

End-to-end 3D shape inverse rendering of different classes of objects from a single input image

Shima Kamyab¹ and S. Zohreh Azimifar¹

¹Computer Science and Engineering and Information Technology, Shiraz university, Shiraz, Iran

November 17, 2017

Abstract

In this paper a semi-supervised deep framework is proposed for the problem of 3D shape inverse rendering from a single 2D input image. The main structure of proposed framework consists of unsupervised pre-trained components which significantly reduce the need to labeled data for training the whole framework. using labeled data has the advantage of achieving to accurate results without the need to predefined assumptions about image formation process. Three main components are used in the proposed network: an encoder which maps 2D input image to a representation space, a 3D decoder which decodes a representation to a 3D structure and a mapping component in order to map 2D to 3D representation. The only part that needs label for training is the mapping part with not too many parameters. The other components in the network can be pre-trained unsupervised using only 2D images or 3D data in each case. The way of reconstructing 3D shapes in the decoder component, inspired by the model based methods for 3D reconstruction, maps a low dimensional representation to 3D shape space with the advantage of extracting the basis vectors of shape space from training data itself and is not restricted to a small set of examples as used in predefined models. Therefore, the proposed framework deals directly with coordinate values of the point cloud representation which leads to achieve dense 3D shapes in the output. The experimental results on several benchmark datasets of objects and human faces and comparing with recent similar methods shows the power of proposed network in recovering more details from single 2D images.

1 Introduction

Inverse rendering stands for a specific class of machine learning techniques aim at recovering the properties of a 3D scene like camera's extrinsic parameters, scene lighting and the shape of the scene from some existing measurements (e.g. 2D images)[1, 21]. It has attracted high interest in the recent academic research due to wide applications in computer vision [15, 1, 33, 35, 38, 10, 26, 31, 32, 25, 27, 30, 13, 21, 7].

Inverse rendering is not a well posed problem because of existing infinite number of 3D structure which can result the same 2D image [10, 25]. Facing with such problems needs additional assumptions and prior knowledge about the class of 3D structures to be recovered from 2D images. For instance a statistical model can be used to determine which solution is likely and which is not for a specific class of objects [25].

The aim of Inverse rendering problem is actually inverting the image formation process called rendering. in Computer graphics, Rendering is expressed as an equation formulated as in 1 [19].

$$F = K(h + Gf) \quad (1)$$

where, F is the reflected radiance (light) from the surface (e.g. intensity of the pixels in the image), K represents the way the object surface reflects the lights in various spatial and angular positions, h describes the lighting of the scene, G is some measurement showing how light is transferred between other existing surfaces in the scene and f is the light which is reflected by other surfaces in scene affecting F . Knowing the quantities K , h , G , f means that the properties of the scene are known and F can be computed using these quantities and this is a direct problem

to solve. The inverse rendering problem arises when F or some information about it are available and the aim is to compute some values in the right-hand side of 1. In this paper, our focus is on recovering shape geometry of an object from a single 2D input image having different poses which is implicitly embedded in variable K as spatial properties of the surface in light reflection.

Inverse rendering problem as a machine learning task, can be faced through 3 main steps to be solved:

- Selection of a model (i.e. assuming a specific class of hypotheses)
- Defining a score criterion
- Defining a Search strategy

Selection of a model: Selection of a model is one of the main necessities for solving machine learning problems due to “no free lunch theorem” [34]. In the case of recovering 3D structures from 2D images, this step is more important because in such problems we deal with an ill-posed problem based on Hadmard definition for well-posed problems: there might be more than one solutions that give the same 2D image and the solution does not depends continuously on the data, which means that small errors in measurements may cause large errors in the solutions [24]. For solving such kinds of problems, some assumptions are usually made about the properties of the solutions to restrict the solution space to feasible regions. Selection of a suitable model for defining the target scene serves as some prior knowledge about the solution space, by which the search can be performed more reliable. Like using any other assumptions, this method may have some costs like losing some promising regions that are not spanned by the selected model. For inverse rendering problems, there are various types of models and prior knowledge for the target scene in the literature that make the existing methods different from each other: some approaches use additional information, like multiple images or video sequence or landmarks, about the scene to find a unique solution [38, 23, 9, 18], some of them work with a certain statistical model to determine the feasibility of the solutions [12] and some other perform some regressing techniques that relates the output measurements to 3D structures [10, 26, 31, 32, 25, 27, 13, 17, 30].

