V4PCS: VOLUMETRIC 4PCS ALGORITHM FOR GLOBAL REGISTRATION

Jida Huang[1], **Tsz-Ho Kwok**[2], **Chi Zhou**[1]

[1]Department of Industrial and Systems Engineering
University at Buffalo, The State University of New York, Buffalo, NY 14260

[2]Department of Mechanical, Industrial and Aerospace Engineering,
Concordia University, Montreal, QC, Canada H3G 1M8

Email: jidahuan@buffalo.edu, tszho.kwok@concordia.ca, chizhou@buffalo.edu

ABSTRACT

With the advances in three-dimensional (3D) scanning and sensing technologies, massive human-related data are now available and create many applications in data-driven design. Similarity identification is one of basic problems in data-driven design and can facilitate many engineering applications and product paradigm such as quality control and mass customization. Therefore, reusing information can create unprecedented opportunities in advancing the theory, method, and practice of product design. To enable information reuse, different models have to be aligned so that their similarity can be identified. This alignment is commonly known as the global registration that finds an optimal rigid transformation to align two 3D shapes (scene and model) without any assumptions on their initial positions. The Super 4-Points Congruent Sets (S4PCS) is a popular algorithm used for this shape registration. While S4PCS performs the registration using a set of 4 coplanar points, we find that incorporating the volumetric information of the models can improve the robustness and the efficiency of the algorithm, which are particularly important for mass customization. In this paper, we propose a novel algorithm, Volumetric 4PCS (V4PCS), to extend the 4 coplanar points to non-coplanar ones for global registration, and theoretically demonstrate the computational complexity is significantly reduced. Experimental tests are conducted on a number of models such as tooth aligner and hearing aid to compare with S4PCS. The experimental results show that the proposed V4PCS can achieve a maximum of 20 times speedup and can successfully compute the valid transformation with very limited number of sample points. An application of the proposed method in mass customization is also investigated.

1. INTRODUCTION

With the arrival of cyber-physical systems era, similarity identification is of great importance in data-driven design since reusing data can not only accelerate system response speed but also enable the utilization of template design in many engineering applications. The concept of information reuse [1] has been demonstrated as a promising technique to take advantage of the similarities and create unprecedented opportunities in advancing the theory, methods, and practice of data-driven product design. The idea of similarity identification is that the process of the similar or even the same features among different objects only need to be done once, and the results can be applied to other instances. To enable similarity identification and further information reuse, the models should be aligned and then their shape similarities are identified, which is naturally a shape registration problem. Besides, this pre-alignment step is also common in many manufacturing applications, e.g., quality control – once a product is manufactured, it needs to be aligned and compared with the gauge model to identify whether it is qualified.

Registration is a process to align different three-dimensional (3D) shapes either of the same object or of two different but similar objects. It is a fundamental task in many applications of manufacturing, computer vision, pattern recognition, computer graphics, medical imaging, etc. It is also referred to as *scene-to-model* registration that finds an optimal rigid transformation $\mathbf{T} \in \mathcal{R}^3$ to align one shape (model $\mathcal{M}$) to another (scene $\mathcal{S}$). "Model" and "scene" are the aliases for two 3D shapes, where "scene" stands for the static one and "model" stands for the moving one, i.e., $\mathbf{T}(\mathcal{M}) \rightarrow \mathcal{S}$. The rigid transformation $\mathbf{T}$ includes rotation $\mathbf{R}$ and translation $\mathbf{t}$ that is defined as following when it acts on any vector or point $\mathbf{p}$:

$$\mathbf{T}(\mathbf{p}) = \mathbf{Rp} + \mathbf{t}$$

If there are four pairs of vertices $\{\mathbf{p_i} \in \mathcal{M}, \mathbf{q_i} \in \mathcal{S}\}, i = 1 \ldots 4$, a linear transformation between the shapes $\mathcal{M}$ and $\mathcal{S}$ can be defined by a linear transformation $\mathbf{L}$ as

$$\mathbf{Lp_i} + \mathbf{t} = \mathbf{q_i}, \qquad \mathrm{i} = 1 \ldots 4. \tag{1}$$

The translation $\mathbf{t}$ can be eliminated by subtracting the last equation from the others, and thus we have $\mathbf{LP} = \mathbf{Q}$ with

$$\mathbf{P} = [\mathbf{p_1} - \mathbf{p_4} \quad \mathbf{p_2} - \mathbf{p_4} \quad \mathbf{p_3} - \mathbf{p_4}] \in \mathcal{M}$$
$$\mathbf{Q} = [\mathbf{q_1} - \mathbf{q_4} \quad \mathbf{q_2} - \mathbf{q_4} \quad \mathbf{q_3} - \mathbf{q_4}] \in \mathcal{S}$$

Therefore, the matrix $\mathbf{L}$ can be obtained by $\mathbf{L} = \mathbf{QP^{-1}}$, which included both the scaling and the rotation. In order to extract the pure rotation $\mathbf{R}$ from $\mathbf{L}$, *singular value decomposition* (SVD) can be applied: $\mathrm{SVD}(\mathbf{L}) = \mathbf{U\Sigma V^T}$. Since $\mathbf{L}$ is a square matrix with positive determinant, $\mathbf{U}$ and $\mathbf{V}$ are the rotational matrices, and $\mathbf{\Sigma}$ is a scaling matrix. Then, the pure rotation can be derived from $\mathbf{L}$ as $\mathbf{R} = \mathbf{UV^T}$. The translation vector $\mathbf{t}$ can then be computed by subtracting the centers of $\{\mathbf{q_i}\}$ and transformed $\{\mathbf{p_i}\}$.

If a rough initial position and orientation of the model relative to the scene is given, the most popular method for fine registration is the Iterative Closest Point (ICP) algorithm [2]. Otherwise, a global registration is needed to find a rigid transformation to roughly align two 3D shapes without any assumptions on their initial positions. The 4-Points Congruent Sets (4PCS) algorithm [3] is an established approach to global registration. 4PCS performs the registration using a set of 4 coplanar points – called a *base* – forming a two-dimensional (2D) plane in one shape and find the congruent bases in another by the RANdom SAmple Consensus (RANSAC) method. It is an iterative method that repeats a constant number of times and find the best transformation, which has a runtime complexity of $O(n^2)$, where n is the size of input point set. 4PCS is further developed to a Super 4PCS (S4PCS) algorithm [4] by using intelligent indexing scheme to reduce the complexity of the original 4PCS algorithm to $O(n)$. Our study finds that using coplanar points may have ambiguities when the shapes are close to planar, and thus results in excess number of congruent bases to find the real optimal solution. In this paper, we incorporate the volumetric information of the shapes and propose a Volumetric 4PCS (V4PCS) algorithm to extend the 4 coplanar points to non-coplanar ones – specifically a tetrahedron is used here. The computational complexity is theoretically reduced, and experimental results show that the proposed method can significantly improve the robustness and the efficiency compared to S4PCS. The contributions of this paper are summarized:

1. The volumetric information of 3D shapes is introduced to the base construction, which can filter the ambiguities and the unnecessary congruent bases, and thus the algorithm is more robust and resistant to extreme cases.
2. The smart indexing technique in S4PCS, which reduces the quadratic time complexity in 4PCS to linear, is further utilized for the tetrahedron construction, so that there is no increase in time complexity in our base construction, but a maximum of 20x speedup overall.
3. With the robustness of the base construction, the concept of information reuse is extended to the registration by reusing the base to efficiently align multiple models together. It is demonstrated by an application of mass customization.

