Shape from semantic segmentation via the geometric Rényi divergence

Tatsuro Koizumi and William A. P. Smith
University of York
York, UK
{tk856,william.smith}@york.ac.uk

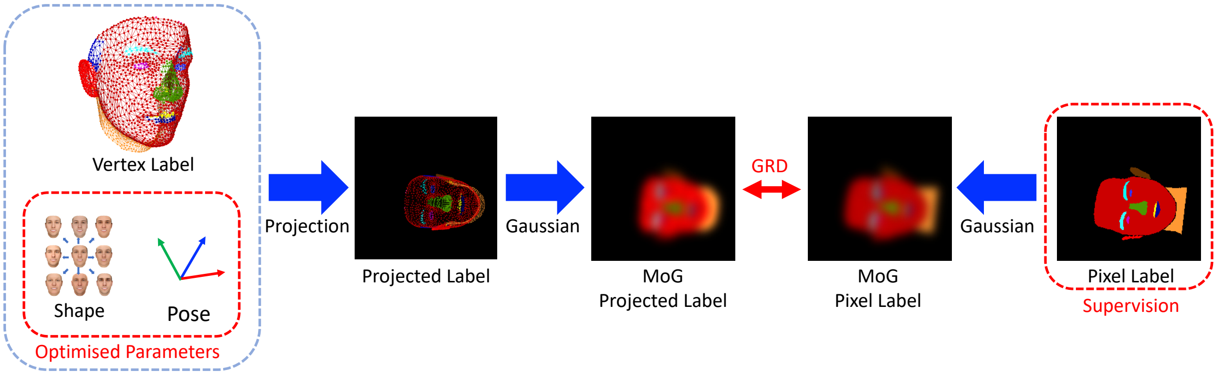


Figure 1: To extract a supervisory signal from a given pixel-wise semantic segmentation, we propose a loss that is differentiable with respect to pose and shape parameters. Given fixed per-vertex semantic labels and pose and shape estimates (col. 1), we project the labelled vertices to 2D. We represent 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.

Abstract

In this paper, we show how to estimate shape (restricted to a single object class via a 3D morphable model) using solely a semantic segmentation of a single 2D image. We propose a novel loss function based on a probabilistic, vertex-wise projection of the 3D model to the image plane. We represent both these projections and pixel labels 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. We propose both classical, direct optimisation of this loss ("analysis-by-synthesis") and its use for training a parameter regression CNN. We show significant advantages over existing segmentation losses used in state-of-the-art differentiable renderers Soft Rasterizer and Neural Mesh Renderer.

1. Introduction

It is widely known that the silhouette of an object provides an important cue for 3D shape estimation and the theory of multiview shape-from-silhouette is well understood [16]. Restricting consideration to a single object class allows the problem to be tackled using model-based methods in which the solution is constrained by a 3D morphable model (3DMM) [8]. Such model-based approaches have been used for reconstruction of faces from multiview silhouettes [21] and even for reconstruction from single silhouettes while simultaneously learning a morphable model for a new object class [6]. A silhouette can be viewed as a binary semantic segmentation of the image into foreground and background. A more fine-grained segmentation into semantic object parts presumably conveys richer information about 3D shape, including occluding and internal contours and the layout of internal features. This is particularly interesting because of the recent great successes in learning automatic semantic segmentation using fully convolutional networks [2]. To the best of our knowledge, the reconstruction of a 3D model using only semantic segmentation information in a single image has not previously been studied and we coin this problem *shape from semantic segmentation*. In contrast to landmarks, which are sparse but have well defined one-to-one correspondence to the reference model, pixel-wise semantic segmentation information is dense but only provides one-to-many correspondence (a pixel may correspond to any vertex within the semantic model part).

In this paper we consider the problem of shape from semantic segmentation using a 3DMM, specifically a human face model. The crucial ingredient is a measure of the discrepancy between an observed and predicted semantic segmentation in image space that is both differentiable and has a wide basin of convergence. This enables the measure to be used as a loss in gradient-based direct optimisation or in training of a parameter regression CNN. To this end, we propose to use geometric Rényi divergence and show that this has benefits over other soft segmentation difference measures. In particular, it is able to converge to a good solution from an initial estimate that is far from ground truth in both pose and shape. The resulting shape estimation improves upon previous face 3DMM fitting approaches by avoiding conservative underfitting, ensuring the model expands to fit boundary features such as ears and neck and by providing robustness to large image space transformations of the input. In addition, we provide an efficient closed form solution for computing the GRD so that it can be incorporated into the training of a parameter regression CNN.

