Toward Realistic Single-View 3D Object Reconstruction with Unsupervised Learning from Multiple Images

Long-Nhat Ho[1] Anh Tuan Tran[1,2] Quynh Phung[1] Minh Hoai[1,3]
[1]VinAI Research, Hanoi, Vietnam, [2]VinUniversity, Hanoi, Vietnam,
[3]Stony Brook University, Stony Brook, NY 11790, USA
{v.nhathl,v.anhtt152,v.quynpt29,v.hoainm}@vinai.io

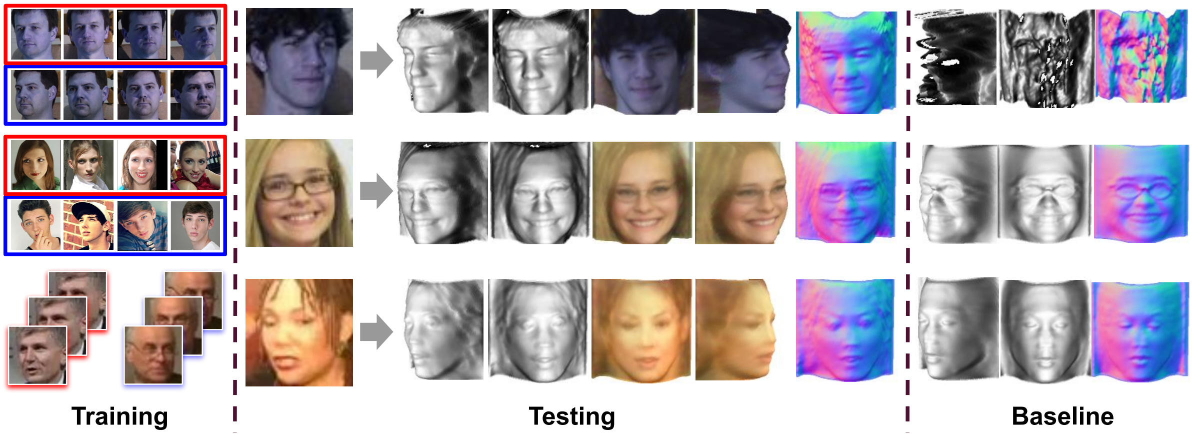


Figure 1: We propose a system to learn single-view 3D object reconstruction from multi-image datasets in an unsupervised manner. From top to bottom: Multi-PIE [23], CASIA-WebFace [54], and Youtube Faces [50] datasets. From left to right: different training data structures, our corresponding 3D reconstructions at test time, and the baseline [52] 3D results. For each 3D model, we provide two textureless views, two textured views, and the canonical normal map.

Abstract

Recovering the 3D structure of an object from a single image is a challenging task due to its ill-posed nature. One approach is to utilize the plentiful photos of the same object category to learn a strong 3D shape prior for the object. This approach has successfully been demonstrated by a recent work of Wu et al. (2020), which obtained impressive 3D reconstruction networks with unsupervised learning. However, their algorithm is only applicable to symmetric objects. In this paper, we eliminate the symmetry requirement with a novel unsupervised algorithm that can learn a 3D reconstruction network from a multi-image dataset. Our algorithm is more general and covers the symmetry-required scenario as a special case. Besides, we employ a novel albedo loss that improves the reconstructed details and realisticity. Our method surpasses the previous work in both quality and robustness, as shown in experiments on datasets of various structures, including single-view, multi-view, image-collection, and video sets. Code is available at: https://github.com/VinAIResearch/LeMul.

1. Introduction

Images are 2D projections of real-world 3D objects, and recovering the 3D structure from a 2D image is an important computer vision task with many applications. Most image-based 3D modeling methods rely on multi-view inputs [42, 43, 16, 17, 11, 57, 48, 21], requiring multiple images of the target object captured from different views. However, these methods are not applicable to the scenarios where only a single input image is available, which is the focus of our work in this paper. This problem is called single-view 3D reconstruction, and it is ill-posed since an image can be a projection of infinitely many 3D shapes. Interestingly, humans are very good at estimating the 3D structure of any known class object from a single image; we can even predict how it looks in unseen views. This is perhaps because humans have strong prior knowledge about the 3D shape and texture of the object class in consideration. Inspired by this observation, many category-specific 3D modeling methods have been proposed for specific object categories such as faces [3, 40, 59, 46, 39, 44, 47, 13], hands [60, 30, 4, 18], and bodies [34, 24].

In this paper, instead of focusing on any individual category, we aim to develop a general framework that can work for any object category, as long as there are many images from that category to train a single-view 3D reconstruction network. Furthermore, given the difficulty of acquiring 3D ground-truth annotation, we also aim to develop an unsupervised learning method which does not require the ground-truth 3D structures for the objects in the training images. However, this is a challenging problem due to the huge variation of the training images, regarding their viewpoint, appearance, illumination, and background.

A recent study [52] made a break-though in solving this problem with a novel end-to-end trainable deep network. Their network consisted of several modules to regress the image formation's components, including the object's 3D shape, texture, viewpoint, and lighting parameters, so that the rendered image was similar to the input. The modules were trained in an unsupervised manner on image datasets. They assumed a single image per training example, so it was still highly under-constrained. To make this training procedure converge, the authors proposed using the symmetry constraint. Their system successfully recovered 3D shape of human faces, cat faces, and synthetic cars after training on respective datasets. For convenience, from now on we will call this Learning from Symmetry method as ***LeSym***.

While showing good initial results, LeSym has several limitations. First, it requires the target object to be almost symmetric, severely restricting its applicability to certain object classes. For highly asymmetric objects, this method does not work, and for nearly symmetric objects, it would not preserve the asymmetric details. Second, with a strong symmetry constraint, an incorrect mirror line estimation would lead to unrealistic 3D reconstruction. Some examples and detailed discussions on these issues can be found in Sec. 4. Third, when multiple images of the same object in the training dataset are available, LeSym cannot correlate and leverage these images to improve the reconstruction accuracy and stability. This is a drawback because there are many imagery datasets that contain multiple images for each object. For example, multiview stereo datasets have photos of each object captured at different views. Some datasets instead have multiple pictures of the same view but with different lighting conditions or focal lengths. Facial datasets often have multiple images for each person, and video datasets have a large number of frames covering the same object in each video.

