Geometric Deep Learning for Shape Correspondence in Mass Customization by 3D Printing

Jida Huang
Mechanical and Industrial Engineering
University of Illinois at Chicago
Chicago, Illinois 60607
Email: jida@uic.edu

Hongyue Sun
Industrial and Systems Engineering
University at Buffalo, SUNY
Buffalo, New York 14260
Email: hongyues@buffalo.edu

Tsz-Ho Kwok
Mechanical, Industrial and Aerospace
Engineering, Concordia University
Montreal, QC H3G 1M8, Canada
Email: tom.thkwok@gmail.com

Chi Zhou *
Industrial and Systems Engineering
University at Buffalo, SUNY
Buffalo, New York 14260
Email: chizhou@buffalo.edu

Wenyao Xu
Computer Science and Engineering
University at Buffalo, SUNY
Buffalo, New York 14260
Email: wenyaoxu@buffalo.edu

Many industries, such as human-centric product manufacturing, are calling for mass customization with personalized products. One key enabler of mass customization is 3D printing, which makes flexible design and manufacturing possible. However, the personalized designs bring challenges for the shape matching and analysis, owing to the high complexity and shape variations. Traditional shape matching methods are limited to spatial alignment and finding a transformation matrix for two shapes, which cannot determine a vertex-to-vertex or feature-to-feature correlation on the two shapes. Hence, such a method cannot measure the deformation of the shape and interested features directly. To measure the deformations widely seen in the mass customization paradigm and address the issues of alignment methods in shape matching, we identify the geometry matching of deformed shapes as a correspondence problem. The problem is challenging due to the huge solution space and nonlinear complexity, which is difficult for conventional optimization methods to solve. According to the observation that the well-established massive databases provide the correspondence results of the treated teeth models, a learning-based method is proposed for the shape correspondence problem. Specifically, a state-of-the-art geometric deep learning method is used to learn the correspondence of a set of collected deformed shapes. Through learning the deformations of the models, the underlying variations of the shapes are extracted and used for finding the vertex-to-vertex mapping among these shapes. We demonstrate the application of the proposed approach in the orthodontics industry, and the experimental results show that the proposed method can predict correspondence fast and accurate, also robust to extreme cases. Furthermore, the proposed method is favorably suitable for deformed shape analysis in mass customization enabled by 3D printing.

1 Introduction

Mass customization is an emerging paradigm to achieve variety and customization in product geometry, functionality, and property at near mass production price [1]. The customized products are challenging to be mass-produced in traditional manners due to high geometric variation and product functionality. As an emerging disruptive technology, 3D printing, also known as additive manufacturing, can rapidly fabricate complex physical object and therefore enables profitable mass customization [2]. For instance, in the orthodontics industry as shown in Figure 1, highly mass-customized transparent dental aligners are fabricated by 3D printing to allow the patient to wear on the teeth and progressively move the misplaced teeth to the desired position and orientation. The patient typically receives a pair of aligners for upper and

---

*Corresponding author.

lower teeth every two weeks during the six-month to twelve-month treatment period. It is reported that the company runs the 3D printers 24 hours and produces 40,000 unique aligners per day [3]. The need for a large amount of different complex shapes in a short period requires mass customization techniques for aligner production.
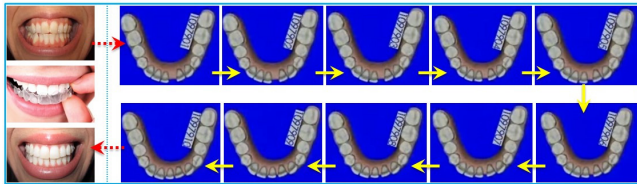

Fig. 1. Teeth aligners for orthodontic treatment [4]. Left-top: aligner before treatment; Left-middle: aligner during treatment; Left-bottom: aligner after treatment; Right: digital models for the progressively fabricated transparent aligners.
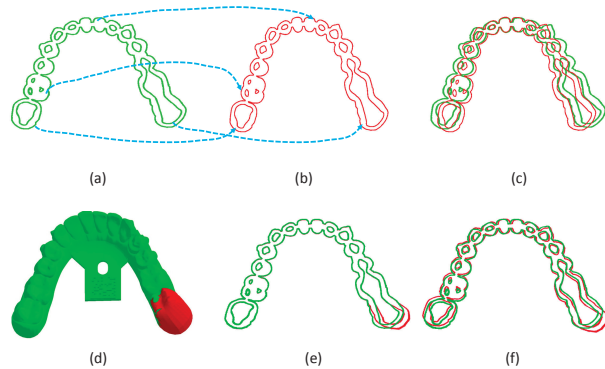

Fig. 2. Shape alignment versus shape correspondence: (a) teeth model with wider opening; (b) teeth model with narrower opening, the dash lines show the correspondence between the two models; (c) the alignment between the two models in (a) and (b); (d, e) alignment between two locally deformed teeth models based on maximum correspondence level in both 3D and 2D views; (f) alignment between the two models in (d-e) based on minimum distance error.

To promote the broad applications of 3D printing and fully realize mass customization, one needs to guarantee the product geometric accuracy during design and manufacturing. This is challenging to achieve due to the high geometric complexity and large variations. One can imagine that the teeth model of different people is totally different, though the general structure looks similar. One of the most common practices of geometry operation in mass customization is the shape geometry matching. For the teeth aligner example, the shapes of the patient's teeth during the whole treatment period have to be systemically tracked and recorded for the aligner design. The dentist firstly needs to manually mark several "feature points" on the scanned teeth model. Then the CAD software is used to match these marked points of the newly scanned teeth model with the initial one (template), based on which each individual tooth can be extracted, marked, and numbered, allowing them to be individually adjusted to a preferable position and orientation. Besides, the scanned teeth model (patients' teeth imprint) will be matched and compared with the most recently used aligner model to check the effectiveness of the treatment in the prior period.

Based on the similarities of the customized models, algorithms have been proposed to address the computational reuse problem [5, 6]. These algorithms tend to utilize the existing geometry and topology for information-reuse in the mass customization applications. However, these algorithms assume the matching between the target model and the template model is given, which may not be available in real practice. What is more, the printed aligner needs to be compared with the target model (prescription from the dentist) to evaluate the quality of the printed product, which is again based on the matching result. It is therefore very desirable to design an effective shape matching procedure to capture the geometry variations (e.g., structure deformations, local feature changes) for mass customization.

For shape matching, the most intuitive way is to find a transformation to align two shapes together, also known as

rigid registration. The registration method tends to find a spatial transformation between the input shapes. Based on the transformation, one can align one shape to the other and observe the overall spatial difference of two shapes. However, the rigid shape registration is not an appropriate approach to depict the deformation and variance between the models in the mass customization applications in two folds:

1. The rigid registration approach minimizes the error of the Euclidean distance between the closest points from the current model to the target model. For the global deformation, the two models cannot be spatially well-aligned regardless of the effectiveness of the optimization algorithm. As shown in Figure 2 (a-c), the two teeth models with global deformation need to be well-mapped through the corresponded individual teeth features; however, they cannot be well-aligned spatially due to the large deformation. For the local deformation, the rigid alignment algorithms tend to align the locally deformed features by sacrificing the non-deformed features, which otherwise can be perfectly aligned. As shown in Figure 2 (d-f), the two teeth models with local deformation (the right-side wisdom tooth is moved) can be well matched based on the maximum correspondence (d-e); however, the traditional alignment algorithms optimize the Euclidean distance error between the two models and result in mismatched alignment (f);

2. The rigid alignment algorithms tend to align the individual vertices from the two models by optimizing the spatial transformation matrix, and it cannot find a vertex-to-vertex and feature-to-feature mapping between the deformed shapes, thus cannot make further analysis of the deformation behavior in the application of mass customization.