In this paper our focus is on the pose invariant recovering 3D structure of an object from a single 2D image with the help of a model describing the properties of that object. The main structure of our proposed framework is based on finding a relation between measurements, i.e. a single 2D image for each scene, and 3D structure of the scene constrained with the used model.

Defining a score criterion: After selection of a suitable model describing the solution space, the optimum solution should be searched in that space. For this aim, first a criterion should be defined which guides the search toward finding the optimum solution satisfying the constraints of the problem at hand. For instance, in the case of inverse rendering of human face from 2D image(s) for face recognition task, first, the main objective is to recover the distinctiveness characteristics of a face in an image and so a good solution should not be too generic so or too over determined and there should be some tradeoff between these two properties in the objective criterion [32]. In this paper, since the objective is to minimize the reconstruction error, we used Root Mean Square Error (RMSE) as loss function for training and evaluation.

Defining a search strategy: There exist different types of search mechanisms to search for a solution based on the defined score criterion. In the case of convex criteria, a closed form solution exists and the search can be done in linear time. However, the nature of inverse rendering problem is highly nonlinear and the inference and finding the relation between 2D image and corresponding 3D structure is intractable. If some assumptions can be incorporated to the problem, like deterministic models formulating behavior of all variables in the rendering equation to be defined accurately, so the solution can be computed in closed form [1]. However we cannot have such strong assumptions in all inverse rendering problems. In such cases we can use learning based methods by training some models and then use them for test samples. The most popular learning framework in recent years are deep networks which show their power in solving nonlinear problems. Deep networks like other learning based structures can be trained in 3 ways: Supervised, unsupervised and semi-supervised.

Supervised methods use a set of labeled data to guide the training. Their result will be accurate because of availability of ground truth in the training procedure but suffer from the lack of enough labeled training data in some domains. In the case of 3D Inverse rendering applications, providing enough realistic training data for a deep network may be impractical. One Solution for this problem is to use a generative model for generating synthetic training data to trained the network [26, 27].

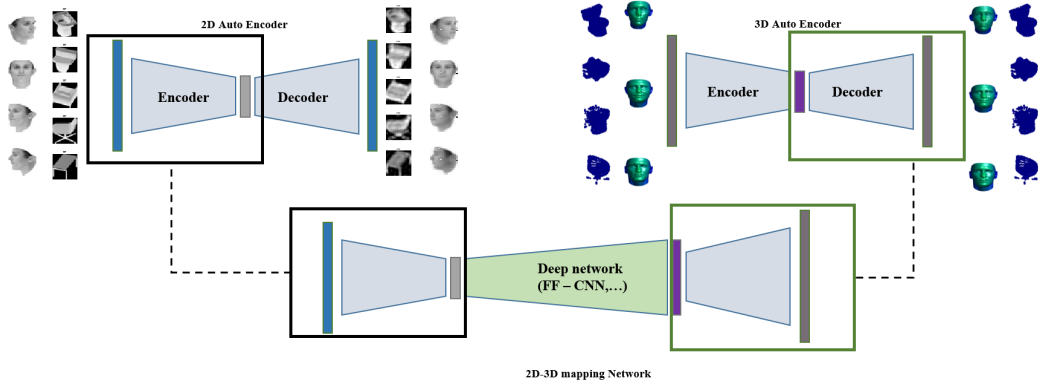


Figure 1: Main configuration of proposed framework

As another solution, There can be found some attempts in the recent literature to Unsupervised training the network for 3D reconstruction[25, 30] in which a 3D structure is generated in a part of the network and some rendering mechanism is used to transform the generated 3D structure to something in the form of input measurement to the network and then try to reduce the difference between input and rendered output. So the network does not need to labeled data for training. In the existing unsupervised methods in the literature, they used an **analytical form** for rendering the 3D structure with some assumptions about image formation process. This characteristic although is a significant improvement in reducing the need for training data but may **affects the quality of solutions** due to the existing assumptions about rendering mechanism (such as camera properties, scene illumination, reflectance properties and so on). In the case of 3D Inverse rendering problem, there are well known 3D models that can help us to generate enough data for training the deep networks.

In this paper, we propose a mechanism in which some parts of network can be **trained using unlabeled data** and then the whole network will be **fine-tuned using labeled training data** for aggregation of the components. We believe this properties reduces the neddd of deep network for large set of training data. Therefore it is a semi-supervised framework for inverse rendering problem.

