Model-based Self-supervision for Dense Face Alignment and 3D Reconstruction

Tatsuro Koizumi

PhD

University of York

Computer Science

October 2023

Abstract

In the field of monocular 3D reconstruction, self-supervision based on differentiable rendering and a statistical 3D model has been proposed to alleviate the need for datasets with ground truth. In theory, this enables training of neural networks only using unannotated images. However, training through self-supervision tends to be unstable and surrogate supervision such as landmarks is required in practice. Moreover, reaching convergence in self-supervised 3D reconstruction is slow or unachievable due to the weak and discontinuous supervisory signal provided by a differentiable renderer. Our research starts from the aim to improve such problems in differentiable renderer-based self-supervision.

Firstly, we combined differentiable linear least-squares fitting of a 3D morphable model (3DMM), pose, and lighting with self-supervision. We propose linear least-squares solutions for geometric and photometric parameters including a novel inverse spherical harmonic lighting model. This assures optimal fitting of photometric components given estimated geometric parameters and improves fidelity in reconstructed appearance. This concept also provides an opportunity to combine 3DMM fitting with image-to-image networks, leading to stable training without requiring landmark supervision.

Secondly, we proposed supervision based on semantic segmentation. In contrast to landmarks, this form of supervision is dense and always well defined. However, it is not one-to-one, meaning more complex loss functions are required to exploit it. We propose two novel cohesive measures for semantic segmentation supervision. First, we show how precomputed distance maps in a 3DMM UV space can be used to supervise pixel-wise estimates of image-model correspondence. Second, we derive a novel differentiable vertex to pixel cohesive measure based on the geometric Rényi divergence. Using this loss, we show that pure shape-from-semantic segmentation is possible via analysis-by-synthesis.

Lastly, we combined both techniques and propose the self-supervised architecture for 3D face reconstruction that does not require a differentiable renderer.

Acknowledgments

I would like to express my profound gratitude to my supervisor, Dr. William Smith, for his continuous support, patience, and invaluable guidance throughout the course of this research. Your insights and wisdom have been instrumental in shaping this work, and I am deeply thankful for the opportunity to learn under your supervision.

I wish to express my deepest love and gratitude to my wife, Ami, who has been my constant support and pillar of strength during this journey. Your unwavering belief in me, your sacrifices, and your endless love have sustained me through challenges, and I dedicate this accomplishment to you.

Declaration

I declare that this thesis is a presentation of original work and I am the sole author. This work has not previously been presented for an award at this, or any other, University. All sources are acknowledged as References. Main contents have been published in the following papers. For all these works I made the major contribution in design, implementation and experiments.

- Tatsuro Koizumi and William AP Smith. Shape from semantic segmentation via the geometric Rényi divergence. In Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, pages 2312–2321, 2021.
- Tatsuro Koizumi and William AP Smith. "Look ma, no landmarks!"–unsupervised, model-based dense face alignment. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II, pages 690–706. Springer, 2020.

Tatsuro Koizumi

September 30, 2023

Contents

List of Tables

List of Figures

Chapter 1

Introduction

1.1 Background

Reconstruction of the shape and appearance of 3D objects from RGB images has been one of the ultimate goals of research in computer vision. This field of research is particularly important for potential in applications such as 3D content creation, augmented reality, telecommunications, medical diagnosis, land surveying, and robotics [18][67]. In 3D reconstruction techniques, the physical process of forming an RGB image of a subject is utilised to derive a formulation to reconstruct the original geometric and photometric information. For instance, multi-view stereo methods reconstruct the 3D geometry of a subject based on constraints from multiple observations, modeled by perspective projections [27]. Perspective projection models the transport of light rays from the surface of the subject to the image plane through the principal point of the lens, in a simplified form. In general, the forward process of image formation inherently loses a certain amount of information about a subject, making the inverse process highly ill-posed. To address this ill-posedness, classical algorithms often rely on hand-crafted priors or models. In the case of multi-view stereo, the local smoothness of a natural object's surface is commonly utilised. This smoothness constraint is typically embedded in the form of patch-based matching [20] or incorporated as post-processing through smoothness regularisation [29]. However, even with sophisticated regularisation, this approach requires a large number of views and is prone

to producing erroneous results. Another approach is 3D reconstruction based on a linear statistical model, which serves as a strong prior for specific object categories. A well-known example of such a model is the 3D Morphable Model (3DMM) [5]. In this approach, both the 3D shape and texture of a human face are represented by a linear statistical model. This model comprises linear basis vectors, normalised by the variance, and a mean vector for the 3D position and albedo at each vertex of the 3D template meshes. In this method, coefficients for the bases are estimated by minimising the error between pixel values in a target image and those in a rendered image, an approach known as analysis-by-synthesis. The effective representational capability of the linear model enables high-fidelity 3D reconstruction from a single image. However, this approach usually necessitates solving a challenging optimisation process, which is both non-convex and non-linear.

In recent years, advancements in deep learning have enabled monocular 3D reconstruction without the need for inference-time optimisation. Neural networks can learn the mapping from an input image to latent 3D information by training on a large dataset of images with ground-truth data. In early works, 3D reconstruction is formulated as a regression task, wherein the network infers 3D representations such as depth maps, 3D meshes, or 3DMM coefficients from input images. The network is trained in a supervised manner, requiring costly annotated data that consists of pairs of an image and its corresponding 3D representation. To relax the requirement for the dataset, a hybrid approach that combines neural networks with analysis-by-synthesis, namely model-based self-supervision, is proposed [66]. In this approach, a fixed rendering layer, implemented in a differentiable manner and known as a differentiable renderer, is integrated into the training pipeline of the neural network. In this pipeline, the 3D representation estimated by the neural network is rendered by the differentiable renderer. The parameters of the neural network are optimised so that the error between the rendered image and the original input image is minimised. As the rendering layer is differentiable, standard gradient descent techniques can be applied during training. This hybrid strategy has enabled network training from images without ground truth, thereby broadening the scope of its application to various domains where ground truth is unattainable. Figure 1.1 shows the summarisation of

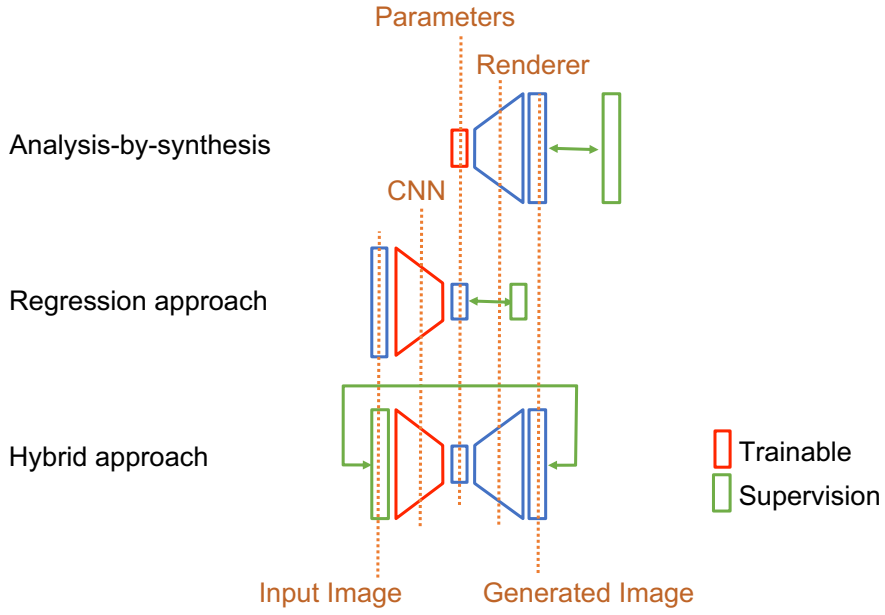


Figure 1.1: Overview of monocular 3D reconstruction approaches

three approaches. Building on this direction, this thesis explores ways to extend the hybrid approach. Specifically, we focus on addressing the limitations associated with a differentiable renderer. The challenges and our proposed solutions will be briefly described in the following sections.

1.2 Challenges

In this thesis, we address the following challenges associated with the use of a differentiable renderer.

1.2.1 Difficulty of optimisation

In general, a sufficiently large convolutional neural network can learn the mapping from an input image to a latent 3D representation, due to its redundant parameterisation, if it is directly supervised. However, if the differentiable rendering layer is applied to the output of a convolutional neural network, parameters obtained through the

differentiable rendering layer have physical meaning based on its design. This means that the obtained parameters are no more redundant. Therefore, training based on a differentiable renderer becomes more difficult and requires careful adjustment of hyperparameters such as parameter-wise learning rates like in conventional gradient descent non-linear optimisation.

1.2.2 Need for additional supervision

Image-based alignment methods generally rely on the smooth-nature of natural images or objects. Therefore, if network training is conducted only based on the difference between an input image and a rendered image, the supervisory signal becomes meaningful only if the estimation is sufficiently close to the optimum. This causes the need for additional supervision such as landmarks that can work as meaningful supervision when the geometric displacement is large. It also stabilises training by preventing a network from updating parameters by erroneous estimates.

1.2.3 Discontinuity in self-occlusion

The difficulty of optimisation through the rendering process lies in the fact that rasterizing a 3D object onto discrete pixels is fundamentally non-differentiable. If a differentiable renderer is naively applied, the gradient through the rendering layer is calculated based on tentative correspondences between a pixel and a mesh. When the network parameters are updated, the tentative correspondences also change. Such updates sometimes cause new self-occlusions, leading to drastic changes in pixel-mesh correspondences. To train 3D reconstruction network for an object that has significant self-occlusions, discontinuity should be addressed or taken into account.

1.3 Our proposals

In this thesis, we will seek ways to address challenges in the use of a differentiable renderer. Our goal is to answer the following questions: (1) How can the convergence of training based on a differentiable renderer be improved? (2) Can we remove auxiliary

landmark supervision for statistical model fitting while still achieving stable training? (3) How can semantic segmentation be leveraged as an alternative to landmarks for auxiliary supervision? (4) Can we train a neural network using a statistical model but without using the differentiable renderer?

To this end, we will introduce two novel technical elements to differentiable rendering. One element is a differentiable linear least-squares layer that can be combined with a 3D reconstruction network. This fixed function layer conducts linear least-squares fitting of a model to inputs or outputs of a neural network in a differentiable manner. To achieve the analytically differentiable model, we introduce the inverse lighting model based on spherical harmonics to linearise the least-squares problem. On the one hand, this technique can improve the optimality of the estimation as linear least-squares assures nearly perfect fitting of the model. On the other hand, we can use this layer to regularise outputs of a neural network. In this thesis, we show regularisation of estimated image-to-model correspondence maps by differentiable linear least-squares. The other new element is distance metrics for a semantic segmentation map. A semantic segmentation map is image-like data in which semantic labels are assigned to each pixel, and can be used as an alternative to landmark supervision. One straightforward way to use segmentation as supervision is assigning segmentation labels to each point on a 3D shape, and rendering it with a differentiable renderer. However, the weakness of differentiable renderers in aligning large displacements reduces the advantage of supervision based on segmentation map. To address this issue, we propose a novel distance metric for segmentation map alignment based on Geometric Rényi Divergence (GRD). Finally, we show applications of these new techniques to self-supervised monocular 3D face reconstruction. One application is fully self-supervised monocular 3D face reconstruction, which combines an image-to-model correspondence prediction network with a linear least-squares layer. This combination enabled the introduction of a robust loss function, which contributes to the stabilisation of the training. Thereby, this approach only relies on input images and does not require additional supervision such as landmarks. Another application is GRD-based supervision of monocular 3D face reconstruction. The long-range non-saturating gradient signal of GRD enables network training based on images with large displacement and rotation. The other

application is monocular 3D face reconstruction without a differentiable renderer. In this approach, we completely remove the forward rasterisation layer from the training architecture while using a statistical face model and a perspective projection model.

1.4 Outline

The remainder of this thesis is organised as follows:

- **Chapter 2. Related work**: We show prior works related to 3D reconstruction, statistical shape modelling, and differentiable rendering. We then discuss limitations of existing works from the perspective of self-supervised monocular 3D reconstruction.
- **Chapter 3. Differentiable least-squares model fitting**: We propose utilising statistical models of shape and texture to regularise the outputs of neural networks in a differentiable manner.
- **Chapter 4. Differentiable cohesive measure**: We propose cohesive measures to use a segmentation map as supervision with a differentiable renderer or neural network.
- **Chapter 5. Self-supervised monocular 3D face reconstruction**: We propose architectures that incorporate the ideas from the preceding two chapters for the purposes of self-supervised learning with model-based regularisation and segmentation-based supervision.
- **Chapter 6. Conclusion**: We summarise our research and discuss limitations and potential future works.

Chapter 2

Related work

In this thesis, we specifically focus on monocular 3D reconstruction based on statistical 3D models. Our research direction is to relax the requirements for supervision and improve the performance in this task by leveraging a physical model of image formation process. We also introduce supervision based on a semantic segmentation map as an alternative to landmarks. In this direction, we employ image-to-model correspondence estimation networks for the sake of reducing the need for landmark supervision and to enable segmentation-based supervision.

In this chapter, we review existing works on 3D face reconstruction based on 3D morphable models and analysis-by-synthesis. Subsequently, we review deep-learning-based regression approaches, and hybrid approaches that combines deep learning with analysis-by-synthesis using a differentiable renderer. Since the practical target domain of our research is a human face, in this part, we mainly focus on 3D face reconstruction methods, and describe the concept of each approach. Then, we go through methods for image-to-model correspondence estimation, which is the key component of our approach. We also review differentiable renderers and discuss limitations. Since one of our key ideas is solving a subproblem within a network, we review in-network optimisation approaches. Lastly, we review utilisation of semantic segmentation for 3D reconstruction.

2.1 3D morphable model and analysis-by-synthesis

A 3D morphable model (3DMM) is the most successful and widely used representation of a human face. Early works in monocular 3D face reconstruction methods used the 3DMM representation. Blanz and Vetter [5] proposed the 3D morphable model, which represents the shape and texture of a face as a linear combination of bases. Each basis is represented as displacement of position and albedo of each vertex from the mean. The mean and basis are calculated via principal components analysis (PCA) of previously captured and aligned multiple faces. In this approach, 3D morphable model coefficients are calculated by minimising the error between pixel values in an input image and a rendered image, which is known as analysis-by-synthesis. Single image 3D face reconstruction usually requires solution of a non-convex and non-linear optimisation process, thus estimation takes significant computation time with no guarantee of obtaining the global minimum. The earliest work on 3DMM fitting used landmark distance as a sparse objective function for approximate initialisation and within an analysis-by-synthesis framework [6]. Subsequently, Romdhani and Vetter [58] used landmarks and occluding contours within a multi-feature fitting approach. Bas and Smith [4] explore to what extent geometric parameters can be estimated from landmarks and contours alone and show that this leads to an ambiguity between shape and face/camera distance. Many state-of-the-art methods still rely on landmarks for supervision. E.g. RingNet [60] trains a CNN to regress geometric parameters (shape and pose) from a single image using only landmark supervision and paired identity images. Beyond landmarks and contours, silhouettes and segmentation information have been much less widely used. In early work, Moghaddam et al. [51] used a binary silhouette loss across multiple images. Since this loss is discontinuous, they use the derivative-free Nelder–Mead optimisation method.

2.2 Deep neural networks for 3D reconstruction

In this section, we firstly review naive applications of convolutional neural networks (CNN) to a 3D face reconstruction task, where the task is formulated as regression. Subsequently, we review self-supervised 3D face reconstruction networks, which com-

bines CNN with analysis-by-synthesis. Lastly, we review 3D face reconstruction methods that formulates the task as image-to-image conversion while representing geometry or correspondence as pixel-wise information.

2.2.1 CNN-based 3D face reconstruction

Recent advancement in deep neural networks have been applied for single image 3D face reconstruction. Richardson *et al.* [57] trains the network so that the difference between estimated 3D morphable coefficients and ground truth is minimised. One problem with these approaches is the scarcity of real 3D face datasets with ground truth. Instead of real data, Richardson *et al.* [57] employed synthetic data for training. Tran *et al.* [69] used parameters estimated by another method for training. However, there is still a huge domain gap between real and synthesised or estimated data. To mitigate this problem, Kim *et al.* [36] used synthesised images with ground truth as well as non-annotated real images with the bootstrapping procedure. This approach uses only synthesised images at first. As training proceeds, the distribution of training data is updated so that it reflects the distribution of estimated parameters based on the current network. Though this method exhibits robustness of reconstruction against pose variations, it requires cropping the central region of a face and relies on given facial landmarks. Besides 3DMM, other representations of a face are also investigated. Feng *et al.* [16] proposed a method which represents a reconstructed face as coordinates in UV map for each pixel. Jackson *et al.* [30] employs the 3D position of each pixel as the representation. Regardless of the representation, these methods rely on the existence of ground truth, pseudo ground truth or synthetic data for 3D faces, and they cannot be applied to objects, for which 3D ground truth is difficult to obtain.

2.2.2 Self-supervised 3D face reconstruction network

To employ a large quantity of image data without annotation for training, Tewari *et al.* [66] combined a CNN encoder with differentiable renderer. This network estimates 3D morphable model coefficients from a single image. The network parameters

are optimised so that the difference between the rendered image and the input image is minimised. The gradient through self-supervision is weak and the optimisation is prone to converge to a local minimum compared with direct supervision based on ground truth [11]. This method relies on 3D morphable basis to resolve the degeneracy between geometry, texture, and lighting. Therefore, the representation power of the reconstructed face is limited to that of 3D morphable model. Therefore, the estimated result tends to be blurry or less realistic. Tran *et al.* [70] uses encoder-decoder network as a nonlinear morphable model while regularising the parameters with the distance from linear 3D morphable model and learning fine textures based on adversarial training. Tewari *et al.* [64] proposed a method which trains corrective functions as well as regression network and optimise the parameters of corrective space at test time to reproduce fine details. These methods can reproduce the detail of input images and improve the quality of the result. However, they relies on the existence of 3DMM basis and facial landmarks. Thus, these methods cannot be applied to other objects, for which no landmarks and basis are provided.

2.2.3 Image-to-image methods

Going beyond model fitting, a number of methods make pixel-wise predictions. SFSNet [62] infers lighting and surface normal and albedo maps from single face images. Their training is bootstrapped using synthetic faces sampled from a model. Sela et al. [61] use an image-to-image network to predict facial depth and correspondence to a canonical model. The network is trained entirely supervised using synthetic data and model fitting requires an offline nonrigid registration to the estimated correspondences. Guler et al. [2] and Yu et al. [81] predict dense correspondence maps using an image-to-image network and supervision provided by landmark-based 3DMM fits. Feng et al. [16] predict a UV map from a 3D face to 2D image coordinates. Zhu et al. [87, 88] propose the projected normalised coordinate code (PNCC) as a representation for dense correspondence. Crispell and Bazik [10] augment PNCC with a predicted 3D offset. All of the above approaches are supervised. Several approaches [61, 81, 10] fit a model to estimated depth or correspondence, but this is done as an offline, nonlinear optimisation.

2.3 Differentiable rendering

Many differentiable renderers employ a simple image formation process. They perform a projection of vertices based on the camera projection model, calculate shading assuming the Lambertian reflection model under point light source or spherical harmonics lighting, and rasterise pixels. On the other hand, there are several different approaches to calculate gradients of losses through the rendering process. Tewari *et al.* [66] calculates pixel values on each vertex and minimises the difference between resampled pixel values and rendered pixel values. Resampled pixel values are calculated based on projected vertices using bilinear interpolation. Occlusion of vertices and correspondence between pixels and vertices are assumed as fixed during gradient calculation. Instead of vertex-wise error, Tran *et al.* [70] calculates error on each pixel comparing rasterised pixels and input pixels. During gradient calculation, the occlusion and pixel-vertex correspondence is fixed. To handle complex objects, which is far from convex shape and has diverse shape variation, calculation of occlusion is required. OpenDR [47] calculates gradient as the combination of vertex position gradient and pixel value gradient on an image. In this process, the gradient on the occlusion boundary is calculated by comparing the nearest face and the second nearest face. The fundamental challenge is that rasterisation of a continuous 3D object onto a discrete pixel grid is fundamentally not differentiable. Therefore, the gradient is only meaningful in neighboring pixels. That makes supervision signals weak and causes convergence to local minimum. Hence, approximations are used that provide useful smooth gradients. Neural 3D Mesh Renderer (NMR) [35] extrapolates a gradient outside triangles based on linear interpolation of the derivative across a triangle edge. Soft Rasterizer (SoftRas) [45] computes a soft (i.e. blurred) rasterisation of each triangle in a mesh. Petersen *et al.* [54] employs cycle adversarial loss to compensate the difference in appearance. However, as both methods still rely on optimisation of pixel value errors, they could fail to reconstruct fine structures of objects, which is difficult to understand and analyse by rasterisation process. Inverse graphics GAN [48] trains a differentiable neural network to approximate the behaviour of a non-differentiable classical renderer. Neural radiance fields (NeRF) [50] uses soft volume rendering of a non-binary density field. The problem with such approaches is that this softened rendering is only an

approximation whose quality depends on parameters controlling the softness. In practice, these parameters must be tuned or scheduled during optimisation to achieve a good fit to data [45].

2.4 Differentiable-renderer-based geometry reconstruction

Differentiable render is also applied to other types of geometry reconstruction tasks including category specific general 3D reconstruction, 3D scene reconstruction, and multi-view 3D reconstruction.

2.4.1 Category specific 3D reconstruction

Differentiable rendering has been applied to single image 3D reconstruction of general objects. Kanazawa [32] *et al.* proposed category specific single image 3D reconstruction based on self-supervision with differentiable renderer. The object is assumed to have the same topology as a sphere, and the deviation of vertex position and texture flow is obtained through the network from a single image. As this inverse problem is highly ill-posed, camera parameters are regularised based on category-wise structure from motion method as well as smoothness prior. Due to strong regularisation to overcome the ill-posedness, the reproduction of fine structures is still limited. Furthermore, this method relies on landmarks.

2.4.2 3D scene reconstruction based on RGBD self-supervision

Kaneko *et al.* [33] tries to apply differentiable rendering to a single image 3D reconstruction for non-fixed mesh topology scene. To handle the non-fixed connectivity of the mesh, the method employs disconnected triangle meshes. It also uses depth information from RGBD images both for supervision and inference. RGBD pixel values are optimised by a differentiable renderer through training. As this method employs disconnected meshes and does not use the explicit context of objects, the reconstructed geometry tends to be noisy.

2.4.3 3D reconstruction based on multi-view self-supervision

Pillai *et al.* [55] shows structure from motion based on CNN and differentiable renderer. The network estimates camera motion and depth map from the input image sequence. This approach intends to replace conventional structure from motion processing with a deep neural network. Though it does not require any landmarks, it uses adjacent frames in a video sequence for training, in which the displacement of geometry is small. Therefore, this method can be used only if a dataset of video sequences is available. Kato *et al.* [34] employed an adversarial loss that discriminates synthesised images and original images using multi-view images to resolve the ambiguity of 3D reconstruction. As this method needs silhouettes of objects for training, the scarcity of training data is still a problem. Lin *et al.* [43] used a 3D shape dataset to obtain a shape prior and trains a multi-view stereo network based on a differentiable renderer. As this method also relies on a 3D shape dataset, which is not abundant and precise, the quality of reconstruction is still limited.

2.5 In-network optimisation

In Chapter 3, we introduce a linear least-squares layer, which enables differentiable in-network optimisation. In this direction, we review methods that perform optimisation within a neural network pipeline. Early works from the pre-deep learning era in closely related areas include the variable projection method [24] and Wiberg matrix factorisation [75]. The variable projection method is a technique for solving nonlinear optimisation problems involving separable linear variables. In this technique, linear subvariables are analytically solved as a function of nonlinear variables, and then the nonlinear independent variables are optimised through a nonlinear optimisation technique. Wiberg matrix factorisation [75] can be viewed as an application of the variable projection method. This method calculates factorised matrices that approximate the original matrix. One factor matrix is solved as a linear problem, and the rest is optimised as a nonlinear problem. The concept of solving a subproblem separately within an entire problem is also applied to the training of a neural network. Kolotouros *et al.* [38] proposed training of a 3D human pose and shape estimation network by con-

ducting keypoint-based fitting from the output of a tentative network and supervising the network with the fitted result. Although this approach involves in-network optimisation, it is not combined with the training pipeline in a differentiable manner. Van Gansbeke *et al.* [73] introduced a layer that solves linear least-squares, which is differentiable, to estimate line parameters for lane detection and train a network by direct supervision on obtained line parameters, leading to end-to-end training. From the perspective that in-network optimisation can be viewed as a constraint, we review implicit representations for neural networks. DeepSDF employs the signed distance function (SDF) to represent the surface of a 3D object. Unlike NeRF [50], which explicitly represents occupancy and colour for each spatial point, DeepSDF represents the surface as an isosurface of the SDF. Although extracting the surface from the SDF involves a search, which can be considered optimisation, the training is conducted in a supervised manner using ground truth SDF. DVR [53] extends the SDF-based approach to unsupervised training with multi-view images. In DVR, surface search is conducted to establish a tentative surface, and then the implicit function theorem, which is analytical, is applied to obtain the gradient for backpropagation. NeuralODE [9] has a similar concept. NeuralODE models the dynamics of a system as an ordinary differential equation (ODE). During gradient calculation, NeuralODE solves the ODE with any arbitrary solver and calculates the gradient using the adjoint sensitivity method.

2.6 3D reconstruction meets semantic segmentation

In Chapter 4, we will show utilisation of semantic segmentation maps to align geometry, and experiments of network training based on semantic segmentation in Chapter 5. To this end, we review 3D reconstruction methods based on semantic segmentation supervision, and points-to-points distance measures, which is closely related to our proposed method in Chapter 4.

2.6.1 3D reconstruction based on semantic segmentation supervision

Recent 3D reconstruction works include a face parsing loss as one of a number of losses with which a face model fitting (i.e. parameter regression) CNN is trained [86, 8]. They do so simply by rasterising the semantic labels on the mesh using a differentiable renderer, [86] using a variant of SoftRas [45] and [8] using TF Mesh Renderer [21]. Note that the latter uses a hard rasterisation and does not provide any useful gradient for changes in rasterisation or, therefore, for aligning discrete semantic segments. Meanwhile, SoftRas compares a soft rasterisation to hard discrete input meaning that the minimum loss does not correspond to optimal alignment. No previous work, including [86, 8, 42], has considered the problem of estimating shape using only semantic segmentation information. Li et al. [42] learn both a deformable model and model fitting in a self-supervised fashion. One of their training objectives is to ensure semantic consistency, measured by projecting the semantically labelled 3D model into the image. They measure semantic loss using the Chamfer distance which is sensitive to sampling differences between pixels and vertices and tends to cause the model to shrink.

2.6.2 Points-to-points distance measures

When aligning point clouds to point clouds or vertices to pixels with unknown correspondence, a variety of soft distance measures have been considered to ensure a useful gradient is provided even from a poor initialisation. Of particular relevance to our work are those methods based on probabilistic representations. Jian and Vemuri [31] use the L2 distance between two mixture of Gaussians (MoG) for point cloud registration. Wang et al. [74] use closed-form Jensen Rényi divergence for MoG for group-wise point cloud registration. Yamashita et al. [79] represent volumetric point clouds using MoG and exploit this for fitting to 2D silhouettes using KL divergence, though they require stochastic Monte Carlo sampling and regularisation to obtain stable performance.

Method	Pros	Cons
3DMM fitting [5]	No training needed	Test time optimisation
Richardson [57]	No test time optimisation	Trained on synthetic data
Kim [36]	No test time optimisation	Trained on existing estimation
MoFA [66]	Unsupervised	Not robust, blurry
Tewari [64]	High fidelity	Need landmarks for training
RingNet [60]	High fidelity	Need landmarks for training
NeRF [50]	Reconstruct any object	Need multiview, no generalisation

Table 2.1: Summary of previous works

2.7 Conclusion

For conclusion, we summarise advantages and disadvantages of previous works in Table 2.1. In this chapter, we mainly reviewed self-supervised 3D face reconstruction methods that rely on a differentiable renderer and 3DMM. Self-supervised approaches have an advantage that they can leverage a large number of unannotated images for training network. A limitation of this approach is that it requires landmark supervision or face region mask for stable training. The reason is that differentiable rendering of 3DMM can explain only modeled face region by the template meshes. Thus, if a face model is rendered outside the face region on the image due to erroneous estimation, that causes a huge penalty in photometric reconstruction loss. This makes the network predict a shrunk shape because it is the statistically safe choice for the network assuming large errors near the boundary are inevitable. In this context, masking background or occluded region and constraining prediction by landmark correspondences are viewed as a straightforward solution. In MoFA [65], the authors state landmark supervision is optional. However, based on our experience to reproduce MoFA work, we can state it is very difficult to train a network without landmark supervision and very careful adjustment of learning rate on each estimated parameters is needed. We also find achieving high fidelity reconstruction is difficult as convergence is slow and unstable. This leads us to pursue four research directions: 1) combine image-model correspondence estimation with 3DMM and differentiable renderer and train a network so that each pixel becomes well explained by the model, which is opposite to previous differentiable-renderer-based methods, 2) improve fidelity of reconstruction

by introducing analytically optimal model fitting into the training pipeline, 3) leverage semantic segmentation dataset to achieve stable training expecting self-supervised segmentation will be available, and 4) seek ways to avoid using a differentiable renderer and hence avoid approximations of the softened rasterisation process or assumptions about self-occlusion.

