electronics

Article

Deep Spectral Meshes: Multi-Frequency Facial Mesh Processing with Graph Neural Networks

Robert Kosk [1,2,*], Richard Southern [1], Lihua You [1], Shaojun Bian [2,3], Willem Kokke [2] and Greg Maguire [4]

1. Centre for Digital Entertainment, National Centre for Computer Animation, Bournemouth University, Poole BH12 5BB, UK; rsouthern@bournemouth.ac.uk (R.S.); lyou@bournemouth.ac.uk (L.Y.)
2. Humain Ltd., Belfast BT1 2LA, UK; shaojun@humain-studios.com (S.B.); willem@humain-studios.com (W.K.)
3. School of Creative and Digital Industries, Buckinghamshire New University, High Wycombe HP11 2JZ, UK
4. Belfast School of Art, Ulster University, Belfast BT15 1ED, UK; g.maguire@ulster.ac.uk

* Correspondence: rkosk@bournemouth.ac.uk

Abstract: With the rising popularity of virtual worlds, the importance of data-driven parametric models of 3D meshes has grown rapidly. Numerous applications, such as computer vision, procedural generation, and mesh editing, vastly rely on these models. However, current approaches do not allow for independent editing of deformations at different frequency levels. They also do not benefit from representing deformations at different frequencies with dedicated representations, which would better expose their properties and improve the generated meshes' geometric and perceptual quality. In this work, spectral meshes are introduced as a method to decompose mesh deformations into low-frequency and high-frequency deformations. These features of low- and high-frequency deformations are used for representation learning with graph convolutional networks. A parametric model for 3D facial mesh synthesis is built upon the proposed framework, exposing user parameters that control disentangled high- and low-frequency deformations. Independent control of deformations at different frequencies and generation of plausible synthetic examples are mutually exclusive objectives. A Conditioning Factor is introduced to leverage these objectives. Our model takes further advantage of spectral partitioning by representing different frequency levels with disparate, more suitable representations. Low frequencies are represented with standardised Euclidean coordinates, and high frequencies with a normalised deformation representation (DR). This paper investigates applications of our proposed approach in mesh reconstruction, mesh interpolation, and multi-frequency editing. It is demonstrated that our method improves the overall quality of generated meshes on most datasets when considering both the L_1 norm and perceptual Dihedral Angle Mesh Error (DAME) metrics.

Keywords: shape modelling; spectral meshes; multi-frequency deformations; graph neural networks

check for updates

Citation: Kosk, R.; Southern, R.; You, L.; Bian, S.; Kokke, W.; Maguire, G. Deep Spectral Meshes: Multi-Frequency Facial Mesh Processing with Graph Neural Networks. *Electronics* **2024**, *13*, 720. https://doi.org/10.3390/electronics13040720

Academic Editors: Moez Bouchouicha and Eric Moreau

Received: 11 December 2023
Revised: 31 January 2024
Accepted: 2 February 2024
Published: 9 February 2024

1. Introduction

The importance of generating high-fidelity digital humans is more relevant now than ever before. Not only are we seeing significant investments in new visual effects-intensive creative content from streaming services, but platforms like *Meta* are also reported to have invested at least USD 36 billion in developing content and tools to populate the next-generation social media platforms based on Virtual Reality. The availability of high-quality facial scans and recent advances in deep learning methods applied to mesh processing have led to the development of data-driven parametric models. These approaches are at the forefront of content creation technologies, allowing artists and users to rapidly generate high-fidelity character assets for these applications. However, parameters exposed by deep learning approaches may not always be meaningful. Furthermore, the perceptual and geometric quality of assets generated by neural models often does not meet standards appropriate for industrial applications.

The above issues can be addressed with suitable representations, which can describe deformations at different frequencies and independently edit deformations at these different frequency levels. This topic has not been investigated in existing work, as discussed below.

Polygon meshes [1], including quad and triangle meshes, are the most popular surface representation and are widely applied in the generation of creative content. They represent 3D surfaces with geometric vertices as absolute coordinates, making it challenging to edit overall shape deformations while preserving local surface details. In contrast, differential coordinates [2], often called the Laplacian operator or Laplacian coordinates, explicitly describe local surface and enable global shape editing while preserving local deformations [3]. Since differential coordinates are only translation-invariant and not scale- and rotation-invariant, some improvements have been made to extend differential coordinates with these additional properties, as reviewed in Section 2.1. Other mesh operators [4] have also been introduced to extend the Laplacian operator. The Laplace–Beltrami operator, a generalisation of the Laplace operator, is used in this paper to represent mesh deformations.

Although mesh operators enable global shape editing while preserving local deformations, they still cannot represent deformations at different frequencies or independently edit deformations at different frequency levels. Spectral mesh processing [5] derives eigenvalues, eigenvectors, or eigenspace projections from the mesh operators and uses them to carry out desired tasks. It provides a powerful means to achieve different approximations of a 3D mesh with different frequencies. In this paper, spectral mesh processing is used to decompose mesh deformations into low- and high-frequency deformations.

Three-dimensional meshes are non-Euclidean data, unlike Euclidean data such as voxels, which have an underlying grid structure and can be treated by extending already-existing 2D deep learning paradigms. The lack of grid structure poses a challenge when attempting to apply classical deep learning techniques to non-Euclidean data. To address this problem, geometric deep learning [6] has been developed explicitly for non-Euclidean data. Consequently, geometric deep learning is used in the proposed method to learn the latent parameters of low- and high-frequency deformations.

Parametric models, such as 3D morphable models [7], are commonly used to synthesise new meshes by altering the coefficients in a parametric space. They are widely applied due to their ability to model intrinsic properties of 3D faces. As reviewed in Section 2.4, parametric models have been used in graph neural networks to represent facial shapes. A parametric model is also used in this paper for 3D facial mesh synthesis.

As the most popular 3D structure for representing 3D models, triangle meshes are graphs. Graph neural networks are suitable for the geometric learning of triangle meshes [8]. Therefore, triangle meshes are considered in our proposed methods, and graph neural networks are used for deep learning.

By integrating the above-discussed Laplace–Beltrami operator for deformation representation, spectral mesh processing for decomposition of mesh deformations, geometric deep learning, and parametric models with graph neural networks, a new 3D facial mesh synthesis model is developed in this paper. The model exposes user parameters to control disentangled low- and high-frequency deformations, generate plausible facial shapes, and allow the user to control deformations independently at low- and high-frequency levels.

Generating plausible facial shapes and allowing controllable deformation can conflict, as plausible faces exist within a joint distribution of low- and high-frequency information. For example, wrinkled skin at higher frequencies correlates with volume loss of fat pads at lower frequencies. This problem is exacerbated when using smaller or biased datasets, which can lead to undesirable correlations. A Conditioning Factor is introduced to overcome the conflict between plausibility and user control. It is a scalar that modulates the influence of mutual conditioning of low- and high-frequency deformations.

Lower-frequency deformations encode most of the mesh volume, while higher-frequency deformations describe fine surface details. This observation is used to improve the quality of generated meshes. Low frequencies are represented with standardised Euclidean coordinates, which capture first-order mesh properties. The highest frequency deformation is represented with normalised deformation representation (DR) to improve perceptual quality. In this way, our proposed method improves the overall quality of generated meshes from geometric and perceptual perspectives, as shown in Section 7.

Our main contributions are:

- The introduction of spectral decomposition of meshes in 3D shape representation learning.
- A novel parametric deep face model which enables independent control of high- and low-frequency deformations.
- Enhanced geometric and perceptual quality of generated meshes, achieved through the use of different representations of deformations at high and low frequencies.

The remaining parts of this paper are organised as follows. Related work is reviewed in Section 2. Section 3 offers an overview of the proposed approach. Deep spectral meshes are described in Section 4. The impacts of the Conditioning Factor are discussed in Section 5. The implementation of the proposed approach is detailed in Section 6. Section 7 investigates applications and comparisons. Finally, Section 8 provides a summary and identifies future work directions.

2. Related Work

The work described in this paper is related to differential coordinate-based 3D shape representations, spectral mesh processing, geometric deep learning in the spectral domain, and parametric models with graph networks. This section offers a brief review of the existing work in these fields.

2.1. Differential Coordinate-Based 3D Shape Representation

A translation-invariant differential representation for surface reconstruction is reviewed in [5], which uses Laplacian coordinates, also called differential coordinates, to represent surfaces. Laplacian coordinates allow for mesh editing and shape approximation. The paper also reviews the work of extending the translation-invariant differential representation to rotation-invariant differential representations.

Gao et al. propose a new differential representation called rotation-invariant mesh difference (RIMD) [9]. Unlike the translation-invariant differential representation, it encodes deformations and is invariant to rigid rotations and translations. RIMD is defined on both vertices and edges. Therefore, it is incompatible with our method, which requires a vertex-based representation. Gao et al. also propose an as-consistent-as-possible (ACAP) [10] deformation representation as an improvement of computational issues of RIMD and its ability to handle large rotations. This deformation representation has also been used in [11] to develop a mesh-based variational autoencoder architecture for dealing with meshes with irregular connectivity and non-linear deformations.

The deformation representation proposed in [10] requires the orientations of rotation axes and rotation angles of adjacent vertices to be as consistent as possible. However, in the context of facial deformations, local rotations never exceed 2π rad. Wu et al. address this and use less complex deformation representation (DR) based on the deformation gradient. DR has been successful in reconstructing caricatures of faces through the combination and extrapolation of deformation features [12].

Quantised DR is chosen to represent high-frequency deformations in our proposed approach. Similarly to other differential representations, quantised DR preserves local surface properties. Moreover, it is compatible with our method, which requires the features to be defined on vertices.

2.2. Spectral Mesh Processing

Various methods of spectral mesh processing have been developed to address the problems of shape analysis, mesh simplification, surface segmentation, and shape correspondence. Sorkine [5] and Zhang et al. [4] comprehensively review existing work until 2005 and 2010, respectively. Recent work in the field is reviewed in the following paragraphs.

Melzi et al. obtain smooth, local, controllable, orthogonal, and efficient basis called Localized Manifold Harmonics (LMH) through the spectral decomposition of new types of intrinsic operators. The authors integrate local details provided by the basis and the global information from the Laplacian eigenfunctions to deal with local spectral shape analysis [13]. Xu et al. fit the original 3D model with a finite subdivision surface and restrict the eigenproblem with a subdivision linear subspace to obtain intrinsic shape information of 3D models through fast calculation of large-scale Laplace–Beltrami eigenproblem on the models [14]. Lescoat et al. propose a new mesh simplification algorithm, which uses a spectrum-preserving mesh decimation scheme to simplify input meshes and makes the Laplacian of simplified meshes spectrally close to the Laplacian of input meshes [15].

Wang et al. use a Laplacian operator and select and combine some of its sub-eigenvectors to compute a single segmentation field to conduct mesh segmentation [16]. Tong et al. build a Laplacian matrix, propose a spectral mesh segmentation method, which converts mesh segmentation into an ℓ_0 gradient minimisation problem, and devise a fast algorithm to solve the minimisation problem [17]. Bao et al. devise a feature-aware simplification algorithm to create a coarse mesh. Then, they use the spectral segmentation method proposed in [17] to perform partition on the coarse mesh to obtain a coarse segmentation. Next, they reverse the simplification process to map the coarse mesh to the input mesh and smooth jaggy boundaries to develop a spectral segmentation method for large meshes [18].

Jain and Zhang first transform two 3D meshes into spectral domain and find and match the embeddings of the two 3D meshes in the spectral domain to obtain vertex-to-vertex correspondence between the two 3D meshes [19]. Dubrovina and Kimmel use the eigenfunctions of the Laplace–Beltrami operator to calculate surface descriptors and match surface descriptors to develop a correspondence detection method [20]. Melzi et al. introduce iterative spectral upsampling to obtain high-quality correspondences with a small number of coefficients in the spectral domain [21].

Spectral methods share a common framework. They define a matrix representing a discretisation of a continuous operator over a mesh. In our case, it is the Laplace–Beltrami operator defined in Equation (4). Then, the matrix is decomposed to obtain eigenvectors and eigenvalues. These are used to solve problems in a specific domain. This paper introduces the concept of spectral mesh processing to 3D shape representation learning. Through the spectral decomposition of meshes, our method benefits from the ability to use different representations for low- and high-frequency data.

2.3. Geometric Deep Learning in Spectral Domain

Deep neural networks have proven to be powerful tools in extracting latent representations of data in the Euclidean domain, especially 2D images. In recent years, these deep learning approaches extended to irregular graphs [6,8,22] and allowed for learning non-linear, lower-dimensional embeddings of meshes, including human faces. There are many publications on geometric deep learning. The current research on geometric deep learning in the spectral domain is briefly reviewed in the following paragraph.

Dong et al. propose a convolutional neural network framework called Laplacian2Mesh by mapping 3D meshes to a multi-dimensional Laplacian–Beltrami space to deal with irregular triangle meshes for shape classification and segmentation [23]. Lemeunier et al. integrate spectral mesh processing and deep learning models consisting of a convolutional autoencoder and a transformer to develop a spectral transformer called SpecTrHuMS to generate human mesh sequences [24]. Qiao et al. present a deep learning approach that uses Laplacian spectral clustering to build a fine-to-coarse mesh hierarchy and integrate Laplacian spectral analysis and mesh feature aggregation blocks to encode mesh connectivity for shape segmentation and classification [25]. Based on the idea of approximating low and mid frequencies on coarse grids, Nasikun and Hildebrandt investigate a new solver called the Hierarchical Subspace Iteration Method that can solve sparse Laplace–Beltrami eigenproblems on meshes faster than existing methods based on Lanczos iterations, preconditioned conjugate gradients, and subspace iterations [26].

