fig. 1 illustrates a manufactured part being measured by the two metrology devices.

In this pair, the CCMM is the high-resolution (HR) device, which can measure up to the resolution of 0.5 μm. Comparatively, the OCMM is the low-resolution (LR) one, whose resolution is usually one order of magnitude lower than that of the CCMM [3]. On the other hand, the OCMM, due to its use of the laser scanning mechanism, can take dense measurements from a medium- to large-sized part reasonably fast, say in hours, while using a CCMM on the same part may take considerably longer time, say days. Even then the CCMM measurements do not cover the part's surface as nearly dense as those of the OCMM. In the end, the two resulting metrology data sets have measurements of *different resolutions* and *different surface-covering densities*. They form a pair of data sets complementing, rather than replacing, one another, as the LR data, with its dense coverage, captures the local and global shape features better, while the HR data, albeit scarce in number, does describe by each of its data points the true yet unknown surface in a more accurate and precise manner.

Researchers recognize the need and benefit of combining the two-resolution metrology data sets. For instance, Xia *et al.* [4] has shown that combining the two-resolution data sets produces better prediction quality of the underlying surface feature than using only one of them. A prerequisite in achieving an effective combination is to match each HR data point to an LR data point that measures approximately the same physical location on the part surface. This matching is, however, a challenge because the two data sets are often misaligned. By misalignment, we mean that the coordinates of a data point cannot serve as a unique reference to the physical location on

Manuscript received July 6, 2015; revised March 7, 2016; accepted June 14, 2016. Date of publication August 9, 2016; date of current version January 4, 2017. This paper was recommended for publication by Associate Editor L. Zhu and Editor M. Wang upon evaluation of the reviewers' comments.

The authors are with the Department of Industrial and System Engineering, Texas A&M University, College Station, TX 77843 USA (e-mail: ypwang@tamu.edu; emc@tamu.edu; yuding@tamu.edu).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. The Supplementary Material contains a data set that includes a total of nine separate data sets, which are used to form six different instance sizes mentioned in the paper. Specifically, they are: Instance 16×400 (uses data sets 'highData16' and 'lowData400'). Instance 25×400 (uses datasets 'highData25' and 'lowData400'). Instance 32×800 (uses data sets 'highData32' and 'lowData800'). Instance 50×800 (uses data sets 'highData50' and 'lowData800'). Instance 64×1560 (uses data sets 'highData64' and 'lowData1560'). Instance 100×1560 (uses data sets 'highData100' and 'lowData1560'). This material is 180 KB in size.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TASE.2016.2587219

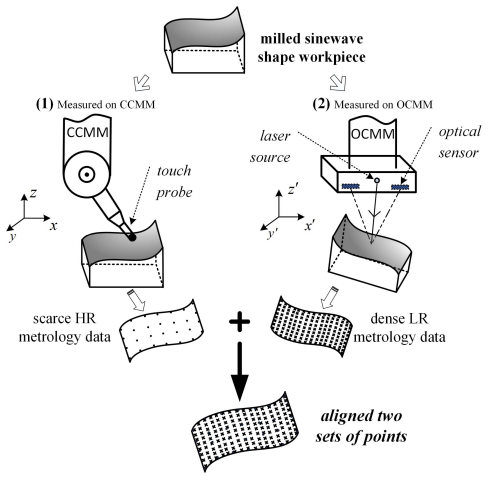


Fig. 1. Two-resolution metrology data.

the part surface where the measurement was actually taken. Given two data sets, it is not immediately clear which point in one data set corresponds to a selected point in the other data set. Misalignment happens, nearly inevitably, because: 1) the coordinate systems used to record the measurements on CCMM and OCMM are usually different and 2) the part is typically reoriented between the two measuring tasks, and hence has different poses while being measured.

The technical objective of our research is to develop a robust algorithm, i.e., misalignment insensitive, for matching the two metrology data sets so as to lay a sound foundation that enables the neighborhood linkage model in [4] to be applied for producing better surface feature predictions.

A. Problem Definition

Our problem is within the class of problems referred to as rigid point set registration (RPSR) problems (also known as point matching problems). Given two finite point sets $\mathcal{A}$ and $\mathcal{B}$, each on a different coordinate system, the RPSR is to find a rigid transformation and/or point-to-point correspondences that minimize the misalignment between transformed point set $\mathcal{A}$ and point set $\mathcal{B}$. The term *set of correspondences* or simply *correspondences* is used here to specify a complete set of point-to-point assignment between two point sets, whereas a pair of matching points specifies only one single point-to-point assignment, i.e., if $a_i \in \mathcal{A}$ is matched to $b_j \in \mathcal{B}$, then (a_i, b_j) is called a pair of matching points. Henceforth, the terms *data set* and *point set* are used interchangeably.

The metrology data matching problem addressed in this paper is defined as follows: given a sparse HR data set and a dense LR data set that are obtained by measuring the same part surface, we want to find point-to-point correspondences from the HR data set to the LR data set under an one-to-one (injection) function such that each HR point is matched to an LR point measuring approximately the same physical location.

Unlike a generic RPSR problem, our metrology data matching problem has the following unique characteristics.

1) Each data set is a collection of unstructured coordinate points. Specifically, the data sets do not include any additional information concerning the nature of the points or the relationships between points, e.g., no labels, polygon mesh representation, or other features like intensity or texture of the surface available.
2) The misalignment between the two data sets may be arbitrarily large.
3) The cardinality of the HR data set is significantly smaller than that of the LR data set, as the HR data set has significantly lower density than the LR data set, yet both data sets fully cover the part surface. In summary, the distinctive characteristic of our problem is the drastic density and cardinality differences between the data sets, distinguishing our problem from the RPSR problems previously addressed, including those whose data sets have no or negligible density differences [5]–[7], or a much smaller cardinality difference [8], or no appreciable density or cardinality difference [9].

B. Our Approach

There are three typical strategies to solve an RPSR problem.

1) Establish the point-to-point correspondences first and then recover the rigid body transformation based on the obtained correspondences (see [10] and [11]). Then, with the established correspondences at hand, one can employ a closed-form least square solution (see [12] and [13]) to recover the rigid transformation that optimally aligns (i.e., minimizes the L_2 distances between) the two metrology data sets.
2) First, estimate the rigid body transformation that best aligns the two data sets, and then find the correspondences (see [14] and [15]). Once the two data sets are aligned by applying the estimated transformation, one can simply use a closest point criterion to obtain the correspondences.
3) Find the transformation and the correspondences jointly (see [5], [6], and [16]). This strategy is generally implemented by either alternating between recovering the transformation parameters and determining the correspondences until convergence or optimizing transformation and correspondences simultaneously using a single probabilistic or optimization model.

Our solution approach follows the first strategy and intends to solve the RPSR problem by focusing on finding the point-to-point correspondences between the two data sets without the need of computing the underlying rigid body transformation beforehand. To establish the point-to-point correspondences, inspired by the matching heuristic proposed in [4], our approach makes use of the invariance property of interpoint distance (IPD) of rigid body transformations (IPD first introduced in [17]). IPD is defined for any two points/measurements in the same data set and is calculated as the Euclidean distance between the two points. The invariance property of IPD means the following: given any pair of physical points in the manufactured part, the Euclidean distance between the two points remains the same after applying any rigid body transformation. However, in our context, the invariance property of IPD holds only approximately because of the resolution scale difference between the CCMM and the OCMM and the randomness in the measurement locations

due to the different measuring plans in the CCMM and the OCMM. Specifically, given a pair of physical points that were (approximately) measured in both data sets, the distance between the pair of measurements in the first data set should be approximately equal to the distance between the pair of measurements in the second data set. Moreover, recall that both data sets cover the part surface evenly and the LR data set is significantly denser than the HR one. Therefore, it is reasonable to assume that each HR point has a corresponding LR point physically residing so close to it that we can deem both points represent approximately the same physical location on the part surface, and thus, they are considered to be a pair of matching points.

The invariance property of IPD allows us to compare the intrinsic pairwise distances internal to one data set with those internal to the other data set. To compare the internal pairwise distances (i.e., IPDs) of two pairs of matching points, with each pair in a respective data set, we compute the IPD difference associated with these two pairs of points and use this difference as a criterion (i.e., a dissimilarity measure) to evaluate how good one pair is matched to the other pair. Let us denote by $H = \{h_i \in \mathbb{R}^d : i = 1, \ldots, n_h\}$ the HR data set and $L = \{l_s \in \mathbb{R}^d : s = 1, \ldots, n_l\}$ the LR data set, where $n_l \gg n_h$ and d is usually 2 or 3. Then, given two pairs of matching points (h_i, l_s) and (h_j, l_t), the associated IPD difference is $|\|h_i - h_j\| - \|l_s - l_t\||$.

Our approach attempts to find the best correspondences between the two data sets whose largest IPD difference is minimized. This goal is achieved by formulating our matching problem as a quadratic integer programming (QIP) model (see the details in Section III-A). However, due to the combinatorial nature of the QIP model, solving its linearized version using a general mixed integer linear programming (MILP) solver is computationally prohibitive even for a small problem size (e.g., 16 HR points and 100 LR points). Even with the help of an effective search space pruning method (discussed in Section III-B), it is still difficult to solve to optimality a medium-sized problem (e.g., 16 HR points and 400 LR points). Therefore, our goal is to obtain a near optimal solution for large-size problems within a reasonable amount of time.

To achieve this goal, we propose a two-stage matching framework (TSMF) combining the branch-and-bound (B&B) search method and approximation algorithms. More specifically, our approach follows a coarse-to-fine search strategy, entailing the following major actions.

1) Downsample both data sets to smaller sizes.
2) Find the optimal correspondences for the downsampled problem.
3) Extend the optimal correspondences of the downsampled problem to the full data sets and find a complete set of correspondences.
4) Finally, employ an iterative local search procedure to refine this complete set of correspondences until there is no appreciable improvement.

The rest of this paper is organized as follows. Section II reviews the related literature. Section III presents the mathematical formulation of our problem. Section IV describes the details of our TSMF approach including the aforementioned major actions. Section V demonstrates the merit of our method through a comparison study with two popular RPSR algorithms using a pair of two-resolution metrology data sets. Finally, Section VI concludes this paper.

II. Related Work

The RPSR problem arises in many different fields, such as computer vision, image processing, pattern recognition, and computational biology, and has thus been extensively studied. Since a few outstanding surveys were published recently (see [18]–[23]), we do not intend to give another comprehensive review here. Instead, this literature review focuses on the RPSR methods that may be applied to our specific problem or that share strong similarities with our approach; most of them also take unstructured data point sets as input data sets. In this paper, we categorize the RPSR solution methods into four different groups: local deterministic optimization methods, probabilistic methods, heuristic and metaheuristic methods, and global optimization methods.