Towards development of image-model correspondence estimation techniques, we reviewed face reconstruction methods that consist of image-to-image neural networks. We find existing works in this field rely on supervised training, and combining this with self-supervision is a promising research direction. We also reviewed 3D reconstruction methods based on semantic segmentation supervision. Most works in this field rely on modified implementations of existing differentiable renderers. This approach is sub-optimal because alignment between semantic segmentation maps sometimes requires: 1) a long-range non-saturating measure, 2) exact matching, and 3) special treatment for a certain semantic class depending of the meaning of the class. These features are unattainable by the existing differentiable renderers.

Chapter 3

Differentiable least-squares model fitting

3.1 Introduction

In this chapter, we propose *linear least-squares layers* to implicitly solve for optimal geometric and photometric parameters and describe methods to integrate these layers within the architecture and training pipeline of neural networks. Since linear least-squares problems can be solved by matrix operations, this layer is naturally differentiable. Our proposal is to use this fixed differentiable layer to regularise outputs of a neural network and generate physically meaningful expressions based on statistical and physical models. In this section, we will introduce the formulation of 3D morphable models and perspective projection (in Section 3.1.1), which are the foundational models in our approach. Subsequently, we show exemplar use-cases of linear least-squares layer for better understanding of the concept (in Section 3.1.2). In the remainder of this chapter, we will show photometirc, geometric, and combined linear least-squares layer.

3.1.1 3D morphable model and image formation

Here, we introduce notations about 3D morphable model and perspective projection of it. 3DMM is a widely used statistical representation of a human face. 3DMM represents the shape and texture of a face as a linear combination of bases. Each basis is represented as displacement of position and albedo of each vertices from the mean. We represent 3D face models based on a 3DMM:

$$\mathbf{v}_j = \sum_{i=1}^{N_g} \alpha_i \mathbf{S}_{ij} + \bar{\mathbf{s}}_j, \quad \mathbf{r}_j = \sum_{i=1}^{N_r} \beta_i \mathbf{A}_{ij} + \bar{\mathbf{a}}_j, \tag{3.1}$$

where $\mathbf{v}_j$ is the 3D position and $\mathbf{r}_j$ is the RGB reflectance of jth vertex respectively. $\mathbf{S}_{ij}$ is the ith linear basis of the vertex position and $\bar{\mathbf{s}}_j$ is its mean. In the same manner, $\mathbf{A}_{ij}$ is the ith linear basis of the vertex reflectance and $\bar{\mathbf{a}}_j$ is its mean. In typical 3DMM, geometric components of the model are separated into shape and expression. Shape components are associated with the identity of individual people and expression components represent facial expression. Since both have the same mathematical meaning, we merged them into the vertex position bases and mean by concatenating the basis vectors and summing the mean vectors. We denote the number of vertex position bases as $N_g = N_s + N_e$, where N_s is the number of shape dimensions and N_e is the number of expression dimensions. α_i and β_i is the coefficient of the linear combination and that is the representation of 3D face model which we use. We use the Basel Face Model 2017 [23] as the basis of our representation which has $N_s = 199$, $N_e = 100$, and $N_r = 199$ dimensions for facial identity shape, facial expression shape, and skin reflectance respectively.

Each vertex is projected onto the image plane based on a full perspective camera model:

$$\lambda_j \begin{bmatrix} \mathbf{x}_j \\ 1 \end{bmatrix} = \mathbf{K}(\mathbf{R}\mathbf{v}_j + \mathbf{t}), \tag{3.2}$$

where $\mathbf{x}_j$ is jth projected vertex position, $\mathbf{K}$ is an intrinsic camera matrix, $\mathbf{R}$ is a 3D rotation matrix, and $\mathbf{t}$ is a 3D translation vector. In addition, each vertex is

shaded using spherical harmonic lighting for image generation and supervision based on photometric discrepancy:

$$\mathbf{i}_j = \mathbf{r}_j \sum_{k=1}^{27} \gamma_k \mathbf{f}_k(\mathbf{n}_j), \tag{3.3}$$

where $\mathbf{i}_j$ is jth shaded vertex colour, $\mathbf{f}_k$ is a function to obtain kth spherical harmonic basis from jth vertex normal $\mathbf{n}_j$, γ_k is the coefficient for the kth basis. We employ second order spherical harmonic lighting, which has 9 bases for each colour channel. We calculate $\mathbf{n}_j$ by averaging the surface normal of neighbouring faces of each vertex.

3.1.2 Exemplar use-cases of linear least-squares layer

Figure 3.1 shows exemplary use-case of linear least-squares layer. Typical use-cases are divided into three categories: photometric, geometric, and hybrid. Figure 3.1(a) is an example of a neural network combined with photometric linear least-squares layer. In the naive context of model-based self-supervision like MoFA [66], the neural network outputs 3DMM coefficients for both shape and albedo as well as pose parameters and illumination parameters. Predicted parameters form a set of projected and illuminated vertices, which has both 2D position and colour values. The neural network is trained so that the difference in pixel values between input pixels and projected vertices is minimised. On the other hand, in our proposed approach, the network only provides 3DMM parameters for geometry, which is typically shape, expression, and pose parameters. Based on the estimated geometric parameters, pixel values are sampled from the input image. The linear least-squares layer takes sampled pixel values and 3DMM mean and basis vectors for albedo, and computes fitted photometric parameters. The loss value for network training is calculated based on the error between the fitted appearance and the input image. As the layer is differentiable, the training signal derived from fitted appearance is transmitted to the neural network beyond the linear least-squares layer. In this scenario, the optimality of output photometric parameters is improved as the linear least-squares assures optimal fitting conditional on the current model-image alignment. The technical detail will be described in Section 3.2.

Next, we explain a use-case for geometric linear least-squares in Figure 3.1(b). In this scenario, the neural network outputs a correspondence map from an input

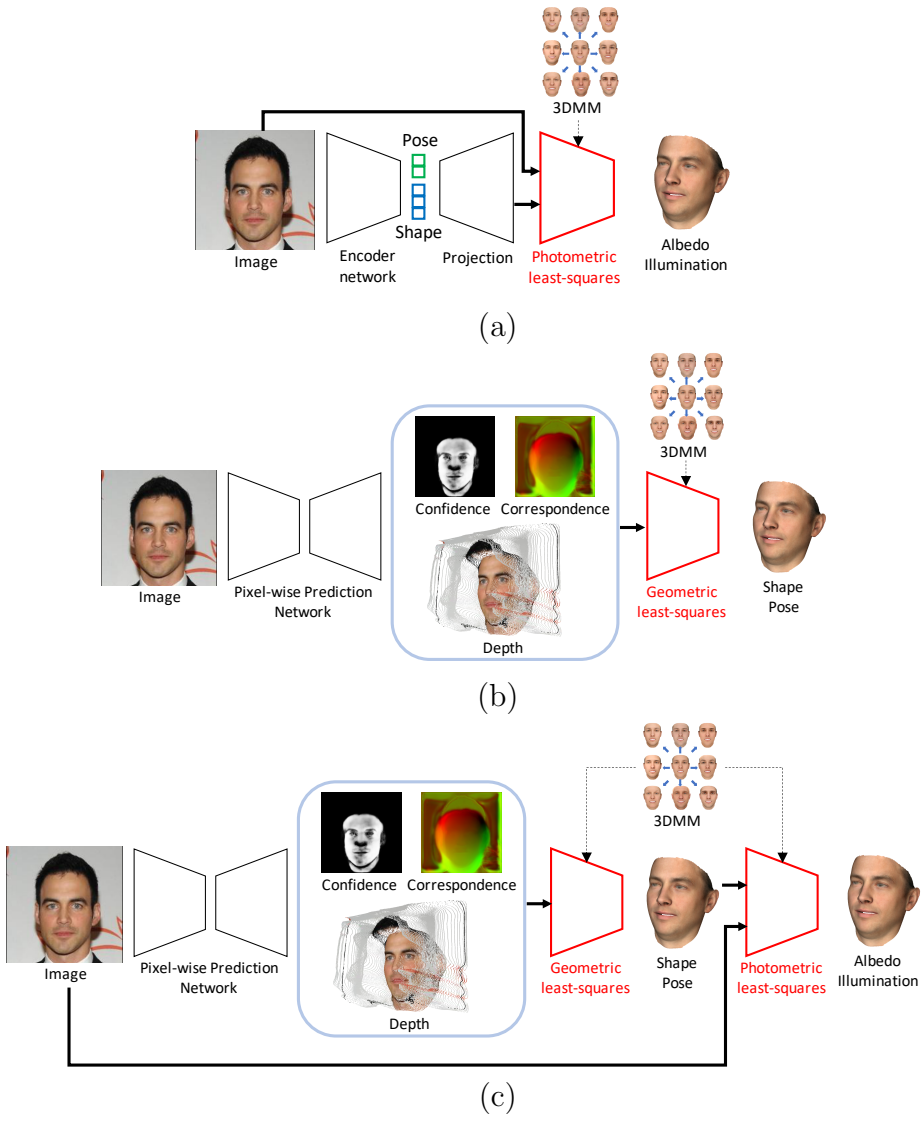


Figure 3.1: Exemplar use-cases of linear least-squares layers: (a) Photometric linear least-squares layer predicts 3DMM albedo coefficients and spherical harmonics coefficients for inverse lighting using input pixel values, shape, expression coefficients, and camera parameters; (b) Geometric linear least-squares layer predicts 3DMM shape coefficients and camera parameters from correspondence, depth, and confidence maps; (c) Hybrid linear least-squares layer combines both photometric and geometric functions to predict 3DMM albedo, shape, expression coefficients, camera parameters, and spherical harmonics coefficients using input pixel values, correspondence map, depth map, and confidence map.

image to template meshes of the 3DMM along with a depth map. We apply the linear least-squares layer to the network output. The model of linear least-squares layer can be weak perspective projection, perspective projection, 3D alignment, and 3DMM geometry, and the layer produces 3D geometry of an object as shape and pose parameters. During training, the residual of least-squares is minimised and the network learns to generate correspondence and depth map, which is consistent with the space of the model. The estimated geometry is also used to project vertices onto the image plane and arbitrary loss function is applied. The technical detail will be described in Section 3.3.

Lastly, a hybrid of both photometric and geometric linear least-squares is shown in Figure 3.1(c). In this scenario, the geometric linear least-squares layer is applied in the same manner as the purely geometric one. Subsequently, based on fitted geometric parameters, photometeric linear least-squares layers is applied (details in Section 3.4).

Motivation It is perhaps not obvious why these alternate formulations might be promising over current methods. The main difference in the second and third exemplar use cases compared to previous work is that, instead of image-to-3DMM parameter regression with a contractive CNN, we propose to estimate a dense image-model correspondence map with an image-to-image CNN architecture. The main difference in the first exemplar use case is that photometric parameters are not regressed, rather they are solved for using the input image data and the estimated geometric parameters. We argue that there are a number of significant benefits in these approaches:

1. A correspondence map is a minimal representation from which all 3DMM parameters can be estimated. One perspective on this is that using a CNN to predict both geometric and photometric parameters, as done in all previous work [66, 36, 22, 64, 11], is redundant.

2. The estimated parameters are least-squares optimal with respect to the input image and estimated correspondence map. Optimality for a given image is not guaranteed for a parameter regression CNN whose training objective seeks optimality only in aggregate over the whole training set.

3. Image-to-image CNNs are well suited to estimating correspondence maps with invariance to 2D transformations. Intuitively, it is enough for the correspondence CNN to learn "part detectors" with robustness to 2D rotation (translation invariance comes from the translation invariance of convolution layers). On the other hand, contractive CNNs are ill-suited to directly regressing geometric parameters with 2D transformation invariance [44]. This is because spatial information is lost in contractive layers and fully connected layers must exhaustively represent both features and their locations to reason about geometric parameters.

4. Image-to-image CNNs are much smaller than parameter regression networks due to the lack of fully connected layers. Concretely, we require roughly $10\times$ fewer parameters than previous CNN based approaches (13.4M parameters for our U-Net versus, for example, 138M parameters in VGG-face used by [22, 66]).

5. Every pixel in the input image can contribute to the losses during training. Previous model-based methods learn only from the parts of the image covered by the geometry of the current 3DMM estimate. In our approach, there is no longer a shortcut for the network to reduce reconstruction loss by shrinking the model to avoid difficult pixels.

6. We defer estimation of actual face geometry. Correspondence is an intermediate representation from which we infer geometry. At test time, if we have access to calibration information or have multiple images from the same camera (e.g. a video), we can exploit these constraints when we finally compute shape from the estimated correspondence map(s). Parameter regression networks cannot do this – they commit to an explanation of the shape and camera parameters for a single image with no way to inject calibration information or constraints post hoc.

Alternatively, our approach can be viewed as a means to learn dense face alignment (correspondence estimation) using model fitting as a form of self-supervision. Correspondence is, in itself, a useful representation. Once trained, the 3DMM can be discarded and the correspondence estimation network used for tasks such as landmarking or semantic segmentation without ever requiring ground truth labels for supervision.

3.2 Photometric linear least-squares

In this section, we describe in detail the formulation of our photometric linear least-squares layer. This layer takes pixel values sampled from a given input image and predefined 3DMM for albedo components, and calculates illumination parameters and albedo coefficients of 3DMM by solving linear least-squares. The key of this technique is to formulate appearance in terms of *inverse lighting*. Relying on inverse lighting, we define the objective function of the least-squares system as the error in albedo space. In the remainder of this section, we show the detail of inverse lighting based on spherical harmonics in Section 3.2.1, and linear least-squares layer based on the inverse lighting in Section 3.2.2.

3.2.1 Inverse spherical harmonics lighting

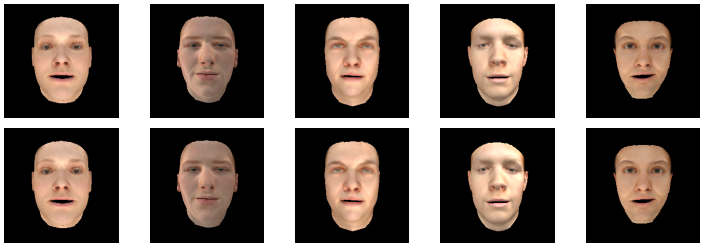
Spherical harmonic lighting [52] is a widely used representation for reflectance under environment illumination [65]. Appearance is modelled as the product of diffuse albedo and shading which is in turn represented as a linear combination of spherical harmonics basis functions shown in Equation (3.3).

This expression is bilinear in diffuse albedo and the spherical harmonic lighting coefficients. We reformulate this model such that it is linear in both simultaneously. This means that, given the geometric information (and hence the surface normals and model-image correspondence), we can directly infer reflectance and lighting parameters by solving a linear least-squares problem.

In contrast to the conventional model, we use spherical harmonics to represent *inverse lighting*. That is, a quantity that (when multiplied by the image intensity) removes the effect of shading, giving the diffuse albedo. In other words, we use the spherical harmonic basis functions to represent the reciprocal of diffuse shading. To this end, we represent the inverse shaded pixel value of the jth sampled pixel $\hat{\mathbf{r}}_j$ as:

$$\hat{\mathbf{r}}_j = \mathbf{i}_j \cdot \sum_{i=1}^{27} \gamma_i \, \mathbf{f}_i(\mathbf{n}_j), \tag{3.4}$$

where $\mathbf{f}_i(\mathbf{n}_j)$ are the spherical harmonic basis functions for normal direction $\mathbf{n}_j$. We



Regular SH					
Inverse SH					
Max error	0.0489	0.1086	0.1229	0.2718	0.1681
RMS error	0.0188	0.0430	0.0294	0.0582	0.0462

Figure 3.2: Empirical validation of inverse spherical harmonic lighting model.

employ 9 spherical harmonics basis for each RGB colour. Contrast this to Equation (3.3). We use the same spherical harmonic basis but now to effectively divide out the shading from the intensity yielding the albedo. Note that the effect of γ_k in Equation (3.4) is totally different from that of γ_i in Equation (3.3).

Since the reciprocal of inverse lighting effect in Equation (3.4) is not exactly the same as the lighting effect in Equation (3.3), it is not guaranteed that reversing the inverse lighting will accurately reproduce the forward lighting. We empirically validate this model in Figure 3.2. The upper row shows randomly generated images based on conventional SH lighting. We generate random SH coefficients by $\sigma = 0.2$ and add the random lighting to constant lighting which intensity is 0.9. We use the same SH coefficients for all RGB channel. Lower row shows images of the same faces rendered based on inverse SH lighting. Inverse SH coefficients are calculated as a least square solution that minimises the difference between estimated inverse lighting and inverted original lighting at random 100,000 sample points on the sphere. We also show the mean and max pixel errors. We assume the pixel value in both images and 3DMM is scaled to $[0, 1]$.

These results demonstrate that a variety of complex illumination conditions that are representable by conventional spherical harmonic lighting can be almost exactly recreated using our inverse lighting model.

However, the limitation of the inverse lighting model becomes apparent with dark pixels, which have values close to zero. In such pixels, the ideal inverse lighting might be excessively steep, leading to significant inconsistencies between forward and inverse

spherical harmonics lighting. Additionally, this can distort the noise distribution.

To mitigate the numerical instability caused by dark pixels, we clamp low pixel values of an input image. Specifically, we apply softplus function to input image as preprocessing: $i_{x,y} = \log(1 + e^{\xi \cdot i_{x,y}})/\xi$ where ξ is a parameter to adjust the scale of softplus function. We also apply inverse function of softplus function to visualise output images. We use $\xi = 4$.

3.2.2 Solving photometric linear least-squares

We calculate optimal reflectance parameters $\{\beta\}_{i=1,2,\ldots,N_r}$ and inverse lighting parameters $\{\gamma\}_{i=1,2,\ldots,27}$ by minimising E_{photo}, the error between the model albedo and that implied by inverse shading the image intensities:

$$E_{\text{photo}} = \sum_{j=1}^{N_v} c_j(\boldsymbol{\rho}) \|\hat{\mathbf{r}}_j(\boldsymbol{\rho}, \gamma_1, \gamma_2 \ldots, \gamma_{27}) - \mathbf{r}_j(\boldsymbol{\rho}, \beta_1, \beta_2 \ldots, \beta_{N_r})\|_2^2 + \sum_{i=1}^{27} \eta_i \gamma_i^2 + \sum_{i=1}^{N_r} \omega_i \beta_i^2 \tag{3.5}$$

where ω_i and η_i are the weight values for regularisation and regarded as a fixed parameter. $\boldsymbol{\rho}$ is geometric parameters that determines correspondence between the input image and the template meshes such as α_i in Equation (3.1) and $\mathbf{K}$,$\mathbf{R}$, and $\mathbf{t}$ in Equation (3.2). We assume an inversely illuminated pixel value $\hat{\mathbf{r}}_j$ and albedo bases and mean regarding to $\mathbf{r}_j$ are properly sampled from the input image and 3DMM based on the geometric parameters $\boldsymbol{\rho}$. c_j is confidence value of each sample, reflecting the confidence of estimates and visibility of the sample. We solve using linear least-squares:

$$\{\acute{\beta}_1, \acute{\beta}_2, \ldots, \acute{\beta}_{N_r}, \acute{\gamma}_1, \acute{\gamma}_2, \ldots, \acute{\gamma}_{N_i}\} = \underset{\beta_1, \beta_2, \ldots, \beta_{N_r}, \gamma_1, \gamma_2, \ldots, \gamma_{N_i}}{\arg\min} E_{\text{photo}}. \tag{3.6}$$

This expression has a closed form solution and derivative via the pseudoinverse. Note that visibility of vertex w_i is regarded as fixed value in each training step though it

depends on geometric parameters $\boldsymbol{\rho}$. Finally, we can compute the residual error:

$$\acute{E}_{\text{photo}} = \sum_{j=1}^{N_v} \|\hat{\mathbf{r}}_j(\boldsymbol{\rho}, \acute{\gamma}_1, \acute{\gamma}_2, \ldots, \acute{\gamma}_{N_i}) - \mathbf{r}_j(\boldsymbol{\rho}, \acute{\beta}_1, \acute{\beta}_2, \ldots, \acute{\beta}_{N_r})\|_2^2. \tag{3.7}$$

As this function is differentiable with respect to geometric parameters $\boldsymbol{\rho}$, $\acute{E}_{\text{photo}}$ can provide supervision through residual error of reconstruction using back propagation.

We now explicitly derive the solution in matrix form for a scenario where photometric errors are defined on each vertex, and pixel values are sampled from the input image based on the projection of the vertices. Assuming, on the jth sampled pixel, $\mathbf{i}_j = \mathbf{i}(\mathbf{x}_j(\boldsymbol{\rho}))$ is the pixel value, F_j the inverse lighting spherical harmonic basis, c_j is a confidence value, $\mathbf{A}_j$ the 3DMM albedo basis matrix, and $\bar{\mathbf{a}}_j$ the 3DMM albedo mean, then the optimal spherical harmonic coefficients $\boldsymbol{\gamma}$ and 3DMM albedo coefficients $\boldsymbol{\beta}$ can be found via the pseudoinverse as:

$$\begin{bmatrix} \boldsymbol{\gamma} \\ \boldsymbol{\beta} \end{bmatrix} = (\boldsymbol{\Lambda}^T \boldsymbol{\Pi} \boldsymbol{\Lambda})^{-1} \boldsymbol{\Lambda}^T \boldsymbol{\Pi} \boldsymbol{\Xi}. \tag{3.8}$$

where

$$\boldsymbol{\Lambda} = \begin{bmatrix} \mathbf{i}_1 \cdot \mathbf{F}_1 & -\mathbf{A}_1 \\ \mathbf{i}_2 \cdot \mathbf{F}_2 & -\mathbf{A}_2 \\ \vdots & \vdots \\ \mathbf{i}_{N_v} \cdot \mathbf{F}_{N_v} & -\mathbf{A}_{N_v} \\ \mathbf{E}_{27\times 27} & \mathbf{0}_{27\times N_r} \\ \mathbf{0}_{N_r\times 27} & \mathbf{E}_{N_r\times N_r} \end{bmatrix}, \tag{3.9}$$

$$\boldsymbol{\Pi} = \text{diag}\left(c_1, c_1, c_1, c_2, c_2, c_2, \ldots, c_{N_v}, c_{N_v}, c_{N_v}, \eta_1, \eta_2, \ldots, \eta_{27}, \omega_1, \omega_2, \ldots, \omega_{N_r}\right), \tag{3.10}$$

and

$$\mathbf{\Xi} = \begin{bmatrix} \bar{\mathbf{a}}_1 \\ \bar{\mathbf{a}}_2 \\ \vdots \\ \bar{\mathbf{a}}_{N_v} \\ \mathbf{0}_{27\times 1} \\ \mathbf{0}_{N_r\times 1} \end{bmatrix}, \tag{3.11}$$

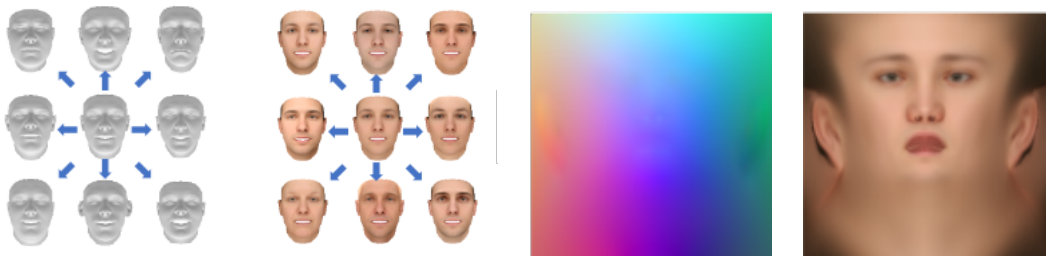
where η_i and ω_i represent the weight for regularisation.

3.3 Geometric linear least-squares

In this section, we describe the application of linear least-squares layer to regularisation of geometric information predicted by a neural network. Specifically, this layer takes a correspondence map that relates each pixel on the input image to a point on the template mesh of the 3DMM. For computational efficiency, we precompute a UV-warped 3D morphable model, which is a multi-channel 2D image representation of a 3DMM. In Section 3.3.1, we introduce the detail of UV-warped 3D morphable model first. Subsequently, in Section 3.3.2, we describe the detail of our geometric linear least-squares layer.

3.3.1 UV-warped 3D morphable model

We assume that the geometric least-squares layer samples the 3DMM mean and basis for each pixel based on predicted correspondences. Therefore, for computational efficiency, we flatten the 3DMM to a 2D parameterisation beforehand. Specifically, we generate a Tutte embedding [19] for each component of the 3DMM. We force the boundary of the embedding to be square. We refer to the flattened 3DMM as UV-3DMM and its domain of definition as UV space. To fill a hole inside the mouth of the Basel Face Model 2017, we introduce an auxiliary vertex inside the hole and connect it with the boundary vertices of the mouth. We set the mean value of mouth boundary



$\mathbf{v}_j$ (a) $\mathbf{r}_j$ (b) $\mathbf{v}_{\boldsymbol{\alpha}}(u,v)$ (c) $\mathbf{r}_{\boldsymbol{\beta}}(u,v)$ (d)

Figure 3.3: A 3D morphable model of geometry (a) and albedo (b) can be interpolated to a UV space (c,d) via an embedding. We refer to this as a UV-3DMM.

vertices for each component of the auxiliary vertex.

Via barycentric interpolation we can compute a linear shape and texture model for any position, $(u,v) \in [-1,1] \times [-1,1]$, in UV space. Accordingly, we write $\mathbf{s}^i(u,v)$, $\bar{\mathbf{s}}(u,v)$, $\mathbf{a}^i(u,v)$ and $\bar{\mathbf{a}}(u,v)$ for the interpolated ith shape basis, shape mean, ith albedo basis and albedo mean at arbitrary location in UV space (u,v). Note that (u,v) is continuous and the barycentric interpolation amounts to taking linear combinations of basis and mean values at the original vertex positions.

The 3D position of the model interpolated at UV coordinate (u,v) is:

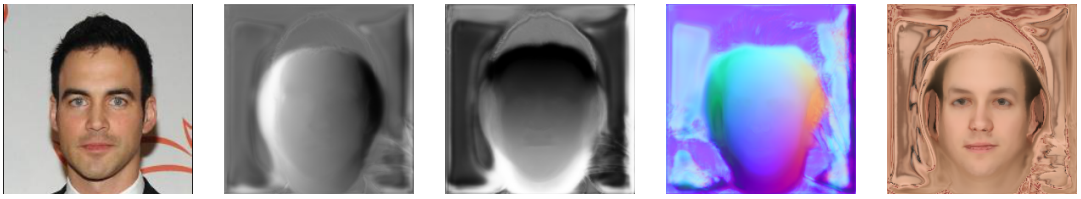
$$\mathbf{v}_{\boldsymbol{\alpha}}(u,v) = \mathbf{S}_{u,v}\boldsymbol{\alpha} + \bar{\mathbf{s}}(u,v), \tag{3.12}$$

where $\mathbf{S}_{u,v} = [\mathbf{s}^1(u,v), \ldots, \mathbf{s}^{N_g}(u,v)]$ are the stacked shape bases for the model interpolated at UV position (u,v). Similarly, we can write the model albedo interpolated at UV position (u,v):

$$\mathbf{r}_{\boldsymbol{\beta}}(u,v) = \mathbf{A}_{u,v}\boldsymbol{\beta} + \bar{\mathbf{a}}(u,v), \tag{3.13}$$

where again $\mathbf{A}_{u,v} = [\mathbf{a}^1(u,v), \ldots, \mathbf{a}^{N_r}(u,v)]$ are the stacked albedo bases for the model interpolated at UV position (u,v).

