Learning Feature Aggregation for Deep 3D Morphable Models

Zhixiang Chen
Imperial College London
zhixiang.chen@imperial.ac.uk

Tae-Kyun Kim
Imperial College London and KAIST
tk.kim@imperial.ac.uk

Abstract

3D morphable models are widely used for the shape representation of an object class in computer vision and graphics applications. In this work, we focus on deep 3D morphable models that directly apply deep learning on 3D mesh data with a hierarchical structure to capture information at multiple scales. While great efforts have been made to design the convolution operator, how to best aggregate vertex features across hierarchical levels deserves further attention. In contrast to resorting to mesh decimation, we propose an attention based module to learn mapping matrices for better feature aggregation across hierarchical levels. Specifically, the mapping matrices are generated by a compatibility function of the keys and queries. The keys and queries are trainable variables, learned by optimizing the target objective, and shared by all data samples of the same object class. Our proposed module can be used as a train-only drop-in replacement for the feature aggregation in existing architectures for both downsampling and upsampling. Our experiments show that through the end-to-end training of the mapping matrices, we achieve state-of-the-art results on a variety of 3D shape datasets in comparison to existing morphable models.

1. Introduction

The 3D morphable models play a core role in many 3D applications, including identity recognition, shape retrieval, shape completion, animation and 3D reconstruction from 2D images [46, 42, 26]. 3D morphable models encode raw 3D shapes into latent variables, from which the shapes can be reconstructed to some approximation by interacting with the 3D shape model. In this work, we are particularly interested in 3D mesh shapes that share a common template and are already aligned to the template. The commonly employed encoding approaches are the linear encoding of Principal Component Analysis [1, 26, 44, 53] or manually defined blendshape [45, 24, 8] for human face, skinned vertex based models like Skinned Multi-Person Linear Model [30] and hand model with articulated and non-rigid de-

formations [38] for body and hand and non-linear encodings of autoencoder neural nets [43]. Compared to the classical linear models, the non-linear models, especially deep learning based models, offer the possibility to capture detailed deformations like wrinkles and surpass them in terms of generalization, compactness and specificity [40].

Recently, there is an emerging interest in generalizing CNNs to meshes with operations directly defined on meshes [37, 9]. Both isotropic [37] and anisotropic [9] convolutions have been introduced for meshes with a fixed topology. Feature aggregation is defined for the downsampling and upsampling of vertex features and combined with convolutions to construct mesh autoencoders [37]. Despite the success of such models, the feature aggregation across hierarchical levels is determined by a preprocessing stage that maintains surface error approximations with quadric metrics [17] rather than by optimizing the learning objective. Such procedure limits the representation power of the morphable model to capture fine grained deformations.

In this work, we propose an attention based feature aggregation strategy to construct hierarchical representations for 3D meshes with a fixed topology (Fig. 1(a)). Instead of using the precomputed matrices (Fig. 1(b)), we propose to learn the mapping matrices along with the convolutions in the network (Fig. 1(c)). The attention based mapping module introduces keys and queries as trainable variables that are shared by all data samples in the dataset. With keys and queries standing for the vertices at the preceding and succeeding levels respectively, the mapping matrices are derived by a compatibility function of the keys and queries. This allows the receptive fields and aggregation weights to be simultaneously learned. By varying the number of keys and queries, this module can be used for either downsampling or upsampling. Since only mapping matrices are needed for inference, we can detach the attention based mapping module once training is finished to avoid additional cost at inference stage. We evaluate our method on the reconstruction task which serves as a fundamental testbed for further applications. We quantitatively and qualitatively show the results on three 3D human shape datasets: faces, bodies, and hands. As a drop-in replacement for ex-
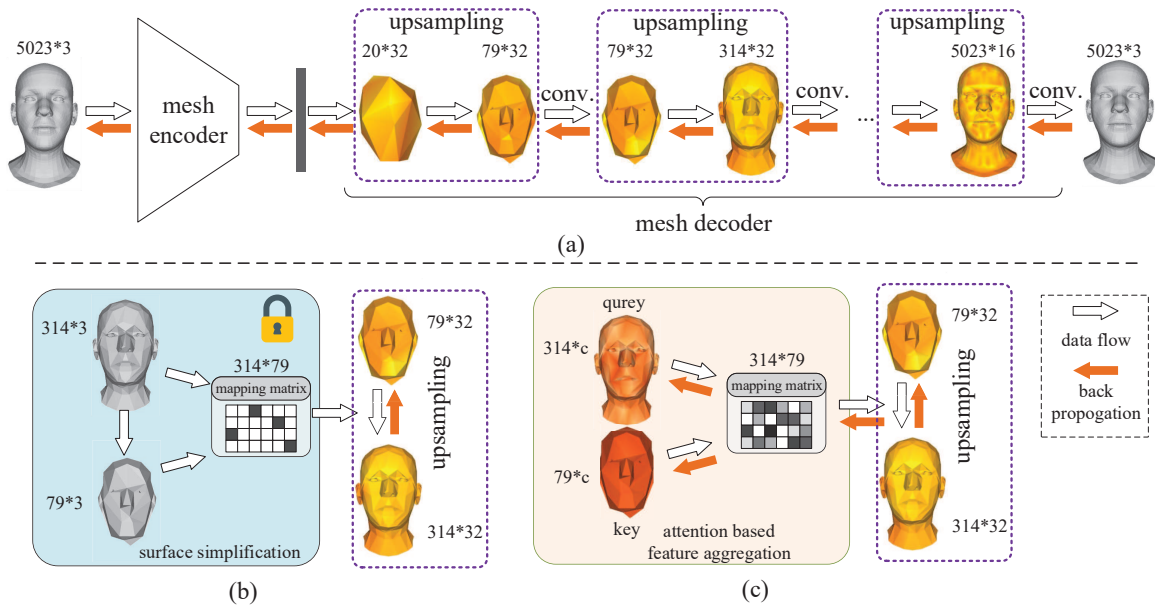
Figure 1. (a) Deep 3D morphable model with mesh encoder and decoder. The mesh encoder encodes a mesh as a compact latent representation, which the mesh decoder takes as input to recover the mesh with hierarchical upsampling and convolution operations. (b) A fixed mapping matrix generated by surface simplification is used for feature aggregation across hierarchical levels in [37, 9]. (c) In this work, the mapping matrix for feature aggregation is generated by an attention based module and jointly learned with other components of the model

isting feature aggregation method, our method boosts the performance of existing models by a large margin, in combination with either isotropic or anisotropic convolution operators.