A. Local Deterministic Optimization Methods

Local deterministic methods intend to minimize the misalignment between the data sets using local neighborhood search. The most famous method is the iterative closest point (ICP) method introduced by Besl and McKay [5]. ICP iteratively registers the two point sets by alternating between the transformation estimation and the correspondence determination. ICP is widely used in many different RPSR applications due to its simplicity and good performance. The main shortcoming is that ICP can easily be trapped in a local minimum [7] without a good initial alignment. To alleviate the local minima issue, different variants of ICP have been proposed [24]–[26]. Another drawback is that ICP does not guarantee to return a set of one-to-one correspondences [27]. More recently, Linh and Hiroshi [28] combined ICP and nested annealing aiming to find the globally optimal alignment between the two point sets, yet, they pointed out that this algorithm is still likely to converge to local minima.

In addition to ICP and its variants, two other local optimization methods are available. Pottmann *et al.* [29] proposed a registration approach using instantaneous kinematics and a local quadratic approximation of a squared distance function of a surface, which they demonstrated to have better convergence than ICP. Mitra *et al.* [30] used a gradient descent based optimization technique to update the rigid transformation parameters iteratively by setting the partial derivatives of the residual error to zero and solving the resulting linear systems. Even though the gradient descent method is more stable and converges faster than ICP and its variants, a good solution from the method still heavily depends on the starting position of the point sets [21]. In fact, all the aforementioned local optimization methods (except for [28]) require more or less a good initial transformation estimation to work properly, which limits considerably their success in handling the arbitrarily large misalignment in our problem.

B. Probabilistic Models

Probabilistic point matching methods can be further divided into two subgroups. The methods in the first subgroup model

one or both of the data sets using a Gaussian mixture model (GMM) and cast the registration process as a maximum likelihood estimation problem (see [6] and [31]—[34]). In other words, these methods aim to maximize the likelihood that one data set fits another via an expectation–maximization (E-M) algorithm. One well-known approach in this subgroup is called coherent point drift (CPD) [6]. CPD poses the registration of two data sets as a probability density estimation problem and models one data set as GMM centroids. CPD is able to preserve the topological structure of the point sets and can efficiently handle large-size data sets for both rigid and nonrigid cases. Lu *et al.* [35] proposed an accelerated CPD algorithm that can register large 3D point clouds more quickly than CPD by further accelerating the Gaussian summation process during the calculation of correspondence probability matrix of CPD. Eckart *et al.* [36] proposed a GMM-based point cloud registration algorithm that applies a so-called dual-model E-M framework to achieve faster and better convergence for a wider range of initial misalignments. However, this algorithm outperforms alternative methods only when the maximum misalignment angle is less than 90° and the translation is less than the length of the data set. The methods in the second subgroup are generally known as the robust point matching algorithms, which combine the so-called soft assign technique and deterministic annealing to determine the correspondences [37]–[39]. It has been shown [39] that the process of alternating between soft assignment of correspondences and transformation estimation is equivalent to the E-M algorithm used in the first subgroup. Note that [39] also models one of its data sets using GMM (as in [6]). In comparison with ICP and its variants, probabilistic methods are more robust to initial misalignment between the two data sets. However, they can still be trapped in local minima if the misalignment degree is relatively large, for instance, CPD can handle a misalignment up to 70° [6].

C. Heuristic and Metaheuristic Methods

The third group of methods uses either heuristic or metaheuristic algorithms to find the correspondences or to estimate the transformation parameters. Xia *et al.* [4] proposed a fast heuristic matching algorithm, referred to as XiaHeur hereafter, which is based on the IPD invariance property of rigid body transformations. Specifically, XiaHeur first randomly selects one HR point as anchor point and then provisionally matches it to an LR point to form an anchor pair. With this anchor pair, XiaHeur matches the remaining HR points, one at a time; specifically, XiaHeur matches each HR point with an unmatched LR point that results in the smallest IPD difference between these newly formed matching pair and the anchor pair. Once all the remaining HR points have been matched to an LR point, we obtain one provisional set of correspondences. After this, XiaHeur matches the HR anchor point to the next LR point to form a new anchor pair and repeats the process of matching the remaining HR points. This is done until all LR points have been tried to form an anchor pair with the HR anchor point. The final set of correspondences, chosen among all the obtained provisional sets of correspondences, is the set of correspondences with the smallest maximum IPD difference. Being a heuristic algorithm, XiaHeur is fast and easy to execute but does not control the resulting maximum IPD difference in each provisional set of correspondences. Indeed, as shown in the computational results (Table V), it produces a poor set of correspondences with quite a sizeable maximum IPD difference.

As for metaheuristic algorithms, both genetic algorithms (GAs) and simulated annealing (SA) are popular choices. A GA was used in [40] to find the transformation parameters that minimize a modified Hausdorff distance between two sets of extracted image features and in [8] to find good correspondences for free-form surfaces, while SA was used in [41], in conjunction with ICP, to deal with two partially overlapping data sets; specifically, SA was used to alleviate the local-optimal-entrapping shortcoming of ICP, while ICP was used to speed up SA. Since this hybrid approach relies on the dead reckoning technique [42] to obtain a coarse position estimation, its applicability is limited.

D. Global Optimization Methods

Global optimization methods cast the RPSR problem as a global mathematical optimization model and aim to align data point sets with any initial misalignment. This group of methods intend to find either an optimal global solution through a B&B based approach or a practical near-to-optimal solution by combining the B&B approach and some approximation algorithms. Li and Hartley [43] presented a method based on the B&B search to globally register two given 3D images. This method is not applicable to our problem because it assumes equal sizes of the two sets and no translation between them. Gelfand *et al.* [11] proposed a method for registering 3D shapes based also on the pair-wise distance consistency (i.e., the IPD invariant property), but this approach relies on strong distinctive features of the input shapes to perform well. The work presented in [7] and [44] employed B&B for image matching applications. However, both methods are specialized for 2D data sets, and it is not a trivial task to generalize them to 3D cases. Raviv *et al.* [10] proposed a nonrigid registration method for 3D shapes that shares a similar coarse-to-fine matching strategy to our approach (elaborated in Section IV). The method in [10] has two limitations, however, making it not suitable for our problem: 1) it requires the data sets to have mesh structure (smooth geometric measure) and 2) its exact coarse matching model can only handle point sets with the same cardinality (see [10, constraint (3.5)]). Recently, Brown *et al.* [45] proposed a B&B-based globally optimal 2D–3D registration algorithm. However, this algorithm relies on features not found on unstructured 3D cloud points.

To evaluate the performance of TSMF in the later section, we choose one representative algorithm from each of the first three groups and compare them with our proposed TSMF, as the methods in the last group are not applicable to our problem. In the first group, ICP is selected due to its popularity and good performance. In the second group, CPD is chosen because of its robustness compared with ICP and ICP's variants. XiaHeur is selected from the third group because it is fast and simple, and thus, is most likely adopted in industrial practice. It is worth pointing out that CPD and ICP algorithms are

not randomized algorithms; they are deterministic algorithms. Specifically, even though CPD considers the alignment of the two data sets as a probability density probability problem, the E-M algorithm to solve the problem is deterministic. Thus, given an instance, running CPD multiple times will always produce the exact same solution (the same is true for ICP).

III. Problem Formulation

This section presents the mathematical formulation of our problem. We first introduce the QIP formulation and then briefly describe its linearized version followed by an effective search space pruning technique.

A. Quadratic Integer Program and Its Linearization

Given that the invariance property of IPD holds approximately for our problem, we formulate the misaligned metrology data matching problem as a min–max quadratic integer program (*minMaxQIP*). We first introduce a few notations. Denote by d_{ij}^H the IPD between point h_i and point h_j in the HR data set, i.e., $d_{ij}^H = \|h_i - h_j\|$. The IPD for the LR data set, d_{st}^L, is likewise defined. Denote by x_{is} the binary assignment variable, such that $x_{is} = 1$ if h_i is matched to l_s, and $x_{is} = 0$ otherwise. As such, the *minMaxQIP* formulation is as follows:

$$\min_{x} \quad \max_{\substack{i,j=1,\ldots,n_h \\ s,t=1,\ldots,n_l}} \left| d_{ij}^H - d_{st}^L \right| x_{is} x_{jt} \tag{1}$$

$$\text{s.t.} \quad \sum_{s=1}^{n_l} x_{is} = 1, \quad i = 1, \ldots, n_h \tag{2}$$

$$\sum_{i=1}^{n_h} x_{is} \leq 1, \quad s = 1, \ldots, n_l \tag{3}$$

$$x_{is} \in \{0, 1\}, \quad i = 1, \ldots, n_h; s = 1, \ldots, n_l. \tag{4}$$

The objective here is to minimize the maximum IPD difference (referred to as *maxIPDdiff* hereafter) across all potential correspondences between the two data sets. Constraint (2) ensures that each HR point is matched to exactly one LR point. Constraint (3) forces an LR point to be assigned to one HR point at most. Constraints (2) and (3) together make sure that the whole HR set is matched to a subset of the LR set under a one-to-one (injective) function.

minMaxQIP is mathematically equivalent to a min–max version of the quadratic assignment problem (QAP) [46], which is proved to be nondeterministic polynomial-time (NP) hard [47] and considered indeed one of the hardest combinatorial optimization problems. The state-of-the-art exact algorithms for QAP can only solve problems with up to 35 facilities [48], which is equivalent to 35 HR points and 35 LR points in our context. For manufacturing applications, we need an approach that can solve much larger instances (e.g., an HR data set size of about 100 and an LR data set over 1000) within a reasonable amount of time. To address the challenge brought forth by the larger problem size, we devised a coarse-to-fine matching strategy such that we only need to solve a much smaller size of the minMaxQIP problem to optimality, where the much smaller minMaxQIP problem is referred to as the downsampled problem.

To prepare for our solution procedure, we linearize the minMaxQIP model. First, we define new binary variables z_{isjt} to replace the quadratic term $x_{is}x_{jt}$ in the objective function and add (6) to ensure that z_{isjt} is 1 when both x_{is} and x_{jt} are 1. Then, we change the original min–max objective function to a minimization one by defining a new continuous variable u to replace the inner maximization, i.e., $\max|d_{ij}^H - d_{st}^L| z_{isjt}$. To reflect that u is the maximum over all combinations of i, j, s, and t, constraint (7) is added, which says that the maximum over all possible terms is greater than or equal to every one of individual terms. The linearized model is given by

$$\min_{x} \quad u \tag{5}$$

$$\text{s.t. } (2-4)$$

$$x_{is} + x_{jt} \leq z_{isjt} + 1, \quad i, j = 1, \ldots, n_h \quad s, t = 1, \ldots, n_l; i < j, s \neq t \tag{6}$$

$$u \geq \left| d_{ij}^H - d_{st}^L \right| z_{isjt}, \quad i, j = 1, \ldots, n_h \quad s, t = 1, \ldots, n_l; i < j, s \neq t \tag{7}$$

$$z_{isjt} \in \{0, 1\}, \quad i, j = 1, \ldots, n_h; \; s, t = 1, \ldots, n_l \quad i < j, s \neq t. \tag{8}$$

B. Search Space Pruning Technique

Before we present the details of our proposed two-stage solution approach, we present that a simple search space pruning that remarkably decreases the computation time for solving the linearized optimization problem.