1.1. Related work

One-to-many 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 [13] use the L2 distance between two mixture of Gaussians (MoG) for point cloud registration. Wang *et al*. [28] use closed-form Jensen Rényi divergence for MoG for group-wise point cloud registration. Yamashita *et al*. [29] 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.

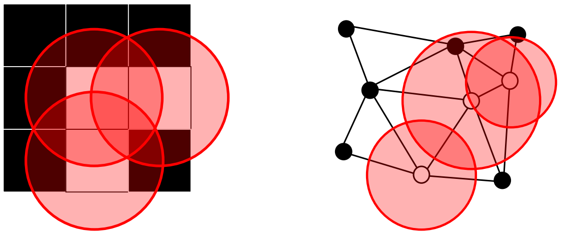
3DMM fitting and shape-from-geometric features The earliest work on 3DMM fitting used landmark distance as a sparse objective function for approximate initialisation and within an analysis-by-synthesis framework [4]. Subsequently, Romdhani and Vetter [23] used landmarks and occluding contours within a multi-feature fitting approach. Bas and Smith [3] 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 [24] 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*. [21] used a binary silhouette loss across multiple images. Since this loss is discontinuous, they use the derivative-free Nelder–Mead optimisation method. In very recent, ambitious work, Li *et al*. [18] 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.

Differentiable rendering The recent topic of differentiable rendering has emerged from a desire to include explicit rendering capability within a neural network such that training can exploit 3D rendering as a supervision signal. The fundamental challenge is that rasterisation of a continuous 3D object onto a discrete pixel grid is fundamentally not differentiable. Hence, approximations are used that provide useful smooth gradients. Neural 3D Mesh Renderer (NMR) [14] extrapolates a gradient outside triangles based on linear interpolation of the derivative across a triangle edge. Soft Rasterizer (SoftRas) [19] computes a soft (i.e. blurred) rasterisation of each triangle in a mesh. Two very recent 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 [32, 7]. They do so simply by rasterising the semantic labels on the mesh using a differentiable renderer, [32] using a variant of SoftRas [19] and [7] using TF Mesh Renderer [10]. 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 [32, 7, 18], has considered the problem of estimating shape using only semantic segmentation information.

2. Overview

A pixel-wise semantic segmentation of a face image is a discrete representation. Similarly, the rasterisation of a 3D face model 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 segmentations (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. Figure 1 shows an overview of our proposed loss. Given estimates of 3DMM shape parameters and the pose (camera parameters), we project the 3D vertices of the 3DMM to 2D. The vertices themselves have fixed semantic labels (which we later show how to automat-



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

Figure 2: Representing pixels (a) and vertices (b) of a given semantic class (shown in white) as mixtures of Gaussians.

ically infer from a given labelled 2D image dataset). 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 contribution 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. For practical application, we propose both direct optimisation of the loss given a single input segmentation ("analysis-by-synthesis") and use it for training a parameter regression CNN.

3. GRD-based semantic segmentation loss

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

3.1. 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 2). Hence we represent softened semantic label P on image coordinate $\mathbf{z}$ with Mixture of Gaussian:

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

where $\mathbf{x_i}$ is the centre of ith Gaussian kernel (corresponding to either a pixel centre or projected vertex position), and σ is the corresponding standard deviation of Gaussian function. α_i is weight of ith Gaussian kernel, and is allocated based on corresponding area on the image. For input pixel-wise semantic labels, α_i is set to 1 so that it represents the area of one pixel. For vertices, α_i is set to the average of the projected area of the neighboring faces.