2. Related work

2.1. 3D Morphable Models

3D Morphable Models represent the shape of objects as 3D meshes that can be deformed to match a particular instance of that object class. These meshes are in dense correspondence with a shared template of the object class. Existing models take the statistical information of a set of representative examples of the class as prior knowledge to model shapes. The most well studied models are on human face [15, 26], body [30, 34] and hand [38]. Principal Component Analysis is the most commonly used approach for statistical modeling and has been used to model faces [3, 35, 11, 5, 36]. SMPL [30, 34] and MANO [38] are the most well know models for body and hand. These linear models filter out the high frequency signal of the shapes, thereby losing details, and require class specific joint localisation for shape deformation.

Deep 3D morphable models apply 3D deep learning to model the nonlinear shape variations and are not limited to a specific class of shapes. Litany *et al.* [28] stack several graph convolutional layers and average the features of all vertices to obtain the mesh representation. Ranjan *et*

al. [37] propose a mesh autoencoder called CoMA with fast localised convolutional filters [12] and mesh downsampling and upsampling layer to support multi-scale hierarchical structure. Bouritsas *et al.* [9] propose to replace the isotropic convolutions with anisotropic spiral convolutions. Vertex-wise weighted convolutions are further proposed in [16, 56] at the cost of increased number of parameters. Tretschk *et al.* [47] add an embedded deformation layer to the mesh autoencoder. Feature aggregation in most of these works adopts direct selection for downsampling and barycentric coordinates for upsampling. The aggregation functions are fixed and determined by quadric mesh simplification rather than minimizing reconstruction error. Although the aggregation weights are learned from data in [56], the aggregation is still performed over fixed local neigborhood. In contrast, we propose to learn both the aggregation weights and receptive fields. Furthermore, rather than directly setting the weights as trainable parameters, we generate the weights by an attention based module to avoid over-parameterization.

2.2. Geometric deep learning on meshes

Extending convolution operations to meshes and graphs has been studied in both spectral-based and spatial-based fields. Spectral approaches build convolutional operations on spectral representations of graphs with graph Laplacian [10, 6, 22, 25]. Ranjan *et al.* [37] apply the truncated Chebyshev polynomials [12, 23] to meshes. On the oth-

er hand, spatial-based approaches take as input per vertex's local spatial structure and features to construct convolutions [20, 49, 50, 33, 52, 39, 16, 56]. Geodesic polar coordinates and anisotropic heat kernels are adopted to parameterize local mesh surfaces for convolutions in [32, 7]. Bouritsas *et al.* [9] and Gong *et al.* [19] propose to define local coordinates by the spiral sequence [27]. Convolutions on edges are defined by Hanocka *et al.* [21] and extended for mesh subdivision [29]. While various convolution operators are proposed in these works, the focus of this work lies in feature aggregation across hierarchical levels, which is in parallel to them.

To build hierarchical representations for meshes, Graclus algorithm [13] and Quadric Error Metrics based mesh simplification [17] are adopted to reduce the number of vertices in [12, 50, 18, 37, 55, 51]. Hanocka *et al.* [21] learn to sequentially collapse edges. These approaches aggregate features with permutation invariant functions, such as $\text{sum}(\cdot)$, $\text{max}(\cdot)$ and $\text{mean}(\cdot)$, for downsampling and interpolation functions for upsampling. There are works proposing to learn the aggregation weights via a dense mapping matrix [54, 2] or a fully-connected layer [14]. As shown by our experiments, they deliver poor performance for high resolution meshes due to over-parameterization. Zhou *et al.* [56] directly learn aggregation weights over predefined local neighbors. Different from these works, we learn the mapping matrices for feature aggregation through an attention based module, which is scalable due to linear complexity and enables learning the receptive fields together with the aggregation weights.

3. Learning feature aggregation for 3DMMs

3.1. Deep 3D Morphable Models

The deep 3D morphable model, as illustrated in Fig. 1, consists of an encoder and a decoder. The encoder has a mirrored structure of the decoder. The encoder takes as input a 3D mesh to generate a compact representation, which captures both global and local shape information by a fine-to-coarse hierarchical network structure. Given the compact representation, the decoder learns to generate a mesh to recover detailed shape descriptions in a coarse-to-fine fashion. At each level of the hierarchical structure, convolution and upsampling operations are applied sequentially in the decoder, whereas convolution and downsampling operations are used for the encoder. While the convolution can be implemented by existing isotropic [37] or anisotropic [9] convolutions, the feature aggregation, including upsampling and downsampling, is usually performed by multiplying input features by a mapping matrix.

Formally, we consider a 3D mesh $\mathcal{O}$ as a set of vertices and edges, $\mathcal{O} = (\mathcal{V}, \boldsymbol{A})$, where $\mathcal{V} = \{1, \ldots, n\}$ denotes the set of n vertices and $\boldsymbol{A} \in \{0, 1\}^{n \times n}$ indicates the s-

parse edge connections between vertices. Assume there are L hierarchical levels in both the encoder and decoder of the morphable model. In the following, we only take the decoder as the example to illustrate our method. However, similar analysis can be applied for the encoder. Let $\boldsymbol{X}^{(l)} \in \mathbb{R}^{n_l \times d_l}$ denote the output features of the decoder at level $l \in \{1, 2, \ldots, L\}$, where n_l and d_l are the number of vertices and the feature dimension. $\boldsymbol{X}^{(l-1)}$ serves as the input features at level l.