2.4. Parametric Models with Graph Networks

Work [27] is the first geometric learning approach to representing facial shapes. It utilises a variational autoencoder architecture with spectral graph convolutional operations [28]. Models [29–33] improve this architecture with custom convolutional and aggregation operators. Cheng et al. introduce MeshGAN, the adversarial approach to learning the facial manifold of 3D meshes [34]. Zhou et al. [35] jointly learn deformation offset and colour residing on vertices of a face mesh [35]. Yuan et al. convolve preprocessed ACAP deformation features [36]. Jiang et al. learn disentangled manifolds of facial identities and expressions in the normalised deformation representation (DR) [37]. Based on the deformation representation proposed in [10], a new variational autoencoder architecture is proposed in [38] to learn a latent representation of 3D human hands for high-accuracy monocular RGB-D/RGB 3D hand reconstruction.

The decomposition proposed in this paper is deformation decomposition, which decomposes a deformation into low- and high-frequency parts. It differs from the attribute decomposition proposed in [37], where a 3D face shape is decomposed into identity and expression parts. However, both this paper and [37] adopt the same deformation representation proposed in [10]. The deep spectral mesh graph neural network in our proposed method also differs from the variational autoencoder architecture in [38]. While [38] encodes and decodes ACAP deformation features using fully connected layers, our proposed approach separately encodes the decomposed deformation using graph convolutional layers and subsequently decodes the decomposed deformation from concatenated latent codes.

As discussed above, none of the parametric models allow for independent editing of deformations at different frequencies. Additionally, these models represent a whole spectrum of deformations with only one representation. Therefore, unlike our proposed approach, they cannot benefit from defining deformations at several frequency bands with different, dedicated representations.

3. Method Overview

Our proposed method is outlined in Figure 1. It consists of three steps: spectral mesh processing, neural networks, and final assembly.

In the first step, preprocessed meshes $\mathbf{P}$ are partitioned into two frequency bands through spectral decomposition. The resulting low- and high-frequency deformations $\mathbf{P}_{low}$ and $\mathbf{P}_{high}$ are transformed to standardised Euclidean coordinate features $\mathbf{F}_{low}$ and normalised deformation representation (DR) features $\mathbf{F}_{high}$. The spectral decomposition and representations of features are discussed in Section 3.1 below.

In the second step, a neural network consisting of graph encoders and graph decoders is used to reconstruct input features. Graph encoders E_{high} and E_{low} encode the features to latent means $\boldsymbol{\mu}_{high}$ and $\boldsymbol{\mu}_{low}$ and deviations $\boldsymbol{\sigma}_{high}$ and $\boldsymbol{\sigma}_{low}$. Means and deviations are concatenated, and latent codes $\mathbf{Z}$ are sampled from the distribution ($[\boldsymbol{\mu}_{high} \mid \boldsymbol{\mu}_{low}]$,$[\boldsymbol{\sigma}_{high} \mid \boldsymbol{\sigma}_{low}]$). Graph decoders D_{high} and D_{low} reconstruct input features from $\mathbf{Z}$ and output reconstructed features $\mathbf{F}'_{high}$ and $\mathbf{F}'_{low}$. The neural network is introduced in Section 3.2.

In the third step, output features $\mathbf{F}'_{high}$ and $\mathbf{F}'_{low}$ are converted back from their representations to Euclidean coordinates $\mathbf{P}'_{high}$ and $\mathbf{P}'_{low}$, which are later combined to obtain the final vertex positions, $\mathbf{P}'$, as described in Section 3.3.

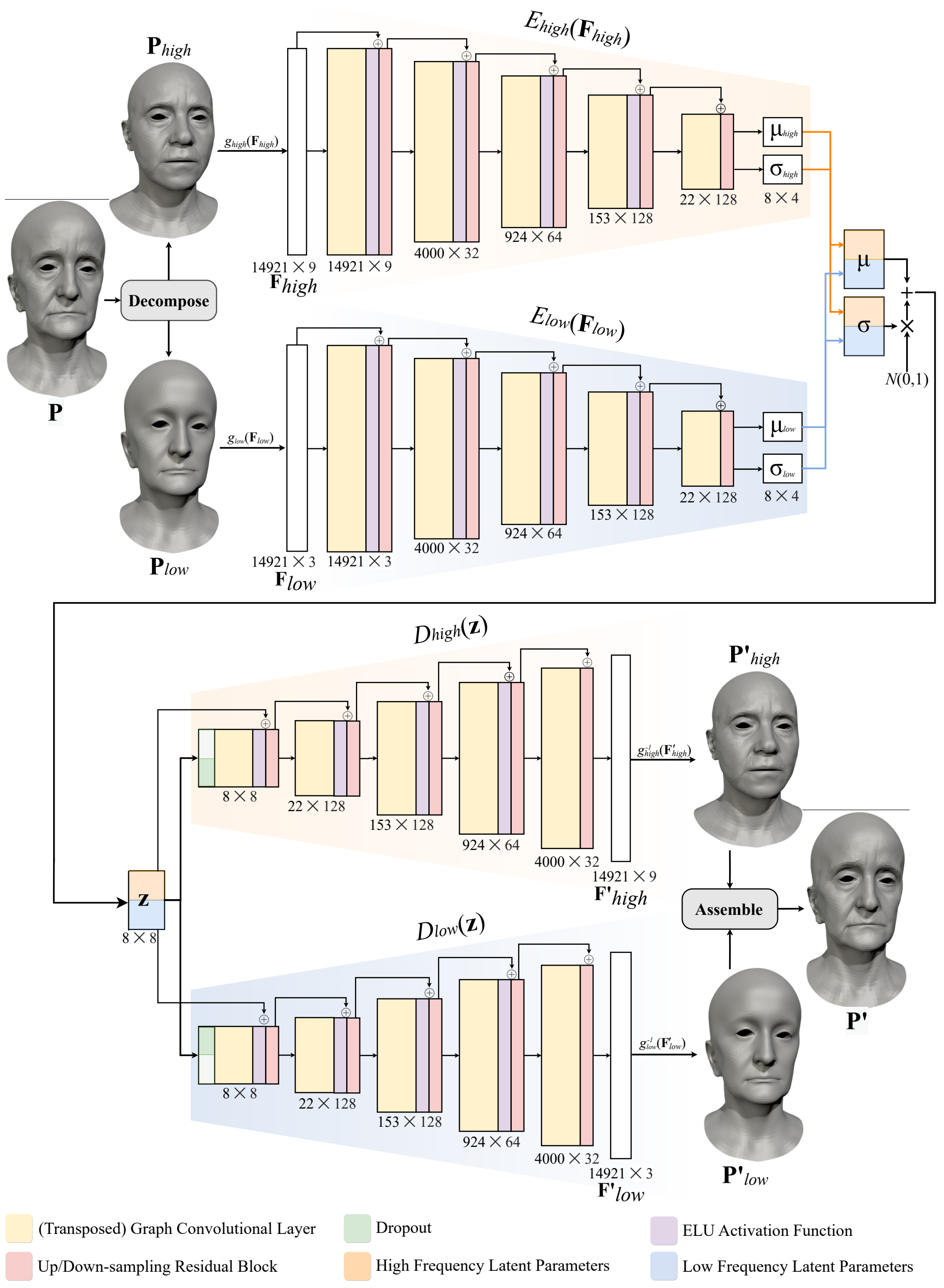


Figure 1. Overview of our Deep Spectral Meshes graph neural network.

3.1. Spectral Partitioning and Representation

The generalised spectral decomposition [4] of a Laplacian matrix, $\mathbf{L}$, has a form of

$$\mathbf{L} = \mathbf{U}\Lambda\mathbf{U}^T\mathbf{M}, \tag{1}$$

where $\mathbf{M}$ is the mass matrix representing Voronoi areas around vertices. The use of mass matrix $\mathbf{M}$ counters the impact of irregular tessellation and, in consequence, improves the geometric quality of meshes generated by our method.

Spectral partitioning is the process of identifying n subsets of eigenvectors $\mathbf{U}$ and projecting vertex positions $\mathbf{P}$ of a mesh onto these subsets of vectors to isolate different frequency bands $\{\mathbf{P}_0, ..., \mathbf{P}_n\}, \mathbf{P} = \sum_{i=0}^{n} \mathbf{P}_i$. The choice of size and number of frequency bands is arbitrary and depends on a specific application. Projection of $\mathbf{P}$ onto a subset of eigenvectors $\mathbf{U}_i$ is defined as

$$\mathbf{P}_i = \mathbf{U}_i \mathbf{U}_i^T \mathbf{M} \mathbf{P}. \tag{2}$$

As a consequence of spectral partitioning, frequency bands $\{\mathbf{P}_0, ..., \mathbf{P}_n\}$ can be transformed into different, dedicated representations, which are more suitable for a given spectral band. A representation transform can be denoted as function $g_i(\mathbf{P}_i)$, which transforms the ith frequency band into representation $\mathbf{F}_i$. An associated inverse transform can be denoted as $g_i^{-1}(\mathbf{F}_i)$. These functions must output a per-vertex representation because $\mathbf{F}_i$ are inputs to graph neural networks which process input signal defined on vertices.

3.2. Neural Network

Variational graph autoencoders [32] are used to encode features of each frequency band into their respective latent distributions $\mathcal{N}(\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i)$, where $\boldsymbol{\mu}_i$ is a mean vector and $\boldsymbol{\sigma}_i$ is a standard deviation vector. If desired, each encoder can output a different number of parameters. The optimal size of the parametric space depends on the training dataset and requirements of a specific application. Therefore, it can be determined as part of the hyperparameter optimisation process.

The neural network consists of n graph encoders E_i with n corresponding decoders D_i. The choice of convolutional, transpose convolutional, pooling, de-pooling or other layers of encoders and decoders is arbitrary. While each encoder outputs a different latent distribution of its frequency band, the encoders take the same set of parameters as input. Specifically, mean vectors are concatenated to $\boldsymbol{\mu} = [\boldsymbol{\mu}_0, ..., \boldsymbol{\mu}_n]$ and deviation vectors are concatenated to $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_0, ..., \boldsymbol{\sigma}_n]$. Afterwards, latent parameters $\mathbf{Z}$ are stochastically sampled from $\mathcal{N}(\boldsymbol{\mu}, \boldsymbol{\sigma})$. The decoders are optimised to output reconstructed features $\mathbf{F}_i'$, with the objective of minimising the discrepancy between $\mathbf{F}_i$ and $\mathbf{F}_i'$.

3.3. Final Assembly

Before combining reconstructed vectors $\mathbf{F}_i'$, it is necessary to convert them back to Euclidean coordinates so that the reconstructed frequency band $\mathbf{P}_i' = g_i^{-1}(\mathbf{F}_i')$. It can be observed that each decoder D_i is optimised to generate outputs which, after transformation with $g_i^{-1}(\cdot)$, are a linear combination of eigenvectors $\mathbf{U}_i$. This implies that $\mathbf{P}_i'$ can be projected onto these vectors without information loss, as the filtered-out frequencies are noise which lies outside the domain for this frequency band. Based on this observation, the final reconstructed vertex positions are obtained through the aggregation of frequency bands $\mathbf{P}_i'$ in the following way:

$$\mathbf{P}' = \sum_{i=0}^{n} \mathbf{U}_i \mathbf{U}_i^T \mathbf{M} \mathbf{P}_i' \quad . \tag{3}$$

4. Deep Spectral Meshes

This section explains the details of our parametric deep face model, which uses the spectral decomposition of meshes. An overview of the model is provided in Figure 1.

4.1. Vertex Feature Representation

Given a dataset of triangle meshes with shared connectivity, i.e., with the same triangle-vertex index incidence table to associate each triangle with its three bounding vertices, the triangle meshes are translated to make their centre coincide with the origin. Then, their mean is computed, which is an averaged shape of all triangle meshes denoted as $\bar{\mathbf{P}}$. Next, each triangle mesh is translated and rotated to align with the mean. Following this, the triangle meshes are uniformly scaled to fit within a cube with sides measuring two units each.

4.1.1. Spectral Decomposition

Laplacian matrix $\mathbf{L}$ in Equation (1) consists of elements $\mathbf{L}_{ij}$, which are determined by the following cotangent Laplacian operator used in [2]:

$$\mathbf{L}_{ij} = \begin{cases} j \in \mathcal{N}_i & \cot\alpha_{ij} + \cot\beta_{ij} \\ j \notin \mathcal{N}_i & 0 \\ i = j & -\sum_{k \neq j} \mathbf{L}_{ik} \end{cases}, \tag{4}$$

where $\mathcal{N}_i$ are vertices ..., $j-1$, j, $j+1$, ... in the one-ring neighbourhood of vertex i and α_{ij} and β_{ij} are the angles opposite to edge ij, as shown in Figure 2.

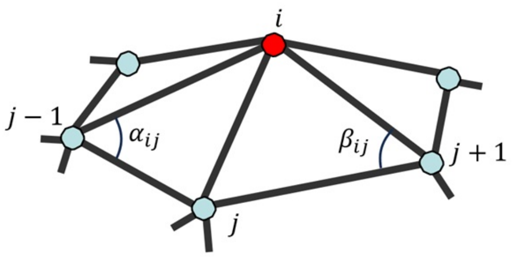


Figure 2. One-ring neighbourhood vertices and the angles used to calculate cotangent weights.