3.2. Geometric Rényi divergence

We employ closed-form geometric Rényi divergence (GRD) as a cohesive measure between two mixture of Gaussian (MoG) distributions, which represent ground truth and projected semantic labels. Wang *et al.* [28] proposed closed-form Jensen-Rényi divergence (JRD) for MoG and applications to group-wise shape registration. JRD for K groups is defined as:

$$\begin{aligned} &JRD_{\pi,q}\left(P_1, P_2, \ldots, P_K\right) = \\ &\qquad H_q\left(\sum_{i=1}^{K} \pi_i P_i\right) - \sum_{i=1}^{K} \pi_i H_q\left(P_i\right), \qquad (2) \end{aligned}$$

where H_q is 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. qth order Rényi entropy is defined as:

$$H_q(P) = \frac{1}{1-q} \log\left(\int P\left(\mathbf{z}\right)^q d\mathbf{z}\right). \qquad (3)$$

When $q \to 1$, (3) is Shannon entropy, and (2) is Jensen-Shannon divergence. Wang *et al.* [28] 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, order 2 JRD is not a preferable measure for alignment of two distributions. To resolve the negativity issue, Antolín *et al.* [1] proposed geometric Rényi divergence (GRD):

$$\begin{aligned} &GRD_{\pi,q}\left(P_1, P_2, \ldots, P_K\right) = \\ &\qquad (q-1)\left[H_q\left(\prod_{i=1}^{K} P_i^{\pi_i}\right) - \sum_{i=1}^{K} \pi_i H_q\left(P_i\right)\right]. \qquad (4) \end{aligned}$$

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.

3.3. Closed-form 2nd order GRD between two MoGs

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

$$\begin{aligned} &GRD_{1/2,2}\left(P_x, P_y\right) = \\ &\qquad H_2\left(\sqrt{P_x P_y}\right) - \frac{1}{2}\left(H_2\left(P_x\right) + H_2\left(P_y\right)\right) \qquad (5) \end{aligned}$$

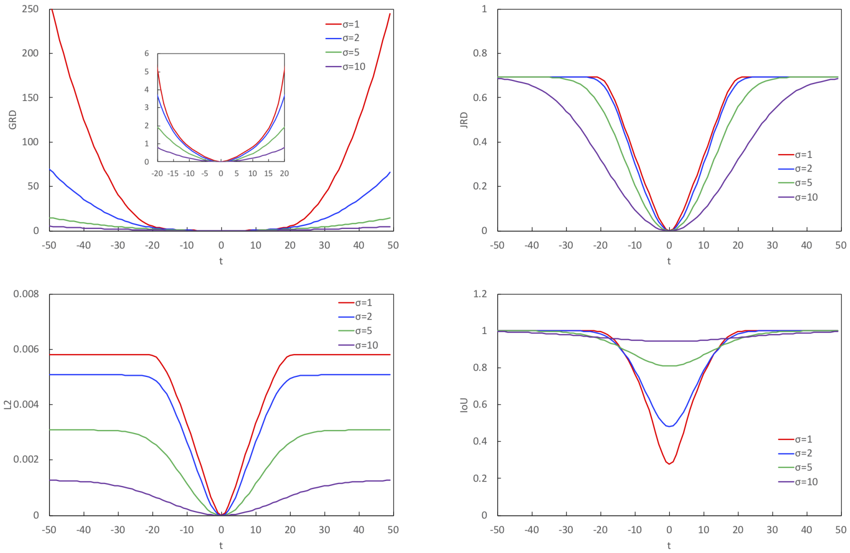


Figure 3: Loss landscape of GRD (top-left), JRD (top-right), L2 (bottom-left), and IoU (bottom-right) with respect to t pixel horizontal translation.

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

$$
\begin{aligned}
H_2\left(\sqrt{P_x P_y}\right) &= \int P_x(\mathbf{z}) P_y(\mathbf{z}) d\mathbf{z} \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j \int G(\mathbf{z}-\mathbf{x}_i,\sigma^2\mathbf{I})G(\mathbf{z}-\mathbf{y}_j,\sigma^2\mathbf{I})d\mathbf{z}\right] \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j G(\mathbf{x}_i-\mathbf{y}_j,2\sigma^2\mathbf{I})\right], \qquad (6)
\end{aligned}
$$

and

$$
\begin{aligned}
H_2\left(P_x\right) &= \int P_x(\mathbf{z})^2 d\mathbf{z} \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{M}\alpha_i\alpha_j \int G(\mathbf{z}-\mathbf{x}_i,\sigma^2\mathbf{I})G(\mathbf{z}-\mathbf{x}_j,\sigma^2\mathbf{I})d\mathbf{z}\right] \\
&= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\alpha_j G(\mathbf{x}_i-\mathbf{x}_j,2\sigma^2\mathbf{I})\right]. \qquad (7)
\end{aligned}
$$

