MC-Net: Mesh Completion for Dental Scans

Olivier Lessard, *Member, IEEE*, François Guibault, *Member, IEEE*, Julia Keren and Farida Cheriet, *Senior Member, IEEE*

***Abstract*— Dental centers need to design hundreds of dental crowns per year using computer assisted design (CAD) software. Typically, technicians manually modify a template shape to create crowns. That process requires a lot of time and experience to be done correctly, which leads to great variability in quality. In recent years, many deep learning methods have been proposed to do point cloud completion by predicting only the missing region. Although these methods are potentially applicable to the task of dental crown design, most of them fail to generate smooth point clouds, which is critical for surface reconstruction. In this paper, we propose an end-to-end approach called MC-Net for automatic mesh completion of dental scans. Using an input point cloud sampled at multiple resolutions and a template shape for the type of tooth to generate, MC-Net extracts features to guide a mesh deformation. The mesh generation follows a coarse-to-fine strategy and uses a mesh-related loss function to make the procedure stable. Our model can generate visually correct and accurate surfaces of the missing regions.**

***Index Terms*—3D shape generation, geometric deep learning, computer aided design, dental mesh completion.**

I. INTRODUCTION

IN dental centers, technicians need to manually design hundreds of crowns per year. Crown generation implies designing an external surface that is visible once the crown is installed on the tooth preparation. That design is challenging even with today's computer assisted design (CAD) software. The technician must design a tooth shape with a complex morphology and patient-specific characteristics. Usually, the dentist starts by making a preparation with the damaged tooth. The preparation serves as a foundation on which the dental crown will be installed. Secondly, the dentist scans the prepared tooth and the surrounding teeth with an intra-oral scanner to get a 3D representation. Finally, the technician uses that surrounding information to design a patient-specific crown.

Once the technician has the 3D scan of the patient's jaw, they will use a CAD system to create the crown shape. Modern systems have many dental application-specific tools to assist this process. However, considerable human intervention is still needed. Usually, the technician starts by picking a template tooth from a library and adjusting it through positioning and scaling to account for functional and esthetic constraints. Then, complex transformations may be applied to create contact points with the surrounding teeth. Thus, an automatic approach to generate patient-specific crowns with the desired positioning and scaling would be very useful. Also, an approach to create patient-specific teeth could help technicians to design more personalized crowns.

In this paper, we focus on the task of tooth completion. An example of tooth completion is shown in Figure 1. From this figure, we can see that tooth completion implies generating a missing shape to get a complete arch. The missing shape corresponds to a mesh surface similar to a crown's external surface. In both cases, the surface needs to be smooth, detailed, and patient-specific.

The completion problem has been addressed in the computer vision community since real world 3D data is often captured by scanners. Due to occlusions, light reflections and limitations of the scanners, 3D data such as point clouds are often incomplete. The task of point cloud completion consists in generating a complete point cloud from a partial observation. Deep Learning (DL) methods became popular because of their ability to complete shapes even with large missing regions. Since the advent of PointNet [1], many methods were proposed to achieve point cloud completion. Among them, Huang *et al.* proposed PF-Net [2]. Their architecture enables a multi-resolution representation of the completions and generates realistic missing regions. Contrary to most shape completion methods in the literature, missing shapes in tooth completion and dental crown design can be seen as complex shapes in and of themselves. Furthermore, in dental crown design, the technicians always know where they must be located. For this reason, we focus on tooth-specific completions.

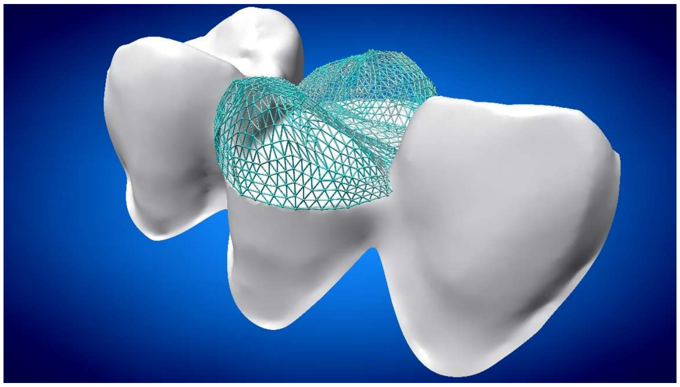


Fig. 1 Illustration of the tooth completion problem. Using a point cloud of the surrounding teeth, our model predicts the missing surface that the tooth should have, as shown by the mesh in cyan.

There have been a few studies on how to automate dental crown design and how to complete teeth. In two of these works [3], [4], the authors converted 3D dental scans to 2D depth images. Then, they used pix2pix-based approaches [5] to generate depth images of dental crowns. In this manner, they could benefit from the advances in image generation to achieve good performance. But their conversion from 3D to 2D necessarily involves information loss, especially on the sides of the teeth. By generating depth maps, these approaches focus on the crown's occlusal surface but neglect the lateral portions,