In existing models [37, 9], feature aggregation can be generally formulated as

$$\boldsymbol{x}_i^{(l)} = \sum_{j=1}^{n_{l-1}} m_{ij}^{(l-1 \to l)} \boldsymbol{x}_j^{(l-1)}, \tag{1}$$

with different aggregation weights $m_{ij}^{(l-1 \to l)} \in \boldsymbol{M}^{(l-1 \to l)}$ assigned for downsampling and upsampling. While these models demonstrate promising performance, the mapping matrices $\boldsymbol{M}^{(l-1 \to l)}$ are calculated by mesh simplification on a template mesh and are fixed during the training of the model. In contrast, we are interested in learning these mapping matrices along with the convolution operations for the training shapes. A straightforward solution would be directly parameterizing the mapping matrices as in [54, 14] and training them along with the other components of the model. However, this suffers from over-parameterization for high resolution meshes and deteriorates the model performance as we show in the experiments. On the contrary, we propose to learn the mapping matrices for feature aggregation through an attention based module.

3.2. Learning feature aggregation via attention

We propose to model the feature aggregation from $\boldsymbol{X}^{(l-1)}$ to $\boldsymbol{X}^{(l)}$ by the attention mechanism [41, 48, 49]. The attention mechanism can be regarded as generating an output with a corresponding query vector and the stored key-value pairs. In the context of feature aggregation, $\boldsymbol{x}_i^{(l)}$ is the output and $\boldsymbol{x}_j^{(l-1)}, j = 1, \ldots, n_{l-1}$ are the stored values. Then, we propose to compute the aggregation weight $m_{ij}^{(l-1 \to l)}$ by a compatibility function $d(\cdot)$ of a query vector $\boldsymbol{q}_i^{(l)} \in \mathbb{R}^c$ and the key vector $\boldsymbol{k}_j^{(l-1)} \in \mathbb{R}^c$ that corresponds to the stored value $\boldsymbol{x}_j^{(l-1)}$. The formulation is as

$$m_{ij}^{(l-1 \to l)} = d\left(\boldsymbol{q}_i^{(l)}, \boldsymbol{k}_j^{(l-1)}\right). \tag{2}$$

By this means, we can break down the construction of mapping matrices into the establishment of the key and query vectors and the design of the compatibility function, as shown in Fig. 2.
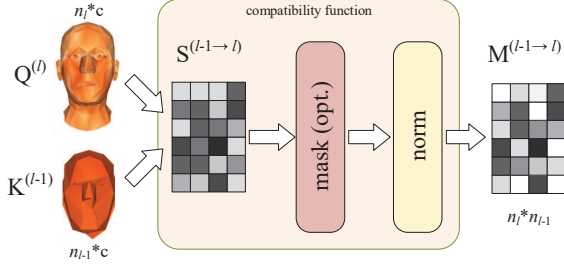
Figure 2. The attention based feature aggregation module

3.2.1 The key and query vectors

Since the key and query vectors are paired with the vertex features, their values are expected to be either associated with the vertex indices or related to the vertex features. Given the fact that feature irrelevant mapping matrices are feasible in the existing models [37, 9], we model the key and query vectors as a function of the vertex indices rather than the vertex features. In other words, we aim to learn object aware mapping matrices instead of instance aware ones. Recall that the meshes considered in this work are already registered, i.e., each vertex has its corresponding semantic meaning. The key and query vectors can, then, be parameterized as trainable variables without taking any data as input. One benefit of such parameterization is that only the learned mapping matrices rather than the attention module are required at inference stage. Furthermore, this also avoids the chicken-and-egg situation between $\boldsymbol{q}_i^{(l)}$ and $\boldsymbol{x}_i^{(l)}$ when associating queries with vertex features. While the numbers of key and query vectors are determined by the numbers of vertices, the feature dimension of each vector c is a hyperparameter to be set.

3.2.2 The compatibility function

Given the key and query vectors, the compatibility function measures how well two vertices at neighboring levels align. In general, arbitrary distance metric can be employed to compute the compatibility score. We propose to compute the compatibility score with the cosine distance between $\boldsymbol{q}_i^{(l)} \in \mathbb{R}^c$ and $\boldsymbol{k}_j^{(l-1)} \in \mathbb{R}^c$ as

$$s_{w,ij}^{(l-1\rightarrow l)} = \cos\left(\boldsymbol{q}_i^{(l)}, \boldsymbol{k}_j^{(l-1)}\right). \qquad (3)$$

Directly applying this score as the aggregation weight would result in a dense mapping matrix. In other words, any vertex feature in a given level is influenced by all vertex features in the preceding level. However, such aggregation is less desirable in the context of hierarchical learning.

In order to capture both the global and local information, the receptive fields of different $\boldsymbol{x}_i^{(l)}$ are expected to

be distinctive to each other. Each vertex in the succeeding level $\boldsymbol{x}_i^{(l)}$ is expected to be related to an unique subset of the vertices in the preceding level. We integrate such prior knowledge into an optional mask operation by setting the cardinality of the subset to a fixed value of k with top-k selection. And the elements of each subset are automatically learned from the training shapes. To be specific, we define a binary mask for the weight score $s_{w,ij}^{(l-1\rightarrow l)}$ as

$$b_{ij}^{(l-1\rightarrow l)} = \begin{cases} 1, & s_{w,ij}^{(l-1\rightarrow l)} \text{ is among the top } k \text{ of } s_{w,i:}^{(l-1\rightarrow l)} \\ 0, & \text{otherwise.} \end{cases}$$

(4)

The masked weight score is then computed by multiplying the weight score by the binary mask as

$$s_{m,ij}^{(l-1\rightarrow l)} = b_{ij}^{(l-1\rightarrow l)} s_{w,ij}^{(l-1\rightarrow l)}. \qquad (5)$$

We then normalize the weight score as

$$m_{a,ij}^{(l-1\rightarrow l)} = \frac{s_{m,ij}^{(l-1\rightarrow l)}}{\sum_{j=1}^{n_{l-1}} s_{m,ij}^{(l-1\rightarrow l)}}. \qquad (6)$$

3.2.3 The mapping matrix

While it is feasible to take Eqn. 6 as the replacement of the precomputed weights in existing models, we further propose to fuse both information to leverage the success of existing methods. The fusion can be thought as a multi-head attention with a fixed head and a learnable head. Specifically, we adopt a weighted combination for fusion as

$$m_{ij}^{(l-1\rightarrow l)} = w_a m_{a,ij}^{(l-1\rightarrow l)} + (1-w_a)m_{p,ij}^{(l-1\rightarrow l)}, \qquad (7)$$

where w_a is a trainable weight and $m_{p,ij}^{(l-1\rightarrow l)}$ stands for the precomputed downsampling or upsampling coefficient in existing models [9, 37]. When w_a is a fixed value of 0, Eqn. 7 is exactly the same as the precomputed ones in existing models. With the aggregation weight computed according to Eqn. 7, we can obtain the mapping matrices $\boldsymbol{M}^{(l-1\rightarrow l)}$ for both upsampling and downsampling in the model.