From (5), (6) and (7), we obtain closed-form divergence:

$$
\begin{aligned}
GRD_{1/2,2}\left(P_x,P_y\right) &= -\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\beta_j G(\mathbf{x}_i-\mathbf{y}_j,2\sigma^2\mathbf{I})\right] \\
&+\frac{1}{2}\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\alpha_i\alpha_j G(\mathbf{x}_i-\mathbf{x}_j,2\sigma^2\mathbf{I})\right] \\
&+\frac{1}{2}\log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\beta_i\beta_j G(\mathbf{y}_i-\mathbf{y}_j,2\sigma^2\mathbf{I})\right]. \qquad (8)
\end{aligned}
$$

3.4. Numerical stability

The GRD becomes numerically unstable when the difference between two MoG distributions is large. That is because all the exponential functions in (6) output zero value for large $\|\mathbf{x_i}-\mathbf{y_i}\|_2^2$. To avoid this issue, in practice we implement (6) as:

$$
\begin{aligned}
H_2\left(\sqrt{P_x P_y}\right) &= \log\left[\sum_{i=1}^{M}\sum_{j=1}^{N}\exp\left(-\mathbf{e}_{ij}+\min\{\mathbf{e}_{ij}\}\right)\right] \\
&- \min\{\mathbf{e}_{ij}\} + \log\left(\frac{\alpha_i}{2\pi\sigma^2}\right), \qquad (9)
\end{aligned}
$$

where $\mathbf{e}_{ij} = -\frac{(\mathbf{x_i}-\mathbf{x_j})^T(\mathbf{x_i}-\mathbf{x_j})}{2\sigma^2} - \log(\alpha_i\alpha_j)$.

3.5. Loss landscape and comparison

We now illustrate the attractive properties of the GRD using a toy example. We draw a circle with 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 3, a horizontal translation of t pixels is applied, and in Figure 4, magnification by factor s is applied. 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 [27] and [19], we define a soft IoU loss L_{IoU} for two distributions P_x and P_y as:

$$
L_{IoU} = 1 - \frac{\|P_x \odot P_y\|_1}{\|P_x + P_y - P_x \odot P_y\|_1} \qquad (10)
$$

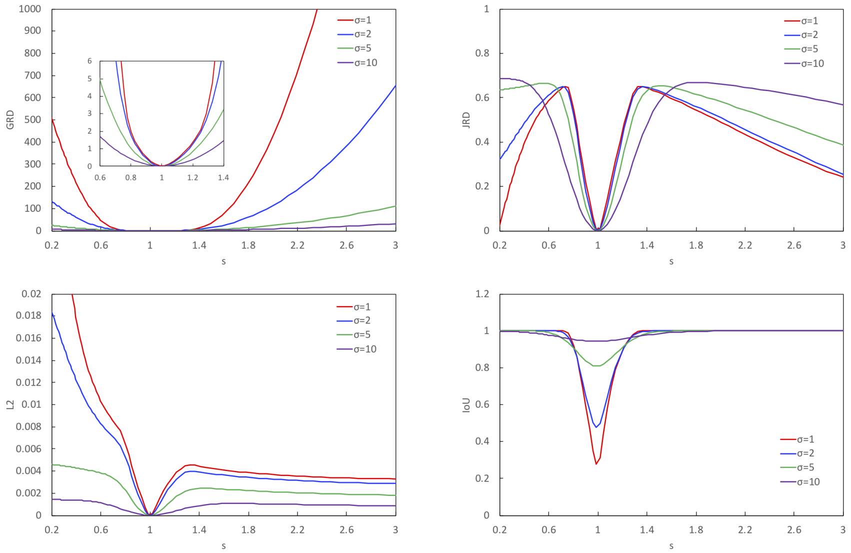


Figure 4: Loss landscape of GRD (top-left), JRD (top-right), L2 (bottom-left), and IoU (bottom-right) with respect to magnification by s.

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 flat gradient. These examples indicate that GRD is more suitable as a measure for region alignment than other metrics.

4. GRD loss based fitting/supervision

We now show how to integrate our GRD-based semantic segmentation loss into either analysis-by-synthesis or CNN-based 3DMM fitting regimes.