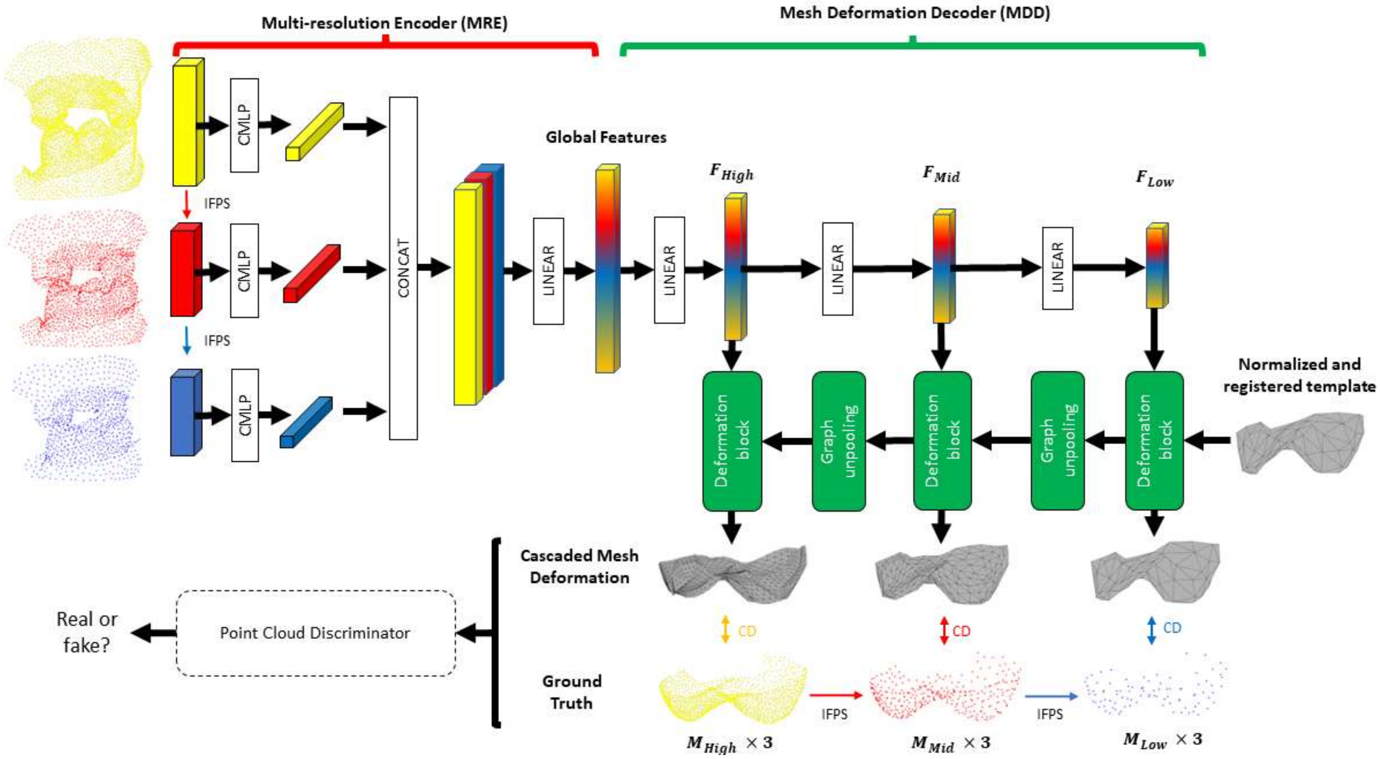


Fig. 2 Architecture of MC-Net. The multi-resolution encoder (MRE) analyzes the three input resolutions to create global features. The mesh deformation decoder (MDD) uses the features to deform the template with three deformation blocks. *(CMLP means Combined Multi-Layer Perceptron; IFPS is Iterative Farthest Point Sampling; CD is Chamfer Distance.)*

which are equally important to produce an appropriate dental crown. The 2D approaches do not generate complete tooth surfaces directly and obtaining the lateral surfaces from the occlusion surface is not a trivial task. To complete dental scans, Ping *et al.* proposed a voxel-based approach called SA-IFN [6]. Their approach uses implicit functions and a self-attention mechanism to capture nonlocal features to do shape completion. Their model can predict 3D voxels that represent the complete shapes, in contrast to previous methods. However, voxel-based approaches need a lot of unnecessary occupancy predictions, and this limits the output resolution. Dental technicians always work with high resolution scans. Thus, a scalable approach that can work at higher resolution would be preferred. In [7], the recently proposed PF-Net was used to complete teeth. The model can predict a variety of tooth shapes, but the predictions are sometimes wrongly positioned and noisy. The approach produces point clouds that cannot be directly used to reconstruct meshes. Zhu *et al.* [8] used a transformer-based model to complete teeth point clouds and apply a new voxel-based reconstruction. However, their method reconstructs a watertight mesh instead of a thin surface with a boundary. In addition, by generating point clouds rather than meshes including connectivity, their method has little control over the smoothness of the resulting shapes.

The present work is an extension of a preliminary conference publication [7] to achieve teeth completion. In practice, technicians use a preparation which is very important to guide their crown design. Similarly, in our proposed approach, we use the base of the partial tooth (referred to as a *sliced tooth*) to guide the tooth completion. The dataset to train our DL model now contains sliced teeth, which improves the positioning of the predictions. In addition, this paper presents 1) a new architecture using a formulation of mesh deformation that generates meshes instead of point clouds; 2) a comparison with the original PF-Net for mesh generation; and 3) an ablation study to evaluate the effectiveness of the key components of our model.

The contributions of this work are twofold: I) We propose to directly learn high-level geometric features of the dental scans for automatic tooth completion; II) We propose an end-to-end DL method called Mesh Completion Network (MC-Net).

MC-Net can be seen as an extension of PF-Net. First, instead of learning directly a low resolution representation of the missing region, the proposed network learns a deformation from a fixed template. The template introduces both mesh connectivity and dental shape characteristics. Second, we propose to replace the concatenation and addition mechanism of PF-Net by the edge-based graph unpooling proposed in Pixel2Mesh [9]. This allows us to add new points at better positions and maintain the uniformity of the meshes. Third, we add a Laplacian regularization [9] to control the coarse-to-fine mesh deformation. This ensures a smooth deformation and avoids creating intersecting triangles and spikes. Finally, graph convolutions and a new formulation of the discriminator input are used to enhance the shape generation.

The rest of the paper is organized as follows. Related works concerning point cloud completion, mesh completion and mesh deformation are briefly reviewed in Section II. The proposed method is explained in Section III. Section IV presents the experimental settings and results, including an ablation study. The limitations and future work are presented in Section V. Finally, Section VI concludes this paper.

II. Related Work

A. Point Cloud Completion

Point cloud completion is crucial in 3D computer vision because 3D data are often incomplete. The task consists in predicting the complete shape of an object from a partial observation. This problem has received increasing attention in the computer vision community since the pioneering works of PointNet and FoldingNet [1], [10], [11]. Many methods have been proposed to do point cloud completion. Some of them are based on generative adversarial networks (GAN) [12]. As mentioned in Fei *et al.* [13], these can be categorized into point-based, convolution-based, graph-based, folding-based, GAN-based, and transformer-based methods. Among these different architectures, some were proposed to predict only the missing region [2], [14]. For instance, the GAN-based architecture PF-Net uses a multi-resolution encoder and a point pyramid decoder to generate the missing points hierarchically. However, their resulting shapes lack geometric details. PF-Net does not use the connectivity between points, and thus the new points added in the upsampling mechanism have a large degree of freedom. Hence, the resulting point cloud does not represent a smooth surface, which makes the mesh reconstruction challenging. Recently, transformer-based approaches like PoinTr and SnowflakeNet [14], [15] have been proposed. These approaches require long training times and are memory-consuming, which is not ideal for practical applications in dentistry.

B. Mesh Completion

Similarly, mesh completion involves predicting a complete mesh from a partial observation. Mesh completion can be used for either small missing regions or large occlusions. For small missing regions like holes, geometric priors, self-similarity or patch encoding can be used as in [16], [17]. However, these methods work only for small missing regions. To complete larger occlusions, model-based approaches are used to capture the variability of a shape category. In [18], the authors propose an autoencoder to learn a latent space of complete shapes to achieve possible completions. Their method can complete meshes directly without any conversion or reconstruction, but the resulting meshes do not have the precision needed for dental applications. In [19], the authors propose a similar method to complete surfaces of the liver. They train a variational auto-encoder (VAE) on complete shapes and perform an optimization at inference time. The method takes as input a partial point cloud and outputs a complete mesh. However, the two latter methods require a time-consuming optimization at inference. This is not suitable in our case as we would like to reduce the overall time required for a dental crown design. Importantly, only the missing tooth region is interesting in our application. Existing mesh completion methods generate the complete shape, but extracting the missing region afterwards is not trivial. By generating the missing surface only, a model can focus on the details of that region and maximize its resolution.