Therefore, instead of finding an optimal spatial transformation, we need to determine the mapping relation between the deformed shapes in mass customization. Such a

mapping relation is usually represented as a vertex-to-vertex correspondence, i.e., finding a corresponding vertex on one shape to the given vertex on the other shape. So this problem is called *shape correspondence problem*. The problem is challenging because the solution space is big and nonlinear. It has $O(N!)$ possibilities for mapping N vertices on both shapes. What is more, in the scenario of mass customization, the number of deformed shapes to be matched is enormous, which makes the problem even more challenging. Currently, in the teeth aligner industry, the common approaches still primarily rely on manual operations (such as marking the feature vertices on the teeth model and mapping the patients' teeth models in different periods) based on the dentists' expertise and experience, which is extremely tedious and inefficient, and the time spent on such manually marking tasks could be 10 minutes to 2 hours and without guarantee of finding the perfect matching to the reference model [7]. This hugely hinders the digital model prepossessing, especially for a large number of models, which is common in the application of mass customization.

To address this challenge, this paper investigates an automated way of finding the shape correspondence with an ultimate goal of integrating mass customization with 3D printing. The optimization for finding the shape correspondence of a large number of complex shapes is challenging. In practice, we observe that the massive databases of the well-established correspondence results for the treated teeth models provide valuable resources for us to predict the correspondence features of the new teeth models. Thus, we hypothesize that the highly similar yet complex teeth models share the intrinsic correspondence relation, which can be learned from the existing models in the databases, and the learning results can be used to automatically map the corresponded features between the new models to the existing models. The objective of this paper is to investigate an effective machine learning approach to solve the shape correspondence problem in mass customization. We will explore the emerging deep learning techniques to extract the intrinsic relation for the shape correspondence. In particular, we will focus on a geometric deep learning approach owing to its potential to extract invariant features among the customized models. The input data are the vertex coordinates of the teeth models, the output data are the elements of the canonical label set, and a new convolution operation is designed based on the metric of geodesic distance, which captures the shape variation. The main contributions of the work can be summarized as follows:

1. We identify the shape matching problem in mass customization as a correspondence problem, which is more suitable to depict the relation of deformed shapes and conduct further analysis of the shape deformation behavior.
2. Based on the problem property, in which the established database of shape correspondence already exists in mass customization, a learning-based method is proposed for the correspondence problem.
3. A geometric deep learning method is used for corre-

spondence learning. Experimental results verify that the proposed method can predict new shape correspondence for deformed shapes. Also, the proposed method is robust to extreme cases and efficient for making new predictions.

We will use the teeth aligner in the orthodontic industry as an application example to present the proposed approach, and it should be noted that the approach is generic and can be easily extended to other applications in mass customization, including medical industry (hearing aid) [8], entertainment industry (movie characters) [9], jewelry industry (customized rings) [9], and toy industry [10]. The rest of the paper is organized as follows. Section 2 will briefly review the related works. The correspondence problem will be discussed in Section 3. Section 4 will introduce the architecture of the proposed deep neural network, and it is followed by the experimental results in Section 5. Section 6 will conclude the paper.

2 Literature Review

In this section, we firstly review the related work on traditional shape alignment in design and manufacturing, the correspondence problem, and then summarize the 3D deep learning models applicable for shape matching.

2.1 Shape alignment in design and manufacturing

Shape matching is naturally associated with a classical problem, shape alignment. Shape alignment is a process to align different three-dimensional (3D) shapes. Many research works have been explored in diverse aspects, and interested readers are referred to a survey paper [11]. In shape registration, the input includes two partial scans of the same object.

However, in many practical applications, the matching objects are different or include a certain degree of deformation, even for the same object. In [12], similar but different shape matching problem is considered. The shape matching is also widely used for geometric variation modeling in the manufacturing area. The majority of the matching problems treat the product as a rigid body. For instance, Tootooni et al. performed a classification study for the fused deposition modeling (FDM) printed part geometric integrity variation using 3D vertex cloud data, which are matched with the CAD design [13]. These methods did not consider the deformation of the products. In contrast, many other studies imply the necessity to investigate the non-rigid bodies in manufacturing [14,15]. For instance, Camelio et al. studied the geometrical variation propagation at the discrete measurement vertices in the automotive body assembly process with a compliant assemble system [14]. Other than just focusing on the limited discrete measurement vertices, Zhou et al. proposed the morphing of geometry from stage to stage and learned the mapping between complex surfaces via affine and non-affine transformations for the surface quality control [16].

2.2 Shape correspondence problem

In general, the non-rigid matching can be solved by shape correspondence problem. The goal of the classical correspondence problem is to find a vertex-wise matching between the vertices of two shapes. For example, a theoretical and computational framework is proposed for isometry invariant recognition of point cloud data in [17]. Mateus et al. proposed an articulated shape matching using Laplacian eigenfunctions and unsupervised point registration. A convex optimization and game theory-based method is used in [18] and [19], respectively. Typically, the computational complexity of such methods is high, but the scalability is an essential issue for mass customization. These methods are thus not suitable within the context of mass customization.

Rather than vertex-wise correspondence, other works used a soft correspondence approach to assign a vertex on one shape to more than one vertex on the other. For instance, a soft mapping between surfaces is proposed in [20], while Ovsjanikov et al. used a function map to represent the correspondence between shapes [21]. In [22], a matrix completion method is proposed for solving the shape correspondence problem.

2.3 Deep learning beyond Euclidean data

As an emerging machine learning technique, deep learning has been widely used in image analysis, computer vision, and manufacturing areas [23,24], and achieved remarkable breakthroughs. In order to extend the deep learning method from 2D learning to 3D learning, many attempts have been made to extend the convolution operation to 3D problems. The most direct way is to use a voxel representation of 3D shapes. Wu et al. represented a geometric 3D shape as a probability distribution of binary variables on a 3D voxel grid, using a convolutional deep belief network to learn the distribution of complex 3D shapes, and achieved object recognition [25]. Similarly, Brock et al. trained voxel-based variational autoencoders for object classification [26]. Balu et al. used voxel data to learn salient features from a CAD model of a mechanical part and determined the part manufacturability [27]. Qi et al. used point cloud as input to deep net architecture for 3D classification [28].

However, the main drawback of such approaches is representing the geometric data in a Euclidean structure. First, for complex 3D objects, the Euclidean representations such as depth images or voxels may lose significant parts of the object or its fine details, or even break its topological structure. Second, the Euclidean representations are not intrinsic and vary as the result of the pose or deformation of the object. Extracting the invariance to shape deformations is extremely difficult with such methods and requires complex models and massive training data sets due to a large number of degrees of freedom involved in describing non-rigid deformations. In order to extend the convolution operation for intrinsic geometric deep learning, Bronstein et al. proposed geometric deep learning, which goes beyond Euclidean data [29]. Masci et al. firstly considered convolutional neural networks (CNN) in non-Euclidean domains (surfaces) by using the geodesic CNN model [30]. The method is improved by Boscaini et al. [31] and further generalized by Monti et al. [32].
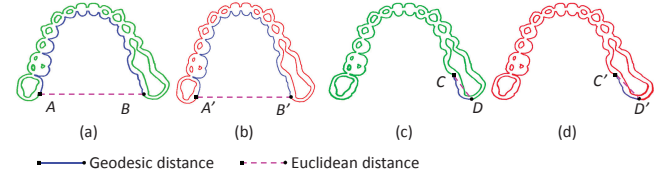

Fig. 3. Geodesic distance versus Euclidean distance: (a-b) the comparison between the geodesic distance and Euclidean distance for globally deformed teeth models; (c-d) the comparison between the geodesic distance and Euclidean distance for locally deformed teeth models.