For a large triangle mesh, Laplacian matrix $\mathbf{L}$ in Equation (1) has many eigenvectors. Calculating all the eigenvectors of such Laplacian matrix is slow. To tackle this problem, only the first k eigenvectors are calculated. Meshes are partitioned into two frequency bands: low and high frequencies. The former band is defined by the first k eigenvectors $\mathbf{U}_{(k)}$, which correspond to the k smallest eigenvalues of Laplacian matrix $\mathbf{L}$. Section 4.1.2 covers the high-frequency band.

We let $\hat{\mathbf{P}}$ be a deformation from a mean so that $\hat{\mathbf{P}} = \mathbf{P} - \bar{\mathbf{P}}$. As eigenvectors $\mathbf{U}_{(k)}$ correspond to the lowest eigenvalues, they form a discrete space of slowly changing values. Therefore, by projecting $\hat{\mathbf{P}}$ onto space $\mathbf{U}_{(k)}$, the resulting vertex positions $\mathbf{P}_{\text{low}}$ are $\hat{\mathbf{P}}$, which is a mean deformed by only low-frequency deformations. $\mathbf{P}_{\text{high}}$ represents the remaining high-frequency deformations of a mean $\bar{\mathbf{P}}$. Meshes $\mathbf{P}_{\text{low}}$ and $\mathbf{P}_{\text{high}}$ are computed as follows:

$$\begin{aligned} \mathbf{X} &= \mathbf{U}_{(k)}\mathbf{U}_{(k)}^T\mathbf{M}, \\ \mathbf{P}_{\text{low}} &= \mathbf{X}\hat{\mathbf{P}} + \bar{\mathbf{P}}, \\ \mathbf{P}_{\text{high}} &= (\mathbf{I} - \mathbf{X})\hat{\mathbf{P}} + \bar{\mathbf{P}}. \end{aligned} \tag{5}$$

4.1.2. High-Frequency Band

High-frequency band $\mathbf{P}_{high}$ is encoded in a normalised deformation representation (DR) [10,12]. This representation encodes per-vertex deformation gradient $\mathbf{T}_i$ between the position of vertex $\mathbf{p}_i$ on mean $\bar{\mathbf{P}}$ and the position of deformed vertex $\mathbf{p}'_i$ on $\mathbf{P}_{high}$. Following [39,40], deformation gradient $\mathbf{T}_i$ is calculated by solving the following weighted least-squares system, which minimises energy $E(\mathbf{T}_i)$ such that

$$E(\mathbf{T}_i) = \sum_{j \in \mathcal{N}_i} c_{ij} \left\| (\mathbf{p}'_i - \mathbf{p}'_j) - \mathbf{T}_i(\mathbf{p}_i - \mathbf{p}_j) \right\|^2, \tag{6}$$

where c_{ij} are cotangent weights calculated on a mean of the training meshes.

As linear interpolation of matrices $\mathbf{T}_i$ is meaningless, they are decomposed into a rotational part and a scale/shear part using polar decomposition so that $\mathbf{T}_i = \mathbf{R}_i\mathbf{S}_i$. The rotation matrix $\mathbf{R}_i$, mapped to $\log \mathbf{R}_i$, can be linearly interpolated and then converted back to $\mathbf{R}_i = \exp(\log \mathbf{R}_i)$. Finally, identity matrix $\mathbf{I}$ is subtracted from $\mathbf{S}_i$.

Although both per-vertex matrices $\mathbf{R}_i$ and $\mathbf{S}_i$ contain nine elements each, only six non-trivial scale elements and three non-trivial rotation elements are combined to construct deformation features $\mathbf{f}_i$, where $|\mathbf{f}_i| = 9$. The results of our experiments, shown in Section 7, demonstrate that the normalisation of DR results in a lower reconstruction error of test data. Consequently, each channel is normalised to range $[-1, 1]$.

4.1.3. Low-Frequency Band

Meshes $\mathbf{P}_{\text{low}}$ remain in the Euclidean coordinate representation. Based on our experiments in Section 7, inputs $\mathbf{F}_{low}$ are standardised by subtraction of the mean and division by standard deviation.

4.2. Graph Network Architecture

Our network consists of two fully convolutional variational graph autoencoders, as depicted in Figure 1. Encoders E_{high} and E_{low} take as input high-frequency features $\mathbf{F}_{high}$ and low-frequency features $\mathbf{F}_{low}$, respectively (see Section 4.1). The encoders compress each input feature to compact distributions with means $\boldsymbol{\mu}_{high}$ and $\boldsymbol{\mu}_{low} \in \mathbb{R}^{8\times4}$, and deviation vectors $\boldsymbol{\sigma}_{high}$ and $\boldsymbol{\sigma}_{low} \in \mathbb{R}^{8\times4}$. Subsequently, output means and deviations are concatenated so that $\boldsymbol{\mu} = [\boldsymbol{\mu}_{high} \mid \boldsymbol{\mu}_{low}]$ and $\boldsymbol{\sigma} = [\boldsymbol{\sigma}_{high} \mid \boldsymbol{\sigma}_{low}]$.

Latent code $\mathbf{Z} \in \mathbb{R}^{8\times8}$ is obtained in a stochastic process. First, standard deviation $\boldsymbol{\sigma}$ is multiplied with scalar $\boldsymbol{\epsilon}$ drawn from a normal distribution. Then, the result is added to mean vector $\boldsymbol{\mu}$ so that $\boldsymbol{\sigma} \times \boldsymbol{\epsilon} + \boldsymbol{\mu} = \mathbf{Z} = [\mathbf{Z}_{high} \mid \mathbf{Z}_{low}]$.

Both decoders, E_{high} and E_{low}, take $\mathbf{Z}$ as input. This way, E_{high} decodes high-frequency deformation features $\mathbf{F}'_{high}$ from code $\mathbf{Z}_{high}$ conditioned on code $\mathbf{Z}_{low}$. Analogically, E_{low} decodes low-frequency deformation features $\mathbf{F}'_{low}$ from code $\mathbf{Z}_{low}$ conditioned on code $\mathbf{Z}_{high}$. Controlling the influence of this conditioning is discussed in Section 5.

4.3. Network Structure

Both variational graph autoencoders have the same structure. They use convolution/transposed convolution operations and upsampling/downsampling residual layers introduced in [32]. However, other types of graph convolutional and pooling operators can be used instead if they accept input features defined solely on vertices. Therefore, the encoders and decoders operating on signal defined on edges would not be suitable, as our method uses low- and high-frequency features defined on vertices.

The encoders put their inputs through five graph convolutional layers. The local convolution kernels are sampled from the global kernel weight basis to perform convolutional operations on irregular graphs. Therefore, the network learns the shared global kernel weight basis and per-vertex sampling functions [32]. We use stride = 2, kernel radius = 2, weight basis = 35, and channel dimensions = [$|\mathbf{f}|$, 32, 62, 128, 128, 4]. All layers, except the last one, are followed by the *ELU* activation function. The outputs from the *ELU* are added to outputs from a downsampling residual layer. The decoders mirror the encoders with five blocks of graph transposed convolutional layers followed by the *ELU*s and upsampling residual layers.

4.4. Training Process

The two variational graph autoencoders are trained simultaneously. At each iteration, the weights and biases of each encoder–decoder pair are updated through backpropagation using two separate Adam optimisers. Learnable parameters of E_{high} and D_{high} are optimised in terms of loss L_{high}, while parameters of E_{low} and D_{low} are updated in terms of loss L_{low}. Losses are calculated as follows:

$$\begin{aligned} L_{high} = {} & ||\mathsf{denorm}(\mathbf{F}_{high}) - \mathsf{denorm}(D_{high}(\mathbf{Z}))||_1 \\ & + \phi \mathsf{KL}(\mathcal{N}(0,1)||p(\mathbf{Z}_{high}|\mathbf{F}_{high})), \end{aligned} \tag{7}$$

$$L_{low} = ||\mathsf{destd}(\mathbf{F}_{low}) - D_{low}(\mathbf{Z})||_1 \\ +\phi \mathsf{KL}(\mathcal{N}(0,1)||p(\mathbf{Z}_{low}|\mathbf{F}_{low})). \tag{8}$$

In other words, both losses are a sum of two terms. The first one is the L_1 norm of a difference between the ground truth feature and its reconstruction. The second one is weighted Kullback–Leibler (KL) divergence, which measures a difference between normal distribution $\mathcal{N}(0,1)$ and latent vector distribution, where ϕ is a scalar weight.

Since D_{high} outputs the normalised DR features (see Section 4.1.2), the output, $\mathbf{F}'_{high}$, as well as the ground truth, $\mathbf{F}_{high}$, are denormalised before calculating the L_1 norm of a difference between them. Denormalisation is denoted with the $\mathsf{denorm}(\cdot)$ function. In the case of D_{low}, the decoder outputs a destandardised feature in Euclidean coordinates. Therefore, only ground truth feature $\mathbf{F}_{low}$ is destandardised before computing the L_1 norm term. Here, destandardisation is denoted with the $\mathsf{destd}(\cdot)$ function.

4.5. Inference

This section describes the postprocessing steps required to obtain reconstructed vertex positions $\mathbf{P}'$. To be a valid input to the network, every mesh with vertex positions $\mathbf{P}$ must be represented with features $\mathbf{F}_{high}$ and $\mathbf{F}_{low}$, following the process described in Section 4.1. At inference, the network generates normalised DR features $\mathbf{F}'_{high}$ and Euclidean coordinates $\mathbf{F}'_{low} = \mathbf{P}'_{low}$. To calculate $\mathbf{P}'_{high}$, $\mathbf{F}'_{high}$ is denormalised and converted back from the normalised DR representation to Euclidean coordinate representation, as in [12]. Subsequently, $\mathbf{P}'_{high}$ and $\mathbf{P}'_{low}$ are combined in the following way:

$$\mathbf{P}' = (\mathbf{I} - \mathbf{X})\mathbf{P}'_{high} + \mathbf{X}\mathbf{P}'_{low}, \tag{9}$$

where $\mathbf{X}$ is the matrix from Equation (5).

5. Conditioning Influence

Conditioning, described in Section 4.2, allows for the network to generate plausible facial meshes from a joint distribution of high- and low-frequency information. However, it can be desirable to tame the influence of conditioning to increase artistic control and prevent overfitting.

In an extreme case, the impact of conditioning could be removed if all the conditioning parameters were set to a constant value, for example, zero. On the other hand, if only a pseudo-randomly drawn fraction of these parameters was set to zero at each training iteration, the impact of conditioning would lower. It is proposed to achieve this effect by applying a dropout [41] to the conditioning parameters.

Specifically, at each iteration, conditioning parameters $\mathbf{Z}_{low}$ of decoder D_{high} are randomly zeroed with probability $(1-\gamma)$ using the samples from the Bernoulli distribution. The same applies to conditioning parameters $\mathbf{Z}_{high}$ of decoder D_{low}. Therefore, scalar γ can be called a Conditioning Factor, where $\gamma = 0$ prevents conditioning and $\gamma = 1$ means full conditioning.

6. Implementation Details

Our parametric models are trained with the following datasets of facial meshes: Facsimile ™ [42] (202 meshes, 14,921 vertices each), FaceScape [43] (807 meshes, 26,317 vertices each) and FaceWarehouse [44] (150 meshes, 11,510 vertices each). As all three datasets contain subjects performing a set of facial expressions, only the shapes with a neutral expression are used. All meshes within each dataset have triangular faces with consistent connectivity. In our experiments, each dataset is split into training, validation and test subsets in proportions 85:5:10.

ARPACK [45] is used to solve the generalised eigenvalue problem in Equation (1). Only the first k eigenvectors are computed. The networks are trained with $k = 500$, the Conditioning Factor $\gamma = 1.0$ (full conditioning) and $\gamma = 0.4$ (partial conditioning).

The networks are implemented in Pytorch 1.8 [46]. In all experiments, the networks are trained for 500 epochs, with a learning rate of 10^{-4} and a learning rate decay of 1% after each epoch. KL divergence weight ϕ from Equations (7) and (8) is 10^{-6}. Adam optimiser's hyper-parameters are set to $\beta_1 = 0.9$ and $\beta_2 = 0.999$. The networks trained with the Facsimile™ and FaceWarehouse datasets are optimised with a batch size of 16, while those trained with FaceScape have a batch size of 8 due to GPU memory constraints.

Matrix $\mathbf{X}$ from Equations (5) and (9) needs to be pre-computed only once. Otherwise, its frequent computation can significantly lower the performance of data preprocessing. Table 1 compares the CPU time and memory required to compute $\mathbf{U}$ and $\mathbf{X}$ on different datasets. Additionally, the table shows the CPU time of the final assembly Equation (9). This operation is performed each time a new mesh is inferred and takes 0.20 to 0.98 s per mesh, depending on the dataset used. For faster eigendecomposition of Laplacian $\mathbf{L}$ from Equation (4), approximation techniques can be used. Nevertheless, they are not used in our implementation.

Table 1. Per-mesh CPU time and CPU memory required to compute terms from Equations (5) and (9). Three datasets of different vertex count are compared: FaceWarehouse [44] (150 meshes, 11,510 verts), Facsimile™ [42] (202 meshes, 14,921 verts) and FaceScape [43] (26,317 verts).

	FaceWarehouse (11,510 verts)	**Facsimile™ (14,921 verts)**	**FaceScape (26,317 verts)**
Computation of $\mathbf{U}$ CPU time [s]	26.92	33.72	58.22
Computation of $\mathbf{X}$ CPU time [s]	2.11	7.70	9.88
Computation of Equation (9) CPU time [s]	0.20	0.36	0.98
Computation of $\mathbf{U}$ CPU memory [MB]	45.0	58.3	102.8
Computation of $\mathbf{X}$ CPU memory [GB]	1.04	1.74	5.41

All the experiments are trained and inferred on a single NVIDIA GeForce GTX 1080 Ti with 16 GB memory. Data processing is performed on an Intel(R) Xeon(R) CPU E5-1630 v4 running at 8×3.70 GHz using 32 GB RAM. Training times range between 2 h for the FaceWarehouse dataset and 19 h for the FaceScape dataset.