C. Mesh Deformation

Deep learning has become a popular approach for mesh deformation because it can learn complex transformations that would be difficult to determine explicitly. Mesh deformation has been employed to reconstruct 3D meshes from images. Early works on mesh reconstruction used voxels to generate 3D shapes [20], [21]. However, these methods are memory consuming, which limits the resolution and constitutes an important limitation of this shape representation in practice. Instead, Fan *et al.* [22] propose to generate a point set from a single image. But due to the large number of degrees of freedom of point clouds, their predictions are noisy, which makes subsequent mesh reconstruction difficult. To overcome that problem, Wang *et al.* propose Pixel2Mesh to generate a 3D mesh directly [9]. They formulated the reconstruction problem as a mesh deformation problem and use mesh losses to control a deformation of a template mesh. Their template is a uniform general shape that provides an initial connectivity. The deformation is done with graph convolution networks (GCN) and a perceptual pooling that provides meaningful local features. They later extended their architecture to reconstruct a mesh from multiple views [23]. More recently, new methods have achieved better performance than Pixel2Mesh, especially on shapes with positive genus [24]–[26]. In this paper, we focus on the generation of shapes with genus 0 (i.e., closed surfaces with no holes) as in Pixel2Mesh. Chen *et al.* proposed MR-Net [27] to reconstruct cardiac meshes from sparse point clouds using mesh deformation. They propose to use voxels to map the point clouds to template vertices to guide the deformation which is done using GCN blocks. However, that voxel correspondence limits the quality of the deformations.

The method we present here takes inspiration from all three approaches seen above. Indeed, point cloud completion methods analyze a 3D input to generate a missing 3D output. They are interesting mainly because dental scans contain a lot of information and point cloud encoders can analyze such input efficiently with light algorithms. However, point cloud outputs cannot be directly used by technicians. For this reason, mesh completion also influenced our methodology because those approaches provide an output in the form of a mesh, as required in our problem. However, such methods are designed for hole filling or completion without high precision requirements. Because of the ability of mesh deformation methods to generate shapes that capture important details, our insight was to propose an architecture that analyzes an input point cloud to deform a template mesh into an output mesh. The details of the architecture will be discussed in the next section.

III. Method

Our model in an end-to-end deep learning framework that takes a partial point cloud as input and generates the missing region in a mesh format. The overview of our framework is illustrated in Figure 2. The network is a GAN-based model [12] like PF-Net but with a new decoder. It includes a generator composed of a Multi-Resolution Encoder (MRE) and a Mesh Deformation Decoder (MDD), and a discriminator network. The encoder network is a point-based network that analyzes three resolutions of the partial point cloud to create global features. With the global features, the decoder progressively deforms a template shape into the desired 3D tooth. Unlike PF-Net, the decoder predicts a 3D mesh instead of a 3D point cloud. The cascaded mesh deformation network uses deformation blocks and two graph-based unpoolings. Each deformation block takes an input mesh and predicts the new coordinates of each vertex using the global features. The graph unpooling layer increases the number of vertices while maintaining the initial connectivity. Starting from a low-resolution template, the model learns to gradually deform and add details to the mesh in a coarse-to-fine strategy. To control the deformation and ensure the smoothness of the output, a Laplacian regularization is used in addition to Chamfer Distance (CD) in the learning process.

A. Initial surface

The model always deforms a tooth-specific template to do patient-specific completions. The template is a downsampled mesh with roughly 90 vertices and 160 edges that has been normalized and registered with the input point cloud. The oriented bounding boxes of neighboring teeth are used to register and scale the template as shown in Figure 3. More specifically, a manual teeth segmentation performed by a technician allows us to create bounding boxes around each tooth. A box is then defined to position and scale the template points. The box has approximately the same width and depth as that of the sliced tooth. Its height is defined by the partial tooth boundary and the adjacent teeth cusps. This box-based registration is only an initial positioning and scaling that is refined by the network. In practice, automatic teeth segmentation [28], [29] can provide the labels to define the bounding boxes.

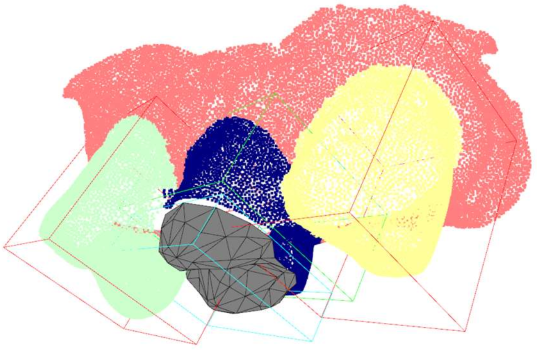


Fig. 3 The template mesh (in gray) is positioned and scaled to fit inside a bounding box defined by the segmented input.

B. Mesh deformation block

Each mesh deformation block contains a linear layer, a 1D convolution layer and a graph convolution layer (see Figure 4). Graph convolutions in the deformations allow sharing of information between neighboring points, which is useful to predict appropriate displacements. In [9], the authors used 14 graph convolution layers in the deformation blocks to deform an ellipsoid into an unknown shape. However, we decided to use a single layer because the model deforms a template that already has general characteristics corresponding to those of the missing tooth. With an input mesh and some learned features, each deformation block learns displacements for each block's input vertices. The first deformation block takes the template with M_{Low} vertices and predicts the $M_{Low} \times 3$ new coordinates using the F_{Low} features. The first intermediate mesh is then up sampled to have M_{Mid} vertices. The second deformation block takes the new upsampled mesh and predicts the $M_{Mid} \times 3$ new coordinates with the F_{Mid} features. Finally, the mesh is up sampled again, and the last deformation block predicts the high-resolution mesh with $M_{High} \times 3$ coordinates using the F_{High} features. With this coarse-to-fine deformation, the network can apply small deformations at the beginning that affect many high-resolutions vertices at once. The network thus learns to move the template vertices to the most representative locations and add details in further deformations.

C. Graph unpooling layer

The goal of the graph unpooling layers is to add new vertices in the mesh while preserving the initial connectivity. We followed the edge-based graph unpooling of Pixel2Mesh [9] to divide each triangle into 4 smaller triangles. This allows us to use two points to add new vertices instead of one like in PF-Net. This way, points are added at optimal positions and the upsampling can be very efficient since there are more edges than points. Furthermore, the edge-based unpooling maintains regular vertex degrees.

