Unsupervised Learning of Probably Symmetric Deformable 3D Objects from Images in the Wild

Shangzhe Wu Christian Rupprecht Andrea Vedaldi

Visual Geometry Group, University of Oxford

{szwu, chrisr, vedaldi}@robots.ox.ac.uk

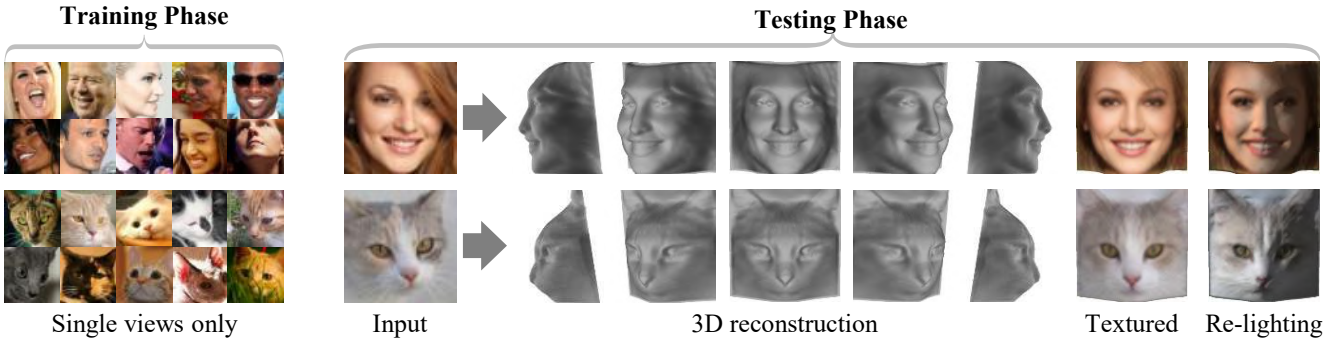


Figure 1: **Unsupervised learning of 3D deformable objects from in-the-wild images.** Left: Training uses *only* single views of the object category with *no* additional supervision at all (*i.e.* no ground-truth 3D information, multiple views, or any prior model of the object). Right: Once trained, our model reconstructs the 3D pose, shape, albedo and illumination of a deformable object instance from a single image with excellent fidelity. Code and demo at https://github.com/elliottwu/unsup3d.

Abstract

We propose a method to learn 3D deformable object categories from raw single-view images, without external supervision. The method is based on an autoencoder that factors each input image into depth, albedo, viewpoint and illumination. In order to disentangle these components without supervision, we use the fact that many object categories have, at least in principle, a symmetric structure. We show that reasoning about illumination allows us to exploit the underlying object symmetry even if the appearance is not symmetric due to shading. Furthermore, we model objects that are probably, but not certainly, symmetric by predicting a symmetry probability map, learned end-to-end with the other components of the model. Our experiments show that this method can recover very accurately the 3D shape of human faces, cat faces and cars from single-view images, without any supervision or a prior shape model. On benchmarks, we demonstrate superior accuracy compared to another method that uses supervision at the level of 2D image correspondences.

1. Introduction

Understanding the 3D structure of images is key in many computer vision applications. Futhermore, while many deep networks appear to understand images as 2D textures [16], 3D modelling can explain away much of the variability of natural images and potentially improve image understanding in general. Motivated by these facts, we consider the problem of learning 3D models for deformable object categories.

We study this problem under two challenging conditions. The *first* condition is that no 2D or 3D ground truth information (such as keypoints, segmentation, depth maps, or prior knowledge of a 3D model) is available. Learning without external supervisions removes the bottleneck of collecting image annotations, which is often a major obstacle to deploying deep learning for new applications. The *second* condition is that the algorithm must use an unconstrained collection of single-view images — in particular, it should not require multiple views of the same instance. Learning from single-view images is useful because in many applications, and especially for deformable objects, we solely have a source of still images to work with. Consequently, our learning algorithm ingests a number of single-view images of a deformable object category and produces as output a deep network that can estimate the 3D shape of any instance given a single image of it (Fig. 1).

We formulate this as an autoencoder that internally decomposes the image into albedo, depth, illumination and viewpoint, *without direct supervision for any of these factors*.

However, without further assumptions, decomposing images into these four factors is ill-posed. In search of minimal assumptions to achieve this, we note that many object categories are *symmetric* (*e.g.* almost all animals and many handcrafted objects). Assuming an object is perfectly symmetric, one can obtain a virtual second view of it by simply mirroring the image. In fact, if correspondences between the pair of mirrored images were available, 3D reconstruction could be achieved by stereo reconstruction [41, 12, 60, 54, 14]. Motivated by this, we seek to leverage symmetry as a geometric cue to constrain the decomposition.

However, specific object instances are in practice never fully symmetric, neither in shape nor appearance. Shape is non-symmetric due to variations in pose or other details (*e.g.* hair style or expressions on a human face), and albedo can also be non-symmetric (*e.g.* asymmetric texture of cat faces). Even when both shape and albedo are symmetric, the appearance may still not be, due to asymmetric illumination.

We address this issue in two ways. First, we explicitly model illumination to exploit the underlying symmetry and show that, by doing so, the model can exploit illumination as an additional cue for recovering the shape. Second, we augment the model to reason about potential lack of symmetry in the objects. To do this, the model predicts, along with the other factors, a dense map containing the probability that a given pixel has a symmetric counterpart in the image.

We combine these elements in an end-to-end learning formulation, where all components, including the confidence maps, are learned from raw RGB data only. We also show that symmetry can be enforced by flipping internal representations, which is particularly useful for reasoning about symmetries probabilistically.

We demonstrate our method on several datasets, including human faces, cat faces and cars. We provide a thorough ablation study using a synthetic face dataset to obtain the necessary 3D ground truth. On real images, we achieve higher fidelity reconstruction results compared to other methods [49, 56] that do not rely on 2D or 3D ground truth information, nor prior knowledge of a 3D model of the instance or class. In addition, we also outperform a recent state-of-the-art method [40] that uses keypoint supervision for 3D reconstruction on real faces, while our method uses no external supervision at all. Finally, we demonstrate that our trained face model generalizes to non-natural images such as face paintings and cartoon drawings without fine-tuning.

2. Related Work

In order to assess our contribution in relation to the vast literature on image-based 3D reconstruction, it is important to consider three aspects of each approach: which information is used, which assumptions are made, and what the output is. Below and in Table 1 we compare our contribution to prior works based on these factors.

Paper	Supervision	Goals	Data
[47]	3D scans	3DMM	Face
[66]	3DV, I	Prior on 3DV, predict from I	ShapeNet, Ikea
[1]	3DP	Prior on 3DP	ShapeNet
[48]	3DM	Prior on 3DM	Face
[17]	3DMM, 2DKP, I	Refine 3DMM fit to I	Face
[15]	3DMM, 2DKP, I	Fit 3DMM to I+2DKP	Face
[18]	3DMM	Fit 3DMM to 3D scans	Face
[28]	3DMM, 2DKP	Pred. 3DMM from I	Humans
[51]	3DMM, 2DS+KP	Pred. N, A, L from I	Face
[64]	3DMM, I	Pred. 3DM, VP, T, E from I	Face
[50]	3DMM, 2DKP, I	Fit 3DMM to I	Face
[13]	2DS	Prior on 3DV, pred. from I	Model/ScanNet
[30]	I, 2DS, VP	Prior on 3DV	ScanNet, PAS3D
[29]	I, 2DS+KP	Pred. 3DM, T, VP from I	Birds
[7]	I, 2DS	Pred. 3DM, T, L, VP from I	ShapeNet, Birds
[23]	I, 2DS	Pred. 3DV, VP from I	ShapeNet, others
[56]	I	Prior on 3DM, T, I	Face
[49]	I	Pred. 3DM, VP, T† from I	Face
[22]	I	Pred. V, L, VP from I	ShapeNet
Ours	I	Pred. D, L, A, VP from I	Face, others

Table 1: Comparison with selected prior work: supervision, goals, and data. I: image, 3DMM: 3D morphable model, 2DKP: 2D keypoints, 2DS: 2D silhouette, 3DP: 3D points, VP: viewpoint, E: expression, 3DM: 3D mesh, 3DV: 3D volume, D: depth, N: normals, A: albedo, T: texture, L: light. † can also recover A and L in post-processing.

Our method uses single-view images of an object category as training data, assumes that the objects belong to a specific class (*e.g.* human faces) which is weakly symmetric, and outputs a monocular predictor capable of decomposing any image of the category into shape, albedo, illumination, viewpoint and symmetry probability.

Structure from Motion. Traditional methods such as Structure from Motion (SfM) [11] can reconstruct the 3D structure of individual rigid scenes given as input multiple views of each scene and 2D keypoint matches between the views. This can be extended in two ways. First, *monocular reconstruction methods* can perform dense 3D reconstruction from a single image without 2D keypoints [74, 62, 20]. However, they require multiple views [20] or videos of rigid scenes for training [74]. Second, *Non-Rigid SfM* (NRSfM) approaches [4, 44] can learn to reconstruct deformable objects by allowing 3D points to deform in a limited manner between views, but require supervision in terms of annotated 2D keypoints for both training and testing. Hence, neither family of SfM approaches can learn to reconstruct deformable objects from raw pixels of a single view.