Let u^* be the optimal solution (maxIPDdiff) of the linearized model. The key of the search space pruning is to find a tight upper bound for u^*, which we call the search space threshold, denoted by $\bar{T}$. That is, $\bar{T}$ is a value that we know for sure is larger than u^*, but we hope is not much larger than u^*. Consequently, one can reject all possible pairwise correspondences whose IPD difference is greater than $\bar{T}$. Specifically, given two HR points (h_i, h_j) and two LR points (l_s, l_t), if the IPD difference between them is greater than $\bar{T}$, then only one HR point can be matched to one of the two LR points. This is to say, if h_i is matched to l_s, then h_j cannot be matched to l_t, or *vice versa*.

Finding a proper $\bar{T}$ is essential for the search space pruning technique to work effectively. $\bar{T}$ should be as small as possible to eliminate a large amount of the potential pairwise correspondences. However, it cannot be too small; otherwise, it may block off the optimal solution; that is, it may render the pruned model infeasible—in which case one would need to increase $\bar{T}$ and resolve the model.

To find an effective and safe $\bar{T}$, we consider an ideal situation where measurements in both data sets are perfectly evenly spaced over a flat part surface. A small section of a hypothetical part surface under this ideal situation is shown in Fig. 2, where each cross represents an LR point and each circle stands for an HR point and τ denotes the maximum distance between an LR point and its closest neighbor in the LR data set. As shown in Fig. 2, to estimate the largest possible IPD difference, we examine the worst case scenario where every HR point sits almost at the center of its closest four surrounding LR points and the HR point h' (h'') sits a little bit closer to LR point l^b (l^c) than to LR point l^a (l^d).

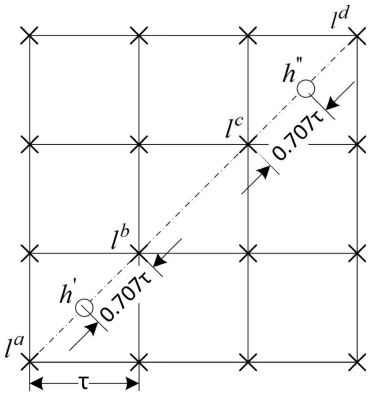


Fig. 2. Ideal case for choosing a proper $\bar{T}$.

As such, the HR points h' and h'' should be matched to the LR points l^b and l^c, respectively, and the IPD difference between these two pairs of matching points is 1.414τ. Under the ideal case, this 1.414τ is approximately the largest value u^* can take, as one can imagine that no matter where we move the HR points, the IPD difference is likely to get no greater. This understanding suggests that $\bar{T}$ can be set to 1.414τ. In practice, of course, the data sets are not perfectly evenly spaced and the part surfaces are usually curved. Consequently, 1.414τ may not be an upper bound of u^*. We believe that this value still represents an effective threshold. To be safer, we relax $\bar{T}$ to 1.5τ. Our later numerical analysis in Section V shows that this search space pruning technique on average eliminates almost 80% of binary variables and never yielded an infeasible pruned model.

It should be noted that in this paper, we assume that the measurements in both HR and LR data sets are evenly spaced over the part surface. This assumption is realistic because the evenly spaced measurements can be readily obtained using today's metrology technology [4]. Said this, we acknowledge that there might be circumstances where taking evenly measurements throughout the surface may not be desirable. For example, if the surface is very wiggly, it may be preferred to take denser measurements near the locations with high curvature than other relative flatter areas so that the critical surface features are captured without an undue increase of measurements (especially in the HR data set). Under these circumstances, it is desirable and practical to maintain the measurements' evenness only locally (with higher density measurements evenly distributed over the high curvature areas and lower density measurements evenly distributed over the not very curvy locations). The Appendix explains in detail why our choice of $\bar{T} = 1.5\tau$ is also appropriate under this circumstance.

To implement the search space pruning technique, for two pairs of potential matching points (h_i, l_s) and (h_j, l_t), we do not define variable z_{isjt} if their IPD difference is greater than $\bar{T}$. To mathematically reflect this in the linearized model [see (2)–(8)], we include constraint (9) to the linearized model and change (6) to (10) as follows:

$$x_{is} + x_{jt} \le 1, \quad i, j = 1, \ldots, n_h;\ s, t = 1, \ldots, n_l \\ i < j;\ s \ne t;\ \text{if } \left|d_{ij}^H - d_{st}^L\right| > \bar{T} \qquad (9)$$

$$x_{is} + x_{jt} \le z_{isjt} + 1,\ i, j = 1, \ldots, n_h;\ s, t = 1, \ldots, n_l \\ i < j;\ s \ne t;\ \text{if } \left|d_{ij}^H - d_{st}^L\right| \le \bar{T}. \qquad (10)$$

A nice property of the pruning technique is that the optimal objective value of the pruned model is independent of the value of $\bar{T}$ in the following sense. If $\bar{T} < u^*$, then the pruned model is infeasible (and thus one would need to increase $\bar{T}$ and resolve the model); while if $\bar{T} \ge u^*$, then the optimal objective value of the pruned model will be equal to u^*. To see this, note that the pruning technique eliminates only the pairwise correspondences (corresponding to the binary variables z_{ijst} of the linearized model) whose IPD differences are greater than $\bar{T}$. In other words, only suboptimal solutions are discarded. Therefore, as long as $\bar{T} \ge u^*$, the pruned model has the same optimal objective value as the unpruned model.

IV. Two-Stage Matching Framework

Even though the search space pruning technique significantly decreases the solution time of small instances, it does not do so sufficiently to medium-to-large instances. Thus, in order to tackle problems with real-life sizes, we relax our optimization goal from solving to optimality to finding a robust near-optimal solution and devise a TSMF to accomplish this relaxed goal.

We start with an overview of our solution framework. Our solution approach is conducted in two stages and each stage comprises two steps. The first stage of TSMF aims to obtain the optimal correspondences for a subset of the HR and LR data points and its steps are as follows.

1) Downsample both data sets.
2) Find the optimal correspondences for the downsampled problem by solving it to optimality using B&B.

The second stage of TSMF extends the partial set of correspondences (i.e., the optimal correspondences for the downsampled problem) found at the first stage to the original problem; its two steps are as follows.

1) Extend the partial set of correspondences of the downsampled problem to a complete set of correspondences on the full data sets (i.e., find LR correspondences for the HR points that were not in the downsampled HR data set).
2) Refine the complete set of correspondences through an iterative local search until there is no appreciable improvement.

Fig. 3 summarizes the proposed framework.

A. First Stage—Obtaining a Partial Set of Correspondences

1) Downsampling Both Data Sets: When downsampling both data sets, we have two objectives: (a) that the optimal correspondences of the downsampled problem are close to the optimal correspondences for the full data sets and (b) that the sizes of downsampled sets should be small enough so that the downsampled problem can be efficiently solved to optimality using a general MILP solver. To achieve these objectives, a downsampling algorithm needs to meet two requirements: (a) the downsampled points should be nearly evenly spread over the part surface and (b) the resulting downsampled set should contain a desired number of points.

To fulfill the two requirements, we propose a greedy downsampling approach, called *greedyDownsampling*, which

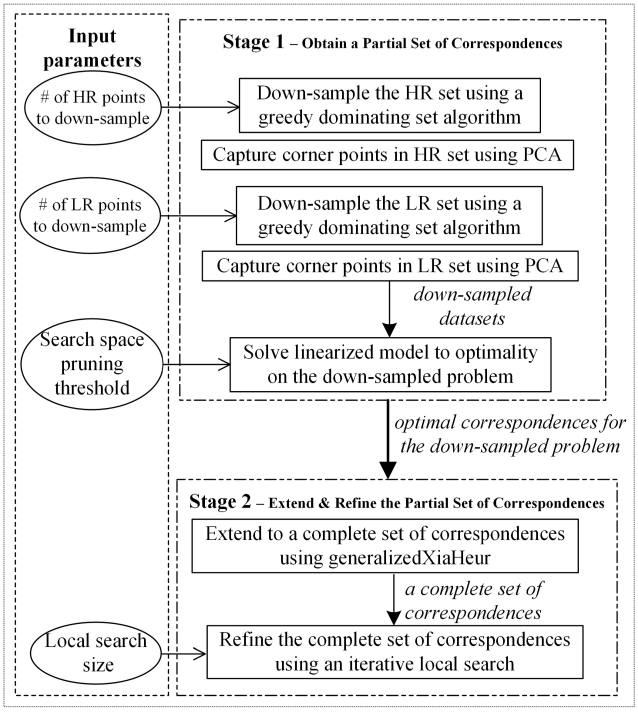


Fig. 3. Flowchart of TSMF.

combines the dominating set method and principal component analysis (PCA). Specifically, the final downsampled set comprises a set of dominating points returned by the dominating set method and the corner points detected by PCA. Note that the greedyDownsampling approach is applied to each of the two data sets in the same manner. Next, we present the details of the two components of the greedyDownsampling approach.

The idea behind the dominating set method is as follows: if each data point is either part of the sampled set, or very close to a data point in the sampled set, then the set of sampled points is guaranteed to spread evenly over the part surface. This is because the full data set is evenly spaced over the part surface. This idea can be implemented by solving the minimum dominating set problem on an undirected graph $G = (V, E)$ appropriately constructed on the full data set. A dominating set is a subset D of V such that every vertex not in D is adjacent to at least one vertex in D. The minimum dominating set problem is to find a dominating set with minimum cardinality. For our purposes, $G = (V, E)$ is constructed as follows: vertex set V comprises all data points in the full data set, and there is an edge between each data point and other points residing within a certain distance of it. We denote this distance by R_n. Building G in this way guarantees evenness of the downsampled set (i.e., the dominating set).

The minimum dominating set problem is NP hard [49]. Yet, for our purposes, using a greedy algorithm to find an approximate solution is good enough. The greedy algorithm starts with an empty dominating set D and iteratively appends to D the vertex $v \in V$ with the maximum degree and updates G by removing that newly added point and all vertices adjacent to it until G becomes empty. Since both data sets are arbitrarily indexed for identification, for both data sets, the algorithm always selects the median point as the first dominating point so that the two separate downsampling process (one for each data set) start approximately from the same physical location of the part surface, and we choose the vertex with the smallest index to break ties when there is more than one vertex having the maximum degree.