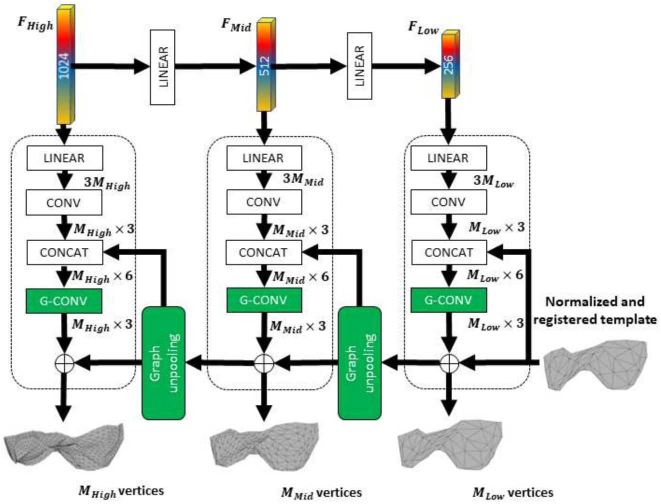


Fig. 4 Details of the Mesh Deformation Decoder. MDD creates an output mesh along with 2 intermediate results. The graph unpooling layers preserve the connectivity during the coarse-to-fine generation. The plus circles represent add operations. Layers operating on points are shown in while; those operating on graphs are shown in green.

D. Discriminator Input

Originally in PF-Net, the discriminator receives as input the real shapes and the generated ones. It classifies them as real or fake regardless of any surrounding context. That formulation is not ideal here because the model may be able to generate shapes that look realistic by themselves but are not compatible with the full input geometry. Since the input contains many more points than the output, we propose to concatenate the sliced tooth from the input point cloud and the generated shape, as shown in Figure 5. This way, the discriminator's input is a complete tooth, which should be smooth everywhere and not have a gap at the intersection of the known and missing regions. A binary feature is added to let the discriminator know which are the input's points versus the predicted ones.

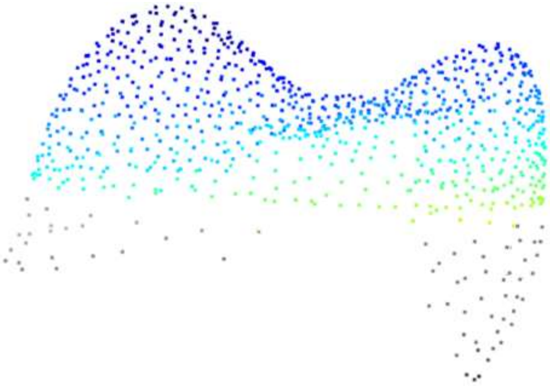


Fig. 5 An example of the discriminator's input. The points in gray in the lower portion are sampled from the input using teeth segmentation and iterative farthest point sampling (IFPS).

E. Loss function

We use three different losses to constrain the output shape and control the coarse-to-fine deformation. We use the Chamfer Distance (CD) loss [22] to constrain the vertex locations, a Laplacian regularization to maintain relative locations between neighboring vertices during the deformations, and an adversarial loss. All the losses except the adversarial loss are applied at each resolution.

The CD measures the mean squared distance between two point clouds S_1 and S_2. The measurements are done between each point and its closest point in the other point set as described in equation (1):

$$d_{CD}(S_1,S_2)=\frac{1}{S_1}\sum_{x\in S_1}\min_{y\in S_2}\|x-y\|_2^2 \quad + \\ \frac{1}{S_2}\sum_{y\in S_2}\min_{x\in S_1}\|y-x\|_2^2, \tag{1}$$

where one term measures the distances from the prediction to the ground truth and the other measures the distances in the opposite direction. The completion loss uses the CD measurement with each output resolution and specific weights as described in equation (2):

$$l_{comp}=d_{CD}\left(Y_{High},Y_{GT}\right)+\alpha\cdot d_{CD}\left(Y_{Med},Y'_{GT}\right) \\ +\beta\cdot d_{CD}\left(Y_{Low},Y''_{GT}\right), \tag{2}$$

where α and β are weights and Y_{GT}, Y'_{GT}, Y''_{GT} are the different resolutions of the ground truth point clouds obtained by iterative farthest point sampling (IFPS). l_{comp} is efficient to predict reasonably good vertex coordinates; however, it does not capture the smoothness and visual appearance of the shapes.

To preserve the template's smoothness, we propose a Laplacian regularization. This term prevents vertices from moving too freely, which can cause intersecting triangles and degrade the template characteristics. It encourages neighboring vertices to have similar movements. The Laplacian loss compares the relative positions of one point to its neighbors before and after the deformation. Each relative position is calculated using equation (3):

$$\delta_p=p-\sum_{k\in N(p)}\frac{1}{\|N(p)\|}k, \tag{3}$$

where p is a vertex in the predicted mesh and $N(p)$ are the neighboring vertices of p. The Laplacian loss is defined in equation (4):

$$l_{Lap}=\sum_p\left\|\delta'_p-\delta_p\right\|_2^2, \tag{4}$$

where $\delta'_p$and δ_p are the relative positions before and after the deformation.

From the set of partial input point clouds X, the generator predicts shapes that should look like the missing point clouds in Y. The final predicted shapes and the ground truth are given to the discriminator network. The discriminator's goal is to classify its input shapes as real or fake. Both the generator and the discriminator are trained together with a GAN-based loss function [12]. The adversarial loss is formulated in equation (5):

$$L_{adv}=\sum_{1\leq i\leq S}\log\left(D(y_i)\right) \\ +\sum_{1\leq j\leq S}\log\left(1-D(G(x_j))\right), \tag{5}$$

where a partial input is represented by $x_i\in X$, a ground truth point cloud is represented by $y_i\in Y$ and S is the size of the dataset.

The final loss function is described by equation (6).

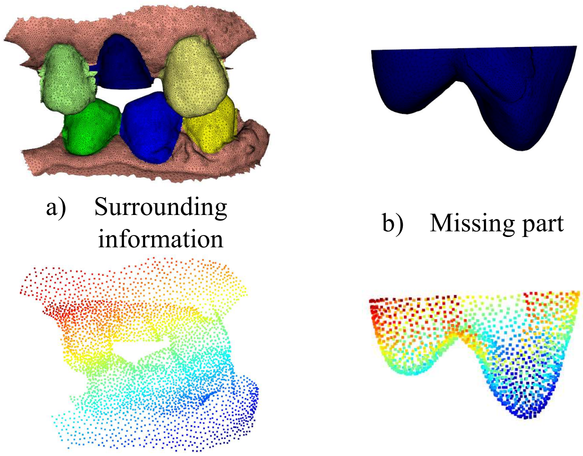
$$L=\ \lambda_{comp}l_{comp}+\ \lambda_{lap}l_{Lap}+\lambda_{adv}l_{adv} \tag{6}$$

where λ_{comp}, λ_{lap} and λ_{adv} are weights set to 0.95, 0.03 and 0.05.

IV. Experiments

A. Dataset and Preprocessing

We used different datasets for each tooth position, each one associated with a tooth-specific template. In total, we used four datasets to generate central incisors, canines, first premolars and first molars. To create each dataset, the full dental arches were first registered to a reference dental arch using oriented bounding boxes. The top part of the tooth to generate was removed using a planar slice. Afterwards, the 3D data was filtered to remove any structures that were not close to the sliced tooth. This step created a partial input that contains only partial tooth and its immediate neighborhood. Each input contained the sliced tooth, two adjacent teeth, three antagonist teeth, and the surrounding gingiva (see Figure 6). Because of lateral symmetry, we only generated teeth from the left side of the mouth, which reduced variability in the data. The preprocessed meshes were then converted to point clouds to be analyzed by the encoder, as shown in the lower part of Figure 6.



a) Surrounding information

b) Missing part

Fig. 6 Illustration of an input-output pair. Half of the tooth to be completed is removed using a planar slice.