For the teeth aligner application, the geodesic distance (distance between geographic vertices along the path conforming to the surface) has little or no changes, though the Euclidean distance (straight-line distance between two vertices in Euclidean space) has large changes under the non-rigid deformation. As shown in Figure 3(a-b), under global deformation, the Euclidean distances between the corresponded vertices are quite different ($\|AB\| > \|A'B'\|$), while the geodesic distance are almost the same ($d(A,B) = d(A',B')$). Similarly, under the local deformation (Figure 3(c-d)), the Euclidean distances are different due to shape stretching ($\|CD\| < \|C'D'\|$), while the geodesic distance are almost the same ($d(C,D) = d(C',D')$). Therefore, the geodesic distance will be used as the metric to capture the invariant features among the shape variations of the mass-customized models in this paper.

3 Problem Definition

The shape matching includes two different problems: shape alignment and shape correspondence. In this section, the shape alignment problem is firstly introduced, then the shape correspondence problem is defined. In both problems, the input is two 3D shapes $\mathcal{X}$ and $\mathcal{Y}$, typially modeled as Riemannian manifolds.

3.1 Shape alignment problem

In the shape alignment problem, the objective is to find a spatial transformation $T \in R^3$: $T(\mathcal{X}) \to \mathcal{Y}$, to align two shapes. This transformation usually includes the rotation and translation components. Furthermore, the alignment is normally solved through minimizing a specific distance function:

$$T^* = \operatorname*{arg\,min}_{T \in R^3} E(T(\mathcal{X}), \mathcal{Y}) \tag{1}$$

The distance function E can be Euclidean distance or any other application-based distance matrices.

From Figure 4 (a), it can be seen that the spatial transformation can provide a rough estimate of the similarity between the two models, but from such rough alignment we cannot tell the vertex or feature relation between the two shapes, i.e., given a vertex on one shape, we cannot tell its corresponding vertex on the other shape. Thus, within the deformation, it cannot determine how a vertex on the shape is moved. Thus, a method to extract the vertex-to-vertex or feature-to-feature relation (mapping between two models) is needed.

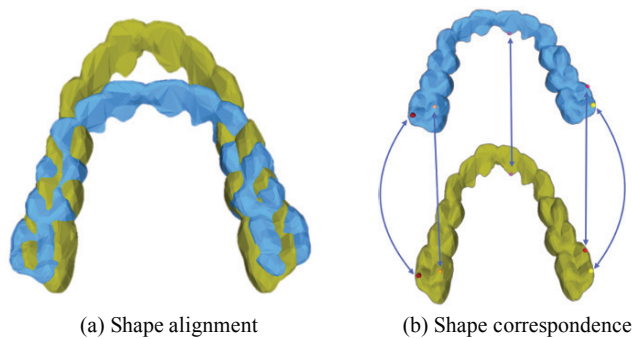(a) Shape alignment   (b) Shape correspondence

Fig. 4. Different shape matching method: (a) minimizing spatial transformation distance; (b) mapping via shape correspondence

3.2 Shape correspondence problem

Figure 4 (b) shows the correspondence of two models, in which each vertex on one model is mapped to a corresponding vertex on the other model. In this case, we extract the vertex-to-vertex relation between two models rather than finding a spatial transformation between them. Once we determined such a relation, we can further identify how each vertex on the model is deformed by comparing the spatial position of the corresponded vertices. Furthermore, we can also compare a vertex and its neighboring vertices with the correspondent one on the other model to see how a local structure is deformed. Thus, shape correspondence is more suitable for depicting the mapping relationship between two deformed models.

In shape correspondence problem, the goal is to find a meaningful vertex-wise correspondence $\varphi: \mathcal{X} \to \mathcal{Y}$. Specifically, shape $\mathcal{X}$ and $\mathcal{Y}$ contain the number of m and n vertices respectively. Here, the number m and n can be selected as the vertices on the triangulated mesh model or through a uniform sampling on the shape. The correspondence of two shapes (mapping relation) can be described as finding a mapping π: $\{x_1, \ldots, x_m\} \to \{y_1, \ldots, y_n\}$. Such a mapping is represented as a permutation matrix $\Pi \in \{0,1\}^{m \times n}$. Denoting the space of $m \times n$ permutation matrices as $\mathcal{P}$, the shape matching approaches frame the correspondence problem as,

$$\pi^* = \arg\min_{\pi \in \mathcal{P}} F(\Pi) \qquad (2)$$

where F is the fidelity term intended to align a set of vertex-wise descriptors encoding the similarity between the vertices [33].

An optimal vertex-to-vertex correspondence is usually challenging to find because the solution space is big and nonlinear, especially when the m and n are large. In practice, the problem can be transformed into a soft correspondence problem, that is for a vertex x on a shape $\mathcal{X}$, the goal of the problem is to find a m-dimensional output which can be interpreted as a correspondence probability of vertex x to the vertices on shape $\mathcal{Y}$. Thus each vertex on shape $\mathcal{X}$ would have m outputs indicating the probability of the vertices corresponding to point x. The output of all the vertices of the shape can be arranged as a $m \times n$ matrix with the element of the probability of vertex x mapped to vertex y.

Theoretically, it is very time-consuming to find the optimal solution of the correspondence problem because the problem cannot be solved in polynomial time. Practically, finding the desired correspondence solution for the mass customization problem is very challenging. Firstly, the number of vertices on the shape is big. For example, an approximated triangulated teeth aligner model usually has more than 8K vertices. Secondly, in teeth aligners industry, the number of teeth models that need to be extracted for the correspondence to the template or previous treatment model is vast. This hugely hinders the computational efficiency of the correspondence extraction. Therefore, a fast and automated way of finding the correspondence between shapes is urgently needed in mass customization.

4 Correspondence Learning

As discussed in the previous section, the shape matching in mass customization is modeled as a correspondence problem. Inspired by the fact that most of the models are similar despite the deformations in the application of the mass customization paradigm, a learning-by-examples approach is introduced to find the correspondence of similar shapes in the same category. In such a scenario, we assume the correspondence of a set of training shapes in the same category is already known and collected. Our goal is to learn from these examples on how to match two deformed shapes with a vertex-to-vertex correspondence. In order to extract the underlying intrinsic information among these deformations, a deep learning method is introduced for such information extraction.

4.1 Overview of the proposed learning-based method

In the learning-based method, the assumption is that the vertex-to-vertex correspondence of a set of samples is already collected, i.e., the ground-truth correspondence of such a group of shapes are already known. From the given data set, the intrinsic correspondence property of the shapes is learned from these examples. Moreover, for the learning-based method, CNN is introduced for correspondence learning in this paper.

Figure 5 depicts a brief overview of the proposed

learning-based method. It can be seen that given the ground-truth correspondence of shapes $\pi^* : X \to Y$ in the training examples, our objective is to learn how to match two new shapes from these ground-truth correspondences. During the learning stage, the relation between each vertex x on a query shape X to its corresponding vertex $\pi^*(x)$ on the reference shape Y in the collected training dataset are learned. Accordingly, the vertex-to-vertex correspondence function $f_\Theta(x)$ needs to be solved and extracted. Here, Θ is the network parameters to be optimized, f is the network which outputs a corresponding vertices vector by giving an input vertex x.
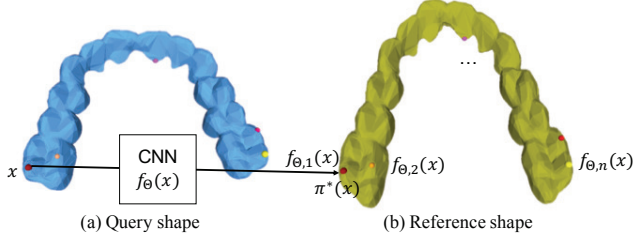

Fig. 5. Overview of correspondence leaning. A CNN is used for the correspondence learning. The goal is to find a n-dimensional output, which can be interpreted as a correspondence probability of vertex x from the query shape to vertices on the reference shape.