Recall that the second requirement of our downsampling approach is to obtain a desired number of points from the full data set. To achieve this, one needs to set R_n such that the greedy algorithm returns a dominating set of the desired size or very close to the desired size. Since the dominating set's cardinality increases monotonically as R_n decreases, to find the proper R_n, one can simply do a binary search over a plausible range of R_n. A safe initial range for R_n is between zero and a half of the longest between-point Euclidean distance in the respective data set. The binary search procedure starts with R_n taking the middle value of the initial range and uses it to construct graph G. With G constructed, our procedure checks if the cardinality of the returned dominating set is close enough to the desired number of downsampled points (say within 5%). If it is, the binary search stops; otherwise: (a) if the dominating set's cardinality is smaller than the desired number, the binary search continues on the lower half of the current R_n range and decreases the current R_n to the midpoint of this lower half range or (b) if the dominating set's cardinality is larger than the desired size, the binary search continues on the upper half of the current R_n range and increases R_n to the midpoint of this upper half range.

The other component in the greedyDownsampling approach is PCA, which is used to compensate the dominating set method for its tendency not to include the edge/corner points. Specifically, we use the first two principal components of the data set to obtain four corner points, two for each principal component. To get the first two corner points, we project the full data set to the first principal component and choose the two points whose projection is farthest apart. The other two points are obtained similarly using the second principal component.

2) Solve the Downsampled Problem to Optimality: After downsampling both data sets, we find the optimal solution (i.e., a partial set of correspondences for the full data sets) for the downsampled sets by solving the linearized (minMaxQIP) model to optimality. This is done using a general MILP solver, but we take advantage of the search space pruning technique described in Section III-B.

This step ensures that each point in the downsampled HR set is matched to its best LR correspondence in the downsampled LR data set. The resulting correspondences is of course only a partial set of the full correspondences, but having it creates a good basis for improvement in the second stage.

B. Second Stage—Extend the Partial Set of Correspondences and Refine the Complete Solution

Once the first stage is completed, each of the two data sets can be thought of having two subsets, the matched subsets and the unmatched subsets comprising the remaining data points, namely, $H = H^{(\text{matched})} \cup H^{(\text{unmatched})}$ and $L = L^{(\text{matched})} \cup L^{(\text{unmatched})}$, so that for each point in $H^{(\text{matched})}$, there is a

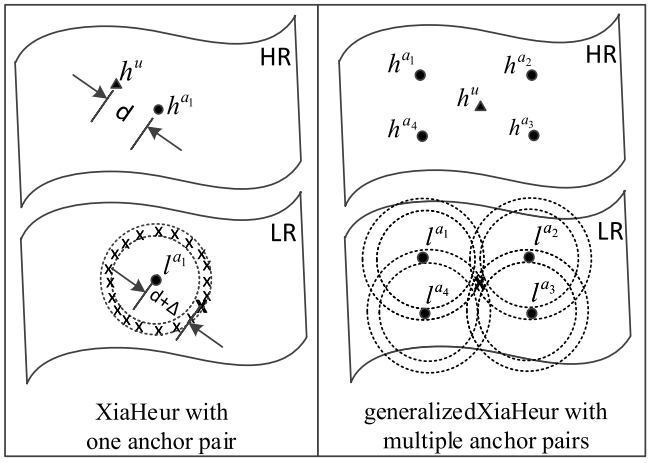


Fig. 4. Advantage of using multiple anchor pairs.

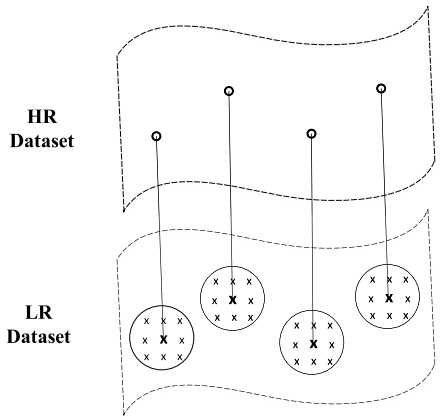


Fig. 5. Local search illustration for one iteration.

point in $L^{(\text{matched})}$ that is matched to it. Our objective in the second stage is to find a point correspondence in $L^{(\text{unmatched})}$ for every point in $H^{(\text{unmatched})}$, conditioned on the partial set of point correspondences that have already been formed between $H^{(\text{matched})}$ and $L^{(\text{matched})}$.

The existence of a set of matched pairs between the HR and LR data sets in fact provides a set of anchor pairs, to borrow the term from XiaHeur. It motivates us to follow the idea of that heuristic to match the remaining HR data points to their LR counterparts. Acknowledging that XiaHeur is not robust in its matching outcome, the two steps in this stage are devised to safeguard the solution quality.

1) Generalizing XiaHeur by Using Multiple Anchor Pairs: Through our investigation, we found that using the plain version of XiaHeur is not robust because it heavily relies on a single anchor pair. To see this, consider the example in Fig. 4 (left). In the top figure there are two HR data points, illustrated by a solid circle and a solid triangle, respectively, while in the bottom figure, there is a group of LR datapoints, one illustrated by a solid circle and the rest illustrated by crosses. The single anchor pair comprises the two solid circles, denoted by h^{a_1} and l^{a_1}, respectively. The solid triangle point, named h^u, is the unmatched HR point. With one anchor pair (h^{a_1}, l^{a_1}), there are multiple plausible LR points that could be matched to h^u. As illustrated in Fig. 4 (left), when considering a degree of measurement uncertainty up to Δ, all LR points residing within the two dashed circles could have the same merit to be matched to h^u. Some solutions could even appear on the opposite direction relative to l^{a_1}, compared to that between h^u and h^{a_1}.

In this step, we propose a generalized version of XiaHeur, called *generalizedXiaHeur*, to overcome this drawback of XiaHeur. The major change made in generalizedXiaHeur is to use the multiple anchor pairs—those formed in the first stage of our solution framework. Specifically, for an unmatched HR point h^u, generalizedXiaHeur finds its matching point in the LR data set such that the largest IPD difference between this new matching pair and each of the anchor pairs is minimized [see the illustration in Fig. 4 (right)]. Through extensive numerical studies, we believe that the generalizedXiaHeur provides a remarkably robust match outcome, even in the presence of measurement noises in the data sets.

2) Iterative Local Search: The generalizedXiaHeur, albeit its robust performance, is still a heuristic, thus leaving room for further improvement. Thus, we propose to use an iterative local search procedure to refine the complete set of correspondences obtained by generalizedXiaHeur.

As its name suggests, the iterative local search procedure comprises a sequence of local search iterations. Each local search iteration aims to find a better set of correspondences than the best set of correspondences found so far. Moreover, such searches are limited to the sets of correspondences that are close/local to the current best set of correspondences. The sequence of local search iterations terminates when there is no appreciable improvement. Note that the input for each local search is the current best set of correspondences; specifically, the input for the first local search is the set of correspondences found by generalizedXiaHeur and the input for each subsequent local search is the set of correspondences found by the preceding local search iteration. The remainder of this section explains one local search iteration.

The basic idea of the local search is illustrated in Fig. 5, where each dot in the top sinewave represents an HR point and each cross in the bottom sinewave stands for an LR point. In the input of the local search, each HR point is matched to an LR point (denoted by a bold cross). During the local search, each HR point is allowed to be rematched to any LR point in the neighborhood of that HR point's current LR correspondence (the crosses within the circle centered at the respective current LR correspondence).

Given a neighborhood size, the local search can be done by solving a modified linearized minMaxQIP model, called *local search model*. Specifically, the local search model is very similar to the linearized minMaxQIP model [(5)–(8)] except that in the local search model, each HR point can only be matched with one of the LR points in the neighborhood of that HR point's current LR correspondence. We do not give the local search model explicitly because we do not solve it directly but solve it as described in the remainder of this section.

Solving the local search model to optimality is very computationally expensive for real-life-sized problems, even when using small neighborhood sizes and even after applying

the search space pruning technique. In contrast, an MILP solver is very fast in determining the feasibility of the local search model after applying to it a search space pruning threshold, called $\bar{T}_{\mathrm{LS}}$. This large complexity difference is because one can greatly simplify the local search model if one is only interested in checking its feasibility. Specifically, to check the feasibility of the model, one can drop the objective function, previously u (equivalently, set it to a constant in the MILP solver, say zero), and consequently eliminate all the constraints related to u as they are not needed for the feasibility check. With these changes, we give below the *feasibility-check model* that one needs to solve to determine the feasibility of the local search model given a specific $\bar{T}_{\mathrm{LS}}$. In this model, NH_i denotes the set of LR points in the neighborhood of the ith HR point's current LR correspondence; constraint (11) forces each HR point, say point h_i, to be matched to exactly one of the LR points in NH_i; constraint (12) ensures that an HR point, say point h_i, is not matched to an LR point outside NH_i; constraint (13) guarantees that each LR point is only matched to at most one HR point; and constraint (14) excludes all correspondences whose maxIPDdiff is greater than $\bar{T}_{\mathrm{LS}}$. To further expedite the feasibility check, one can properly set a parameter in the MILP solver to emphasize feasibility over optimality; in CPLEX, this is to set *MIPEmphasis* to 1.

$$\begin{aligned}
&\min \quad 0\\
&\text{s.t.} \quad \sum_{s\in NH_i} x_{is} = 1, i = 1,\ldots,n_h && (11)\\
&\qquad x_{is} = 0, \quad i = 1,\ldots,n_h; s \in \{1,\ldots,n_l\} \setminus NH_i && (12)\\
&\qquad \sum_{i=1}^{n_h} x_{is} \le 1, s = 1,\ldots,n_l && (13)\\
&\qquad x_{is} + x_{jt} \le 1, \quad i,j = 1,\ldots,n_h;\ s \in NH_i\\
&\qquad\quad t \in NH_j;\ \text{if } \left|d_{ij}^H - d_{st}^L\right| > \bar{T}_{\mathrm{LS}} && (14)\\
&\qquad x_{is} \in \{0,1\}, \quad i = 1,\ldots,n_h;\ s = 1,\ldots,n_l. && (15)
\end{aligned}$$

Next, we explain how to find the local search model's optimal solution by taking advantage of the MILP solver's efficiency to solve the feasibility-check model. Note that the feasibility-check model is feasible if and only if the applied $\bar{T}_{\mathrm{LS}}$ is greater than or equal to the maxIPDdiff of the local search model's optimal solution. Therefore, as explained below, one can perform a binary search over $\bar{T}_{\mathrm{LS}}$ in order to find the optimal solution to the local search model.

In the binary search, the initial range of $\bar{T}_{\mathrm{LS}}$ is between zero and the maxIPDdiff of the best set of correspondences found so far. The binary search starts with $\bar{T}_{\mathrm{LS}}$ taking the midpoint of its initial range and solves the feasibility-check model built using that $\bar{T}_{\mathrm{LS}}$. If the model is feasible, then the binary search continues on the lower half of $\bar{T}_{\mathrm{LS}}$'s current range and decreases $\bar{T}_{\mathrm{LS}}$ to the midpoint of that lower half range; otherwise, the binary search continues on the upper half of $\bar{T}_{\mathrm{LS}}$'s current range and increases $\bar{T}_{\mathrm{LS}}$ to the midpoint of that upper half range. The binary search proceeds until the range length is within a predefined tolerance, say 0.01.