The input point clouds were first normalized using z-score normalization along each axis. Each of the four datasets contained roughly 40 teeth. The training, validation and test sets contained 28 teeth, 6 and 6 teeth. The training sets were augmented using small random translations, rotations, and scaling.

B. Implementation Details

The input point clouds were downsampled with IFPS to 4000, 2000, and 1000 points. The intermediate resolutions M_{Low} and M_{Mid} were set to 93 and 351 vertices and the final output resolution M_{High} to 1365 vertices. 100 input points were added to the 1365 points of the missing regions to get complete teeth to provide to the discriminator. The encoder and the discriminator were the same as in PF-Net [2]. The MRE predicted a 1920 feature vector. The displacements learned by the MDD were scaled down using a linear activation function with a slope of 0.01. This scaling ensured that, during training, the first deformations did not introduce distortions or artefacts that would be difficult to remove in later iterations.

We implemented our network on PyTorch. The generator and the discriminator were trained alternately using the ADAM optimizer with a learning rate of 0.001 and a batch size of 6. Batch normalization and RELU activation units were used in the encoder and the discriminator but only RELU was used in the MDD. The model was trained for 40 epochs, which took 15.25 hours using an NVIDIA GeForce RTX 3080 graphic card. During testing, our model took 4.98ms to infer a mesh with 1365 vertices.

C. Comparison with State Of The Art Methods

We evaluated our method against two baselines. The first one was the original architecture of PF-Net which is used to predict the vertex coordinates only. The second one was a slightly modified PF-Net called PF-Net*. This second baseline consists of a PF-Net with a point-based template used in the architecture to predict the low-resolution points similarly to MC-Net. The template is the same one as in MC-Net but without edges between the points. PF-Net* then learns low resolution relative coordinates instead of direct coordinates as in the original article [2]. That modification makes sure the model starts its coarse-to-fine generation with a smooth and uniform point cloud. The model still uses the concatenation and addition mechanism to add new points and to deform the point template. PF-Net* is interesting experimentally because it allows us to evaluate to what extent the model can keep the smoothness and uniformity of the template when working only with point clouds. All three architectures were trained using the same datasets and training methodology. To evaluate the methods, we used the CD between the predicted shapes and the ground truth shapes. For a fair comparison, all the methods have a final output resolution of 1365 points. However, their intermediate resolutions differ due to their respective upsampling mechanisms. The CD is a bidirectional measurement, including a prediction to ground truth error $Pred \rightarrow GT$ and a ground truth to prediction error $GT \rightarrow Pred. Pred \rightarrow GT$ indicates how different the prediction is from the ground truth, while $GT \rightarrow Pred$ indicates how much of the ground truth surface is covered by the predicted one.

1) Quantitative evaluations

Table I shows the results for the three architectures with different tooth position datasets. MC-Net's hyperparameters were determined by experiments on premolars only. Our method outperforms the two other networks for both the $GT \rightarrow Pred$ and $Pred \rightarrow GT$ errors, in the case of premolars. Our network can predict shapes that are closer to the ground truth and that cover more of the ground truth regions. The premolars generated by MC-Net have an average CD measure of $0.85 * 10^{-2}$. By de-normalizing each CD measurement and averaging them all, the mean CD corresponds to a physical distance of 0.22 millimeters. The results of PF-Net* are consistently better than those of PF-Net. For incisors, canines and molars, PF-Net* yields the lowest errors of all three methods. However, lower CD does not necessarily lead to a better mesh model. To demonstrate this, qualitative results are shown in the next section.

Table I Tooth completion results after 6-fold cross-validation. The reported metrics are [*Pred → GT / GT → Pred*]. Resulting normalized CD values are scaled by 100.

Category	PF-Net	PF-Net*	MC-Net
Incisor #11, #21	4.92/3.38	**2.41/1.83**	3.08/2.71
Canine #13, #23	2.99/1.94	**1.59/0.99**	1.66/ 1.32
Premolar #14, #24	2.71/1.85	1.08/0.90	**0.89/0.81**
Molar #16, #26	3.09/2.28	**1.22/1.02**	1.24/1.24
Mean	3.43/2.36	**1.58/1.19**	1.71/1.52

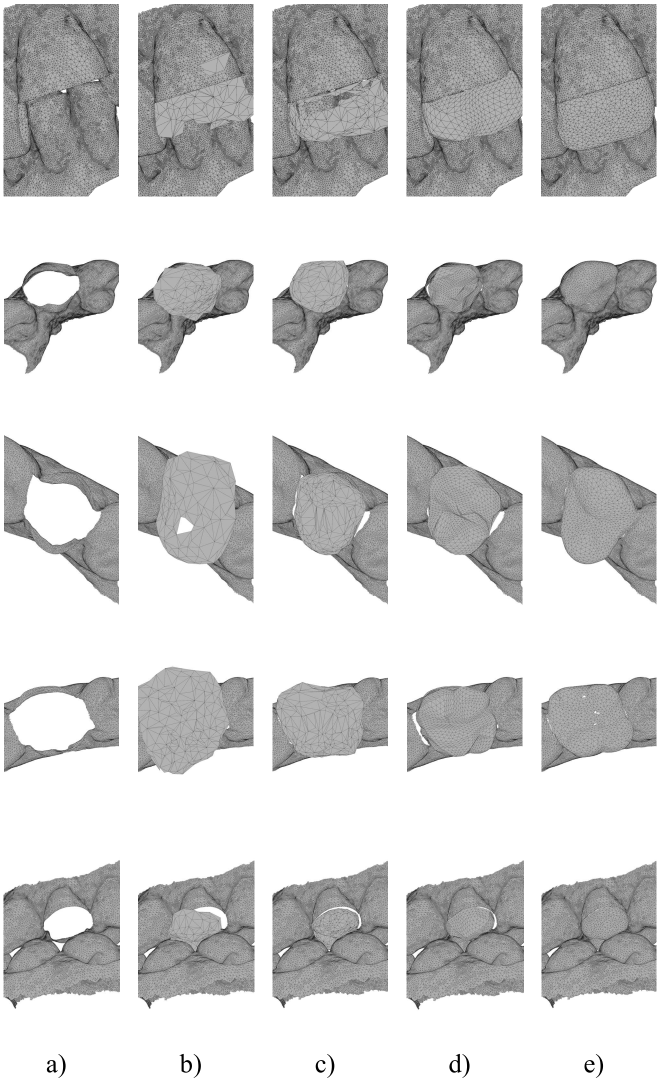


Fig. 7 Examples of mesh completions by the different architectures. **a)** partial input meshes used to create points clouds; **b)** generated teeth with PF-Net; **c)** generated teeth with PF-Net*; **d)** generated teeth with MC-Net; **e)** ground truth meshes. In all cases except **d)**, ball pivoting reconstruction was used to create meshes from point clouds. Holes in the ground truth and the prediction meshes are due to reconstruction artefacts.