In this paper, we propose a more general framework, called ***LeMul***, that effectively Learns from Multi-image datasets for more flexible and reliable unsupervised training of 3D reconstruction networks. It employs loose shape and texture consistency losses based on component swapping across views. This is an "unsupervised" method since it does not require any 3D ground-truth data in training. Although it exploits multiple images per training instance, these images are so diverse and cannot be combined in traditional approaches to form any 3D supervision. LeMul can cover the symmetric object addressed in LeSym by using the original and the flipped image with less regularized results. More importantly, it handles a wider range of training datasets and object classes.

Besides, we employ an albedo loss in LeMul, which accurately recovers fine details of the 3D shape. This loss is inspired by a well-known Shape-from-Shading (SfS) literature [32]. It greatly improves the realisticity of the reconstructed 3D model, sometimes approaching laser-scan quality, from a low-res single image input.

In short, our contributions are: (1) we introduce a general framework, called LeMul, that can exploit multi-image datasets in learning 3D object reconstruction from a single image without the symmetry constraint; (2) we employ shape and texture consistency losses to make that unsupervised learning converge; (3) we apply an albedo loss to improve realisticity of the reconstruction results; (4) LeMul shows state-of-the-art performance, qualitatively and quantitively, on a wide range of datasets.

2. Related Work

In this section, we briefly review the existing image-based 3D reconstruction approaches, from classical to deep-learning-based algorithms.

Multi-view 3D reconstruction. This approach requires multiple images of the target object captured at different viewpoints. It consists of two sub-tasks: Structure-from-Motion (SfM) and Multi-view Stereo (MVS). SfM estimates from the input images the camera matrices and a sparse 3D reconstructed point-cloud [42, 43]. SfM requires robust keypoints extracted from each input view for matching and reconstruction. MVS assumes known camera matrices for a dense 3D reconstruction [17, 16]. These tasks are often combined to form end-to-end systems: SfM provides camera matrix estimation as an input to MVS [51]. These approaches were well-studied in classical literature, and they have been further improved with deep learning [57, 48, 21]. These methods, however, are unfit for our objective of 3D reconstruction from a single image at inference time. Even at training time, they hardly work with our in-the-wild inputs with low image quality, diverse capturing conditions, and freely non-rigid deformation.

Shape from X is another common 3D modeling approach that relies on a specific aspect of the image(s) such as silhouettes [27], focus [12], symmetry [31, 14, 45, 41], and shading [55, 26, 2, 32]. These methods only work on restricted conditions, thus do not apply to in-the-wild data. We focus on two latter directions since they are applicable to our problem. Shape-from-symmetry assumes the target object is symmetric, thus using the original and flipped image as a stereo pair for 3D reconstruction. Shape-from-shading (SfS) relies on some shading model, normally Phong shading [36] or Spherical Harmonic Lighting [22], and solves an inverse rendering problem to decompose image's intrinsic components, including 3D shape, albedo, and illumination. SfS methods often either refine an initial 3D [26, 32] or solve an optimization problem with multiple heuristic constraints [2]. We are particularly interested in [32], which employs bilateral-like loss functions to obtain fine-details on an initial raw depth-map.

Deep-learning-based 3D modeling. Deep learning provides a powerful tool to handle challenging computer vision problems, including 3D reconstruction from a single image. Some studies managed to solve the monocular depth estimation [9, 53, 38, 15] from a single image via supervised learning on ground-truth datasets. Some other studies learned a 3D shape representation from 3D datasets, using a generative model such as GAN or VAE, and fit it into the input image either with or without supervision [5, 20, 58, 58, 28]. These methods, however, require ground-truth data for supervision or 3D shape datasets for prior learning. They are not unsupervised and cannot handle a new object class that has no available 3D data.

Category-specific 3D reconstruction. Some research focus on reconstructing 3D models of a specific object class, such as human faces [3, 40, 59, 46, 39, 44, 47, 13], hands [60, 30, 4, 18], and bodies [34, 24]. The 3D modeling process often heavily relies on well-defined shape priors. For instance, early 3D face modeling studies used simple PCA models learned from facial landmarks such as Active Appearance Model (AAM) [6, 7] and Constrained Local Model (CLM) [8, 1]. Later, statistical models for 3D face shape and albedo learned from 3D face scans, called 3D Morphable Models (3DMMs) [35, 19], were used as an effective prior in 3D face modeling algorithms [3, 40, 59, 46, 39, 44]. Recently, many works have explored other 3D face presentations, such as non-linear 3DMMs [47] or GCN-based features [37, 49]. Instead of learning specific models based on characteristics of each object class, we target a general framework that can extract 3D shape prior for any class just from in-the-wild images.

LeSym [52] was the first work that could handle the task of 3D modeling from a single image in a general and unsupervised manner. It followed the SfS approach to extract the image's intrinsic components, including 3D shape, texture, view, and illumination parameters. The network was trained to minimize the reconstruction loss, comparing the rendered image and the input, using a differentiable renderer on a large image set of same-class objects. The optimization problem was under-constrained, so the authors assumed symmetry on the target object and incorporated the flipped image as in Shape-from-Symmetry algorithms. LeSym showed impressive reconstruction results on human faces, cat faces, and synthetic cars. However, the symmetry assumption strongly regularized the estimated 3D models and restricted LeSym's applications. Also, the reconstructed 3D models are still raw, with many details missing.

3. Learning from Multi-Image Datasets

3.1. Overview

We revise the mechanism used in LeSym to get LeMul as a more general, effective, and accurate unsupervised 3D reconstruction method. Two key ideas in our proposal are: a multi-image based unsupervised training and a novel albedo loss. The system overview is illustrated in Fig. 2.

Unlike LeSym, we do not require the modeling target to be symmetric. Instead, we assume more than one image for each object in the training data. We run the network modules over each image and enforce shape and albedo consistency. Note that having a single image of a symmetric object is a special case of ours; we can simply use the original and flipped input as two images of each training instance, and the 3D model consistency will enforce the object's symmetry. Moreover, this configuration can account for many other common scenarios such as multi-view, multi-exposure, multi-frame datasets. The multi-image configuration is only needed in training. During inference, the system can output a 3D model from a single input image.