Once the leaning stage is completed, the correspondence function f can be learned. Based on the optimized network parameters, Θ, we can directly use $f_\Theta(x)$ to infer the vertex correspondence on the new shapes. During the stage of inference, we assume there are n vertices on the reference shape Y. By passing the vertex x as input into the learning-based function $f_\Theta(x)$, the output will be an n-dimensional metric, which represents the probability of vertex x corresponding to vertices on the reference shape Y.

$$f_\Theta(x) = (f_{\Theta,1}(x), ..., f_{\Theta,n}(x)) \qquad (3)$$

In summary, the deep learning method, CNN, is introduced for the shape correspondence learning in the mass customization application. In the following sections, the details of how to solve the leaning function $f_\Theta(x)$ and the detailed steps of CNN in the learning stage will be introduced.

4.2 Convolution operation on mesh data
One of the key elements for feature learning in CNN is the convolution operation. However, most of the existing works are limited to image data, in which the convolution operation is well defined in Euclidean grid-like data. For the data in the correspondence problem, the shapes are represented as a Riemannian manifold with the format of mesh in the 3D non-Euclidean domain. Given such mesh data, the convolution operation in the image domain is no longer suitable for non-Euclidean manifold data learning. Hence, to
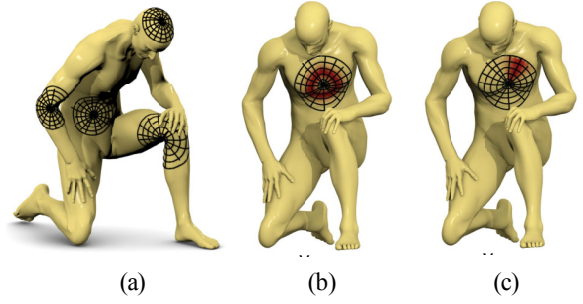

Fig. 6. Patch operator construction. (a) Local geodesic patch examples; (b) Geodesic radial weight w_ρ; (c) Angular weight w_θ. (Image courteous of [30]).

utilize the CNN for mesh data learning, a new convolution operation should be designed in 3D non-Euclidean domain.

In order to design such a convolution operation and represent the intrinsic variations of the deformations of the manifolds, Masci et al. [30] proposed a generalization of convolution operation to mesh data. In this generalized method, the operation is based on the definition of a local charting procedure in geodesic polar coordinates, named as patch operator.

Patch operator is initially designed for constructing an intrinsic shape context descriptor by Kokkinos et al. [34]. It mainly considers the local neighboring area around a given vertex on the manifold to describe such a vertex. The definition of the patch operator is

$$(D(x)f)(\rho, \theta) = \int_X w_{\rho,\theta}(x, \xi) f(\xi) d\xi \qquad (4)$$

The patch operator maps the values of a function f at a neighborhood of the vertex $x \in X$ into the local polar coordinates ρ, θ. Here $d\xi$ denotes the area element induced by the Riemannian metric, and $w_{\rho,\theta}(x, \xi)$ is a weighting function localized around vertex x with geodesic radius ρ and angle θ. Figure 6 shows examples of the construction of local geodesic patches with two different types of weights w_ρ and w_θ.

Intrinsic convolution. $D(x)f$ can be regarded as a patch on the manifold and $(D(x)f)(\rho, \theta)$ is interpolating f in the local coordinates, which can be used to define the convolution operator for manifold data.

$$(f * a)(x) = \int_0^{2\pi} \int_0^{\rho_{max}} a(\rho, \theta)(D(x)f)(\rho, \theta) d\rho d\theta \qquad (5)$$

Here the convolution operation can be thought of as matching a template $a(\rho, \theta)$ with the extracted patch at each vertex. In angular coordinate, the patch can be rotated at any angle, which would lead to angular ambiguities [32]. A maximum is taken over all possible rotations of the template to eliminate such ambiguity.

$$(f * a)(x) = \max_{\Delta\theta \in [0, 2\pi)} \int_0^{2\pi} \int_0^{\rho_{max}} a(\rho, \theta + \Delta\theta)(D(x)f)(\rho, \theta) d\rho d\theta \quad (6)$$

The above operator is used to define an analogy of traditional convolution operation. For discrete triangulated mesh data, it can be implemented through a discrete local system of geodesic polar coordinates containing N_θ and N_ρ radial bins [35].

4.3 Non-Euclidean CNN

With the defined non-Euclidean convolution operation for mesh data, it can be directly used in the convolution layer to learn the templates of a in Equation (6). The templates represent different local features of each vertex on the mesh. The proposed network consists of various subsequent layers. The architecture of the proposed network mainly consists of the following different type of layers.

Intrinsic convolution (IC) layer uses the operator from Equation (6) to replace the classical Euclidean convolution. The layer is specified by a certain number of filters, a_{qp}, along with additive biases b, and it operates by computing the convolution of the previous layer with each of those filters, afterwards adding the biases. The IC layer contains PQ filters arranged in banks (P filters in Q bank), each bank corresponds to an output dimension.

$$g_q^{out}(x) = \sum_{p=1}^P (f_p^{in} * a_{qp})(x) + b_q; \quad p = 1,...,P; q = 1,...,Q \quad (7)$$

where a_{qp} is the learnable coefficients of the pth filter in the qth filter bank. The IC layer is mainly used to extract the hierarchy composites of the feature associated with the vertex on the mesh data.

Fully connected (FC) layer is a linearly connected layer to adjust the input and output dimensions. Given a P-dimensional input $X^{in} = (x_1^{in},...,x_P^{in})$, the fully connected layer produces a Q-dimensional output $Y^{out} = (y_1^{out},...,y_Q^{out})$ by using a learnable weight vector w,

$$g_q^{out}(x) = \eta(\sum_{p=1}^P w_{qp} f_p^{in}(x)); q = 1,...,Q \quad (8)$$

The output is optionally passed through a non-linear function such as the ReLU [36], $\eta(t) = max\{0, t\}$. The ReLU is an activation function which can have a better gradient propagation and scale-invariant, also have the effect of sparse activation for the network [37].

Softmax layer is used to classify the output from the previous layer. In this paper, the output of vertex j is a n-dimensional probability vector, whose element represents the probability of vertex j corresponding to vertex i on the other shape.

$$(g_i^{out})_j = softmax(f_i^{in}) = \frac{\exp(f_i^{in})}{\sum_{i=1}^n \exp(f_i^{in})} \quad (9)$$

where $i = 1,...,n; j = 1,...,m$ are the number of vertices on each shape, respectively.

Dropout layer is a fixed layer to prevent overfitting [38]. The term "dropout" refers to dropping out units (hidden and visible) in a neural network. Dropping a unit out means we temporarily remove the unit from the network, also remove all of incoming and outgoing connections of the unit. The selection of which units to drop is random.

Batch normalization layer is another fixed layer to reduce the training time of large network [39]. It normalizes each mini-batch during stochastic optimization to keep zero mean and unit variance, and then performs a linear transformation of the form:

$$g_p^{out} = \frac{f_p^{in} - \mu}{\sqrt{\sigma^2 + \varepsilon}} \gamma + \beta \quad (10)$$

where μ and σ^2 are the mean and the variance of the training dataset by using exponential moving average method. To avoid numerical errors, a small positive constant ε is used here.

4.4 Learning the correspondence

Once the non-Euclidean CNN is constructed, we can apply it to the collected ground-truth data to train the network. When training the network, a cross-entropy function is used as the objective function to be minimized for obtaining the optimal network parameters.