Shape from X. Many other monocular cues have been used as alternatives or supplements to SfM for recovering shape from images, such as shading [25, 71], silhouettes [33], texture [65], symmetry [41, 12] etc. In particular, our work is

inspired from *shape from symmetry* and *shape from shading*. Shape from symmetry [41, 12, 60, 54] reconstructs symmetric objects from a single image by using the mirrored image as a virtual second view, provided that symmetric correspondences are available. [54] also shows that it is possible to detect symmetries and correspondences using descriptors. Shape from shading [25, 71] assumes a shading model such as Lambertian reflectance, and reconstructs the surface by exploiting the non-uniform illumination.

Category-specific reconstruction. Learning-based methods have recently been leveraged to reconstruct objects from a single view, either in the form of a raw image or 2D keypoints (see also Table 1). While this task is ill-posed, it has been shown to be solvable by learning a suitable object prior from the training data [47, 66, 1, 48]. A variety of supervisory signals have been proposed to learn such priors. Besides using 3D ground truth directly, authors have considered using videos [2, 74, 43, 63] and stereo pairs [20, 38]. Other approaches have used single views with 2D keypoint annotations [29, 40, 55, 6] or object masks [29, 7]. For objects such as human bodies and human faces, some methods [28, 18, 64, 15] have learn to reconstruct from raw images, but starting from the knowledge of a predefined shape model such as SMPL [36] or Basel [47]. These prior models are constructed using specialized hardware and/or other forms of supervision, which are often difficult to obtain for deformable objects in the wild, such as animals, and also limited in details of the shape.

Only recently have authors attempted to learn the geometry of object categories from raw, monocular views *only*. Thewlis *et al*. [58, 59] uses equivariance to learn dense landmarks, which recovers the 2D geometry of the objects. DAE [52] learns to predict a deformation field through heavily constraining an autoencoder with a small bottleneck embedding and lift that to 3D in [49] — in post processing, they further decompose the reconstruction in albedo and shading, obtaining an output similar to ours.

Adversarial learning has been proposed as a way of hallucinating new views of an object. Some of these methods start from 3D representations [66, 1, 75, 48]. Kato *et al*. [30] trains a discriminator on raw images but uses viewpoint as addition supervision. HoloGAN [42] only uses raw images but does not obtain an explicit 3D reconstruction. Szabo *et al*. [56] uses adversarial training to reconstruct 3D meshes of the object, but does not assess their results quantitatively. Henzler *et al*. [23] also learns from raw images, but only experiments with images that contain the object on a white background, which is akin to supervision with 2D silhouettes. In Section 4.3, we compare to [49, 56] and demonstrate superior reconstruction results with much higher fidelity.

Since our model generates images from an internal 3D representation, one essential component is a differentiable renderer. However, with a traditional rendering pipeline, gradients across occlusions and boundaries are not defined. Several soft relaxations have thus been proposed [37, 31, 34]. Here, we use an implementation[1] of [31].

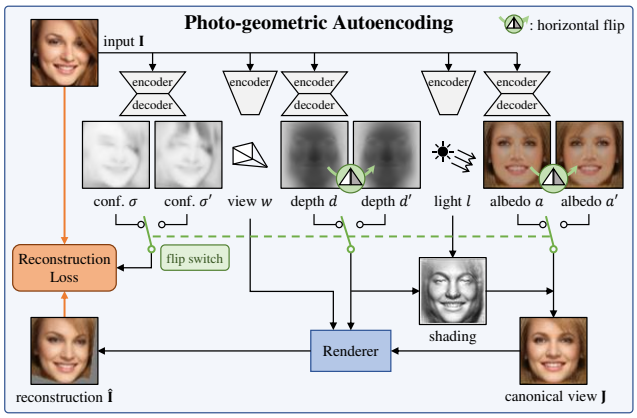


Figure 2: **Photo-geometric autoencoding**. Our network Φ decomposes an input image $\mathbf{I}$ into depth, albedo, viewpoint and lighting, together with a pair of confidence maps. It is trained to reconstruct the input without external supervision.

3. Method

Given an unconstrained collection of images of an object category, such as human faces, our goal is to learn a model Φ that receives as input an image of an object instance and produces as output a decomposition of it into 3D shape, albedo, illumination and viewpoint, as illustrated in Fig. 2.

As we have only raw images to learn from, the learning objective is reconstructive: namely, the model is trained so that the combination of the four factors gives back the input image. This results in an autoencoding pipeline where the factors have, due to the way they are recomposed, an explicit photo-geometric meaning.

In order to learn such a decomposition without supervision for any of the components, we use the fact that many object categories are *bilaterally symmetric*. However, the appearance of object instances is never perfectly symmetric. Asymmetries arise from shape deformation, asymmetric albedo and asymmetric illumination. We take two measures to account for these asymmetries. First, we explicitly model asymmetric illumination. Second, our model also estimates, for each pixel in the input image, a confidence score that explains the probability of the pixel having a symmetric counterpart in the image (see **conf** σ, σ' in Fig. 2).

The following sections describe how this is done, looking first at the photo-geometric autoencoder (Section 3.1), then at how symmetries are modelled (Section 3.2), followed by details of the image formation (Section 3.3) and the supplementary perceptual loss (Section 3.4).

[1] https://github.com/daniilidis-group/neural_renderer

3.1. Photo-geometric autoencoding

An image $\mathbf{I}$ is a function $\Omega \to \mathbb{R}^3$ defined on a grid $\Omega = \{0, \dots, W-1\} \times \{0, \dots, H-1\}$, or, equivalently, a tensor in $\mathbb{R}^{3\times W\times H}$. We assume that the image is roughly centered on an instance of the object of interest. The goal is to learn a function Φ, implemented as a neural network, that maps the image $\mathbf{I}$ to four factors (d, a, w, l) comprising a *depth map* $d : \Omega \to \mathbb{R}_+$, an *albedo image* $a : \Omega \to \mathbb{R}^3$, a global *light direction* $l \in \mathbb{S}^2$, and a *viewpoint* $w \in \mathbb{R}^6$ so that the image can be reconstructed from them.

The image $\mathbf{I}$ is reconstructed from the four factors in two steps, *lighting* Λ and *reprojection* Π, as follows:

$$\hat{\mathbf{I}} = \Pi\left(\Lambda(a, d, l), d, w\right). \tag{1}$$

The lighting function Λ generates a version of the object based on the depth map d, the light direction l and the albedo a as seen from a canonical viewpoint $w = 0$. The viewpoint w represents the transformation between the canonical view and the viewpoint of the actual input image $\mathbf{I}$. Then, the reprojection function Π simulates the effect of a viewpoint change and generates the image $\hat{\mathbf{I}}$ given the canonical depth d and the shaded canonical image $\Lambda(a, d, l)$. Learning uses a reconstruction loss which encourages $\mathbf{I} \approx \hat{\mathbf{I}}$ (Section 3.2).

Discussion. The effect of lighting could be incorporated in the albedo a by interpreting the latter as a texture rather than as the object's albedo. However, there are two good reasons to avoid this. First, the albedo a is often symmetric even if the illumination causes the corresponding appearance to look asymmetric. Separating them allows us to more effectively incorporate the symmetry constraint described below. Second, shading provides an additional cue on the underlying 3D shape [24, 3]. In particular, unlike the recent work of [52] where a shading map is predicted independently from shape, our model computes the shading based on the predicted depth, mutually constraining each other.

3.2. Probably symmetric objects

Leveraging symmetry for 3D reconstruction requires identifying symmetric object points in an image. Here we do so implicitly, assuming that depth and albedo, which are reconstructed in a canonical frame, are symmetric about a fixed vertical plane. An important beneficial side effect of this choice is that it helps the model discover a 'canonical view' for the object, which is important for reconstruction [44].

To do this, we consider the operator that flips a map $a \in \mathbb{R}^{C\times W\times H}$ along the horizontal axis[2]: $[\operatorname{flip} a]_{c,u,v} = a_{c,W-1-u,v}$. We then require $d \approx \operatorname{flip} d'$ and $a \approx \operatorname{flip} a'$. While these constraints could be enforced by adding corresponding loss terms to the learning objective, they would be difficult to balance. Instead, we achieve the same effect indirectly, by obtaining a second reconstruction $\hat{\mathbf{I}}'$ from the flipped depth and albedo:

$$\hat{\mathbf{I}}' = \Pi\left(\Lambda(a', d', l), d', w\right), \quad a' = \operatorname{flip} a, \quad d' = \operatorname{flip} d. \tag{2}$$

Then, we consider two reconstruction losses encouraging $\mathbf{I} \approx \hat{\mathbf{I}}$ and $\mathbf{I} \approx \hat{\mathbf{I}}'$. Since the two losses are commensurate, they are easy to balance and train jointly. Most importantly, this approach allows us to easily reason about symmetry probabilistically, as explained next.

The source image $\mathbf{I}$ and the reconstruction $\hat{\mathbf{I}}$ are compared via the loss:

$$\mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) = -\frac{1}{|\Omega|}\sum_{uv\in\Omega} \ln \frac{1}{\sqrt{2}\sigma_{uv}} \exp -\frac{\sqrt{2}\ell_{1,uv}}{\sigma_{uv}}, \tag{3}$$

where $\ell_{1,uv} = |\hat{\mathbf{I}}_{uv} - \mathbf{I}_{uv}|$ is the L_1 distance between the intensity of pixels at location uv, and $\sigma \in \mathbb{R}_+^{W\times H}$ is a *confidence map*, also estimated by the network Φ from the image $\mathbf{I}$, which expresses the *aleatoric uncertainty* of the model. The loss can be interpreted as the negative log-likelihood of a factorized Laplacian distribution on the reconstruction residuals. Optimizing likelihood causes the model to self-calibrate, learning a meaningful confidence map [32].