4.1. 3DMM and image formation model

We represent 3D face models based on a 3DMM:

$$\mathbf{v}_j = \sum_{i=1}^{N_s+N_e} \alpha_i \mathbf{b}_{ij} + \mathbf{a}_j, \quad \mathbf{r}_j = \sum_{i=1}^{N_r} \beta_i \mathbf{c}_{ij} + \mathbf{d}_j \tag{11}$$

where $\mathbf{v}_j$ is the 3D position and $\mathbf{r}_j$ is the RGB reflectance of jth vertex respectively. $\mathbf{b}_{ij}$ is the ith linear basis of the vertex position and $\mathbf{a}_j$ is its mean. In the same manner, $\mathbf{c}_{ij}$ is the ith linear basis of the vertex reflectance and $\mathbf{d}_j$ is its mean. 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 [11] 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:

$$\lambda_j \begin{bmatrix} \acute{\mathbf{x}}_j \\ 1 \end{bmatrix} = \mathbf{A}(\mathbf{R}\mathbf{v}_j + \mathbf{t}) \tag{12}$$

where $\acute{\mathbf{x}}_j$ is jth projected vertex position, $\mathbf{A}$ 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:

$$\mathbf{i}_j = \mathbf{r}_j \sum_{k=1}^{27} \gamma_k \mathbf{H}_k(\mathbf{n}_j), \tag{13}$$

where $\mathbf{i}_j$ is jth shaded vertex colour, $\mathbf{H}_k$ 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.

4.2. Automatic labelling of model vertices

In order to apply our GRD loss, we require semantic labels for each vertex in the 3D face model that are consistent with ground truth semantic labels provided on images. In order to transfer the semantics of the image labels to the model automatically, we propose the following process. First, we pre-train an image-to-image face parsing network using the given labelled image dataset. Specifically we use CelebAMask-HQ [17]. Next, we randomly sample

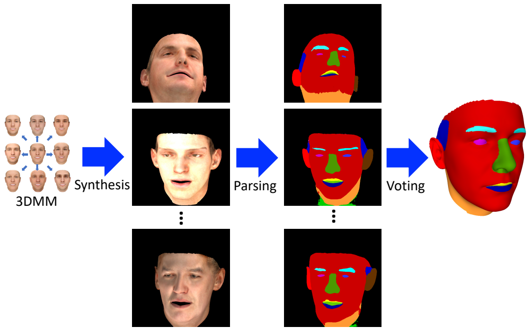


Figure 5: Automatic semantic labelling of model vertices.

face models from the 3DMM, render to images and pass them through the face parsing network. For a given image, each visible vertex is assigned the semantic label of highest probability from the face parsing network output. Then, across all images we take a majority vote to assign a single semantic label to each vertex in the model. 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 5.

4.3. Analysis-by-synthesis

We use 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 1. We place a Gaussian kernel on each projected vertex $\acute{\mathbf{x}}_j$ calculated from (12), and obtain softened semantic label $\acute{P}_l$ of lth label on image coordinate $\mathbf{z}$:

$$\acute{P}_l(\mathbf{z}) = \frac{\sum_{j=1}^{N_v} \acute{\lambda}_{lj} v_j G\left(\mathbf{z} - \acute{\mathbf{x}}_j, \sigma^2 \mathbf{I}\right)}{\sum_{j=1}^{N_v} \acute{\lambda}_{lj} v_j}, \tag{14}$$

where N_v represents the number of vertices, $\acute{\lambda}_{lj}$ represents lth label on jth vertex, which returns 1 if a vertex belongs to the label and 0 otherwise, v_j represents average area of three neighboring faces of jth 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 softened semantic label $\hat{P}_l$ of lth label on image coordinate $\mathbf{z}$:

$$\hat{P}(\mathbf{z}) = \frac{\sum_{v=0}^{N_H-1} \sum_{u=0}^{N_W-1} \hat{\lambda}_l(u, v) G\left(\mathbf{z} - [u, v]^T, \sigma^2 \mathbf{I}\right)}{\sum_{v=0}^{N_H-1} \sum_{u=0}^{N_W-1} \hat{\lambda}_l(u, v)} \tag{15}$$

where $\hat{\lambda}_l(u, v)$ represents lth label on image coordinate $[u, v]$, N_W is a number of horizontal pixels, and N_H is a number of vertical pixels.