4. Experiments

In this section, we demonstrate the effectiveness of our proposed attention based feature aggregation for 3D reconstruction on human face, human body, and hand shape datasets in comparison to existing 3DMMs [37, 9]. The code is available at https://github.com/zxchen110/Deep3DMM.

4.1. Datasets

COMA [37]. The face expression dataset presented by Ranjan *et al.* [37] consists of 20K+ well registered 3D face
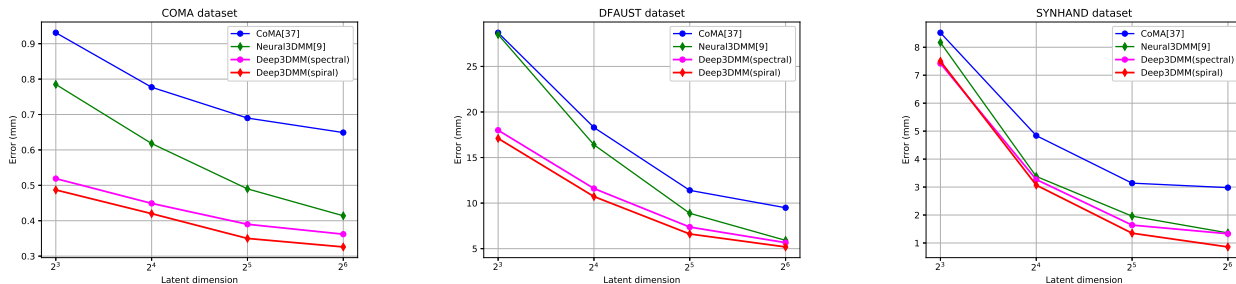
Figure 3. Reconstruction errors of different methods on COMA (left), DFAUST (middle), and SYNHAND (right) datasets

shapes with 5,023 vertices for each shape. The dataset provides twelve extreme expression sequences, such as high smile and bareteeth, from twelve unique identities. We follow the interpolation setting in [37] to split the dataset.

DFAUST [4]. This dataset contains 40K+ dynamic human body shapes with all shapes aligned to a common reference topology with 6,890 vertices [4]. The shapes are captured from ten identities in over one hundred moving sequences, covering actions like punching and shaking shoulders and hips. We follow the data split setting in [9].

SYNHAND [31]. A total of 5 million synthetic hand meshes with varying hand shapes and poses are presented in [31]. We follow [47] to randomly sample 100K meshes for the experiments with hands, where 90K meshes are used for training and validation and the remaining 10K meshes are used to evaluate the trained model.

4.2. Implementation details

We choose to evaluate the proposed feature aggregation method on two state-of-the-art 3D morphable models with isotropic spectral convolution [37] and anisotropic spiral convolution [9], denoted as Deep3DMM(spectral) and Deep3DMM(spiral), respectively. We follow the settings in [37] to configure the network architecture as an encoder-decoder with four downsampling and upsampling layers. The mapping matrices are generated by our method while the remaining components of the model are consistent with existing models. The whole network is trained with the recommended settings in the original works [37, 9].

For our attention based feature aggregation module, we empirically set c to 21 and k to 2 and 32 for downsampling and upsampling, respectively. The trainable weight w_a is initialized as 0.2. Different from the standard initialization of the remaining components in the network, we use the results of mesh simplification to initialize the key and query vectors. In particular, we initialize the first three dimensions of the key and query vectors by the spatial position of vertices at the corresponding level. The remaining dimensions are randomly initialized by an uniform distribution.

4.3. Results of attention based feature aggregation

4.3.1 Quantitative results of reconstruction

We follow previous works [37, 9] to evaluate the performance of shape representation in terms of Euclidean distance based reconstruction error on the test set. In Fig. 3, we show the quantitative results of the averaged reconstruction error on three datasets with the dimension of latent representation varying from 8 to 64.

As shown in the figure, morphable models equipped with our attention based feature aggregation module consistently outperform the corresponding baseline models under all evaluated settings. For the spectral convolution based models, our feature aggregation reduces the reconstruction errors by a large margin for all tested latent dimensions on all three datasets (60% lower on COMA dataset). That should be attributed to the mechanism of learning both the receptive fields and the aggregation weights. This mechanism allows each vertex to be affiliated with a selected portion of vertices at the preceding level with different weights, which compensates the isotropy of spectral convolutions. For the spiral convolution based models, our feature aggregation also enhances the model capacity in every tested case. A possible reason of the improvement is the adaptive learning of the receptive field for each vertex.

We also observe that the gap between spectral convolution based model and spiral convolution based model is narrowed after applying our feature aggregation. We attribute this to the fact that our feature aggregation also introduces certain anisotropic operations which partly overlap with the spiral convolutions. Moreover, we observe that the reduced error introduced by our feature aggregation, especially in combination with spiral convolutions, is more significant for shorter latent representations, which may benefit downstream tasks to capture semantically meaningful and discriminative representations with a lightweight model.