We refer to $\mathbf{v}_{\boldsymbol{\alpha}}(u,v)$ and $\mathbf{r}_{\boldsymbol{\beta}}(u,v)$ as a *UV-3DMM* (see Figure 3.3). Now, suppose that we are given a correspondence map between a face image, $\mathbf{i}(x,y)$, and the UV space of our 3DMM, i.e. we are given two maps: $u(x,y)$ and $v(x,y)$ defined for each pixel $(x,y) \in \{1,\ldots,N_W\} \times \{1,\ldots,N_H\}$ in the face image. Each pixel provides a



$\mathbf{i}(x,y)$ (a) $u(x,y)$ (b) $v(x,y)$ (c) $\mathbf{v}_{\boldsymbol{\alpha}}(u(x,y),v(x,y))$ (d) $\mathbf{r}_{\boldsymbol{\beta}}(u(x,y),v(x,y))$ (e)

Figure 3.4: Using estimated correspondences (b,c) from an image (a) to the UV space of the 3DMM, we can define a pixel-3DMM of geometry (d) and albedo (e) in pixel space as a function of 3DMM parameters.

correspondence between image and model. We can now interpolate our 3DMM at each pixel, via the correspondence map, giving a *pixel-3DMM*: $\mathbf{v}_{\boldsymbol{\alpha}}(u(x,y),v(x,y))$ and $\mathbf{r}_{\boldsymbol{\beta}}(u(x,y),v(x,y))$ (see Figure 3.4)

3.3.2 Solving geometric linear least-squares

We calculate optimal 3DMM shape coefficients $\{\alpha\}_{i=1,2,\ldots,N_g}$, where N_g is the total number of shape bases and expression bases, and camera parameters $\mathbf{H}$, which is a 3×4 matrix, by minimising E_{geo}, the error between points on a depth map $d(\mathbf{x})$ and corresponding points $\mathbf{v}_{\boldsymbol{\alpha}}(u(\mathbf{x}),v(\mathbf{x}))$ on 3DMM accumulated over $\mathbf{x} = (x,y) \in \{1,\ldots,N_W\} \times \{1,\ldots,N_H\}$:

$$E_{\text{geo}} = \sum_{x=1}^{N_w} \sum_{y=1}^{N_H} c(\mathbf{x}) \|\hat{\mathbf{v}}(\mathbf{x},\mathbf{H}) - \mathbf{v}_{\boldsymbol{\alpha}}(u(\mathbf{x}),v(\mathbf{x}))\|_2^2 + \sum_{i=1}^{N_g} \lambda_i \alpha_i^2, \tag{3.14}$$

where

$$\hat{\mathbf{v}}(\mathbf{x},\mathbf{H}) = \mathbf{H} \begin{bmatrix} d(\mathbf{x})\mathbf{x} \\ d(\mathbf{x}) \\ 1 \end{bmatrix}, \tag{3.15}$$

and λ_i is the weight value for regularisation and regarded as a fixed parameter. $\mathbf{x}$ represents predefined image coordinates and a confidence value $c(\mathbf{x})$, UV-correspondence $(u(\mathbf{x}),u(\mathbf{x}))$, and a depth value $d(\mathbf{x})$ are variables on each point. We solve using linear

least-squares:

$$\{\acute{\mathbf{H}}, \acute{\boldsymbol{\alpha}}\} = \arg\min_{\mathbf{H},\boldsymbol{\alpha}} E_{\text{geo}}. \tag{3.16}$$

This expression has a closed form solution and derivative via the pseudoinverse. Finally, we can compute the residual error:

$$\acute{E}_{\text{geo}} = \sum_{x=1}^{N_w} \sum_{y=1}^{N_H} \|\hat{\mathbf{v}}(\mathbf{x}, \acute{\mathbf{H}}) - \mathbf{v}_{\acute{\boldsymbol{\alpha}}}(u(\mathbf{x}), v(\mathbf{x}))\|_2^2. \tag{3.17}$$

As this function is differentiable with respect to a confidence value $c(\mathbf{x})$, UV-correspondence $(u(\mathbf{x}), u(\mathbf{x}))$, and a depth value $d(\mathbf{x})$, E_{geo} can provide supervision through residual error of reconstruction using back propagation.

Again, we derive the optimal solution in matrix form. Assuming, on jth pixel, $\mathbf{x}_j = (x_j, y_j, 1)^T$ is pixel coordinates, d_j is depth value, c_j is confidence value, $\mathbf{S}_j$ is 3DMM shape basis matrix, and $\bar{\mathbf{s}}_j$ is 3DMM shape mean, then each element of camera parameters $\mathbf{H}$ and 3DMM shape coefficients $\boldsymbol{\alpha}$ are given by:

$$\begin{bmatrix} \acute{H}_{1,1} \\ \acute{H}_{1,2} \\ \acute{H}_{1,3} \\ \acute{H}_{1,4} \\ \acute{H}_{2,1} \\ \acute{H}_{2,2} \\ \acute{H}_{2,3} \\ \acute{H}_{2,4} \\ \acute{H}_{3,1} \\ \acute{H}_{3,2} \\ \acute{H}_{3,3} \\ \acute{H}_{3,4} \\ \acute{\boldsymbol{\alpha}} \end{bmatrix} = (\boldsymbol{\Theta}^T \boldsymbol{\Omega} \boldsymbol{\Theta})^{-1} \boldsymbol{\Theta}^T \boldsymbol{\Omega} \boldsymbol{\Upsilon}, \tag{3.18}$$

where

$$\mathbf{\Psi}_j = \begin{bmatrix} d_j\mathbf{x}_j^T & 1 & \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & d_j\mathbf{x}_j^T & 1 & \mathbf{0}_{3\times3} \\ \mathbf{0}_{3\times3} & \mathbf{0}_{3\times3} & d_j\mathbf{x}_j^T & 1 \end{bmatrix}, \tag{3.19}$$

$$\mathbf{\Theta} = \begin{bmatrix} \mathbf{\Psi}_1 & -\mathbf{S}_1 \\ \mathbf{\Psi}_2 & -\mathbf{S}_2 \\ \vdots & \vdots \\ \mathbf{\Psi}_{N_p} & -\mathbf{S}_{N_p} \\ \mathbf{0}_{N_g\times12} & \mathbf{E}_{N_g\times N_g} \end{bmatrix}, \tag{3.20}$$

$$\mathbf{\Omega} = \mathrm{diag}\left(c_1, c_1, c_1, c_2, c_2, c_2, \ldots, c_{N_p}, c_{N_p}, c_{N_p}, \lambda_1, \lambda_2, \ldots, \lambda_{N_g}\right), \tag{3.21}$$

$$\mathbf{\Upsilon} = \begin{bmatrix} \mathbf{a}_1 \\ \mathbf{a}_2 \\ \vdots \\ \mathbf{a}_{N_p} \\ \mathbf{0}_{N_g\times1} \end{bmatrix}, \tag{3.22}$$

and λ_i represents the weight for regularisation. $\mathbf{0}$ represents zero matrix and $\mathbf{E}$ represents identity matrix.

The obtained camera parameters $\acute{\mathbf{H}}$ are an inverse form of a perspective projection matrix. Now, we show the conversion of $\acute{\mathbf{H}}$ to camera parameters shown in Equa-

tion (3.2). By inverting $\acute{\mathbf{H}}$, we can describe camera parameters:

$$\psi\mathbf{KR} = \begin{bmatrix} \acute{\mathbf{p}}_1^t \\ \acute{\mathbf{p}}_2^t \\ \acute{\mathbf{p}}_3^t \end{bmatrix} = \begin{bmatrix} \acute{H}_{11} & \acute{H}_{12} & \acute{H}_{13} \\ \acute{H}_{21} & \acute{H}_{22} & \acute{H}_{23} \\ \acute{H}_{31} & \acute{H}_{32} & \acute{H}_{33} \end{bmatrix}^{-1}, \tag{3.23}$$

$$\psi\mathbf{Kt} = \acute{\mathbf{q}} = -\begin{bmatrix} \acute{\mathbf{p}}_1^t \\ \acute{\mathbf{p}}_2^t \\ \acute{\mathbf{p}}_3^t \end{bmatrix} \begin{bmatrix} \acute{H}_{14} \\ \acute{H}_{24} \\ \acute{H}_{34} \end{bmatrix}, \tag{3.24}$$

where $\mathbf{K}[\mathbf{R}\ \mathbf{t}]$ represents a classical projective camera matrix. To obtain a camera matrix, we decompose $\acute{\mathbf{H}}$, $\mathbf{q}$ into $\mathbf{K}, \mathbf{R}, \mathbf{t}$ as:

$$s = \|\acute{\mathbf{p}}_3\|_2, \tag{3.25}$$

$$\mathbf{r}_3 = \frac{\acute{\mathbf{p}}_3}{s}, \tag{3.26}$$

$$k_5 = \acute{\mathbf{p}}_3^t \mathbf{r}_3, \tag{3.27}$$

$$k_4 = \|\acute{\mathbf{p}}_2 - k_5\mathbf{r}_3\|_2, \tag{3.28}$$

$$\mathbf{r}_2 = \frac{\acute{\mathbf{p}}_2 - k_5\mathbf{r}_3}{k_4}, \tag{3.29}$$

$$k_3 = \acute{\mathbf{p}}_1^t \mathbf{r}_3, \tag{3.30}$$

$$k_2 = \acute{\mathbf{p}}_1^t \mathbf{r}_2, \tag{3.31}$$

$$k_1 = \|\acute{\mathbf{p}}_1 - k_2\mathbf{r}_2 - k_3\mathbf{r}_3\|_2, \tag{3.32}$$

$$\mathbf{r}_1 = \frac{\acute{\mathbf{p}}_1 - k_2\mathbf{r}_2 - k_3\mathbf{r}_3}{k_1}, \tag{3.33}$$

$$\mathbf{K} = \begin{bmatrix} k_1 & k_2 & k_3 \\ 0 & k_4 & k_5 \\ 0 & 0 & 1 \end{bmatrix}, \tag{3.34}$$

$$\mathbf{R} = \begin{bmatrix} \mathbf{r}_1^t \\ \mathbf{r}_2^t \\ \mathbf{r}_3^t \end{bmatrix}, \tag{3.35}$$

$$\mathbf{t} = \frac{1}{\psi}\mathbf{K}^{-1}\acute{\mathbf{q}}. \tag{3.36}$$

3.3.3 Variants of geometric least-squares

Geometric linear least-squares can be also applied to other models. Specifically, a lower-dimensional model can be used for the initialisation stages of neural network training as is described in Section 5.4.

3D alignment

We can apply linear least-squares to naive 3D alignment model without using a 3DMM. In this example, we use the 3DMM mean shape $\bar{\mathbf{s}}$ as target geometry and formulate target function E_{geo} as:

$$E_{\text{geo}} = \sum_{x=1}^{N_w}\sum_{y=1}^{N_H} c(\mathbf{x})\|\hat{\mathbf{v}}(\mathbf{x},\mathbf{H}) - \bar{\mathbf{s}}(u(\mathbf{x}),v(\mathbf{x}))\|_2^2, \tag{3.37}$$

where

$$\hat{\mathbf{v}}(\mathbf{x},\mathbf{H}) = \mathbf{H}\begin{bmatrix} d(\mathbf{x})\mathbf{x} \\ d(\mathbf{x}) \\ 1 \end{bmatrix}, \tag{3.38}$$

and $\mathbf{H} \in \mathbb{R}^{3\times 4}$ is 3D affine transformation matrix.

Full perspective camera fitting

In the 3D alignment case, a depth map $d(\mathbf{x})$ must be given. If the target shape is fixed, we can also define a target function E_{geo} without using a depth map $d(\mathbf{x})$ as:

$$E_{\text{geo}} = \sum_{x=1}^{N_w} \sum_{y=1}^{N_H} c(\mathbf{x}) \left\| \mathbf{x} P_2(\mathbf{x}, \mathbf{H_f}) - \begin{bmatrix} P_0(\mathbf{x}, \mathbf{H_f}) \\ P_1(\mathbf{x}, \mathbf{H_f}) \end{bmatrix} \right\|_2^2, \tag{3.39}$$

where

$$\begin{bmatrix} P_0(\mathbf{x}, \mathbf{H_f}) \\ P_1(\mathbf{x}, \mathbf{H_f}) \\ P_2(\mathbf{x}, \mathbf{H_f}) \end{bmatrix} = \mathbf{H}_f \begin{bmatrix} \bar{\mathbf{s}}(u(\mathbf{x}), v(\mathbf{x})) \\ 1 \end{bmatrix}, \tag{3.40}$$

and $\mathbf{H}_f \in \mathbb{R}^{3\times 4}$ is a perspective projection matrix.

Weak perspective camera fitting

Lastly, we show the model based on weak perspective camera model. The target function E_{geo} is defined as:

$$E_{\text{geo}} = \sum_{x=1}^{N_w} \sum_{y=1}^{N_H} c(\mathbf{x}) \left\| \mathbf{x} - \mathbf{P}(\mathbf{x}, \mathbf{H}_w) \right\|_2^2, \tag{3.41}$$

where

$$\mathbf{P}(\mathbf{x}, \mathbf{H}_w) = \mathbf{H}_w \begin{bmatrix} \bar{\mathbf{s}}(\mathbf{x}) \\ 1 \end{bmatrix}, \tag{3.42}$$

and $\mathbf{H}_w \in \mathbb{R}^{2\times 4}$ is a weak perspective projection matrix.

3.4 Integration of geometric and photometric least-squares

In Section 3.2, we assumed that geometric information is given and appropriate normal vectors are provided for points where photometric least-squares is applied. Normal vectors can be provided through geometric least-squares. Here, we show the integration

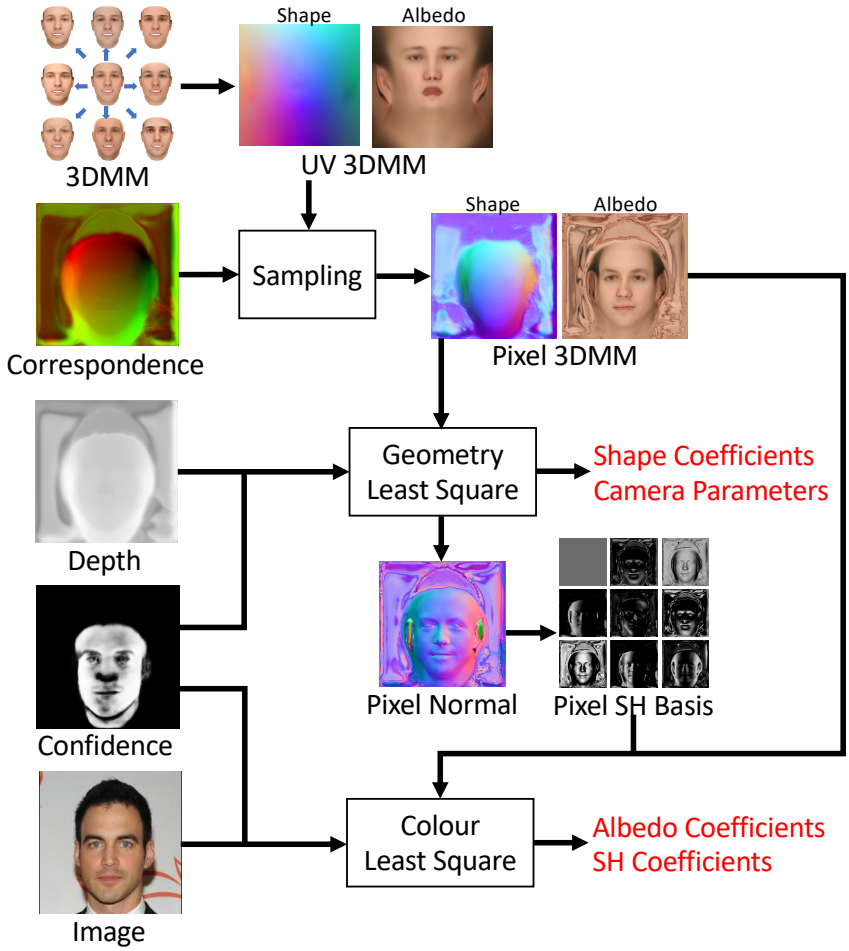
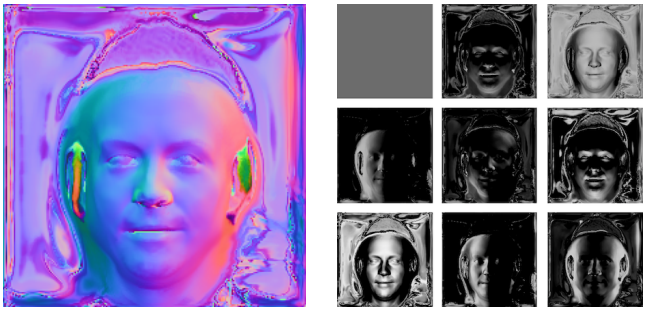


Figure 3.5: Overview of integration of geometric and photometric least-squares layer (outputs in red).

of geometric and photometric least-squares layers by Figure 3.5. From the geometry estimated by the geometric least-squares layer, we can obtain a surface normal map in UV space. A surface normal vector $\mathbf{n}_{\boldsymbol{\alpha}}(u, v)$ at UV position (u, v) in UV space is given by:

$$\mathbf{n}_{\boldsymbol{\alpha}}(u, v) = \frac{(\mathbf{v}_{\boldsymbol{\alpha}}(u, v+1) - \mathbf{v}_{\boldsymbol{\alpha}}(u, v)) \times (\mathbf{v}_{\boldsymbol{\alpha}}(u+1, v) - \mathbf{v}_{\boldsymbol{\alpha}}(u, v))}{|(\mathbf{v}_{\boldsymbol{\alpha}}(u, v+1) - \mathbf{v}_{\boldsymbol{\alpha}}(u, v)) \times (\mathbf{v}_{\boldsymbol{\alpha}}(u+1, v) - \mathbf{v}_{\boldsymbol{\alpha}}(u, v))|}, \tag{3.43}$$



$\mathbf{n}_{\boldsymbol{\alpha}}(u(x,y),v(x,y))$ $\mathbf{f}_i(\mathbf{n}_{\boldsymbol{\alpha}}(u(x,y),v(x,y)))$

(a) (b)

Figure 3.6: From shape parameters $\boldsymbol{\alpha}$ we calculate surface normals in UV space and interpolate via $u(x,y)$ and $v(x,y)$ to a pixel space normal map (a). From this we define an SH basis in pixel space (b).

where $\mathbf{v}_{\boldsymbol{\alpha}}(u,v)$ is provided by Equation (3.12). Given the estimated image-to-model correspondence map, we can interpolate a pixel space normal map $\mathbf{n}_{\boldsymbol{\alpha}}(u(x,y),v(x,y))$ (see Figure 3.6(a)). Now, we consider photometric linear least-squares (Equation (3.5)) for pixels instead of vertices. We modify Equation (3.4) to apply it to pixels, and the inverse shaded pixel value $\hat{\mathbf{r}}(\mathbf{x})$ at $\mathbf{x}=(x,y)$ in image space is given by:

$$\hat{\mathbf{r}}(\mathbf{x}) = \mathbf{i}(\mathbf{x}) \cdot \sum_{i=1}^{27} \gamma_i \, \mathbf{f}_i(\mathbf{n}_{\boldsymbol{\alpha}}(u(\mathbf{x}),v(\mathbf{x}))), \tag{3.44}$$

where $\mathbf{i}(\mathbf{x})$ is the pixel value at (x,y) position of the input image. We can write the objective function for the linear least-squares problem in pixel space as:

$$\begin{aligned} E_{\text{photo}} = \sum_{x=1}^{N_w}\sum_{y=1}^{N_H} c(\mathbf{x}) \|\hat{\mathbf{r}}(\mathbf{x}) - \mathbf{r}_{\boldsymbol{\alpha}}(u(\mathbf{x}),v(\mathbf{x}))\|_2^2 \\ + \sum_{i=1}^{27} \eta_i \gamma_i^2 + \sum_{i=1}^{N_r} \omega_i \beta_i^2 \end{aligned} \tag{3.45}$$

where $\mathbf{r}_{\boldsymbol{\alpha}}(u(\mathbf{x}),v(\mathbf{x}))$ is given by Equation (3.13) via bilinear sampling of the UV-3DMM. The solution for this system can be obtained by the calculation in Section 3.2.2. We will show the example of self-supervised training based on this integrated archi-

tecture in Section 5.2.

3.5 Conclusion

In this chapter, we have presented ways to integrate a linear least-squares layer that fits parameters related to pose, lighting, and/or 3DMM coefficients as a fixed differentiable component in a pipeline of neural networks.

Firstly, we propose a linear least-squares layer for photometric elements. We show the layer can be implemented in a differentiable way using a pseudoinverse matrix. To achieve this, we devise inverse spherical harmonics lighting to linearise the problem and the idea is validated through experiments to reproduce appearance of randomly generated 3DMM samples. This layer assures the optimal fit to given appearance under 3DMM and the lighting model. Hence, this improves the fidelity of the final estimation.

Secondly, we propose a linear least-squares layer for geometric elements. We show variations of different camera models and simultaneous fitting of 3DMM coefficients and solution by a pseudoinverse matrix, which is differentiable. For fitting of full perspective camera parameters with 3DMM coefficients, obtained camera parameters can be decomposed into intrinsic and extrinsic camera parameters. This method is applicable to a dense image-to-model correspondence map and can be used to provide constraints to the estimated correspondence.

Lastly, we propose an integrated pipeline of photometric and geometric linear least-squares layers. This method will be used for neural network training in Chapter 5 and the advantage will be demonstrated.

From the perspective of nonlinear optimisation involving linear variables, both photometric and geometric linear least-squares layers are viewed as the extension of the variable projection approach [24] and Wiberg matrix factorisation [75] to neural network training. In both approaches, linear variables are solved with respect to the objective function and treated as dependent variables of nonlinear variables. In this context, we set a sub-target function as either a linear photometric or geometric problem and optimise network parameters as nonlinear variables by minimising the entire

loss function. From the perspective of a neural network that includes constraints in implicit form, our approach linearises the constraints by modifying the formulation and directly solves them. Alternatively, we can retain the constraints in the original formulation and implement them as an implicit layer. In this approach, we can solve the constraints using any arbitrary solver and calculate the gradient through the implicit function theorem.

Chapter 4

Differentiable cohesive measure

4.1 Introduction

A segmentation map is image-like data, in which semantic labels are assigned to each pixel. In the case of a human face, each label represents a semantically distinguishable region of a face, such as eyes, ears, nose, and lips. Similarly, semantic labels can be defined on meshes of a 3D face model. Each vertex of the 3DMM corresponds to the same semantic point, thanks to the dense correspondence among samples with different identity and expression, which was established when the model was constructed. Therefore, semantic segmentation on the 3DMM remains fixed under identity and expression variations. This provides a rationale for using semantic segmentation labels as information about dense correspondence between an image and a 3D model. When considered as information for fitting, landmarks represent one-to-one sparse information and require exact annotation, whereas segmentation labels represent many-to-many dense information and can handle regions that cannot be defined by points inherently. To utilise a segmentation map as a constraint for fitting a 3D model, we need a measure that represents consistency between a 2D segmentation map corresponding to a target image and the projection of semantic labels on a 3D model. We refer to this measure as a cohesive measure. To fit a model by gradient-based parameter optimisation or a neural network, this cohesive measure must be differentiable.

In this chapter, we propose differentiable cohesive measures for two different use-

cases. One use-case is optimisation of image-to-model correspondence based on a differentiable cohesive measure on the surface of the 3D model. For this use-case, we propose to utilise bilinear sampling of a precomputed distance map. In this approach, the minimisation of the cohesive measure pushes the target points into the corresponding regions. This is particularly effective when not all points on the model are visible due to occlusion. The details of the distance-transform-based cohesive measure will be described in Section 4.3. The other use-case is optimisation of pose and shape parameters of the 3DMM based on a differentiable cohesive measure on image space. For this use-case, we propose to apply the geometric Rényi divergence (GRD) to the alignment of segmentation labels on the vertices of the 3DMM, projected onto the image plane, with a given 2D semantic segmentation map. This approach is intended to achieve exact matching between target points and regions. Therefore, this is suitable for a case that all the points on the model are visible or occlusion is negligible. The details of the GRD-based cohesive measure will be described in Section 4.4.2. For semantic segmentation supervision, cross entropy is widely utilised. However, it delivers supervisory signals only based on local pixel value differences, and its capability to effectively push target points into the correct regions is limited. The key of the proposed cohesive measures is its ability to extend gradient signals to spatially distant points.

Both approaches rely on predefined semantic segmentation labels on the 3DMM template vertices. In the remainder of this section, we will describe how to construct semantic segmentation labels on the 3DMM vertices.

4.2 Automatic labelling of model vertices

In order to utilise semantic segmentation labels for image-to-model alignment, semantic labels for each vertex in the 3D face model that are consistent with given semantic labels on target images are required. We annotate semantic segmentation labels on the vertices by transferring the labels on the images to the model automatically via the following process.

First, we pre-train an image-to-image face parsing network using the given labelled

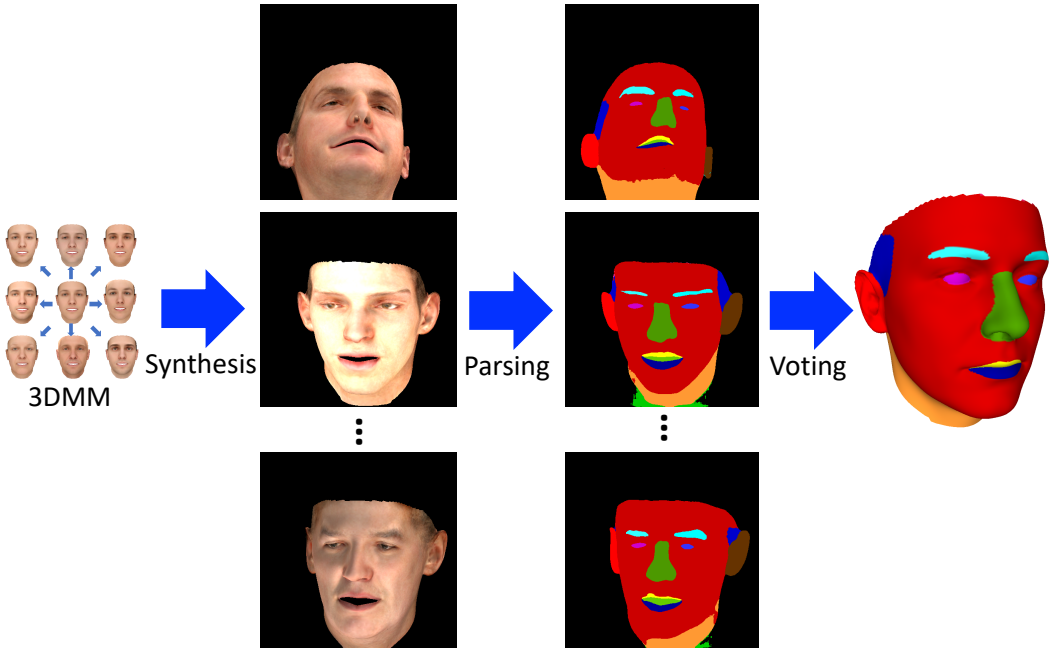


Figure 4.1: Automatic semantic labelling of model vertices.

image dataset. Specifically we use CelebAMask-HQ [40]. Next, we randomly generate 3D face meshes sampling from the 3DMM. Subsequently, we render images and feed them into the face parsing network. Based on output semantic segmentation labels from the network, we assign the most probable label to each visible vertex as one vote. Then, we count the number of votes for each vertex and each label. We take the most assigned label for each vertex as a semantic segmentation label on the vertex. We note that human annotators may not be entirely consistent in how they segment face regions (e.g. how they delineate the boundary of the nose region). Our automatic labelling seeks to be optimal in aggregate across the training set. We show a visual overview of this process in Figure 4.1.