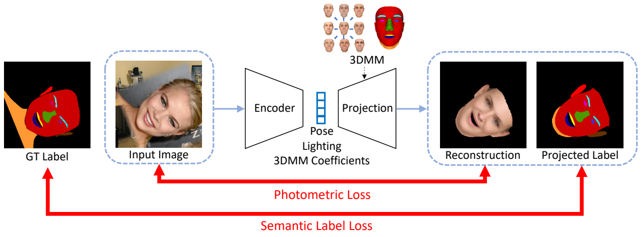


Figure 6: Parameter regression CNN architecture with semantic segmentation supervision.

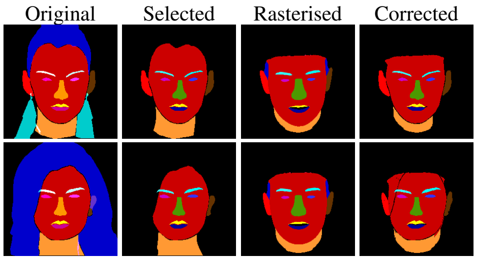


Figure 7: Label correction based on rasterised semantic labels generated by a provisional network.

We calculate GRD for each label based on (8) and minimise average GRD of all the labels while optimising all shape and camera parameters.

4.4. CNN-based regression

Our semantic label loss can be combined with a CNN-based regression network. Figure 6 shows a network for 3D face reconstruction based on semantic label loss. This can be viewed as a variant of MoFA [25] with additional semantic segmentation supervision. An encoder network predicts pose parameters and 3DMM coefficients. Semantic label loss is calculated as it is for analysis-by-synthesis application. To reconstruct colour information, we also estimate lighting and 3DMM reflectance coefficients, and minimise L2 norm of the difference of the colour between input image and shaded vertices based on (13).

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

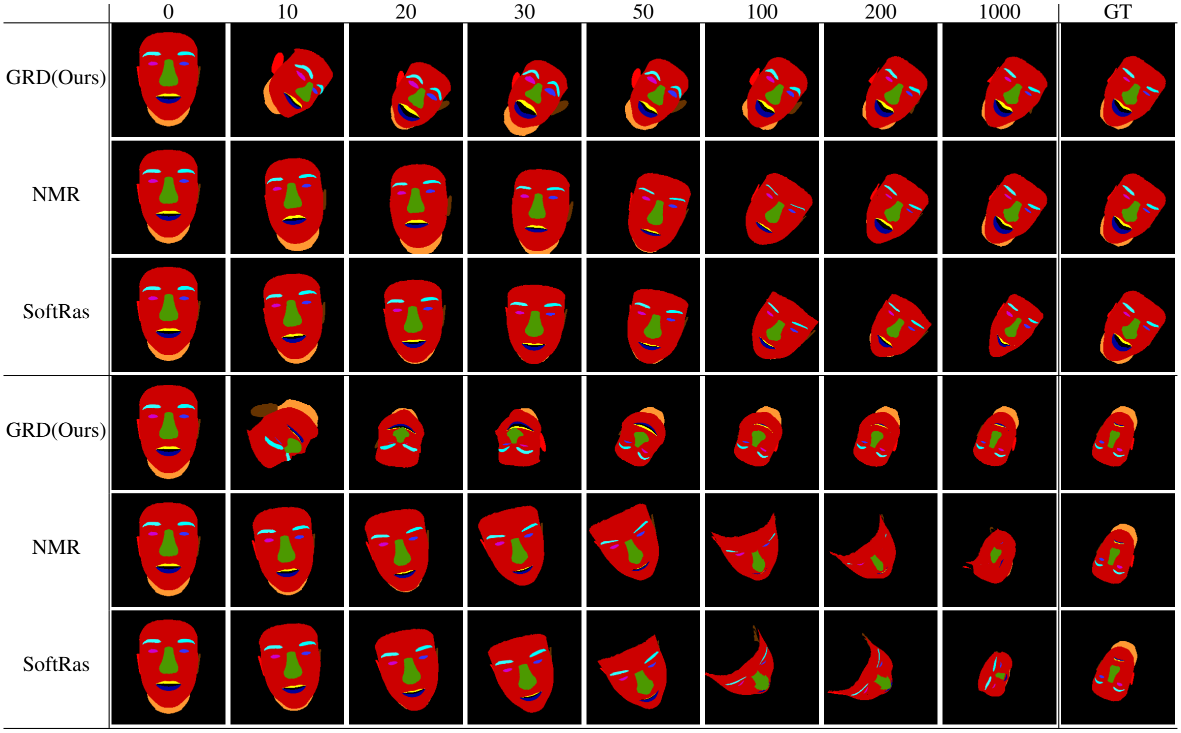


Figure 8: Convergence of direct optimisation of our GRD, NMR [14], and SoftRas [19] segmentation losses. Upper rows show an easy case, lower rows a challenging one. Target ground truth labels are shown in the final column.