4.3.2 Qualitative results of reconstruction

In Fig. 4, we visualize the per vertex Euclidean error of different morphable models on several shapes from the three
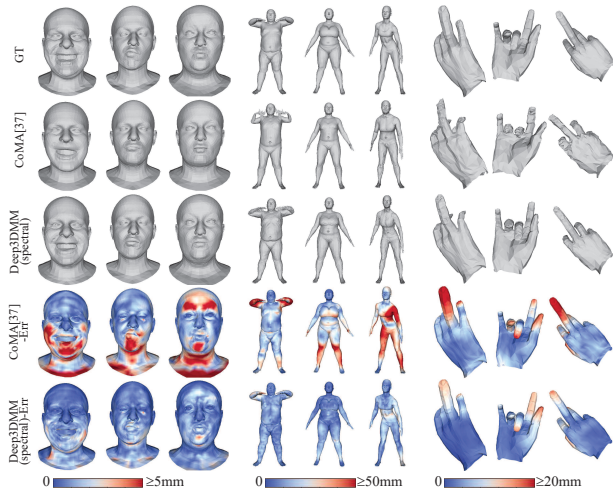
Figure 4. Qualitative comparison of spectral convolution based models on COMA (left), DFAUST (middle), and SYNHAND (right) datasets. The first row is the ground truth shapes. The second and third rows show the reconstructed shapes, while the fourth and fifth rows show the corresponding reconstruction errors
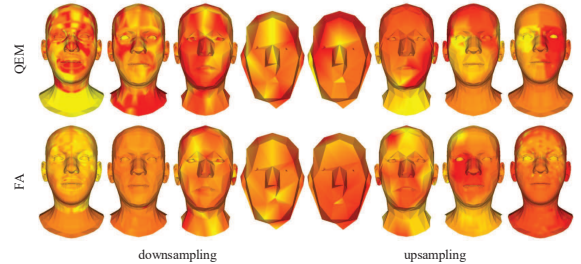

Figure 5. Visualization of mapping matrices of down-sampling and up-sampling on COMA dataset with t-SNE. The first and second rows are the results of quadric error minimization in [37, 9] and our proposed feature aggregation, respectively

datasets for qualitative comparison. The latent dimension is set as 8. We can observe that our feature aggregation reduces the reconstruction error not only on the vertices with large errors but also on other vertices. While existing morphable models may have wrong predictions in some parts of the shape, such as the forehead of face, the hands and legs of human, and the fingers of hand, our proposed feature aggregation can alleviate this problem by providing rough correct predictions. Meanwhile, our feature aggregation leads to more realistic reconstructions by recovering more details.

4.3.3 Visualization of mapping matrices

To understand the learning of mapping matrices for feature aggregation, we provide the visualization of mapping matrices in Figs. 5 and 6 with the spectral convolution on COMA dataset. In Fig. 5, we show the mapping matrices of both downsampling and upsampling on the corresponding shapes for both quadric error minimization (QEM) in existing morphable models [37, 9] and our proposed feature aggregation (FA). We map the column/row vectors of the mapping matrices of down-/up-sampling by t-SNE to a 1-dimension manifold for visualization. We can observe that the learned mapping matrices are notably different from those computed by mesh decimation. The learned aggregation weights are more likely to be similar for vertices that are close to each other, such as the region around eyes in the rightmost shape.

In Fig. 6, we visualize the receptive fields of some exemplar vertices for the last level of upsampling. Since the traced downsampling and barycentric coordinates are used

for upsampling in QEM, we can find that the receptive fields are small local regions around the traced vertices, as shown in Fig. 6(b). In contrast, we can observe that the receptive fields are randomly distributed on the face with random initialization of the keys and queries in our attention based feature aggregation, as shown in the leftmost of Fig. 6(c). By training the feature aggregation together with the convolutions, most of the receptive fields also converge to the local regions around the traced vertices, but with larger sizes than that of QEM. Besides the geodesic neighboring vertices, there are some non-local receptive fields, such as vertices at the back of head as shown in the bottom row. Note that the training of mapping matrices is based on the vertex features, rather than the spatial information of each vertex. It is possible that non-local vertices contain the identity and expression information that is helpful for recovering shape details. Along with the training of receptive fields, the aggregation weights are simultaneously trained for the receptive fields. This is shown in the figure by the variation of saturation and brightness, for example on the chin and neck.

4.4. Comparison with other aggregation methods

In order to show the effectiveness of simultaneously learning receptive fields and aggregation weights by the attention mechanism, we also conduct experiments with different aggregation methods. In Table 1, we show the results with latent dimension of 8 on COMA dataset with spectral convolutions. Full mapping method directly parameterizes the mapping matrices as trainable parameters to learn both the aggregation weights and receptive fields. However, this method delivers poor performance since it suffers from over-parameterization. We migrate DiffPool [56] for mesh autoencoders by learning the mapping matrices with additional network layers as suggested by the authors. This method also does not perform well due to learning the matrics directly. Average aggregation uses the same receptive fields as QEM, but the assigned aggregation weights are identical for vertices in the receptive field. QEM in [37, 9] adopts mesh decimation for downsampling and barycentric
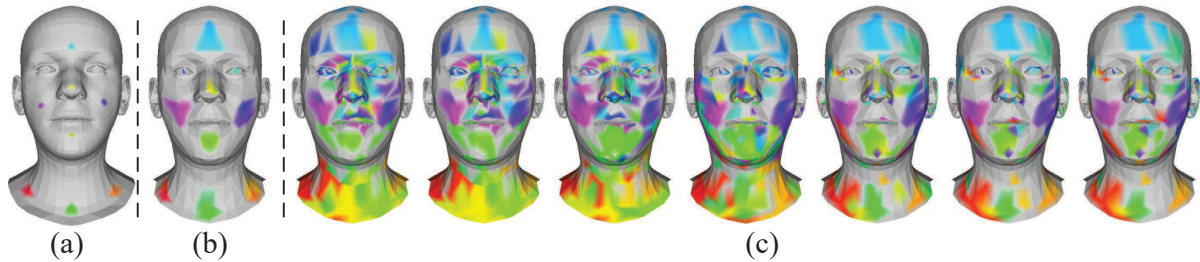
Figure 6. Illustrations of receptive fields of selected vertices at the finest level of upsampling on COMA dataset (best viewed in color). The first column (a) shows the selected vertices with colors. The second column (b) shows the receptive fields of the existing QEM method at the neighboring coarser level. The remaining columns (c) show the receptive fields of our proposed attention based feature aggregation at epochs of 1, 50, 100, 150, 200, 250, and 300 (from left to right) with convolutions in [37]