4.3 Cohesive measure based on distance transform

In this section, we propose a distance-transform-based soft cohesive measure, which can be used to encourage a group of points to align with a region in an image via

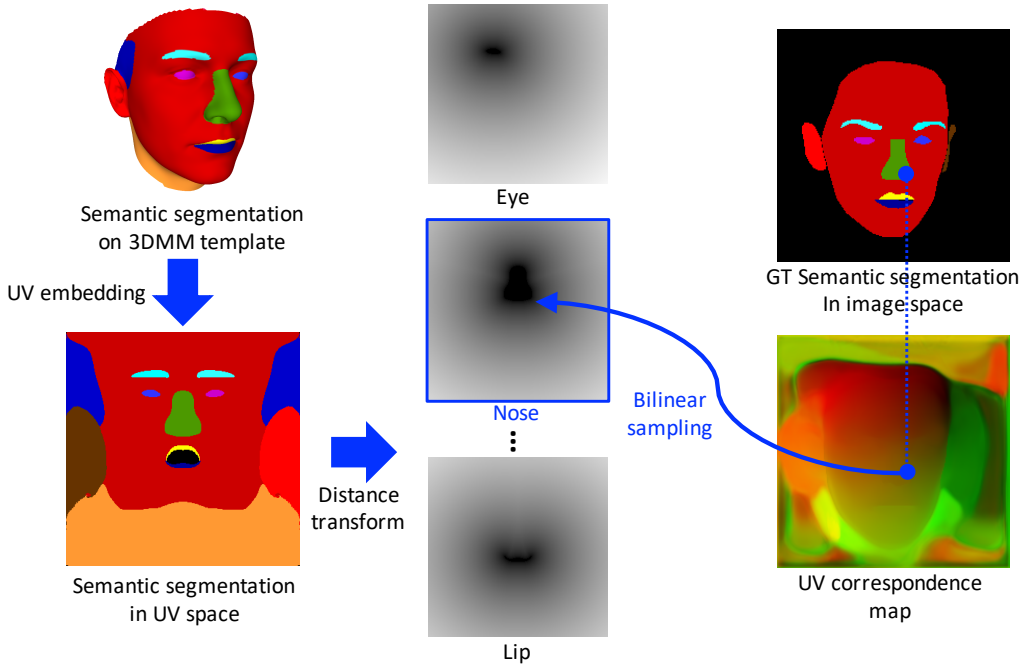


Figure 4.2: Distance-transform-based supervision. A fixed semantic segmentation of the model (top left) is mapped to UV space (bottom left) from which a distance transform for each segment is computed (middle). The estimated correspondence map (bottom right) provides a UV coordinate for each pixel. The corresponding ground truth semantic label for that pixel (top right) is used to select the appropriate distance transform map and the distance at the predicted UV coordinate bilinearly interpolated to provide the loss.

minimisation. Intuitively, a semantic pixel label restricts the possible image-to-model correspondence to only the region with the correct semantic class on the model. If the target point is inside the corresponding region, where both point and region have the same segmentation class, the distance between the target point and the nearest point in the region is zero. On the other hand, if the target point is distant from the nearest point in the region, the distance shows a larger value. Minimising such a measure provides supervisory signals for establishing dense pixel-to-model correspondence. To improve computational efficiency and to achieve differentiability, we precompute a distance map for each semantic class and bilinearly sample it to calculate the distance. Figure 4.2 shows the overview of this scheme. First, we construct a semantic segmen-

tation map in UV space (Section 4.3.1). Subsequently, distance maps for respective semantic classes are generated (in Section 4.3.2). Finally, segmentation maps corresponding to respective target points are bilinearly sampled and aggregated to calculate the cohesive measure, which can be used as a loss function for neural network training or gradient-based optimisation.

4.3.1 Semantic segmentation of 3DMM in UV space

To calculate a cohesive measure by sampling a distance map associated with the surface of 3DMM, we precompute a segmentation map in UV space in the same manner as we generate UV-3DMM in Section 3.3.1. Now, we assume l_{ij} is a binary flag for the ith vertex in the 3DMM and the jth segmentation label, where $[l_{i1}, l_{i2} \cdots, l_{iN_{\text{seg}}}]$ is a one-hot vector, and N_{seg} represents the number of types of semantic classes. If the $\hat{j}$th label is assigned to the ith vertex, the binary flag l_{ij} satisfies:

$$l_{ij} = \begin{cases} 1 & \text{if } j = \hat{j} \\ 0 & \text{otherwise} \end{cases} . \tag{4.1}$$

We render a segmentation map in UV space as an N_{seg}-channel image $\mathbf{L} = [L_0, L_1, \cdots, L_{N_{seg}}]$ from N_{seg}-dimensional flags on vertices via barycentric interpolation. To apply distance transform, we binarise $\mathbf{L}$ and obtain $\hat{\mathbf{L}}$ as:

$$\hat{L}_j(u,v) = \begin{cases} 1 & \text{if } \forall i \in \{1, 2, \cdots, N_{\text{seg}} | i \neq j\}, L_j(u,v) > L_i(u,v) \ \wedge \ L_j(u,v) > \epsilon \\ 0 & \text{otherwise} \end{cases} . \tag{4.2}$$

In our experiment, we choose $\epsilon = 0.01$. By this process, we obtain a segmentation map in UV space (UV-segmentation map). This labelling is done once and remains fixed during experiments. An example of the generated semantic segmentation map in UV space is show in (middle left) in Figure 4.2.

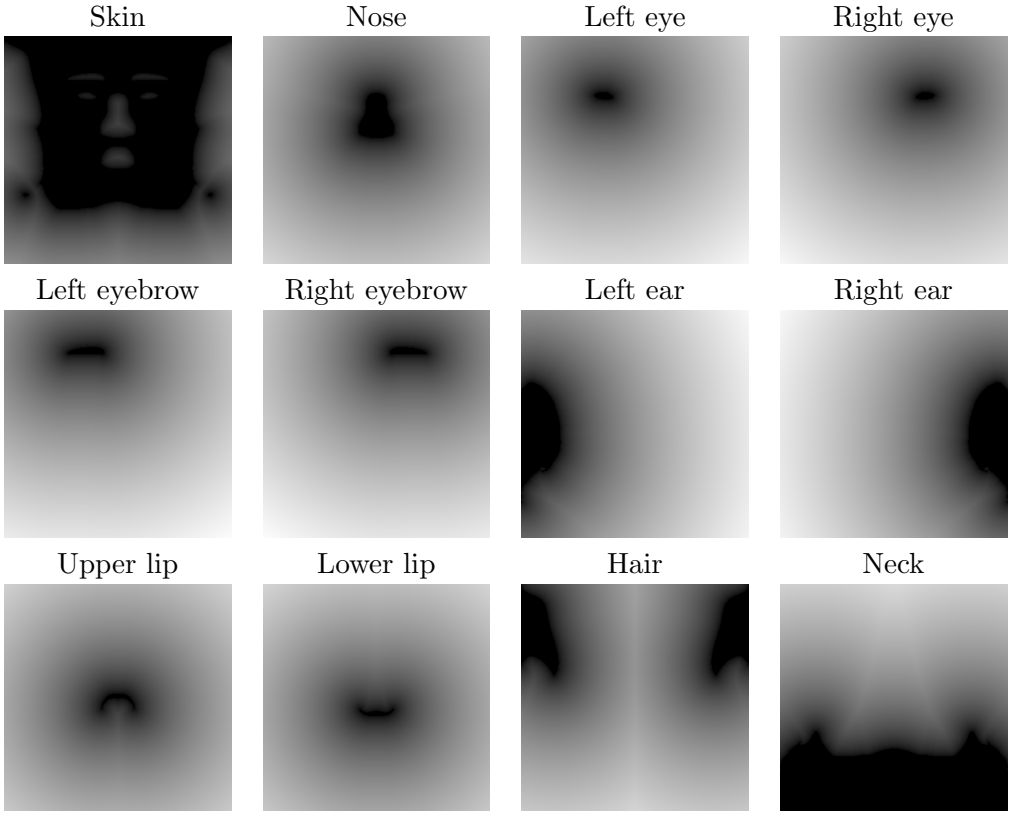


Figure 4.3: Visualisation of Euclidean distance maps of facial segmentation labels in UV space.

4.3.2 Distance transform of a segmentation map

Once the UV-segmentation map is obtained, we apply the Euclidean distance transform [14] to each label. A value of a distance map on (u,v) position for the jth segmentation label satisfies:

$$\hat{D}_j(u,v) = \{(u-\hat{u})^2 + (v-\hat{v})^2 | (\hat{u},\hat{v}) \in [1,2,\cdots,N_H] \times [1,2,\cdots,N_W], L_j(\hat{u},\hat{v}) = 1\} \tag{4.3}$$

$$D_j(u,v) = \begin{cases} \min(\hat{D}_j(u,v)) & \text{if} \quad \hat{D}_j(u,v) \neq \emptyset \\ D_{inf} & \text{otherwise} \end{cases}, \tag{4.4}$$

where D_{inf} represents a constant value corresponding to the infinite distance. We show examples of the Euclidean distance map for each label in UV space in Figure 4.3. Similar to the UV-segmentation map, the distance map in UV space is precomputed and remains fixed during experiments.

4.3.3 Calculation of a cohesive measure based on bilinear sampling of a distance map

Using the precomputed distance maps, we calculate a differentiable cohesive measure by aggregating distance values among target points. We calculate the cohesive measure E_{edt} as:

$$E_{\text{edt}} = \sum_{j=1}^{N_{\text{seg}}} w_j \left[\sum_{i=1}^{N_p} l'_{ij} D_j(u_i, v_i) \right], \tag{4.5}$$

where l'_{ij} represents a binary label for the ith point and the jth label class, N_p represents the number of points, $D_j(u_i, v_i)$ represents a distance map on (u_j, v_j) position in UV space for the jth label class, and w_j represents a weight for jth label class. (u_j, v_j) can have fractional values and $D_j(u_i, v_i)$ is bilinear interpolation of the UV-segmentation map D_j at (u_i, v_i). Through bilinear interpolation, E_{edt} has differentiability and can provide a meaningful supervisory signal for network training and gradient-based optimisation. In this method, target points can be either pixels or projected vertices. In the case of pixels, the target points can be image-to-model correspondences predicted by a neural network. In the case of vertices, the target points can be projected vertices of 3DMM, which is described in Section 3.1.1, and we can define E_{edt} by using a distance map $D_j(u_i, v_i)$ generated from a segmentation map in image space instead of that in UV space. The minimisation of the distance-transform-based cohesive measure is underconstrained and tends to push all the points into a small region. In Section 5.4, we will show the network training that uses pixel-based loss, projected-vertex-based loss, and model-based regularisation altogether to introduce reasonable constraints.

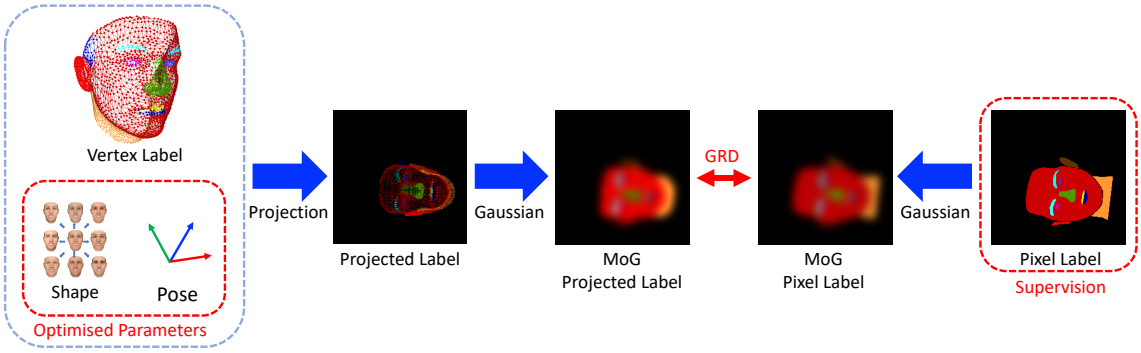


Figure 4.4: To extract a supervisory signal from a given pixel-wise semantic segmentation, we propose a loss that is differentiable with respect to pose and shape parameters. Given fixed per-vertex semantic labels and pose and shape estimates (col. 1), we project the labelled vertices to 2D. We represent both these vertex projections (col. 2) and the given pixel-wise labels (col. 5) as mixtures of Gaussians (col. 3-4) and measure segmentation loss using the geometric Rényi divergence.

4.4 Cohesive measure based on the geometric Rényi divergence

In this section, we show a novel cohesive measure based on the geometric Rényi divergence, which can be used to align segmentation labels. We assume that an entity to be aligned, referred as segmentation labels, is either an image-like segmentation map or 3D meshes that have segmentation labels and are projected onto the image plane. A pixel-wise semantic segmentation is a discrete representation. Similarly, the rasterisation of 3D meshes into an image (and the corresponding pixel-wise semantic segmentation) is also discrete. This means that pixel-based measures for comparing the similarity of the two semantic segmentation maps (such as intersection over union) are discontinuous. Therefore, the gradient of such measures provides no information about how to adjust the parameters of the 3D model to achieve a similar semantic segmentation to the given pixel-wise one.

For this reason, we propose a soft, probabilistic measure for comparing pixel-wise and vertex-wise semantic segmentations in 2D. We illustrate our proposed method using an example of 3DMM fitting in Figure 4.4. Given estimates of 3DMM shape

parameters and the pose (camera parameters), we project the 3D vertices of the 3DMM to 2D (see Section 3.1.1). The vertices themselves have fixed semantic labels (see Section 4.2). We assume that we are given a target pixel-wise semantic segmentation (i.e. in the context of CNN training, we assume a supervised scenario). These input labels could themselves be predicted by a 2D semantic segmentation network. Then, we represent both the projected vertices and the pixel labels probabilistically as a mixture of Gaussians. Our key idea is to measure the difference between these two distributions using the geometric Rényi divergence. This new measure has advantages that: 1) it varies smoothly with respect to the displacement of the projection; 2) optimal alignment corresponds to the minimum value; 3) the gradient does not vanish even if the displacement is large. Hence, this method can enable accurate and stable 2D-3D alignment of the model. We validate our method through experiments on direct optimisation of the loss given a single input segmentation, i.e. shape-from-semantic segmentation (see Section 4.5.2) and parameter regression CNN training (see Section 5.3).

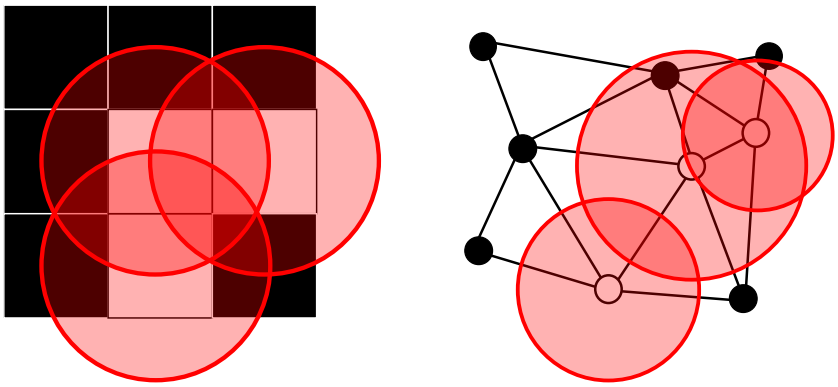
We begin by showing how to compute a semantic segmentation loss between pixels and projected vertices of a given semantic class.

4.4.1 Pixel and vertex labels as mixtures of Gaussians

In order to obtain long-range gradients from the discrepancy between semantic labels on input images and projected vertices, we soften both labels by analytically convolving Gaussian kernels on representative points (see Figure 4.5). Hence, we represent softened semantic label P on image coordinate $\mathbf{z}$ with a mixture of Gaussians:

$$\begin{aligned} P(\mathbf{z}) &= \sum_{i=1}^{N} \frac{\alpha_i}{2\pi\sigma^2} \exp\left(-\frac{1}{2\sigma^2}\left(\mathbf{z}-\mathbf{x}_i\right)^T\left(\mathbf{z}-\mathbf{x}_i\right)\right) \\ &= \sum_{i=1}^{N} \alpha_i G\left(\mathbf{z}-\mathbf{x}_i, \sigma^2\mathbf{I}\right), \end{aligned} \tag{4.6}$$

where $\mathbf{x_i}$ is the centre of the ith Gaussian kernel (corresponding to either a pixel centre or projected vertex position), and σ is the corresponding standard deviation



(a) Soft label on pixels (b) Soft label on vertices

Figure 4.5: Representing pixels (a) and vertices (b) of a given semantic class (shown in white) as mixtures of Gaussians. The size of each circle represents the weight of its corresponding Gaussian kernel, which is proportional to the area of the relevant pixel or neighboring triangles.

of the Gaussian function. α_i is the weight of the ith Gaussian kernel, which satisfies $\alpha_i > 0$ and $\sum_{i=1}^{N} \alpha_i = 1$, and is allocated based on the corresponding area on the image. For input pixel-wise semantic labels, α_i is set to $1/N$ so that it represents the normalised area of one pixel. For vertices, α_i is set to the average of the projected area of the neighboring faces normalised by the total area of the projected faces.

4.4.2 Geometric Rényi divergence

We employ a closed-form geometric Rényi divergence (GRD) as a cohesive measure between two mixtures of Gaussian (MoG) distributions, which represent pixel-wise and projected semantic labels. Wang et al. [74] proposed the closed-form Jensen-Rényi divergence (JRD) for MoG and applications to group-wise shape registration. Assuming we calculate JRD among K distributions, which are two (K=2) MoGs in our case, JRD is defined as:

$$JRD_{\pi,q}\left(P_1, P_2, \ldots, P_K\right) = H_q\left(\sum_{i=1}^{K} \pi_i P_i\right) - \sum_{i=1}^{K} \pi_i H_q\left(P_i\right), \tag{4.7}$$

where H_q is the qth-order Rényi entropy, and $\pi = \{\pi_1, \pi_2, \ldots, \pi_n | \pi_i > 0, \sum_i \pi_i = 1\}$ are the weights for the weighted arithmetic mean of the distributions and the entropies. The qth-order Rényi entropy is defined as:

$$H_q(P) = \frac{1}{1-q} \log \left(\int P(\mathbf{z})^q \, d\mathbf{z} \right). \tag{4.8}$$

When $q \to 1$, (4.8) is the Shannon entropy, and (4.7) is the Jensen-Shannon divergence. Wang et al. [74] employed $q = 2$ as it has a closed-form for MoG. However, non-negativity of JRD is not guaranteed when $q > 1$, and optimal registration does not necessarily correspond to minimal divergence. Therefore, the second-order JRD is not a preferable measure for alignment of two distributions. To resolve the negativity issue, Antolín et al. [3] proposed geometric Rényi divergence (GRD):

$$GRD_{\pi,q}(P_1, P_2, \ldots, P_K) = (q-1) \left[H_q \left(\prod_{i=1}^{K} P_i^{\pi_i} \right) - \sum_{i=1}^{K} \pi_i H_q(P_i) \right]. \tag{4.9}$$

For arbitrary q, non-negativity of GRD is guaranteed. In addition, when $q = 2$, a closed-form GRD can be derived for comparison of two distributions in the same way as JRD.

4.4.3 Closed-form second-order GRD between two MoGs

We now derive a closed-form second-order GRD between two MoGs, i.e. for the special case $\pi = \frac{1}{2}$, $q = 2$:

$$GRD_{1/2,2}(P_x, P_y) = H_2 \left(\sqrt{P_x P_y} \right) - \frac{1}{2} \left(H_2(P_x) + H_2(P_y) \right). \tag{4.10}$$

Based on the closed-form integral of the product of two Gaussians, we obtain:

$$
\begin{aligned}
H_2\left(\sqrt{P_x P_y}\right) &= \int P_x(\mathbf{z}) P_y(\mathbf{z}) d\mathbf{z} \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j \int G(\mathbf{z}-\mathbf{x}_i, \sigma^2\mathbf{I}) G(\mathbf{z}-\mathbf{y}_j, \sigma^2\mathbf{I}) d\mathbf{z}\right] \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j G(\mathbf{x}_i-\mathbf{y}_j, 2\sigma^2\mathbf{I})\right], \qquad (4.11)
\end{aligned}
$$

and

$$
\begin{aligned}
H_2\left(P_x\right) &= \int P_x(\mathbf{z})^2 d\mathbf{z} \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j \int G(\mathbf{z}-\mathbf{x}_i, \sigma^2\mathbf{I}) G(\mathbf{z}-\mathbf{x}_j, \sigma^2\mathbf{I}) d\mathbf{z}\right] \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j G(\mathbf{x}_i-\mathbf{x}_j, 2\sigma^2\mathbf{I})\right]. \qquad (4.12)
\end{aligned}
$$

From Equation (4.10), Equation (4.11) and Equation (4.12), we obtain the closed-form divergence:

$$
\begin{aligned}
GRD_{1/2,2}\left(P_x, P_y\right) = &-\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j G(\mathbf{x}_i-\mathbf{y}_j, 2\sigma^2\mathbf{I})\right] \\
&+ \frac{1}{2}\log\left[\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j G(\mathbf{x}_i-\mathbf{x}_j, 2\sigma^2\mathbf{I})\right] \\
&+ \frac{1}{2}\log\left[\sum_{i=1}^{N}\sum_{j=1}^{N}\beta_i\beta_j G(\mathbf{y}_i-\mathbf{y}_j, 2\sigma^2\mathbf{I})\right]. \qquad (4.13)
\end{aligned}
$$

4.4.4 Numerical stability

The GRD becomes numerically unstable when the difference between two MoG distributions is large. That is because all the exponential functions in Equation (4.11) output zero value for large $\|\mathbf{x_i} - \mathbf{y_i}\|_2^2$. To avoid this issue, in practice, we implement

Equation (4.11) as:

$$H_2\left(\sqrt{P_x P_y}\right) = -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\exp\left(\mathbf{e}_{ij} - \max\{\mathbf{e}_{ij}\}\right)\right] - \max\{\mathbf{e}_{ij}\} - \log\left(\frac{1}{2\pi\sigma^2}\right), \tag{4.14}$$

where $\mathbf{e}_{ij} = -\frac{\left(\mathbf{x_i}-\mathbf{y_j}\right)^T\left(\mathbf{x_i}-\mathbf{y_j}\right)}{2\sigma^2} + \log(\alpha_i\beta_j)$.

4.4.5 Analysis-by-synthesis

We now show how to integrate our GRD-based semantic segmentation loss into analysis-by-synthesis. We use the GRD for MoG to optimise shape and pose parameters so that the discrepancy of given semantic labels on vertices and pixels is minimised. We directly minimise parameters in an analysis-by-synthesis manner as shown in Figure 4.4.

We place a Gaussian kernel on each projected vertex $\acute{\mathbf{x}}_j$ calculated from Equation (3.2), and obtain the softened semantic label $\acute{P}_j$ of the jth label on the image coordinate $\mathbf{z}$:

$$\acute{P}_j(\mathbf{z}) = \frac{\sum_{i=1}^{N_v} \acute{l}_{ij} w_i G\left(\mathbf{z} - \acute{\mathbf{x}}_i, \sigma^2\mathbf{I}\right)}{\sum_{i=1}^{N_v} \acute{l}_{ij} w_i}, \tag{4.15}$$

where N_v represents the number of vertices, $\acute{l}_{ij}$ represents the jth label on the ith vertex, which returns 1 if the vertex belongs to the label and 0 otherwise, w_i represents the average area of three neighboring faces of the ith vertex projected on the image plane. The area is regarded as zero if the vertex normal points away from the camera (i.e. self-occluded).

For pixel labels, we place a Gaussian kernel on each pixel with image coordinate $[u, v]$, and obtain the softened semantic label $\hat{P}_j$ of the jth label on the image coordinate $\mathbf{z}$:

$$\hat{P}(\mathbf{z}) = \frac{\sum_{v=1}^{N_H}\sum_{u=1}^{N_W} \hat{l}_j(u,v) G\left(\mathbf{z} - [u,v]^T, \sigma^2\mathbf{I}\right)}{\sum_{v=1}^{N_H}\sum_{u=1}^{N_W} \hat{l}_j(u,v)}, \tag{4.16}$$

where $\hat{l}_j(u,v)$ represents the jth label on the image coordinate $[u,v]$, N_W is the number of horizontal pixels, and N_H is the number of vertical pixels.

We calculate GRD for each label based on Equation (4.13) and minimise average

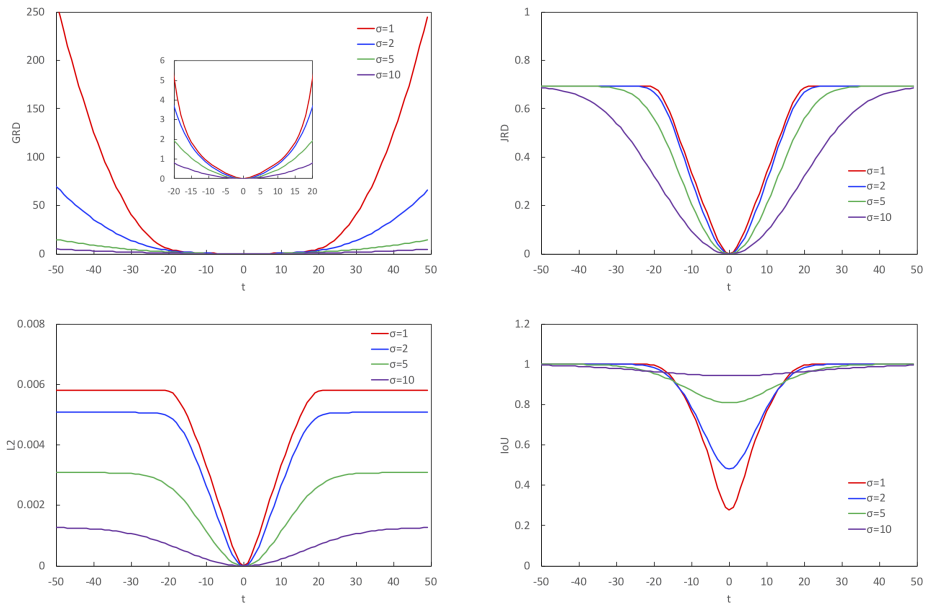


Figure 4.6: Loss landscape of GRD (top-left), JRD (top-right), L2 (bottom-left), and IoU (bottom-right) with respect to t pixel horizontal translation.

GRD of all the labels while optimising all shape and camera parameters.

4.5 Experiments

In this section, we evaluate the GRD-based cohesive measure by visualising the loss landscape and conducting optimisation in an analysis-by-synthesis scenario. Experiments of neural network training based on the distance-transform-based cohesive measure and the GRD-based cohesive measure will be shown in Chapter 5.

4.5.1 Loss landscape and comparison

We now illustrate the attractive properties of the GRD using a toy example. We draw a circle with a 10-pixel diameter onto a 100×100 pixel image. We generate two MoGs by putting Gaussian kernels on each pixel in the circle, and transform one MoG while fixing the other. In Figure 4.6, a horizontal translation of t pixels is applied, and in Figure 4.8, magnification by factor s is applied. We visualise how MoG changes by

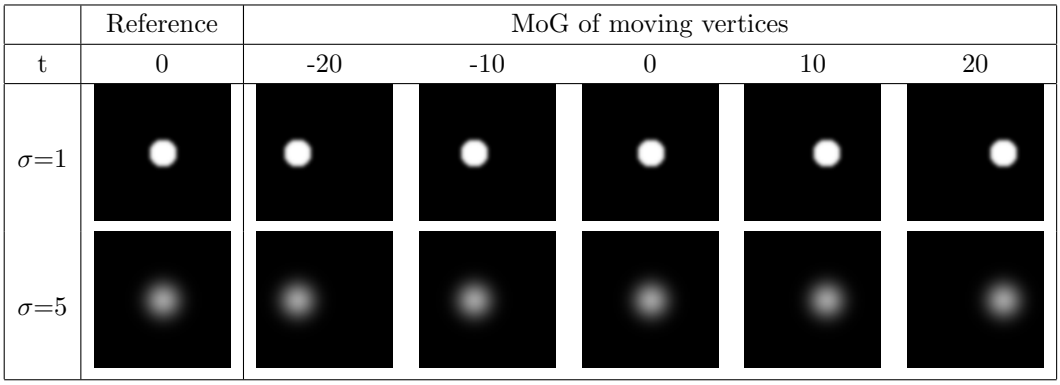


Figure 4.7: Visualisaion of the experiment for loss landscape evaluation. According to t, the MoG of vertices moves horizontally.