Consider a training example and let $\{\mathbf{I}_i\}$ denote the set of M images of an object taken at different conditions. Each image $\mathbf{I}_i \in R^{H\times W\times 3}$ can be decomposed into four components $(\hat{d}_i, \hat{a}_i, \hat{l}_i, \hat{v}_i)$. The first two components represent the object's 3D model in a **canonical** view that is independent to camera pose, with $\hat{d}_i \in \mathbb{R}^{H\times W}$ is the depth-map and $\hat{a}_i \in \mathbb{R}^{H\times W\times 3}$ is the albedo-map. The latter components model the capturing conditions, with $\hat{l}_i \in \mathbb{R}^L$ is a vector of L illumination parameters and $\hat{v}_i \in \mathbb{R}^6$ is the viewing vector. The image is formed by a shading function $\mathcal{R}$:

$$\mathbf{I}_i = \mathcal{R}(\hat{d}_i, \hat{a}_i, \hat{l}_i, \hat{v}_i) + \eta_i. \tag{1}$$

where η_i is the noise term for factors such as background clutter and occlusions. The shading model $\mathcal{R}$ is a differentiable renderer [25], which uses a perspective projection camera, Phong shading model, and Lambertian surface assumption. There are L=4 illumination parameters, including the weighting coefficients for the ambient term k_s and the diffuse term k_d and the light direction (l_x, l_y). Other details are described in [52].

Our **decomposing network** consists of four modules to estimate the four intrinsic components $(\hat{d}_i, \hat{a}_i, \hat{l}_i, \hat{v}_i)$ of an input image I_i. We denote these modules as $\mathcal{F}_d, \mathcal{F}_a, \mathcal{F}_l$, and $\mathcal{F}_v$ respectively. $\mathcal{F}_d$ and $\mathcal{F}_a$ translate the input to output maps that have the same spatial resolution. $\mathcal{F}_l$

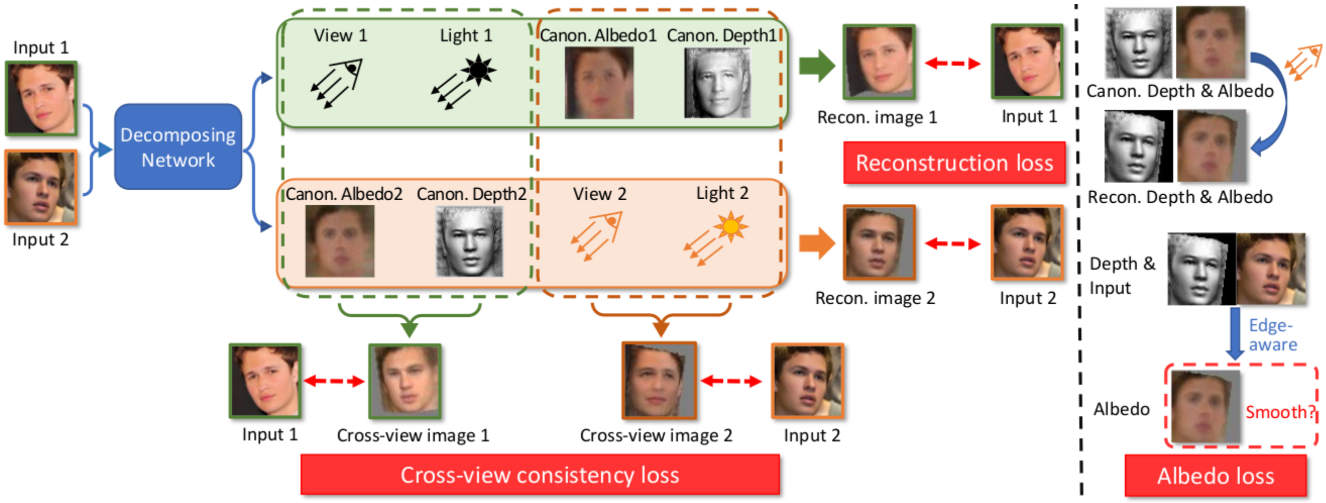


Figure 2: **Overview of the proposed system.** We train a decomposing network to optimize different loss components. Note that we omit the confidence maps in this figure for simplicity. Also, we use diffuse shading images to visualize depth maps.

and $\mathcal{F}_v$ are regression networks that output parameter vectors. The outputs of these modules components, denoted as (d_i, a_i, l_i, v_i), are used to reconstruct the input image:

$$\mathbf{I}_i^r = \mathcal{R}(\mathcal{F}_d(\mathbf{I}_i), \mathcal{F}_a(\mathbf{I}_i), \mathcal{F}_l(\mathbf{I}_i), \mathcal{F}_v(\mathbf{I}_i)) = \mathcal{R}(d_i, a_i, l_i, v_i).$$

There are two desired criteria: (1) the reconstructed image $\mathbf{I}_i^r$ should be similar to the input $\mathbf{I}_i$; (2) for any pair of images coming from the same training sample $\mathbf{I}_i$ and $\mathbf{I}_j$, the estimated canonical depth and albedo maps (d_i, a_i) and (d_j, a_j) should be almost similar and interchangeable. These criteria can be formulated into two losses $\mathcal{L}^{rec}$ and $\mathcal{L}^{rec}_{cross}$ respectively. Furthermore, we employ novel loss functions, called albedo losses, inspired by [32] to further improve the reconstruction of fine details. These losses follow the same-view and cross-view settings, and we denote them as $\mathcal{L}^{al}$ and $\mathcal{L}^{al}_{cross}$. The total training loss, therefore, will be:

$$\mathcal{L} = \mathcal{L}^{rec} + \lambda^{cross}\mathcal{L}^{rec}_{cross} + \lambda^{al}(\mathcal{L}^{al} + \lambda^{cross}\mathcal{L}^{al}_{cross}), \quad (2)$$

with λ^{al} and λ^{cross} being weighting hyper-parameters. We will now discuss each loss component above.

3.2. Reconstruction loss

We inherit this loss from LeSym. It enforces the reconstructed image to be similar to the input. To discard the effect of the noise η, another sub-network, called $\mathcal{F}_c$ is used to regress a pair of confidence maps (c^{l_1}, c^{pe}) that weigh the pixels in computing the reconstruction loss. The total reconstruction loss is summed over all input views:

$$\mathcal{L}^{rec} = \sum_{i=1}^{M}\left(\mathbb{L}^{l_1}(\mathbf{I}_i, \mathbf{I}_i^r, c_i^{l_1}) + \lambda^{pe}\mathbb{L}^{pe}(g(\mathbf{I}_i), g(\mathbf{I}_i^r), c_i^{pe})\right),$$

where $\mathbb{L}^{l_1}$ and $\mathbb{L}^{pe}$ are functions to compute the l_1 and perceptual loss components, g is a function to extract the k-th layer feature $\mathbf{f}$ of a VGG-16 network pre-trained on ImageNet, and λ^{pe} is a weighting hyper-parameter.