	GRD (Ours)	NMR [14]	SoftRas [19]
IoU mean	0.931	0.789	0.423
IoU std	0.013	0.150	0.124

Table 1: Direct optimisation results for semantic labels randomly synthesised from the BFM [11].

5. Experiments

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 Rendenderer (NMR) [14] and Soft Rasterizer (SoftRas) [19].

Synthetic pixel label images are generated by perturbing 3DMM coefficents, focal length, image centre, pose rotation, and pose translation. Pose rotation is parameterised by Euler angle. We directly optimise 299 dimensional 3DMM coefficients, and 9 dimensional camera parameters with respect to average GRD between projected MoG and pixel MoG among 11 labels. We employ Adam optimiser with learning rate 0.2 for GRD, and 0.01 for NMR and SoftRas. For GRD, we chose $\sigma = 5$ as a parameter of Gaussian kernel. In optimisation with NMR, we differentiably rasterise semantic labels as an 11 channel image, and compute L2 norm between a rasterised image and a ground truth pixel label image. In optimisation with SoftRas, we differentiably rasterise semantic labels as an 11 channel image. We also reasterise an object silhouette and multiply it to the semantic labels. L2 norm between a rasterised image and a ground truth pixel label image is employed as a loss function. We chose $\sigma = 10^{-3}$ and $\gamma = 10^{-3}$ for SoftRas parameters.

Figure 8 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 large rotation from the initial pose to the ground truth. In a successful case, both NMR and SofTRas converges 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 converges to a local minima. We also calculate mean and standard deviation of IoU between the ground truth and the rasterised semantic labels (Table 1). The result indicates our method successfully converges to the ground truth in all 16 cases, whereas NMR and SoftRas fails in some cases.

CNN-based parameter regression We now use our loss to train a network to reconstruct 3D faces from a single image and show qualitative results and landmark evaluation.

The network estimates 3DMM coefficents for both shape

	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 [22]	7.17	17.72	10.64	6.17	16.19	9.87
ESR [5]	5.58	12.07	7.33	4.38	11.72	8.04
CFSS [31]	4.68	12.51	9.49	3.44	13.02	10.08
MDM [26]	5.14	13.40	9.72	4.64	13.07	10.07
SDM [30]	4.67	9.19	6.10	3.56	9.37	7.23
3DDFA [33]	4.11	4.55	0.54	2.84	3.79	1.08
PRNet [9]	4.19	4.77	-	2.75	3.62	-
Guo [12]	3.98	4.43	-	2.63	3.51	-
Ours	3.98	10.14	5.99	4.97	10.49	5.64

Table 2: Normalized Mean Error on AFLW [20] and AFLW2000-3D [33] datasets.

and albedo, pose parameters, and lighting parameters. Pose parameters are represented by a 3D translation vector, a rotation matrix, and a parameter to express perspective effect. We use 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 vector. Rotation matrix is generated from the vectors in Gram-Schmidt process. We employ individual VGG19 networks to estimate 3DMM coefficients, lighting parameters, and pose parameters respectively.

We train our network using 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 1000 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 [15] 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 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 Adadelta optimiser to train the networks with learning rate 0.001 for the final training of lighting and pose and 0.01 for the rest of the training.

Figure 9 shows qualitative results of the reconstruction. Our method successfully reconstructs 3D face including ears under arbitrary 2D similarity transformation. We quantitatively evaluate our method based on landmarks (Table 2). We follow the evaluation protocol proposed in Zhu *et al.* [33] and compare our result with supervised facial landmark detection methods. Our network shows comparable result to landmark-based methods for modest pose angles.