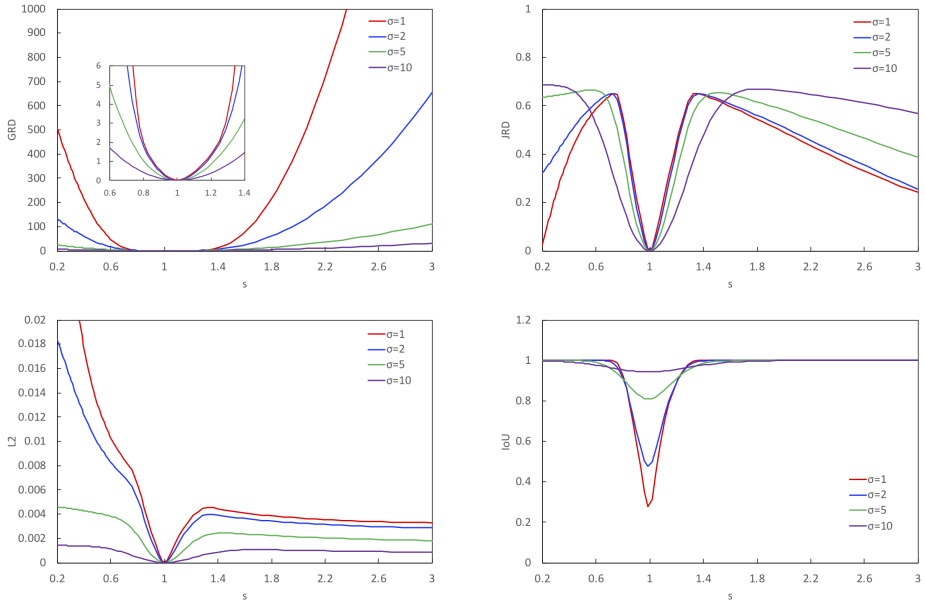


Figure 4.8: Loss landscape of GRD (top-left), JRD (top-right), L2 (bottom-left), and IoU (bottom-right) with respect to magnification by s.

translation in Figure 4.7 and how MoG changes by magnification in Figure 4.9. We compare GRD with JRD, L2 loss, and IoU loss. L2 loss L_{L2} for two distributions P_x and P_y is defined as $L_{L2} = \|P_x - P_y\|_2^2$. Following [72] and [45], we define a soft IoU

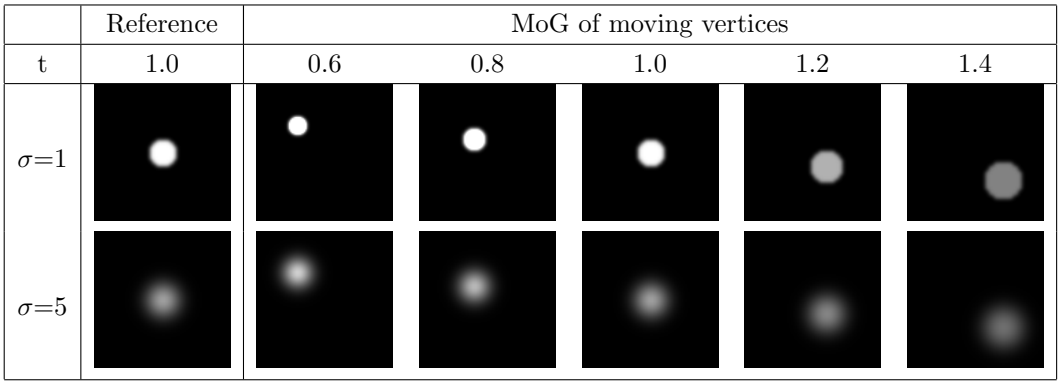


Figure 4.9: Visualisaion of the experiment for loss landscape evaluation. According to s, the MoG of vertices expands.

	GRD (Ours)	NMR [35]	SoftRas [45]
IoU mean	0.931	0.789	0.423
IoU std	0.013	0.150	0.124

Table 4.1: Direct optimisation results for semantic labels randomly synthesised from the BFM [23].

loss L_{IoU} for two distributions P_x and P_y as:

$$L_{IoU} = 1 - \frac{\|P_x \odot P_y\|_1}{\|P_x + P_y - P_x \odot P_y\|_1}. \tag{4.17}$$

In the case of translation, the gradient of JRD, L2, and IoU becomes flat when the displacement is large, whereas GRD increases quadratically. That means only GRD is suitable for large-scale alignment. In the case of scaling, JRD goes negative when the difference in scale is large, while L2 exhibits non-optimal local minima and IoU shows a flat gradient. These examples indicate that GRD is more suitable as a measure for region alignment than other metrics.

4.5.2 Analysis-by-synthesis

We apply our approach to analysis-by-synthesis and evaluate it both quantitatively and qualitatively based on synthetic data. We also compare our approach with Neural Mesh Renderer (NMR) [35] and Soft Rasterizer (SoftRas) [45].

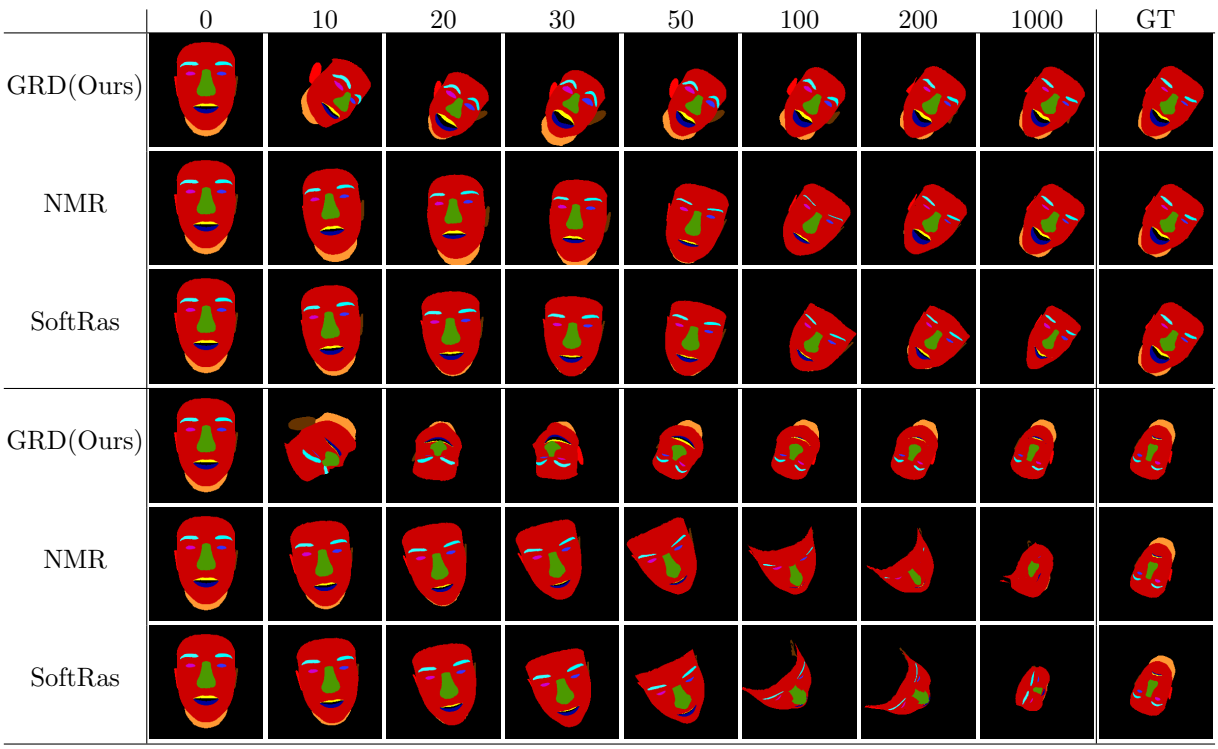


Figure 4.10: Convergence of direct optimisation of our GRD, NMR [35], and SoftRas [45] segmentation losses. Upper rows show an easy case, lower rows a challenging one. Target ground truth labels are shown in the final column.

Synthetic pixel label images are generated by perturbing 3DMM coefficients, focal length, image centre, pose rotation, and pose translation. Pose rotation is parameterised by the Euler angles. We directly optimise the 299-dimensional 3DMM coefficients, and the 9-dimensional camera parameters with respect to the average GRD between the projected MoG and the pixel MoG among 11 labels. We employ the Adam optimiser with a learning rate 0.2 for GRD, and 0.01 for NMR and SoftRas. For GRD, we choose $\sigma = 5$ as a parameter of the Gaussian kernel. In optimisation with NMR, we differentiably rasterise semantic labels as an 11-channel image, and compute the L2 norm between the rasterised image and the ground truth pixel label image. In optimisation with SoftRas, we differentiably rasterise semantic labels as an 11-channel image. We also rasterise an object silhouette and multiply it to the semantic labels. The L2 norm between the rasterised image and the ground truth pixel label

image is employed as a loss function. We choose $\sigma = 10^{-3}$ and $\gamma = 10^{-3}$ for SoftRas parameters.

Figure 4.10 shows the convergence of projected semantic labels during direct optimisation of GRD (Ours), NMR, and SoftRas losses. Upper rows show an example of a successful case, and lower rows show an example of a difficult case. In both cases, our approach converges well to the ground truth despite the large rotation from the initial pose to the ground truth. In a successful case, both NMR and SoftRas converge to the ground truth. The result of SoftRas shows slight shrinking due to the gap between original semantic label images and blurred rasterisation. In a difficult case, both NMR and SoftRas converge to a local minima. We also calculate mean and standard deviation of IoU between the ground truth and the rasterised semantic labels (Table 4.1). The result indicates our method successfully converges to the ground truth in all 16 cases, whereas NMR and SoftRas fail in some cases.

4.6 Conclusion

In this chapter, we have presented approaches to supervise image-to-model correspondence via minimisation of a cohesive measure on semantic segmentation labels.

Firstly, we propose the distance-transform-based cohesive measure, which can be used on UV space as the canonical space of the surface of the 3D statistical model, as well as on image space. We show how to generate a semantic segmentation map in UV-space and apply the distance transform to the map. By doing this as precomputation, a differentiable cohesive measure can be efficiently computed during network training or gradient-based optimisation. In this scenario, the distance-transform-based cohesive measure is calculated on pixels associated with the visible parts of the object. Therefore, it does not require occlusion calculations and is suitable for conditions where self-occlusions need to be considered. An application of this approach will be described in Section 5.4.

Secondly, we propose the cohesive measure based on the geometric Rényi divergence (GRD). This metric enables exact matching of two groups of points. We show how semantic segmentation labels on pixels and rasterised meshes can be represented

by a mixture of Gaussian (MoG) distributions. We derive a numerically stable implementation of GRD between two MoGs. Advantages of GRD are demonstrated by visualisation of the loss landscape and experiments of image-to-model alignment based on gradient-based optimisation. Alignment based on the minimisation of GRD of MoGs enables exact matching between point clouds, but it cannot explicitly handle occlusions. Thus, it is suitable for cases where occlusions can be ignored or for scenarios where the minimisation process works by computing occlusions each time and incorporating them as weights to mask invisible vertices. Further applications to neural networks will be shown in Chapter 5.

Chapter 5

Self-supervised monocular 3D face reconstruction

5.1 Introduction

In this chapter, we will describe three novel paradigms to supervise a 3D face reconstruction network while utilising and demonstrating techniques introduced in Chapter 3 and Chapter 4.

One approach is fully self-supervised training with a differentiable linear least-squares layer. This approach enables the self-supervised training of a monocular 3D face reconstruction network based on 3DMM, eliminating the need for any auxiliary supervision, such as landmarks. The goal of this study is to explore ways to reduce the need for annotations and to investigate viable network structures and training procedures. We apply differentiable linear least-squares layer, described in Chapter 3, to regularise pixel-wise prediction network, which estimates pixel-wise image-to-model correspondence, confidence, and depth. Additionally, we introduce a robust loss function on pixel-wise image-to-model correspondence that stabilises the training. The details will be shown in Section 5.2. We refer to this approach as LSDR (Least Squares with Differentiable Renderer).

Another approach is GRD-based segmentation supervision. This supervision enables training of a monocular 3D face reconstruction network based on semantic seg-

mentation maps. We aim to replace landmark supervision by a semantic segmentation map. This approach uses closed-form GRD, described in Chapter 4, to align segmentation labels on the 3D face model with a semantic segmentation map of a face image. Owing to the non-saturating gradient of GRD for large displacement, this technique proves effective as an alternative to landmark supervision. The details will be shown in Section 5.3. We refer to this approach as GRDDR (GRD based Differentiable Renderer).

The other approach is self-supervised training without a differentiable renderer using semantic segmentation. This approach applies a distance-transform-based cohesive measure in UV space, introduced in Chapter 4, and regularisation on outputs of the pixel-wise prediction network by the linear least-squares layer, introduced in Chapter 3. By using segmentation supervision defined on visible pixels, this approach removes the need for a differentiable renderer in self-supervised monocular 3D face reconstruction, yet keeping estimates plausible by regularising outputs by the linear least-squares layer. The details will be shown in Section 5.4. We refer to this approach as LSDT (Least Squares with Distance Transform).

5.2 Self-supervision without using landmarks

In this section, we show how an image-to-image network for dense face alignment can be trained using self-supervision. The idea is that the network predicts a correspondence map from which we implement the fitting process described in Chapter 3 as differentiable layers. We use a U-Net [59] as the pixel-wise prediction network though any image-to-image architecture would suffice. The network learns from losses measuring the quality of the fit to the correspondence map as well as an appearance loss computed via differentiable rendering. Thanks to this architecture, we can introduce a pixel-wise robust loss function (see Section 5.2.2), while benefiting from regularisation based on 3DMM, thereby self-supervised training of monocular 3D reconstruction without landmark annotations is realised.

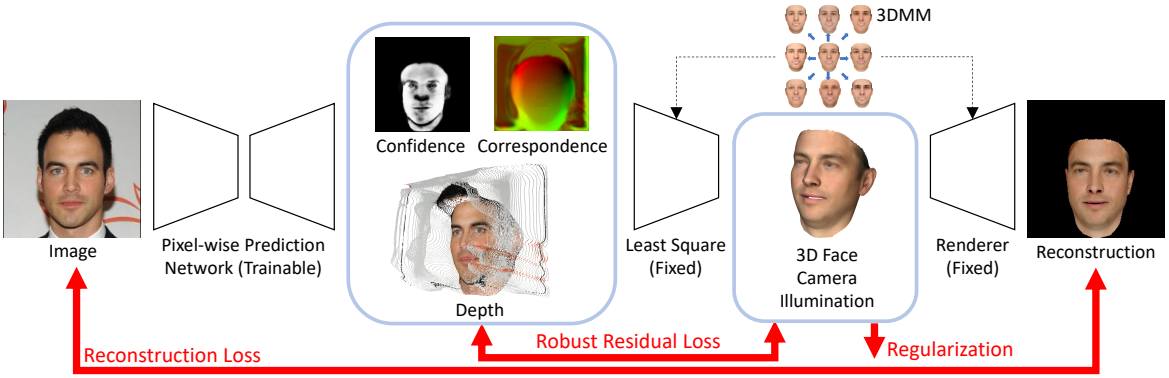


Figure 5.1: Overview of LSDR. In addition to correspondence, the network also predicts a confidence map (for robustness) and a depth map (enabling uncalibrated reconstruction). The least-squares layer solves first for geometric and then for photometric parameters.

5.2.1 Architecture

Figure 5.1 shows the architecture of the training pipeline. In this architecture, the pixel-wise prediction network infers a correspondence map, a depth map, and a confidence map. Predicted maps, pixel coordinates, and input pixel values are fed into the linear least-squares layer, described in Section 3.4, and fitted 3DMM shape and albedo coefficients, camera parameters, and lighting coefficients are calculated. Based on fitted parameters, colour and position of template mesh vertices are reconstructed. During training, the residual of linear least-squares, the regularisation terms for the fitted parameters, and the error between the reconstructed colour and input pixel value sampled from the corresponding point on the input image are minimised. The gradient signals are transmitted through the fixed differentiable layers and the network parameters of the pixel-wise prediction network are optimised. The details of the loss functions will be described in Section 5.4.

Per-pixel confidence

In general, not all of the image will contain face parts. In addition, the face may be occluded by non-face objects such as glasses or unmodelled features such as beards. We do not wish these pixels to contribute to the least-squares solutions. Therefore,

the network also predicts a scalar confidence map $w(x, y) \in [0, 1]$ indicating whether pixel (x, y) is believed to belong to the face. As with correspondence, this is learnt unsupervised without ever providing the network with ground truth face segmentation labels. Note that the prediction of a confidence map is learnt indirectly through self-supervision and, in practice, it functions more as a mask than as an exact confidence map in a statistical context.

In-network linear least-squares

We apply the differentiable linear least-squares layer to the outputs of the pixel-wise prediction network. The outputs of the network consist of a depth map $d(x, y)$, a confidence map $c(x, y)$, and a UV-correspondence map $[u(x, y), v(x, y)]$, which are defined for all $(x, y) \in \{1, \ldots, N_W\} \times \{1, \ldots, N_H\}$, where N_W and N_H are the number of horizontal and vertical pixels of input images. We apply the integrated photometric and geometric least-squares layer described in Section 3.4. On the first stage, geometric least-squares with precomputed UV-3DMM, described in Section 3.3.1, is applied to $d(x, y)$, $c(x, y)$, and $[u(x, y), v(x, y)]$ to fit 3DMM shape coefficients and camera parameters. Based on fitted geometry, a per-pixel normal map $\mathbf{n}(x, y)$ is calculated and fed into the photometric least-squares layer together with the input image $\mathbf{i}(x, y)$ to fit 3DMM albedo coefficients and inverse spherical harmonics lighting coefficients.

Stochastic sampling

Solving a linear system over all pixels for all images in a minibatch within the network during training is prohibitively computationally expensive. For this reason, we introduce a stochastic sampling of pixels for the linear least square process to reduce memory consumption. We randomly select 10,000 pixels which have confidence value larger than $0.001\times$ the maximum confidence value. If the number of pixels which fulfil the above criteria is less than 10,000, we select the rest of the pixels randomly.

5.2.2 Losses

Here, we describe the details of the loss functions to train the pixel-wise prediction network. We employ four losses:

$$E_{\text{total}} = \eta_{\text{res}} E_{\text{res}} + \eta_{\text{rec}} E_{\text{rec}} + \eta_{\text{stat}} E_{\text{stat}} + \eta_{\text{int}} E_{\text{int}}, \tag{5.1}$$

where $\eta_{\text{rec}} = 1.0$, $\eta_{\text{res}} = 3.0$, $\eta_{\text{stat}} = 1.0$, and $\eta_{\text{int}} = 1.0$. We now describe each of these four losses.

Least-squares residual loss

The least-squares layer in the LSDR network solves for optimal shape, albedo, camera, and lighting parameters by minimising the geometric and photometric residuals. The network can learn from these residuals since they indicate how consistent the 3DMM fit is with the estimated correspondence map (and depth/confidence maps) and the image. Whereas the least-squares layer required a closed-form solution and therefore uses linear least-squares, the loss used for network training is not so constrained. For this reason, we use a robust loss on the residuals:

$$E_{\text{res}} = \sum_{x,y} \min\left(\varepsilon(x,y), 1\right), \quad \text{where } \varepsilon(x,y) = \eta_{\text{geo}}\varepsilon_{\text{geo}}(x,y) + \eta_{\text{photo}}\varepsilon_{\text{photo}}(x,y), \tag{5.2}$$

where

$$\varepsilon_{\text{photo}}(x,y) = \left\| \mathbf{i}(x,y) \odot \mathbf{f}(\mathbf{n}_{\boldsymbol{\acute{\alpha}}}(u(x,y), v(x,y))) \boldsymbol{\acute{\gamma}} - \mathbf{r}_{\boldsymbol{\acute{\beta}}}(u(x,y), v(x,y)) \right\|_2, \tag{5.3}$$

$$\varepsilon_{\text{geo}}(x,y) = \left\| \mathbf{H}[d(x,y)x, d(x,y)y, d(x,y), 1]^T - \mathbf{v}_{\boldsymbol{\acute{\alpha}}}(u(x,y), v(x,y)) \right\|_2, \tag{5.4}$$

and $\eta_{\text{geo}} = 20$ and $\eta_{\text{photo}} = 5$. The variables in Equation (5.3) and Equation (5.4) have the same notation as those in Chapter 3, and they are calculated by the method in Section 3.4. This loss has an important effect: it encourages the model to expand so that more pixels in the input image can be explained by the model in both geometry and colour. For example, suppose that the pixel-wise network detects an ear with high confidence and estimates good correspondence to the ear region in the model.

If the ear of the least-squares 3DMM fit is not close to the detected ear pixels, this incurs a residual loss, encouraging the model to expand towards the ear. However, we must make the loss robust since every pixel in the image contributes to it, even the background (we do not use the confidence map here). The clamping suppresses the effect from outlier pixels such as occlusion and background.

Reconstruction loss based on differentiable rendering

We also compute a conventional reconstruction loss using differentiable rendering to compare the fitted model to the image. Without this, the clamped residual loss does not penalise the growing of the face to fit to the background. We render the 3DMM geometry given by the geometric least-squares solution. The differentiable renderer calculates a projection of each vertex as a 2D point on the image as well as its visibility and RGB albedo. We apply our inverse lighting model to the sampled intensities and measure the discrepancy to the RGB albedo per-vertex:

$$E_{\text{rec}} = \frac{1}{\sum_{j=1}^{N_v} w_j} \sum_{j=1}^{N_v} w_j \left\| \mathbf{i}(x_j, y_j) \odot \mathbf{f}(\mathbf{n}_j(\acute{\boldsymbol{\alpha}}))\acute{\boldsymbol{\gamma}} - \mathbf{r}_j(\acute{\boldsymbol{\beta}}) \right\|_2, \tag{5.5}$$

where N_v is the number of the vertices and $w_j = 1$ if a vertex is visible, and zero otherwise (computed using self-occlusion testing and depth testing against a z-buffer). We use differentiable bilinear sampling, and $\mathbf{i}(x_j, y_j)$ represents bilinear sampling of the input image at the non-integer pixel position (x_j, y_j) given by projection of vertex $\mathbf{v}_j(\acute{\boldsymbol{\alpha}})$ using the estimated camera parameters. The variables in Equation (5.5) have the same notation as those in Chapter 3, and they are calculated by the method in Section 3.4.

Statistical regularisation loss

The statistical regularisation loss encourages the network to keep the estimated face plausible in terms of the shape and albedo parameters. It is the weighted squared

average of the estimated 3DMM coefficients $\boldsymbol{\acute{\alpha}}$ and $\boldsymbol{\acute{\beta}}$:

$$E_{\text{stat}} = \sum_{i=1}^{N_g} \acute{\lambda}_i \acute{\alpha}_i^2 + \sum_{i=1}^{N_r} \acute{\omega}_i \acute{\beta}_i^2. \tag{5.6}$$

Since the 3DMM bases are normalised by their standard deviation, the statistical average of $\acute{\alpha}_i^2$ and $\acute{\beta}_i^2$ should be kept to 1 during training. We do this by controlling the loss weight $\acute{\lambda_i}$ and $\acute{\omega}_i$.

During training, the weights for each 3DMM coefficient in statistical regularisation E_{stat} are adaptively adjusted so that the exponential moving average of the squared value of each coefficient is kept to 1, which is equivalent to the variance defined in the 3DMM. Assuming an arbitrary element 3DMM parameters in the jth iteration is α_j, an arbitrary element of the weight vector $\acute{\lambda}_j$ is set by:

$$\acute{\lambda}_j = \max(\min(k_j E[\acute{\lambda}]_j, \acute{\lambda}_{\max}), \acute{\lambda}_{\min}), \tag{5.7}$$

$$E[\acute{\lambda}]_j = (1-\theta)\acute{\lambda}_{j-1} + \theta E[\acute{\lambda}]_{j-1}, \tag{5.8}$$

$$k_j = \max(\min(E[\alpha^2]_j, \alpha^2_{\max}), \alpha^2_{\min}), \tag{5.9}$$

$$E[\alpha^2]_j = (1-\theta)\alpha^2_{j-1} + \theta E[\alpha^2]_{j-1}, \tag{5.10}$$

where θ is the weight for exponential moving averaging, and $\acute{\lambda}_{max}$,$\acute{\lambda}_{min}$,α_{max}, and α_{min} are the bounds for the update. This weight control is also applied to photometric 3DMM coefficients, replacing α by β and $\acute{\lambda}$ by $\acute{\omega}$. Note that $\acute{\lambda}$ and $\acute{\omega}$ are different from regularisation weights for least-squares λ and ω in Section 3.3 and 3.4. We optimise λ and ω as trainable parameters by the optimiser together with the network parameters. We implicitly control them through the statistical regularisation loss.

Camera intrinsics regularisation loss

Finally, we employ regularisation on the estimated camera intrinsic parameters. This penalises the difference between vertical and horizontal focal length as well as the

shear:

$$E_{\text{int}} = \eta_{\text{asp}} \frac{(k_{11} - k_{22})^2}{k_{11}^2 + k_{22}^2} + \eta_{\text{sh}} \frac{k_{12}^2}{k_{11}^2 + k_{22}^2}, \tag{5.11}$$

where the k_{ij} are the elements of the intrinsic camera parameter matrix $\mathbf{K}$. The first term represents the difference between vertical and horizontal focal length and the second term represents the shear component. We normalise the loss by the horizontal and vertical focal length to avoid reducing the scale of focal length. We set $\eta_{\text{asp}} = 1.0$ and $\eta_{\text{sh}} = 1.0$.

5.2.3 Training

Initialisation

Supervision of the LSDR network relies on the difference of appearance between the input image and the estimated face. Therefore, the initial estimation must be sufficiently close to the optimal parameters to obtain a meaningful gradient from the loss function. We pretrain the network using a small number of roughly aligned images by applying data augmentation through 2D similarity transformation. In pretraining, we directly supervise the pixel-wise prediction network using a constant value depth map, synthetic confidence map, and synthetic correspondence map. We align the mean shape of the 3DMM to pretraining images using the averaged positions of five landmarks, and generate a synthetic confidence map, in which the face region is set to 1 and the rest to 0, and a synthetic correspondence map. The same supervision data is used for all the pretraining images. We apply a random similarity transformation to both input images and supervision data. An example of an input image and supervision data is shown in Figure 5.2. Though we use roughly aligned images for pretraining, we never use the landmarks of each image and the 3D ground truth. Thus, the LSDR network can be regarded as unsupervised training in the conventional context. We initially pretrain the network using 1k images from the pre-aligned CelebA dataset. Here, the batch size is 5, and the number of iterations is 14k.

During early iterations of the main training, we additionally regularise the camera translation parameters in the linear least-squares system, as the calculation of full perspective camera parameters from planar depth tends to be unstable. Camera

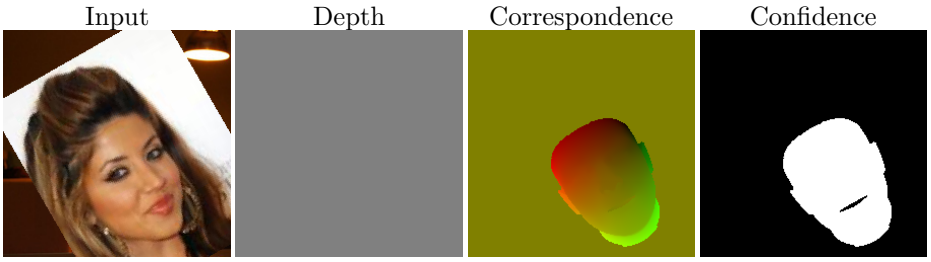


Figure 5.2: Example of training data for pretraining of LSDR.

translation parameters are regularised by applying L2 distance regularisation between the camera viewpoint and a fixed point placed in front of the face.

Training data

We train on $\sim$ 200k images from the pre-aligned CelebA dataset [46]. We augment with random 2D similarity transformations (magnification ratio: $[0.77, 1.3]$, translation: $[0, 75]$ pixels horizontally and vertically, rotation $[-180°, 180°]$). The background region is filled by random images from ImageNet[39] with a blended boundary. Finally, we crop the image by 224×224 pixels.

Optimisation

We use the Adadelta optimizer [82] with a learning rate of 0.01, batch size of 3, and 300k iterations. Network weights and biases are initialised by He initialisation [28]. Training takes approximately 120 hours on an Nvidia GTX 1080Ti.