Assuming Gaussian distributions, the mentioned loss components have detailed expressions as following:

$$\mathbb{L}^{l_1}(\mathbf{I}, \mathbf{I}', c) = \frac{1}{|\Omega|}\sum_{p\in\Omega}\frac{\sqrt{2}|\mathbf{I}(p) - \mathbf{I}'(p)|_1}{c(p)} + \ln(c(p)), \quad (3)$$

$$\mathbb{L}^{pe}(\mathbf{f}, \mathbf{f}', c) = \frac{1}{|\Omega_k|}\sum_{p\in\Omega_k}\frac{\|\mathbf{f}(p) - \mathbf{f}'(p)\|^2}{2(c(p))^2} + \ln(c(p)), \quad (4)$$

with Ω and Ω_k as pixel sets in image and feature space.

3.3. Cross-view consistency loss

Reconstruction loss alone is not enough to constrain the reconstruction outcome. Since we have multiple images per training instance, we can enforce the reconstructed 3D models (d, a) to be consistent via a cross-view consistency loss.

In theory, we can simply minimize the distance $\sum_{i\neq j}(\|d_i - d_j\| + \|a_i - a_j\|)$, but we found it ineffective in practice, making the training unstable to converge. Instead, we propose to implement the consistency loss based on a component swapping mechanism. For each pair of views $i \neq j$, we can swap the estimated 3D model from one view (d_j, a_j) to the other to render a cross-model image:

$$\mathbf{I}_{ij}^r = \mathcal{R}(d_j, a_j, l_i, v_i). \quad (5)$$

This image should be almost the same as the input $\mathbf{I}_i$. Similar to the reconstruction loss, we employ some confidence maps for loss computation. However, these maps correlate two input images $(\mathbf{I}_i, \mathbf{I}_j)$, requiring another confidence network. We call this network $\mathcal{F}_{cc}$ that inputs the image pair $(\mathbf{I}_i, \mathbf{I}_j)$ stacked by channels and returns a pair of confidence maps $(c^{l_1}_{ij}, c^{pe}_{ij})$. The cross-view loss item for this image pair can be computed as follows:

$$\mathcal{L}^{rec}_{cross}(i, j) = \mathbb{L}^{l_1}(\mathbf{I}_i, \mathbf{I}^r_{ij}, c^{l_1}_{ij}) + \lambda^{pe}\mathbb{L}^{pe}(g(\mathbf{I}_i), g(\mathbf{I}^r_{ij}), c^{pe}_{ij}).$$

We can compute the cross-view entropy loss for all pairs of $i \neq j$, but this can be computationally expensive if M is large. For computational efficiency, we select the first view as a pivot and use only the pairs related to the first view:

$$\mathcal{L}^{rec}_{cross} = \sum_{i=2}^{M}(\mathcal{L}^{rec}_{cross}(i, 1) + \mathcal{L}^{rec}_{cross}(1, i)). \tag{6}$$

3.4. Albedo losses

Although the 3D reconstructed shapes obtained with the above losses are reasonably accurate already, the 3D shapes tend to be over-smooth with many fine details of the 3D surface being inaccurately transferred to the albedo map. For sharper 3D reconstruction, we apply a regularization on the albedo map to avoid overfitting to pixel intensities. This regularization should guarantee that the albedo is smooth at non-edge pixels while preserving the edges. Following [32], we implement such regularization by albedo loss terms.

An albedo loss requires three aligned inputs, including an input image $\mathbf{I}$ and the corresponding maps for depth d and albedo a. It enforces smoothness on a:

$$\mathbb{L}^{al}(\mathbf{I}, a, d) = \frac{1}{|\Omega|}\sum_{p\in\Omega}\Big\|\sum_{p_k\in\mathcal{N}(p)} w^c_k w^d_k (a(p) - a(p_k))\Big\|^2.$$

where $\mathcal{N}(p)$ defines the neighbors of a pixel p, w^c_k is the intensity weighting term:

$$w^c_k = \exp\left(-\frac{\|I(p) - I(p_k)\|^2}{2\sigma^2_c}\right), \tag{7}$$

and w^d_k is the depth weighting term:

$$w^d_k = \exp\left(-\frac{\|d(p) - d(p_k)\|^2}{2\sigma^2_d}\right). \tag{8}$$

The weighting terms suppress the effect of neighbor pixels that likely come from other regions due to a large gap in intensity/depth compared with the current one. We use σ_c and σ_d to control the allowed intensity and depth discontinuity.

Note that the three inputs of the albedo loss needed to be aligned pixel-by-pixel. We keep the original input $\mathbf{I}$, which is at an estimated view v. Therefore, we cannot use the canonical maps (d, a) directly, so we transform them to the view v. This process can be done by a warping function $\mathcal{W}$. This function first computes the 3D shape from the canonical depth d, then project and render it at the view v. The outputs are transformed depth and albedo maps:

$$(d^v, a^v) = \mathcal{W}((d, a), d, v). \tag{9}$$

Similar to the previous loss terms, we compute the albedo loss in same-view and cross-view settings:

$$\mathcal{L}^{al} = \sum_{i=1}^{M}\mathbb{L}^{al}(\mathbf{I}_i, a^{v_i}_i, d^{v_i}_i), \tag{10}$$

$$\mathcal{L}^{al}_{cross} = \sum_{i=2}^{M}\left(\mathbb{L}^{al}(\mathbf{I}_1, a^{v_1}_i, d^{v_1}_i) + \mathbb{L}^{al}(\mathbf{I}_i, a^{v_i}_1, d^{v_i}_1)\right).$$

4. Experiments

4.1. Experimental setups

4.1.1 Implementation details

We implemented our system in PyTorch. The networks $\mathcal{F}_d, \mathcal{F}_a, \mathcal{F}_l, \mathcal{F}_v$, and $\mathcal{F}_c$ had the same structure as in the official released code of LeSym[1]. The cross-view confidence network $\mathcal{F}_{cc}$ was similar to $\mathcal{F}_c$, except for having six input channels instead of three. In all experiments, we used the same input image size H=W=64. The hyper-parameters were set as λ^{pe}=1, λ^{cross}=λ^{al}=0.5, σ_c=0.05 and σ_d=2. The networks were jointly trained with Adam optimizer at a fixed learning rate 0.0001 until convergence.

4.1.2 Datasets

To evaluate the proposed algorithm, we run experiments on datasets with various capturing settings and data structures (single-view, multi-view, image-collection, or video):

BFM is a synthetic dataset of 200K human face images proposed by LeSym. Each image is rendered with a 3D shape and texture randomly sampled from the Basel Face Model [35], a random view, and one of the spherical harmonics lights estimated from CelebA images [29]. Besides RGB images, the ground-truth 3D depth-maps are also provided. We use this dataset to quantitatively evaluate our approach as well as comparing it with other baselines.