TABLE I
DESIRED NUMBER OF POINTS TO DOWNSAMPLE

Instance	100×1560	64×1560	50×800	32×800	25×400	16×400
HR points	8	8	8	8	8	8
LR points	180	180	95	95	50	50

V. COMPUTATIONAL EXPERIMENTS AND RESULTS

This section describes the experimental setup and compares the performance of TSMF with those of two widely-used point set registration methods: ICP and CPD. For this purpose, we used two 3D metrology data sets of a milled sinewave surface. All experiments were done on a Linux (Ubuntu 14.04) machine with Intel E5-1620 3.4-GHz processor and 32-GB RAM.

A. Experimental Setup

1) Data Sets (Available as an Online Supplement in the Journal Website): The HR and LR 3D metrology data sets (see Fig. 1) were obtained from a manufactured part of size $101 \times 101 \times 51$ mm^3, measured by a CCMM (*Sheffield Discovery II D-8* with a TB 20 touch probe) and an OCMM (*LDI Surveyor DS-2020* with an RPS 150 laser unit), respectively. The resolutions of the CCMM and the OCMM are roughly 5 and 50 μm, respectively. The two data sets were originally obtained by the study in [4], and each data set consists of 1560 data points that are evenly spaced over the surface. The typical number of data points collected by a CCMM over this size of product is usually an order, or orders, of magnitude fewer than that collected by an OCMM. The reason is that the study of [4] collected the same number of data points in the HR set as in the LR set because the study needed the additional HR data points for validation purposes. In fact, the largest number of data points used as the HR set in [50] is 80, and the remaining 1480 HR points were used to assess the quality of the combined prediction made by their proposed model. In this paper, we believe it is practical to increase the HR data points slightly but not substantially more. Therefore, we chose 100 as the maximum number of points in the HR set while using all the 1560 LR points.

To test the effectiveness and scalability of TSMF, we generated six instances as test cases of various sizes. Smaller sized data sets were created through thinning the two original data sets. Sizes of all six instances are listed in the top row of Table I. Each instance size is indicated by its name, which comprises two parts: the number before "×" denotes the cardinality of the HR data set, whereas the number after "×" denotes the cardinality of the LR data set.

2) TSMF Settings and Implementation: There are four key parameters used in TSMF: 1) the search space pruning threshold ($\bar{T}$); 2) the neighborhood size of the iterative local search step in the second stage; and 3) and 4) the downsampled sizes of the HR and LR data sets.

As mentioned in Section III-B, $\bar{T}$ is chosen to be 1.5 times the maximum distance between an LR point and its closest neighbor in the LR data set. The neighborhood size of the local search is set to 10; this provides a good balance between the search size and the time required to solve each local search iteration.

To set the last two algorithmic parameters—the downsampled sizes of the HR and LR data sets—we conducted extensive experiments and determined that the largest downsampled problem size that an MILP solver can solve to optimality within a couple of minutes is roughly ten HR points by 180 LR points. Meanwhile, we also observed that for each HR data set, eight data points are enough to form a good anchor set leading to an effective generalizedXiaHeur step.

For the above reasons, we chose to downsample every HR data set into eight HR points: four using PCA and four using the dominating set algorithm. In contrast, the sizes of the downsampled LR data sets were proportional to the sizes of the respective original LR data set. Specifically, we downsampled the largest LR data set into 180 points—four using PCA and 176 using the dominating set algorithm. Together with the eight downsampled HR points, this formed a combined set of 8×180 points, which is in the ballpark of the problem size that an MILP solver can solve to optimality in a desirable duration. Note that for the largest LR data set, the dominating set algorithm chose 176 points out of the 1560 points, which is roughly 11.3%. Thus, for the remaining LR data sets, we used the dominating set algorithm to select roughly the same 11.3% of points out of the respective LR data set, i.e., obtaining 91 and 46 points from the original LR data sets of size 800 and size 400, respectively. Table I summarizes the downsample set sizes for each test instance.

TSMF was implemented in C++ using Concert Technology interface for the MILP solver CPLEX (version 12.4). The PCA function we used at the downsampling step is from the Armadillo C++ linear algebra library [51].

3) CPD and ICP Implementations and Settings: Multiple ICP implementations are available online. We selected the ICP code developed by Per Bergström due to its popularity. The MATLAB code can be downloaded from http://www.mathworks.com/matlabcentral/fileexchange/12627-iterative-closest-point-method. Since no initial starting transformation is available in our experiments for ICP, we changed the parameter `init_flag` from default value 1 to 0 to reflect this fact. All other input parameters were left as default. As we want to match the entire HR data set to a subset of the LR data set, when applying ICP to our data instances, we treat the LR and HR data sets as *model* and *data*, respectively.

For CPD, we chose its most recent implementation code in MATLAB, available at https://sites.google.com/site/myronenko/research/cpd. Given the nature of our problem, we selected the rigid registration option of CPD, i.e., `opt.method` = *rigid*. Out of nine remaining input parameters of CPD, we changed three parameters to a non-default value:

1) `opt.scale` = *0* to disallow scaling in the context of a rigid body transformation;
2) `opt.corresp` = *1* to compute the correspondences at end of the registration;
3) `opt.normalize` = *0* to disallow data set normalization.

We do not normalize the data because doing so results in the best CPD performance for solving our problem.

TABLE II
PERFORMANCE COMPARISON OF DOWNSAMPLING METHODS

Instance	FPS maxIPDdiff (w)	greedyDownsampling maxIPDdiff (w)
100×1560	2.0324	1.3979
64×1560	1.9704	1.3866
50×800	2.1425	2.0197
32×800	2.1379	2.0311
25×400	1.6056	1.4966
16×400	1.2542	1.0418

B. Results and Performance Analysis

In this section, we conduct the following analyses.

- Evaluate the performance of greedyDownsampling.
- Show the effectiveness of search space pruning technique.
- Compare our TSMF with XiaHeur (with and without local search) and show the effectiveness of the local search.
- Evaluate TSMF's performance against ICP and CPD.

We want to note that as the density of each HR data set is different and the cardinality ratio of the HR data set and LR data set in each instance is also different (thus the underlying maxIPDdiff is different for each instance), throughout this section, we report all performance metrics as a multiple of the average smallest IPD in the LR set, denoted by w. Specifically, w is calculated for each instance by averaging the distances between each LR point and its closest neighbor in the LR data set. This allows us to compare the results across the different instances.

The first analysis is about the performance of the greedyDownsampling method. We compare it with a downsampling alternative, the farthest point sampling (FPS) method proposed in [52]. Since it is difficult to compare these two downsampling methods directly, what we did was the following. For each downsampling method, we first downsampled both data sets using one of the methods and then solved the downsampled problem to optimality using the search space pruning technique. The downsampling method that resulted in a better solve-to-optimality solution, i.e., a smaller maxIPDdiff, was deemed as a better option.

Table II presents the performance comparison results of the greedyDownsampling and FPS methods. For each instance, the numbers in columns 2 and 3 represent the maxIPDdiff (expressed in multiples of w) of the optimal solution of the downsampled problem obtained using FPS and greedyDownsampling, respectively. The greedyDownsampling method outperforms FPS for all instances. In addition, the execution time of greedyDownsampling and FPS are comparable across all instances and both took less than 1 s. Thus, the greedyDownsampling method suits our purposes better.

Next, we evaluate the effectiveness of the search space pruning technique when it is employed during the solve-to-optimality step in the first stage of TSMF. We first show the effectiveness of the suggested pruning threshold $\bar{T} = 1.5\ \tau$ in terms of percentage of solution time reduced and percentage of binary variables pruned after applying the suggested $\bar{T} = 1.5\ \tau$

TABLE III
EFFECTIVENESS OF THE SUGGESTED $\bar{T} = 1.5\ \tau$

Instance	100×1560	64×1560	50×800	32×800	25×400	16×400
w/o Search Space Pruning Time (s)	1616.5	1400.1	1701.9	280.6	94.4	51.6
w/ Search Space Pruning Time (s)	86.8	90.3	57.5	33.9	47.4	10.6
% of Solution Time Reduced	94.6%	93.6%	96.6%	87.9%	49.8%	79.5%
% of Binary Variables Pruned	84.2%	83.4%	81.0%	79.5%	70.0 %	67.4%

TABLE IV
SENSITIVITY STUDY RESULTS WITH DIFFERENT $\bar{T}$ VALUES

Instance		1τ	1.5τ	2τ	3τ	4τ
16×400	Time (s)	10.6	10.6	30.4	33.3	43.3
	% of Time Reduced	79.5%	79.5%	41.1%	35.5%	16.1%
	% of Vars Pruned	67.4%	67.4%	40.4%	20.0%	8.0%
25×400	Time (s)	47.7	47.4	80.2	47.7	71.5
	% of Time Reduced	49.5%	49.8%	15.0%	49.5%	24.3%
	% of Vars Pruned	70.0%	70.0%	44.0%	23.5%	11.0%
32×800	Time (s)	33.9	33.9	69.0	116.3	193.0
	% of Time Reduced	87.9%	87.9%	75.4%	58.6%	31.2%
	% of Vars Pruned	79.5%	79.5%	60.4%	43.6%	29.5%
50×800	Time (s)	58.0	57.5	239.1	179.1	477.4
	% of Time Reduced	96.6%	96.6%	86.0%	89.5%	71.9%
	% of Vars Pruned	81.0%	81.0%	63.2%	47.0%	33.0%
64×1560	Time (s)	89.6	90.3	204.4	500.0	964.9
	% of Time Reduced	93.6%	93.6%	85.4%	64.3%	31.1%
	% of Vars Pruned	83.4%	83.4%	67.6%	53.1%	40.0%
100×1560	Time (s)	86.9	86.8	336.5	593.9	1227.6
	% of Time Reduced	94.6%	94.6%	79.2%	63.3%	24.1%
	% of Vars Pruned	84.2%	84.2%	69.1%	55.1%	42.3%

to the solve-to-optimality model. Then, we further demonstrate the effectiveness of the suggested $\bar{T} = 1.5\ \tau$ by studying how the solution time and number of pruned binary variables change as a function of $\bar{T}$.

Table III summarizes the performance results of applying the suggested $\bar{T} = 1.5\tau$ to the solve-to-optimality model for each instance size. The percentage of solution time reduced and the percentage of binary variables pruned are calculated by comparing the with-pruning results to the without-pruning results. On average, the search space pruning technique eliminated 78% of the binary variables and reduced the solution time by 84%. Overall, the search space pruning technique is very effective in reducing the solution time of the solve-to-optimality model by eliminating a significant amount of binary variables from the model; moreover, its effectiveness increases as the problem sizes become larger. Note that we compute the value of τ (in the suggested $\bar{T} = 1.5\ \tau$) based on the downsampled LR data set instead of the full LR data set since the search pruning technique is applied when solving the downsampled problem to optimality.