The paper is organized as follows: we first briefly describe our framework in section 2. In section 3 we review and analyze some recent related works. The proposed framework and network structure will be define in more details in section 4. Some experimental results to verify the performance of our proposed structure on several benchmarks will be demonstrated in section 5 and finally some discussions and conclusions will be found in sections 6 and 7.

2 Overview

Our proposed framework is composed of 3 main components based on the fact that in inverse rendering problem we look for a way of **transforming the 2D input image to its corresponding 3D structure**. We do this by mapping the 2D input image and corresponding 3D structure to suitable representations and then **finding a transformation between obtained representations**. We will show experimentally that nonlinear mapping of suitable representations results more accurate solutions. finding suitable representations can be done using autoencoders as unsupervised tools for finding representations or by pre-trained convolutional networks for finding suitable representations. We did this either by different types of autoencoder or by some existing well known networks from literature.

In the case of using autoencoders for finding representations, inspired by 3D Morphable Models (3DMMs) [3] the network extracts the basis vectors from training data of each size and uses these basis functions to reconstruct more accurate 3D shape of an object rather that using a pre-defined 3DMM. Figure 1 shows the overall structure of the proposed framework. More details about the proposed idea can be found in section 4

3 Related works

In this section we review and analyze some of the most recent attempts for 3D inverse rendering from single 2D images using deep frameworks as related works to ours concentrating on their search strategies.

In [26], Richardson et al. proposed a **deep iterative framework** for 3D inverse rendering of human faces. In each iteration of this framework, the output of the network will be fed into the input as additional information to fine-tune the previous result. The training process in this method is supervised by using a 3DMM for generating synthetic labeled training data. The final output will be more fine-tuned using some form of Shape From Shading (SFS) method to achieve more details. The authors improve their work in [27] where, the SFS component of their network is implemented as another deep network. The output of this framework is the coefficients of a predefined 3DMM computed using a few hundred 3D samples and so this framework does not directly deal with 3D shape structures.

In [32], a supervised framework for 3D reconstruction of human faces from a single 2D image is proposed where they also used a 3DMM for generating training data. For this aim they fitted a 3DMM to 2D images via an optimization method and used that model as ground truth for the training data. This strategy of finding ground truth may cause the found solutions to be dependent on the power of the optimization method so that some powers of new work may be ignored. A **loss function** is also used for the training the network for human faces which can be tuned to make trade-off between generic face generation and overdetermined one, which is suitable for the purpose of face recognition.

In [25] an unsupervised framework for 3D reconstruction of objects from single or multiple input images or volumes is proposed. The authors suggested an inference network for encoding the input data to a low dimensional representation which will be fed into a generative network for generating 3D objects using the encoded data. The obtained 3D object will be rendered as a 2D image using a fixed differentiable rendering method. The objective of training of this network is to **minimize the difference between rendered and input images**. Using a fixed rendering component may restrict the accuracy of generated 3D shapes if the input image rendering mechanism differs from network rendering mechanism. The authors also suggested using a convolutional network as rendering component. Using another deep network as rendering component may overcome the mentioned limitation.

In [30] an autoencoder network is proposed for 3D reconstruction of human faces using an unsupervised training mechanism. In this network the encoder part of the autoencoder is a convolutional network like well-known alexnet [16] or VGG-face [20] networks and the decoder part is an analytical differentiable formula which uses the encoded data and some assumptions about the image formation process to form to a 3D face and then render it to a 2D image. In this paper the objective is also to minimize the difference between input image and rendered 3D shape using fix rendering method.

In [8] a Generative Adversarial Network (GAN) is trained for generating 3D shapes of different objects. In this method, in an unsupervised training approach, a random vector from a distribution will be fed to the network and a 3D shape of an object will be generated and then will be rendered in the form of a 2D image as input to a discriminator. In this way, the discriminator does not need to 3D shape to control the process of learning 3D shapes. The rendering method here is fixed and this network can be used to generate training data for inverse rendering frameworks of 3D objects.

In [12] a CNN is designed to recover the 3D geometry and texture of human faces from a single 2D image. This framework directly works with 3D coordinates in the form of proposed binary volumes and maps 2D images directly to the coordinates of binary volume. Using binary volume as 3D coordinates for representing human face may result in blur shapes.