Table 1. Comparison of different aggregation methods. ∗ denotes our implementation of the method for the mesh autoencoder in this paper

Aggregation method	Error (mm)
Full mapping	4.319
DiffPool [54]*	3.964
Average	0.975
QEM [37]	0.931
Variant weight [56]*	0.928
Ours	**0.519**

Table 2. Reconstruction errors with different settings of filters

| Method | simple | | wider | |
	error	# of par.	error	# of par.
CoMA [37]	0.939	30,056	0.682	179,656
Ours	0.519	30,056	0.441	179,656

Table 3. Ablation study of the feature aggregation

| Method | encoder/decoder | | | |
	QEM/QEM	FA/QEM	QEM/FA	FA/FA
Error (mm)	0.939	0.886	0.525	0.519

coordinates for upsampling. Variant weight [56] automatically learns the aggregation weights by setting them as trainable parameters. Since the local region defined in [56] relies on extra connection matrices between hierarchical levels which are unavailable in existing models [37, 9], we set 1-ring neighbors as the local region. The main drawback of these methods is the inability to learn the receptive fields. We do not compare with MeshCNN [21] since it is unclear how to incorporate the edge based operations to existing models [37, 9] defined on vertices. It is also computation inefficient and worse than Neural3DMM [9] as show in [56]. Our proposed feature aggregation balances well between the number of parameters and the flexibility to learn receptive fields and aggregation weights via the attention mechanism.

4.5. Results with different settings of filters

We also conduct experiments on two settings of the number of convolution filters to explore the effectiveness of our proposed feature aggregation mechanism with different networks. The simple setting represents the configuration as in [37], where the number of filters are (3,16,16,16,32) and (32,32,16,16,16,3) for the encoder and decoder, respectively. The wider setting represents the network with larger numbers of filters, where they are (3,16,32,64,128) and

(128,64,32,32,16,3) for the encoder and decoder, respectively. Table 2 shows the reconstruction errors on CO-MA dataset with the latent dimension of 8, together with the number of parameters at the inference stage. We can see that our feature aggregation mechanism consistently reduces the error under both settings with the same number of parameters at the inference stage.

4.6. Ablation study of the feature aggregation

In Table 3, we show the reconstruction errors by applying the proposed feature aggregation (FA) on either the encoder or the decoder. The results show that the performance can be enhanced by learning the feature aggregation on either the encoder or the decoder, especially on the decoder.

4.7. Latent space arithmetic

We also evaluate the representational power of the learned latent space by performing arithmetic operations on latent representations.

Interpolation. Given the latent representations z_1 and z_2 of two sufficient different shapes $\mathcal{O}_1$ and $\mathcal{O}_2$, we linearly interpolate between them to obtain a new latent representation $z_0 = \alpha z_1 + (1 - \alpha)z_2$ with $\alpha \in (0, 1)$. As shown
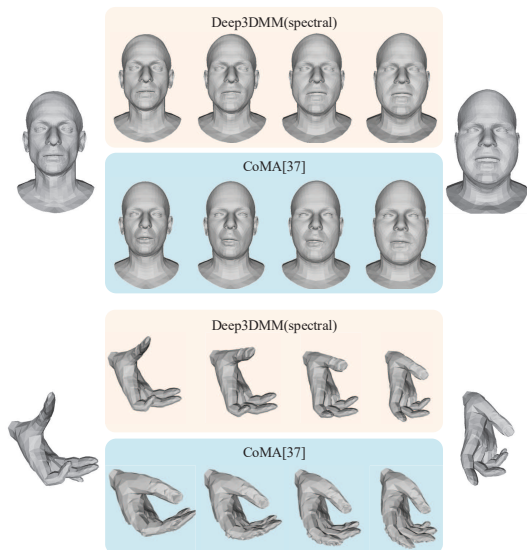
Figure 7. Interpolations between different shapes. The left-most and right-most shapes are input shapes

in Fig. 7, by decoding z_0 with the mesh decoder, we can obtain the in-between shapes for different α. The results of our model are better.

Extrapolation. Similar to the linear combination for interpolation, we obtain an extrapolated latent representation by performing $z_0 = \alpha z_1 + (1 - \alpha)z_2$ but with $\alpha \in (-\infty, 0) \cup (1, \infty)$. As shown in Fig. 8, we choose z_1 to be the neutral shape of the same identity as z_2 and show the overdrawn shape deformation of a given identity. The results of our model are more natural.

Deformation transfer. It is also possible to transfer shape deformation between different identities. Let $\mathcal{S}_0, \mathcal{T}_0$ be the shapes of two different identities with the neutral expression/pose, $\mathcal{S}_1$ be the shape of the same identity as $\mathcal{S}_0$ but with a specific expression/pose. We compute the shape deformation by $d = \mathcal{S}_1 - \mathcal{S}_0$ and apply this deformation to $\mathcal{T}_0$ by $\mathcal{T}_1 = \mathcal{T}_0 + d$ to get the deformed shape. Examples of face and human body are shown in Fig. 9. Our model can successfully transfer the deformation while the baseline model fails.

5. Conclusion

In this paper, we propose to learn feature aggregation for deep 3D morphable models. Specifically, we introduce keys and queries as trainable parameters and adopt the attention mechanism to compute the mapping matrices for vertex feature upsampling and downsampling. Our attention based feature aggregation allows the receptive fields and aggregation weights to be simultaneously learned. Our experiments on three benchmarks show that the proposed feature aggregation can enhance the capacity of 3D morphable models
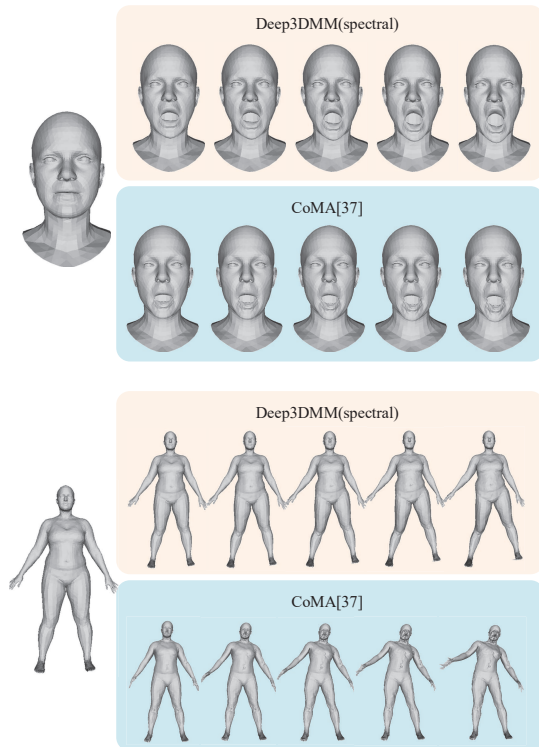