The rest of this paper is organized as follows. Section 2 gives a brief review of literature. Section 3 discusses the 4PCS and S4PCS algorithms, and our proposed V4PCS algorithm is presented in Section 4. To demonstrate the advantages of our framework, several human-centered and manufacturing models and an application in mass customization are studied in Section 5. Finally, Section 6 concludes the paper.

2. RELATED WORKS

Surface registration is a fundamental problem in many applications [5]. Many research works have been explored on multifarious aspects, and readers are referred to a survey paper [6]. When the initial inputs of registration are close to each other, fine registration is done to refine the alignment. Iterative Closest Point (ICP) and its variants [6,11] are widely used and become 'gold standard' for fine registration. Recently, Bouaziz et al. [8] proposed a sparse ICP formulation to robustly handle data with large amounts of outlier, and Fitzgibbon [9] has introduced a nonlinear least-square optimization to develop a robust error function that increases the radius of convergence. When the inputs have arbitrary positions and orientations, global registration methods are used to find an optimal transformation to roughly align the inputs. One approach to global registration is based on the branch-and-bound framework which has theoretical optimality guarantee [10, 11, 12]. However, the systematic search can be extremely time-consuming when the input size is large. Since two sets of 3 non-degenerated points can uniquely determine a rigid transformation including translation and rotation, most global alignment methods use RANSAC method to find a set of candidate correspondences from two shapes and choose the best alignment[13, 14, 15]. The complexity of RANSAC method regularly degrades to its worst case $O(n^3)$ complexity in the number n of data samples. As mentioned earlier, 4PCS [3] and S4PCS [4] intelligently ameliorate the searching process to have the time complexity of $O(n^2)$ and $O(n)$.

There are other research results that use discriminative descriptor and shape distribution to pick sample points for registration and object matching. The shape distribution either sample from a shape function measuring global geometric properties [16] or base on the histogram of point cloud [17]. Discriminative descriptor mainly considers the local information of the shape by estimating the features of points and their neighbors. Plentiful of shape descriptors methods mainly include two categories [18]: spatial distribution histogram and geometric attribute histogram based descriptors. Typical spatial distribution methods include Heat Kernel Signatures [19], Unique Shape Context [20], Rotational Projection Statistics [21]. The well-known geometric attribute methods are Point Feature Histogram [22], Fast Point Feature Histogram [23] and Signature of Histogram of Orientations [24]. Robust shape descriptors are easy to compare but suffer from expensive computational cost especially for large data size and under the situation of noise and outliers. Other methods mainly include voting methods and correspondence based methods [25]. The well-known representatives for voting methods are the generalized Hough transform [26], geometric hashing [27] and tensor matching algorithm [28].

3. TERMINOLOGY AND 4PCS

This section briefly describes the terminologies in surface registration and the basic idea of the 4PCS and S4PCS [7,8].

The goal of global registration is to compute a rough alignment as the initial position for the fine registration, so that the registration will not be trapped into local optimum.

From Eq.(1), it can be seen that the success of finding the optimal transformation $\mathbf{T}$ highly depends on the selection of the four vertices $\{\mathbf{p_i} \in \mathcal{M}, \mathbf{q_i} \in \mathcal{S}\}$ on both shapes. For example, a hand model is used in Fig. 1 for demonstration, where the left one is the scene $\mathcal{S}$ and the right one is the model $\mathcal{M}$. Assume there are 4 points $\{\mathbf{q_i}\}$ selected on $\mathcal{S}$ at its 4th finger, 2nd finger, and center of palm. If the corresponding vertices $\{\mathbf{p_i}\}$ on $\mathcal{M}$ are also selected correspondingly on the fingers and palm (as shown in top row of Fig. 1), the computed $\mathbf{T}$ is correct and good enough for a rough alignment. However, if $\{\mathbf{p_i}\}$ are selected on the 2nd finger, 1st finger, palm and 5th finger (Fig. 1 bottom) for the model $\mathcal{M}$, the computed $\mathbf{T}$ is incorrect, and fine registration cannot be applied on this result. These kinds of vertex sets having wrong correspondences will lead to wrong alignment and shouldn't be used, but the challenge is how to find the good set of vertices $\{\mathbf{p_i}, \mathbf{q_i}\}$ efficiently.

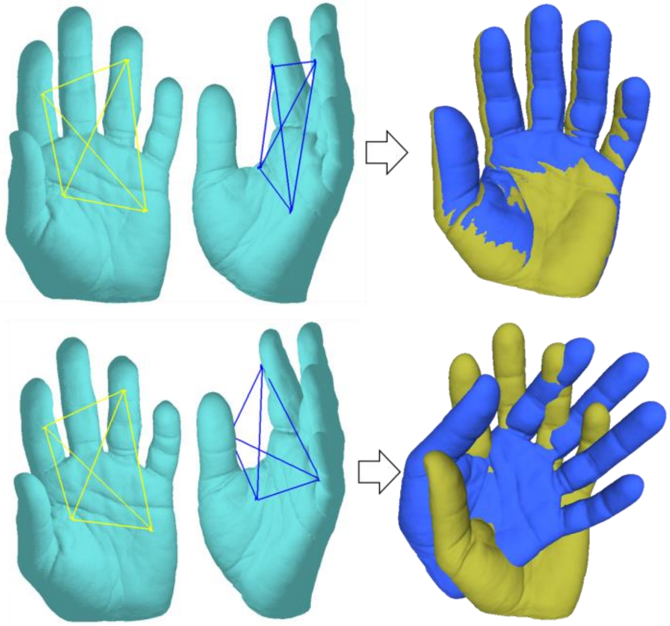
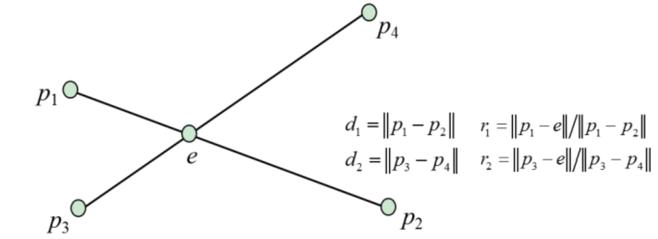


Fig. 1: *The idea of picking 4 points to align two models.*

Let the inputs $\mathcal{S}$ and $\mathcal{M}$ be sampled with m and n points, a brute-force method to test all the combinations and find the best one will result in a time complexity of $O(n^4)$ for $m \leq n$. Based on the key fact that certain ratios defined on a planar congruent set remain invariant under rigid motion, 4PCS [3] significantly reduces the number of trials by only extracting the coplanar 4-points sets and matching with the another set of coplanar 4-points that are approximately congruent. This extraction procedure runs roughly in $O(n^2)$ time and the algorithm of 4PCS is summarized by the following steps:

1. **Pair Generation** (PG): Select 4 coplanar points $\{\mathbf{q_i}\}$ from $\mathcal{S}$ – also named as a base, and then calculate the point distances d_1, d_2 and define the affine invariant ratios r_1, r_2 using these 4-coplanar points as shown in Fig. 2. After that, find all point pairs in $\mathcal{M}$ with distance d_1 and d_2 respectively. This step takes $O(m + n^2)$ time.
2. **Congruent Set Extraction** (CSE): For k reported pairs in $\mathcal{M}$ from the PG step, extract all conjugate pairs which could form a 4 point set congruent to $\{\mathbf{q_i}\} \in \mathcal{S}$, i.e., these four points have an intersection point and same ratios with r_1, r_2, in the extracted pairs. A range tree can be used to improve this query time, and this step takes $O(k \log k + c)$ time, where k is the number of reported pairs and c is the number of congruent sets extracted.
3. **Congruent Set Verification** (CSV): With the extracted c congruent sets $U = \{\{\mathbf{p_i}\}_1, \{\mathbf{p_i}\}_2, \ldots, \{\mathbf{p_i}\}_c\} \in \mathcal{M}$, a rigid transform $\mathbf{T}$ can be computed for each set by Eq.**Error! Reference source not found.**). Then, all the n sample points in $\mathcal{M}$ can be transformed by $\mathbf{T}$ and then the quality of $\mathbf{T}$ is verified by computing how many points of $\mathbf{T}(\mathbf{p})$ are within ϵ-distance to those sample points in $\mathcal{S}$. Finally, the best $\mathbf{T}$ with the most points matched is returned. Assuming the complexity of nearest neighbor search is $O(\log m)$, this step takes $O(c\, n \log m)$ time.



p_4
p_1
e
p_3
p_2
$d_1 = \|p_1 - p_2\|$
$r_1 = \|p_1 - e\| / \|p_1 - p_2\|$
$d_2 = \|p_3 - p_4\|$
$r_2 = \|p_3 - e\| / \|p_3 - p_4\|$

Fig. 2: *Based on a 4-coplanar point base, two distance and ratios can be computed*

Remark 1: In the Congruent Set Extraction (CSE) step, judging of whether a set is congruent is cleverly converted to a matching by d_1, d_2, r_1, and r_2, which takes only $O(1)$ time.

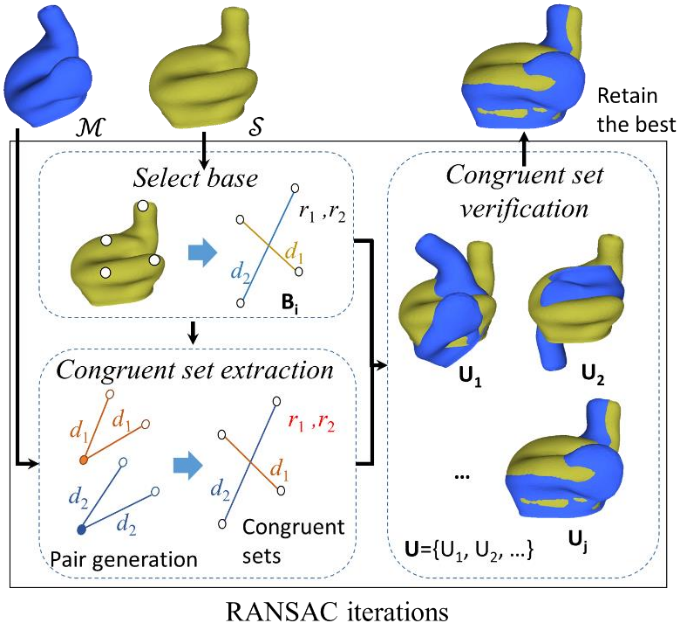


Fig. 3: *The flowchart of 4PCS*

The flowchart of 4PCS is shown as in Fig. 3, the procedure of 4PCS randomly picks a base from $\mathcal{S}$, computes transformations between the selected base and all possible congruent bases from $\mathcal{M}$, and then verifies the resulting registration transformations. To achieve a certain probability of success, this procedure is repeated for L different choices of bases from $\mathcal{S}$ as in RANSAC method. Each iteration runs in $O(n^2 + k \log k + c)$ time, where n is the number of points in $\mathcal{M}$, k is the number of reported pairs and c is the number of congruent sets extracted. The uncertainty of success is from the selection of base on $\mathcal{S}$. If the selected base is generic, e.g., 4 points on a spherical feature, then the alignment can hardly be correct even a number of congruent sets are found. Therefore, we have the following observation.

Observation 1: If the selected base on $\mathcal{S}$ is rare or even unique, there are few similar bases could be found and the number of iteration can be reduced.

The most expensive step in 4PCS is the Pair Generation (PG) step, which requires finding all point pairs in a points set with a given distance takes $O(n^2)$ time. In order to improve the n^2 term, the Super 4PCS (S4PCS) [4] algorithm is proposed to use a new data structure to ameliorate this bottleneck. In short, they organized the points on $\mathcal{M}$ in a 3D grid, and implemented an efficient hyper-sphere rasterization to extract the pairs by intersecting two spheres of radius d_1 and d_2 on this 3D grid. This development makes the S4PCS lower the quadratic complexity from $O(n^2)$ to linear complexity $O(n)$. S4PCS also uses smart indexing techniques to produce the same congruent set as 4PCS but in linear time, i.e., $O(k \log k)$ is reduced to $O(k)$. Therefore, the algorithm of S4PCS runs in $O(n + k + c)$ time.

Remark 2: The most attractive capability of S4PCS is that given a value d, all the point pairs with length $d \pm \varepsilon$ can be extracted efficiently in $O(n)$ time, where ε is a given margin.

4. VOLUMETRIC 4PCS (V4PCS)

4.1 Overview

In S4PCS, the time complexity is reduced and it is linear to the number of points n, the number of reported pairs k, and the number of congruent sets c as $O(n + k + c)$. Hence, the originally insignificant k and c terms become important now. As the congruent sets are the subset of all the reported pairs, it is obvious that $k \geq c$, so we use c alone in the following discussion. The value of c could be varied by the shape of inputs, the width of the selected base $\{\mathbf{q}_\mathbf{i}\}$ in $\mathcal{S}$, and even the value of n. Our study finds that c can be much larger than n in some cases, which can be seen from Fig. 4. Two models (teeth and hearing aid) are tested to understand the correlation between the size of base and the number of congruent set. Both models are sampled with 800 points, i.e., $n = 800$, and the width of a base is using different percentages of the model diameter. As the number of congruent set could be different in each iteration, the average value of all iteration is taken after the alignment is done. The y-axis is the size of congruent set c and it is plotted in log scale. It can be seen that the value of c can be as high as 1×10^6 while n is just 800, which means $c > n^2$ or $c > 125000n$.

Furthermore, for each congruent base, there is a transformation whose quality must be measured by solving the nearest-neighbor search problem. If c is large, the term $O(c\, n \log m)$ in the CSV step becomes dominant. Therefore, there is a critical need to minimize the number of congruent set c without increasing the extraction time. There are some heuristics being reported, such as choosing a wide base in 4PCS and using the angle between the two intersecting lines to filter non-congruent 4-point sets in S4PCS. While these methods are useful, there are still excess number of congruent sets need to be verified. We have the following observation.

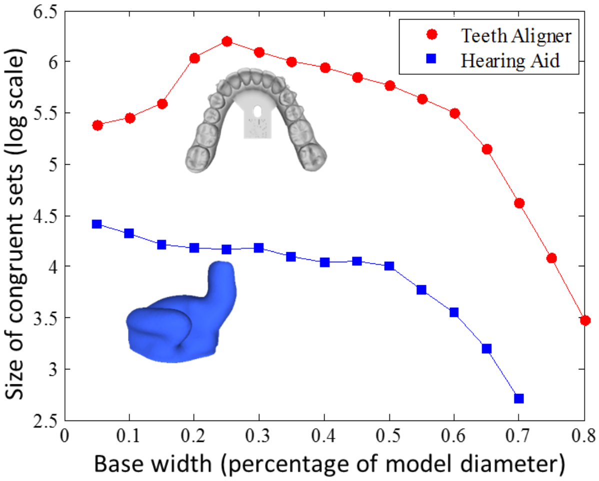


Fig. 4: *Relation between base width and size of congruent sets*

Observation 2: The input is a 3D shape, but the coplanar points form a 2D plane, which can be viewed as a degenerated base describing the 3D shape. A 2D plane is not unique, i.e., there could be numerous number of 2D planes found in a 3D shape, and it could have ambiguities in representing a 3D shape.