To further demonstrate the effectiveness of the suggested $\bar{T} = 1.5\ \tau$, a sensitivity study is performed by applying to the solve-to-optimality model each of the following candidate $\bar{T}$ values: 0.5 τ, 0.75 τ, 1 τ, 1.5 τ, 2 τ, 2.5 τ, 3 τ, 3.5 τ, and 4 τ. Table IV tabulates the main sensitivity study results. For each $\bar{T}$ value, three performance metrics are reported: solution time, percentage of solution time reduced, and percentage of binary variables pruned. Note that the results for $\bar{T}$ values of 0.5 τ, 0.75 τ, 2.5 τ, and 3.5 τ are not recorded in Table IV. This is because 0.5 τ and 0.75 τ yield infeasible pruned solve-to-optimality models, and 2.5 τ (3.5 τ) leads to the same pruned solve-to-optimality model as 2 τ (3 τ). Table IV shows that for all instance sizes, 1 τ also gives the same results (and the same pruned model) as the suggested 1.5 τ. The equivalence of the models obtained using 1 τ, 2 τ, and 3 τ and 1.5 τ, 2.5 τ, and 3.5 τ, respectively, is due to our conservative definition of τ as the *maximum* distance between an LR point and its closest neighbor in the LR data set. Therefore, we decide to use 1.5 τ as in our tests it always yielded the same pruned model as 1 τ, yet we prefer to err on the safer side. In general, the solution time decreases significantly as $\bar{T}$ decreases from 4 τ to 1.5 τ. Specifically, on average, decreasing $\bar{T}$ from 4 τ to 1.5 τ saves 51% of the original solution time. An interesting observation is that the solution time does not always increase as the value of $\bar{T}$ increases for the two smallest instances. For example, for the instance 25 × 400, the solution time decreases by 8.7 s when $\bar{T}$ increases from 2 τ to 4 τ. These counterintuitive results are rare (and only occur in the smallest instances); moreover, they can be explained by the well-known variability of the solvers' solution times (most observable when the solution times are small). Despite this, 1.5 τ always requires significantly less solution time compared with 2 τ, 3 τ, and 4 τ. In sum, the effectiveness of the suggested $\bar{T}$ value of 1.5 τ in reducing the solution time is significant for all instance sizes, especially for large instances.

TABLE V
PERFORMANCE OF LOCAL SEARCH AND XiaHeur

Instance	XiaHeur		XiaHeur + local search		TSMF w/o local search		Full TSMF	
	maxIPD -diff(w)	Time (s)	maxIPD -diff(w)	Time (s)	maxIPD -diff(w)	Time (s)	maxIPD -diff(w)	Time (s)
100×1560	39.63	1.7	24.79	33.8	1.465	86.9	0.128	162.1
64×1560	35.47	1.2	23.15	10.0	1.427	89.5	0.052	120.4
50×800	33.20	0.3	19.31	5.8	2.203	58.0	0.054	73.8
32×800	25.53	0.2	15.91	2.6	2.679	34.4	0.045	43.3
25×400	0.02	0.1	0.02	0.4	1.497	47.5	0.025	51.2
16×400	8.99	0.05	4.26	1.3	1.404	10.8	0.014	12.3

The third analysis shows the local search effectiveness. Table V compares four alternative approaches: XiaHeur, XiaHeur with local search, TSMF without local search, and full TSMF (i.e., TSMF with local search).

In Table V, the results of each alternative approach are tabulated in a pair of columns, where the maxIPDdiff and solution time of applying that particular alternative approach for all six instances are tabulated in the left column and the right column, respectively. By doing a pair-wise comparison for all approaches in Table V, one can observe the following.

1) Full TSMF significantly outperforms both XiaHeur and XiaHeur with local search in five out of six instances except for the second smallest instance, where the maxIPDdiff obtained by applying XiaHeur is only slightly smaller than that obtained by full TSMF.
2) Comparing the first two pairs of columns shows that local search improved the solution quality of XiaHeur by 34%, and comparing the last two pairs of columns shows that the solution quality was improved by 97% by including local search in TSMF.
3) Even though XiaHeur and XiaHeur with local search outperform TSMF with respect to solution time, all solution times of TSMF are within a reasonable limit so that TSMF can very well serve as an offline application.

In summary, full TSMF produces significantly better solutions than both XiaHeur and XiaHeur with local search within a reasonable amount of time. In addition, the local search is very effective in improving the solution quality. More importantly, local search appears to give greater improvements when starting from a better solution.

The remainder of this section evaluates the overall performance of TSMF by comparing it with those of both ICP and CPD algorithms. Since the misalignment between the two data sets is not known and can be arbitrarily large, it is important to check the robustness of each approach to the change of the underlying rigid transformation between the two data sets. For this purposes, we created 100 variants for each of the six original instances listed in Table I. These 100 variants of each original instance are created by applying 100 uniformly distributed random rotation matrices and random translations (i.e., 100 random rigid body transformations) to the HR data set of that original instance. In this paper, we generate these 100 uniformly distributed random rotation matrices using the random rotation matrix generation approach proposed in [53]. Note that our TSMF approach is insensitive to the change of initial misalignment degree between the two data sets, but practically, it is reasonable to only allow the manufactured part rotate within the range of $-90°$ and $90°$ along axes x and y and rotate any degree along the vertical axis z (see Fig. 1). This restriction allows us to make fair comparison between TSMF and other alternative methods.

To reach an unbiased conclusion on the performance evaluation, we use three registration error metrics: maxIPDdiff, the summation of IPD differences (sumIPDdiff), and the root-mean-squared error (RMSE). The metric maxIPDdiff is used because it is the metric optimized by TSMF. The metric sumIPDdiff is reported because it is used as the objective function of many IPD-based point set registration algorithms (see [7], [10]). RMSE is also reported because it is a very popular measure of alignment error between the two data sets in point set registration problems (see [5], [24], and [25]). Moreover, RMSE is also the objective function that ICP aims to optimize. For each test instance, RMSE is calculated as follows: 1) estimating the rigid body transformation between the two data sets based on the obtained correspondences; 2) applying the estimated transformation to one data set in order to align the two data sets; and 3) calculating RMSE as the square root of the average squared Euclidean distance of all point correspondences.

TABLE VI
PERFORMANCE COMPARISON IN TERMS OF maxIPDdiff

Instance	Method	# of variants worse than TSMF	maxIPDdiff (w)			
			Min	Avg.	Std. Dev.	Max
100×1560	TSMF	-	0.13	0.13	0	0.13
	ICP	100	1.04	2.05	1.32	13.05
	CPD	100	2.26	3.21	0.51	3.59
64×1560	TSMF	-	0.052	0.052	0	0.052
	ICP	100	1.03	2.72	2.71	11.60
	CPD	100	2.28	3.26	0.41	3.60
50×800	TSMF	-	0.054	0.054	0	0.054
	ICP	100	1.03	3.05	1.67	10.24
	CPD	100	3.62	5.12	1.25	6.33
32×800	TSMF	-	0.045	0.045	0	0.045
	ICP	100	1.02	3.00	1.88	9.49
	CPD	100	4.05	4.79	0.88	10.57
25×400	TSMF	-	0.025	0.025	0	0.025
	ICP	100	0.04	1.70	0.67	4.00
	CPD	100	1.43	1.43	0.002	1.44
16×400	TSMF	-	0.014	0.014	0	0.014
	ICP	100	0.06	1.64	0.77	4.21
	CPD	100	0.03	0.40	0.47	1.04

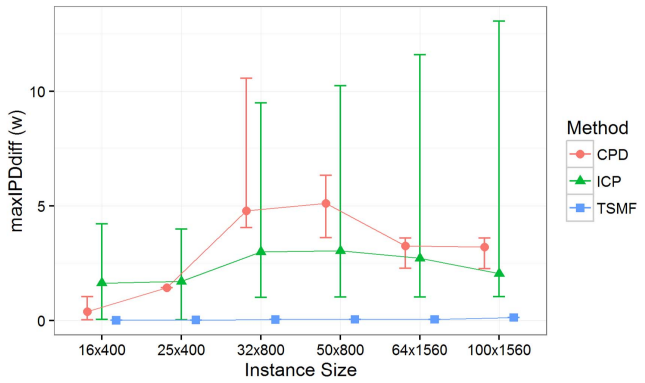


Fig. 6. Performance in terms of maxIPDdiff.

Table VI summarizes the performance of TSMF, ICP, and CPD on all 600 test cases (100 variants per instance size) in terms of maxIPDdiff. Specifically, last four columns of Table VI give the minimum maxIPDdiff, average maxIPDdiff, standard derivation of maxIPDdiff, and maximum maxIPDdiff for the 100 variants of each instance size, respectively. The third column of Table VI gives the number of variants, out of the 100, on which TSMF outperforms ICP and CPD. Fig. 6 visualizes the comparison among the three methods via an error bar plot with respect to maxIPDdif. Each error bar is plotted using the minimum value and the maximum value from Table VI.

Similar comparisons were done using RMSE and sumIPDdiff. Table VII and Fig. 7 give the results using RMSE. Fig. 8 gives the results using sumIPDdiff (we do not include a sumIPDdiff table because it does not provide additional insights). It should be noted that the 100 variants of each instance size in Tables VI and VII (also in Figs. 6–8) are 100 different instances of the same instance size each with a different misalignment degree between the two data sets. Therefore, for each instance size, the results for ICP and CPD are not 100 different runs of ICP and CPD in the

TABLE VII
PERFORMANCE COMPARISON IN TERMS OF RMSE

Instance	Method	# of variants worse than TSMF	RMSE (w) Min	Avg.	Std. Dev.	Max
100×1560	TSMF	-	0.59	0.59	0	0.59
	ICP	80	0.34	0.78	0.25	1.85
	CPD	100	0.62	0.99	0.20	1.22
64×1560	TSMF	-	0.06	0.06	0	0.06
	ICP	100	0.41	0.85	0.25	1.97
	CPD	100	0.68	1.03	0.20	1.32
50×800	TSMF	-	0.08	0.08	0	0.08
	ICP	100	0.30	0.98	0.30	1.86
	CPD	100	2.11	1.57	0.21	2.11
32×800	TSMF	-	0.07	0.07	0	0.07
	ICP	100	0.35	1.01	0.29	2.04
	CPD	100	1.04	1.41	0.26	2.70
25×400	TSMF	-	0.38	0.38	0	0.38
	ICP	98	0.29	0.65	0.19	1.20
	CPD	100	0.62	0.70	0.04	0.77
16×400	TSMF	-	0.41	0.41	0	0.41
	ICP	100	0.33	0.64	0.13	1.08
	CPD	12	0.13	0.28	0.09	0.46

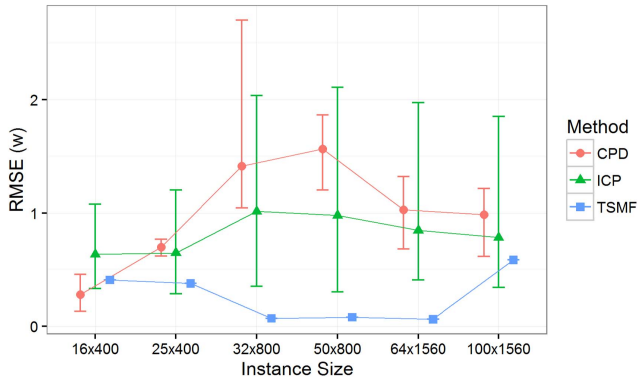


Fig. 7. Performance in terms of RMSE.

same instance. Therefore, the variance showed in the results of ICP and CPD is due to their sensitivity to the misalignment degree change between the two data sets.