CelebA [29] is a popular facial dataset of more than 200K celebrity images. The images were captured under in-the-wild conditions. It is split into three subsets for training, validation, and testing with 162K, 20K, and 20K images, respectively. We use this dataset to compare LeMul and LeSym under the "single-view" and "symmetric-objects" settings. We generate two image inputs for each training instance, including the original and the flipped image.

Cat Faces is a dataset of 11.2K images capturing cat faces in-the-wild. This dataset was constructed in LeSym by combining two previous datasets [56, 33]. This set is split into 8930 training and 2256 testing instances. This dataset is also under the "single-view" and "symmetric-objects" settings, and its two-view data is formed similar to CelebA.

[1] https://github.com/elliottwu/unsup3d

Multi-PIE [23] is a large human-face dataset captured in studio settings. It contains more than 750K images of 337 people involved in from one to four different recording sessions. In each session, each subject has a collection of images captured at 15 view-points, 19 illuminations, and with several expressions. We excluded images with extreme light or overwhelmed expression and selected ones at three viewing angles corresponding to frontal, 15°-to-the-left, and 15°-to-the-right views, to form a multi-view image set. Each training instance is a set of three images of each person, captured at the selected views. We use random illumination, causing three input views drastically different and unable to be used by traditional multi-view stereo methods.

CASIA-WebFace [54] has 500K face images of 10K people collected from the Internet. Each person has on average 50 in-the-wild images with drastically different conditions. We keep the last 200 subjects for testing and use the rest for training. In each training epoch, for each subject, we randomly select M=3 images of that person regardless of pose, expression, and illumination to form a training example.

YouTube Faces (YTF) [50] is a video dataset that consists of 3425 videos of 1595 people. The videos have low-quality frames, which were severely degraded by video compression. Many videos are also bad for 3D face modeling, with the target faces at non-frontal views and barely moving. Still, we aim to evaluate our method on such extreme conditions. For each video, we extract the frames and crop them around the target faces. We split the videos for training (3299) and testing (126). Similar to CASIA, in each training epoch and with each video, we randomly select M=3 frames to form a training instance.

Quantitative Metrics. For fair comparison results, we use the same metrics used in LeSym. The first metric is *Scale-Invariant Depth Error* (SIDE) [10], which computes the standard deviation of the difference between the estimated depth map at the input view and the ground-truth depth map at the log scale. However, we argue that this metric is not a strong indicator of the reconstruction quality. A reasonable error in the object distance estimation, while not affecting the projected image, can cause SIDE varying. In contrast, it is ineffective in evaluating the reconstructed surface quality. We can smooth out the depth-map or add small random noise to it but cause a minimal change in SIDE value.

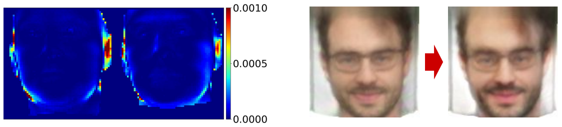
Instead, we focus on the second metric, which is the ***Mean Angle Deviation* (MAD) [52]** between normal maps computed from estimated depth map d^v and ground-truth depth map d^*. It can measure how well the surface is captured and is sensitive to surface noise.

4.2. Quantitative experiments

In this section, we perform quantitative evaluations on the BFM dataset with provided ground-truth data.

No	Baseline	SIDE(x10^{-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)	LeSym	**0.793±0.140**	16.51 ± 1.56
(5)	LeMul (proposed)	0.834 ± 0.169	**15.49±1.50**

Table 1: BFM results comparison with baselines.



(a) (b)

Figure 3: **Qualitative analyses.** (a) LeMul vs. LeSym (SIDE) and (b) Texture refinement.

No	Method	SIDE(x10^{-2})↓	MAD(deg.)↓
(1)	Baseline [52]	0.793 ± 0.140	16.51 ± 1.56
(2)	+ multi-view	**0.728±0.135**	15.73 ± 1.54
(3)	+ albedo loss	0.899 ± 0.217	16.35 ± 1.79
(4)	+ mul+al (full)	0.834 ± 0.169	**15.49±1.50**

Table 2: **Ablation studies** on BFM dataset

BFM results. We trained and tested our algorithm on the BFM dataset, and the results are reported in Table 1, along with some baselines: (1) supervised 3D reconstruction network as the upper bound, (2) a dummy network returning a constant null depth, (3) a dummy network producing a constant mean depth computed over the ground-truth one, and (4) LeSym. As can be seen, LeMul outperforms the dummy networks by a wide margin. Compared with LeSym, it achieves a better MAD number with 1° decrease, implying better reconstructed 3D surfaces with details recovered.

As for SIDE, we examine the error maps and find that LeMul provides a better overall depth estimation. However, the outliers, particularly on the face boundary or outer components (ears, neck), are more unstable and skew the average score. If we compute the SIDE metric over the facial region bounded by Dlib's 68-landmarks, LeMul has a lower error (***0.00534***) compared with LeSym (***0.00564***), confirming this observation. Fig. 3a provides a common scenario in which LeMul provides a lower error on most facial areas but higher errors on the boundary and an ear.

Ablation studies. We run ablation experiments to evaluate each proposed component's contribution to our result on the BFM dataset. From LeSym as the baseline, we can modify it to follow our multi-view scheme or integrate the albedo losses. As reported in Table 2, each of our proposals positively affects the MAD numbers. We achieve the best reconstructed 3D surfaces when combining both techniques.

4.3. Qualitative experiments

We qualitatively compare our method to the LeSym baseline on the mentioned datasets in Fig. 1 and Fig. 4. In all experiments, LeSym uses each single image as a training instance and applies the symmetry constraint. Our method also assumes that symmetry property on BFM, CelebA, and Cat Faces by using pairs of original and flipped images as training instances. However, on Multi-PIE, CASIA-WebFace, and Youtube Faces, we completely drop that assumption and use multi-image examples in training.

Three symmetry-assumed datasets. On BFM, CelebA, and Cat Faces, both LeSym and LeMul can reconstruct reasonable 3D models. However, thanks to the albedo loss, LeMul can recover more 3D details such as human hairs, beards, and cat furs. The 3D models are well recognizable even without texture.

Multi-PIE results. LeSym completely collapsed, perhaps due to the limited number of poses and the asymmetric lights. LeMul, instead, performed well on this data configuration with high quality produced 3D models.