Modelling uncertainty is generally useful, but in our case is particularly important when we consider the "symmetric" reconstruction $\hat{\mathbf{I}}'$, for which we use the same loss $\mathcal{L}(\hat{\mathbf{I}}', \mathbf{I}, \sigma')$. Crucially, we use the network to estimate, also from the same input image $\mathbf{I}$, a *second* confidence map σ'. This confidence map allows the model to learn which portions of the input image might *not* be symmetric. For instance, in some cases hair on a human face is not symmetric as shown in Fig. 2, and σ' can assign a higher reconstruction uncertainty to the hair region where the symmetry assumption is not satisfied. Note that this depends on the *specific* instance under consideration, and is learned by the model itself.

Overall, the learning objective is given by the combination of the two reconstruction errors:

$$\mathcal{E}(\Phi; \mathbf{I}) = \mathcal{L}(\hat{\mathbf{I}}, \mathbf{I}, \sigma) + \lambda_{\mathrm{f}}\mathcal{L}(\hat{\mathbf{I}}', \mathbf{I}, \sigma'), \tag{4}$$

where $\lambda_{\mathrm{f}} = 0.5$ is a weighing factor, $(d, a, w, l, \sigma, \sigma') = \Phi(\mathbf{I})$ is the output of the neural network, and $\hat{\mathbf{I}}$ and $\hat{\mathbf{I}}'$ are obtained according to Eqs. (1) and (2).

3.3. Image formation model

We now describe the functions Π and Λ in Eq. (1) in more detail. The image is formed by a camera looking at a 3D object. If we denote with $P = (P_x, P_y, P_z) \in \mathbb{R}^3$ a 3D point expressed in the reference frame of the camera, this is mapped to pixel $p = (u, v, 1)$ by the following projection:

$$p \propto KP, \quad K = \begin{bmatrix} f & 0 & c_u \\ 0 & f & c_v \\ 0 & 0 & 1 \end{bmatrix}, \quad \begin{cases} c_u = \frac{W-1}{2}, \\ c_v = \frac{H-1}{2}, \\ f = \frac{W-1}{2\tan\frac{\theta_{\mathrm{FOV}}}{2}}. \end{cases} \tag{5}$$

[2]The choice of axis is arbitrary as long as it is fixed.

This model assumes a perspective camera with *field of view* (FOV) θ_{FOV}. We assume a nominal distance of the object from the camera at about 1m. Given that the images are cropped around a particular object, we assume a relatively narrow FOV of $\theta_{\text{FOV}} \approx 10°$.

The depth map $d : \Omega \to \mathbb{R}_+$ associates a depth value d_{uv} to each pixel $(u, v) \in \Omega$ in the canonical view. By inverting the camera model (5), we find that this corresponds to the 3D point $P = d_{uv} \cdot K^{-1}p$.

The viewpoint $w \in \mathbb{R}^6$ represents an Euclidean transformation $(R, T) \in SE(3)$, where $w_{1:3}$ and $w_{4:6}$ are rotation angles and translations along x, y and z axes respectively.

The map (R, T) transforms 3D points from the canonical view to the actual view. Thus a pixel (u, v) in the canonical view is mapped to the pixel (u', v') in the actual view by the warping function $\eta_{d,w} : (u, v) \mapsto (u', v')$ given by:

$$p' \propto K(d_{uv} \cdot RK^{-1}p + T), \tag{6}$$

where $p' = (u', v', 1)$.

Finally, the reprojection function Π takes as input the depth d and the viewpoint change w and applies the resulting warp to the canonical image $\mathbf{J}$ to obtain the actual image $\hat{\mathbf{I}} = \Pi(\mathbf{J}, d, w)$ as $\hat{\mathbf{I}}_{u'v'} = \mathbf{J}_{uv}$, where $(u, v) = \eta^{-1}_{d,w}(u', v')$.[3]

The canonical image $\mathbf{J} = \Lambda(a, d, l)$ is in turn generated as a combination of albedo, normal map and light direction. To do so, given the depth map d, we derive the normal map $n : \Omega \to \mathbb{S}^2$ by associating to each pixel (u, v) a vector normal to the underlying 3D surface. In order to find this vector, we compute the vectors t^u_{uv} and t^v_{uv} tangent to the surface along the u and v directions. For example, the first one is: $t^u_{uv} = d_{u+1,v} \cdot K^{-1}(p + e_x) - d_{u-1,v} \cdot K^{-1}(p - e_x)$ where p is defined above and $e_x = (1, 0, 0)$. Then the normal is obtained by taking the vector product $n_{uv} \propto t^u_{uv} \times t^v_{uv}$.

The normal n_{uv} is multiplied by the light direction l to obtain a value for the directional illumination and the latter is added to the ambient light. Finally, the result is multiplied by the albedo to obtain the illuminated texture, as follows: $\mathbf{J}_{uv} = (k_s + k_d \max\{0, \langle l, n_{uv}\rangle\}) \cdot a_{uv}$. Here k_s and k_d are the scalar coefficients weighting the ambient and diffuse terms, and are predicted by the model with range between 0 and 1 via rescaling a `tanh` output. The light direction $l = (l_x, l_y, 1)^T/(l_x^2 + l_y^2 + 1)^{0.5}$ is modeled as a spherical sector by predicting l_x and l_y with `tanh`.

3.4. Perceptual loss

The L_1 loss function Eq. (3) is sensitive to small geometric imperfections and tends to result in blurry reconstructions. We add a *perceptual loss* term to mitigate this problem. The k-th layer of an off-the-shelf image encoder e (VGG16 in our case [53]) predicts a representation $e^{(k)}(\mathbf{I}) \in \mathbb{R}^{C_k \times W_k \times H_k}$ where $\Omega_k = \{0, \dots, W_k - 1\} \times \{0, \dots, H_k - 1\}$ is the corresponding spatial domain. Note that this feature encoder does not have to be trained with supervised tasks. Self-supervised encoders can be equally effective as shown in Table 3.

Similar to Eq. (3), assuming a Gaussian distribution, the perceptual loss is given by:

$$\mathcal{L}^{(k)}_{\text{p}}(\hat{\mathbf{I}}, \mathbf{I}, \sigma^{(k)}) = -\frac{1}{|\Omega_k|} \sum_{uv \in \Omega_k} \ln \frac{1}{\sqrt{2\pi(\sigma^{(k)}_{uv})^2}} \exp -\frac{(\ell^{(k)}_{uv})^2}{2(\sigma^{(k)}_{uv})^2}, \tag{7}$$

where $\ell^{(k)}_{uv} = |e^{(k)}_{uv}(\hat{\mathbf{I}}) - e^{(k)}_{uv}(\mathbf{I})|$ for each pixel index uv in the k-th layer. We also compute the loss for $\hat{\mathbf{I}}'$ using $\sigma^{(k)'}$. $\sigma^{(k)}$ and $\sigma^{(k)'}$ are additional confidence maps predicted by our model. In practice, we found it is good enough for our purpose to use the features from only one layer `relu3_3` of VGG16. We therefore shorten the notation of perceptual loss to $\mathcal{L}_{\text{p}}$. With this, the loss function $\mathcal{L}$ in Eq. (4) is replaced by $\mathcal{L} + \lambda_{\text{p}}\mathcal{L}_{\text{p}}$ with $\lambda_{\text{p}} = 1$.

4. Experiments

4.1. Setup

Datasets. We test our method on three human face datasets: **CelebA** [35], **3DFAW** [21, 27, 73, 69] and **BFM** [47]. CelebA is a large scale human face dataset, consisting of over 200k images of real human faces in the wild annotated with bounding boxes. 3DFAW contains 23k images with 66 3D keypoint annotations, which we use to evaluate our 3D predictions in Section 4.3. We roughly crop the images around the head region and use the official train/val/test splits. BFM (Basel Face Model) is a synthetic face model, which we use to assess the quality of the 3D reconstructions (since the in-the-wild datasets lack ground-truth). We follow the protocol of [51] to generate a dataset, sampling shapes, poses, textures, and illumination randomly. We use images from SUN Database [68] as background and save ground truth depth maps for evaluation.

We also test our method on cat faces and synthetic cars. We use two **cat datasets** [72, 46]. The first one has 10k cat images with nine keypoint annotations, and the second one is a collection of dog and cat images, containing 1.2k cat images with bounding box annotations. We combine the two datasets and crop the images around the cat heads. For cars, we render 35k images of synthetic cars from **ShapeNet** [5] with random viewpoints and illumination. We randomly split the images by 8:1:1 into train, validation and test sets.

Metrics. Since the scale of 3D reconstruction from projective cameras is inherently ambiguous [11], we discount it in the evaluation. Specifically, given the depth map d predicted by our model in the canonical view, we warp it to a depth map $\bar{d}$ in the actual view using the predicted viewpoint and compare the latter to the ground-truth depth map d^* using the **scale-invariant depth error** (SIDE) [10]

[3] Note that this requires to compute the *inverse* of the warp $\eta_{d,w}$, which is detailed in Section 6.1.

No	Baseline	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Supervised	0.410 ± 0.103	10.78 ± 1.01
(2)	Const. null depth	2.723 ± 0.371	43.34 ± 2.25
(3)	Average g.t. depth	1.990 ± 0.556	23.26 ± 2.85
(4)	Ours (unsupervised)	0.793 ± 0.140	16.51 ± 1.56

Table 2: **Comparison with baselines.** SIDE and MAD errors of our reconstructions on the BFM dataset compared against a fully-supervised and trivial baselines.