7. Applications and Comparisons

The approach proposed in this paper has a large number of applications. Besides those identified in Section 8, this section investigates the applications of the proposed approach in mesh reconstruction, mesh interpolation and multi-frequency mesh editing. It also compares the proposed approach with previously published methods.

7.1. Mesh Reconstruction

Mesh reconstruction can be used in representation learning, mesh compression, mesh smoothing and fairing. In this subsection, our proposed method is applied to reconstruct 3D meshes for mesh compression. For illustration, all the reconstruction experiments on the Facsimile™ dataset use a compression rate of 0.14%. These reconstruction experiments compare our method and common representations used in other methods: Euclidean coordinates, standardised Euclidean coordinates and the normalised deformation representation (DR).

Implementation of our method follows the description from Sections 4 and 6, with the Conditioning Factor $\gamma = 1.0$ and frequencies split at $k = 500$. All other representations do not decompose the mesh into deformations with multiple frequencies. Therefore, they are evaluated on the fully convolutional variational graph autoencoder with a single encoder and decoder. The encoder is the same as E_{high} or E_{low}, and the decoder is the same as D_{high} or D_{low}, without the dropout layer. All the experiments encode to latent space $\mathbf{Z}$ of 64 parameters.

7.1.1. Quantitative Evaluation

Two metrics, the point-wise L_1 norm and the Dihedral Angle Mesh Error (DAME) [47], are used to quantitatively assess the quality of meshes generated by our method. The commonly used L_1 norm $||\mathbf{P} - \mathbf{P}'||_1$ between reconstructed Euclidean coordinates $\mathbf{P}'$ and the ground truth $\mathbf{P}$ is chosen due to its sensitivity to overall shape changes. In other words, the L_1 norm is capable of capturing low-frequency error. Nonetheless, it poorly correlates with high-frequency discrepancies perceived by the human visual system [48]. Therefore, the results are also evaluated on a perceptual metric, as described below.

Given that the final meshes generated by our method are viewed by the human observer, the perceptual quality of the results is essential. Quantitative evaluation of perceptual mesh quality is often overlooked in previous work on parametric models with graph networks. In this work, DAME [47] is used to capture high-frequency discrepancies between the reconstructed and the ground truth meshes, and measure the perceptual quality of the generated results. DAME is selected due to its highest correlation with human judgement, measured on the compression task, compared to other common perceptual metrics. Moreover, DAME is dedicated to datasets with shared connectivity [48].

DAME consists of three elements: the difference between oriented dihedral angles, the masking effect and the visibility weighting. The last one is application-specific, as it changes with the viewing angles and the rendering resolution. Therefore, following the recommendation in [47], the visibility term is replaced with triangle areas. Border edges are ignored in the calculation of DAME, as oriented dihedral angles cannot be calculated on these edges.

The reconstruction results generated with our approach are compared with those output by other methods: Mesh Autoencoder [32], SpiralNet++ [49], Neural 3DMM [29] and FeaStNet [33]. Table 2 presents the quantitative outcomes of this comparison. Our proposed method consistently outperforms all counterparts in terms of the perceptual DAME metric across both the Facsimile™ and FaceWarehouse training and test datasets. Regarding point-wise accuracy measured with the L_1 norm, our method outperforms other compared methods on the FaceWarehouse training dataset. However, on the Facsimile™ training and test sets and the FaceWarehouse test dataset, our method performs less favourably than SpiralNet++ [49] and Neural 3DMM [29]. Nonetheless, a comprehensive assessment encompasses both perceptual and point-wise accuracy perspectives, and the perceptual results generated by [29,49] have significantly higher perceptual DAME error. These quantitative results are further supported by a qualitative user study in Section 7.1.2.

Table 2. Numerical comparison of the reconstruction results of the Facsimile™ and FaceWarehouse datasets using our method ($k = 500$, $\gamma = 1$, $\mathbf{Z} = 64$) and four other methods: Mesh Autoencoder [32], SpiralNet++ [49], Neural 3DMM [29] and FeaStNet [33].

	Training		**Test**	
	L_1 Norm $\times 10^{-3} \downarrow$	**DAME $\times 10^{-2} \downarrow$**	**L_1 Norm $\times 10^{-3} \downarrow$**	**DAME $\times 10^{-2} \downarrow$**
Facsimile™				
Ours	1.61	**2.76**	6.42	**3.17**
Mesh Autoencoder.	2.60	6.04	8.32	5.81
SpiralNet++	**1.35**	5.35	6.38	4.87
Neural 3DMM	1.71	3.84	**5.95**	3.81
FeaStNet	2.02	5.30	9.07	5.35
FaceWarehouse				
Ours	**0.91**	**1.10**	6.27	**1.29**
Mesh Autoencoder.	2.54	5.27	5.33	5.50
SpiralNet++	1.21	6.06	4.69	5.63
Neural 3DMM	1.58	4.84	**4.02**	4.46
FeaStNet	1.92	6.81	8.17	6.30

In Table 3, the L_1 norm and DAME metrics are used to evaluate reconstructed meshes with our proposed approach and the 3D shape representations from other methods. It is demonstrated that, across the majority of datasets, our method outperforms other methods in reconstructing examples from the training set. Reconstruction of examples seen by the network during training has applications in 3D mesh compression.

Table 3. Quantitative comparison of the reconstruction results with our method ($k = 500$, $\gamma = 1$) and with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates [27,29–31,49] and the normalised deformation representation (DR) [12,37]. To ensure a fair comparison between our method and other input representations, they are evaluated on the fully convolutional variational graph autoencoder with a single encoder and a single decoder. The encoder is the same as E_{high} or E_{low}, and the decoder is the same as D_{high} or D_{low}, without the dropout layer. All the comparisons encode to latent space **Z** of 64 parameters. Our method outperforms the reconstruction of examples from the training set on most datasets. On the test set, our method favourably compromises between the point-wise L_1 precision and the perceptual DAME metric. Other methods considerably sacrifice one of these in favour of another.

	Training		**Test**	
	L_1 **Norm** $\times 10^{-3} \downarrow$	**DAME** $\times 10^{-2} \downarrow$	L_1 **Norm** $\times 10^{-3} \downarrow$	**DAME** $\times 10^{-2} \downarrow$
Facsimile™				
Ours	**1.61**	**2.76**	6.42	3.17
DR	4.77	3.05	9.29	**3.00**
Euclidean Std.	2.36	4.90	**5.78**	3.89
Euclidean	2.60	6.04	8.32	5.81
FaceWarehouse				
Ours	**0.91**	**1.10**	6.27	1.29
DR	2.20	1.14	7.42	**1.22**
Euclidean Std.	1.11	3.23	**5.33**	2.53
Euclidean	2.54	5.27	5.34	5.50
FaceScape				
Ours	1.27	2.25	1.65	2.41
DR	6.06	**1.64**	5.92	**1.61**
Euclidean Std.	**0.96**	1.81	**1.30**	1.82
Euclidean	1.32	2.20	1.71	2.26

The reconstructions of meshes from the test set reveal a pattern. The standardised Euclidean representation achieves the lowest L_1 norm error, and the normalised DR has the lowest DAME error. Nonetheless, in the case of the Facsimile™ [42] and FaceWarehouse [44] datasets, Euclidean and standardised Euclidean coordinates perform the worst on the DAME metric. Analogously, DR has the highest L_1 norm. It can be concluded that, with these representations, either point-wise precision or perceptual quality must be sacrificed significantly.

Our method balances these metrics favourably, as demonstrated in Figure 3. The results on the Facsimile™ dataset reveal that, compared to normalised DR, our method achieves a 30.9% lower L_1 norm error, with only a 5.7% increase in DAME error. Contrasted with standardised Euclidean coordinates, our approach yields an 18.5% lower DAME error at the cost of an 11.5% higher L_1 norm error. Now, turning to the FaceWarehouse [44] dataset, our method has a 15.5% lower L_1 norm error than normalised DR, with just a 5.7% higher DAME error. In comparison to standardised Euclidean coordinates, our method results in a 49% lower DAME error at the cost of just a 17.6% higher L_1 norm error.

However, for the FaceScape [43] dataset, our method does not improve upon the reconstruction results from other representations on both training and validation sets. Scatter plots in Figure 3 reveal the reason. The proposed approach can improve the quality of reconstructed meshes by utilising dedicated mesh representations for each spectral band. Nevertheless, on the FaceScape dataset, the DR representation yields little improvement of the perceptual quality compared to standardised Euclidean representation. Therefore, our proposed method cannot benefit from using different representations in this particular case. Experiments on the FaceScape dataset demonstrate that our method can elevate the overall quality of reconstructed meshes only when there is a substantial difference between the point-wise and the perceptual accuracy yielded by at least two different mesh representations.

Table 4 compares the deformation representation (DR) with and without the normalisation preprocessing step described in Section 4.1.2. Moreover, the table contrasts Euclidean coordinates with and without the standardisation preprocessing step, as described in Section 4.1.3. Standardisation of low-frequency input features represented in Euclidean coordinates improves training and validation results across all datasets in the comparison. Normalising high-frequency input features represented with DR improves the L_1 norm on validation sets and yields a similar or a lower DAME error. The impact of normalising DR is not conclusive on the training dataset, as normalisation of DR improves the L_1 norm and DAME metrics on the FaceWarehouse dataset, while DR without normalisation yields better results on the Facsimile™ dataset.

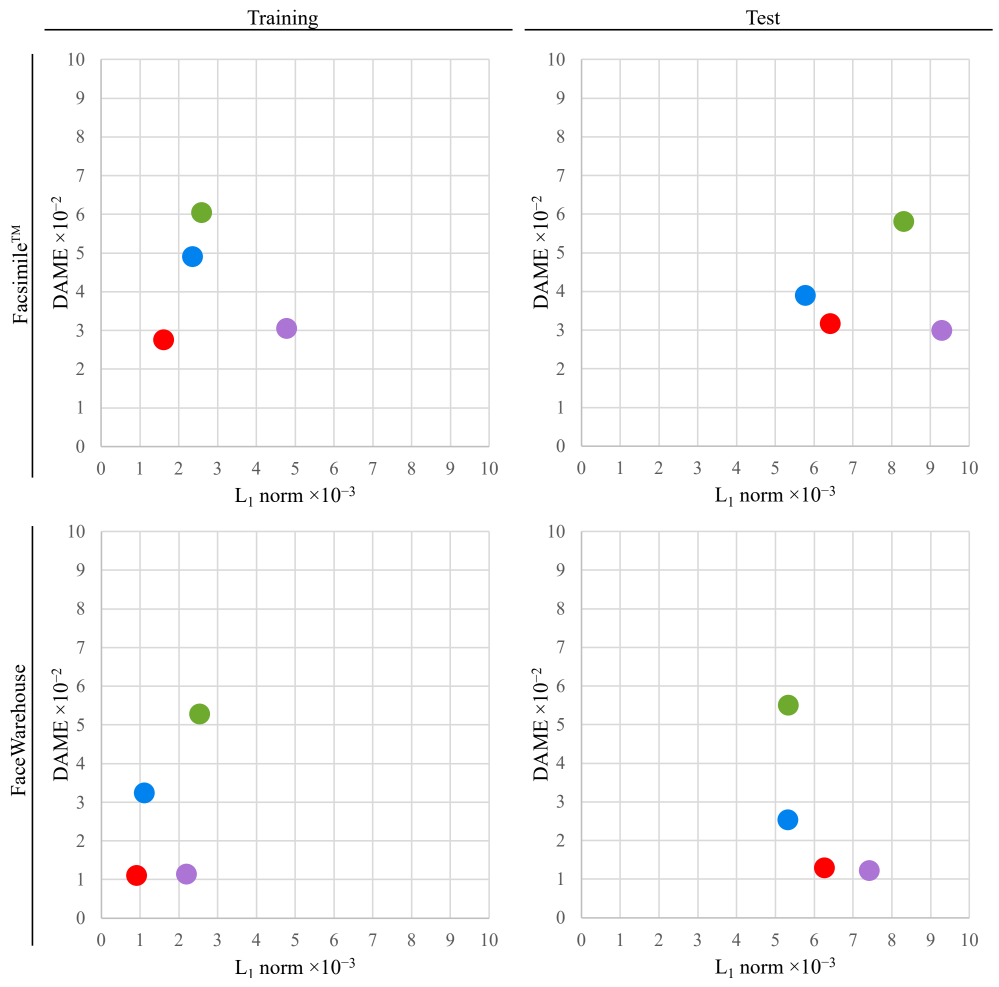


Figure 3. *Cont.*

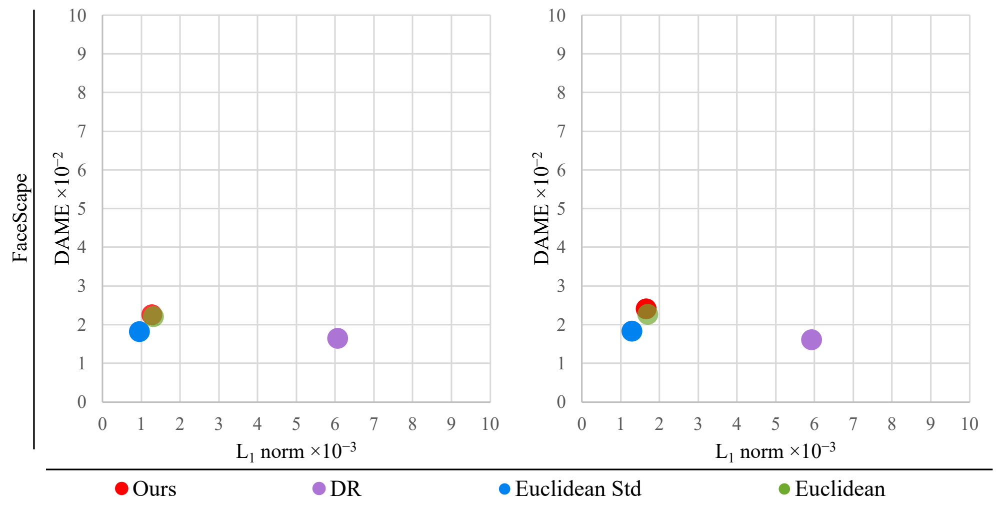


Figure 3. Comparison of the reconstruction results with our method ($k = 500$, $\gamma = 1$, $\mathbf{Z} = 64$) and with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates [27,29–31,49] and normalised deformation representation (DR) [12,37]. Across Facsimile and FaceWarehouse datasets, our method outperforms in reconstructing examples from the training set and favourably balances perceptual and geometric quality on the Pareto-front of optimal solutions. Our method underperforms on the FaceScape [43] dataset because the benefit of using normalised DR representation for high-frequency information is minuscule compared to standardised Euclidean representation.