CASIA-WebFace results. LeMul can learn well the 3D face structure. It is impressive since the images used in each training example are wildly different; they are even challenging for humans to correlate, as illustrated in Fig. 1 (second row). In both Fig. 1 and Fig. 4, LeMul can capture asymmetric details such as one-sided hairstyle and lopsided smile. In contrast, LeSym over-regulated the 3D shapes with the symmetry constraint, producing incorrect 3Ds.

Youtube Faces results. This dataset is pretty challenging to our training due to low-quality images and limited variation between frames in each video. Still, LeMul manages to converge and produce reasonable results at test time. When the input image is not too blurry, LeMul can reconstruct a 3D model with more details compared with LeSym, while it does not suffer from the symmetry assumption.

4.4. User surveys

We further compared our method with the baseline via user surveys. We skipped this test on BFM, which was already used in quantitative evaluations, and Multi-PIE, in which LeSym completely failed. For each remaining dataset, we created a survey with 30 testing images randomly sampled from the respective test set. We generated two 3D models, estimated by LeSym and LeMul, for each image and produced corresponding videos to illustrate these models in various viewing angles. Each tester was asked to pick which model was better. At least 40 people took each survey, leading to at least 1200 answers per dataset.

We report the rate that each method is selected in Table 3. LeMul outperforms LeSym on CelebA and CASIA datasets by a wide margin, showing that LeMul can recover better 3D models in good training conditions. Notably, it was selected near 80% of time on CASIA, proving the superiority of the multi-image setting over the symmetry constraint. It also beats LeSym on YTF and Cat Faces datasets but with smaller gaps. We found many YTF frames too blurry, making both 3D models smooth and hard to compare. The small gap of LeMul over LeSym came from clear frames, which is still very meaningful. Finally, on Cat Faces, while our models are more detailed, some testers preferred smooth 3Ds from LeSym, decreasing our selected rate. This phenomenon suggests that it is not always good to have many details, opening a future research to improve our method.

Method	CelebA	CatFaces	CASIA	YTF
LeSym [52]	36.01	47.03	20.86	45.23
Ours	**63.99**	**52.97**	**79.14**	**54.77**

Table 3: **User survey results.** For each dataset, we report the rate (%) that each method is selected by the tester for providing a better 3D model.

4.5. In-the-wild tests

Finally, we run evaluation on in-the-wild facial images collected from the Internet. We select the LeMul model trained on the CASIA dataset since it can capture even the asymmetric details. In contrast, among LeSym models for human face, the released model trained on the CelebA dataset shows the best reconstruction quality. We compare these models on some in-the-wild images in Fig. 5. The 3D shapes generated from LeSym are often distorted by symmetry regulation. Our results, instead, look more natural and detailed. Particularly, LeMul can create a realistic-looking 3D model from a cartoon drawing (the fourth row).

4.6. Texture refinement.

We observe that the regressed texture with models trained on CASIA and YTF datasets is a bit blurry, possibly due to two reasons. First, these datasets have lower image quality compared to CelebA; many images have blur, noise, or JPEG artifacts. Second, the models have to learn the subject's albedo from vastly different inputs, causing blurry texture. We propose a simple solution to fix the second issue. After getting a trained model, we can finetune $\mathcal{F}_a$, $\mathcal{F}_l$, and $\mathcal{F}_c$ while freezing the other modules for a few epochs on single-image inputs of the same training set. As shown in Fig. 3b, the estimated texture is significantly improved. Note that this refinement preserves the high-quality 3D shape evaluated in previous experiments.

5. Discussions

In this paper, we present a novel system that shows the state-of-the-art 3D modeling quality in unsupervised learning for single-view 3D object reconstruction. The key insights are to exploit multi-image datasets in training and to

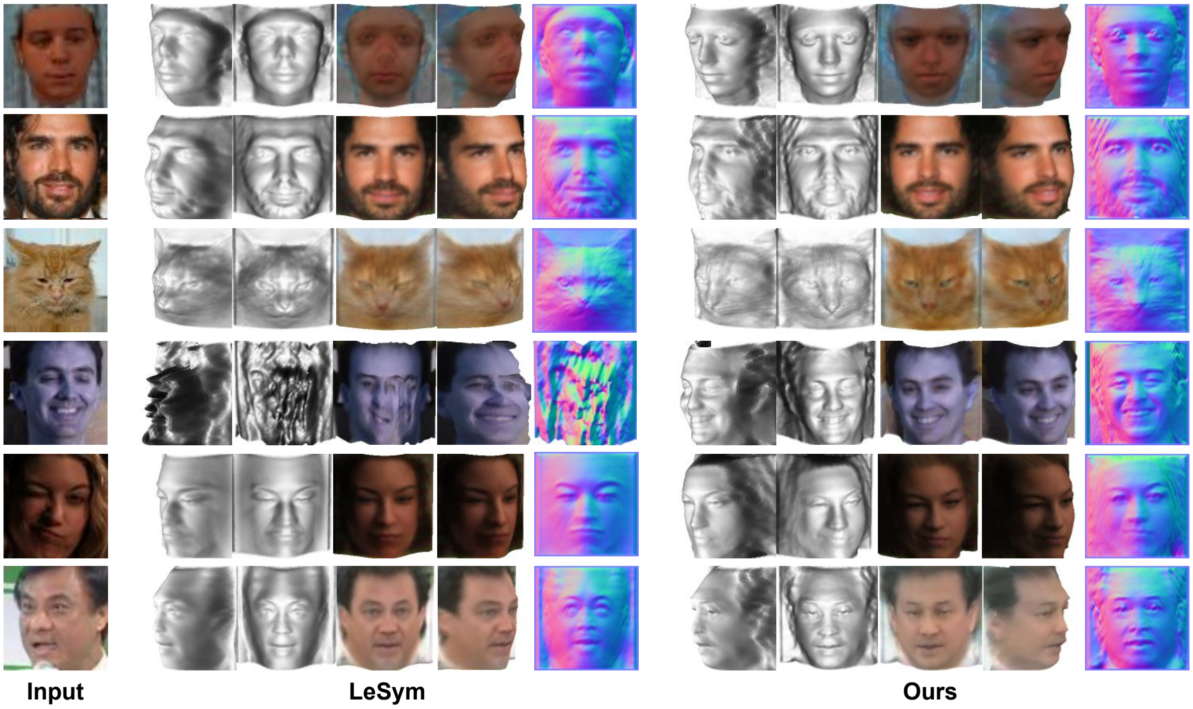


Figure 4: Comparing the reconstructed 3D models from the baseline method LeSym model LeSym [52] and ours. The datasets from top to bottom: BFM [52], CelebA [29], Cat Faces [52], Multi-PIE [23], CASIA-WebFace [54], and Youtube Faces [50]. For each 3D model, we provide two textureless views, two textured views, and the canonical normal map.