One example of the ambiguity is shown in Fig. 7 that when two teeth models are aligning by a 2D plane using S4PCS, a flipped result is obtained if the sample size is not sufficiently large. Both Observations 1 and 2 point to the development of new base format that can better represent the input 3D shape, so we make the following two hypotheses:

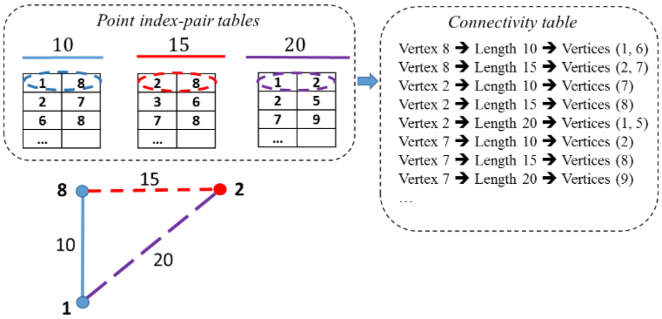
Hypothesis 1: Incorporating volumetric information to form a 3D base is a richer representation for a 3D shape, and it can eliminate the ambiguities in the 2D base, so that the algorithm will be more robust and resistant to extreme cases.

Hypothesis 2: Maximizing the volume of the base can filter many false positive cases in the CSE step, and the number of congruent sets c will be largely reduced, so that the algorithm will be speeded up and more efficient.

In this paper, we utilize the new data structure developed in S4PCS and develop a Volumetric 4PCS (V4PCS) which theoretically decrease the CSE time. In the rest of this section, we will discuss our methodology and implementation to test our hypotheses. Since the CSV step is the same, we only discuss the PG and CSE steps here.

4.2 Pair Generation (PG)

As stated in Remark 2 (at the end of Section 3), S4PCS has developed a new data structure that extracts all the pairs with required distances in a linear time. In order to utilize this capability, we propose to construct a base that only requires the distance information, so that we can have more flexibility to simplify the algorithm. Specifically, a tetrahedron is used in this paper. In geometry, we know that two tetrahedrons are congruent if all of their corresponding laterals have the same length, so the judging can be purely based on the distance information. Besides, the relationship between the lengths of different vertices can be stored by a connectivity table at the time when the pairs are extracted. After that, we only need to look up from the tables and check if there are laterals with the same length to form a congruent tetrahedron.



Point index-pair tables

10	
1	8
2	7
6	8
...	

15	
2	8
3	6
7	8
...	

20	
1	2
2	5
7	9
...	

Connectivity table

Vertex 8 ➔ Length 10 ➔ Vertices (1, 6)
Vertex 8 ➔ Length 15 ➔ Vertices (2, 7)
Vertex 2 ➔ Length 10 ➔ Vertices (7)
Vertex 2 ➔ Length 15 ➔ Vertices (8)
Vertex 2 ➔ Length 20 ➔ Vertices (1, 5)
Vertex 7 ➔ Length 10 ➔ Vertices (2)
Vertex 7 ➔ Length 15 ➔ Vertices (8)
Vertex 7 ➔ Length 20 ➔ Vertices (9)
...

Fig. 5: *Congruent set extraction process illustrated by a triangle*

This idea is illustrated by a triangle case shown in Fig. 5. First, three distances corresponding to the laterals of a triangular base selected from $\mathcal{S}$ are calculated, e.g., 10, 15, and 20 respectively. Then, by using the pair extraction algorithm in S4PCS, we can extract all the pairs with the lengths 10, 15, and 20 from $\mathcal{M}$ in a linear time. After that, we can build three tables of point index-pair based on the lengths to record all the extracted point pairs with the corresponding length. These tables can also be stored with connectivity rooted at each vertex, e.g., Vertex 8 ➔ Length 10 ➔ Vertices (1, 6 …). The connectivity is built during the pair extraction process, and it does not take extra time as compared to simply storing all the extracted pairs in an unordered array. It is similar for tetrahedron, except there are six tables instead of three.

In terms of selecting the tetrahedral base from $\mathcal{S}$ at the first step, similar to finding a wide base in S4PCS by running a constant number of iterations to choose the one with the largest area, we also run a constant number of iteration to choose a tetrahedron in $\mathcal{S}$ with the largest volume.

4.3 Congruent Set Extraction (CSE)

After a tetrahedral base is selected from $\mathcal{S}$, the next step is to extract all the congruent tetrahedrons from $\mathcal{M}$. Again, the idea is illustrated by the triangle case in Fig. 5. With the connectivity tables generated from the PG step, extracting a congruent triangle becomes a simple lookup problem. For example, to find a congruent triangle in $\mathcal{M}$, we can start with the index-pair table with the length of 10, e.g., the pair 1-8, and find a pair with length 15 connecting to it. Both of the vertices 1 and 8 are retrieved and checked if there is a pair in the list having a length 15, e.g., Vertex 8 ➔ Length 15 ➔ Vertices (2, 7). After that, it can very easily compute if the pairs 1-2 and 1-7 are 20 in length. Courtesy to the beauty of 4PCS stated in Remark 1, we construct the bases that contain only the distance information, and we only need to query the distance pair index table to extract the congruent pairs directly. All these operations can be done in $O(1)$ time. There is no need to compute the ratios and angle to judge whether the sets are congruent. It is also worth to remark that the pairs extracted have a tolerance ε in terms of its length, i.e., $d \pm \varepsilon$ (see Remark 2), so it won't be over-constrained that no congruent sets can be found.

4.4 Computational Complexity

Based on the discussion of pair generation and congruent set extraction processes, the pseudo code of proposed V4PCS algorithm is shown in Algorithm 1. Similar to the S4PCS, V4PCS also runs in a RANSAC fashion. By solving a problem of largest common point set, the algorithm finds a largest size of consensus set and subsequently obtains the best rigid transformation $\mathbf{T}$. When the matching rate h is 95% or higher, the algorithm is terminated since this is when a correct registration is probably found.