$E_{\text{SIDE}}(\bar{d}, d^*) = (\frac{1}{WH}\sum_{uv}\Delta_{uv}^2 - (\frac{1}{WH}\sum_{uv}\Delta_{uv})^2)^{\frac{1}{2}}$ where $\Delta_{uv} = \log \bar{d}_{uv} - \log d^*_{uv}$. We compare only valid depth pixel and erode the foreground mask by one pixel to discount rendering artefacts at object boundaries. Additionally, we report the **mean angle deviation** (MAD) between normals computed from ground truth depth and from the predicted depth, measuring how well the surface is captured.

Implementation details. The function $(d, a, w, l, \sigma) = \Phi(\mathbf{I})$ that preditcs depth, albedo, viewpoint, lighting, and confidence maps from the image $\mathbf{I}$ is implemented using individual neural networks. The depth and albedo are generated by encoder-decoder networks, while viewpoint and lighting are regressed using simple encoder networks. The encoder-decoders do not use skip connections because input and output images are *not* spatially aligned (since the output is in the canonical viewpoint). All four confidence maps are predicted using the same network, at different decoding layers for the photometric and perceptual losses since these are computed at different resolutions. The final activation function is `tanh` for depth, albedo, viewpoint and lighting and `softplus` for the confidence maps. The depth prediction is centered on the mean before `tanh`, as the global distance is estimated as part of the viewpoint. We do *not* use any special initialization for all predictions, except that two border pixels of the depth maps on both the left and the right are clamped at a maximal depth to avoid boundary issues.

We train using Adam over batches of 64 input images, resized to 64×64 pixels. The size of the output depth and albedo is also 64×64. We train for approximately 50k iterations. For visualization, depth maps are upsampled to 256. We include more details in Section 6.2.

4.2. Results

Comparison with baselines. Table 2 uses the BFM dataset to compare the depth reconstruction quality obtained by our method, a fully-supervised baseline and two baselines. The supervised baseline is a version of our model trained to regress the ground-truth depth maps using an L_1 loss. The trivial baseline predicts a constant uniform depth map, which provides a performance lower-bound. The third baseline is a constant depth map obtained by averaging all ground-truth depth maps in the test set. Our method largely outperforms the two constant baselines and approaches the results of supervised training. Improving over the third baseline (which has access to GT information) confirms that the model learns an *instance specific* 3D representation.

No	Method	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
(1)	Ours full	0.793 ± 0.140	16.51 ± 1.56
(2)	w/o albedo flip	2.916 ± 0.300	39.04 ± 1.80
(3)	w/o depth flip	1.139 ± 0.244	27.06 ± 2.33
(4)	w/o light	2.406 ± 0.676	41.64 ± 8.48
(5)	w/o perc. loss	0.931 ± 0.269	17.90 ± 2.31
(6)	w/ self-sup. perc. loss	0.815 ± 0.145	15.88 ± 1.57
(7)	w/o confidence	0.829 ± 0.213	16.39 ± 2.12

Table 3: **Ablation study.** Refer to Section 4.2 for details.

	SIDE ($\times 10^{-2}$) ↓	MAD (deg.) ↓
No perturb, no conf.	0.829 ± 0.213	16.39 ± 2.12
No perturb, conf.	0.793 ± 0.140	16.51 ± 1.56
Perturb, no conf.	2.141 ± 0.842	26.61 ± 5.39
Perturb, conf.	0.878 ± 0.169	17.14 ± 1.90

Table 4: **Asymmetric perturbation.** We add asymmetric perturbations to BFM and show that confidence maps allow the model to reject such noise, while the vanilla model without confidence maps breaks.

	Depth Corr. ↑
Ground truth	66
AIGN [61] (**supervised**, from [40])	50.81
DepthNetGAN [40] (**supervised**, from [40])	58.68
MOFA [57] (**model-based**, from [40])	15.97
DepthNet [40] (from [40])	26.32
DepthNet [40] (from GitHub)	35.77
Ours	48.98
Ours (w/ CelebA pre-training)	54.65

Table 5: **3DFAW keypoint depth evaluation.** Depth correlation between ground truth and prediction evaluated at 66 facial keypoint locations.

Ablation. To understand the influence of the individual parts of the model, we remove them one at a time and evaluate the performance of the ablated model in Table 3. Visual results are reported in Fig. 9.

In the table, **row (1)** shows the performance of the full model (the same as in Table 2). **Row (2)** does not flip the albedo. Thus, the albedo is not encouraged to be symmetric in the canonical space, which fails to canonicalize the viewpoint of the object and to use cues from symmetry to recover shape. The performance is as low as the trivial baseline in Table 2. **Row (3)** does not flip the depth, with a similar effect to row (2). **Row (4)** predicts a shading map instead of computing it from depth and light direction. This also harms performance significantly because shading cannot be

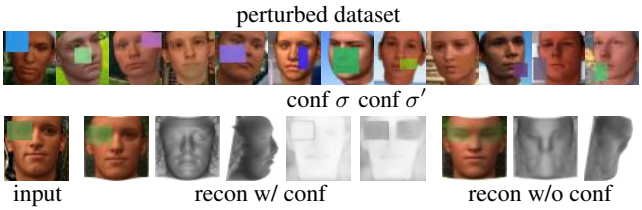


Figure 3: **Asymmetric perturbation.** Top: examples of the perturbed dataset. Bottom: reconstructions with and without confidence maps. Confidence allows the model to correctly reconstruct the 3D shape with the asymmetric texture.

used as a cue to recover shape. **Row (5)** switches off the perceptual loss, which leads to degraded image quality and hence degraded reconstruction results. **Row (6)** replaces the ImageNet pretrained image encoder used in the perceptual loss with one[4] trained through a self-supervised task [19], which shows no difference in performance. Finally, **row (7)** switches off the confidence maps, using a fixed and uniform value for the confidence — this reduces losses (3) and (7) to the basic L_1 and L_2 losses, respectively. The accuracy does not drop significantly, as faces in BFM are highly symmetric (*e.g.* do not have hair), but its variance increases. To better understand the effect of the confidence maps, we specifically evaluate on partially asymmetric faces using perturbations.

Asymmetric perturbation. In order to demonstrate that our uncertainty modelling allows the model to handle asymmetry, we add asymmetric perturbations to BFM. Specifically, we generate random rectangular color patches with 20% to 50% of the image size and blend them onto the images with α-values ranging from 0.5 to 1, as shown in Fig. 3. We then train our model with and without confidence on these perturbed images, and report the results in Table 4. Without the confidence maps, the model always predicts a symmetric albedo and geometry reconstruction often fails. With our confidence estimates, the model is able to reconstruct the asymmetric faces correctly, with very little loss in accuracy compared to the unperturbed case.

Qualitative results. In Fig. 4 we show reconstruction results of human faces from CelebA and 3DFAW, cat faces from [72, 46] and synthetic cars from ShapeNet. The 3D shapes are recovered with high fidelity. The reconstructed 3D face, for instance, contain fine details of the nose, eyes and mouth even in the presence of extreme facial expression.

To further test generalization, we applied our model trained on the CelebA dataset to a number of paintings and cartoon drawings of faces collected from [9] and the Internet. As shown in Fig. 5, our method still works well even though it has never seen such images during training.

[4]We use a RotNet [19] pretrained VGG16 model obtained from https://github.com/facebookresearch/DeeperCluster.

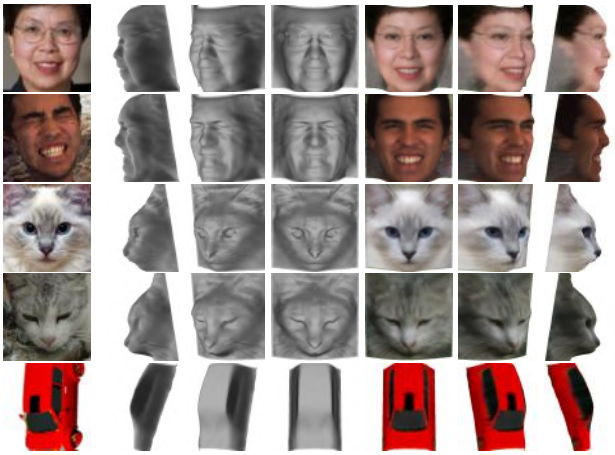


Figure 4: **Reconstruction of faces, cats and cars.**

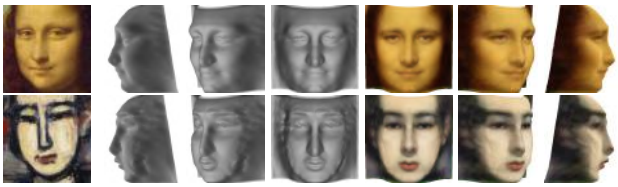
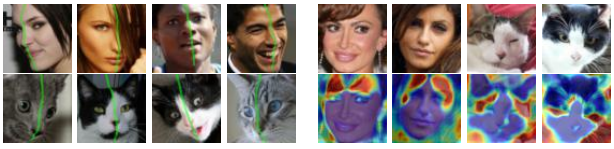


Figure 5: **Reconstruction of faces in paintings.**



(a) symmetry plane (b) asymmetry visualization

Figure 6: **Symmetry plane and asymmetry detection.** (a): our model can reconstruct the "intrinsic" symmetry plane of an in-the-wild object even though the appearance is highly asymmetric. (b): asymmetries (highlighted in red) are detected and visualized using confidence map σ'.