As expected, since the optimization model formulation of TSMF is a fully IPD-based formulation, it is insensitive to the change of the initial misalignment between the two data sets. In contrast, both ICP and CPD are very sensitive, thus not robust to the change of the misalignment between the two data sets (overall, ICP is more sensitive than CPD). From Figs. 6 and 8, it is clear that TSMF always outperforms ICP and CPD; specifically, for every instance size, TSMF's solution has lower maxIPDdiff and sumIPDdiff than the best solution of ICP and CPD. The results with respect to RMSE are as follows.

1) TSMF outperforms ICP in all 100 test variants of four out of six instance sizes except for the largest and the second smallest instances, where our TSMF performs better than ICP for 80 out of 100 test variants and for 98 out of 100 test variants, respectively.
2) TSMF also outperforms CPD in all 100 test variants of the five largest instances except for the smallest instance where TSMF outperforms CPD in only 12 test variants.

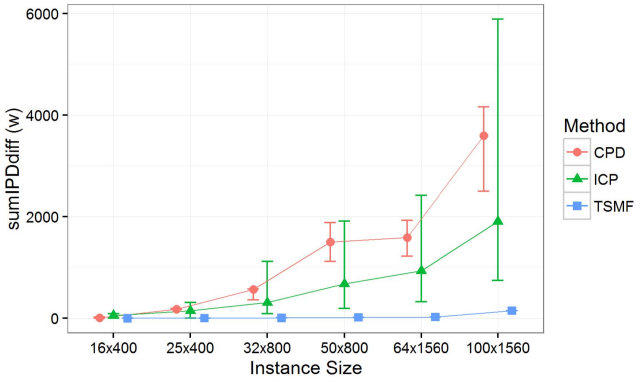


Fig. 8. Performance in terms of sumIPDdiff.

The average computation time of ICP and CPD is short and in seconds. Even though TSMF is slower than ICP and CPD, its solution time is well acceptable for it to be practically useful in an offline precision inspection setting—especially considering that obtaining the HR data set can take hours. Finally, it is clear from Figs. 6–8 that TSMF's performance scales very well with respect to every performance metric, while ICP's and CPD's performances deteriorate as the instance size increases.

VI. CONCLUSION

This paper proposed a TSMF approach to establish the correspondences between misaligned two-resolution metrology data that differ by a nearly rigid body transformation. The proposed framework has two stages: in the first stage, a coarse alignment is obtained by using downsampled data sets, while the second stage extends the partial set of correspondences on the downsampled data sets to a complete set of correspondences on the full data sets and refines the complete set of correspondences to improve that coarse alignment. This approach is a hybrid algorithm combining heuristics and exact optimization techniques. Numerical experiments showed that our approach can solve real-life-sized metrology alignment problems within a reasonable amount of time. Specifically, the execution time of TSMF is less than the time spent obtaining the LR and HR measurements from the part surface.

Our method is able to deal with two fully-overlapping metrology data with different resolutions and cardinalities in arbitrary initial positions and outperforms ICP and CPD in all 600 testing cases in terms of both maximum IPD difference and summation IPD difference metrics and almost always produces better solution than ICP and CPD with respect to RMSE metric for the five largest instance sizes. Unlike ICP and CPD, our approach is insensitive to the change of the initial misalignment between the two metrology data sets and its performance, in terms of all performance metrics, scales remarkably well as the instance size increases.

A promising but not trivial research direction is to parallelize the TSMF approach as it will allow us to solve instances with even larger sizes. As another promising future research direction, we plan to derive a Bayesian-based probabilistic model for our misaligned matching problem to account for different sources of noises and infer the optimal set of correspondences implicitly. In addition, we also plan to generalize

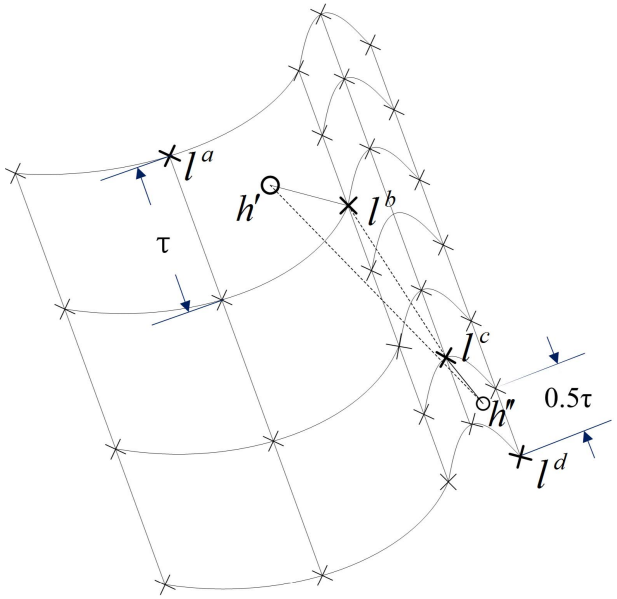


Fig. 9. Illustration of uneven case on a wiggly surface.

our framework to other practical applications in the areas of computer vision and object recognition where point set registration techniques play a key role.

APPENDIX

This Appendix illustrates that the suggested pruning threshold $\bar{T} = 1.5\tau$ (and especially our definition of τ—the maximum distance between an LR point and its closest neighbor in the LR set) is also appropriate for wiggly surfaces where the measurements may not be evenly spaced throughout the part surface. Specifically, for wiggly surfaces, the preferred measurement plans for both HR and LR data sets may still maintain the measurements' evenness locally (with higher density measurements evenly distributed over high curvature areas and lower density measurements evenly distributed over not very curvy locations). Throughout this Appendix, the terms measurements and points are used interchangeably.

Consider a hypothetical wiggly part surface shown in Fig. 9. This surface comprises a relatively flat section with a sparse set of evenly spaced measurements and a relatively high curvature section with a dense set of evenly spaced measurements. In Fig. 9, each cross represents an LR point and each circle denotes an HR point; each HR point sits almost at the center of its closest four surrounding LR points; h' sits a little bit closer to l^b than to l^a, and h'' is slightly closer to l^c than to l^d; and thus HR points h' and h'' should be matched to the LR points l^b and l^c, respectively. Note that this setup, just like the setup in Section III-B, was created in order to have the largest possible IPD difference between correct pairs of matchings. Hereafter, for brevity, we denote the line segment and its length between two points, say points A and B, by AB and $|AB|$, respectively. The triangle inequality implies that $|h'h''| < |h'l^b| + |l^b l^c| + |l^c h''|$, which in turn implies that $|h'h''| - |l^b l^c| < |h'l^b| + |l^c h''|$ (that is, the IPD difference between the two pairs of matching points is less than $|h'l^b| + |l^c h''|$). Therefore, a safe upper bound for the largest possible IPD difference is $|h'l^b| + |l^c h''|$. Now, due to the surface curvature, $|h'l^b| \approx 0.707\tau$ (where 0.707τ is half of the diagonal length of a square with side length of τ), and similarly, $|h''l^c| \approx 0.353\tau$. Finally, since $0.707\tau + 0.353\tau = 1.06\tau$, we conclude that our suggested threshold value of $\bar{T} = 1.5\tau$ is a safe upper bound on the maximum possible IPD difference.

Remark: The above discussion considers only the case where one HR point is selected from the relatively flat surface section and the other HR point is selected from the relatively high curvature surface section. Two other possible cases are: 1) both HR points are selected from the relatively flat section or 2) both HR points are selected from the high curvature section. For these two cases, one can follow the derivation discussed in Section III-B to justify 1.5τ's appropriateness.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their insightful comments and suggestions. The authors would also thank Dr. P. Bergstrom at the Luleo University of Technology and Dr. A. Myronenko at Accuray Inc., for generously sharing ICP and CPD codes, respectively. They also want to express their gratitude to Prof. P. A. Vela at the School of Electrical and Computer Engineering, Georgia Institute of Technology, and his two previous Ph.D. students Dr. H. Kingravi and Dr. I. Kolesov for their constructive suggestions and discussions on the early stage of this research.

REFERENCES

[1] I. Ainsworth, M. Ristic, and D. Brujic, "CAD-based measurement path planning for free-form shapes using contact probes," *Int. J. Adv. Manuf. Technol.*, vol. 16, no. 1, pp. 23–31, 2000.

[2] R. Minguez, A. Arias, O. Etxaniz, E. Solaberrieta, and L. Barrenetxea, "Framework for verification of positional tolerances with a 3D non-contact measurement method," *Int. J. Interact. Design Manuf.*, vol. 10, no. 2, pp. 85–93, 2014.

[3] T.-S. Shen, J. Huang, and C.-H. Menq, "Multiple-sensor integration for rapid and high-precision coordinate metrology," *IEEE/ASME Trans. Mechatronics*, vol. 5, no. 2, pp. 110–121, Jun. 2000.

[4] H. Xia, Y. Ding, and B. K. Mallick, "Bayesian hierarchical model for combining misaligned two-resolution metrology data," *IIE Trans.*, vol. 43, no. 4, pp. 242–258, 2011.

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

[6] 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.

[7] F. Pfeuffer, M. Stiglmayr, and K. Klamroth, "Discrete and geometric branch and bound algorithms for medical image registration," *Ann. Oper. Res.*, vol. 196, no. 1, pp. 737–765, 2012.

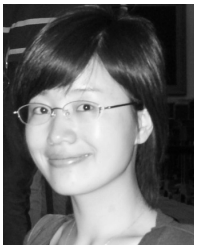
[8] K. Brunnstrom and A. J. Stoddart, "Genetic algorithms for free-form surface matching," in *Proc. 13th Int. Conf. Pattern Recognit.*, vol. 4. Vienna, Austria, Aug. 1996, pp. 689–693.

[9] J. Williams and M. Bennamoun, "A multiple view 3D registration algorithm with statistical error modeling," *IEICE Trans. Inf. Syst.*, vol. 83, no. 8, pp. 1662–1670, 2000.

[10] D. Raviv, A. Dubrovina, and R. Kimmel, "Hierarchical framework for shape correspondence," *Numer. Math., Theory, Methods Appl.*, vol. 6, no. 1, pp. 245–261, 2013.