Table 4. Ablation study on the reconstruction task demonstrating the impact of normalisation of the deformation representation (DR) and the standardisation of Euclidean coordinate input features.

	Training		**Validation**	
	L_1 Norm $\times 10^{-3} \downarrow$	**DAME $\times 10^{-2} \downarrow$**	**L_1 Norm $\times 10^{-3} \downarrow$**	**DAME $\times 10^{-2} \downarrow$**
Facsimile™				
DR without normalisation	**4.01**	**2.77**	12.05	3.07
DR with normalisation	4.77	3.05	**9.29**	**3.00**
Euclidean without standardisation	2.60	6.07	8.32	5.81
Euclidean with standardisation	**2.36**	**4.90**	**5.78**	**3.89**
FaceWarehouse				
DR without normalisation	2.57	1.71	10.03	**1.19**
DR with normalisation	**2.21**	**1.14**	**7.42**	1.22
Euclidean without standardisation	2.54	5.27	**5.33**	5.50
Euclidean with standardisation	**1.12**	**3.23**	**5.33**	**2.53**

7.1.2. Qualitative Evaluation

Figures 4 and 5 provide a visual assessment of the mesh quality in the reconstruction experiments, comparing our method with other commonly used representations. The meshes generated using Euclidean and standardised Euclidean representations display visible surface artefacts and struggle to capture high-frequency details. The severity of surface discrepancies corresponds with the colour visualisation of the DAME error. Results obtained with the normalised deformation representation (DR) are perceptually more similar to the ground truth meshes. Nonetheless, there is noticeable volume loss in the neck, chin and cheeks. In contrast, our method produces results that are perceptually similar to ground truth meshes without experiencing noticeable volume changes.

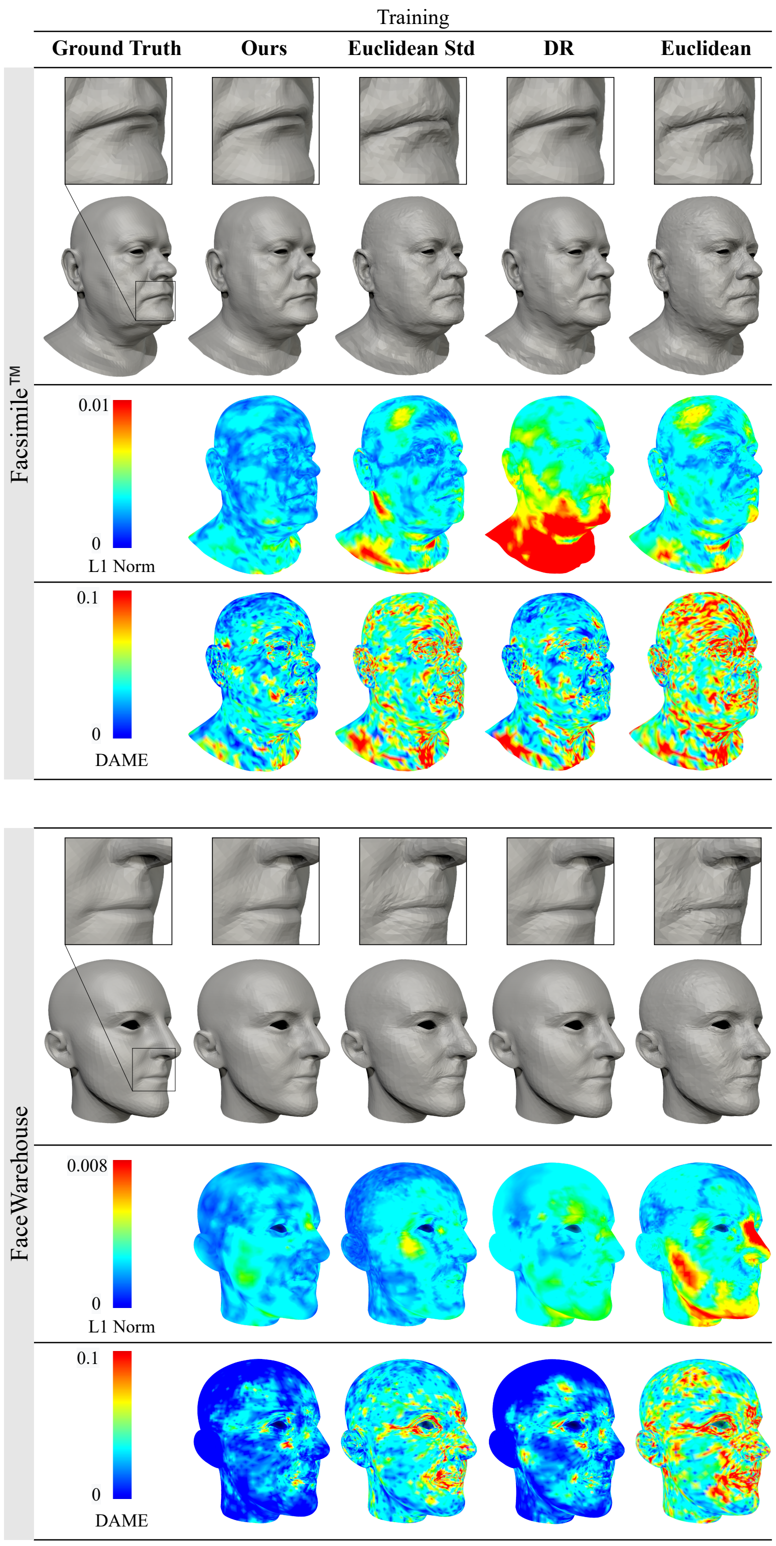


Figure 4. Qualitative comparison of the reconstruction results of training data with our method ($k = 500$, $\gamma = 1$) and with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates [27,29–31,49] and the normalised deformation representation (DR) [12,37]. The meshes generated by our method achieve superior results compared to other feature representations. Zooming into the digital version is recommended to see the surface artefacts on the results generated with Euclidean and standardised Euclidean representations.

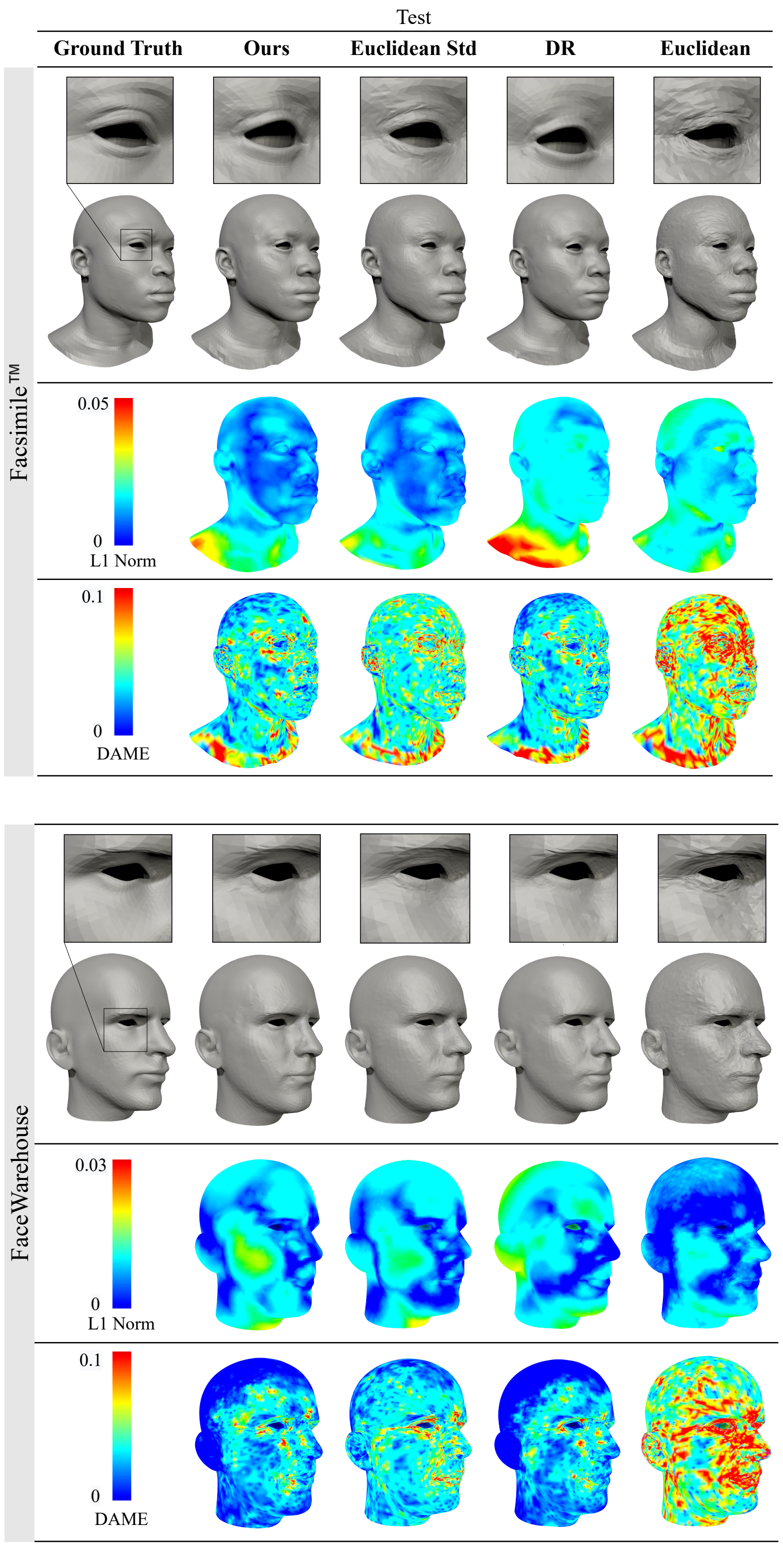


Figure 5. Qualitative comparison of the reconstruction results of test data with our method ($k = 500$, $\gamma = 1$) and with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates [27,29–31,49] and the normalised deformation representation (DR) [12,37]. The meshes generated by our method have similar surface quality to the outputs with DR while achieving much lower volume loss in the neck, chin, and cheek areas.

Figure 6 compares the rendered meshes from the Facsimile™ and FaceWarehouse datasets reconstructed with our proposed method and other popular methods: Mesh

Autoencoder [32], SpiralNet++ [49], Neural 3DMM [29] and FeaStNet [33]. A user study was conducted to qualitatively assess the perceptual quality of meshes generated by our method ($k = 500$, $\gamma = 1$, $\mathbf{Z} = 64$) compared with meshes synthesised using alternative methods and common feature representations. The evaluation focused on reconstructions of training and test subsets of the Facsimile™ and FaceWarehouse datasets. The study was conducted online on a representative sample of 94 participants (42 female, 51 male, 1 non-binary), with the following age distribution: 25 participants aged 18–25, 52 aged 26–35, 11 aged 36–45 and 6 aged 46–55. Participants viewed three images of rendered 3D models and were instructed to compare the ground truth reference model (always in the middle) against models A (left) and B (right). Subsequently, participants selected the model they perceived as more similar to the reference model (either option "A" or "B"). In cases where a clear distinction was challenging, participants had the option to choose "Difficult to say". Each participant completed a total of 48 comparisons. In each comparison, one 3D model was generated by our method, while the other was produced by an alternative method or using a different commonly used representation. Participants were instructed to use a desktop monitor or a tablet and complete the study in a full-screen view.