2) Qualitative evaluation

As mentioned above, the CD metric does not tell us about the smoothness and visual appearance of the predicted shapes. Therefore, a qualitative evaluation is necessary to judge the correctness of the tooth completions. Figure 7 provides several examples of visual results from the three models, as surface meshes. The figure shows the partial input, the three predictions and the ground truth in each case. To convert the point cloud results into meshes, we used a standard algorithm (ball pivoting reconstruction) [30].

As we can see in Figure 7, the mesh reconstruction from PF-Net and PF-Net*'s point clouds is non-trivial. With PF-Net, the reconstructed meshes are sometimes wrongly positioned, which significantly affects the quality of the predictions. In contrast, with PF-Net*, the meshes are better positioned and contain more details. However, with both architectures, the final meshes contain spikes, intersecting triangles, and holes. These artefacts result from the large number of degrees of freedom when no regularization is used. The resulting noise in the point clouds degrades the quality of the reconstruction results.

Unlike the baselines, our model predicts meshes directly. Its components allow us to carefully deform the template into a smooth continuous surface. Indeed, as revealed in Figure 7, MC-Net's predictions are smoother, more uniform and detailed than the baseline shapes from PF-Net and PF-Net*.

D. Ablation Study

We performed an ablation study to evaluate the effectiveness of different components of MC-Net. The ablation study is done using the premolar dataset. The validation set was used to report results. We introduced a new metric called ICP-CD. It measures the CD after an Iterative Closest Point (ICP) registration of the prediction with respect to the ground truth. This metric evaluates the shape without any positional bias.

1) Effectiveness of the tooth template

Using a half sphere template instead of a tooth template implies learning a more challenging deformation. Indeed, the generated shapes starting from a half sphere are smoother and contain fewer curvatures. As seen in Figure 8, the model deforms the half sphere into teeth that lack grooves that are characteristic of normal teeth. This is problematic because the designed shape should have similar grooves as the other patient's teeth to be realistic. To get grooves using a half sphere template, the deformation blocks would need to be more complex, and the model would have to learn all the characteristics of dental shapes from scratch. As shown in Table II and Figure 8, our full model can generate correct teeth with simple deformation blocks because the tooth template provides important knowledge of the intended shape.

2) Effectiveness of Laplacian regularization

As shown in Figure 8, the Laplacian loss is important for the smoothness of the predictions. Without it, the generated shapes are much less realistic. The noise introduced causes overlaps with the surrounding teeth, which is not desirable. The noise also increases the CD as shown in Table II. The Laplacian regularization is hence a key component of MC-Net, ensuring that the predictions remain smooth.

3) Effectiveness of graph convolutions

As shown by the qualitative and quantitative results, the single graph convolution layer in each mesh deformation block contributes very little to the results. It changes the shape slightly, especially on the occlusion surface. The predictions using graph convolutions have smoother grooves compared to those without them. Indeed, with only one layer, a point receives only information from its direct neighbors. This helps the model to predict better displacements locally to avoid undesired surface folds. By adding more graph convolution layers, each point would receive information from a larger neighborhood, but this would increase the model complexity.

4) Effectiveness of the new discriminator input

According to Table II., the points describing the sliced tooth added to the discriminator's input help to reduce the CD. This result is not surprising because the predicted shape is supposed to form a realistic whole tooth when joined with the partial input. Therefore, the positions of the boundary points are an important aspect of the prediction. Using the new input formulation, the adversarial loss can provide a more meaningful gradient. The cost function implicitly encourages the predicted boundary points to be close to the partial input's boundary, i.e. the base of the tooth. As shown in Figure 8, the predictions have a better fit with the partial input when the new input formulation is used.

Table II Ablation study to evaluate the effect of different components of our MC-Net model. Each row reports the average error metrics when removing the corresponding component. Resulting normalized CD values are scaled by 100.

Experiment	CD	ICP-CD
- Tooth template	1.119	0.562
- Laplacian	0.971	0.421
- Graph convolution	**0.659**	0.302
- Discriminator input modification	0.670	0.307
Full model	**0.659**	**0.291**

V. Discussion

The improvements of PF-Net* over PF-Net as shown by the last row in Fig. 7 are mainly due to the positioning of the point cloud template. The initial positioning helps the model understand precisely where the predicted mesh should be located with respect to the partial input. Thus, the template positioning allows both PF-Net* and MC-Net to reduce the CD error metric. PF-Net* produces a point cloud with fewer constraints than MC-Net. This greater freedom allows the model to reduce the CD, but it does not improve the visual appearance of the meshes, as revealed in our qualitative evaluation (see Figure 7).

With PF-Net*, even if the provided point template is smooth and uniform, the decoder still introduces noise and artefacts that affect the CD and the smoothness. The template characteristics are lost and the model is not able to preserve its smoothness. This problem is due to the concatenation and addition mechanism that does not exploit connectivity information when adding new points. Therefore, new points in PF-Net* and PF-Net have a large degree of freedom, which creates noise. Since we want to complete a surface of a known shape category, the flexibility of the point cloud approach does not provide any advantage; instead, it creates a smoothness problem.

By contrast, MC-Net's predictions are very smooth. They benefit from the edge-based graph unpooling and the Laplacian regularization. The unpooling preserves the initial connectivity and maintains the smoothness during the upsampling. Meanwhile, the Laplacian regularization provides a meaningful gradient to generate smooth meshes with relevant details as demonstrated in the ablation study. That regularization reduces the noise and the CD at the same time. Its effect is more apparent in the qualitative results as shown in Figure 7. By finetuning the Laplacian regularization weight for each category of teeth, as we did for the premolars, MC-Net could provide optimal results for any tooth position.

We made several design choices with the aim of having more control over the mesh generation process and producing appropriate shapes for tooth completion. However, our method may produce meshes with incorrect boundaries. The boundary seems to be a difficult region for the model to understand. This is unexpected because the partial input contains a boundary that tells the network where the generated mesh must begin. However, our results indicate that that constraint is not explicit enough yet. This may be due to the standard CD formulation, in which all the points are equally important. A weighted CD giving more importance to the boundary points could help with this problem. A better template initialization could also be explored. Furthermore, a transformer-based encoder could help to capture dependencies between the boundary points at a broader scale.

Other improvements to our current approach could be considered. Among them, a mesh encoder could potentially provide better features to guide the shape generation than the currently used point encoder. Additionally, a functionality-aware loss function could help the predicted tooth to have better contact points with adjacent teeth. Finally, adding more graph convolution layers would allow information sharing between more distant points in the mesh generation.