5.2.4 Evaluation

Qualitative evaluation

We qualitatively evaluate the LSDR method based on test images from the CelebA dataset (Figure 5.3). The LSDR method successfully predicts a 3D face including ears under arbitrary 2D similarity transformation. We compare LSDR method with MoFA [66], which can only reconstruct the centre region of a face, whereas LSDR can reconstruct a uncropped face. LSDR also has better fidelity of reconstruction due to the optimality of the least-squares. We also test multiframe aggregation of the

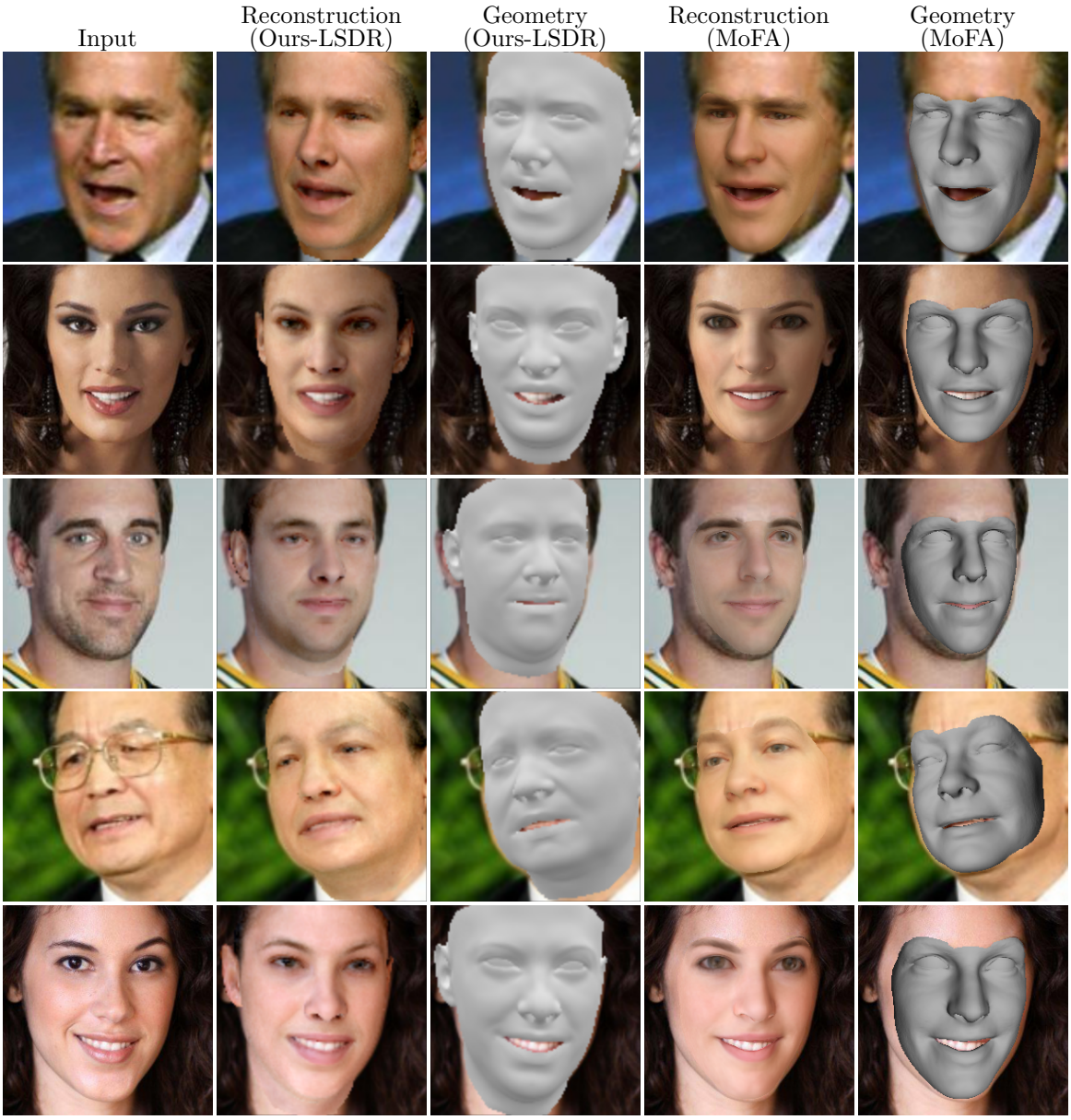


Figure 5.3: Reconstruction result of MoFA [66] and LSDR from images in MoFA-test dataset.

pixel-wise prediction (Figure 5.4). By optimising multiframe geometry and albedo to the intermediate output in a single optimisation, a superior quality of output can be obtained.

Figure 5.4: Results of multiframe aggregation from five frames based on LSDR.

	Median	Mean	Std	Supervision
Tran [69]	1.83	2.33	2.05	Fully supervised
PRNet [16]	1.51	1.99	1.90	Fully supervised
RingNet [60]	1.23	1.55	1.32	Landmarks, ID
Ours-LSDR	1.52	1.89	1.57	None

Table 5.1: Quantitative evaluation of LSDR on NoW dataset [60]. Figures in the table are in the unit of millimetres.

	Error(HQ)	Error(LQ)	Error(Full)
MTCNN-CNN6-eos [17]	2.70 ± 0.98	2.78 ± 0.95	2.75 ± 0.93
MTCNN-CNN6-3DDFA [17]	2.04 ± 0.67	2.19 ± 0.70	2.14 ± 0.69
SCU-BRL [68]	2.65 ± 0.67	2.87 ± 0.81	2.81 ± 0.80
Ours-LSDR (w/o E_{int})	2.65 ± 0.98	2.60 ± 0.83	2.62 ± 0.88
Ours-LSDR	2.39 ± 0.81	2.55 ± 0.82	2.49 ± 0.82

Table 5.2: Quantitative evaluation of LSDR on Stirling/ESRC 3D Face Database [1][17]. Figures in the table are in the unit of millimetres.

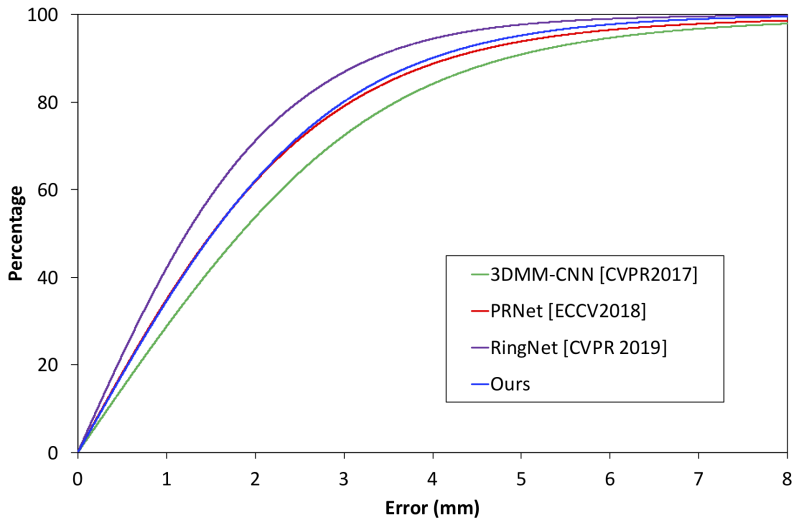


Figure 5.5: Cumulative error of LSDR(Ours) for the NoW dataset [60].

Quantitative evaluation

We quantitatively evaluate the LSDR method based on landmarks (Table 5.3). We follow the evaluation protocol proposed in Zhu et al. [88] and compare LSDR with supervised facial landmark detection methods. We evaluate landmarks obtained from both direct correspondence and fitted model. LSDR shows comparable result to some supervised methods, despite LSDR being unsupervised.

We quantitatively evaluate LSDR on the NoW dataset [60] (Table 5.1, Figure 5.5) and the Stirling/ESRC 3D Face Database (Table 5.2), in which the error of reconstructed neutral face shape is calculated. LSDR does not outperform other methods that use richer supervision, though it is comparable to some supervised methods despite LSDR being unsupervised.

Ablation study

We investigate the contribution of each loss function, qualitatively (Figure 5.6) and quantitatively (Table 5.2). The right column in Figure 5.6 shows the result trained by only the reconstruction loss and the statistical regularisation. This is a clear example of a shrinking problem, and the robust residual loss significantly improves the

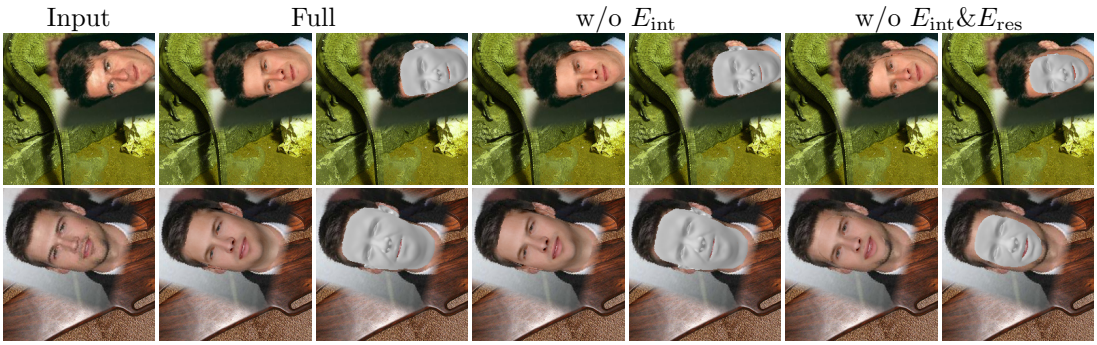


Figure 5.6: Ablation study to show the contribution of intrinsic parameter regularisation E_{int} and robust residual loss E_{res} in LSDR. We show input, then for each condition we show overlaid reconstruction followed by overlaid geometry.

problem. From Figure 5.6 and Table 5.2, it is also clear that the intrinsic parameter regularisation enables the reconstruction of plausible and a precise shape.

Performance versus training iteration

Figure 5.7 shows the convergence of reconstructed images during the training. The initial estimate is based on the pretrained network, which only requires a small amount of roughly aligned images for supervision. Reconstructed face region expands gradually as training proceeds (odd rows in Figure 5.7). The number of inlier pixels, which has a larger robust residual error than the threshold, also increases during the training (dark pixels in even row images in Figure 5.7).

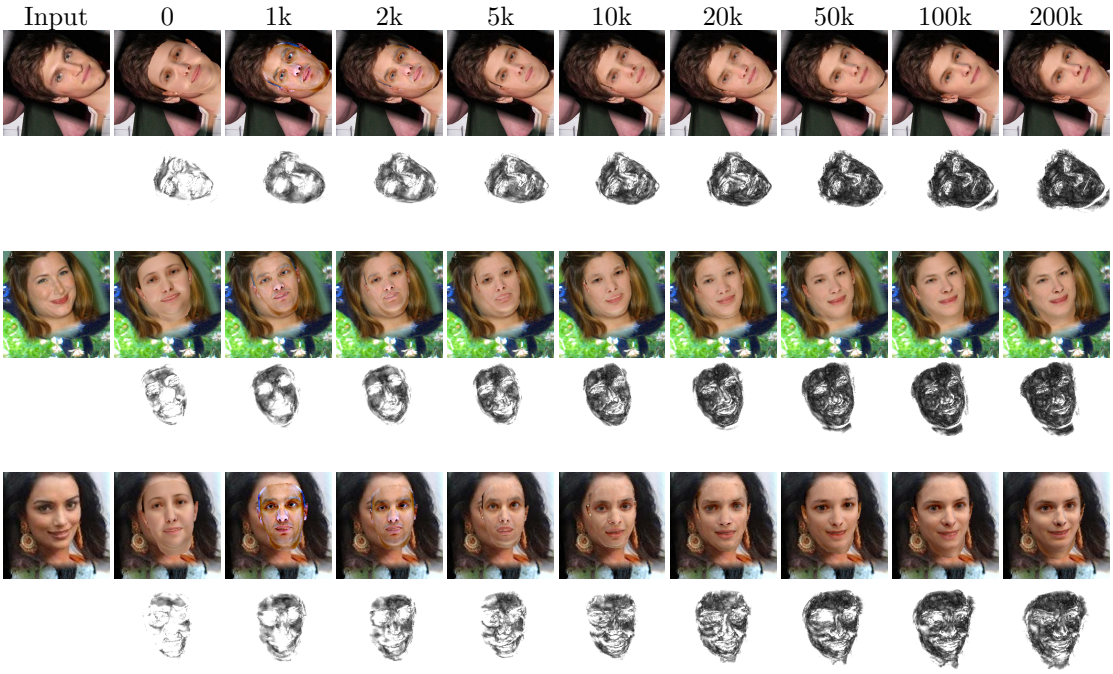


Figure 5.7: Convergence of images reconstructed by LSDR during training. Odd rows show the overlay of the reconstructed image. Even rows show the visualisation of the robust residual loss on each pixel.

5.3 GRD-based segmentation supervision

In this section, we show how to train a network to estimate a shape (restricted to a single object class via a 3D morphable model) using a semantic segmentation map of a single 2D image. To this end, we use the geometric Rényi divergence, discussed in Chapter 4, as a loss function to train a neural network. We represent both the projection of semantic labels on model vertices and the semantic labels on pixels as mixtures of Gaussians, and compute the discrepancy between the two based on the geometric Rényi divergence. The resulting loss is differentiable, and has a wide basin of convergence.

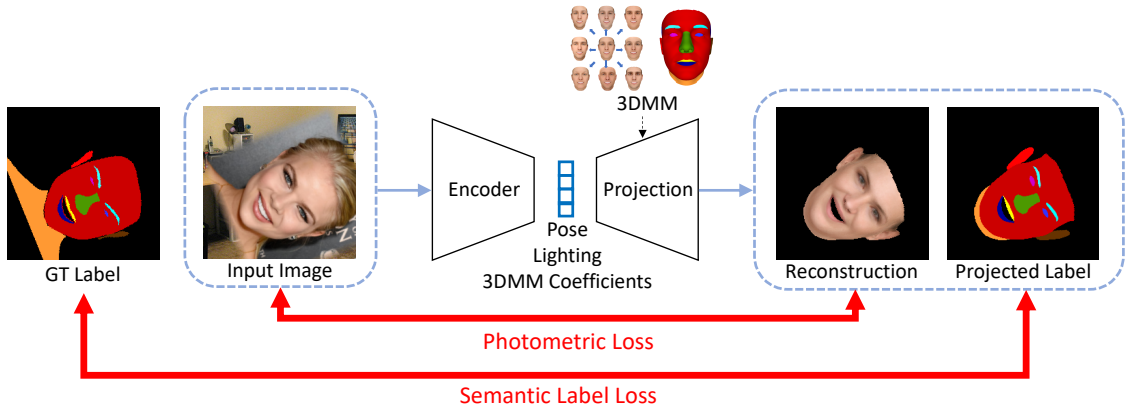


Figure 5.8: Parameter regression CNN architecture with semantic segmentation supervision.

5.3.1 Architecture

Figure 5.8 shows a network for 3D face reconstruction based on semantic label loss. This can be viewed as a variant of MoFA [65] with additional semantic segmentation supervision. An encoder network predicts pose (camera) parameters and 3DMM coefficients. The semantic label loss is calculated in the same manner as the analysis-by-synthesis experiment in Section 4.4.5. To reconstruct colour information, we also estimate lighting and 3DMM albedo coefficients, and minimise the L2 norm of the difference in colour between the input image and shaded vertices based on Equation (3.3).

5.3.2 Label correction

Pixel-wise labels contain some classes or face regions not present in the model. For example, glasses may occlude the face while the neck and forehead are cropped in the model. Since the network learns the alignment between the image and the model, inconsistencies in the label definitions could cause the network training to fail. Therefore, we propose to correct these labels using a provisional network. Having trained using classes from the original labels that are present in the model (see Figure 5.9, col. 2), we obtain initial model-based estimates (col. 3). We update the original labels by allowing a potential occluder class to be replaced with a face class or a face class

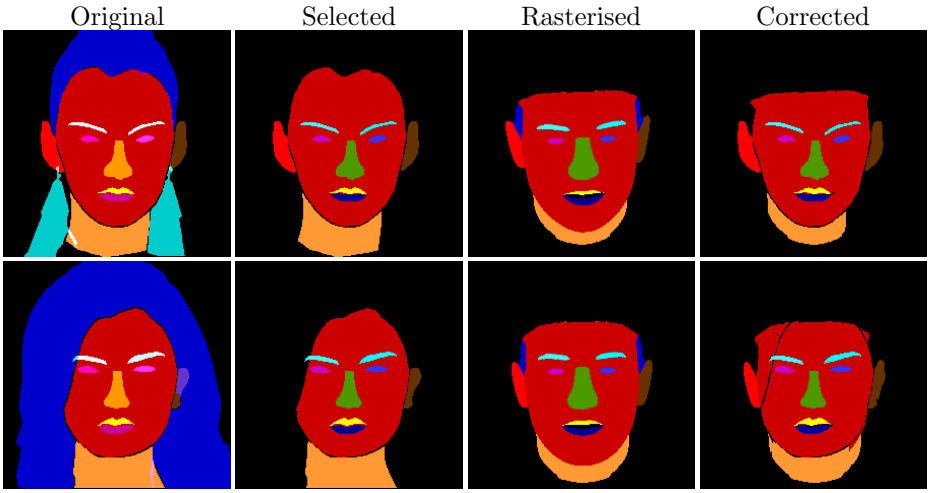


Figure 5.9: Label correction based on rasterised semantic labels generated by a provisional network.

to be replaced with background (col. 4). This can be viewed as a statistical inpainting of occluded regions.

5.3.3 Camera parameterisation

To take advantage of supervision based on the GRD for semantic labels, which can address large displacement and rotation, we employ a 6D redundant expression for camera rotation and an explicit perspective distortion parameter, following Zhou et al. [84]. The network estimates 13-dimensional parameters $[r_1, r_2, r_3, r_4, r_5, r_6, t_x, t_y, t_z, f, g, c_x, c_y]$. From the estimated parameters, the intrinsic camera matrix $\mathbf{K}$,the rotation matrix $\mathbf{R}$, and the translation vector $\mathbf{t}$ are given by:

$$\mathbf{r}_y = [r_1, r_2, r_3]^t, \mathbf{r}_z = [r_4, r_5, r_6]^t, \tag{5.12}$$

$$\acute{\mathbf{r}}_z = \mathbf{r}_z - (\mathbf{r}_z^t \mathbf{r}_y)\frac{\mathbf{r}_y}{\|\mathbf{r}_y\|^2}, \tag{5.13}$$

$$\bar{\mathbf{r}}_y = \frac{\mathbf{r}_y}{\|\mathbf{r}_y\|}, \bar{\mathbf{r}}_z = \frac{\acute{\mathbf{r}}_z}{\|\acute{\mathbf{r}}_z\|}, \bar{\mathbf{r}}_x = \bar{\mathbf{r}}_y \times \bar{\mathbf{r}}_z, \tag{5.14}$$

$$\mathbf{R} = \begin{bmatrix} \bar{\mathbf{r}}_x & \bar{\mathbf{r}}_y & \bar{\mathbf{r}}_z \end{bmatrix}, \tag{5.15}$$

$$\mathbf{t} = [gt_x, gt_y, gt_z]^t, \tag{5.16}$$

$$\mathbf{K} = \begin{bmatrix} gf & 0 & c_x + (1-g)ft_x/t_z \\ 0 & gf & c_y + (1-g)ft_y/t_z \\ 0 & 0 & 1 \end{bmatrix}. \tag{5.17}$$

5.3.4 Training

We now use the GRD-based loss to train a network to reconstruct 3D faces from a single image. The network estimates 3DMM coefficients for both shape and albedo, pose parameters, and lighting parameters. The visibility and the weight for each vertex used to calculate the GRD are computed in each iteration of the training. Pose parameters are represented by a 3D translation vector, a rotation matrix, and a parameter to express perspective effect. We use the Basel Face Model 2017 as 3DMM, which has 299 bases for shape and 100 for albedo. Rotation matrix is parameterised by 6D redundant expression, which consists of two 3D vectors. Rotation matrix is generated from the vectors using the Gram-Schmidt process. We employ individual VGG19 networks to estimate 3DMM coefficients, lighting parameters, and pose parameters respectively.

We train the GRDDR network using the CelebAMask-HQ dataset. We use left/right ears, left/right eyes, left/right eyebrows, upper/lower lips, nose, face, and neck labels for training and visualisation. We split the dataset into 29,000 training images and 1,000 test images. We augment with random 2D similarity transformations (magnification ratio: $[0.654, 1.105]$, translation: $[-56, 56]$ pixels, rotation: $[-180°, 180°]$). The background region is filled by random images from ImageNet [39] with blended boundary. Finally, we crop the image by 224×224 pixels.

We begin by only training the pose estimation network for 10,000 iterations with batch size 5 using the original labels. Then, using the corrected labels, we train the pose and lighting estimation networks for 40,000 iterations with batch size 5. Consequently, we add the 3DMM estimation network and train the networks for 240,000 iterations with batch size 2. We employ the Adadelta optimiser to train the networks with a

learning rate of 0.001 for the final training of lighting and pose and 0.01 for the rest of the training.

5.3.5 Evaluation

Figure 5.10 shows qualitative results of the reconstruction. The GRDDR method successfully reconstructs the 3D face including ears under arbitrary 2D similarity transformation. We quantitatively evaluate our method based on landmarks (Table 5.3). We follow the evaluation protocol proposed in Zhu et al. [88] and compare GRDDR with supervised facial landmark detection methods. GRDDR shows comparable results to landmark-based methods for modest pose angles.

5.4 Backwards rasterisation: Distance-transform-based segmentation supervision without differentiable rendering

Model-based self-supervision uses explicit physics-based or geometric models that are implemented in-network, providing a supervisory signal via a reconstruction error, forming a model-based autoencoder [65]. Such approaches avoid the need for ground truth supervision and have been applied to tasks such as 3D face reconstruction, inverse rendering, and monocular depth estimation.

In many cases, the model includes a renderer to compute a 2D image from 3D geometry and material properties. However, rendering is not differentiable. Whether or not a surface point is visible to a particular pixel location is a binary (and hence discontinuous) function. Rasterisation computes this mapping explicitly, taking as input a model that depends on continuous parameters (e.g. vertex positions, transformation and camera parameters) and outputting a series of buffers that encode the discrete pixel to model correspondences and other per-pixel quantities required for screen space rendering. Similarly, sampling image intensities onto model vertices in 3D (for 3D model fitting [65]) is not differentiable since it is discontinuous when a sample point moves outside the image boundary or when a vertex becomes self-occluded.

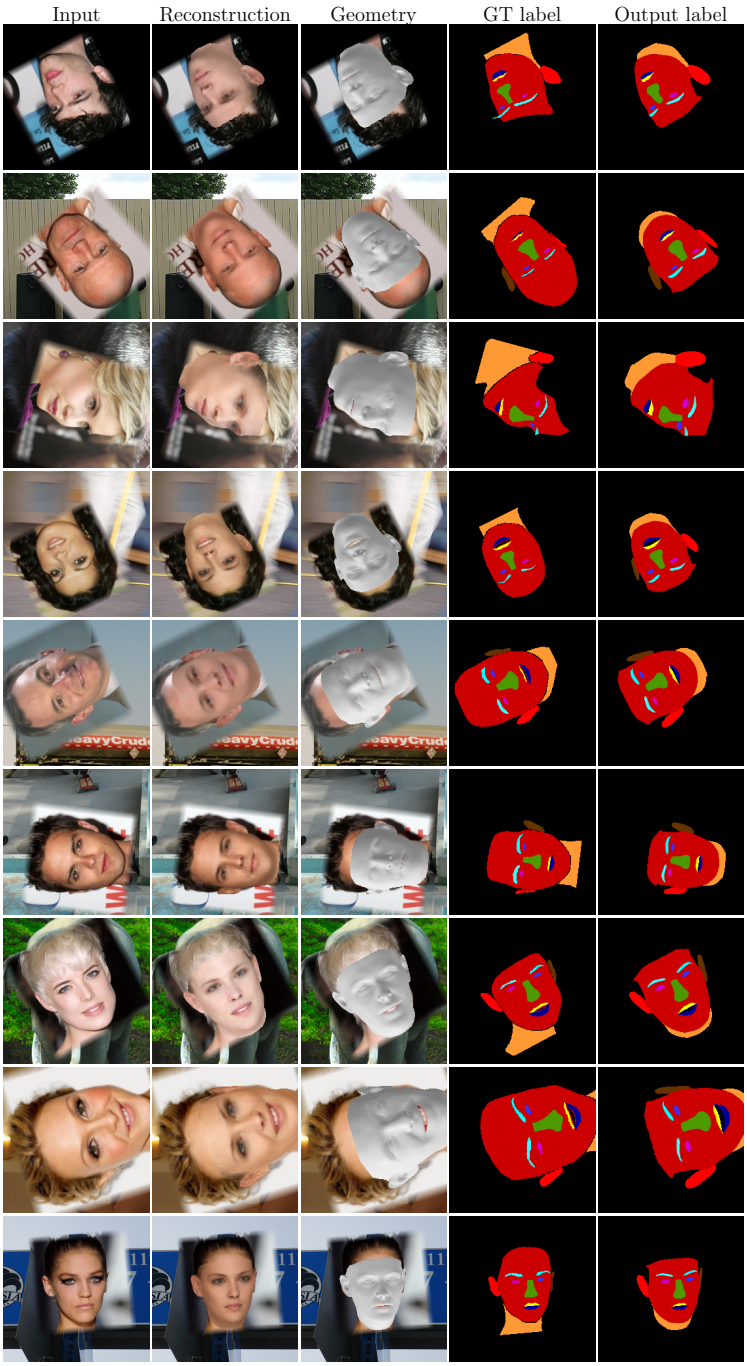


Figure 5.10: Reconstruction results of GRDDR.

When a network is trained using a loss that includes rendering or image sampling, the supervisory signal (gradient) cannot convey information about the effect of these discontinuous changes. Therefore the network cannot learn how changes in visibility affect the loss. Differentiable renderers seek to soften the rendering pipeline in some way, for example by blurring the rasterisation as in SoftRas [45], smoothly extrapolating the rasterisation as in Neural Mesh Renderer [35] or volume rendering soft density as in NeRF [50]. This weakness applies to the LSDR method proposed in Section 5.2. Here, we seek to overcome this limitation.

Our idea avoids forward rasterisation entirely, enabling model-based self-supervision *without a renderer and without a reconstruction loss*. The idea is to task a network with predicting the buffers that would have arisen in the forward rendering that produced the input image. The desired model, e.g. geometry, albedo and lighting, is recovered by solving an optimisation problem to find the model that best fits the buffers. The buffers are chosen such that the optimisation problem is easier to solve than the original problem (ideally with a closed-form solution) and the solution must be differentiable. If this is the case, the optimisation can be implemented in-network and the residuals of the model fit become the learning signal that is backpropagated through the optimisation. Our approach eliminates a reconstruction loss (and hence forward rendering) entirely. If part of the model is occluded, then there will simply be no pixel predicting correspondence to that part of the model, sidestepping the non-differentiability of occlusions.

Note that the method in Section 5.2 is already very close to this idea. However, here we introduce the conceptual step of viewing this process as *backwards rasterisation*. This is a more general idea of which we only consider one realisation, however to gain the benefits of avoiding forwards rasterisation, the differentiable renderer must be removed. The cost of removing supervision from the differentiable renderer entirely is that the problem becomes too ill-posed and training cannot converge. However, this problem can be addressed by introducing segmentation supervision. The distance transform based method that we introduced in Section 4.3 is ideally suited for this purpose. It provides weak supervision for the correspondence estimation task, ensuring that face parts predict correspondences at least to the right region of the face

model, while the model-based regularisation further constrains correspondences to be consistent with a shape explainable by the model.

5.4.1 Architecture

We now describe one realisation of the backwards rasterisation idea. Unlike existing 3D reconstruction methods, which directly predict 3D shape or regress latent parameters of a 3D morphable model (3DMM), the LSDT method reconstructs buffers of the rendering pipeline in image pixel space with a neural network. While many possible implementations are possible, we choose a minimum viable implementation in order to illustrate our idea. We predict a correspondence map (comparable to a face buffer in conventional rasterisation), a depth map (equivalent to a Z-buffer) and a confidence map (comparable to a stencil buffer). Specifically, the correspondence map predicts, for every pixel in the input image, the corresponding UV coordinate on the template 3DMM. The depth map predicts the distance to the face surface at every pixel. The confidence map predicts a probability, indicating whether the network believes a pixel is explainable by the model. We refer to this approach as LSDT (Least Squares with Distance Transform).

We train the LSDT network using four losses and only minimal supervision (semantic segmentation maps):

1. **Residual loss**: this measures the goodness of fit of the 3DMM to the predicted buffers. This acts as model-based regularisation, requiring the network to predict depth and correspondence maps that are close to a face that is realisable within the model. The model fit itself is made robust by using the predicted confidence map to weight the contribution of pixels.

2. **Segmentation loss**: we use ground truth semantic segmentation labels on the input images. Each input pixel is transported to UV space via the estimate in the correspondence map, where we measure distance to the segment with the ground truth class label for that pixel. This is our UV space distance transform based cohesive measure from Section 4.3.2.

3. **Confidence loss**: we binarise the ground truth segmentation of face parts and use this to supervise confidence.