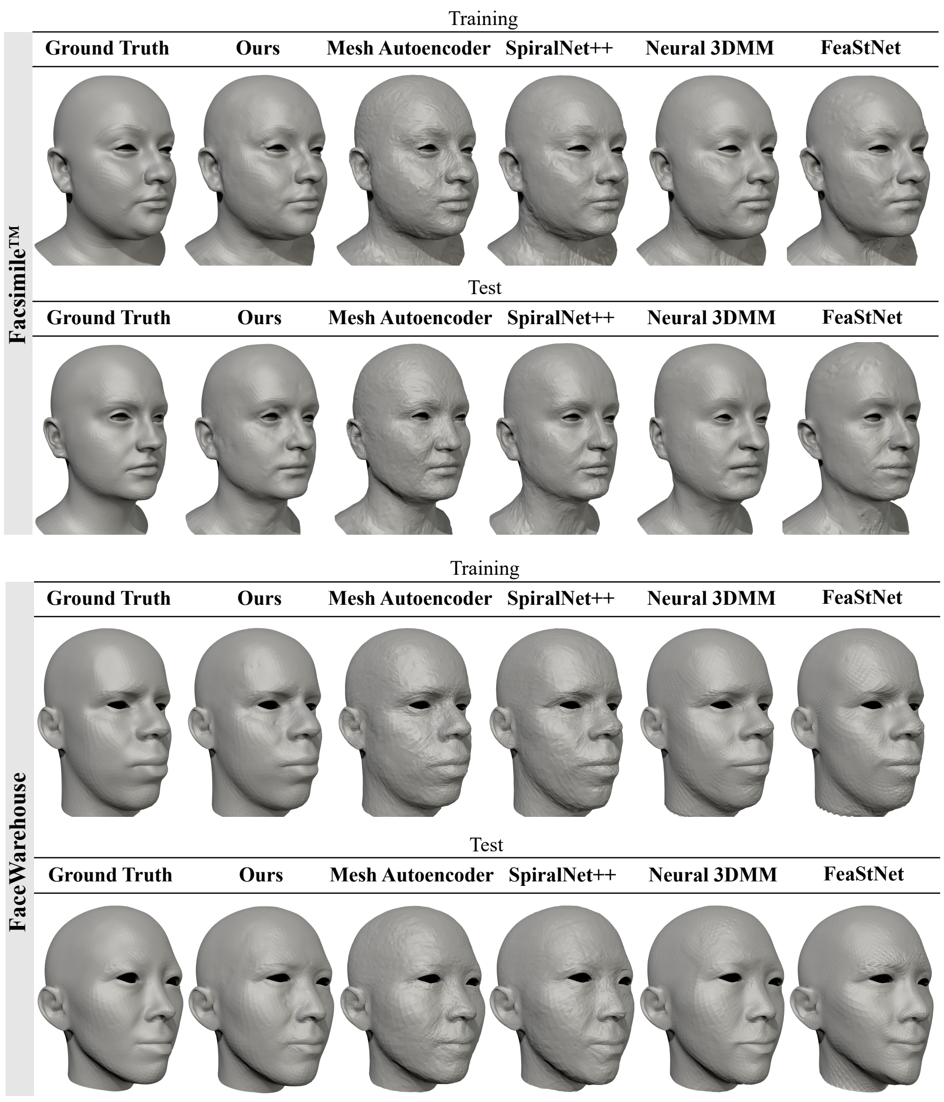


Figure 6. Visual comparison of the reconstruction results of the Facsimile™ and FaceWarehouse datasets using our method ($k = 500$, $\gamma = 1$, $\mathbf{Z} = 64$) and four other methods: Mesh Autoencoder [32], SpiralNet++ [49], Neural 3DMM [29] and FeaStNet [33]. It is recommended to zoom into the digital version to compare the reconstructed meshes.

Figure 7 displays the aggregated responses from the user study, which compared the visual similarity of the meshes generated by our proposed method with those produced by other methods [29,32,33,49]. In all cases, a strong majority of participants perceived the meshes generated by our method as more similar to the reference than those generated by other compared methods. Participants expressed a high certainty of their responses, with only a median of 4.3% choosing the "Difficult to say" option.

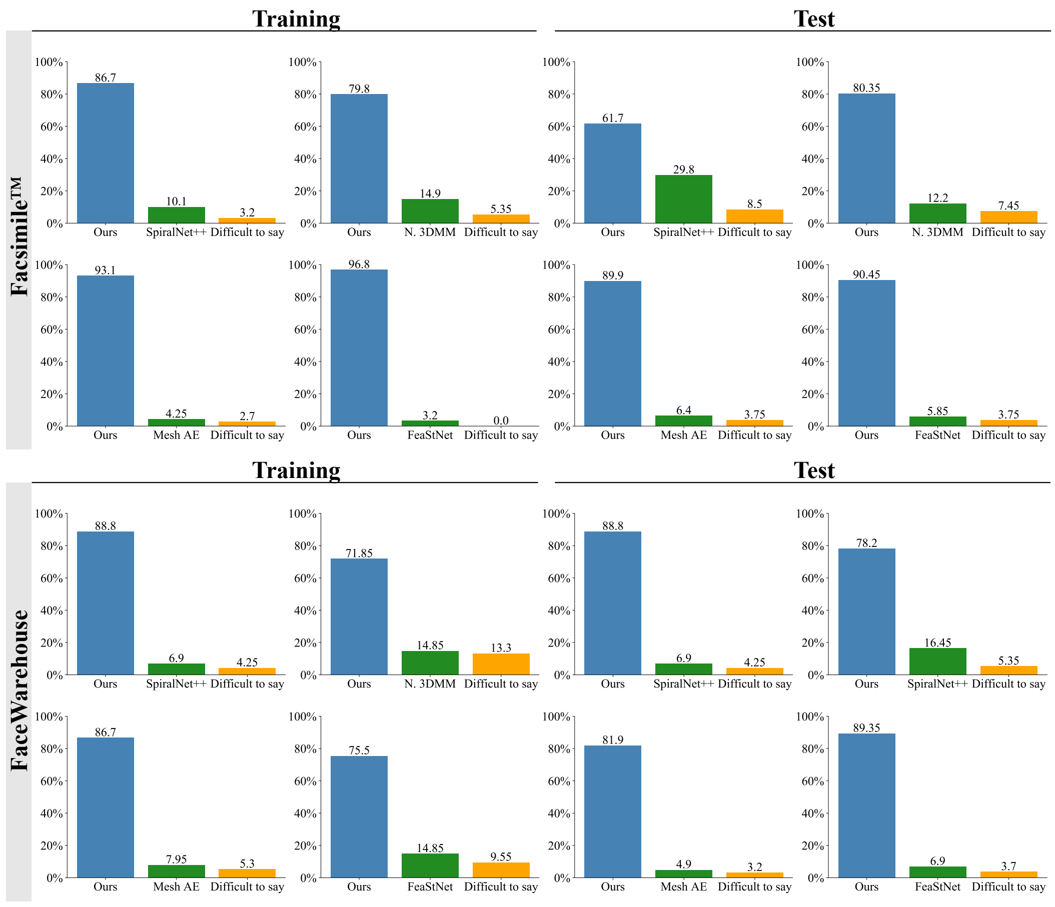


Figure 7. The outcomes of the user study, which compared the visual similarity to the ground truth of the meshes generated by our method and other methods: Mesh Autoencoder [32], SpiralNet++ [49], Neural 3DMM [29] and FeaStNet [33]. The bars show the percentage of participants who selected the mesh generated by the given method as more similar to the ground truth mesh. The participants were asked to select "Difficult to say" only when they had to guess between the generated models.

Figure 8 presents the results from the user study comparing our method with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates [27,29–31,49] and the normalised deformation representation (DR) [12,37]. Participants' responses align with our quantitative analysis of perceptual DAME error from Table 3. On average, 87.9% and 65.7% of participants perceived our methods' results as more similar than those from methods using Euclidean coordinates and standardised Euclidean coordinates, respectively. Regarding the comparison with normalised DR representation, the participants were divided. The training sets reconstructed with our method received 10.7 and 8.5 more percentage points of participants' preference. In comparison, the meshes produced with the normalised DR representation were chosen as more similar on the test sets by 4.3 and 13.3 more percentage points of participants. Consequently, the perceptual similarity of meshes generated by our method and the normalised DR representation is comparable, while our method yields significantly lower point-wise accuracy error, as demonstrated in quantitative analysis.

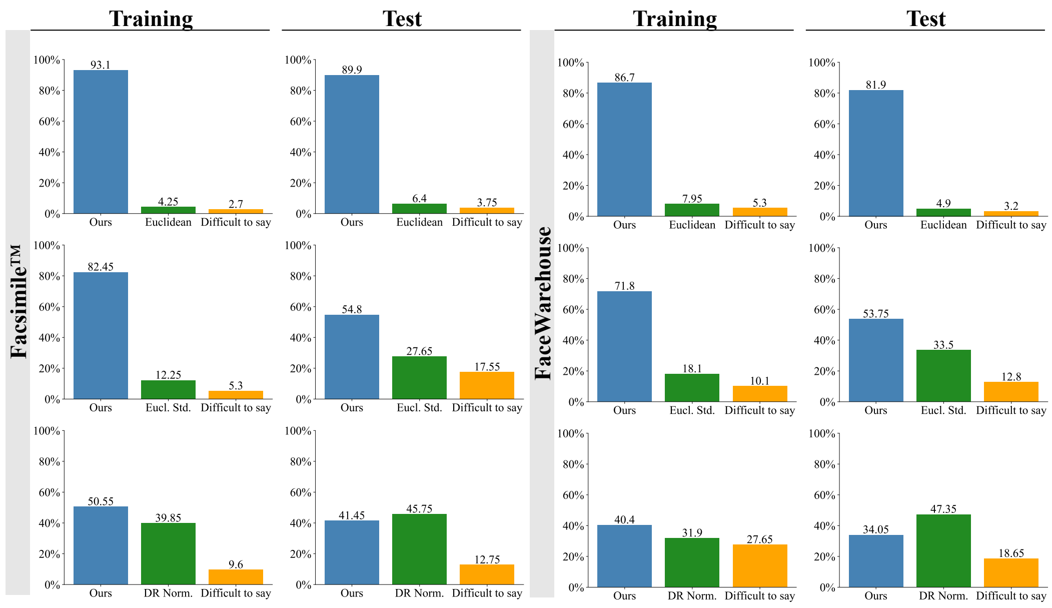


Figure 8. The results from the user study comparing our method with common representations used in other methods: Euclidean coordinates [32,34,50], standardised Euclidean coordinates (Eucl. Std.) [27,29–31,49] and the normalised deformation representation (DR Norm.) [12,37]. The bars show the percentage of participants who selected the mesh generated by the given method as more similar to the ground truth mesh. The participants were asked to select "Difficult to say" only when they had to guess between the generated models.

7.2. Mesh Interpolation

Mesh interpolation is widely applied in facial animation to produce new facial meshes from two known facial meshes. In contrast to existing interpolation methods that interpolate two facial meshes, our proposed approach interpolates low- and high-frequency parts of two facial meshes.

In Figure 9, the subject in the green outline is encoded into parameters $[\mathbf{Z}_{1l}|\mathbf{Z}_{1h}]$ and the subject in the purple outline is encoded into parameters $[\mathbf{Z}_{2l}|\mathbf{Z}_{2h}]$. High-frequency weight α and low-frequency weight β are used to interpolate between these latent parameters so that the interpolated latent parameters are

$$\mathbf{Z}_{\alpha,\beta} = [(1-\beta)\mathbf{Z}_{1l} + \beta\mathbf{Z}_{2l}|(1-\alpha)\mathbf{Z}_{1h} + \alpha\mathbf{Z}_{2h}]. \tag{10}$$

The meshes resulting from Equation (10) are shown in Figure 9A,B. In the figure, the meshes arranged in a grid are decoded from latent parameters $\mathbf{Z}_{\alpha,\beta}$. When using the linear interpolation in the vertex space between the mesh in the green outline and the mesh in the purple outline, the interpolation equation is

$$\mathbf{P}_{\delta} = \mathbf{P}_1 + \delta(\mathbf{P}_2 - \mathbf{P}_1), \tag{11}$$

where $\mathbf{P}_1$ and $\mathbf{P}_2$ are the vertex coordinates of the meshes in the green outline and the purple outline, respectively, and $0 \le \delta \le 1$.

The meshes obtained from Equation (11) are shown in Figure 9C. Interpolating low- and high-frequency latent parameters with two different conditioning values, 0.4 and 1.0, generates 28 new meshes. In contrast, interpolating the meshes in the green and purple outlines creates only two new meshes.

Our discussion indicates that multi-frequency interpolation noticeably raises the capacity to create novel facial meshes from two known facial meshes. In addition, Figure 9 demonstrates the disentanglement of low and high frequencies in the parametric space and illustrates the impact of the Conditioning Factor γ.