In this paper, our aim is to design a deep framework for 3D shape inverse rendering of different types of 3D shape representations such as point clouds and volume representations from a single 2D input image. The training process utilizes from both labeled and unlabeled realistic data for improving the quality of resulted 3D shapes. The following section demonstrates the proposed approach in more details.

4 Proposed framework

The main idea in our work is to **end-to-end mapping of some representation of a single 2D input image to some representation of corresponding 3D shape structure.** Defining such structure is actually based on the main functionality of deep networks, i. e. using different layers of nonlinear computations in order to feature extraction and transformation [37]. Our proposed framework is also made of components for feature extraction and transformation.

As stated in section 2, we used three main components in our framework: the first component is trained to compute some representations for 2D input images. the second component is trained for reconstructing a 3D shape from some other representations and the third components is used to map the representation computed in the first component to one used in second component. The representations in each component should satisfy its objective. for instance, the component for computing representation from 2D image, should computes a representation that is suitable for being mapped to 3D representation. On the other hand, the 3D representation should be suitable for reconstructing the 3D shape in the form of point cloud or binary volume. We observed in our experiments that more complex representations for 2D representation component achieve better performance and decoder part of a linear autoencoder as 3D reconstruction component is suitable for reconstructing the 3D shape in he form of point cloud.

The third component is a structure for **mapping 2D low-dim representation to corresponding 3D representation.** We actually proposed this component because the manifold of 2D images may be different from the manifold of 3D shapes especially in the case of complicated shapes like human faces and therefore a nonlinear mapping function should be found to obtain an accurate 3D representation from 2D image representation.

Note that our proposed frameworks deals directly with dense 3D shapes of the objects instead of using predefined 3D models for representing a face. This characteristic has the advantage of not being restricted to a spanning obtained by a few samples by using larger set of training data for extracting suitable basis vectors for 3D and 2D shape spaces.

The main contributions of this paper are as follows:

- Using deep structures as different components in an interpretable framework for mapping the representations from a manifold to another and directly reconstructing 3D results. Using deep structures in each component improves handling the non-linearity of each stage.
- Possibility of **unsupervised pre-training of each component** using unlabeled data so that the representations can be improved using realistic data in-the-wild and therefore reducing the need for labeled data for training because of suitable initialization for training. In fact the labeled data is needed only for the final aggregation of pre-trained components.
- Using the training data itself to extract the bases of each representation space instead of using a predefined analytical models.
- Ability of training the framework for inverse rendering of different object classes from a single 2D input image.

In the following subsection, the proposed idea in this paper is demonstrated in the simplest case (linear) in terms of Singular Value Decomposition (SVD).

4.1 Describing proposed idea in a simple case (i.e. linear) in terms of Singular Value Decomposition (SVD)

In this section we analytically describe the proposed idea in this paper in the simplest case, i.e. the linear case. These calculations gives a mathematical insight into the motivation of proposing this framework. In section 5, we will evaluate our idea using different datasets and compare it with other possible method in the case of solving 3D shape inverse rendering problem. For this aim, assume the linear case for each component, i. e. linear autoencoder for 2D encoder and 3D decoder components and linear feedforward network as the mapping component.

In [2] it is proven that, the optimal representation space resulted by a linear autoencoder, using L2-norm as loss function, has the same spanning as the sub-space resulted by performing PCA on the data. Therefore we can consider the use of a linear autoencoder equivalent to performing PCA on training data. It is therefore can be inferred that the weights of hidden to output layer of the

trained linear autoencoder are in fact the bases of the resulted representation subspace which are found using the training data of autoencoder. This characteristic means that, by using a linear autoencoder, we can obtain the bases of a representation space for data of any size without the concern of basis vector decomposition of a large data matrix.