In the main loop of V4PCS, the first step is the Pair Generation (PG) from line 3 to 6. First, a large-volume tetrahedral base $\{\mathbf{q_i}\}$ is picked randomly from $\mathcal{S}$, and then the distance information of the base can be calculated as $d_1, d_2, \ldots, d_6$. With these distance information, the six pairs of laterals could be extracted from $\mathcal{M}$ through the $O(n)$ method developed in S4PCS. After that, all the extracted pairs are stored in six connectivity tables H. Therefore, this step take $O(6n)$ time. The second step is the Congruent Set Extraction (CSE) from line 7 to 9, which basically loops through all the extracted pairs from $\mathcal{M}$ and searches the congruent sets of tetrahedron U through the connectivity tables H. As the complexity of searching is $O(1)$, this step take $O(k)$ time. The last step is Congruent Set Verification (CSV) from line 10 to 16. All the congruent sets are tested by finding the rigid transformation to align $\{\mathbf{p_i}\}$ to $\{\mathbf{q_i}\}$ as well as to transform $\mathcal{M}$ to $\mathcal{S}$ for computing the matching score using the largest common point set method as in the S4PCS. We haven't altered this step, and the time complexity is the same. In summary, the time complexity of proposed V4PCS compared to S4PCS is decreased from $O(n + k + c)$ to $O(n + k)$, where n is the number of points in $\mathcal{M}$, k is the number of reported pairs and

```
Algorithm 1 Volumetric 4PCS
Input: Target and source point sets, $\mathcal{S}$ and $\mathcal{M}$
Output: Best transformation $\mathbf{T}$
$h = 0; \mathbf{T} = 0; l = 0;$
while $h < 0.95$ and $l < L$ do
    //Pair Generation (PG)
    $\{\mathbf{q_i}, i = 1 \dots 4\} =$ SelectTetrahedralBase $(\mathcal{S})$
    $d_1, d_2, \dots, d_6$ are the laterals of $\{\mathbf{q_i}\}$
    Extract $d_1, d_2, \dots, d_6$ pairs from $\mathcal{M}$ [4]
    Create connectivity table $H$ to store extracted pair index
    //Congruent Set Extraction (CSE)
    for all the pairs extracted from $\mathcal{M}$
        $U \leftarrow$ Search H to find other 5 pairs to form a congruent
        tetrahedron to $\{\mathbf{q_i}\}$
    end for
    //Congruent Set Verification (CSV)
    for all congruent tetrahedral sets $\{\mathbf{p_i}\} \in U$ do
        $\mathbf{t} \longleftarrow$ rigid transformation aligns $\{\mathbf{p_i}\}$ to $\{\mathbf{q_i}\}$
        $s =$ compute the score for $\mathbf{t}$ by common point set
        if $s > h$ then
            $h \leftarrow s; \mathbf{T} \leftarrow \mathbf{t};$
        end if
    end for
    $l \leftarrow l + 1;$
end for
return T
```

c is the number of congruent sets extracted. As all the extracted sets are the congruent sets, we don't need the ratio information to extract matching angle for filtering the false ones in the CSE step. Although the theoretical improvement does not look significant, it can practically speedup the process significantly and the magnitude of k is also reduced, which will be shown in the next section.

Mohamad et al. [29] had a similar insight and proposed to use a non-coplanar base by introducing one more dimension with an additional intermediate point in the base construction. However, in order to find the congruent set after pair generation, they need to rasterized a circle on the 3D grid that stores intermediate points by moving in discrete circular steps to check all grid cells that intersect the circle centered at e (see Fig. 6). This angular increment for the circular motion can be calculated using the following formula: $cos^{-1}(\frac{2h^2-2\epsilon^2}{2h^2})$, where h is the distance between two segments and ϵ is the given margin. There is a constant factor of $k_1 = 360/cos^{-1}(\frac{2h^2-2\epsilon^2}{2h^2})$ for each query, hence the time complexity compared with S4PCS is increased from $O(n+k+c)$ to $O(n+k_1k\ +c)$. Their method theoretically increases the time complexity in the CSE step due to the need of finding the overlapping intermediate point, which damages the beauty of the 4PCS algorithm stated in Remark 1. In fact, the efficiency gained in the CSV step in their method is trumped by the efficiency lost in the CSE step. To remedy the effects of the increased search time, they fixed the distances d_1 and d_2 to reduce the number of iteration, which reduces certain flexibility of the algorithm. In contrast, our method is able to take advantage of volumetric information and at the same time theoretically reduce the time complexity.

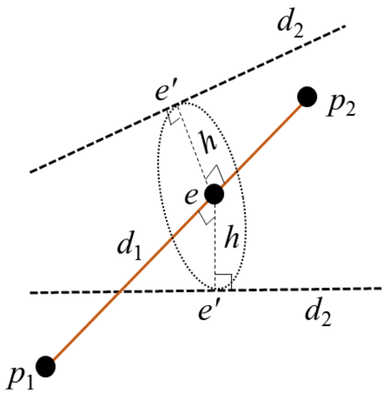


Fig. 6: *Congruent set extraction by Mohamad et al.* [29]. *Given a segment of length* $\boldsymbol{d_1}$ *with endpoints* $\boldsymbol{p_1 p_2}$ *and an intermediate point* $\boldsymbol{e}$*, the other segments of length* $\boldsymbol{d_2}$ *are found based on the sphere centered at* $\boldsymbol{e}$ *with radius* $\boldsymbol{h}$*.*

In terms of space complexity, S4PCS uses an efficient hyper-sphere rasterization method to extract pairs and store the indexes, the rasterized grid need $O(n)$ space for the input of n points. In addition, the extracted k pairs are stored in a hash table with angle information for finding the congruent set, which needs $O(k)$ space. Therefore, the total space complexity for S4PCS is $O(n+k)$. In the proposed V4PCS, the pair generation step is same with S4PCS, hence it needs $O(n)$ space for given n points. We construct a connectivity table for the extracted k pairs, where each edge is stored twice with both its end vertices, and thus it takes $O(2k)$ space. Therefore, V4PCS has a space complexity of $O(n+2k)$, which is slightly higher than S4PCS.

5. RESULTS

In this section, we will test the performance of proposed V4PCS on different data set and compared it with S4PCS in terms of robustness and efficiency. The results of S4PCS are generated by the code provided on their project webpage, which is implemented in C++. Our V4PCS is also implemented based on their code. Therefore, all the environmental settings and parameters are consistent in the testing. All experiments in this paper are run on a machine with Intel Core i7-4510U CPU @ 2.00GHz and 8 GB RAM. An application of mass customization using our method is also discussed at the end.

5.1 Robustness Test

For the sake of process rate, it is desirable to minimize the problem size and the number of parameters. However, it always comes with a trade-off in stability. When an algorithm is applied in real world applications, it is always desired that the algorithm is stable even in some extreme cases. Therefore, in order to test the robustness of the algorithms, we have conducted two experiments. One is testing with a small number of sample size n, which is a major contribution in the time complexity. Another one is testing with a small number of RANSAC iterations L, which is a multiplier on the total process time. These two parameters are also having a direct effect to the process speed, and there will be a large impact to the efficiency if an algorithm is robust enough to work with small values of the parameters.