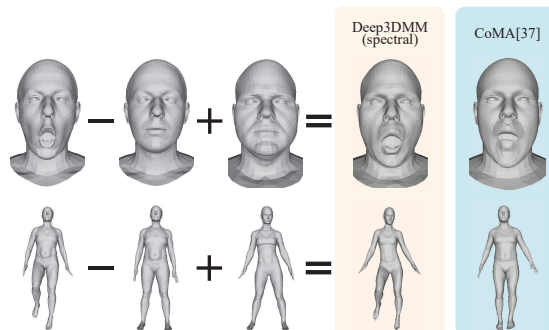Figure 8. Extrapolation. The left-most shapes are the neutral shapes

Figure 9. Deformation transfer with arithmetic operations on latent representations

with either isotropic or anisotropic convolutions.

There are several possible future directions for our work. While the attention mechanism is introduced for feature aggregation in this work, it would be interesting to apply the attention mechanism for the other components of the morphable models. In addition, while current models require the input shapes to be registered to a template mesh, we are also interested in extending our method to arbitrary meshes.

Acknowledgment

References

[1] Brian Amberg, Reinhard Knothe, and Thomas Vetter. Expression invariant 3D face recognition with a morphable model. In *FG*, pages 1–6, 2008.

[2] Filippo Maria Bianchi, Daniele Grattarola, and Cesare Alippi. Mincut pooling in graph neural networks. *CoRR*, abs/1907.00481, 2019.

[3] Volker Blanz and Thomas Vetter. A morphable model for the synthesis of 3D faces. In *SIGGRAPH*, pages 187–194, 1999.

[4] Federica Bogo, Javier Romero, Gerard Pons-Moll, and Michael J. Black. Dynamic FAUST: registering human bodies in motion. In *CVPR*, pages 5573–5582, 2017.

[5] James Booth, Anastasios Roussos, Allan Ponniah, David Dunaway, and Stefanos Zafeiriou. Large scale 3D morphable models. *IJCV*, 126(2-4):233–254, 2018.

[6] Davide Boscaini, Jonathan Masci, Simone Melzi, Michael M. Bronstein, Umberto Castellani, and Pierre Vandergheynst. Learning class-specific descriptors for deformable shapes using localized spectral convolutional networks. *Comput. Graph. Forum*, 34(5):13–23, 2015.

[7] Davide Boscaini, Jonathan Masci, Emanuele Rodolà, and Michael M. Bronstein. Learning shape correspondence with anisotropic convolutional neural networks. In *NeurIPS*, pages 3189–3197, 2016.

[8] Sofien Bouaziz, Yangang Wang, and Mark Pauly. Online modeling for realtime facial animation. *TOG*, 32(4):40:1–40:10, 2013.

[9] Giorgos Bouritsas, Sergiy Bokhnyak, Stylianos Ploumpis, Stefanos Zafeiriou, and Michael M. Bronstein. Neural 3D morphable models: Spiral convolutional networks for 3D shape representation learning and generation. In *ICCV*, pages 7212–7221, 2019.

[10] Joan Bruna, Wojciech Zaremba, Arthur Szlam, and Yann LeCun. Spectral networks and locally connected networks on graphs. In *ICLR*, 2014.

[11] Chen Cao, Yanlin Weng, Shun Zhou, Yiying Tong, and Kun Zhou. Facewarehouse: A 3D facial expression database for visual computing. *TVCG*, 20(3):413–425, 2014.

[12] Michaël Defferrard, Xavier Bresson, and Pierre Vandergheynst. Convolutional neural networks on graphs with fast localized spectral filtering. In *NeurIPS*, pages 3837–3845, 2016.

[13] Inderjit S. Dhillon, Yuqiang Guan, and Brian Kulis. Weighted graph cuts without eigenvectors A multilevel approach. *PAMI*, 29(11):1944–1957, 2007.

[14] Bardia Doosti, Shujon Naha, Majid Mirbagheri, and David J. Crandall. Hope-net: A graph-based model for hand-object pose estimation. In *CVPR*, pages 6607–6616, 2020.

[15] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhöfer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3D morphable face models - past, present, and future. *ACM TOG*, 39(5):157:1–157:38, 2020.

[16] Zhongpai Gao, Guangtao Zhai, Juyong Zhang, Junchi Yan, Yiyan Yang, and Xiaokang Yang. Learning local neighboring structure for robust 3D shape representation. In *AAAI*, 2021.

[17] Michael Garland and Paul S. Heckbert. Surface simplification using quadric error metrics. In *SIGGRAPH*, pages 209–216, 1997.

[18] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3D hand shape and pose estimation from a single RGB image. In *CVPR*, pages 10833–10842, 2019.

[19] Shunwang Gong, Lei Chen, Michael M. Bronstein, and Stefanos Zafeiriou. Spiralnet++: A fast and highly efficient mesh convolution operator. In *ICCV Workshops*, pages 4141–4148, 2019.

[20] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *NeurIPS*, pages 1024–1034, 2017.

[21] Rana Hanocka, Amir Hertz, Noa Fish, Raja Giryes, Shachar Fleishman, and Daniel Cohen-Or. MeshCNN: a network with an edge. *TOG*, 38(4):90:1–90:12, 2019.

[22] Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *CoRR*, abs/1506.05163, 2015.

[23] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *ICLR*, 2017.

[24] Hao Li, Thibaut Weise, and Mark Pauly. Example-based facial rigging. *TOG*, 29(4):32:1–32:6, 2010.