MC-Net addresses the task of dental mesh completion, which is closely related but not identical to the task of dental crown generation. One difference is that tooth preparations have quite variable shapes, and the crown must fit “around” the base as opposed to extending a previously sliced tooth. Another difference lies in the crown itself, which must be a closed shape rather than the completion of one. On the other hand, real teeth (that our model learns to complete) have greater variability than crowns; the latter are typically adapted from templates using standard techniques for sizing and contact points. Therefore, the two problems are of comparable complexity, and an appropriate adaptation of our methodology could lead to dental crowns with better personalization than existing techniques.

VI. Conclusion

We have presented an approach to perform mesh completion from a partial input point cloud. Our model, called MC-Net, controls the mesh completion process using different mesh-related losses, graph unpooling, information propagation along edges and a mesh template. Our results show that MC-Net can complete single surfaces accurately compared to the state-of-the-art PF-Net. The predicted surfaces are always smooth and are sized accurately. Thus, we believe that this work will be beneficial to subsequent research on tooth completion and to the related problem of dental crown generation.

Acknowledgment

This work was funded by Kerenor Dental Studio, Intellident Dentaire Inc., iMD Research Inc., the Natural Science and Engineering Research Council of Canada (NSERC), and MEDTEQ. We thank Compute Canada for providing computational resources for our experimentation. We also acknowledge the help provided by JACOBB Inc. and Object

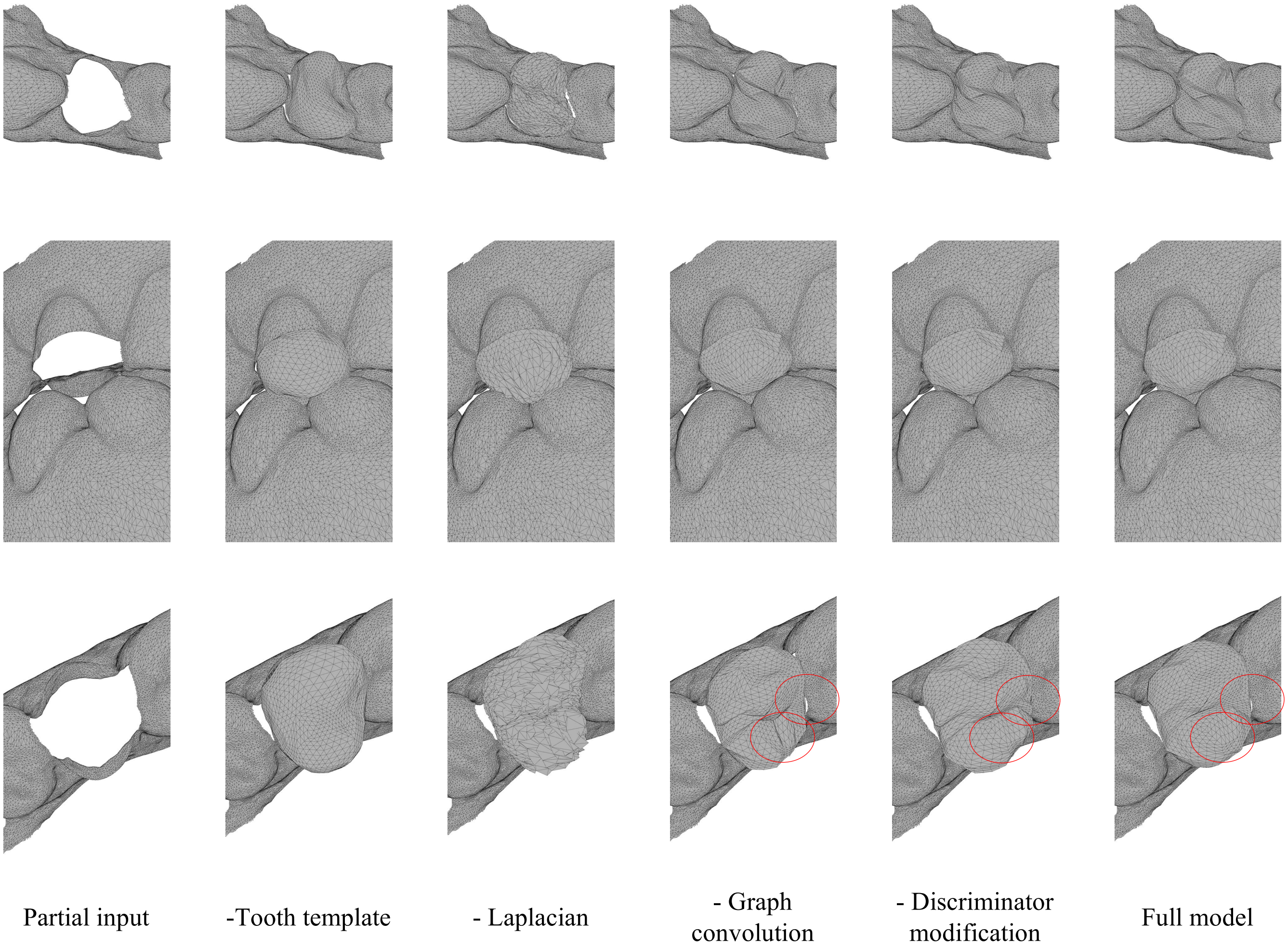


Fig. 8 Qualitative results of the ablation study. The images in each column reflect the contribution of each component of MC-Net by showing the resulting mesh without that component. The red circles overlaid on the images at bottom right show subtle differences between the results.

Research Systems Inc. We thank Philippe Debanné for revising this manuscript. This work was approved by the research ethics committee of Polytechnique Montréal. It has not been submitted elsewhere for publication.

VII. References

[1] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[2] Z. Huang, Y. Yu, J. Xu, F. Ni and X. Le, "PF-Net: Point Fractal Network for 3D Point Cloud Completion," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 7662–7670.

[3] J.-J. Hwang, S. Azernikov, A. A. Efros, and S. X. Yu, "Learning Beyond Human Expertise with Generative Models for Dental Restorations," *arXiv Prepr. arXiv1804.00064*, 2018.

[4] F. Yuan, N. Dai, S. Tian, B. Zhang, Y. Sun, Q. Yu, H. Liu, "Personalized design technique for the dental occlusal surface based on conditional generative adversarial networks," *Int. j. numer. method. biomed. eng.*, vol. 36, no. 5, p. e3321, 2020, doi: 10.1002/cnm.3321.

[5] P. Isola, J.-Y. Zhu, T. Zhou and A. A. Efros, "Image-to-Image Translation with Conditional Adversarial Networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1125–1134, Accessed: Sep. 27, 2021. [Online]. Available: https://github.com/phillipi/pix2pix.

[6] Y. Ping, G. Wei, L. Yang, Z. Cui, and W. Wang, "Self-attention implicit function networks for 3D dental data completion," *Comput. Aided Geom. Des.*, vol. 90, p. 102026, Oct. 2021, doi: 10.1016/J.CAGD.2021.102026.