Let X denotes the matrix which is formed by concatenating n_1 samples of 2D input images vectorized as D -dimensional column vectors as 2:

$$X = [x_1, \dots, x_{n_1}]_{D \times n_1} \quad (2)$$

By applying SVD on X , after subtracting its mean value, the bases (Eigen vectors) of the PCA subspace can be obtained using SVD as 3:

$$[U_x, \sigma_x, V_x] = SVD(X) \quad (3)$$

where, U_x is the matrix with the columns representing the eigenvectors of XX^T , σ_x is a diagonal matrix with eigenvalues of XX^T as diagonal elements and V_x denotes the matrix with the eigenvectors of $X^T X$ on its columns. Therefore we can use the columns of U_x as the basis vectors of the resulted representation subspace for 2D images, and the k -dimensional representation of X in the resulting sub-space, spanned by the first k eigenvectors in U_x with largest eigen values in σ_x , can be expressed as Y in 4:

$$Y_{k \times n_1} = U_x^T X_{D \times n_1} \quad (4)$$

Similarly, considering Z as the column matrix containing n_2 p -dimensional vectorized training samples of 3D shape structures, represented as 3D point clouds or binary volumes, the resulted k' -dimensional representation of Z using another autoencoder can be expressed as $B_{k' \times n_2}$ in 7:

$$Z = [z_1, \dots, z_{n_2}]_{p \times n_2} \quad (5)$$

$$[U_z, \sigma_z, V_z] = SVD(Z) \quad (6)$$

$$B_{k' \times n_2} = U_z^T Z_{p \times n_2} \quad (7)$$

To find a transformation between two obtained representations Y and B for the inverse rendering problem, if a linear feedforward net is used, it is equivalent to performing a least square mechanism to find a linear relation between Y and B as 8:

$$T_{k \times k'} = \operatorname{argmin}_T |B - TY| \quad (8)$$

and the result can be used for reconstructing the 3D shape, say \hat{z} , as 9:

$$\hat{z} = U_z TY \quad (9)$$

On the other hand, we can also solve a direct optimization problem that linearly tries to find the linear relation between 2D images and corresponding 3D shapes as 10-11.

$$\hat{B} = \operatorname{argmin}_B |Z - BX| \quad (10)$$

$$\hat{z} = \hat{B} X \quad (11)$$

These two approaches look for a subspace that best describes the 3D shape of the objects in 2D image. The first approach first tries to find a low dimensional representation of 2D input image and corresponding 3D structure for a set of training data and then a linear solution is found for transforming the 2D representation to 3D one via least squares. Note that the matrices \hat{B} and $U_z T$ are used as mapping functions for the test data. And the examples used for calculating U_x do not necessarily correspond to those which were used to obtain U_z and Y . Such mechanism shows the possibility of unsupervised training of different components in this method.

The second closed form approach in 11 directly finds a linear mapping between 2D input image and corresponding 3D structure. We believe that the first approach significantly improves the process of finding a solution for inverse rendering problem and in Table 2 in the section 5.4, we verified this claim experimentally compared with second approach in 10.

4.2 Loss function

As we mentioned before, the purpose of our proposed framework in this paper is to recover the shape of an object. So we deal with a regression problem. Therefore, we used Root Mean Squared Error (RMSE) as in 12 as the loss criterion for training the components in the proposed framework. This criterion is the standard deviation of prediction error and is a standard way to regression analysis [6, 14].

$$E_{RMSE} = \sqrt{\frac{\sum_{i=1}^N |\hat{x}_o^i - x_{gt}^i|^2}{N}} \quad (12)$$

Where, \hat{x}_o^i and x_{gt}^i denote the prediction and ground truth of the i th training sample and N stands for the sample size. Here, the sample size stands for the size of the mini batch. There are different loss functions are proposed to be used for inverse rendering problems in different researches [12, 32] for different applications like reconstruction or recognition. In [12] a binary voxel representation is used for representing the faces and sigmoid cross entropy loss function is used for training the network. In our framework we deal with both mesh representation and voxel representations. So we choose the standard RMSE for training the networks.

5 Experiments

In this section we perform some experiments to evaluate the performance of our proposed framework in different 3D inverse rendering scenarios. We used 2 types of datasets, objects and human faces having different 3D shape representations, like point clouds and binary volumes, in order to report and compare the reconstruction results. we also analyse some structures as different components of the framework.

5.1 Datasets

We report the results of the performance of proposed framework on two types of datasets: objects and human faces. In each case we trained a separate structure using that type of data representation. we also used some unlabeled 2D and 3D datasets in order to unsupervised pre-training the components.