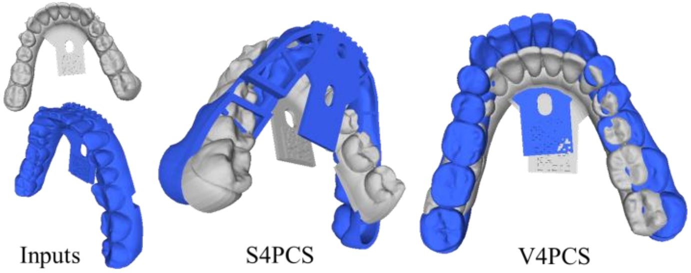


Fig. 7: *Alignment results for teeth models with n=32.*

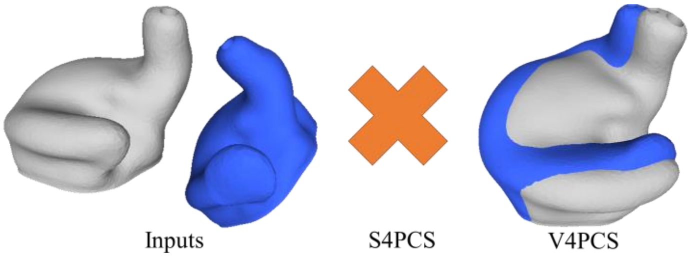


Fig. 8: *Alignment results for hearing aid models with n=24.*

Two examples are used in the first experiment as shown in Fig. 7 and Fig. 8. From the previous discussion, we know that the PG step takes $O(n)$ steps to generate all the pairs with the given lengths. The sample size n also directly affects the number of generated pairs k. It cannot be too small, otherwise no congruent pairs can be found, but it should be as small as possible. A sample size of 32 is used for aligning the teeth models in Fig. 7, which means there are only 32 sampling points on $\mathcal{M}$. Both S4PCS and V4PCS have been tested with the same setting and same number of iterations, and the results are shown in Fig. 7. Due to the reason of ambiguities from using a 2D plane to represent a 3D shape as stated in Observation 2, S4PCS returns a flipped alignment as the best result. Impressively, our proposed V4PCS can still find a proper alignment with this small number of sample size given. For the hearing aid models in Fig. 8, a sample size of 24 is used. In this example, the model is not in a planar shape making the S4PCS even fails to extract any congruent coplanar sets in the CSE step with the small number of sample size, i.e., $U = \emptyset$, but V4PCS is able to find the congruent tetrahedral bases and compute a close alignment result.

The second experiment is conducted to test with a small number of RANSAC iterations L. The whole procedure is run for L times to select different bases from $\mathcal{S}$ and to expect there are some iterations give good results. For a successful probability a larger number of L is preferred, but with a trade-off in time. Here, we demonstrate that the V4PCS as a richer representation for the input 3D shape can improve the probability and require fewer iterations. Two human models are shown in Fig. 9 as an example, where a small value of $L = 10$ is used. In this experiment, S4PCS returns a transformation that is flipped similar to the example of the teeth models, but V4PCS can roughly align the two human models pretty well.

From the two experiments discussed above, we have verified our Hypothesis 1 that the V4PCS algorithm is more robust and resistant to extreme cases by incorporating volumetric information to construct the base. Fig. 10 shows some other alignment results on different models by V4PCS.

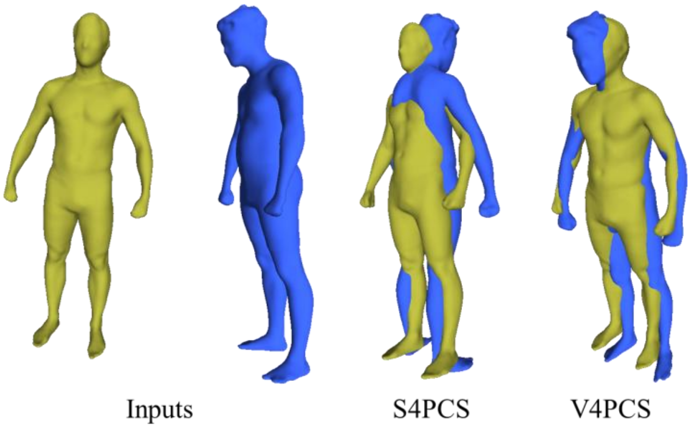


Fig. 9: *Alignment results for human models with L=10.*

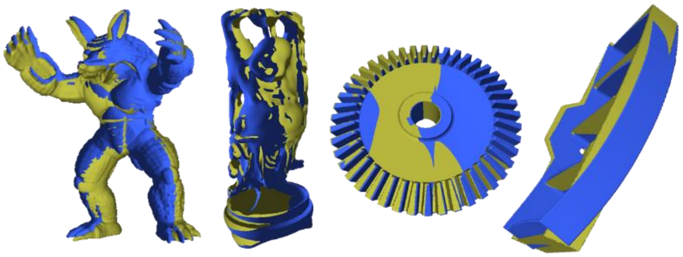


Fig. 10: *Alignment results for other models by V4PCS.*

5.2 Computational Test

To compare the computational complexity between V4PCS and S4PCS, the details of time statistics are plotted in Fig. 11. The experiment is conducted on the teeth models shown in Fig. 7 but not using a small sample size that S4PCS would fail. Fig. 11 also demonstrates the impact of sample size by using different number of n, and reports the total time and the time spent in each step: PG, CSE, and CSV. From Fig. 11, we can see that the PG time of two algorithms are similar, and V4PCS is slightly larger than S4PCS. This is because V4PCS extracts six laterals of tetrahedral base, while S4PCS extracts only two. This step could be further optimized for V4PCS, but since the PG step is performed in linear time, this constant difference in complexity is not significant.

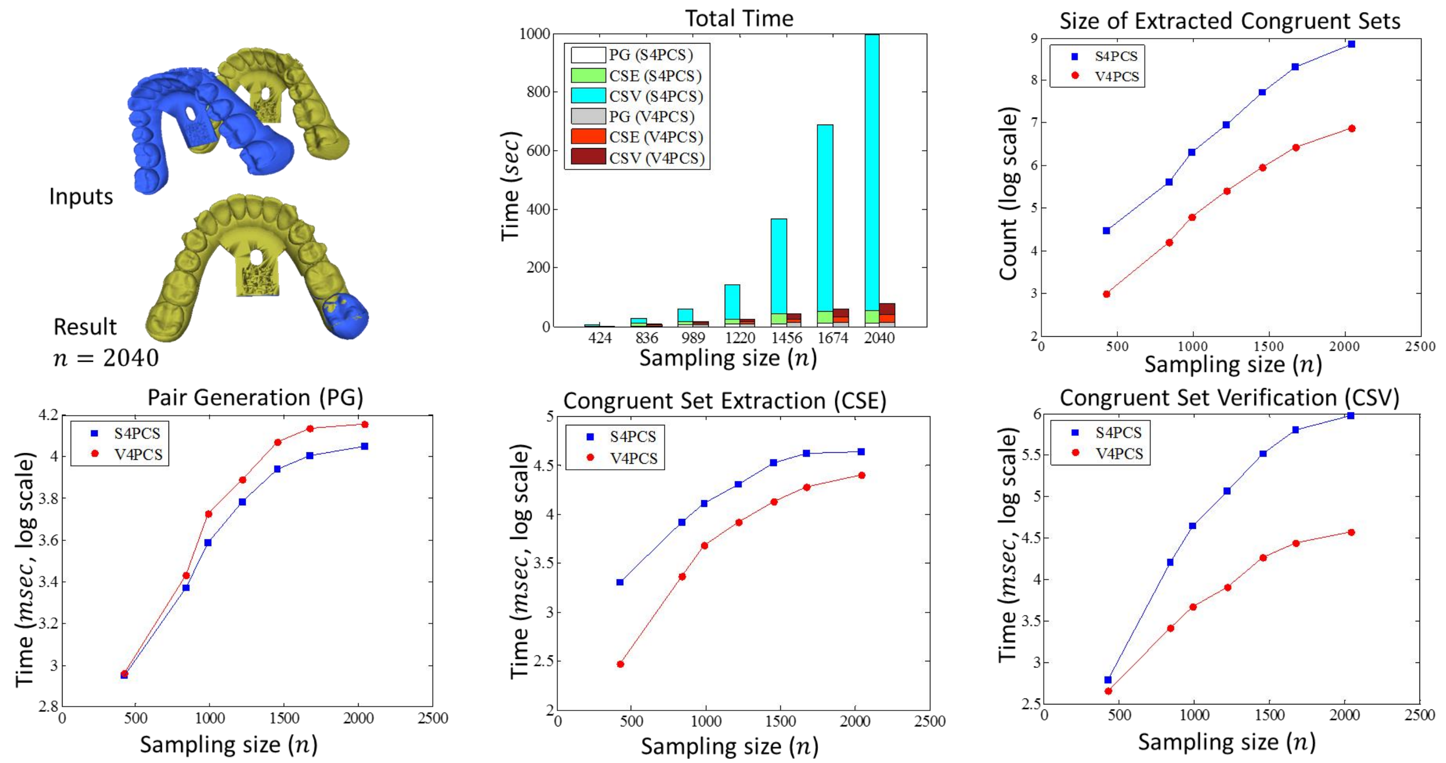


Fig. 11: *Computational time for V4PCS and S4PCS on two teeth models. Note that the number of extracted congruent sets and the timings for the subtotals of the PG, CSE, and CSV steps are in log-scale.*