Symmetry and asymmetry detection. Since our model predicts a canonical view of the objects that is symmetric about the vertical center-line of the image, we can easily visualize the symmetry plane, which is otherwise non-trivial to detect from in-the-wild images. In Fig. 6, we warp the center-line of the canonical image to the predicted input viewpoint. Our method can detect symmetry planes accurately despite the presence of asymmetric texture and lighting effects. We also overlay the predicted confidence map σ' onto the image, confirming that the model assigns low confidence to asymmetric regions in a sample-specific way.

4.3. Comparison with the state of the art

As shown in Table 1, most reconstruction methods in the literature require either image annotations, prior 3D models

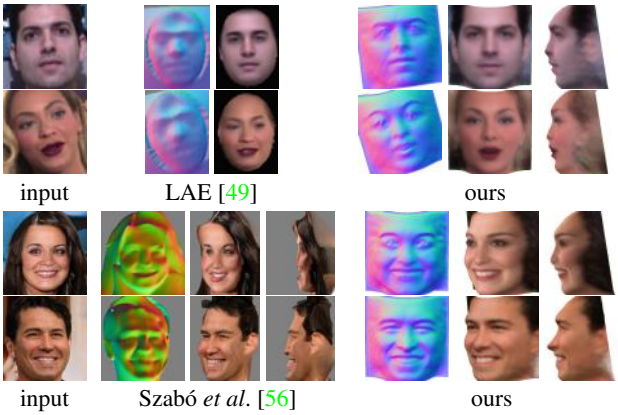


Figure 7: **Qualitative comparison to SOTA.** Our method recovers much higher quality shapes compared to [49, 56].

or both. When these assumptions are dropped, the task becomes considerably harder, and there is little prior work that is directly comparable. Of these, [22] only uses synthetic, texture-less objects from ShapeNet, [56] reconstructs in-the-wild faces but does not report any quantitative results, and [49] reports quantitative results only on keypoint regression, but not on the 3D reconstruction quality. We were not able to obtain code or trained models from [49, 56] for a direct quantitative comparison and thus compare qualitatively.

Qualitative comparison. In order to establish a side-by-side comparison, we cropped the examples reported in the papers [49, 56] and compare our results with theirs (Fig. 7). Our method produces much higher quality reconstructions than both methods, with fine details of the facial expression, whereas [49] recovers 3D shapes poorly and [56] generates unnatural shapes. Note that [56] uses an unconditional GAN that generates high resolution 3D faces from random noise, and cannot recover 3D shapes from images. The input images for [56] in Fig. 7 were generated by their GAN.

3D keypoint depth evaluation. Next, we compare to the DepthNet model of [40]. This method predicts depth for selected facial keypoints, but uses 2D keypoint annotations as input — a much easier setup than the one we consider here. Still, we compare the quality of the reconstruction of these sparse point obtained by DepthNet and our method. We also compare to the baselines MOFA [57] and AIGN [61] reported in [40]. For a fair comparison, we use their public code which computes the depth correlation score (between 0 and 66) on the frontal faces. We use the 2D keypoint locations to sample our predicted depth and then evaluate the same metric. The set of test images from 3DFAW and the preprocessing are identical to [40]. Since 3DFAW is a small dataset with limited variation, we also report results with CelebA pre-training.

In Table 5 we report the results from their paper and the slightly improved results we obtained from their publicly-available implementation. The paper also evaluates a supervised model using a GAN discriminator trained with ground-truth depth information. While our method does not use any supervision, it still outperforms DepthNet and reaches close-to-supervised performance.

Figure 8: **Failure cases.** See Section 4.4 for details.

4.4. Limitations

While our method is robust in many challenging scenarios (e.g., extreme facial expression, abstract drawing), we do observe failure cases as shown in Fig. 8. During training, we assume a simple Lambertian shading model, ignoring shadows and specularity, which leads to inaccurate reconstructions under extreme lighting conditions (Fig. 8a) or highly non-Lambertian surfaces. Disentangling noisy dark textures and shading (Fig. 8b) is often difficult. The reconstruction quality is lower for extreme poses (Fig. 8c), partly due to poor supervisory signal from the reconstruction loss of side images. This may be improved by imposing constraints from accurate reconstructions of frontal poses.

5. Conclusions

We have presented a method that can learn a 3D model of a deformable object category from an unconstrained collection of single-view images of the object category. The model is able to obtain high-fidelity monocular 3D reconstructions of individual object instances. This is trained based on a reconstruction loss without any supervision, resembling an autoencoder. We have shown that symmetry and illumination are strong cues for shape and help the model to converge to a meaningful reconstruction. Our model outperforms a current state-of-the-art 3D reconstruction method that uses 2D keypoint supervision. As for future work, the model currently represents 3D shape from a canonical viewpoint using a depth map, which is sufficient for objects such as faces that have a roughly convex shape and a natural canonical viewpoint. For more complex objects, it may be possible to extend the model to use either multiple canonical views or a different 3D representation, such as a mesh or a voxel map.

Acknowledgements We would like to thank Soumyadip Sengupta for sharing with us the code to generate synthetic face datasets, and Mihir Sahasrabudhe for sending us the reconstruction results of Lifting AutoEncoders. We are also indebted to the members of Visual Geometry Group for insightful discussions. This work is jointly supported by Facebook Research and ERC Horizon 2020 research and innovation programme IDIU 638009.

References

[1] Panos Achlioptas, Olga Diamanti, Ioannis Mitliagkas, and Leonidas Guibas. Learning representations and generative models for 3D point clouds. In *Proc. ICML*, 2018. 2, 3

[2] Pulkit Agrawal, Joao Carreira, and Jitendra Malik. Learning to see by moving. In *Proc. ICCV*, 2015. 3

[3] Peter N. Belhumeur, David J. Kriegman, and Alan L. Yuille. The bas-relief ambiguity. *IJCV*, 1999. 4

[4] Christoph Bregler, Aaron Hertzmann, and Henning Biermann. Recovering non-rigid 3D shape from image streams. In *Proc. CVPR*, 2000. 2

[5] Angel X. Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, Jianxiong Xiao, Li Yi, and Fisher Yu. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 5

[6] Ching-Hang Chen, Ambrish Tyagi, Amit Agrawal, Dylan Drover, Rohith MV, Stefan Stojanov, and James M. Rehg. Unsupervised 3d pose estimation with geometric self-supervision. In *Proc. CVPR*, 2019. 3

[7] Wenzheng Chen, Huan Ling, Jun Gao, Edward Smith, Jaako Lehtinen, Alec Jacobson, and Sanja Fidler. Learning to predict 3d objects with an interpolation-based differentiable renderer. In *NeurIPS*, 2019. 2, 3

[8] Joon Son Chung, Arsha Nagrani, and Andrew Zisserman. VoxCeleb2: Deep speaker recognition. In *INTERSPEECH*, 2018. 12

[9] Elliot J. Crowley, Omkar M. Parkhi, and Andrew Zisserman. Face painting: querying art with photos. In *Proc. BMVC*, 2015. 7, 12

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NeurIPS*, 2014. 5

[11] Olivier Faugeras and Quang-Tuan Luong. *The Geometry of Multiple Images*. MIT Press, 2001. 2, 5

[12] Alexandre R. J. François, Gérard G. Medioni, and Roman Waupotitsch. Mirror symmetry $\Rightarrow$ 2-view stereo geometry. *Image and Vision Computing*, 2003. 2, 3

[13] Matheus Gadelha, Subhransu Maji, and Rui Wang. 3D shape induction from 2D views of multiple objects. In *3DV*, 2017. 2

[14] Yuan Gao and Alan L. Yuille. Exploiting symmetry and/or manhattan properties for 3d object structure estimation from single and multiple images. In *Proc. CVPR*, 2017. 2

[15] Baris Gecer, Stylianos Ploumpis, Irene Kotsia, and Stefanos Zafeiriou. GANFIT: Generative adversarial network fitting for high fidelity 3D face reconstruction. In *Proc. CVPR*, 2019. 2, 3

[16] Robert Geirhos, Patricia Rubisch, Claudio Michaelis, Matthias Bethge, Felix A. Wichmann, and Wieland Brendel. Imagenet-trained CNNs are biased towards texture; increasing shape bias improves accuracy and robustness. In *Proc. ICML*, 2019. 1

[17] Zhenglin Geng, Chen Cao, and Sergey Tulyakov. 3D guided fine-grained face manipulation. In *Proc. CVPR*, 2019. 2

[18] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Lüthi, Sandro Schönborn, and Thomas Vetter. Morphable face models - an open framework. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2018. 2, 3

[19] Spyros Gidaris, Praveer Singh, and Nikos Komodakis. Unsupervised representation learning by predicting image rotations. In *Proc. ICLR*, 2018. 7

[20] Clément Godard, Oisin Mac Aodha, and Gabriel J. Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proc. CVPR*, 2017. 2, 3

[21] Ralph Gross, Iain Matthews, Jeffrey Cohn, Takeo Kanade, and Simon Baker. Multi-pie. *Image and Vision Computing*, 2010. 5

[22] Paul Henderson and Vittorio Ferrari. Learning single-image 3D reconstruction by generative modelling of shape, pose and shading. *IJCV*, 2019. 2, 8

[23] Philipp Henzler, Niloy Mitra, and Tobias Ritschel. Escaping plato's cave using adversarial training: 3d shape from unstructured 2d image collections. In *Proc. ICCV*, 2019. 2, 3

[24] Berthold Horn. Obtaining shape from shading information. In *The Psychology of Computer Vision*, 1975. 4

[25] Berthold K. P. Horn and Michael J. Brooks. *Shape from Shading*. MIT Press, Cambridge Massachusetts, 1989. 2, 3