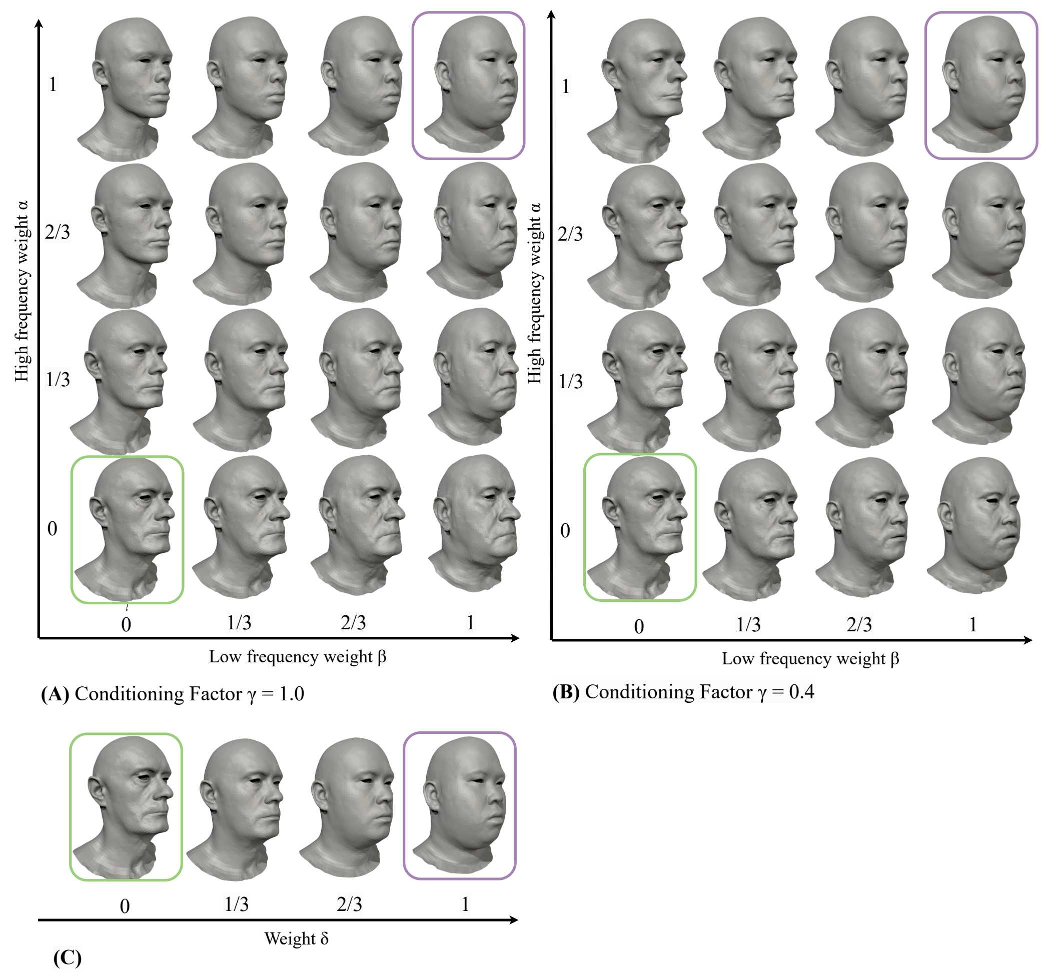
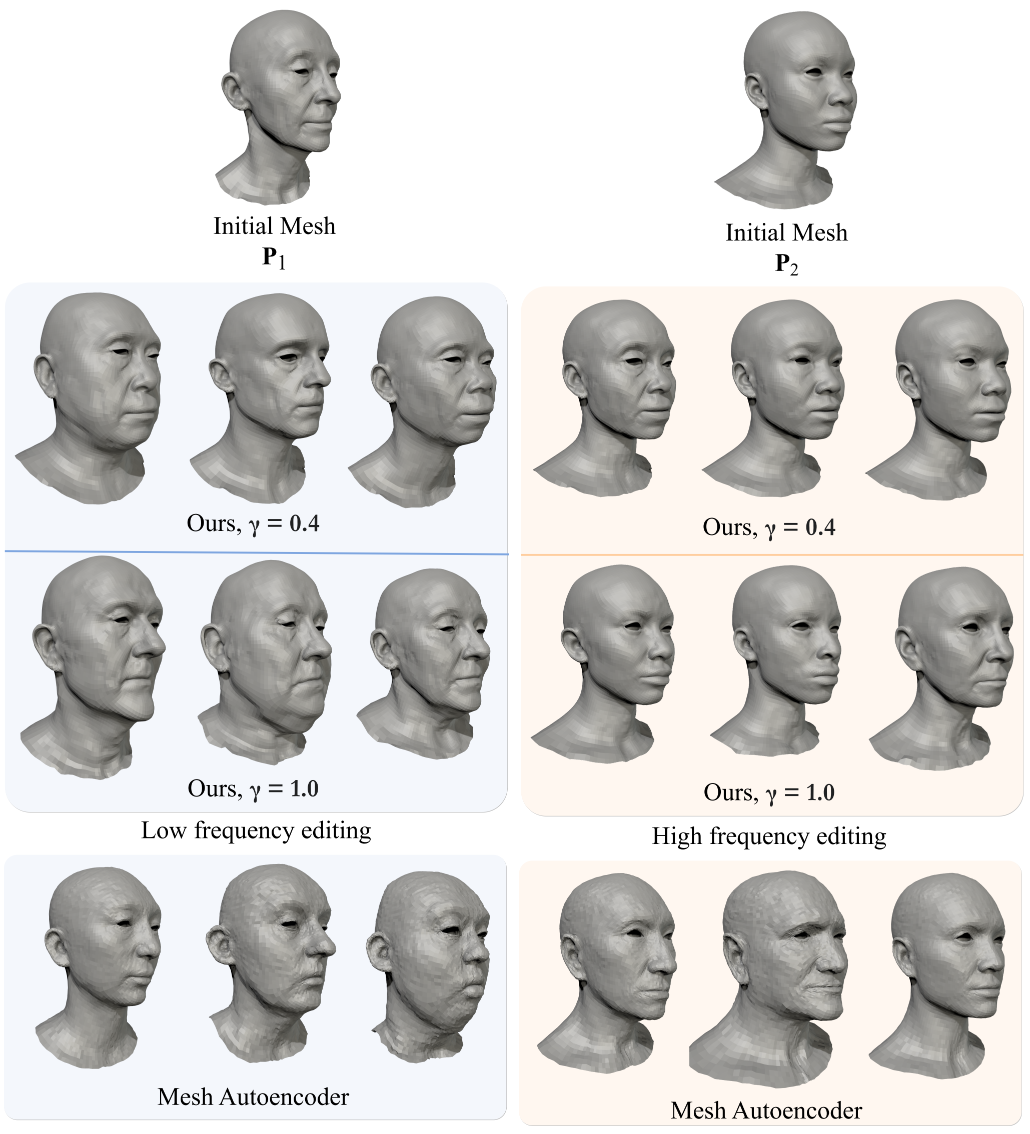


Figure 9. Interpolation of low-frequency and high-frequency latent parameters, $k = 500$. Two facial meshes (in green and purple outlines) from the Facsimile™ [42] dataset are encoded. In (**A**), the model is trained with the Conditioning Factor $\gamma = 1.0$. In (**B**), the Conditioning Factor $\gamma = 0.4$. The meshes arranged in a grid are decoded from interpolated latent parameters. In (**C**), the meshes in green and purple outlines are interpolated in the vertex space.

When $\gamma = 1.0$, the high-frequency parameters of an elderly subject influence mid-frequencies to impose the generation of plausible faces. However, due to bias towards younger subjects in the dataset, this conditioning significantly restricts the domain of generated faces. For an artist, such editing behaviour may be undesirable. In contrast, interpolation of low- and high-frequency parameters with $\gamma = 0.4$ is more predictable, as high- and low-frequency deformations are almost fully disentangled. Nevertheless, plausible faces exist within a joint distribution of low- and high-frequency deformations. Consequently, the network may generate implausible examples, like the older man with young, smooth skin depicted in Figure 9B at ($\alpha = 1, \beta = 0$). Therefore, the choice of γ depends on application-specific requirements, whether prioritising more precise and flexible artistic control or the ability to generate only plausible faces.

7.3. Multi-Frequency Editing

Figure 10 presents the application of our method in editing low- and high-frequency deformations independently. Additionally, the editing capabilities of our model are compared with the method in [32], which does not disentangle low- and high-frequency parameters. In our approach, low-frequency parameters affect the overall head shape while preserving high-frequency details. The Conditioning Factor γ impacts the nature of editing. With $\gamma = 1.0$ (full conditioning), high frequencies condition low-frequency deformations to a more narrow domain of only plausible faces. In contrast, when $\gamma = 0.4$ (partial conditioning), edited heads are more diverse, despite some falling outside of the domain of real faces. When editing high-frequency parameters, the overall head shape remains unchanged, with only fine details being affected. Notably, with $\gamma = 0.4$, the editing of high frequencies becomes more precise and predictable since high- and low-frequency parameters barely influence each other.



(A) Comparison between low-frequency editing of the method and the latent code editing of Mesh Autoencoder.

(B) Comparison between high-frequency editing of the method and the latent code editing of Mesh Autoencoder.

Figure 10. Comparison of latent code editing between the proposed method and Mesh Autoencoder [32]. In (**A**), the editing of low-frequency latent codes of encoded mesh $\mathbf{P}_1$. In (**B**), the editing of high-frequency latent codes of encoded mesh $\mathbf{P}_2$. Top and middle row: the examples decoded using our model with $k = 500$ and Conditioning Factor $\gamma = 0.4$ and $\gamma = 1.0$. Bottom row: the results of editing a subset of latent parameters using the method in [32]. The parameters of our method successfully disentangle high and low frequencies. While subjective, it can be observed that lower γ provides more control and produces more diverse results. Meanwhile, altering the parameters of Mesh Autoencoder [32] affects the entire frequency spectrum.

8. Summary and Future Work

In this paper, a new approach called Deep Spectral Meshes was proposed to address two crucial challenges: the absence of dedicated representations for describing deformations at different frequencies and the inability to independently edit deformations at different frequency levels. These problems may lead to semantically meaningless parameters used in deep learning and result in the unsatisfactory geometric and perceptual quality of assets failing to meet standards in industrial applications.

To develop our new approach, spectral meshes were introduced, overcoming these challenges. They were employed to decompose mesh deformations into low- and high-frequency parts. Subsequently, these parts were converted into features represented with standardised Euclidean coordinates and the normalised deformation representation, respectively. Graph neural networks were proposed to reconstruct features, which were later converted back to Euclidean coordinates to obtain the reconstructed 3D models. A Conditioning Factor was introduced to control the level of mutual conditioning of deformations at different frequencies.

Extensive experiments were conducted to validate and compare our proposed approach with previously published methods, both quantitatively and qualitatively. Results demonstrate the capability of our approach to independently edit low- and high-frequency deformations of facial meshes. Moreover, the experiments illustrate the impact of the Conditioning Factor on balancing mutually exclusive objectives of independent control of deformations at different frequencies and generating plausible synthetic examples. Comparisons with existing methods demonstrate the superiority of our approach over Euclidean coordinates, standardised Euclidean coordinates, the normalised deformation representation (DR) and other methods, as assessed by both L_1 and perceptual metric evaluations.

The significance of our proposed approach is underscored through its applications presented in this paper. As described in Section 7, our method has been applied in mesh compression, enhancing the quality of the reconstructed meshes. Additionally, it has been employed in mesh interpolation, expanding the capability to generate a larger variety of new shapes compared to direct interpolation between two facial meshes. This was achieved by independently interpolating both low-frequency and high-frequency parameters. Moreover, our proposed method has found application in multi-frequency editing, satisfying different editing requirements. It allows the alteration of the overall shape while preserving details by adjusting only high-frequency parameters. It also accommodates modifying fine details while maintaining the overall shape by editing only low-frequency parameters.

Applications of Deep Spectral Meshes are potentially far-reaching, but we identified several promising applications and areas of research which warrant further investigation:

- This work restricts its application to low- and high-frequency bands. The partition of mesh data into more than two frequency bands and an investigation into the relationship between the learning model and the frequency bands remain unexplored. The limitation to two frequency bands in our approach is linked to the properties of the chosen mesh representations. Standardised Euclidean coordinates represent low-frequency information to ensure high point-wise accuracy of the generated meshes. The normalised deformation representation (DR) encodes high-frequency information for superior perceptual quality of the results. Future work on partitioning mesh data into more than two frequency bands could further explore the benefits of alternative feature representations at different frequency levels.
- Further research could describe the relationship between the choice of parameter k and the quality of generated meshes. Designing an algorithm to determine the optimal split between frequency bands is a potential avenue for future research. To achieve this, an objective function relating the quality of generated meshes to parameter k could be formulated. The selection of a suitable optimisation method would be crucial to efficiently obtain the optimal split.

- Our proposed method can be extended to address the problem of multi-frequency-based deformation transfer, which has not been investigated in existing research studies. The basic idea of multi-frequency-based deformation transfer involves decomposing source and target meshes into mean, low- and high-frequency parts. The differences between the source model and these parts at two different poses are determined and transferred to the corresponding bands of the target mesh. Subsequently, the graph neural network proposed in this paper can be employed to reconstruct a new shape for the target mesh with the pose of the source mesh.
- While our approach has been applied to deformable facial meshes, future work could extend the proposed method to articulated shapes like hands and bodies. Existing research has proposed various methods to relate articulated shapes to their underlying skeleton. With this extension, articulated shapes can be decomposed into mean, low- and high-frequency parts. Then, the relationships between these parts and the movements of the skeleton of the articulated shapes can be investigated. These relationships can be used to synthesise mean, low- and high-frequency parts of new poses. The graph neural network proposed in this paper can then extract features, reconstruct them, and synthesise new shapes from the reconstructed features.
- It may be possible to apply the approach proposed in this paper to 3D tumour image analysis. First, 3D shapes containing tumours can be reconstructed from medical images. Then, the approach could be employed to disentangle normal deformations and abnormal deformations caused by tumours and extract tumour features.

Author Contributions: Conceptualisation, methodology, software, validation, visualisation, writing—original draft preparation, R.K.; writing—review and editing, R.K., R.S., L.Y., S.B., W.K. and G.M.; supervision, R.S., L.Y., W.K. and G.M.; resources and data curation, W.K. All authors have read and agreed to the published version of the manuscript.

Funding: This work was supported by grant funding of the Centre for Digital Entertainment at Bournemouth University by the UK Engineering and Physical Sciences Research Council (EPSRC) EP/L016540/1 and Humain Ltd.