[7] O. Lessard, F. Guibault, J. Keren, and F. Cheriet, "Dental Restoration using a Multi-Resolution Deep Learning Approach," *2022 IEEE 19th Int. Symp. Biomed. Imaging*, pp. 1–4, Mar. 2022, doi: 10.1109/ISBI52829.2022.9761622.

[8] H. Zhu, X. Jia, C. Zhang, and T. Liu, "ToothCR: A Two-Stage Completion and Reconstruction Approach on 3D Dental Model," in *Advances in Knowledge Discovery and Data Mining: 26th Pacific-Asia Conference, PAKDD 2022, Chengdu, China*, 2022, pp. 161–172, doi: 10.1007/978-3-031-05981-0_13.

[9] N. Wang, Y. Zhang, Z. Li, Y. Fu, W. Liu, and Y.-G. Jiang, "Pixel2Mesh: Generating 3D Mesh Models from Single RGB Images," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 52–67.

[10] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *Advances in Neural Information Processing Systems*, Jun. 2017, vol. 30, Accessed: Nov. 01, 2021. [Online]. Available: https://arxiv.org/abs/1706.02413v1.

[11] Y. Yang, C. Feng, Y. Shen, and D. Tian, "FoldingNet: Point Cloud Auto-encoder via Deep Grid Deformation," in *Proceedings of the*

IEEE Conference on Computer Vision and Pattern Recognition, Dec. 2018, pp. 206–215, Accessed: May 21, 2021. [Online]. Available: http://arxiv.org/abs/1712.07262.

[12] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville and Y. Bengio, "Generative Adversarial Nets," 2014, Accessed: Jan. 25, 2022. [Online]. Available: http://www.github.com/goodfeli/adversarial.

[13] B. Fei, W. Yang, W. Chen, Z. Li, Y. Li, T. Ma, X. Hu and L. Ma, "Comprehensive Review of Deep Learning-Based 3D Point Clouds Completion Processing and Analysis," *arXiv Prepr. arXiv2203.03311*, 2022.

[14] X. Yu, Y. Rao, Z. Wang, Z. Liu, J. Lu, and J. Zhou, "PoinTr: Diverse Point Cloud Completion with Geometry-Aware Transformers," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Aug. 2021, pp. 12498–12507, Accessed: Dec. 01, 2021. [Online]. Available: https://arxiv.org/abs/2108.08839v1.

[15] P. Xiang, X. Wen, Y.-S. L, Y.-P. Cao, P.Wan, W. Zheng and Z. Han, "SnowflakeNet: Point Cloud Completion by Snowflake Point Deconvolution with Skip-Transformer," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, Aug. 2021, pp. 5499–5509, doi: 10.48550/arxiv.2108.04444.

[16] K. Sarkar, K. Varanasi, and D. Stricker, "Learning quadrangulated patches for 3D shape parameterization and completion," in *International Conference on 3D Vision (3DV)*, 2017, pp. 383–392.

[17] M. Kazhdan, J. Hopkins University, and H. Hoppe, "Screened Poisson Surface Reconstruction," *ACM Trans. Graph.*, vol. 32, no. 3, pp. 1–13, 2013.

[18] O. Litany, A. Bronstein, M. Bronstein, and A. Makadia, "Deformable Shape Completion with Graph Convolutional Autoencoders," *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, pp. 1886–1895, Dec. 2017, Accessed: May 13, 2021. [Online]. Available: http://arxiv.org/abs/1712.00268.

[19] S. Foti, B. Koo, T. Dowrick, J. Ramalhinho, M. Allam, B. Davidson, D. Stoyanov and M. J. Clarkson, "Intraoperative Liver Surface Completion with Graph Convolutional VAE," in *Uncertainty for Safe Utilization of Machine Learning in Medical Imaging, and Graphs in Biomedical Image Analysis*, 2020, pp. 198–207.

[20] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3D-R2N2: A Unified Approach for Single and Multi-view 3D Object Reconstruction," in *European conference on computer vision*, 2016, pp. 628–644.

[21] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta, "Learning a Predictable and Generative Vector Representation for Objects," in *European Conference on Computer Vision*, 2106, pp. 484–499.

[22] H. Fan, H. Su, and L. Guibas, "A Point Set Generation Network for 3D Object Reconstruction from a Single Image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613, Accessed: Oct. 14, 2021. [Online]. Available: https://github.com/fanhqme/PointSetGeneration.

[23] C. Wen, Y. Zhang, Z. Li, and Y. Fu, "Pixel2Mesh++: Multi-View 3D Mesh Generation via Deformation," in *Proceedings of the IEEE International Conference on Computer Vision*, Aug. 2019, pp. 1042–1051, doi: 10.48550/arxiv.1908.01491.

[24] E. J. Smith, S. Fujimoto, A. Romero, and D. Meger, "GEOMetrics: Exploiting Geometric Structure for Graph-Encoded Objects," *arXiv Prepr. arXiv1901.11461*, 2019, Accessed: Apr. 14, 2022. [Online]. Available: https://github.com/EdwardSmith1884/GEOMetrics.

[25] B. Guillard , E. Remelli, A. Lukoianov, S. Richter, T. Bagautdinov, P. Baque and P. Fua, "DeepMesh: Differentiable Iso-Surface Extraction," *arXiv Prepr. arXiv2106.11795*, 2021.

[26] Y. Shi, B. Ni, J. Liu, D. Rong, Y. Qian, and W. Zhang, "Geometric Granularity Aware Pixel-to-Mesh," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 13097–13106.

[27] X. Chen, N. Ravikumar, Y. Xia, R. Attar, A. Diaz-Pinto, S. K Piechnik, S. Neubauer, S. E Peteresen and A. F Frangi, "Shape registration with learned deformations for 3D shape reconstruction from sparse and incomplete point clouds," *Med. Image Anal.*, vol. 74, p. 102228, Dec. 2021, doi: 10.1016/J.MEDIA.2021.102228.

[28] C. Lian L. Wang, T.-H. Wu, F. Wang, P.-T. Yap, C.-C. Ko and D. Shen, "Deep Multi-Scale Mesh Feature Learning for Automated Labeling of Raw Dental Surfaces from 3D Intraoral Scanners," *IEEE Trans. Med. Imaging*, vol. 39, no. 7, pp. 2440–2450, Jul. 2020, doi: 10.1109/TMI.2020.2971730.

[29] A. Alsheghri, F. Ghadiri, Y. Zhang, O. Lessard, J. Keren, F. Cheriet and F. Guibault, "Semi-supervised segmentation of tooth from 3D scanned dental arches," in *Medical Imaging 2022: Image Processing*, Apr. 2022, vol. 12032, pp. 766–771, doi: 10.1117/12.2612655.

[30] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, and S. Member, "The Bal I-Pivoting Algorithm for Surface Reconstruction," *IEEE Trans. Vis. Comput. Graph. Vis. Comput. Graph.*, vol. 5, no. 4, pp. 349–359, 1999.