[26] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. In *NeurIPS*, 2015. 11

[27] László A. Jeni, Jeffrey F. Cohn, and Takeo Kanade. Dense 3d face alignment from 2d videos in real-time. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2015. 5

[28] Angjoo Kanazawa, Michael J. Black, David W. Jacobs, and Jitendra Malik. End-to-end recovery of human shape and pose. In *Proc. CVPR*, 2018. 2, 3

[29] Angjoo Kanazawa, Shubham Tulsiani, Alexei A. Efros, and Jitendra Malik. Learning category-specific mesh reconstruction from image collections. In *Proc. ECCV*, 2018. 2, 3

[30] Hiroharu Kato and Tatsuya Harada. Learning view priors for single-view 3d reconstruction. In *Proc. CVPR*, 2019. 2, 3

[31] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proc. CVPR*, 2018. 3, 11

[32] Alex Kendall and Yarin Gal. What uncertainties do we need in bayesian deep learning for computer vision? In *NeurIPS*, 2017. 4

[33] Jan J Koenderink. What does the occluding contour tell us about solid shape? *Perception*, 1984. 2

[34] Shichen Liu, Tianye Li, Weikai Chen, and Hao Li. Soft rasterizer: A differentiable renderer for image-based 3d reasoning. In *Proc. ICCV*, 2019. 3

[35] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proc. ICCV*, 2015. 5

[36] Matthew Loper, Naureen Mahmood, Javier Romero, Gerard Pons-Moll, and Michael J Black. SMPL: A skinned multi-person linear model. *ACM TOG*, 34(6):248, 2015. 3

[37] Matthew M. Loper and Michael J. Black. OpenDR: An approximate differentiable renderer. In *Proc. ECCV*, 2014. 3

[38] Yue Luo, Jimmy Ren, Mude Lin, Jiahao Pang, Wenxiu Sun, Hongsheng Li, and Liang Lin. Single view stereo matching. In *Proc. CVPR*, 2018. 3

[39] Andrew L. Maas, Awni Y. Hannun, and Andrew Y. Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proc. ICML*, 2013. 12

[40] Joel Ruben Antony Moniz, Christopher Beckham, Simon Rajotte, Sina Honari, and Christopher Pal. Unsupervised depth estimation, 3d face rotation and replacement. In *NeurIPS*, 2018. 2, 3, 6, 8

[41] Dipti P. Mukherjee, Andrew Zisserman, and J. Michael Brady. Shape from symmetry – detecting and exploiting symmetry in affine images. *Philosophical Transactions of the Royal Society of London*, 351:77–106, 1995. 2, 3

[42] Thu Nguyen-Phuoc, Chuan Li, Lucas Theis, Christian Richardt, and Yong-Liang Yang. Hologan: Unsupervised learning of 3d representations from natural images. In *Proc. ICCV*, 2019. 3

[43] David Novotny, Diane Larlus, and Andrea Vedaldi. Learning 3d object categories by looking around them. In *Proc. ICCV*, 2017. 3

[44] David Novotny, Nikhila Ravi, Benjamin Graham, Natalia Neverova, and Andrea Vedaldi. C3DPO: Canonical 3d pose networks for non-rigid structure from motion. In *Proc. ICCV*, 2019. 2, 4

[45] Augustus Odena, Vincent Dumoulin, and Chris Olah. Deconvolution and checkerboard artifacts. *Distill*, 2016. 11

[46] Omkar M. Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *Proc. CVPR*, 2012. 5, 7, 12

[47] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3D face model for pose and illumination invariant face recognition. In *Advanced video and signal based surveillance*, 2009. 2, 3, 5

[48] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J. Black. Generating 3D faces using convolutional mesh autoencoders. In *Proc. ECCV*, 2018. 2, 3

[49] Mihir Sahasrabudhe, Zhixin Shu, Edward Bartrum, Riza Alp Guler, Dimitris Samaras, and Iasonas Kokkinos. Lifting autoencoders: Unsupervised learning of a fully-disentangled 3d morphable model using deep non-rigid structure from motion. In *Proc. ICCV Workshops*, 2019. 2, 3, 8

[50] Soubhik Sanyal, Timo Bolkart, Haiwen Feng, and Michael J. Black. Learning to regress 3D face shape and expression from an image without 3D supervision. In *Proc. CVPR*, 2019. 2

[51] Soumyadip Sengupta, Angjoo Kanazawa, Carlos D. Castillo, and David Jacobs. SfSNet: Learning shape, refectance and illuminance of faces in the wild. In *Proc. CVPR*, 2018. 2, 5

[52] Zhixin Shu, Mihir Sahasrabudhe, Alp Guler, Dimitris Samaras, Nikos Paragios, and Iasonas Kokkinos. Deforming autoencoders: Unsupervised disentangling of shape and appearance. In *Proc. ECCV*, 2018. 3, 4

[53] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *Proc. ICLR*, 2015. 5

[54] Sudipta N. Sinha, Krishnan Ramnath, and Richard Szeliski. Detecting and reconstructing 3d mirror symmetric objects. In *Proc. ECCV*, 2012. 2, 3

[55] Supasorn Suwajanakorn, Noah Snavely, Jonathan Tompson, and Mohammad Norouzi. Discovery of latent 3d keypoints via end-to-end geometric reasoning. In *NeurIPS*, 2018. 3

[56] Attila Szabó, Givi Meishvili, and Paolo Favaro. Unsupervised generative 3d shape learning from natural images. *arXiv preprint arXiv:1910.00287*, 2019. 2, 3, 8

[57] Ayush Tewari, Michael Zollhöfer, Hyeongwoo Kim, Pablo Garrido, Florian Bernard, Patrick Pérez, and Christian Theobalt. MoFA: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. In *Proc. ICCV*, 2017. 6, 8

[58] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Unsupervised learning of object frames by dense equivariant image labelling. In *NeurIPS*, 2017. 3

[59] James Thewlis, Hakan Bilen, and Andrea Vedaldi. Modelling and unsupervised learning of symmetric deformable object categories. In *NeurIPS*, 2018. 3

[60] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Proc. ICCV*, 2005. 2, 3

[61] Hsiao-Yu Fish Tung, Adam W. Harley, William Seto, and Katerina Fragkiadaki. Adversarial inverse graphics networks: Learning 2d-to-3d lifting and image-to-image translation from unpaired supervision. In *Proc. ICCV*, 2017. 6, 8

[62] Benjamin Ummenhofer, Huizhong Zhou, Jonas Uhrig, Nikolaus Mayer, Eddy Ilg, Alexey Dosovitskiy, and Thomas Brox. Demon: Depth and motion network for learning monocular stereo. In *Proc. CVPR*, 2017. 2

[63] Chaoyang Wang, Jose Miguel Buenaposada, Rui Zhu, and Simon Lucey. Learning depth from monocular videos using direct methods. In *Proc. CVPR*, 2018. 3

[64] Mengjiao Wang, Zhixin Shu, Shiyang Cheng, Yannis Panagakis, Dimitris Samaras, and Stefanos Zafeiriou. An adversarial neuro-tensorial approach for learning disentangled representations. *IJCV*, 2019. 2, 3

[65] Andrew P. Witkin. Recovering surface shape and orientation from texture. *Artificial Intelligence*, 1981. 2

[66] Jiajun Wu, Chengkai Zhang, Tianfan Xue, William T. Freeman, and Joshua B. Tenenbaum. Learning a probabilistic latent space of object shapes via 3d generative-adversarial modeling. In *NeurIPS*, 2016. 2, 3

[67] Yuxin Wu and Kaiming He. Group normalization. In *Proc. ECCV*, 2018. 12

[68] Jianxiong Xiao, James Hays, Krista A. Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Proc. CVPR*, 2010. 5

[69] Lijun Yin, Xiaochen Chen, Yi Sun, Tony Worm, and Michael Reale. A high-resolution 3d dynamic facial expression database. In *Proc. Int. Conf. Autom. Face and Gesture Recog.*, 2008. 5

[70] Matthew D. Zeiler, Graham W. Taylor, and Rob Fergus. Adaptive deconvolutional networks for mid and high level feature learning. In *Proc. ICCV*, 2011. 11

[71] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE PAMI*, 1999. 2, 3

[72] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection - how to effectively exploit shape and texture features. In *Proc. ECCV*, 2008. 5, 7, 12

[73] Xing Zhang, Lijun Yin, Jeffrey F. Cohn, Shaun Canavan, Michael Reale, Andy Horowitz, Peng Liu, and Jeffrey M. Girard. Bp4d-spontaneous: a high-resolution spontaneous 3d dynamic facial expression database. *Image and Vision Computing*, 32(10):692–706, 2014. 5

[74] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G. Lowe. Unsupervised learning of depth and ego-motion from video. In *Proc. CVPR*, 2017. 2, 3

[75] Jun-Yan Zhu, Zhoutong Zhang, Chengkai Zhang, Jiajun Wu, Antonio Torralba, Joshua B. Tenenbaum, and William T. Freeman. Visual object networks: Image generation with disentangled 3D representations. In *NeurIPS*, 2018. 3

6. Supplementary Material

6.1. Differentiable rendering layer

As noted in Section 3.3, the reprojection function Π *warps* the canonical image $\mathbf{J}$ to generate the actual image $\mathbf{I}$. In CNNs, image warping is usually regarded as a simple operation that can be implemented efficiently using a bilinear resampling layer [26]. However, this is true only if we can easily send pixels (u', v') in the warped image $\mathbf{I}$ back to pixels (u, v) in the source image $\mathbf{J}$, a process also known as *backward warping*. Unfortunately, in our case the function $\eta_{d,w}$ obtained by Eq. (6) sends pixels in the opposite way.