The major improvement of the proposed V4PCS is the decrease in time at the CSE and CSV steps. Since we constructed a tetrahedron only using the distance information, the congruent bases can be extracted directly by index querying through a connectivity table, and hence we can see that the proposed method is much faster than the S4PCS method. Furthermore, the average numbers of extracted congruent sets for the base in each iteration are compared in Fig. 11, from which we can see that with increasing of samples, the congruent set is exponentially increased. As V4PCS makes use of volumetric information to filter many false pairs, the number of extracted congruent sets is far less than that from S4PCS. At $n = 2040$, the number of V4PCS is only about 0.3% of the S4PCS's one. Since the number of congruent sets is decreased with the rich representation in V4PCS, the time needed for verification is also greatly decreased. This also agrees with our previous deduction that if the number of congruent sets c is large, the term $O(c\, n \log m)$ in the CSV step is dominant.

The comparison in computational time is also done with other examples, and the results are shown in Table 1. Due to the reason that the algorithms are run by the RANSAC method, the performance can be different in different trials even with the same setting. To make the comparison fair, we run 50 times for each setting (i.e., different n) and report the average runtime. It can be seen from Table 1 that the average runtime of proposed V4PCS achieves about $70\% - 90\%$ improvement compared

Table 1 *Average time of registration with different sampling sizes*

model	#Sample in $\mathcal{S}$ (m)	#Sample in $\mathcal{M}$ (n)	S4PCS (sec)	V4PCS (sec)	Improve %
Hearing Aid	8638	449	6.820	1.755	74%
		1187	18.506	4.159	78%
		2000	88.230	20.304	77%
Teeth Aligner	7658	424	4.030	1.245	69%
		989	61.365	12.798	79%
		2040	990.397	51.305	95%
Head	2219	430	8.005	2.334	71%
		1074	96.150	8.691	91%
		2146	158.851	20.266	87%
Hand	15765	833	8.172	2.301	72%
		1574	40.651	8.816	78%
		2298	134.093	18.308	86%
Human Model	5620	423	5.460	0.704	87%
		852	12.426	3.456	72%
		2090	87.977	13.129	85%
Armadil lo	5503	387	1.459	0.241	83%
		899	15.392	3.527	77%
		1874	385.434	20.396	95%
Buddha	7932	422	15.909	2.871	82%
		817	31.560	4.192	87%
		2064	3793.584	208.65	94%
Gear	34705	436	14.334	1.283	91%
		861	127.494	20.517	84%
		2052	992.744	81.032	92%
Boeing part	26830	459	63.644	4.951	92%
		900	991.353	63.431	94%
		2109	1663.245	181.689	89%

to S4PCS. With the increase in the sample rate, the difference is more significant, and the V4PCS can achieve a maximum improvement of 95%, which is 20 times faster than the S4PCS.

From all the testing and comparison, we have verified our Hypothesis 2 that the V4PCS algorithm can minimize the number of congruent sets k and c, and it can improve a lot in the computational efficiency.

5.3 Mass Customization

As discussed in the introduction, the mass human-related data in mass customization can create unprecedented opportunities in advancing the theory, methods, and practice of design and manufacturing for products, system, and services. To exploit such huge, versatile, and highly contextualized data, the first step is always the registration to compute the correspondence between data, and then the feature and the similarity of the data can be identified and reused. Therefore, the robustness and efficiency of the V4PCS as shown in previous sections and experiments are the keys for the success in mass customization. Moreover, we extend the concept of information reuse to the registration step to make a further speedup in the context of mass customization as described here.

Table 2 shows the time statistics of each subroutine in V4PCS for matching two models (teeth aligner and hearing aid models) that are commonly seen in customization. It takes 240 iterations to align a pair of teeth models for the sample size is about 1000, and the total time is $2.61 + 4.15 + 6.04 = 12.8s$, which is fast. However, under the context of mass customization, assume we have 1001 models and we align all other models to the first one, the total matching time for the teeth models would be $1000 \times 12.8 \approx 3.6$ hours, which is too long time as a pre-processing step. Similarly, 2.1 hours are needed for the hearing aid models.

Table 2 *Time of each subroutines in V4PCS*

Model	Sample (n)	PG	CSE	CSV	Iterations
Teeth Aligner	989	2.61s	4.15s	6.04s	~240
Hearing Aid	1271	1.04s	2.48s	3.99s	~90

Fortunately, with the robustness of V4PCS, the algorithm can be customized for the application of mass customization. Since most of the models are very similar in mass customization (e.g., the teeth models of different patients are similar) and the volumetric base represents the shape of the 3D model very well. Therefore, if a proper base is selected for a same group of models, it can be safely reused. A proper base is defined as a large-volume tetrahedron that can capture the unique features of the given models. For example, the base shown in Fig. 12 is manually picked by hand on a hearing aid model, and we find that using this base can successful match with other hearing aid models in our dataset. In other words, it only requires one iteration in V4PCS for matching similar models, i.e., $L = 1$. As a result, the total registration time for the teeth models and the hearing aid models would be only $1000 \times 12.8/240 \approx 53s$ and $1000 \times 7.5/90 = 83s$ respectively, which is significantly decreased and becomes more favorable for mass customization.

It is worth to remark that there are two types of information reuse mentioned in this paper: one is reusing the similarities between models by alignment and identification; another is extending the concept of information reuse to speed up the registration step that aligns many models together. To the best of our knowledge, it is the first time that the concept of information reuse is extended to the registration step, which is enabled by the robustness of V4PCS.

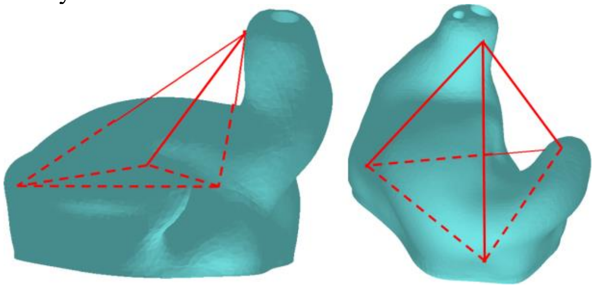


Fig. 12: *A pre-selected base for a hearing aid model*

6. CONCLUSION

This paper proposes a new algorithm – Volumetric 4PCS (V4PCS) framework for global registration of 3D shapes. Based on our observations and developments, we verified our hypotheses that incorporating volumetric information to create the 4-points base can improve the robustness and efficiency of the algorithm. Compared to Super 4PCS (S4PCS), we have theoretically demonstrate the computational complexity is reduced from $O(n + k + c)$ to $O(n + k)$, and we have also experimentally shown that V4PCS can work very well with small input points size and can achieve a maximum of 20 times speedup. The results show that the proposed method is reliable and can be applied to the applications such as mass customization in dealing massive human-related data. The concept of information used is also extended to the registration.