Let m and n denote the number of vertices of shape $\mathcal{X}$ and $\mathcal{Y}$, respectively. For a vertex x on shape $\mathcal{X}$, the network produces a n-dimensional output as described in Section 4.1, which can be interpreted as a correspondence probability on the reference shape $\mathcal{Y}$. The output of the network is arranged as a $m \times n$ matrix. For each matrix element $f_\Theta(x,y)$, it means the probability of vertex x being mapped to y. And $y^*(x)$ denotes the ground-truth correspondence. The ground-truth correspondences are collected as $\mathcal{T} = \{(x, y^*(x))\}$, the optimal parameters of the network Θ are determined by minimizing the following logistic regression loss function.

$$\ell(\Theta) = -\sum_{(x,y^*(x)) \in \mathcal{T}} \log f_\Theta(x, y^*(x)) \quad (11)$$

which represents the divergence between the probability distribution produced by the network and the ground-truth distribution.

5 Experimental Study

In this section, several different types of experiments are conducted to evaluate the performance of the proposed geometric deep learning method for the correspondence problem. The method is tested with a set of non-rigid shapes with various degrees of deformations.

For the training dataset, we collect 100 teeth aligner models from ten different patients with ten different treatment stages. Since the correspondence is in pair-wise, i.e.,
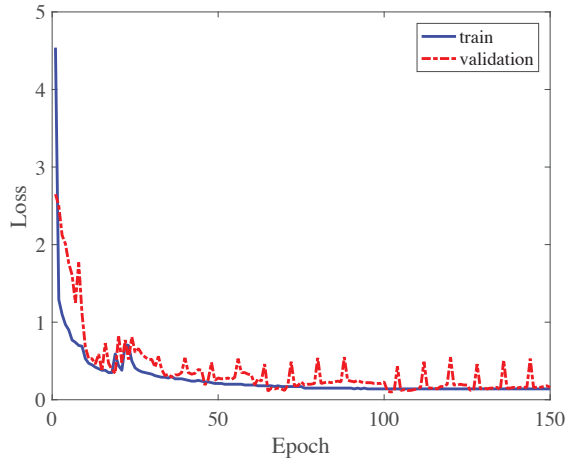
Fig. 7. Convergence curve of the proposed learning method

FC512 + Softmax. That is, the network architecture begins with a fully connected layer with 64 neuron nodes, followed by three convolution layers with 64, 128 and 256 filter bank sizes, two fully connected layers with dimensions of 1024 and 512 respectively, and lastly, a softmax layer is included. The main rationale of designing such a structure is based on the fact that the depth of the network dominantly determines the training time of the network. Figure 7 shows the convergence curve of the network training process, from which it can be seen that after 50 epochs, the network is converging to a small loss (~ 0.016) for both training and validation set. It reveals that the proposed geometric deep learning method can learn the shape correspondence of the ground-truth data and archive a good fitting performance.

It is worth to mention that in this work, we use a machine learning method to transform a traditional optimization problem, which is challenging to solve in polynomial time, into a fast and solvable problem. The prediction time for finding a correspondence between two shapes is approximately 0.5 seconds. This is very significant for identifying the shape correspondences in the mass models and satisfying the time requirement of mass customization.

Figure 8 visualizes some typical samples of correspondence predicted by the geometric deep learning method using colorized mapping, where colors are transformed using raw vertex-wise correspondence as the input to the functional maps. That is, the corresponded vertices are coded with the same color, for example, the ith vertex on shape $\mathcal{X}$ corresponds to the jth vertex on shape $\mathcal{Y}$, then these two vertices are assigned the same color on both shapes. The alignment results of shapes by the registration method are also presented in Figure 8. It can be seen from Figure 8 (a) that in the shape registration approach, it attempts to minimize the distance between the shapes and aims to find an optimal spatial transformation to transform two models as close as possible. However, a close alignment can only represent the rough spatial similarity and cannot represent the corresponding vertices relationship between two models. Thus, the shape registration method cannot reflect the deformation in deformed shapes. On the other hand, the shape correspondence method can find vertex-wise correspondence as in Figure 8 (b). Based on such vertex-wise relationship, one can easily map the information on one model to the other, which is much more utilizable for deformed shapes analysis, especially for a large number of shapes in the application of mass customization.

any two shapes can form a correspondence relation. Therefore, there are $C_{100}^2 = 4950$ correspondence shapes in total for the training dataset. Hence the set of models includes a variety of near-isometric deformations in the same model category. Each teeth aligner model has 9202 vertices on the shape of the mesh, and the vertex-wise ground truth correspondence, i.e., the vertex-to-vertex correspondence is already known between all of the shapes among the dataset. The CNN is implemented in Theano [40]. The ADAM stochastic optimization algorithm [41] is used with initial learning rate of 10^{-3}, $\beta_1 = 0.9$, $\beta_2 = 0.999$, and the dropout probability is 0.5. The input of each vertex in the network uses a local SHOT descriptor with 544 dimensions [42]. The output is a soft correspondence matrix, which can be interpreted as the probability of the vertex corresponded to each vertex on the reference shape, and the loss function is shown in Equation (11) for network training. Typically the training time on the teeth aligner shapes is approximately 40 seconds for one epoch. Forward propagation of the trained model takes approximately 0.5 seconds to produce the dense, soft correspondence for all the vertices.

5.1 Correspondence learning results

A suitable learning-based method should have a good learning ability in which the trained model can represent the intrinsic statistical properties of the training data and also can fit well the new data. The learning performance of the proposed Non-Euclidean CNN for mesh data is studied in the first experiment to investigate the effectiveness of the proposed method. In this experiment, for each vertex on the query shape, the output of the network is a soft correspondence with 9202-dimensional vector, which was then converted to the vertex correspondence. Since the correspondence is in pairs, i.e., the shapes of two models form a correspondence relation. Here we use the correspondences of first 80 models for training, there are $C_{80}^2 = 3160$ correspondence shapes in total in the training dataset.

Inspired by [31], the network structure in this experiment is set as FC64 + IC64 + IC128 + IC256 + FC1024 +

The shape registration method is used as a comparison to demonstrate the effectiveness of the proposed method. For the registration method, the classical iterative closest vertex (ICP) algorithm is applied in the experiment. Figure 9 shows the comparison results of the Non-Euclidean CNN and the registration method for shape matching. The protocol in [43] is applied to plot the percentages of correct correspondence matches under at most r-geodesically distant from the ground-truth correspondence on the reference shape. In this protocol, when the network predicts a correspondence of one vertex to its corresponding vertex on the other shape, we compute the geodesic distance between this predicted vertex