In the case of human faces, we used LFW dataset [11] as 2D unlabeled data for unsupervised pre-training and Besel Face Model [22] for generating synthetic data as labeled training data along with the Bosphorus dataset [28] as realistic labeled data for training and fine-tuning respectively. Note that for aggregating all components using supervised training, we used faces generated by BESEL Face Model (BFM) having natural expression rendered with 3 poses, i.e. -45, 0, 45 degrees about z axis. Therefore the objective of human face shape inverse rendering in our work is to pose invariant identity recovering of faces from a single 2D image. The 3D faces in our framework are represented as 3D point clouds with 54903 vertices in 3D space which are normalized between 0,1 in each dimension. We generated 4000 training and 1000 test samples in natural expression and rendered 2D images of size 32*32 with mentioned poses as 2D input images.

In the case of objects, we used some categories of imagenet [5] dataset as unlabeled data for pre-training and ModelNet10 [36] and ShapeNet [4] datasets as labeled training data which are to the best of our knowledge, well-known datasets for 3D object reconstructions. we used 10 categories for training the networks in the form of $30 \times 30 \times 30$ binary voxel grid and rendered each training sample as a 32×32 gray scale image from 8 viewpoints (between 0 and 180 about z axis). In the case of linear autoencoders we vectorized the voxel grid as a binary column vector of size 27000.

We incorporate well-known alexnet in our experiments for evaluating the use of a deep CNN for computing 2D representations for the face dataset. In this case we rendered and used 2D colored images of the faces in each existing dataset with the size $227 \times 227 \times 3$ which is the standard input size to alexnet framework.

5.2 Parameter setting

Table 1 includes the configuration of each structure used for each type of dataset for the type of used structure. Since we applied inverse rendering on different data types, we had to use different structures for the network components in each case.

Object (binary Volume)			Human Face (Point cloud)			
	Linear	Nonlinear	Convolutional	Linear	Nonlinear	Convolutional
Encoder	1024-60	1024 - 100- 60	$3 \times 3 \times 1 -$ $3 \times 3 \times \times 32 -$ $3 \times 3 \times 16$	1024 - 60	1024 - 100 - 60	$3 \times 3 \times 1 -$ $3 \times 3 \times 32$ $3 \times 3 \times 16$
Mapping	60-100-400	60-100-400	$1 \times 3 \times 1 \times 100 -$ 400	60-100-98	60-100-98	$1 \times 3 \times 100 -$ 98
Decoder	400-27000	400-800-27000	$3 \times 3 \times 3 \times 32$ $3 \times 3 \times 3 \times 16$ $30 \times 30 \times 30$	98-160470	98-200-160470	-

Table 1: Network configuration for each components using different structures

	LS using low dim representation	Directly finding Ls between 2D and 3D
Besel face Dataset	0.8727	2.2869
ModelNet Dataset	0.0970	0.1469
ShapeNet Dataset	0.0712	0.2455

Table 2: Comparison of average test RMSE obtained by different datasets using closed form linear solutions in section 4.1

Note that in the case of object datasets, since there are 10 classes of objects, there were need to layers with larger sizes for representation units.

The learning rate was set for each configuration so that it gives better performance. In most cases we uses 0.001 for 1000 epochs and then reduce it to 0.00001 for another 1000 epochs. In the case of mesh representation the learning was slower and needed smaller learning rate for longer training time. Batch size was set to a constant, in most cases equal to 40 for training. We trained our networks from scratch on waterloo university servers: gpu-pr1-01 and gpu-pt1-01, saspc73, and sharcnet-graham machines which are well-known servers having several gpus suitable for training deep networks.

5.3 Evaluation mechanism

We used RMSE as the evaluation criterion on test and validation sets in different parts of our experiments. In the case of face datasets we also showed the RMSE in the form of Heat maps, showing point-wise RMSE of reconstructed face on the ground truth face, as a suitable tool for error visualization in different regions of the reconstructed face.

We start the experiments by first reporting the evaluation results of closed form linear solutions described in section 4.1 using different data types. in the next step we will use equivalent deep structure by changing the mapping component to nonlinear structure and show that the results can be significantly improved. As a nonlinear component for encoder part of our framework, we fine-tuned alexnet using our labeled human face dataset and report the results. We then show some reconstruction results on realistic human face dataset and compare our results with some of the most recent methods for 3D reconstruction.

5.4 Results of solving linear closed form formula for inverse rendering

Regarding the analytical description of proposed idea in section 4.1, here we show that finding a linear mapping between low dimensional representations of 2D and 3D data achieves significantly better results rather than finding a direct linear mapping between them (2D and 3D data). Table 2 and figures ?? include the numerical and visual results of applying these methods on different datasets, where “LS using low dim representation” stands for finding the mapping between low dimensional representations of 2D and 3D data using least squares and “Directly finding ls between 2D and 3D” stands for finding a least squares solution for direct mapping 2D input images to 3D data.