There are some limitations of this work. First, similar to the S4PCS, most of the parameters used in the algorithm have to be tested and set by hand for different models, which will be a problem in generalizing the algorithm in wider applications. Our future work will study how the parameters are related to the geometry of models, and develop methods to set the parameters automatically. Second, we have only tested the registration on intact models, which is the main focus in our application of mass customization. We will also test and study the performance of the algorithm on range data, i.e., partial and incomplete surfaces. Third, the Pair Generation step of V4PCS is slower than that of S4PCS due to there are six laterals and thus six distances have to be searched, which S4PCS has only two. This step could be optimized in some ways, for example, if equilateral tetrahedron (all sides are equal) is used, then only one search is needed. We will further investigate the possibility and potential of the method. Last, we are also using the RANSAC method to construct the first base and hoping that the algorithm can be success by chance. Although some heuristics are used to improve the probability, we plan to develop new geometric algorithm to pick the best base and thus no more iteration is needed.

ACKNOWLEDGMENTS

This material is based upon work partially supported by the National Science Foundation (NSF) grant # CNS- #1547167 and the Natural Sciences & Engineering Research Council of Canada (NSERC) grant # RGPIN-2017-06707. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the agencies.

REFERENCES

[1] T.-H. Kwok, H. Ye, Y. Chen, C. Zhou, and W. Xu, “Mass Customization: Reuse of Digital Slicing for Additive Manufacturing,” in *Volume 1A: 36th Computers and Information in Engineering Conference*, 2016, p. V01AT02A041.

[2] Paul J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes.pdf,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 14. pp. 239–256, 1992.

[3] D. Aiger, N. J. Mitra, D. Cohen-Or, D. Aiger, N. J. Mitra, and D. Cohen-Or, “4-points congruent sets for robust pairwise surface registration,” in *ACM SIGGRAPH 2008 papers on - SIGGRAPH ’08*, 2008, vol. 27, no. 3, p. 1.

[4] N. Mellado, D. Aiger, and N. J. Mitra, “Super 4PCS Fast Global Pointcloud Registration via Smart Indexing,” *Comput. Graph. Forum*, vol. 33, no. 5, pp. 205–215, Aug. 2014.

[5] T.-H. Kwok, K.-Y. Yeung, and C. C. L. Wang, “Volumetric template fitting for human body reconstruction from incomplete data,” *J. Manuf. Syst.*, vol. 33, no. 4, pp. 678–689, 2014.

[6] 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.

[7] S. Rusinkiewicz and M. Levoy, “Efficient variants of the ICP algorithm,” in *Proceedings Third International Conference on 3-D Digital Imaging and Modeling*, pp. 145–152.

[8] S. Bouaziz, A. Tagliasacchi, and M. Pauly, “Sparse Iterative Closest Point,” *Comput. Graph. Forum*, vol. 32, no. 5, pp. 113–123, Aug. 2013.

[9] A. W. Fitzgibbon, “Robust registration of 2D and 3D point sets,” *Image Vis. Comput.*, vol. 21, no. 13, pp. 1145–1153, 2003.

[10] O. Enqvist, K. Josephson, and F. Kahl, “Optimal correspondences from pairwise constraints,” in *2009 IEEE 12th International Conference on Computer Vision*, 2009, pp. 1295–1302.

[11] N. Gelfand, N. J. Mitra, L. J. Guibas, and H. Pottmann, “Robust global registration,” *Symp. Geom. Process.*, vol. 2, no. 3, p. 5, 2005.

[12] R. I. Hartley and F. Kahl, “Global Optimization through Searching Rotation Space and Optimal Estimation of the Essential Matrix,” in *2007 IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[13] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381–395, Jun. 1981.

[14] S. Irani and P. Raghavan, “Combinatorial and experimental results for randomized point matching algorithms,” in *Proceedings of the twelfth annual symposium on Computational geometry - SCG ’96*, 1996, pp. 68–77.

[15] Chu-Song Chen, Yi-Ping Hung, and Jen-Bo Cheng, “RANSAC-based DARCES: a new approach to fast automatic registration of partially overlapping range images,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 21, no. 11, pp. 1229–1234, 1999.

[16] R.Osada, T.Funkhouser, B.Chazelle, and D. Dobkin, “Matching 3D models with Shape distribution,” *Shape Model. Int.*, pp. 154–166, 2001.

[17] B. Drost, M. Ulrich, N. Navab, and S. Ilic, “Model globally, match locally: Efficient and robust 3D object recognition,” in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010, pp. 998–1005.

[18] Y. Guo, M. Bennamoun, F. Sohel, M. Lu, J. Wan, and N. M. Kwok, “A Comprehensive Performance Evaluation of 3D Local Feature Descriptors,” *Int. J. Comput. Vis.*, vol. 116, no. 1, pp. 66–89, Jan. 2016.

[19] J. Sun, M. Ovsjanikov, and L. Guibas, “A Concise and Provably Informative Multi-Scale Signature Based on Heat Diffusion,” *Comput. Graph. Forum*, vol. 28, no. 5, pp. 1383–1392, Jul. 2009.

[20] F. Tombari, S. Salti, and L. Di Stefano, “Unique shape context for 3d data description,” in *Proceedings of the ACM workshop on 3D object retrieval - 3DOR ’10*, 2010, p. 57.

[21] Y. Guo, F. Sohel, M. Bennamoun, M. Lu, and J. Wan, “Rotational Projection Statistics for 3D Local Surface Description and Object Recognition,” *Int. J. Comput. Vis.*, vol. 105, no. 1, pp. 63–86, Oct. 2013.

[22] R. B. Rusu, N. Blodow, Z. C. Marton, and M. Beetz, “Aligning point cloud views using persistent feature histograms,” in *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2008, pp. 3384–3391.

[23] R. B. Rusu, N. Blodow, and M. Beetz, “Fast Point Feature Histograms (FPFH) for 3D registration,” in *2009 IEEE International Conference on Robotics and Automation*, 2009, pp. 3212–3217.

[24] S. Salti, F. Tombari, and L. Di Stefano, “SHOT: Unique signatures of histograms for surface and texture description,” *Comput. Vis. Image Underst.*, vol. 125, pp. 251–264, 2014.

[25] C. Papazov and D. Burschka, “An Efficient RANSAC for 3D Object Recognition in Noisy and Occluded Scenes,” Springer, Berlin, Heidelberg, 2011, pp. 135–148.

[26] D. H. Ballard, “Generalizing the Hough transform to detect arbitrary shapes,” *Pattern Recognit.*, vol. 13, no. 2, pp. 111–122, Jan. 1981.

[27] Y. Lamdan and H. J. Wolfson, “Geometric Hashing: A General And Efficient Model-based Recognition Scheme,” in *[1988 Proceedings] Second International Conference on Computer Vision*, pp. 238–249.

[28] A. S. Mian, M. Bennamoun, and R. Owens, “Three-Dimensional Model-Based Object Recognition and Segmentation in Cluttered Scenes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 10, pp. 1584–1601, Oct. 2006.

[29] M. Mohamad, M. T. Ahmed, D. Rappaport, and M. Greenspan, “Super Generalized 4PCS for 3D Registration,” *Proc. - 2015 Int. Conf. 3D Vision, 3DV 2015*, pp. 598–606, 2015.