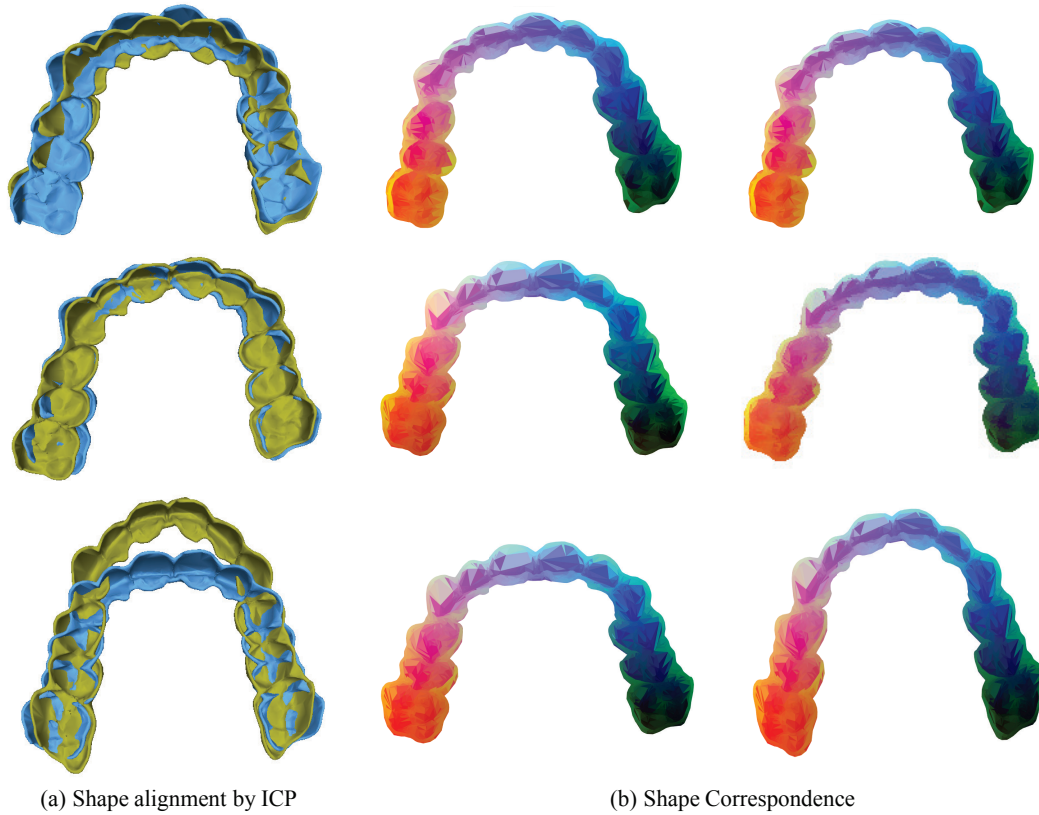(a) Shape alignment by ICP   (b) Shape Correspondence

Fig. 8. Comparison between shape alignment and shape correspondence. The level of correspondence is coded with color, i.e., the same color on the two models represent the corresponded vertices, and the prediction time for a correspondence between two shapes is approximately $0.5sec$.

and the ground-truth corresponding vertex. If this distance d is smaller than or equal to a predefined threshold of r, i.e., $d \leq r$, we consider the vertex is correctly corresponded. The threshold value of r can be determined according to the practical quality requirement.
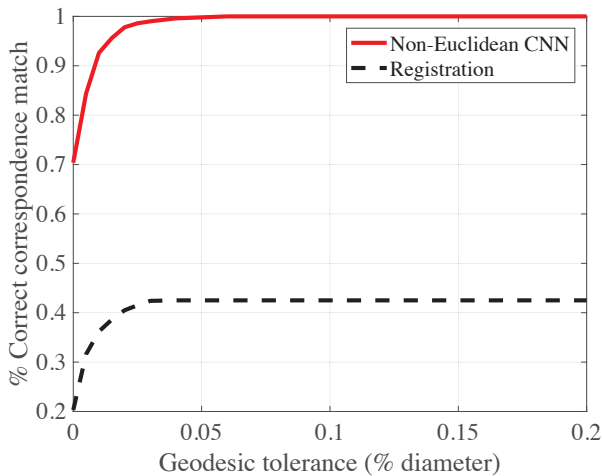

Fig. 9. Shape matching accuracy with geometric deep learning method and registration method on teeth aligner models.

It can be seen from Figure 9 that the performance of the proposed geometric deep learning method is much better than the registration method for shape correspondence matching. It can be seen that when the threshold geodesic distance is 5% of the diameter of teeth aligner model (3.11mm), the correspondence of models in testing achieves a high accuracy of 99% correct matching to the ground-truth while the registration method can only find approximately 40% of the correct correspondence. The main reason is that the registration method can only find a spatial alignment between shapes, which cannot represent the variation of the deformations among different shapes. On the contrary, the geometric deep leaning method learns the vertex local features and matches these features under different degrees of deformations on the model. Besides, from this experiment, it can be seen the correspondence method is more suitable for shape matching among non-rigid deformations, since it can find a vertex-wise correspondence between models, and such a correspondent relationship between models can be further utilized for deformation analysis and topology comparison and reconstruction. Thanks to this characteristic of the proposed method, it can be easily applied to shape matching in mass customization.

Figure 10 shows three sample models of predicted shape correspondence by the trained network. Three randomly chosen models are matched to a reference model. The trained

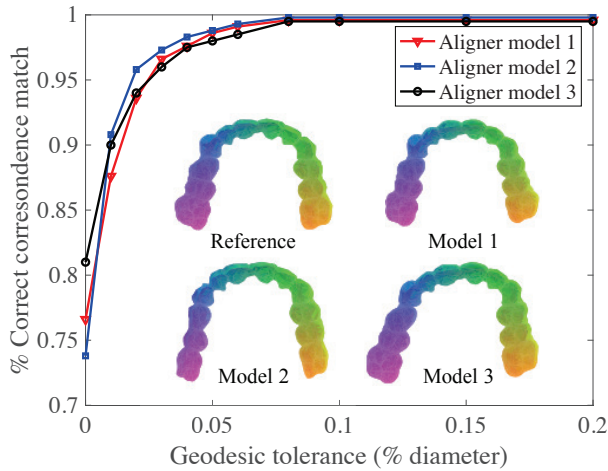9   Copyright © by ASME

Fig. 10. Examples of shape correspondence predicted via the proposed geometric deep learning method. The level of correspondence is coded with color, i.e., the same color on the two models represents the corresponded vertices. For example, the ith vertex on shape $\mathcal{X}$ corresponds to jth vertex on shape $\mathcal{Y}$, and then these two vertices have the same color on both shapes.

network can predict the vertex-wise correspondences, and the matched results of different aligner models are coded with colors as presented in the figure, among which the same color on the two models represents the corresponding vertices. From the results, it can be seen that the trained network can find a well-matched vertex correspondence between the selected model and the reference model. This experiment reveals that the proposed learning method is effective for shape correspondence matching, especially for models with deformations.

5.2 Robustness to extreme cases

It is desired that a correspondence method is robust and stable, however, due to the limitation of the scanning resolution and the reliability of the digital data transfer and processing, the digital shapes always suffer from information missing, resulting in incomplete models. To validate the robustness of the proposed method and test its performance on the incomplete models, in this section, we use the trained network to predict the correspondence of the incomplete models to a complete reference model.

In the experiment, two incomplete models are used, as shown in Figure 11. In these two models, the first one (Case 1) has a small hole, while the second one (Case 2) only has a portion of the original model. Then we attempt to match these two incomplete models to a randomly selected complete reference model in the database. The color-coded results are represented in Figure 11. It can be seen that the proposed method can predict well-matched correspondences for the two incomplete models (Case 1 and Case 2) to the reference model. It indicates that the proposed method can predict the correspondence of the incomplete model to the complete model. It also reveals that the network can learn the underlying features of the 3D model to predict the correspon-

dence which does not rely on the completeness of the mesh data. This is mainly because the network is trained on the correspondence directly based on the intrinsic shape descriptor (input SHOT descriptor) of the vertex on the shape and output a vertex-to-vertex relation. This experiment demonstrated that the proposed geometric deep learning method is effective and robust to extreme cases such as predicting shape correspondence of the incomplete models to a reference shape.

Table 1. Prediction results for incomplete models

	Number of Vertices	Prediction accuracy	Time(s)
Case 1	8756	91.2%	0.4
Case 2	6171	89.6%	0.3

Table 1 shows the results of using the trained model from Section 5.1 to predict the correspondence of the above two models. It can be seen that the prediction process is fast by propagating the trained model, which only takes 0.3-0.4 *sec*. It is worthwhile to mention that the low computation cost does not sacrifice the accuracy of the prediction, specifically both cases achieved around 90% of the ground-truth correspondence within 2% of the model diameter. The high efficiency and accuracy demonstrate that the proposed geometric deep learning method is robust and resilient to extreme cases, which enables broader practical applications such as those with severe data noises.