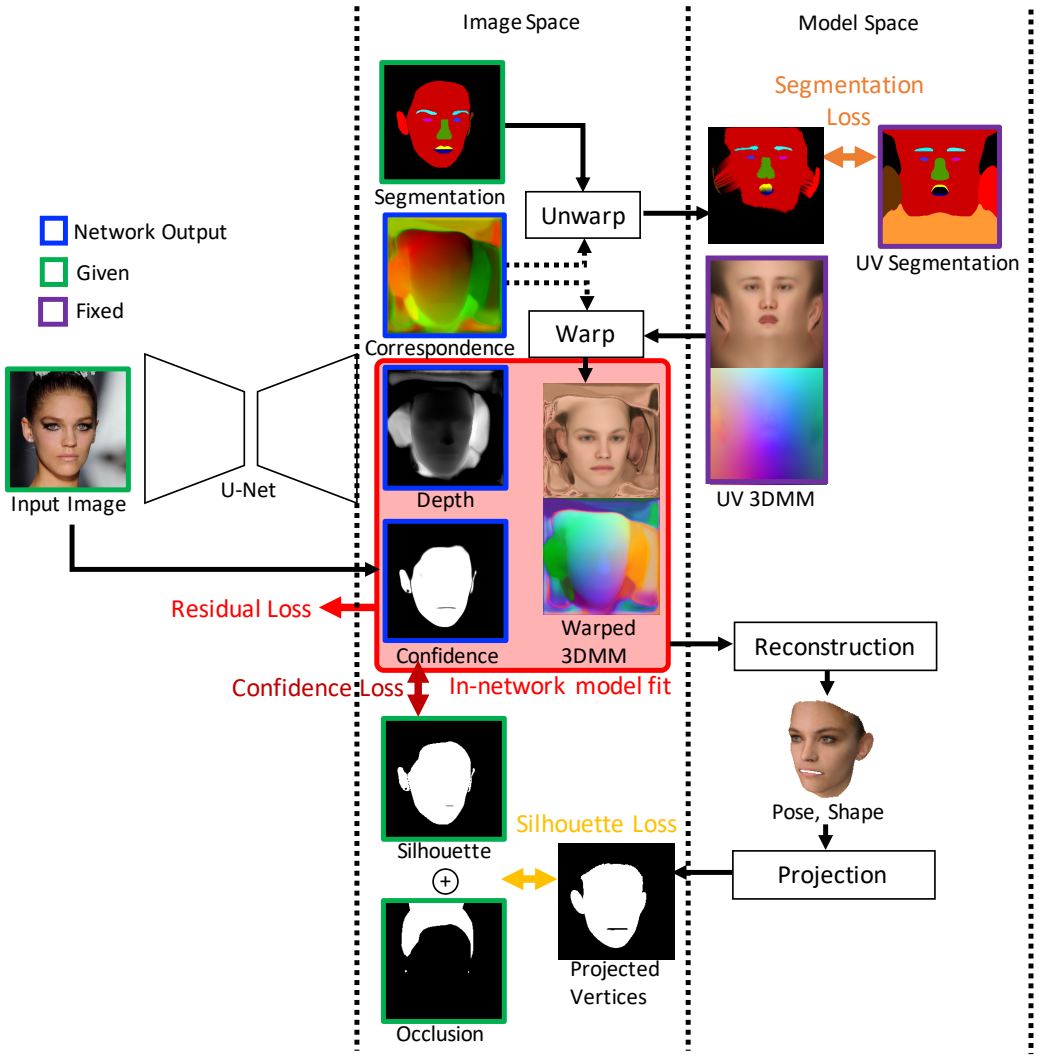


Figure 5.11: The LSDT network predicts correspondence, depth and confidence maps from a single image. At training time, ground truth semantic labels are unwarped to UV space via the estimated correspondences and a semantic segmentation loss is computed against the fixed semantic labels of the model. The 3DMM is warped from UV space to image using the estimated image-to-model correspondences. The model is fitted to the estimated depths, weighted by confidence and the residuals of the fit provide another training signal. We directly supervise the confidence map estimates with ground truth face part segments, then add segments that potentially occlude face parts, project fitted model vertices into the image and compute a silhouette loss.

4. **Silhouette loss**: this is a counterpart of segmentation loss. Once the 3DMM has been reconstructed from the estimated buffers, we project the reconstructed vertices back into the input image and penalise any vertices lying outside the known silhouette. Note that we can require all vertices to lie inside the silhouette without computing visibility or rasterising the model. This is our image space distance transform based cohesive measure from Section 4.3.2.

An overview of the training architecture is shown in Figure 5.11. The silhouette and segmentation loss together ensure that our estimated models closely adhere to occluding boundaries of the face surface. The LSDT approach enables prediction of both accurate dense depth maps (with no depth supervision) as well as the best fit meshes and the albedo map within the space of the 3DMM (computed in-network via the solution of an efficient linear least-squares problem). At inference time, given only an image, we predict the four maps and the corresponding least-squares model fit parameters.

Buffer reconstruction network

We employ a standard U-Net [59] architecture for buffer reconstruction. The number of output channels is four: two for the U and V components of the correspondence map, one for depth, and one for confidence. To restrict the range of output values, we use the sigmoid activation for correspondence and confidence and the absolute value for depth.

5.4.2 Weak supervision based on distance transform

For both segmentation loss and silhouette loss, we use a distance-transform-based soft cohesive measure, shown in Section 4.3, to encourage a group of points (either points in UV space or projected vertices) to align with a region in an image. Intuitively, a semantic pixel label restricts the possible image-to-model correspondence to only the region with the correct semantic class on the model, while a binary silhouette restricts the projection of model vertices to only lie within the silhouette. Minimising such a measure provides weak supervisory signals for establishing dense image-to-model correspondence. In this scenario, some parts of the template shape are invisible from

the input image. Therefore, the geometric Rényi divergence, which aims to achieve exact matching, is inappropriate. For this reason, we use the distance-transform as a soft cohesive measure and combine the semantic segmentation loss and the silhouette loss.

5.4.3 Hierarchical model-based regularisation

Simultaneous optimisation of both segmentation loss and residual loss is not straightforward. When far from a good solution, often the descent direction to reduce segmentation loss increases residual loss. Such conflicts between both losses cause local minima in the loss landscape and the network fails to learn. Furthermore, buffers predicted by the network must always be coherent since least-squares model fitting only provides a meaningful result with plausible inputs. To overcome these issues, we introduce a hierarchical training scheme, in which model complexity increases as training proceeds.

Initialisation

At the initial stage of training, we supervise the network with a fixed synthetic target. We render the mean shape of the 3DMM with fixed camera parameters, which represent a frontal view in an appropriate scale, and generate a ground truth correspondence map and confidence map. We directly supervise the network so that it predicts the fixed synthetic correspondence map and confidence map from any given input image. No model-based regularisation is applied during this stage.

Weak perspective camera fitting

Subsequently, the network is trained with model-based regularisation, assuming a weak perspective camera model described in Section 3.3.3. This model is very stable and can align large pose deviation.

Full perspective camera fitting

Next, we relax the camera model to a less restrictive full perspective camera model. This model only relies on pose fitting and does not fit 3DMM coefficients. In addition to the supervision for output correspondence maps and confidence maps, we introduce supervision for output depth maps simultaneously during this phase. During this phase, the model learns both correspondence and depth maps that conform to a best fit of the mean shape to the correspondence map under full perspective. Because of the erroneous outputs of the network and model mismatch, the fitted camera could result in the face being behind the camera. To detect and omit such cases, we decompose fitted $\mathbf{H_f}$ into an upper triangular matrix $\mathbf{K}_f$, $\mathbf{R}_f \in SO(3)$, and $\mathbf{t}_f \in \mathbb{R}^3$, such that $\mathbf{K}_f \begin{bmatrix} \mathbf{R}_f \mathbf{t}_f \end{bmatrix} = \mathbf{H}_f$. If the sign of the third element of $\mathbf{t}_f$ is negative, the result is determined as a failure case and omitted from the loss. This treatment is also applied to the final model.

Full 3DMM fitting

Finally, we apply the full linear least-squares layer, which is discussed in Section 3.4 and used in Section 5.2.

5.4.4 Loss functions

We train the LSDT network with four losses: $E_{\text{total}} = \eta_{\text{res}} E_{\text{res}} + \eta_{\text{seg}} E_{\text{seg}} + \eta_{\text{sil}} E_{\text{sil}} + \eta_{\text{int}} E_{\text{int}} + \eta_{\text{z}} E_{\text{z}}$ where, η_{res}, η_{seg}, η_{sil}, η_{int}, and η_{z} are weights for respective loss functions.

Segmentation loss

Segmentation loss is supervision for the correspondence map prediction. We apply the distance-transform-based cohesive measure, discussed in Section 4.3, and calculate the loss function based on (4.5). In this loss function, we treat each pixel in the predicted correspondence map as a movable target point, and calculate the distance-transform-based cohesive measure, in UV space using the precomputed distance maps. We assign a weight of 10 to the lip, eye, and eyebrow segments, and assign a weight of 1 to the rest, to encourage accurate reconstruction of important internal face details.

Residual loss

Residual loss is the residual of in-network fitting as discussed in Sections 3.3 and 3.4. It is computed by each model in the hierarchical training, described in Section 5.4.3, with the least-squares optimal parameters. In the loss calculation, the union of semantic segmentation labels is used as the confidence map instead of the predicted, and mask background and occluded region from the loss calculation.

Silhouette loss

Silhouette loss encourages the reconstructed 3D shape to project within the union of semantic segmentation labels additionally augmented by segments corresponding to classes that potentially occlude face parts (see Figure 5.11 bottom middle). We precompute the distance transform of this augmented segmentation mask as D_{sil} and calculate the loss: $E_{\text{sil}} = \sum_{i=1}^{N_v} D_{\text{sil}}(x'_i, y'_i)$, where x'_i and y'_i are projected the ith vertices of the 3DMM. The projection is calculated based on the fitted model parameters.

Confidence loss

We directly supervise a confidence map c by the union of semantic segmentation labels b. The loss function is defined based on binary cross entropy:

$$E_{\text{conf}} = \sum_{x=1}^{W}\sum_{y=1}^{H}\{-b(x,y)\log(c(x,y)) - (1 - b(x,y))\log(1 - c(x,y))\}. \quad (5.18)$$

Camera intrinsics regularisation loss

To prevent the LSDT network from predicting implausible outputs, we use the same camera intrinsics regularisation loss as that in Section 5.2.

Camera distance regularisation loss

To prevent the LSDT network from predicting implausible camera position in early iterations and the initialisation phase. We apply penalisation on too-close and too-far

camera position as:

$$E_{\mathrm{z}} = (|t_z - t_n| - (t_z - t_n))^2 + (|t_f - t_z| - (t_f - t_z))^2, \tag{5.19}$$

where t_z is the third element of fitted camera translation vector, t_n is near-side penalisation boundary, t_f is far-side penalisation boundary. If t_z is outside the range between t_n and t_f, quadratic penalty is applied.

5.4.5 Training

Network

We employ U-Net for the buffer reconstruction network. The number of channels at the original scale is 32, and downsampling is applied five times, in which the horizontal/vertical resolution is halved and the number of channels is doubled with subsequent upsampling.

Training data

We train the LSDT network using the CelebAMask-HQ dataset [40] and the Face Synthetics dataset [77]. We use left/right ears, left/right eyes, left/right eyebrows, upper/lower lips, nose, and face labels for supervision through segmentation loss, and glass, hair, hat, earring, necklace, and cloth labels are treated as occluded region. We split the CelebAMask-HQ dataset into 29k training images and 1k test images. We downsample images into 224×224 pixels. We use the Basel Face Model 2017 [23] as the 3DMM, which has 299 bases for shape and 100 for albedo. In the initialisation stages and early iterations, we only use the Face Synthetics dataset to leverage its large variations in pose and occlusions. To mitigate a problem caused by the difference between the 3DMM template and input images in the coverage of the face region, we reduce confidence values for pixels corresponding to the region near the face-neck boundary and the upper boundary in UV-space.

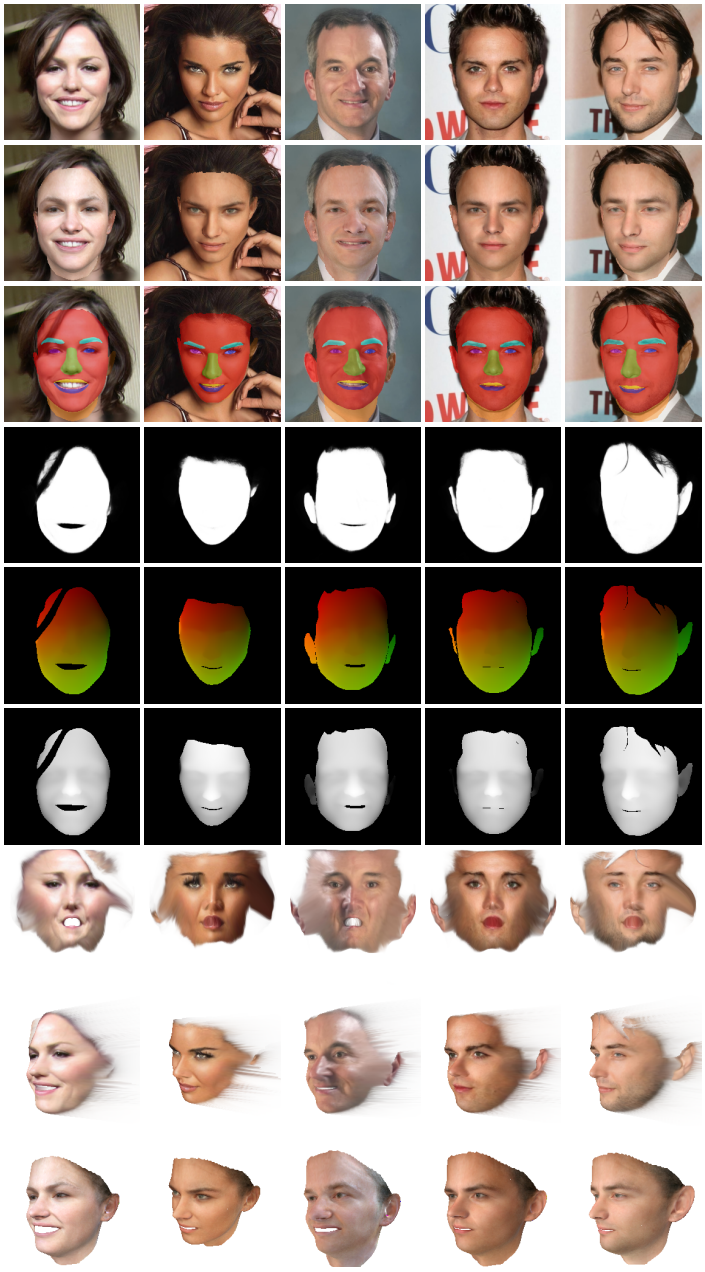


Figure 5.12: Results of reconstruction. Row 1: input image, Row 2: rendered reconstruction, Row 3: fitted segmentation, Row 4: output confidence, Row 5: output UV correspondence (cropped by silhouette), Row 6: output depth map (cropped by silhouette), Row 7: unwarped input image, Row 8: estimated depth map textured by input, Row 9: reconstructed 3DMM.

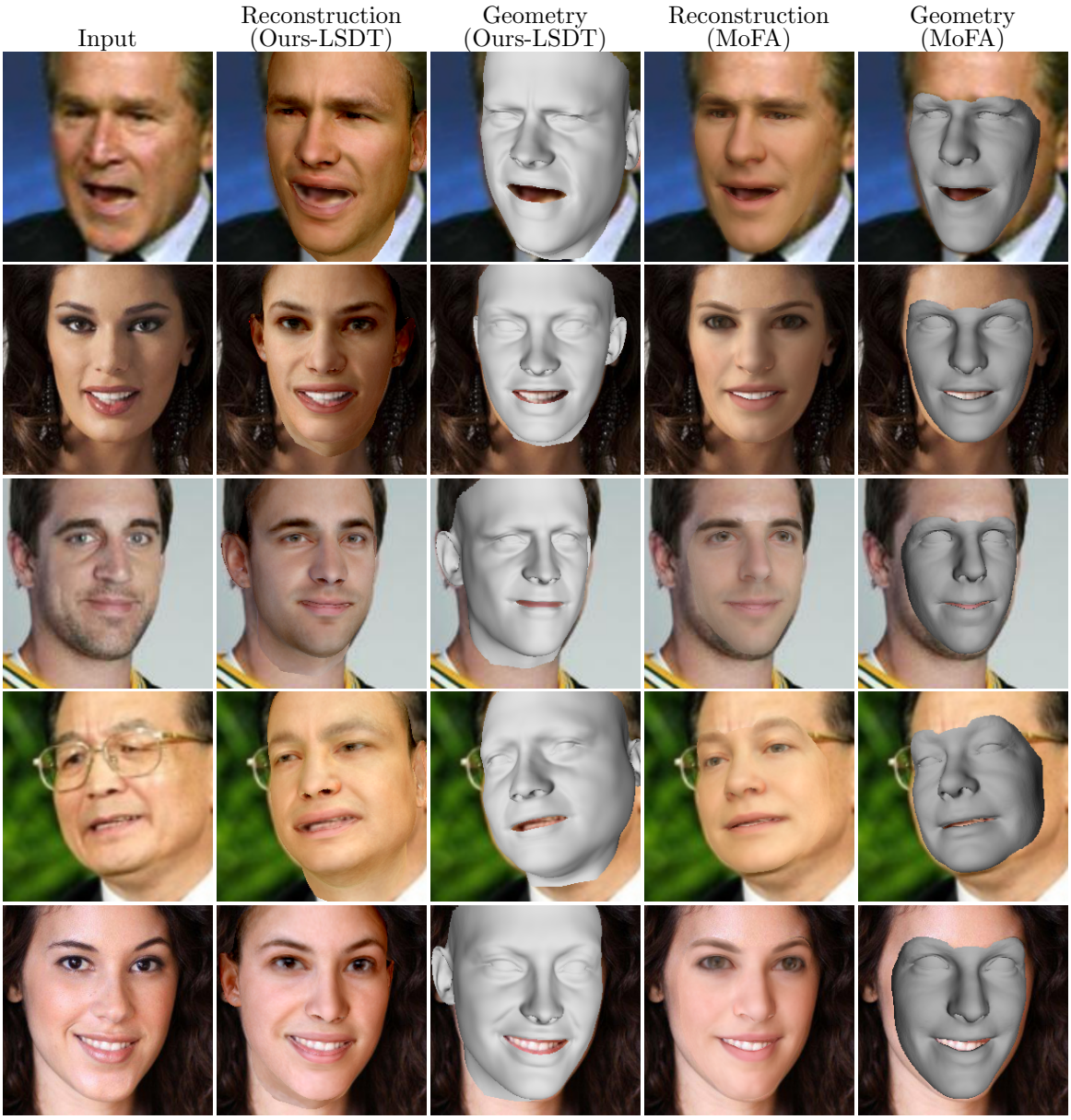


Figure 5.13: Reconstruction result of MoFA [66] and LSDT from images in MoFA-test dataset.

Optimisation

In the initialisation step, we use the Adam optimiser with a learning rate of 0.001 and optimise parameters for 100k iterations with a batch size of 1. After initialisation, we

use the Adadelta optimiser with a learning rate of 0.1. In the weak perspective step, full perspective step, and final step, we optimise parameters for 120k, 500k, and 1800k iterations with a batch size of 10, 5, and 2, respectively. Before the last 60k iterations in the final step, only the CelebAMask-HQ is used. In the full 3DMM fitting phase, we adaptively control the weights to regularise linear least-squares, corresponding to λ and ω in Section 3.3 and 3.4. Unlike the statistical regularisation loss in Section 5.2, we directly update the weights while updating them by the same process as those in Equation (5.7), Equation (5.8), Equation (5.9), and Equation (5.10).

5.4.6 Evaluation

Qualitative evaluation

We show network outputs and the reconstructed 3D model on the test data from the CelebAMask-HQ dataset (Figure 5.12). The LSDT method can align labels on the template mesh with input images (3rd row) and reconstruct fine structures and textures, while using a relatively inexpressive 3DMM. Estimated UV correspondence produces well-aligned unwarped images (7th row) and a good quality of depth prediction (rows 6 and 8). We also qualitatively evaluate LSDT on the MoFA-test images (Figure 5.13). LSDT can align whole head meshes of the 3DMM with input facial images, accurately matching occluding contours thanks to semantic-segmentation-based supervision. You can also see that LSDT can reconstruct fine structures. Additionally, in Figure 5.14 we visualise the progress of training with hierarchical model-based regularisation. This visualisation shows that LSDT can successfully align the input image with the reference in UV-space and reduce the error.

Quantitative evaluation

We quantitatively evaluate our LSDT on the NoW dataset [60] and the AFLW2000-3D [88] dataset. To align evaluation images with the input size, scale, and position of the network, we detect landmarks by using dlib face landmark detection [37] and MTCNN [83], and align and crop the image by a similarity transformation so that the MSE of landmarks is minimised. We conduct a landmark-based quantitative eval-

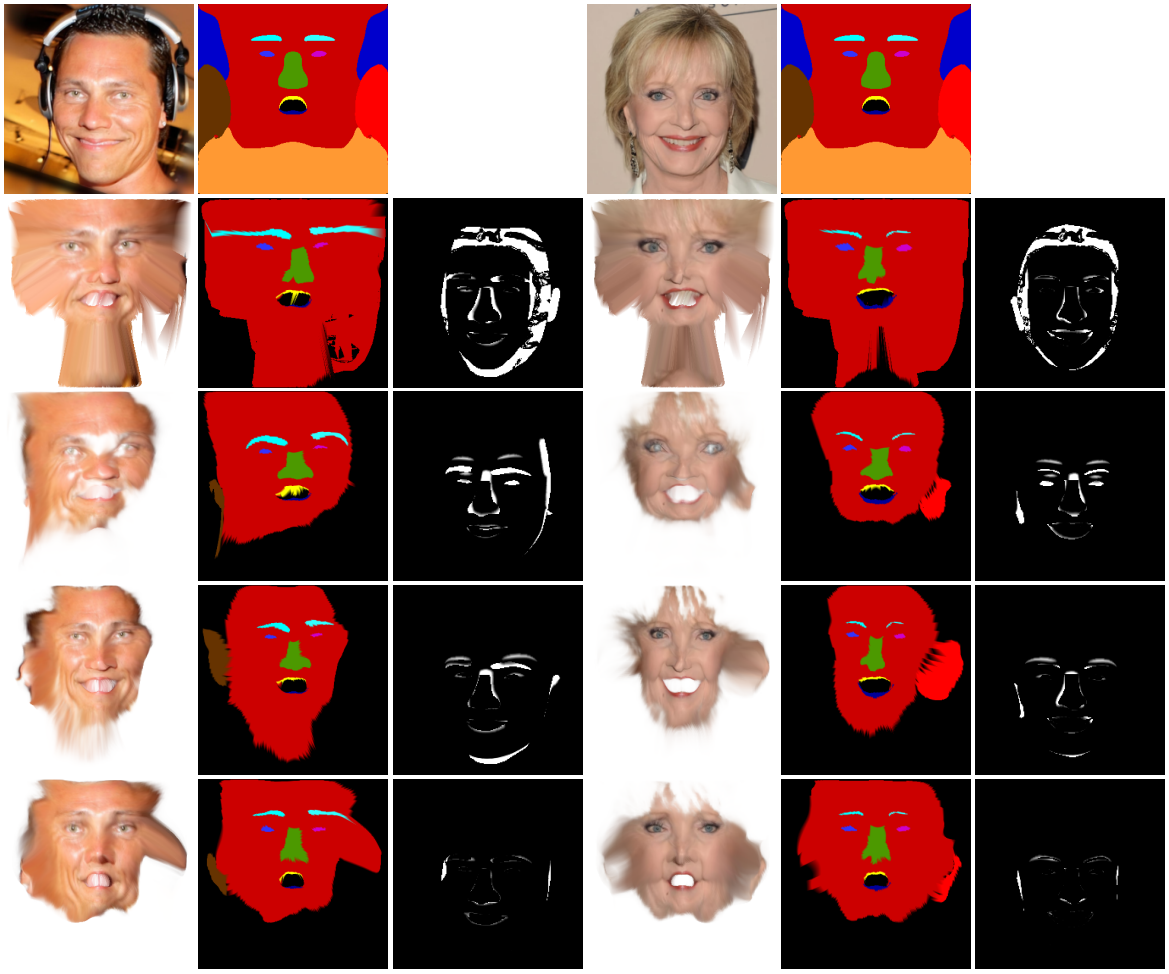


Figure 5.14: Visualisation of the progress of training with hierarchical model-based regularisation. Top row shows input images (left) and ground truth semantic segmentation labels (middle). In each row, visualisation of the output of each model: initialisation (2nd), weak perspective (3rd), full perspective (4th), 3DMM full (5th) is shown. Left column shows unwarped input images. Middle column shows unwarped semantic segmentation labels. Right column shows segmentation loss for each pixel.

uation on the AFLW 2000-3D dataset based on projection of landmarks on the reconstructed 3D model (Table 5.3). We use the evaluation method proposed by Zhu et al. [88]. While LSDT uses only weak supervision and model-based regularisation, LSDT shows comparable results to some supervised methods. We also conduct 3D-reconstruction-based quantitative evaluation on the NoW challenge (Table 5.4). Due

	AFLW Dataset			AFLW2000-3D Dataset		
Method	Mean[0-30]	Mean[0-90]	Std[0-90]	Mean[0-30]	Mean[0-90]	Std[0-90]
LBF [56]	7.17	17.72	10.64	6.17	16.19	9.87
ESR [7]	5.58	12.07	7.33	4.38	11.72	8.04
CFSS [85]	4.68	12.51	9.49	3.44	13.02	10.08
MDM [71]	5.14	13.40	9.72	4.64	13.07	10.07
SDM [80]	4.67	9.19	6.10	3.56	9.37	7.23
3DDFA [88]	4.11	4.55	0.54	2.84	3.79	1.08
PRNet [16]	4.19	4.77	-	2.75	3.62	-
Guo [25]	3.98	4.43	-	2.63	3.51	-
Ours-LSDR (Direct)	5.51	16.00	10.74	4.98	16.63	10.98
Ours-LSDR (Fitted)	5.87	18.63	13.20	4.74	18.55	13.38
Ours-GRDDR	3.98	10.14	5.99	4.97	10.49	5.64
Ours-LSDT	-	-	-	3.73	6.87	2.95

Table 5.3: Evaluation of LSDR, GRDDR, and LSDT on AFLW [49] and AFLW2000-3D [88] datasets. The accuracy is evaluated by the Normalised Mean Error, which is the dimensionless average error of landmark positions normalised by the $\sqrt{width \cdot height}$ of the face bounding box. [0-30] and [0-90] indicate the absolute yaw angle ranges of a face, measured in degrees.

to the relatively small representation power of the Basel Face Model 2017, which we use as the 3DMM, the performance of LSDT is moderate. However, it is still comparable with some supervised methods.

Post-optimisation with known intrinsics

One advantage of the buffer prediction network is the option to incorporate known camera parameters as post-optimisation after training. We minimise the objective function (Equation (3.17)) in Section 3.3 with fixed known camera intrinsics. Unlike the original formulation, this optimisation problem is nonlinear. Thus, we apply alternating optimisation, which consists of iterative PnP to obtain camera pose and linear least-squares to obtain 3DMM coefficients. We also show the result with known intrinsics in Table 5.4. Post-optimisation greatly improves the performance. Figure 5.15 is an example of 3D face reconstruction from an image in the NoW dataset. You can see that post-optimisation with known intrinsics reduces shape distortion.

	Non-metric			Metric		
	Median	Mean	Std	Median	Mean	Std
Tran [69]	1.84	2.33	2.05	3.91	4.84	4.02
PRNet [16]	1.50	1.98	1.88	-	-	-
RingNet [60]	1.21	1.53	1.31	1.50	1.98	1.77
Deng et al. [12]	1.11	1.41	1.21	1.62	2.21	2.08
3DDFA V2 [26]	1.23	1.57	1.39	1.53	2.06	1.95
MGCNet [63]	1.31	1.87	2.63	1.70	2.47	3.02
DECA [15]	1.09	1.38	1.18	1.35	1.80	1.64
SynergyNet [78]	1.27	1.59	1.31	2.28	2.86	2.39
Dib et al. [13]	1.26	1.57	1.31	1.59	2.12	1.93
MICA [89]	0.90	1.11	0.92	1.08	1.37	1.17
FOCUS [41]	1.04	1.30	1.10	1.41	1.85	1.70
Wood et al. [76]	1.02	1.28	1.08	1.36	1.73	1.47
Ours-LSDR	1.52	1.89	1.57	-	-	-
Ours-LSDT (w/o intrinsics)	1.42	1.75	1.44	2.33	2.99	2.62
Ours-LSDT (w/ intrinsics)	1.17	1.49	1.27	1.51	1.99	1.82

Table 5.4: Evaluation of LSDR and LSDT on NoW dataset [60]. Figures in the table are in the unit of millimetres.