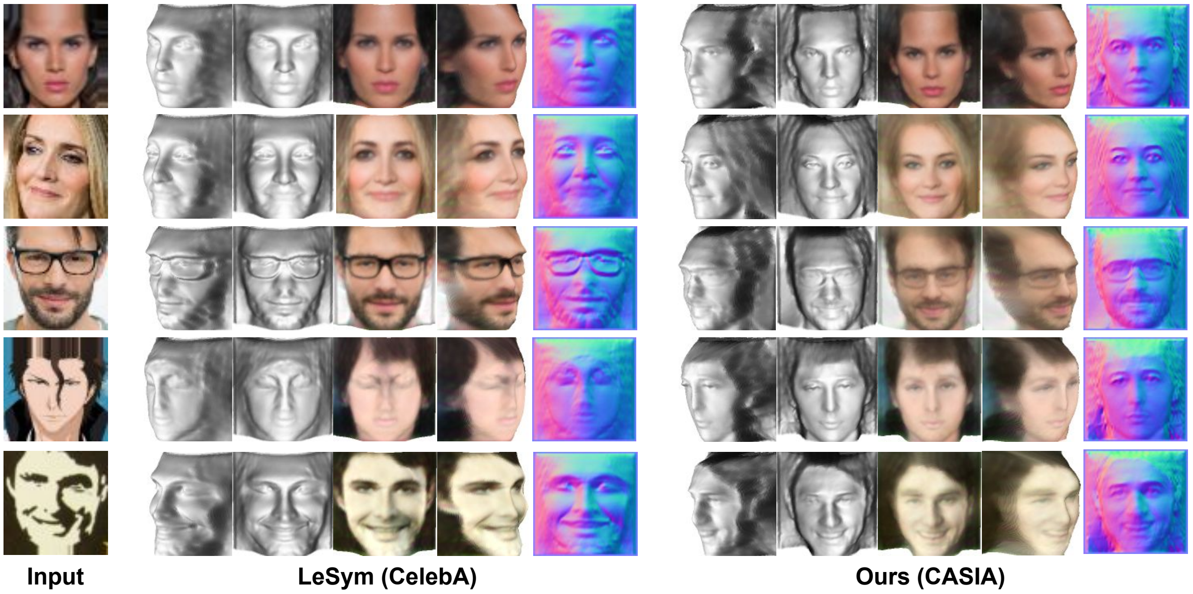


Figure 5: Reconstructed 3D models from in-the-wild images. We compare the baseline model LeSym [52] trained on CelebA dataset [29], and our method trained on CASIA-WebFace [54] dataset. For each 3D model, we provide two textureless views, two textured views, and the canonical normal map.

employ albedo losses for improved detailed reconstruction.

Our method can work on various training datasets ranging from single- and multi-view datasets to image collection and video data. However, a current limitation of our work is that the images of the target object need to be compatible to the depth-map representation, being primarily frontal view without self-occlusion. We plan to address this limitation in future work to increase the applicability of our method.

References

[1] Tadas Baltruvsaitis, Peter Robinson, and Louis-Philippe Morency. 3d constrained local model for rigid and non-rigid facial tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 3

[2] Jonathan T Barron and Jitendra Malik. Shape, illumination, and reflectance from shading. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1670–1687, 2014. 2, 3

[3] V. Blanz and T. Vetter. Morphable model for the synthesis of 3D faces. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*, 1999. 1, 3

[4] Adnane Boukhayma, Rodrigo de Bem, and Philip HS Torr. 3d hand shape and pose from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3

[5] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *Proceedings of the European Conference on Computer Vision*, 2016. 3

[6] Timothy F Cootes, Gareth J Edwards, and Christopher J Taylor. Active appearance models. In *Proceedings of the European Conference on Computer Vision*, 1998. 3

[7] Timothy F. Cootes, Gareth J. Edwards, and Christopher J. Taylor. Active appearance models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 681–685, 2001. 3

[8] David Cristinacce and Tim Cootes. Automatic feature localisation with constrained local models. *Pattern Recognition*, pages 3054–3067, 2008. 3

[9] David Eigen and Rob Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *Proceedings of the International Conference on Computer Vision*, 2015. 3

[10] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In Z. Ghahramani, M. Welling, C. Cortes, N. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*. Curran Associates, Inc., 2014. 6

[11] Olivier Faugeras, Quang-Tuan Luong, and Theo Papadopoulo. *The geometry of multiple images: the laws that govern the formation of multiple images of a scene and some of their applications*. 2001. 1

[12] P. Favaro and S. Soatto. A geometric approach to shape from defocus. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 406–417, 2005. 2

[13] 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*, 2018. 1, 3

[14] Alexandre RJ Franccois, Gérard G Medioni, and Roman Waupotitsch. Mirror symmetry 2-view stereo geometry. *Image and Vision Computing*, pages 137–143, 2003. 2

[15] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[16] Yasutaka Furukawa, Brian Curless, Steven M Seitz, and Richard Szeliski. Towards internet-scale multi-view stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2010. 1, 2

[17] Yasutaka Furukawa and Jean Ponce. Accurate, dense, and robust multi-view stereopsis (pmvs). In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2007. 1, 2

[18] Liuhao Ge, Zhou Ren, Yuncheng Li, Zehao Xue, Yingying Wang, Jianfei Cai, and Junsong Yuan. 3d hand shape and pose estimation from a single rgb image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3

[19] Thomas Gerig, Andreas Morel-Forster, Clemens Blumer, Bernhard Egger, Marcel Luthi, Sandro Schönborn, and Thomas Vetter. Morphable face models-an open framework. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2018. 3

[20] Rohit Girdhar, David F Fouhey, Mikel Rodriguez, and Abhinav Gupta. Learning a predictable and generative vector representation for objects. In *Proceedings of the European Conference on Computer Vision*, 2016. 3

[21] Clément Godard, Oisin Mac Aodha, and Gabriel J Brostow. Unsupervised monocular depth estimation with left-right consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[22] Robin Green. Spherical harmonic lighting: The gritty details. 2003. 3

[23] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker. Multi-pie. In *Proceedings of the International Conference on Automatic Face and Gesture Recognition*, 2008. 1, 6, 8

[24] Wen Jiang, Nikos Kolotouros, Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis. Coherent reconstruction of multiple humans from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 3