6. Conclusion

We have presented the first method that uses closed-form GRD for spatial alignment of two MoG distributions based on gradient-based optimisation. Our segmentation loss shows preferable characteristics over alternative measures and state-of-the-art differentiable renderers on both direct optmisation and training of neural networks. Our approach has further potential of application to other computer vision tasks such as point cloud registration, image registration, and general 3D reconstruction. Especially, our approach is suitable for alignment based on soft landmarks, which predicts landmark position with uncertainty. Our loss could also be used for multiview silhouette fitting, extended to other object classes or combined with pixel-wise semantic segmentation for a self-supervised pipeline.

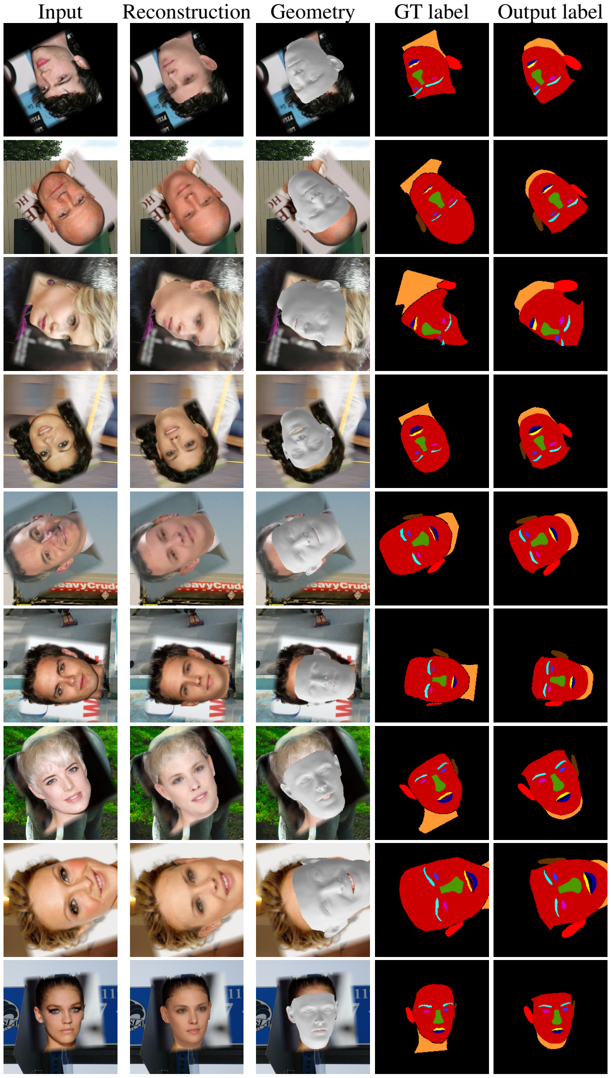


Figure 9: Reconstruction results based on CNN trained with semantic label loss and photometric loss.

Acknowledgements T.Koizumi was supported by Canon Inc. W. Smith was supported by a Royal Academy of Engineering/The Leverhulme Trust Senior Research Fellowship.

References

[1] 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.

[2] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. SegNet: A deep convolutional encoder-decoder architecture for image segmentation. *IEEE transactions on pattern analysis and machine intelligence*, 39(12):2481–2495, 2017.

[3] 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.

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

[5] 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.

[6] Thomas J Cashman and Andrew W Fitzgibbon. What shape are dolphins? building 3D morphable models from 2D images. *IEEE transactions on pattern analysis and machine intelligence*, 35(1):232–244, 2012.

[7] 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.

[8] Bernhard Egger, William A. P. Smith, Ayush Tewari, Stefanie Wuhrer, Michael Zollhoefer, Thabo Beeler, Florian Bernard, Timo Bolkart, Adam Kortylewski, Sami Romdhani, Christian Theobalt, Volker Blanz, and Thomas Vetter. 3d morphable face models – past, present and future. *ACM Transactions on Graphics*, 39(5), 2020.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] Aldo Laurentini. The visual hull concept for silhouette-based image understanding. *IEEE Transactions on pattern analysis and machine intelligence*, 16(2):150–162, 1994.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

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

[24] 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)*, pages 7763–7772, June 2019.

[25] 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.

[26] 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 Con-*

ference on Computer Vision and Pattern Recognition, pages 4177–4187, 2016.

[27] 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.

[28] 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.

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

[30] 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.

[31] 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.

[32] 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.

[33] 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.