5.3 Application in mass customization

Based on the experiments discussed in Sections 5.1 and 5.2 that the geometric deep learning method can learn the intrinsic variety of deformation among a collected set of deformed shapes. The correspondence can be efficiently predicted through the trained CNN. The proposed method is particularly suitable for mass customization applications as the trained network takes only $0.5s$ to predict a full vertex-to-vertex correspondence of two shapes. In mass customization, we need to process a large number of deformed yet similar shapes. In this section, we will study a practical application of the proposed geometric deep learning method for mass customization in the orthodontics industry.

One common practice in the orthodontics industry is that the dentist needs to manually choose several landmark vertices on the patient's teeth model. When a new patient's teeth model (or aligner model) arrives, the dentist needs to select several landmark vertices on this new model manually, then, according to these selected vertices, matches the new model to the template (or previous) model. Furthermore, the selected landmarks are mapped to a reference model to determine a suitable alignment treatment strategy. This process is manually operated and mainly based on the experience of the dentist. The time spent on such manually marking tasks could be 10 minutes to 2 hours and without guarantee of finding the perfect matching to the reference model [7].
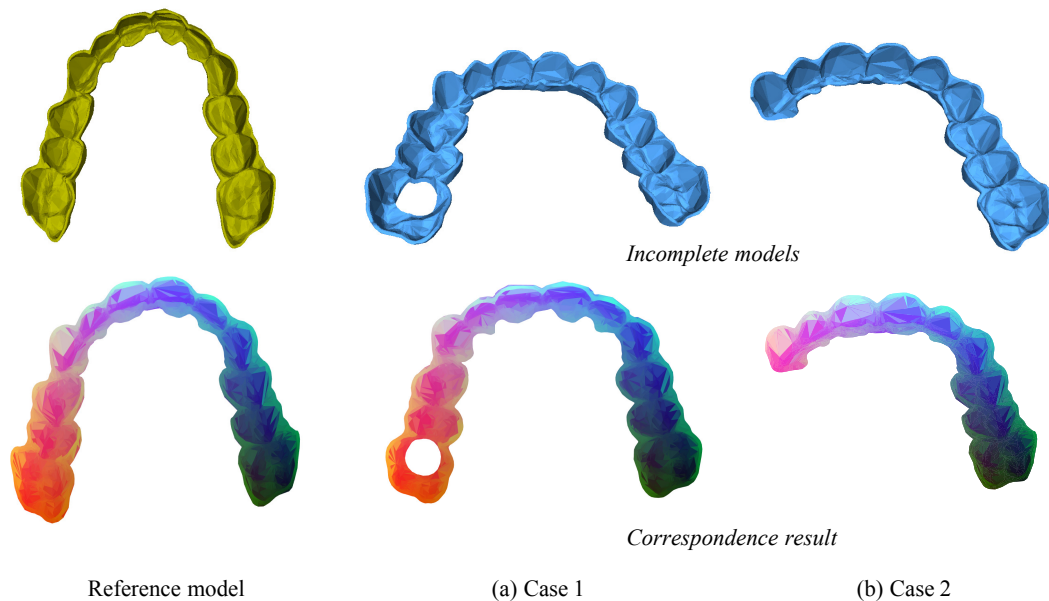
Copyright © by ASME

Fig. 11. Correspondence result on two incomplete models: the top row shows the reference model and two incomplete models, the bottom row shows the correspondence result. The level of correspondence is coded with color, i.e., the same color on the two models represents the corresponded vertices.

Because of the effectiveness and robustness of the geometric deep learning method, it can be used for automatically identifying the shape correspondences. Hence, through this shape correspondence, the dentist can identify all of the vertex-to-vertex relations on the two shapes and does not need to select the landmarks to find a mapping manually. From the previous experiments, when using the proposed geometric deep learning methods, it takes approximately $0.5s$ to predict a soft correspondence to a reference model for a given model with 9.2K vertices. Table 2 shows the prediction time for generating the full correspondence of a new model based on the trained network. Assuming there are a batch (1000) of teeth aligner models, and they all need to be marked and matched to the reference model. The total time for manual marking would be at least $1000 \times 0.167 = 167h$. However, with a trained network, the forward propagation for prediction only needs $1000 \times 0.5 = 500s$. This can significantly reduce the landmark marking and mapping time for massive models.

Table 2. Time comparison of different correspondence methods

Model	Vertices	Manually marking(min)	Time(s)
Teeth	127189	∼90	∼2
Aligner	9202	∼30	∼0.5

It is worth to remark that our method not only produces a correspondence of all vertices on the model but also output a soft-correspondence matrix. Indeed, our method can predict a vector for each vertex, i.e., each vector element representing the probability of the vertex corresponding to all of the vertices on the reference model. According to this information, we can output several optional vertices for dentists rather than only one according to the ranking of the probability of the reference model. This would provide more choices for the dentist to select the desired landmark. Based on the above analysis, it can be seen that the proposed geometric deep learning is excessively suitable for the orthodontics industry and can provide an efficient tool for mass customization applications. Furthermore, the proposed method can be used for the geometry integrity and quality investigation, for example, we can use the method to predict a shape correspondence between two shapes, then based on this vertex correspondence relation to measure the deformation of the vertices. In particular, we can determine whether the critical vertices on the shape are deformed within an acceptable distance.

6 Conclusions

The movement towards mass customization poses significant challenges to product design and manufacturing. 3D printing is becoming more and more mature to fulfill the mass customization. The product's geometric integrity is essential to guarantee the proper product design and manufacturing. To investigate the geometric integrity, the shape matching is the pillar-stone, where researchers propose various rigid or non-rigid body matching algorithms. However, these algorithms do not address the deformation problem. In this paper, we extend the conventional shape matching problem to shape correspondence problem, which includes the larger size of manifold correspondence, to extract the intrinsic deformations. A geometric deep learning method is introduced to learn the correspondence relation among the

11

models. The experimental results show the effectiveness and robustness of the proposed method.

This work is a pioneering work for correspondence based geometric integrity investigation. In the future, several directions will be explored. First, quantifiable assessment of the design and manufacturing after learning the correspondence would be studied. Second, how to get interpretable and semantics results for dentists/practitioners to understand the meaning of correspondence results will be explored. Third, incorporation of dentists/practitioners' knowledge in deep learning will be studied.

Acknowledgements

References

[1] Pine, B. J., 1993. *Mass customization: the new frontier in business competition*. Harvard Business Press.

[2] Bourell, D. L., Leu, M. C., and Rosen, D. W., 2009. Roadmap for additive manufacturing: identifying the future of freeform processing. Tech. rep.

[3] Gibson, I., Rosen, D. W., and Stucker, B., 2010. *Additive Manufacturing Technologies Rapid Prototyping to Direct Digital Manufacturing*, 1st ed. Springer, New York, NY.

[4] Invisalign Inc. http://www.invisalign.com/.

[5] Kwok, T.-H., Ye, H., Chen, Y., Zhou, C., and Xu, W., 2017. "Mass customization: Reuse of digital slicing for additive manufacturing". *Journal of Computing and Information Science in Engineering,* **17**(2), p. 021009.

[6] Ye, H., Kwok, T.-H., Zhou, C., and Xu, W., 2018. "Information reuse to accelerate customized product slicing for additive manufacturing". In ASME 2018 International Design Engineering Technical Conferences and Computers and Information in Engineering Conference, pp. V01AT02A050–V01AT02A050.

[7] http://www.youtube.com/watch?v=L2JjNk8JgUk.

[8] Siemens. http://www.siemens.com/.