Implementing a *forward warping* layer is surprisingly delicate. One way of approaching the problem is to regard this task as a special case of rendering a textured mesh. The *Neural Mesh Renderer* (NMR) of [31] is a differentiable renderer of this type. In our case, the mesh has one vertex per pixel and each group of 2×2 adjacent pixels is tessellated by two triangles. Empirically, we found the quality of the texture gradients of NMR to be poor in this case, likely caused by high frequency content in the texture image $\mathbf{J}$.

We solve the problem as follows. First, we use NMR to warp only the depth map d, obtaining a version $\bar{d}$ of the depth map as seen from the input viewpoint. This has two advantages: backpropagation through NMR is faster and secondly, the gradients are more stable, probably also due to the comparatively smooth nature of the depth map d compared to the texture image $\mathbf{J}$. Given the depth map $\bar{d}$, we then use the inverse of Eq. (6) to find the warp field from the observed viewpoint to the canonical viewpoint, and bilinearly resample the canonical image $\mathbf{J}$ to obtain the reconstruction.

6.2. Training details

We report the training details including all hyperparameter settings in Table 6, and detailed network architectures in Tables 7 to 9. We use standard encoder networks for both viewpoint and lighting predictions, and encoder-decoder networks for depth, albedo and confidence predictions. In order to mitigate checkerboard artifacts [45] in the predicted depth and albedo, we add a convolution layer after each deconvolution layer and replace the last deconvolotion layer with nearest-neighbor upsampling, followed by 3 convolution layers. Abbreviations of the operators are defined as follows:

- Conv$(c_{in}, c_{out}, k, s, p)$: convolution with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.
- Deconv$(c_{in}, c_{out}, k, s, p)$: deconvolution [70] with c_{in} input channels, c_{out} output channels, kernel size k, stride s and padding p.

Parameter	Value/Range
Optimizer	Adam
Learning rate	1×10^{-4}
Number of epochs	30
Batch size	64
Loss weight λ_{f}	0.5
Loss weight λ_{p}	1
Input image size	64×64
Output image size	64×64
Depth map	$(0.9, 1.1)$
Albedo	$(0, 1)$
Light coefficient k_s	$(0, 1)$
Light coefficient k_d	$(0, 1)$
Light direction l_x, l_y	$(-1, 1)$
Viewpoint rotation $w_{1:3}$	$(-60°, 60°)$
Viewpoint translation $w_{4:6}$	$(-0.1, 0.1)$
Field of view (FOV)	10

Table 6: Training details and hyper-parameter settings.

Encoder	Output size
Conv(3, 32, 4, 2, 1) + ReLU	32
Conv(32, 64, 4, 2, 1) + ReLU	16
Conv(64, 128, 4, 2, 1) + ReLU	8
Conv(128, 256, 4, 2, 1) + ReLU	4
Conv(256, 256, 4, 1, 0) + ReLU	1
Conv(256, c_{out}, 1, 1, 0) + Tanh $\rightarrow$ *output*	1

Table 7: Network architecture for viewpoint and lighting. The output channel size c_{out} is 6 for viewpoint, corresponding to rotation angles $w_{1:3}$ and translations $w_{4:6}$ in x, y and z axes, and 4 for lighting, corresponding to k_s, k_d, l_x and l_y.

- Upsample(s): nearest-neighbor upsampling with a scale factor of s.
- GN(n): group normalization [67] with n groups.
- LReLU(α): leaky ReLU [39] with a negative slope of α.

7. Qualitative Results

We provide more qualitative results in the following and 3D animations in the supplementary video[5]. Fig. 9 reports the qualitative results of the ablated models in Table 3. Fig. 11 shows reconstruction results on human faces from CelebA and 3DFAW. We also show reconstruction results on face paintings and drawings collected from [9] and the Internet in Figs. 12 and 13. Figs. 14 to 16 show results on real cat faces from [72, 46], abstract cats collected from the Internet and synthetic cars rendered using ShapeNet.

[5] https://www.youtube.com/watch?v=5rPJyrU-WE4

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 256, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(256, 512, 4, 1, 0) + ReLU	4
Conv(512, 512, 3, 1, 1) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Conv(256, 256, 3, 1, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
Conv(128, 128, 3, 1, 1) + GN(32) + ReLU	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	32
Upsample(2)	64
Conv(64, 64, 3, 1, 1) + GN(16) + ReLU	64
Conv(64, 64, 5, 1, 2) + GN(16) + ReLU	64
Conv(64, c_{out}, 5, 1, 2) + Tanh $\rightarrow$ *output*	64

Table 8: Network architecture for depth and albedo. The output channel size c_{out} is 1 for depth and 3 for albedo.

Encoder	Output size
Conv(3, 64, 4, 2, 1) + GN(16) + LReLU(0.2)	32
Conv(64, 128, 4, 2, 1) + GN(32) + LReLU(0.2)	16
Conv(128, 256, 4, 2, 1) + GN(64) + LReLU(0.2)	8
Conv(256, 512, 4, 2, 1) + LReLU(0.2)	4
Conv(512, 128, 4, 1, 0) + ReLU	1
Decoder	**Output size**
Deconv(128, 512, 4, 1, 0) + ReLU	4
Deconv(512, 256, 4, 2, 1) + GN(64) + ReLU	8
Deconv(256, 128, 4, 2, 1) + GN(32) + ReLU	16
↳ Conv(128, 2, 3, 1, 1) + SoftPlus $\rightarrow$ *output*	16
Deconv(128, 64, 4, 2, 1) + GN(16) + ReLU	32
Deconv(64, 64, 4, 2, 1) + GN(16) + ReLU	64
Conv(64, 2, 5, 1, 2) + SoftPlus $\rightarrow$ *output*	64

Table 9: Network architecture for confidence maps. The network outputs two pairs of confidence maps at different spatial resolutions for photometric and perceptual losses.

Re-lighting. Since our model predicts the intrinsic components of an image, separating the albedo and illumination, we can easily re-light the objects with different lighting conditions. In Fig. 10, we demonstrate results of the intrinsic decomposition and the re-lit faces in the canonical view.

Testing on videos. To further assess our model, we apply the model trained on CelebA faces to VoxCeleb [8] videos *frame by frame* and include the results in the supplementary video. Our trained model works surprisingly well, producing consistent, smooth reconstructions across different frames and recovering the details of the facial motions accurately.