[11] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, "Robust global registration," in *Proc. 3rd Eurograph. Symp. Geometry Process.*, Vienna, Austria, Jul. 2005, pp. 197–206.

[12] D. W. Eggert, A. Lorusso, and R. B. Fisher, "Estimating 3-D rigid body transformations: A comparison of four major algorithms," *Mach. Vis. Appl.*, vol. 9, no. 5, pp. 272–290, 1997.
[13] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 13, no. 4, pp. 376–380, Apr. 1991.
[14] X. Li and I. Guskov, "Multi-scale features for approximate alignment of point-based surfaces," in *Proc. 3rd Eurograph. Symp. Geometry Process.*, Vienna, Austria, Jul. 2005, pp. 217–226.
[15] J. Yang, H. Li, and Y. Jia, "Go-ICP: Solving 3D registration efficiently and globally optimally," in *Proc. IEEE Int. Conf. Comput. Vis.*, Sydney, Australia, Dec. 2013, pp. 1457–1464.
[16] P. J. Green and K. V. Mardia, "Bayesian alignment using hierarchical models, with applications in protein bioinformatics," *Biometrika*, vol. 93, no. 2, pp. 235–254, 2006.
[17] S. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognit.*, vol. 12, no. 4, pp. 269–275, 1980.
[18] B. Zitová and J. Flusser, "Image registration methods: A survey," *Image Vis. Comput.*, vol. 21, pp. 977–1000, Oct. 2003.
[19] J. Salvi, C. Matabosch, D. Fofi, and J. Forest, "A review of recent range image registration methods with accuracy evaluation," *Image Vis. Comput.*, vol. 25, no. 5, pp. 578–596, 2007.
[20] O. van Kaick, H. Zhang, G. Hamarneh, and D. Cohen-Or, "A survey on shape correspondence," *Comput. Graph. Forum*, vol. 30, no. 6, pp. 1681–1707, 2011.
[21] G. K. L. Tam *et al.*, "Registration of 3D point clouds and meshes: A survey from rigid to nonrigid," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 7, pp. 1199–1217, Jul. 2013.
[22] B. Bellekens, V. Spruyt, R. Berkvens, and M. Weyn, "A survey of rigid 3D pointcloud registration algorithms," in *Proc. 4th Int. Conf. Ambient Comput., Appl., Services Technol.*, Rome, Italy, Aug. 2014, pp. 8–13.
[23] X. Li and S. S. Iyengar, "On computing mapping of 3D objects: A survey," *ACM Comput. Surv.*, vol. 47, no. 2, pp. 34:1–34:45, Dec. 2014. [Online]. Available: http://doi.acm.org/10.1145/2668020
[24] S. Rusinkiewicz and M. Levoy, "Efficient variants of the ICP algorithm," in *Proc. 3rd Int. Conf. 3-D Dig. Imag. Modeling*, 2001, pp. 145–152.
[25] Y. Liu, "Improving ICP with easy implementation for free-form surface matching," *Pattern Recognit.*, vol. 37, no. 2, pp. 211–226, 2004.
[26] R. Marani, V. Renò, M. Nitti, T. D'Orazio, and E. Stella, "A modified iterative closest point algorithm for 3D point cloud registration," *Comput.-Aided Civil Infrastruct. Eng.*, vol. 31, no. 7, pp. 515–534, 2016.
[27] 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, Feb. 2003.
[28] T. N. Linh and H. Hiroshi, "Global iterative closet point using nested annealing for initialization," in *Proc. Comput. Sci.*, vol. 60, pp. 381–390, Sep. 2015.
[29] H. Pottmann, S. Leopoldseder, and M. Hofer, "Registration without ICP," *Comput. Vis. Image Understand.*, vol. 95, no. 1, pp. 54–71, 2004.
[30] N. J. Mitra, N. Gelfand, H. Pottmann, and L. Guibas, "Registration of point cloud data from a geometric optimization perspective," in *Proc. 2nd Eurograph./ACM SIGGRAPH Symp. Geometry Process.*, Nice, France, Jul. 2004, pp. 22–31.
[31] G. McNeill and S. Vijayakumar, "A probabilistic approach to robust shape matching," in *Proc. IEEE Int. Conf. Image Process.*, Atlanta, GA, USA, Oct. 2006, pp. 937–940.
[32] W. M. Wells, III, "Statistical approaches to feature-based object recognition," *Int. J. Comput. Vis.*, vol. 21, nos. 1–2, pp. 63–98, 1997.
[33] B. Luo and E. R. Hancock, "A unified framework for alignment and correspondence," *Comput. Vis. Image Understand.*, vol. 92, no. 1, pp. 26–55, 2003.
[34] B. Jian and B. C. Vemuri, "A robust algorithm for point set registration using mixture of Gaussians," in *Proc. 10th IEEE Int. Conf. Comput. Vis.*, vol. 2, Beijing, China, Oct. 2005, pp. 1246–1251.
[35] M. Lu, J. Zhao, Y. Guo, and Y. Ma, "Accelerated coherent point drift for automatic three-dimensional point cloud registration," *IEEE Geosci. Remote Sens. Lett.*, vol. 13, no. 2, pp. 162–166, Feb. 2016.
[36] B. Eckart, K. Kim, A. Troccoli, A. Kelly, and J. Kautz, "MLMD: Maximum likelihood mixture decoupling for fast and accurate point cloud registration," in *Proc. Int. Conf. 3D Vis. (3DV)*, Lyon, France, Oct. 2015, pp. 241–249.
[37] A. Rangarajan *et al.*, "A robust point-matching algorithm for autoradiograph alignment," *Med. Image Anal.*, vol. 1, no. 4, pp. 379–398, 1997.
[38] S. Gold, A. Rangarajan, C.-P. Lu, S. Pappu, and E. Mjolsness, "New algorithms for 2D and 3D point matching: Pose estimation and correspondence," *Pattern Recognit.*, vol. 31, no. 8, pp. 1019–1031, 1998.
[39] H. Chui and A. Rangarajan, "A feature registration framework using mixture models," in *Proc. IEEE Workshop Math. Methods Biomed. Image Anal.*, Hilton Head Island, SC, USA, Jun. 2000, pp. 190–197.
[40] F. Meskine, N. Taleb, M. C. El-Mezouar, K. Kpalma, and A. Almhdie, "A rigid point set registration of remote sensing images based on genetic algorithms and Hausdorff distance," in *Proc. World Acad. Sci., Eng. Technol.*, vol. 7, 2013, pp. 1095–1100.
[41] J. Luck, C. Little, and W. Hoff, "Registration of range data using a hybrid simulated annealing and iterative closest point algorithm," in *Proc. IEEE Int. Conf. Robot. Autom.*, vol. 4, San Francisco, CA, USA, Apr. 2000, pp. 3739–3744.
[42] H. Rashid and A. K. Turuk, "Dead reckoning localisation technique for mobile wireless sensor networks," *IET Wireless Sensor Syst.*, vol. 5, no. 2, pp. 87–96, Apr. 2015.
[43] H. Li and R. Hartley, "The 3D-3D registration problem revisited," in *Proc. IEEE 11th Int. Conf. Comput. Vis. (ICCV)*, Oct. 2007, pp. 1–8.
[44] T. M. Breuel, "Implementation techniques for geometric branch-and-bound matching methods," *Comput. Vis. Image Understand.*, vol. 90, no. 3, pp. 258–294, 2003.
[45] M. Brown, D. Windridge, and J.-Y. Guillemaut, "Globally optimal 2D-3D registration from points or lines without correspondences," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Dec. 2015, pp. 2111–2119.
[46] R. E. Burkard, "Quadratic assignment problems," *Eur. J. Oper. Res.*, vol. 15, no. 3, pp. 283–289, 1984.
[47] S. Sahni and T. Gonzalez, "P-complete approximation problems," *J. ACM*, vol. 23, no. 3, pp. 555–565, 1976.
[48] R. Burkard, S. Karisch, and F. Rendl, "QAPLIB—A quadratic assignment problem library," 2002. [Online]. Available: http://anjos.mgi.polymtl.ca/qaplib/
[49] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco, CA, USA: Freeman, 1979.
[50] H. Xia, "Bayesian hierarchical model for combining two-resolution metrology data," Ph.D. dissertation, Dept. Ind. Syst. Eng., Texas A&M Univ., College Station, TX, USA, Dec. 2008.
[51] C. Sanderson and R. Curtin, "Armadillo: A template-based C++ library for linear algebra," *J. Open Source Softw.*, vol. 1, p. 26, 2016.
[52] C. Wang, M. M. Bronstein, and N. Paragios, "Discrete minimum distortion correspondence problems for non-rigid shape matching," INRIA, France, Res. Rep. RR-7333, 2010.
[53] J. Arvo, *Graphics Gems III*. San Diego, CA, USA: Academic, 1992, ch. 3, pp. 117–120.



Yaping Wang received the B.S. degree in mechanical engineering and automation from the Harbin Institute of Technology, Harbin, China, in 2003, the M.S. degree in computer science from Northwestern Polytechnical University, Xi'an, China, in 2007, and the M.E. degree in industrial engineering from Texas A&M University, College Station, TX, USA, in 2010, where she is currently pursuing the Ph.D. degree in industrial engineering.

Her current research interests include design and analysis of optimization algorithms for solving large-scale optimization computational problems (e.g., rigid point set registration problem and power system corrective topology control), and big data analytical modeling for complex problem.

Erick Moreno-Centeno received the B.S. degree in industrial physics engineering from the Instituto Tecnológico y de Estudios Superiores de Monterrey, Monterrey, Mexico, in 2002, and the M.S. and Ph.D. degrees in industrial engineering and operations research, in 2006 and 2010, respectively, and the M.S. degree in computer science in 2010, all from the University of California at Berkeley, Berkeley, CA, USA.

He is currently an Assistant Professor with the Department of Industrial and Systems Engineering, Texas A&M University, College Station, TX, USA. His current research interests include mathematical programming, particularly in the design and analysis of combinatorial optimization models and algorithms to solve complex problems.

Yu Ding (M'06–SM'11) received the B.S. degree in precision engineering from the University of Science and Technology of China, Hefei, China, in 1993, the M.S. degree in precision instruments from Tsinghua University, Beijing, China, in 1996, the M.S. degree in mechanical engineering from Pennsylvania State University, State College, PA, USA, in 1998, and the Ph.D. degree in mechanical engineering from the University of Michigan, Ann Arbor, MI, USA, in 2001.

He is currently the Mike and Sugar Barnes Professor with the Industrial and Systems Engineering Department, Texas A&M University, College Station, TX, USA. His current research interests include quality and reliability engineering and systems informatics.

Dr. Ding is a Fellow of the Institute of Industrial Engineers (IIE) and a member of the Institute for Operations Research and the Management Sciences (INFORMS) and the American Society of Mechanical Engineers (ASME). He serves as the Department Editor of *IIE Transactions*.