[9] Envisiontec Inc. http://envisiontec.com/.

[10] Digital Forming. https://home.digitalforming.com/.

[11] Tam, G. K. L., Cheng, Z., Lai, Y., Langbein, F. C., Liu, Y., Marshall, D., Martin, R. R., Sun, X., and Rosin, P. L., 2013. "Registration of 3d point clouds and meshes: A survey from rigid to nonrigid". *IEEE Transactions on Visualization and Computer Graphics,* **19**(7), July, pp. 1199–1217.

[12] Huang, J., Kwok, T.-H., and Zhou, C., 2017. "V4pcs: Volumetric 4pcs algorithm for global registration". *Journal of Mechanical Design,* **139**(11), p. 111403.

[13] Tootooni, M. S., Dsouza, A., Donovan, R., Rao, P. K.,

[14] Kong, Z. J., and Borgesen, P., 2017. "Classifying the dimensional variation in additive manufactured parts from laser-scanned three-dimensional point cloud data using machine learning approaches". *Journal of Manufacturing Science and Engineering,* **139**(9), p. 091005.

[14] Camelio, J., Hu, S. J., and Ceglarek, D., 2003. "Modeling variation propagation of multi-station assembly systems with compliant parts". *Journal of Mechanical Design,* **125**(4), pp. 673–681.

[15] Yue, X., Wen, Y., Hunt, J. H., and Shi, J., 2018. "Surrogate model-based control considering uncertainties for composite fuselage assembly". *Journal of Manufacturing Science and Engineering,* **140**(4), p. 041017.

[16] Zhou, L., Wang, H., Berry, C., Weng, X., and Hu, S. J., 2012. "Functional morphing in multistage manufacturing and its applications in high-definition metrology-based process control". *IEEE Transactions on Automation Science and Engineering,* **9**(1), pp. 124–136.

[17] Mémoli, F., and Sapiro, G., 2005. "A theoretical and computational framework for isometry invariant recognition of point cloud data". *Foundations of Computational Mathematics,* **5**(3), Jul, pp. 313–347.

[18] Chen, Q., and Koltun, V., 2015. "Robust nonrigid registration by convex optimization". In 2015 IEEE International Conference on Computer Vision (ICCV), pp. 2039–2047.

[19] Rodol, E., Bronstein, A. M., Albarelli, A., Bergamasco, F., and Torsello, A., 2012. "A game-theoretic approach to deformable shape matching". In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 182–189.

[20] Solomon, J., Nguyen, A., Butscher, A., Ben-Chen, M., and Guibas, L., 2012. "Soft maps between surfaces". *Computer Graphics Forum,* **31**(5), pp. 1617–1626.

[21] Ovsjanikov, M., Ben-Chen, M., Solomon, J., Butscher, A., and Guibas, L., 2012. "Functional maps: A flexible representation of maps between shapes". *ACM Trans. Graph.,* **31**(4), July, pp. 30:1–30:11.

[22] Kovnatsky, A., Bronstein, M. M., Bresson, X., and Vandergheynst, P., 2015. "Functional correspondence by matrix completion". In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 905–914.

[23] Imani, F., Chen, R., Diewald, E., Reutzel, E., and Yang, H., 2019. "Deep learning of variant geometry in layer-wise imaging profiles for additive manufacturing quality control". *Journal of Manufacturing Science and Engineering,* **141**(11).

[24] Bastani, K., Barazandeh, B., and Kong, Z. J., 2018. "Fault diagnosis in multistation assembly systems using spatially correlated bayesian learning algorithm". *Journal of Manufacturing Science and Engineering,* **140**(3).

[25] Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., and Xiao, J., 2015. "3d shapenets: A deep representation for volumetric shapes". In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1912–1920.

[26] Brock, A., Lim, T., Ritchie, J. M., and Weston, N., 2016. "Generative and discriminative voxel modeling with convolutional neural networks". *arXiv preprint arXiv:1608.04236*.

[27] Balu, A., Lore, K. G., Young, G., Krishnamurthy, A., and Sarkar, S., 2016. "A deep 3d convolutional neural network based design for manufacturability framework". *arXiv preprint arXiv:612.02141*.

[28] Qi, C. R., Su, H., Mo, K., and Guibas, L. J., 2016. "Pointnet: Deep learning on point sets for 3d classification and segmentation". *arXiv preprint arXiv: 1612.00593*.

[29] Bronstein, M. M., Bruna, J., LeCun, Y., Szlam, A., and Vandergheynst, P., 2016. "Geometric deep learning: going beyond euclidean data". *arXiv preprint arXiv:1611.08097*.

[30] Masci, J., Boscaini, D., Bronstein, M. M., and Vandergheynst, P., 2015. "Geodesic convolutional neural networks on riemannian manifolds". In Proceedings of the 2015 IEEE International Conference on Computer Vision Workshop (ICCVW), ICCVW '15, IEEE Computer Society, pp. 832–840.

[31] Boscaini, D., Masci, J., Rodoià, E., and Bronstein, M., 2016. "Learning shape correspondence with anisotropic convolutional neural networks". In Proceedings of the 30th International Conference on Neural Information Processing Systems, NIPS'16, Curran Associates Inc., pp. 3197–3205.

[32] Monti, F., Boscaini, D., Masci, J., Rodolà, E., Svoboda, J., and Bronstein, M. M., 2016. "Geometric deep learning on graphs and manifolds using mixture model cnns". *arXiv preprint arXiv:1611.08402*.

[33] Lähner, Z., Vestner, M., Boyarski, A., Litany, O., Slossberg, R., Remez, T., Rodolà, E., Bronstein, A., Bronstein, M., Kimmel, R., et al., 2017. "Efficient deformable shape correspondence via kernel matching". *arXiv preprint arXiv:1707.08991*.

[34] Kokkinos, I., Bronstein, M. M., Litman, R., and Bronstein, A. M., 2012. "Intrinsic shape context descriptors for deformable shapes". In 2012 IEEE Conference on Computer Vision and Pattern Recognition, pp. 159–166.

[35] Kimmel, R., and Sethian, J. A., 1998. "Computing geodesic paths on manifolds". *Proceedings of the National Academy of Sciences,* **95**(15), pp. 8431–8435.

[36] Nair, V., and Hinton, G. E., 2010. "Rectified linear units improve restricted boltzmann machines". In Proceedings of the 27th International Conference on International Conference on Machine Learning, ICML'10.

[37] Glorot, X., Bordes, A., and Bengio, Y., 2011. "Deep sparse rectifier neural networks". In Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, Vol. 15 of *Proceedings of Machine Learning Research*, PMLR, pp. 315–323.

[38] Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. R., 2012. "Improving neural networks by preventing co-adaptation of feature detectors". *arXiv preprint arXiv:1207.0580*.

[39] Ioffe, S., and Szegedy, C., 2015. "Batch normalization: Accelerating deep network training by reducing internal covariate shift". *arXiv preprint arXiv:1502.03167*.

[40] Bergstra, J., Breuleux, O., Bastien, F., Lamblin, P., Pascanu, R., Desjardins, G., Turian, J. P., Warde-Farley, D., and Bengio, Y., 2010. "Theano: A cpu and gpu math compiler in python".

[41] Kingma, D. P., and Ba, J., 2014. "Adam: A method for stochastic optimization". *arXiv preprint arXiv:1412.6980*.

[42] Salti, S., Tombari, F., and Stefano, L. D., 2014. "Shot: Unique signatures of histograms for surface and texture description". *Computer Vision and Image Understanding,* **125**, pp. 251 – 264.

[43] Kim, V. G., Lipman, Y., and Funkhouser, T., 2011. "Blended intrinsic maps". *ACM Trans. Graph.,* **30**(4), July, pp. 79:1–79:12.