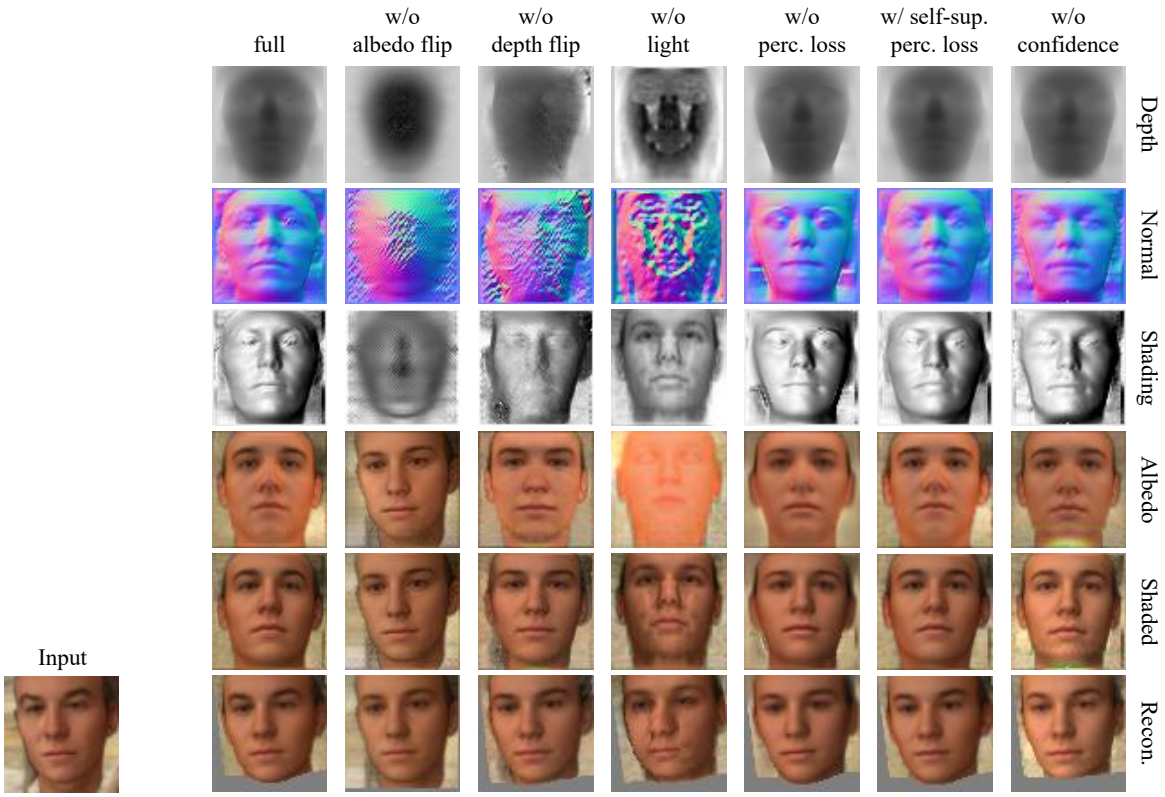


Figure 9: **Qualitative results of the ablated models.**

Figure 10: **Re-lighting effects.**

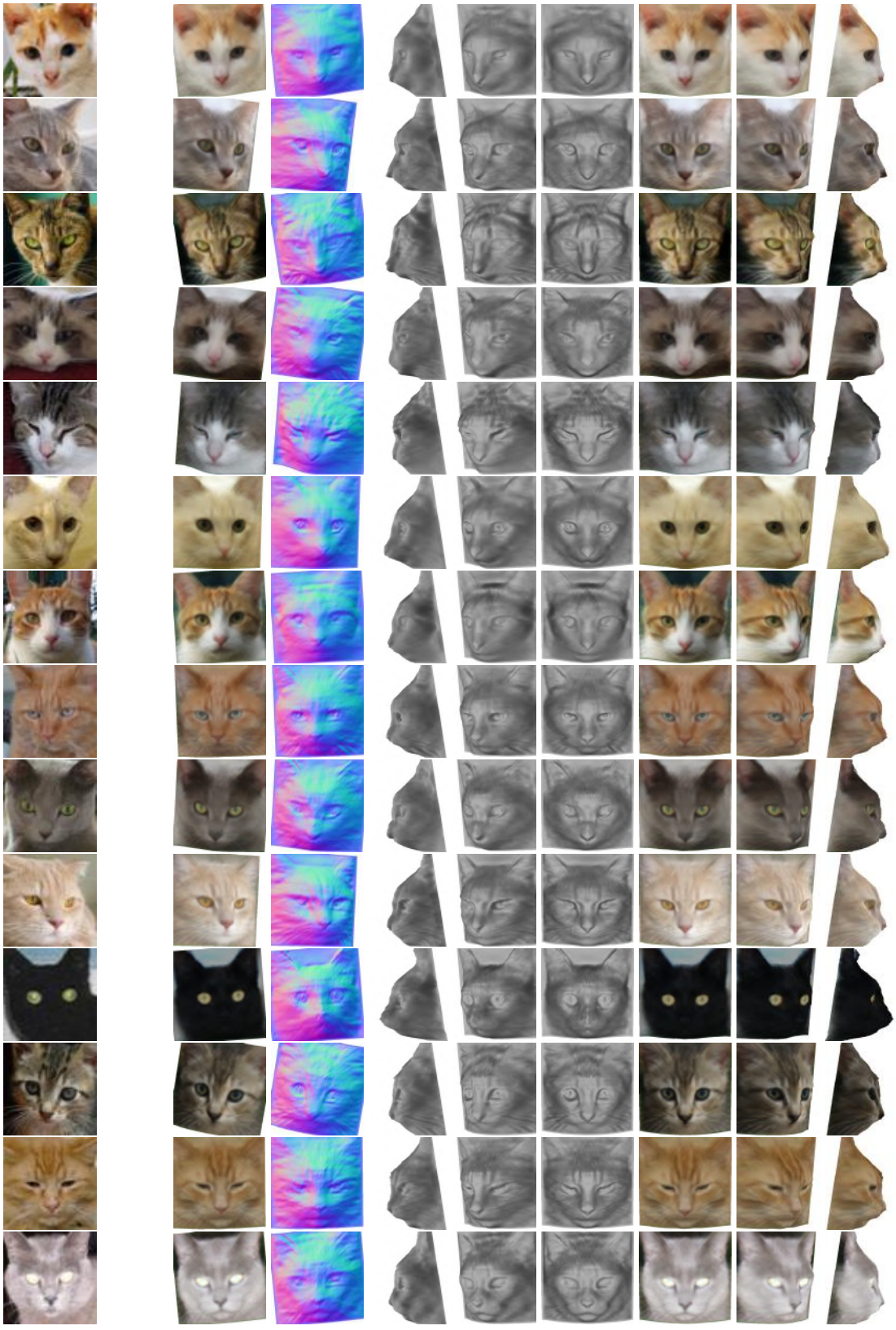
Input Reconstruction

Figure 11: **Reconstruction of human faces.**

Input Reconstruction

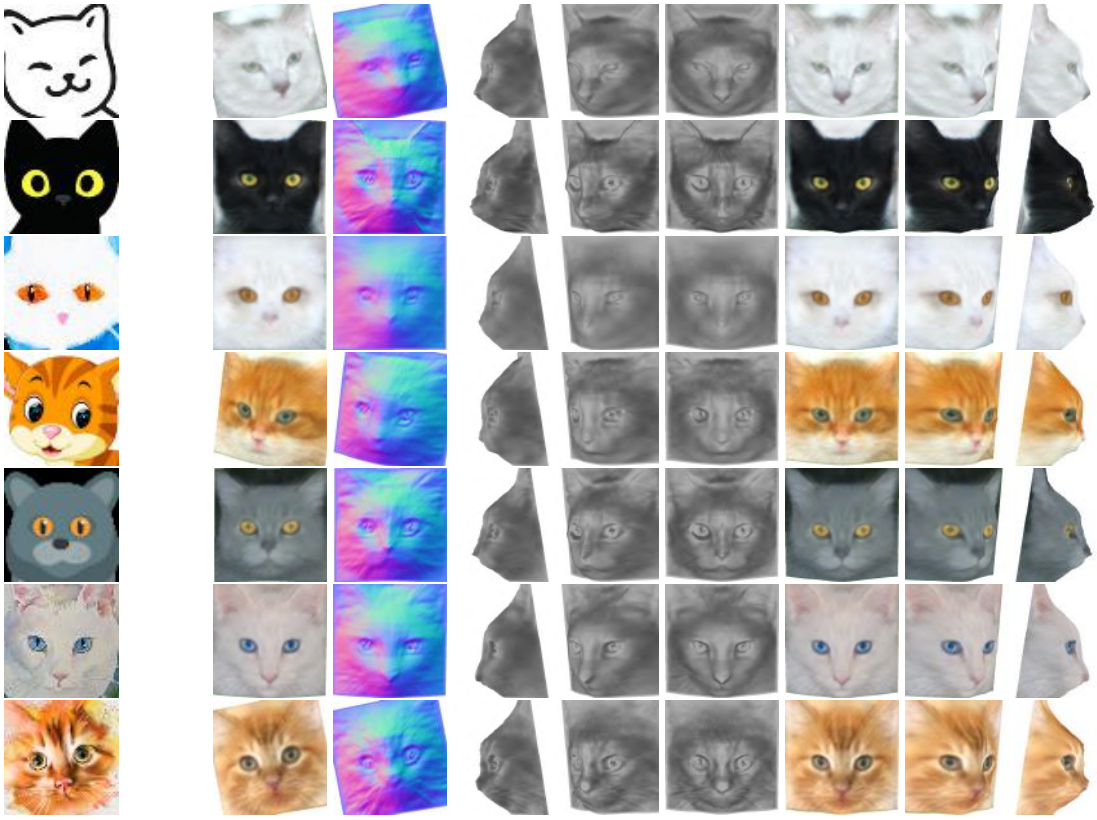
Figure 12: **Reconstruction of face paintings.**

Figure 13: **Reconstruction of abstract faces.**



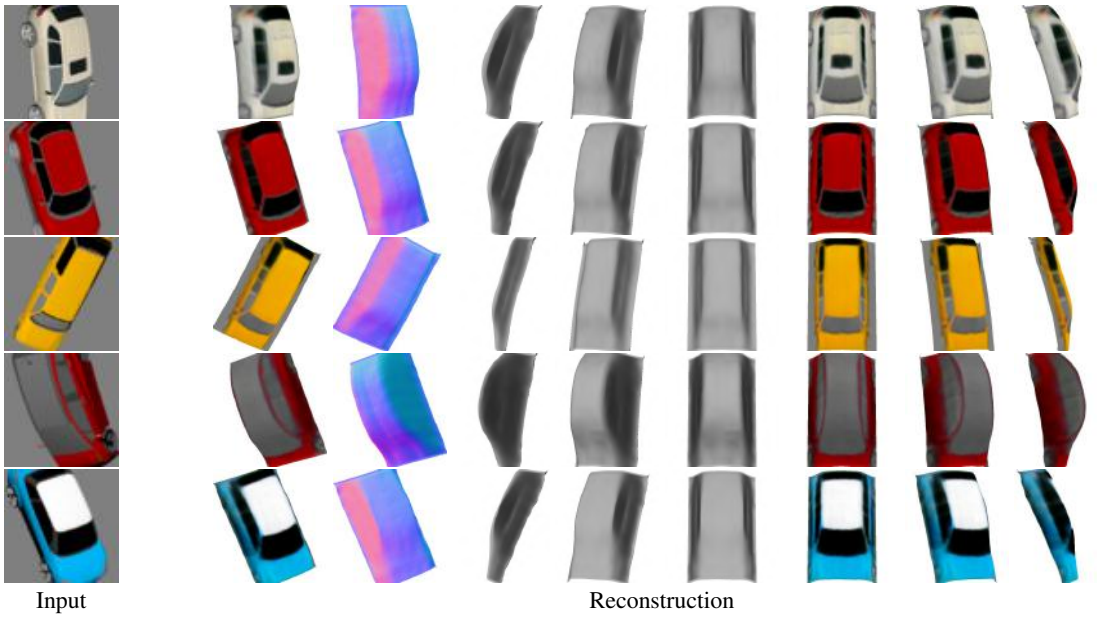
Input Reconstruction

Figure 14: **Reconstruction of cat faces.**



Input Reconstruction

Figure 15: **Reconstruction of abstract cats.**



Input Reconstruction

Figure 16: **Reconstruction of synthetic cars.**