[25] Ruoyu Li, Sheng Wang, Feiyun Zhu, and Junzhou Huang. Adaptive graph convolutional neural networks. In *AAAI*, pages 3546–3553, 2018.

[26] Tianye Li, Timo Bolkart, Michael J. Black, Hao Li, and Javier Romero. Learning a model of facial shape and expression from 4D scans. *TOG*, 36(6):194:1–194:17, 2017.

[27] Isaak Lim, Alexander Dielen, Marcel Campen, and Leif Kobbelt. A simple approach to intrinsic correspondence learning on unstructured 3D meshes. In *ECCV*, pages 349–362, 2018.

[28] Or Litany, Alexander M. Bronstein, Michael M. Bronstein, and Ameesh Makadia. Deformable shape completion with graph convolutional autoencoders. In *CVPR*, pages 1886–1895, 2018.

[29] Hsueh-Ti Derek Liu, Vladimir G. Kim, Siddhartha Chaudhuri, Noam Aigerman, and Alec Jacobson. Neural subdivision. *TOG*, 39(4):124, 2020.

[30] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J. Black. SMPL: a skinned multi-person linear model. *TOG*, 34(6):248:1–248:16, 2015.

[31] Jameel Malik, Ahmed Elhayek, Fabrizio Nunnari, Kiran Varanasi, Kiarash Tamaddon, Alexis Héloir, and Didier Stricker. Deephps: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth. In *International Conference on 3D Vision, 3DV*, pages 110–119, 2018.

[32] Jonathan Masci, Davide Boscaini, Michael M. Bronstein, and Pierre Vandergheynst. Geodesic convolutional neural networks on riemannian manifolds. In *ICCV Workshops*, pages 832–840, 2015.

[33] Federico Monti, Davide Boscaini, Jonathan Masci, Emanuele Rodolà, Jan Svoboda, and Michael M. Bronstein. Geometric deep learning on graphs and manifolds using mixture model CNNs. In *CVPR*, pages 5425–5434, 2017.

[34] Ahmed A. A. Osman, Timo Bolkart, and Michael J. Black. STAR: sparse trained articulated human body regressor. In *ECCV*, pages 598–613, 2020.

[35] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS*, pages 296–301, 2009.

[36] Stylianos Ploumpis, Haoyang Wang, Nick E. Pears, William A. P. Smith, and Stefanos Zafeiriou. Combining 3D morphable models: A large scale face-and-head model. In *CVPR*, pages 10934–10943, 2019.

[37] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *ECCV*, pages 725–741, 2018.

[38] Javier Romero, Dimitrios Tzionas, and Michael J. Black. Embodied hands: modeling and capturing hands and bodies together. *TOG*, 36(6):245:1–245:17, 2017.

[39] Jonas Schult, Francis Engelmann, Theodora Kontogianni, and Bastian Leibe. Dualconvmesh-net: Joint geodesic and euclidean convolutions on 3D meshes. In *CVPR*, pages 8609–8619, 2020.

[40] Martin Styner, Kumar T. Rajamani, Lutz-Peter Nolte, Gabriel Zsemlye, Gábor Székely, Christopher J. Taylor, and Rhodri H. Davies. Evaluation of 3D correspondence methods for model building. In *International Conference on Information Processing in Medical Imaging IPMI*, pages 63–75, 2003.

[41] Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. End-to-end memory networks. In *NeurIPS*, pages 2440–2448, 2015.

[42] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *CVPR*, pages 1701–1708, 2014.

[43] Qingyang Tan, Lin Gao, Yu-Kun Lai, and Shihong Xia. Variational autoencoders for deforming 3D mesh models. In *CVPR*, pages 5841–5850, 2018.

[44] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *ICCV*, pages 3735–3744, 2017.

[45] Justus Thies, Michael Zollhöfer, Matthias Nießner, Levi Valgaerts, Marc Stamminger, and Christian Theobalt. Realtime expression transfer for facial reenactment. *TOG*, 34(6):183:1–183:14, 2015.

[46] Anh Tuan Tran, Tal Hassner, Iacopo Masi, Eran Paz, Yuval Nirkin, and Gérard G. Medioni. Extreme 3D face reconstruction: Seeing through occlusions. In *CVPR*, pages 3935–3944, 2018.

[47] Edgar Tretschk, Ayush Tewari, Michael Zollhöfer, Vladislav Golyanik, and Christian Theobalt. DEMEA: deep mesh autoencoders for non-rigidly deforming objects. In *ECCV*, pages 601–617, 2020.

[48] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, pages 5998–6008, 2017.

[49] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *ICLR*, 2018.

[50] Nitika Verma, Edmond Boyer, and Jakob Verbeek. Dynamic filters in graph convolutional networks. *CoRR*, abs/1706.05206, 2017.

[51] Huawei Wei, Shuang Liang, and Yichen Wei. 3D dense face alignment via graph convolution networks. *CoRR*, abs/1904.05562, 2019.

[52] Ruben Wiersma, Elmar Eisemann, and Klaus Hildebrandt. CNNs on surfaces using rotation-equivariant features. *TOG*, 39(4):92, 2020.

[53] Fei Yang, Jue Wang, Eli Shechtman, Lubomir D. Bourdev, and Dimitris N. Metaxas. Expression flow for 3D-aware face component transfer. *TOG*, 30(4):60, 2011.

[54] Zhitao Ying, Jiaxuan You, Christopher Morris, Xiang Ren, William L. Hamilton, and Jure Leskovec. Hierarchical graph representation learning with differentiable pooling. In *NeurIPS*, pages 4805–4815, 2018.

[55] Yu-Jie Yuan, Yu-Kun Lai, Jie Yang, Qi Duan, Hongbo Fu, and Lin Gao. Mesh variational autoencoders with edge contraction pooling. In *CVPR Workshops*, pages 1105–1112, 2020.

[56] Yi Zhou, Chenglei Wu, Zimo Li, Chen Cao, Yuting Ye, Jason M. Saragih, Hao Li, and Yaser Sheikh. Fully convolutional mesh autoencoder using efficient spatially varying kernels. In *NeurIPS*, 2020.