[25] Hiroharu Kato, Yoshitaka Ushiku, and Tatsuya Harada. Neural 3d mesh renderer. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 3

[26] Ira Kemelmacher-Shlizerman and Ronen Basri. 3d face reconstruction from a single image using a single reference face shape. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 394–405, 2010. 2, 3

[27] Jan J Koenderink. What does the occluding contour tell us about solid shape? *Perception*, pages 321–330, 1984. 2

[28] Abhijit Kundu, Yin Li, and James M Rehg. 3d-rcnn: Instance-level 3d object reconstruction via render-and-compare. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[29] Ziwei Liu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Deep learning face attributes in the wild. In *Proceedings of the International Conference on Computer Vision*, 2015. 5, 8

[30] Franziska Mueller, Florian Bernard, Oleksandr Sotnychenko, Dushyant Mehta, Srinath Sridhar, Dan Casas, and Christian Theobalt. Ganerated hands for real-time 3d hand tracking from monocular rgb. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3

[31] Dipti Prasad Mukherjee, Andrew Peter Zisserman, Michael Brady, and FT Smith. Shape from symmetry: Detecting and exploiting symmetry in affine images. *Philosophical Transactions of the Royal Society of London. Series A: Physical and Engineering Sciences*, pages 77–106, 1995. 2

[32] Roy Or-El, Guy Rosman, Aaron Wetzler, Ron Kimmel, and Alfred M Bruckstein. Rgbd-fusion: Real-time high precision depth recovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015. 2, 3, 4, 5

[33] Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and CV Jawahar. Cats and dogs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2012. 5

[34] Georgios Pavlakos, Vasileios Choutas, Nima Ghorbani, Timo Bolkart, Ahmed AA Osman, Dimitrios Tzionas, and Michael J Black. Expressive body capture: 3d hands, face, and body from a single image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019. 1, 3

[35] Pascal Paysan, Reinhard Knothe, Brian Amberg, Sami Romdhani, and Thomas Vetter. A 3d face model for pose and illumination invariant face recognition. In *Proceedings of International Conference on Advanced Video and Signal based Surveillance*, 2009. 3, 5

[36] Bui Tuong Phong. Illumination for computer generated pictures. *Communications of the ACM*, pages 311–317, 1975. 3

[37] Anurag Ranjan, Timo Bolkart, Soubhik Sanyal, and Michael J Black. Generating 3d faces using convolutional mesh autoencoders. In *Proceedings of the European Conference on Computer Vision*, 2018. 3

[38] Elisa Ricci, Wanli Ouyang, Xiaogang Wang, Nicu Sebe, et al. Monocular depth estimation using multi-scale continuous crfs as sequential deep networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1426–1440, 2018. 3

[39] Elad Richardson, Matan Sela, and Ron Kimmel. 3d face reconstruction by learning from synthetic data. In *Proceedings of International Conference on 3D Vision*, 2016. 1, 3

[40] Sami Romdhani and Thomas Vetter. Efficient, robust and accurate fitting of a 3D morphable model. In *Proceedings of the International Conference on Computer Vision*, 2003. 1, 3

[41] Sudipta N Sinha, Krishnan Ramnath, and Richard Szeliski. Detecting and reconstructing 3d mirror symmetric objects. In *Proceedings of the European Conference on Computer Vision*, 2012. 2

[42] Noah Snavely, Steven M Seitz, and Richard Szeliski. Photo tourism: exploring photo collections in 3d. In *Proceedings of the ACM SIGGRAPH Conference on Computer Graphics*. 2006. 1, 2

[43] Noah Snavely, Steven M Seitz, and Richard Szeliski. Modeling the world from internet photo collections. *International Journal of Computer Vision*, pages 189–210, 2008. 1, 2

[44] 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 International Conference on Computer Vision*, 2017. 1, 3

[45] Sebastian Thrun and Ben Wegbreit. Shape from symmetry. In *Proceedings of the International Conference on Computer Vision*, 2005. 2

[46] Anh Tran, Tal Hassner, Iacopo Masi, and Gérard Medioni. Regressing robust and discriminative 3D morphable models with a very deep neural network. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. Available: http://www.openu.ac.il/home/hassner/projects/CNN3DMM/. 1, 3

[47] Luan Tran and Xiaoming Liu. Nonlinear 3d face morphable model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 1, 3

[48] 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 *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[49] Huawei Wei, Shuang Liang, and Yichen Wei. 3d dense face alignment via graph convolution networks. *arXiv preprint arXiv:1904.05562*, 2019. 3

[50] Lior Wolf, Tal Hassner, and Itay Maoz. Face recognition in unconstrained videos with matched background similarity. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2011. 1, 6, 8

[51] Changchang Wu et al. Visualsfm: A visual structure from motion system. 2

[52] Shangzhe Wu, Christian Rupprecht, and Andrea Vedaldi. Unsupervised learning of probably symmetric deformable 3d objects from images in the wild. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2020. 1, 2, 3, 6, 7, 8

[53] Dan Xu, Wei Wang, Hao Tang, Hong Liu, Nicu Sebe, and Elisa Ricci. Structured attention guided convolutional neural fields for monocular depth estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018. 3

[54] Dong Yi, Zhen Lei, Shengcai Liao, and Stan Z Li. Learning face representation from scratch. *arXiv preprint arXiv:1411.7923*, 2014. Available: http://www.cbsr.ia.ac.cn/english/CASIA-WebFace-Database.html. 1, 6, 8

[55] Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 690–706, 1999. 2

[56] Weiwei Zhang, Jian Sun, and Xiaoou Tang. Cat head detection-how to effectively exploit shape and texture features. In *Proceedings of the European Conference on Computer Vision*, 2008. 5

[57] Tinghui Zhou, Matthew Brown, Noah Snavely, and David G Lowe. Unsupervised learning of depth and ego-motion from video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017. 1, 2

[58] Rui Zhu, Hamed Kiani Galoogahi, Chaoyang Wang, and Simon Lucey. Rethinking reprojection: Closing the loop for pose-aware shape reconstruction from a single image. In *Proceedings of the International Conference on Computer Vision*, 2017. 3

[59] Xiangyu Zhu, Zhen Lei, Xiaoming Liu, Hailin Shi, and Stan Z. Li. Face alignment across large poses: A 3D solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 1, 3

[60] Christian Zimmermann and Thomas Brox. Learning to estimate 3d hand pose from single rgb images. In *Proceedings of the International Conference on Computer Vision*, 2017. 1, 3