Data Availability Statement: The FaceWarehouse [44] dataset is available from Kun Zhou, Zhejiang University. Data are available at http://kunzhou.net/zjugaps/facewarehouse/ accessed on 2 October 2020 with the permission of Kun Zhou. The FaceScape [43] dataset is available from the CITE Lab, Nanjing University. Data are available at https://facescape.nju.edu.cn/ accessed on 3 July 2020 with the permission of the CITE Lab. Both of the datasets are available to approved researchers due to the sensitive nature of human face data. The datasets were used under licenses for this study. For commercial reasons, the Facsimile™ [42] dataset is not publicly available.

Conflicts of Interest: R.K., S.B., W.K. and G.M. are employees of Humain Ltd. R.S. is an employee of The Foundry Visionmongers Ltd. The authors declare no conflicts of interest.

References

1. Russo, M. *Polygonal Modeling: Basic and Advanced Techniques*; Jones & Bartlett Learning: Burlington, MA, USA, 2010.
2. Feng, X.; Shi, M. Surface representation and processing. In Proceedings of the 2009 8th IEEE International Conference on Cognitive Informatics, Hong Kong, China, 15–17 June 2009; IEEE: Piscatway, NJ, USA, 2009; pp. 542–545. [CrossRef]
3. Sorkine, O.; Cohen-Or, D.; Lipman, Y.; Alexa, M.; Rössl, C.; Seidel, H.P. Laplacian surface editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Nice, France, 8–10 July 2004; pp. 175–184.
4. Zhang, H.; Kaick, O.V.; Dyer, R. Spectral mesh processing. *Comput. Graph. Forum* **2010**, *29*, 1865–1894. [CrossRef]
5. Sorkine, O. Laplacian Mesh Processing. In *Eurographics (STARs)*; The Eurographics Association: Eindhoven The Netherlands, 2005.
6. Bronstein, M.M.; Bruna, J.; Lecun, Y.; Szlam, A.; Vandergheynst, P. Geometric Deep Learning: Going beyond Euclidean data. *IEEE Signal Process. Mag.* **2017**, *34*, 18–42. [CrossRef]
7. Egger, B.; Smith, W.; Tewari, A.; Wuhrer, S.; Zollhoefer, M.; Beeler, T.; Bernard, F.; Bolkart, T.; Kortylewski, A.; Romdhani, S.; et al. 3D Morphable Face Models—Past, Present, and Future. *ACM Trans. Graph.* **2020**, *38*, 157. [CrossRef]
8. Xiao, Y.P.; Lai, Y.K.; Zhang, F.L.; Li, C.; Gao, L. A survey on deep geometry learning: From a representation perspective. *Comput. Vis. Media* **2020**, *6*, 113–133. [CrossRef]
9. Gao, L.; Lai, Y.K.; Liang, D.; Chen, S.Y.; Xia, S. Efficient and flexible deformation representation for data-driven surface modeling. *ACM Trans. Graph.* **2016**, *35*, 158. [CrossRef]

10. Gao, L.; Lai, Y.K.; Yang, J.; Ling-Xiao, Z.; Xia, S.; Kobbelt, L. Sparse Data Driven Mesh Deformation. *IEEE Trans. Vis. Comput. Graph.* **2021**, *27*, 2085–2100. [CrossRef] [PubMed]
11. Tan, Q.; Zhang, L.; Yang, J.; Lai, Y.; Gao, L. Variational Autoencoders for Localized Mesh Deformation Component Analysis. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**, *44*, 6297–6310. [CrossRef]
12. Wu, Q.; Zhang, J.; Lai, Y.K.; Zheng, J.; Cai, J. Alive Caricature from 2D to 3D. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; IEEE: Piscataway, NJ, USA, 2018; pp. 7336–7345. [CrossRef]
13. Melzi, S.; Rodol, E.; Castellani, U.; Bronstein, M. Localized Manifold Harmonics for Spectral Shape Analysis. *Comput. Graph. Forum* **2018**, *37*, 20–34. [CrossRef]
14. Xu, C.; Lin, H.; Hu, H.; He, Y. Fast calculation of Laplace-Beltrami eigenproblems via subdivision linear subspace. *Comput. Graph.* **2021**, *97*, 236–247. [CrossRef]
15. Lescoat, T.; Liu, H.; Thiery, J.; Jacobson, A.; Boubekeur, T.; Ovsjanikov, M. Spectral Mesh Simplification. *Comput. Graph. Forum* **2020**, *39*, 315–324. [CrossRef]
16. Wang, H.; Lu, T.; Au, O.; Tai, C. Spectral 3D mesh segmentation with a novel single segmentation field. *Graph. Model.* **2014**, *76*, 440–456. [CrossRef]
17. Tong, W.; Yang, X.; Pan, M.; Chen, F. Spectral mesh segmentation via ℓ_0 gradient minimization. *IEEE Trans. Vis. Comput. Graph.* **2020**, *26*, 440–456. [CrossRef]
18. Bao, X.; Tong, W.; Chen, F. A Spectral Segmentation Method for Large Meshes. *Commun. Math. Stat.* **2023**, *11*, 583–607. [CrossRef]
19. Jain, V.; Zhang, H. Robust 3D Shape Correspondence in the Spectral Domain. In Proceedings of the IEEE International Conference on Shape Modeling and Applications 2006 (SMI'06), Matsushima, Japan, 14–16 June 2006; pp. 1–12. [CrossRef]
20. Dubrovina, A.; Kimmel, R. Matching shapes by eigendecomposition of the Laplace-Beltrami operator. In Proceedings of the 5th International Symposium 3D Data Processing, Visualization and Transmission, Paris, France, 17–20 May 2010; pp. 1–8.
21. Melzi, S.; Ren, J.; Rodolà, E.; Sharma, A.; Wonka, P.; Ovsjanikov, M. ZoomOut: Spectral upsampling for efficient shape correspondence. *ACM Trans. Graph.* **2019**, *38*, 155. [CrossRef]
22. Wu, Z.; Pan, S.; Chen, F.; Long, G.; Zhang, C.; Yu, P.S. A Comprehensive Survey on Graph Neural Networks. *IEEE Trans. Neural Netw. Learn. Syst.* **2021**, *32*, 4–24. [CrossRef]
23. Dong, Q.; Wang, Z.; Li, M.; Gao, J.; Chen, S.; Shu, Z.; Xin, S.; Tu, C.; Wang, W. Laplacian2Mesh: Laplacian-Based Mesh Understanding. *IEEE Trans. Vis. Comput. Graph.* **2023**, 1–13 . [CrossRef]
24. Lemeunier, C.; Denis, F.; Lavoué, L.; Dupont, F. SpecTrHuMS: Spectral transformer for human mesh sequence learning. *Comput. Graph.* **2023**, *115*, 191–203. [CrossRef]
25. Qiao, Y.; Gao, L.; Yang, J.; Rosin, P.; Lai, Y.; Chen, X. Learning on 3D Meshes With Laplacian Encoding and Pooling. *IEEE Trans. Vis. Comput. Graph.* **2022**, *28*, 1317–1327. [CrossRef]
26. Nasikun, A.; Hildebrandt, K. The Hierarchical Subspace Iteration Method for Laplace–Beltrami Eigenproblems. *ACM Trans. Graph.* **2022**, *41*, 17. [CrossRef]
27. Ranjan, A.; Bolkart, T.; Sanyal, S.; Black, M.J. Generating 3D Faces Using Convolutional Mesh Autoencoders. In Proceedings of the European Conference on Computer Vision (ECCV), Munich, Germany, 8–14 September 2018; pp. 725–741.
28. Defferrard, M.; Bresson, X.; Vandergheynst, P. Convolutional neural networks on graphs with fast localized spectral filtering. In Proceedings of the Advances in Neural Information Processing Systems, Barcelona, Spain, 5–10 December 2016; pp. 3844–3852.
29. Bouritsas, G.; Bokhnyak, S.; Ploumpis, S.; Zafeiriou, S.; Bronstein, M. Neural 3D morphable models: Spiral convolutional networks for 3D shape representation learning and generation. In Proceedings of the IEEE International Conference on Computer Vision, Seoul, Republic of Korea, 27 October–2 November 2019; pp. 7212–7221. [CrossRef]
30. Chen, Z.; Kim, T.K. Learning feature aggregation for deep 3D morphable models. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Nashville, TN, USA, 20–25 June 2021; pp. 13159–13168. [CrossRef]
31. Gao, Z.; Yan, J.; Zhai, G.; Zhang, J.; Yang, Y.; Yang, X. Learning Local Neighboring Structure for Robust 3D Shape Representation. In Proceedings of The Thirty-Fifth AAAI Conference on Artificial Intelligence (AAAI-21) Learning, Virtual, 2–9 February 2021; Volume 35, pp. 1397–1405.
32. Zhou, Y.; Wu, C.; Li, Z.; Cao, C.; Ye, Y.; Saragih, J.; Li, H.; Sheikh, Y. Fully Convolutional Mesh Autoencoder using Efficient Spatially Varying Kernels. *Adv. Neural Inf. Process. Syst.* **2020**, *33*, 9251–9262.
33. Verma, N.; Boyer, E.; Verbeek, J. FeaStNet: Feature-Steered Graph Convolutions for 3D Shape Analysis. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 2598–2606. [CrossRef]
34. Cheng, S.; Bronstein, M.; Zhou, Y.; Kotsia, I.; Pantic, M.; Zafeiriou, S. MeshGAN: Non-linear 3D Morphable Models of Faces. *arXiv* **2019**. https://doi.org/10.48550/arXiv.1903.10384. [CrossRef]
35. Zhou, Y.; Deng, J.; Kotsia, I.; Zafeiriou, S. Dense 3D Face Decoding over 2500FPS: Joint Texture & Shape Convolutional Mesh Decoders. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; pp. 1097–1106.
36. Yuan, Y.J.; Lai, Y.K.; Yang, J.; Fu, H.; Gao, L. Mesh Variational Autoencoders with Edge Contraction Pooling. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 274–275.

37. Jiang, Z.H.; Wu, Q.; Chen, K.; Zhang, J. Disentangled representation learning for 3D face shape. In Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Long Beach, CA, USA, 15–20 June 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 11949–11958. [CrossRef]
38. Zheng, X.; Jiang, B.; Zhang, J. Deformation representation based convolutional mesh autoencoder for 3D hand generation. *Neurocomputing* **2021**, *444*, 356–365. [CrossRef]
39. Baran, I.; Vlasic, D.; Grinspun, E.; Popović, J.P. Semantic Deformation Transfer. In *ACM SIGGRAPH 2009 Papers, Proceedings of the Special Interest Group on Computer Graphics and Interactive Techniques Conference, New Orleans, LA, USA, 3–7 August 2009;* ACM: New York, NY, USA, 2009; pp. 1–6. [CrossRef]
40. Sorkine, O.; Alexa, M.; Berlin, T.U. As-Rigid-As-Possible Surface Modeling. In *Proceedings of the Symposium on Geometry Processing, Barcelona, Spain, 4–6 July 2007*; Belyaev, A., Ed.; The Eurographics Association: Eindhoven The Netherlands, 2007; Volume 4, pp. 109–116.
41. Hinton, G.E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; Salakhutdinov, R.R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv* **2012**. [CrossRef]
42. Humain Limited. Humain Limited—Research & Development. 2022. Available online: https://www.humain-studios.com/ (accessed on 12 May 2022)
43. Yang, H.; Zhu, H.; Wang, Y.; Huang, M.; Shen, Q.; Yang, R.; Cao, X. FaceScape: A Large-scale High Quality 3D Face Dataset and Detailed Riggable 3D Face Prediction. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, Seatle, WA, USA, 13–19 June 2020; pp. 601–610.
44. Cao, C.; Weng, Y.; Zhou, S.; Tong, Y.; Zhou, K. FaceWarehouse: A 3D facial expression database for visual computing. *IEEE Trans. Vis. Comput. Graph.* **2014**, *20*, 413–425. [CrossRef] [PubMed]
45. Lehoucq, R.B.; Sorensen, D.C.; Yang, C. *ARPACK Users' Guide: Solution of Large-Scale Eigenvalue Problems with Implicitly Restarted Arnoldi Methods*; Society for Industrial and Applied Mathematics: Philadelphia, PA, USA, 1998.
46. Paszke, A.; Gross, S.; Massa, F.; Lerer, A.; Bradbury, J.; Chanan, G.; Killeen, T.; Lin, Z.; Gimelshein, N.; Antiga, L. Pytorch: An imperative style, high-performance deep learning library. *Adv. Neural Inf. Process. Syst.* **2014**, *32*, 8026–8037.
47. Váša, L.; Rus, J. Dihedral Angle Mesh Error: A fast perception correlated distortion measure for fixed connectivity triangle meshes. *Eurographics Symp. Geom. Process.* **2012**, *31*, 1715–1724. [CrossRef]
48. Corsini, M.; Larabi, M.C.; Lavoué, G.; Petřík, O.; Váša, L.; Wang, K. Perceptual metrics for static and dynamic triangle meshes. *Comput. Graph. Forum* **2013**, *32*, 101–125. [CrossRef]
49. Gong, S.; Chen, L.; Bronstein, M.; Zafeiriou, S. SpiralNet++: A fast and highly efficient mesh convolution operator. In Proceedings of the IEEE/CVF International Conference on Computer Vision Workshop, ICCVW 2019, Seoul, Republic of Korea, 27–28 October 2019; pp. 4141–4148. [CrossRef]
50. Hanocka, R.; Fleishman, S.; Hertz, A.; Fish, N.; Giryes, R.; Cohen, D. MeshCNN: A Network with an Edge. *ACM Trans. Graph. (TOG)* **2019**, *38*, 1–12. [CrossRef]