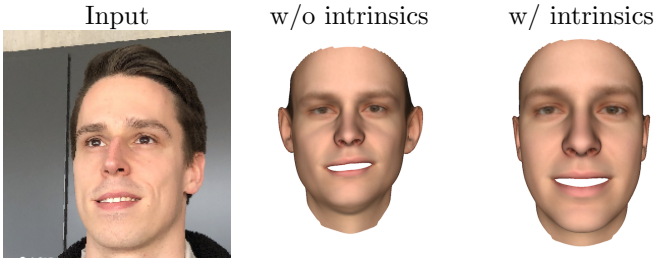


Figure 5.15: Reconstructed 3D face with and without known intrinsics from a NoW dataset image.

5.5 Conclusion

In this chapter, we have presented three novel paradigms for supervision of a 3D face reconstruction network: fully self-supervised training with a differentiable linear least-squares layer, GRD-based segmentation supervision, and self-supervised training without a differentiable renderer using semantic segmentation.

Firstly, fully self-supervised training with a differentiable linear least-squares layer

(LSDR) is a method which combines trainable pixel-wise face alignment with linear least-squares to reconstruct a 3D face model. To the best of our knowledge, this is the first method that enables the reconstruction of an uncropped face model under arbitrary in-plane transformation based on unsupervised training. LSDR has further potential to boost the performance of conventional supervised face alignment methods by harnessing abundant unlabeled images as well as application to a domain in which annotated images are scarce. In future work, LSDR can be further improved by incorporating an occlusion model, specular reflection, and a perceptual metric to alleviate the vulnerability of photometric error-based optimisation.

Secondly, GRD-based segmentation supervision (GRDDR) is a method that uses the closed-form GRD for spatial alignment of two MoG distributions based on gradient-based optimisation. GRD-based segmentation loss shows preferable characteristics in that it can reconstruct a 3D face from images with arbitrary in-plane rotation and large displacement. GRD has further potential for application to other computer vision tasks such as point cloud registration, image registration, and general 3D reconstruction. In particular, GRD is suitable for alignment based on soft landmarks, which predicts landmark position with uncertainty. GRD can be used for multiview silhouette fitting, extended to other object classes, and combined with pretrained semantic segmentation networks.

Lastly, self-supervised training without a differentiable renderer using semantic segmentation (LSDT) is a fundamentally different approach to model-based self-supervision compared to the wide array of existing methods that incorporate a differentiable renderer and a reconstruction loss. The motivation for doing so is that any differentiable renderer can only approximate the true, hard rendering process while the output of parameter-regression-based approaches is restricted to the fixed set of parameters chosen at training time. This makes it difficult to incorporate information such as known camera calibration at test time. On the other hand, image-space buffers provide an intermediate representation such that the actual model fitting process can be deferred, solved in a different way at test time or even not done at all if a depth or correspondence map provides a useful output. We present here only one realisation of backwards rasterisation. Obvious extensions would estimate additional buffers, for example, the

normal map (regularised by depth and 3DMM shape) and albedo (regularised by 3DMM) allowing out-of-model details to be reconstructed.

Additionally, when comparing our proposed methods through evaluations on 3D reconstruction (Figure 5.4) and landmarks (Figure 5.3), LSDT and GRDDR demonstrate improved performance over LSDR due to stronger supervision from segmentation labels. Specifically, LSDT excels by utilizing occlusion-free segmentation-based supervision, showing the best results among the three.

Chapter 6

Conclusion

In this thesis, we explored the utilisation of a differentiable linear least-squares layer and semantic-segmentation-based supervision for 3DMM-based monocular 3D face reconstruction. We have shown that a linear least-squares layer is useful for the regularisation of image-model correspondence estimation and demonstrated a fully self-supervised monocular 3D face reconstruction network. Additionally, we have presented supervision of a monocular 3D face reconstruction network based on semantic segmentation, and demonstrated self-supervised monocular 3D face reconstruction network without a differentiable renderer. In this chapter, we will summarise the contributions of this thesis and present them as overarching conclusions. We will also discuss the limitations of the proposed methods and suggest ideas of potential work building on the results of this thesis.

6.1 Summary of contributions

In Chapter 3, we proposed combining differentiable 3D model fitting with a neural network. Specifically, we focused on fitting of 3DMM coefficients, pose parameters, and low-dimensional lighting parameters. We tackled the photometric part and geometric part of the fitting problem separately. We formulated both problems as linear least-squares, which can be analytically solved via a pseudoinverse matrix in a differentiable manner. To linearise the problem for the photometric elements, we devised

an inverse spherical harmonics lighting technique. This lighting technique was empirically validated in Section 3.2.1. In-network model fitting is particularly beneficial for self-supervised learning as it can improve optimality of the output without necessitating the adjustment of learning rates for individual physical parameters of the model. The advantages were verified through network training experiments in Chapter 5. For the geometric part, we formulated the perspective camera model as a 3D affine transformation from the image space to the model space by introducing an auxiliary depth map. In this technique, image-model correspondence is established based on pixel-wise estimation by a neural network, and the 3DMM bases and mean corresponding to each pixel are efficiently obtained through bilinear interpolation of precomputed UV-3DMM, which is a 3DMM in UV-texture space. The obtained 3D affine transformation can be converted and decomposed into common intrinsic and extrinsic camera parameters in a differentiable way. We combined the geometric linear least-squares with the photometric one in a sequential manner. The combined techniques are demonstrated and verified through neural network training in Section 5.2 and Section 5.4.

In Chapter 4, we explored supervision based on the minimisation of a cohesive measure on a semantic segmentation labels. Specifically, we proposed the utilisation of bilinear sampling of precomputed distance maps for image-model correspondence as well as the application of geometric Rényi divergence to segmentation label alignment. In Section 4.3, we showed the framework of segmentation label alignment using the distance transform. The advantage of this method is its ability to handle self-occlusion, demonstrated in Section 5.4. In Section 4.4.2, we proposed a novel application of geometric Rényi divergence of Gaussian mixtures to the alignment of semantic segmentation labels. We devised a method to represent segmentation labels on both pixels and projected mesh triangles by weighted Gaussian mixtures and introduced a technique for numerical stability. The advantage of this method is that the values do not saturate, even when the displacement is large. This is important to make network training stable and to ease the difficulty of initialisation in model alignment tasks. Such characteristics bring about the possibility that semantic segmentation maps can be used as more informative source of supervision than landmarks. This advantage

is verified through loss landscape visualisation in Section 4.5.1, direct optimisation experiments in Section 4.5.2, and through network training in Section 5.3.

In Chapter 5, we proposed novel network training paradigms based upon the techniques proposed in Chapters 3 and 4. In Section 5.2, we explored completely self-supervised monocular 3D face reconstruction that does not require landmarks as supervision. Regularisation by the differentiable linear least-squares layer, as discussed in Chapter 3, allows a pixel-wise prediction network such as U-Net to be combined with a 3DMM. Thereby, we introduced a pixel-wise robust residual loss that assures the coverage of a fitted 3D model. In doing so, we achieved self-supervised training with no landmarks. In Section 5.3, we demonstrated the advantage of GRD-based supervision for network training based on images with large displacement and rotation. In Section 5.4, we proposed the training of a monocular 3D face reconstruction network that does not rely on a differentiable renderer, utilising a distance-transform-based metric as discussed in Section 4.3.

6.2 Overarching conclusions

Landmarks are not essential for training a model-based 3D reconstruction network.

In Section 5.2, we demonstrated the self-supervised training of a monocular 3D reconstruction network based on 3DMM. A careful design of the loss function can achieve stable training without auxiliary landmark supervision. The key is to introduce a robust residual loss on each pixel that encourages increased face coverage. By leveraging self-supervised training, we can improve the performance of a network by using a large number of unannotated images for training. Additionally, we can train a 3D reconstruction network in a domain where annotated images are unavailable, using self-supervision.

Model-based regularisation can be achieved using in-network linear least-squares.

To the best of our knowledge, this is the first study to utilise the in-network linear least-squares for introducing 3DMM-based regularisation on network outputs. We demonstrated its feasibility through network training experiments in Chapter 5. The photometric linear least-squares can contribute to improving the optimality of the fitting in appearance. The geometric linear least-squares allow an image-model correspondence estimation network to leverage model constraints provided by a 3DMM and a camera model. This enabled the introduction of new types of supervision. One example is the robust residual loss discussed in Section 5.2, which enables self-supervised training of a monocular 3D reconstruction network without landmark supervision. The other example is semantic segmentation loss in Section 5.4.

Spherical harmonics can model inverse lighting and this linearises lighting and albedo parameter estimation.

To the best of our knowledge, our research is the first to utilise inverse lighting, based on spherical harmonics, to linearise the 3DMM albedo fitting problem. In this thesis, we demonstrated its effectiveness through empirical validation in Section 3.2.1 and all the network training examples in Chapter 5.

Semantic segmentation can serve as useful supervision for 3D reconstruction networks.

We demonstrated that a model-based monocular 3D face reconstruction network can be trained using semantic segmentation maps, as discussed in Section 5.4 and Section 5.3. This provides an alternative means of supervision when a target object possesses an appearance that is challenging to annotate with landmarks.

6.3 Limitations

In-network linear least-squares necessitate a linear model.

A notable limitation of in-network linear least-squares model fitting is its inability to handle non-linear objects. In this thesis, we selected a human face as target object domain due to the ease of obtaining a linear statistical model. However, in the domain of objects surrounding us, such models usually do not exist. This fact could constrain the applicability of this method.

Linear least-squares needs careful initialisation.

As demonstrated in Section 5.2.3 and Section 5.4.3, the linear least-squares layer requires careful initialisation to stabilise training. This implies that sufficient realistic synthetic data or adequate domain knowledge to preform pre-alignment is needed. This requirement could somewhat weaken our claim that we achieved self-supervised training without landmark supervision.

Adjusting the regularisation weight in linear least-squares is non trivial.

To mitigate overfitting of linear least-squares to noisy input data, adaptive weight adjustment is introduced in Section 5.2.2. In this method, a constant regularisation weight vector is updated to ensure the distributions of fitted coefficients are plausible. However, this method does not guarantee the plausibility of every fitted model. If the network generates highly implausible data, the fitted result also becomes implausible.

Calculation of the geometric Rényi divergence for a large image or 3D meshes is prohibitively expensive.

In our GRD calculation in Section 4.4.2, all possible combinations between a group of pixels and a group of vertices must be taken into account. This could require unrealistic computational cost when the number of pixels and/or vertices is large.

Inconsistency between semantic labels on the image and those on the model is not fully resolved.

We generate semantic segmentation labels on a 3DMM by automatic labelling and accumulation based on a pretrained semantic segmentation network (see Section 4.2). This label is always consistent, no matter what coefficients appear. On the other hand, semantic segmentation labels on natural images vary from image to image due to self-occlusion and limited coverage of a face in the 3DMM. This results in the inconsistency of labels between the image and the model. Minimising our GRD loss moves points so that exact matching will be achieved. Hence, inconsistency in labels could cause erroneous results. We tried to resolve this problem by a coarse alignment and update approach in Section 5.3.2. However, this is still imperfect, and misalignment occurs in some images.

6.4 Future work

Generalising in-network linear least-squares to non-linear objects

In this thesis, we only focused on a human face that can be well represented by a 3D morphable model, a purely linear statistical model. However, most objects in our surrounding environment are not as simple as a linear model. To leverage in-network linear least-squares model fitting for broader domains and applications, it is necessary to combine it with a non-linear model that consists of linear model elements and non-linear elements (e.g. correctives, joints, neural-network-based bases representation).

Application of in-network photometric linear least-squares to NeRF

Recently, as a 3D scene representation, the neural radiance field (NeRF) is attracting attention due to its explosive evolution. A typical problem with NeRF is its huge computation cost for training. We believe that in-network least-squares for lighting can contribute to reducing the computational cost of training variants of NeRF, which disentangle lighting and reflectance from observed images. The least-squares layer could remove the burden of explicitly estimating lighting by a network.

Application of the geometric Rényi divergence to broader 2D/3D alignment problems

The GRD on Gaussian mixtures is a long-range, non-saturating distance metric, which is suitable for 2D/3D model or image alignment. This measure has not yet been extensively studied in the field of computer vision. We believe GRD can be used as a substitute for keypoint-based distance metrics and is even suitable for the integration of multiple geometric cues including correspondence of keypoints, lines, and regions. In recent years, self-supervised learning for detection of such geometric features has significantly advanced. This trend makes this research direction more promising, as GRD can handle network outputs without applying global pooling or non-maximum suppression, and can benefit from the advancement of the field.

Lightweight calculation of the geometric Rényi divergence

To apply GRD to broader applications, reducing the computation cost of GRD is essential. To this end, we could explore the ways to approximate the distribution of points by a small number of Gaussian distributions, by merging neighbouring points and removing points whose effects on the final GRD value is negligible.

Combination of self-supervised semantic segmentation and backward rasterisation

What is disappointing in our backward rasterisation network, which reconstructs a 3D human face without a differentiable renderer, is that it requires semantic segmentation labels as additional supervision to train the network. Combining a self-supervised semantic segmentation approach with backward rasterisation is a possible research direction to establish a fully self-supervised monocular 3D reconstruction method without using a differentiable renderer.

List of References

[1] Psychological image collection at stirling (pics).

[2] Riza Alp Guler, George Trigeorgis, Epameinondas Antonakos, Patrick Snape, Stefanos Zafeiriou, and Iasonas Kokkinos. Densereg: Fully convolutional dense shape regression in-the-wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 6799–6808, 2017.

[3] J Antolín, PA Bouvrie, and JC Angulo. Geometric rényi divergence: A comparative measure with applications to atomic densities. *Physical Review A*, 84(3):032504, 2011.

[4] Anil Bas and William A. P. Smith. What does 2D geometric information really tell us about 3D face shape? *International Journal of Computer Vision*, 127(10):1455–1473, 2019.

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

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

[7] Xudong Cao, Yichen Wei, Fang Wen, and Jian Sun. Face alignment by explicit shape regression. *International Journal of Computer Vision*, 107(2):177–190, 2014.

[8] Bindita Chaudhuri, Noranart Vesdapunt, Linda Shapiro, and Baoyuan Wang. Personalized face modeling for improved face reconstruction and motion retargeting. In *IEEE European Conference on Computer Vision (ECCV)*, 2020.

[9] Ricky TQ Chen, Yulia Rubanova, Jesse Bettencourt, and David K Duvenaud. Neural ordinary differential equations. *Advances in neural information processing systems*, 31, 2018.

[10] Daniel Crispell and Maxim Bazik. Pix2face: Direct 3d face model estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2512–2518, 2017.

[11] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3d face reconstruction with weakly-supervised learning: From single image to image set. In *IEEE Computer Vision and Pattern Recognition Workshops*, 2019.

[12] Yu Deng, Jiaolong Yang, Sicheng Xu, Dong Chen, Yunde Jia, and Xin Tong. Accurate 3D face reconstruction with weakly-supervised learning: From single image to image set. In *Computer Vision and Pattern Recognition Workshops (CVPR-W)*, pages 285–295, 2019.

[13] Abdallah Dib, Cedric Thebault, Junghyun Ahn, Philippe-Henri Gosselin, Christian Theobalt, and Louis Chevallier. Towards high fidelity monocular face reconstruction with rich reflectance using self-supervised learning and ray tracing. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2021.

[14] Ricardo Fabbri, Luciano Da F Costa, Julio C Torelli, and Odemir M Bruno. 2d euclidean distance transform algorithms: A comparative survey. *ACM Computing Surveys (CSUR)*, 40(1):1–44, 2008.

[15] Yao Feng, Haiwen Feng, Michael J. Black, and Timo Bolkart. Learning an animatable detailed 3D face model from in-the-wild images. *ACM Transactions on Graphics (ToG), Proc. SIGGRAPH*, 40(4):88:1–88:13, 2021.

[16] Yao Feng, Fan Wu, Xiaohu Shao, Yanfeng Wang, and Xi Zhou. Joint 3d face reconstruction and dense alignment with position map regression network. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 534–551, 2018.

[17] Zhen-Hua Feng, Patrik Huber, Josef Kittler, Peter Hancock, Xiao-Jun Wu, Qijun Zhao, Paul Koppen, and Matthias Rätsch. Evaluation of dense 3d reconstruction from 2d face images in the wild. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 780–786. IEEE, 2018.

[18] Daniele Ferdani, Bruno Fanini, Maria Claudia Piccioli, Fabiana Carboni, and Paolo Vigliarolo. 3d reconstruction and validation of historical background for immersive vr applications and games: The case study of the forum of augustus in rome. *Journal of Cultural Heritage*, 43:129–143, 2020.

[19] Michael S Floater. Parametrization and smooth approximation of surface triangulations. *Computer aided geometric design*, 14(3):231–250, 1997.

[20] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multiview stereopsis. *IEEE transactions on pattern analysis and machine intelligence*, 32(8):1362–1376, 2009.

[21] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T. Freeman. Unsupervised training for 3D morphable model regression. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[22] Kyle Genova, Forrester Cole, Aaron Maschinot, Aaron Sarna, Daniel Vlasic, and William T Freeman. Unsupervised training for 3d morphable model regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8377–8386, 2018.

[23] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *International Conference on Automatic Face & Gesture Recognition (FG)*, pages 75–82. IEEE, 2018.

[24] Gene H Golub and Victor Pereyra. The differentiation of pseudo-inverses and nonlinear least squares problems whose variables separate. *SIAM Journal on numerical analysis*, 10(2):413–432, 1973.

[25] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3d dense face alignment. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2020.

[26] Jianzhu Guo, Xiangyu Zhu, Yang Yang, Fan Yang, Zhen Lei, and Stan Z Li. Towards fast, accurate and stable 3D dense face alignment. In *European Conference on Computer Vision (ECCV)*, pages 152–168, 2020.

[27] Richard Hartley and Andrew Zisserman. *Multiple view geometry in computer vision*. Cambridge university press, 2003.

[28] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034, 2015.

[29] Heiko Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 807–814. IEEE, 2005.

[30] Aaron S Jackson, Adrian Bulat, Vasileios Argyriou, and Georgios Tzimiropoulos. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *International Conference on Computer Vision*, 2017.

[31] Bing Jian and Baba C Vemuri. Robust point set registration using gaussian mixture models. *IEEE transactions on pattern analysis and machine intelligence*, 33(8):1633–1645, 2010.

[32] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *ECCV*, 2018.

[33] Masaya Kaneko, Ken Sakurada, and Kiyoharu Aizawa. Meshdepth: Disconnected mesh-based deep depth prediction, 2019.

[34] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[35] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3D mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3907–3916, 2018.

[36] Hyeongwoo Kim, Michael Zollöfer, Ayush Tewari, Justus Thies, Christian Richardt, and Theobalt Christian. InverseFaceNet: Deep Single-Shot Inverse Face Rendering From A Single Image. In *Proceedings of Computer Vision and Pattern Recognition (CVPR 2018)*, 2018.

[37] Davis E. King. Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, 10:1755–1758, 2009.

[38] Nikos Kolotouros, Georgios Pavlakos, Michael J Black, and Kostas Daniilidis. Learning to reconstruct 3d human pose and shape via model-fitting in the loop. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 2252–2261, 2019.

[39] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In F. Pereira, C. J. C. Burges, L. Bottou, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[40] Cheng-Han Lee, Ziwei Liu, Lingyun Wu, and Ping Luo. Maskgan: Towards diverse and interactive facial image manipulation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] Chunlu Li, Andreas Morel-Forster, Thomas Vetter, Bernhard Egger, and Adam Kortylewski. To fit or not to fit: Model-based face reconstruction and occlusion segmentation from weak supervision, 2022.

[42] Xueting Li, Sifei Liu, Kihwan Kim, Shalini De Mello, Varun Jampani, Ming-Hsuan Yang, and Jan Kautz. Self-supervised single-view 3D reconstruction via semantic consistency. In *Proc. ECCV*, 2020.

[43] Chen-Hsuan Lin, Oliver Wang, Bryan C Russell, Eli Shechtman, Vladimir G Kim, Matthew Fisher, and Simon Lucey. Photometric mesh optimization for video-aligned 3d object reconstruction. *arXiv preprint arXiv:1903.08642*, 2019.

[44] Rosanne Liu, Joel Lehman, Piero Molino, Felipe Petroski Such, Eric Frank, Alex Sergeev, and Jason Yosinski. An intriguing failing of convolutional neural networks and the CoordConv solution. In *Advances in Neural Information Processing Systems*, pages 9605–9616, 2018.

[45] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3D reasoning. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 7708–7717, 2019.

[46] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of International Conference on Computer Vision (ICCV)*, 2015.

[47] Matthew M Loper and Michael J Black. Opendr: An approximate differentiable renderer. In *European Conference on Computer Vision*, pages 154–169. Springer, 2014.

[48] Sebastian Lunz, Yingzhen Li, Andrew Fitzgibbon, and Nate Kushman. Inverse graphics GAN. In *NeurIPS-W*, 2020.

[49] Peter M. Roth Martin Koestinger, Paul Wohlhart and Horst Bischof. Annotated Facial Landmarks in the Wild: A Large-scale, Real-world Database for Facial Landmark Localization. In *Proc. First IEEE International Workshop on Benchmarking Facial Image Analysis Technologies*, 2011.

[50] Ben Mildenhall, Pratul P. Srinivasan, Matthew Tancik, Jonathan T. Barron, Ravi Ramamoorthi, and Ren Ng. NeRF: Representing scenes as neural radiance fields for view synthesis. In *ECCV*, 2020.

[51] Baback Moghaddam, Jinho Lee, Hanspeter Pfister, and Raghu Machiraju. Model-based 3D face capture with shape-from-silhouettes. In *IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG)*, pages 20–27. IEEE, 2003.

[52] Claus Müller. Spherical harmonics. *Lecture Notes in Mathematics*, 1966.

[53] Michael Niemeyer, Lars Mescheder, Michael Oechsle, and Andreas Geiger. Differentiable volumetric rendering: Learning implicit 3d representations without 3d supervision. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 3504–3515, 2020.

[54] Felix Petersen, Amit H Bermano, Oliver Deussen, and Daniel Cohen-Or. Pix2vex: Image-to-geometry reconstruction using a smooth differentiable renderer. *arXiv preprint arXiv:1903.11149*, 2019.

[55] Sudeep Pillai, Rares Ambrus, and Adrien Gaidon. Superdepth: Self-supervised, super-resolved monocular depth estimation. *arXiv preprint arXiv:1810.01849*, 2018.

[56] Shaoqing Ren, Xudong Cao, Yichen Wei, and Jian Sun. Face alignment at 3000 fps via regressing local binary features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1685–1692, 2014.

[57] Elad Richardson, Matan Sela, and Ron Kimmel. 3D face reconstruction by learning from synthetic data. In *Proc. 3DV*, pages 460–469, 2016.

[58] Sami Romdhani and Thomas Vetter. Estimating 3d shape and texture using pixel intensity, edges, specular highlights, texture constraints and a prior. In *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, volume 2, pages 986–993. IEEE, 2005.

[59] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[60] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael Black. Learning to regress 3d face shape and expression from an image without 3d supervision. In *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[61] Matan Sela, Elad Richardson, and Ron Kimmel. Unrestricted facial geometry reconstruction using image-to-image translation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1576–1585, 2017.

[62] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D Castillo, and David W Jacobs. SfSNet: Learning shape, reflectance and illuminance of faces 'in the wild'. In *Proc. ECCV*, 2018.

[63] Jiaxiang Shang, Tianwei Shen, Shiwei Li, Lei Zhou, Mingmin Zhen, Tian Fang, and Long Quan. Self-supervised monocular 3D face reconstruction by occlusion-aware multi-view geometry consistency. In *European Conference on Computer Vision (ECCV)*, volume 12360, pages 53–70, 2020.

[64] Ayush Tewari, Michael Zollhöfer, Pablo Garrido, Florian Bernard, Hyeongwoo Kim, Patrick Pérez, and Christian Theobalt. Self-supervised multi-level face model learning for monocular reconstruction at over 250 hz. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2018.

[65] Ayush Tewari, Michael Zollhofer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Christian Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 1274–1283, 2017.

[66] Ayush Tewari, Michael Zollöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Perez, and Theobalt Christian. MoFA: Model-based Deep Convolutional Face Autoencoder for Unsupervised Monocular Reconstruction. In *The IEEE International Conference on Computer Vision (ICCV)*, 2017.

[67] Sebastian Thrun et al. Robotic mapping: A survey. 2002.

[68] Wan Tian, Feng Liu, and Qijun Zhao. Landmark-based 3d face reconstruction from an arbitrary number of unconstrained images. In *2018 13th IEEE International Conference on Automatic Face & Gesture Recognition (FG 2018)*, pages 774–779. IEEE, 2018.

[69] Anh Tuan Tran, Tal Hassner, Iacopo Masi, and Gerard Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proc. CVPR*, pages 5163–5172, 2017.

[70] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *IEEE Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, UT, 2018.

[71] George Trigeorgis, Patrick Snape, Mihalis A Nicolaou, Epameinondas Antonakos, and Stefanos Zafeiriou. Mnemonic descent method: A recurrent process applied for end-to-end face alignment. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4177–4187, 2016.

[72] Floris van Beers, Arvid Lindström, Emmanuel Okafor, and Marco A Wiering. Deep neural networks with intersection over union loss for binary image segmentation. In *ICPRAM*, pages 438–445, 2019.

[73] Wouter Van Gansbeke, Bert De Brabandere, Davy Neven, Marc Proesmans, and Luc Van Gool. End-to-end lane detection through differentiable least-squares fitting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pages 0–0, 2019.

[74] Fei Wang, Tanveer Syeda-Mahmood, Baba C Vemuri, David Beymer, and Anand Rangarajan. Closed-form jensen-renyi divergence for mixture of gaussians and applications to group-wise shape registration. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 648–655. Springer, 2009.

[75] T Wiberg. Computation of principal components when data are missing. In *Proc. of Second Symp. Computational Statistics*, pages 229–236, 1976.

[76] Erroll Wood, Tadas Baltrusaitis, Charlie Hewitt, Matthew Johnson, Jingjing Shen, Nikola Milosavljevic, Daniel Wilde, Stephan Garbin, Chirag Raman, Jamie Shotton, Toby Sharp, Ivan Stojiljkovic, Tom Cashman, and Julien Valentin. 3D face reconstruction with dense landmarks. In *European Conference on Computer Vision (ECCV)*, pages 160–177, 2022.

[77] Erroll Wood, Tadas Baltrušaitis, Charlie Hewitt, Sebastian Dziadzio, Matthew Johnson, Virginia Estellers, Thomas J. Cashman, and Jamie Shotton. Fake it till you make it: Face analysis in the wild using synthetic data alone, 2021.

[78] Cho-Ying Wu, Qiangeng Xu, and Ulrich Neumann. Synergy between 3DMM and 3D landmarks for accurate 3D facial geometry. In *International Conference on 3D Vision (3DV)*, pages 453–463, 2021.

[79] Kohei Yamashita, Shohei Nobuhara, and Ko Nishino. 3D-GMNet: Single-view 3D shape recovery as a gaussian mixture. In *Proc. BMVC*, 2020.

[80] Junjie Yan, Zhen Lei, Dong Yi, and Stan Z. Li. Learn to combine multiple hypotheses for accurate face alignment. *2013 IEEE International Conference on Computer Vision Workshops*, pages 392–396, 2013.

[81] Ronald Yu, Shunsuke Saito, Haoxiang Li, Duygu Ceylan, and Hao Li. Learning dense facial correspondences in unconstrained images. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 4723–4732, 2017.

[82] Matthew D. Zeiler. Adadelta: An adaptive learning rate method. *CoRR*, abs/1212.5701, 2012.

[83] Yang Zhang, Peihua Lv, Xiaobo Lu, and Jun Li. Face detection and alignment method for driver on highroad based on improved multi-task cascaded convolutional networks. *Multimedia Tools and Applications*, 78:26661–26679, 2019.

[84] Yi Zhou, Connelly Barnes, Jingwan Lu, Jimei Yang, and Hao Li. On the continuity of rotation representations in neural networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5745–5753, 2019.

[85] Shizhan Zhu, Cheng Li, Chen Change Loy, and Xiaoou Tang. Face alignment by coarse-to-fine shape searching. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4998–5006, 2015.

[86] Wenbin Zhu, HsiangTao Wu, Zeyu Chen, Noranart Vesdapunt, and Baoyuan Wang. Reda:reinforced differentiable attribute for 3D face reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[87] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z Li. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 146–155, 2016.

[88] Xiangyu Zhu, Xiaoming Liu, Zhen Lei, and Stan Z Li. Face alignment in full pose range: A 3D total solution. *IEEE transactions on pattern analysis and machine intelligence*, 41(1):78–92, 2017.

[89] Wojciech Zielonka, Timo Bolkart, and Justus Thies. Towards metrical reconstruction of human faces. In *European Conference on Computer Vision (ECCV)*, pages 250–269, 2022.