The numerical results in Table 2 indicate the effectiveness of the proposed idea of mapping low dimensional representations instead of direct mapping in terms of RMSE for all existing datasets.

Looking at Figure 2, first, it is clear that using linear methods is not powerful enough for addressing 3D shape inverse rendering problem from a single 2d input image and there is need

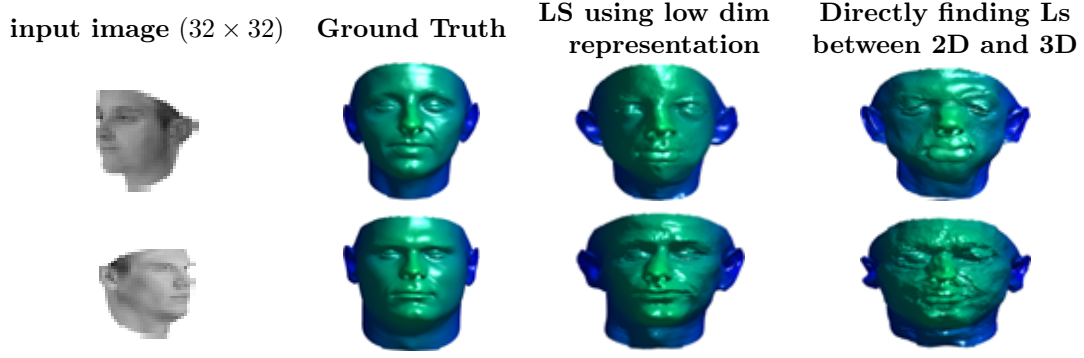


Figure 2: Visual results of using linear methods for finding a mapping between 2D to 3D spaces for BESE face dataset

to use nonlinear tools in this fields especially in the case of human face inverse rendering. And second, we can see that using the low dimensional representation mapping works better than direct mapping visually.

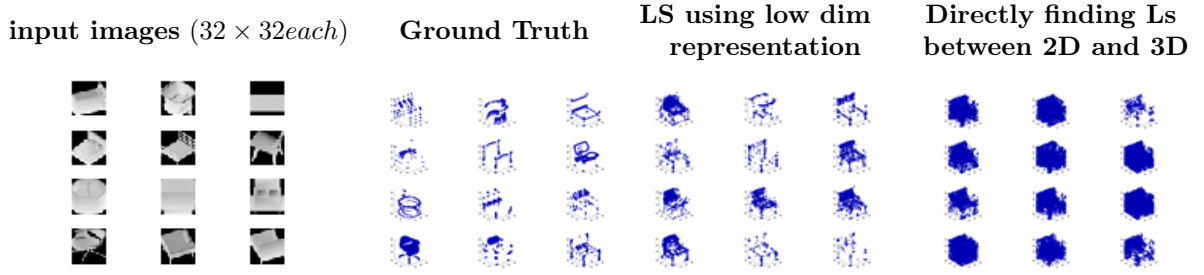


Figure 3: Visual results of comparing linear least squares for finding a mapping between 2D and 3D space for Modelnet10 dataset

In the next step of our experiments, we set up a deep structure consist of linear autoencoders for finding representations and a nonlinear feedforward network for mapping the representations as a strat point for using nonlinearity in the proposed framework.

5.5 Using linear encoders and decoders with nonlinear feedforward network as mapping component:

As a start point of using nonlinear deep structure in our framework, we only used a nonlinear feedforward network as the mapping component and trained the network using 2000 samples of BESEL face model dataset and training set of Modelnet10 dataset for 1000 epochs. Figures 4,5 show the results on test data for face and object datasets, respectively.

It can be observed that using nonlinearity only in the mapping component in our framework results in significant improvement Visually and numerically. For instance, In the case of face datasets, we have face-like reconstructions. this observation also shows that the relation between subspaces of 2D and 3D representations have nonlinear relation.

In the next subsection, after fixing the nonlinear feedforward network as mapping component, we consider the use of more complicated representations for 2D and 3D data. To do this, we fine-tuned alexnet as encoder of 2D input image.

5.6 Transfer learning from Alexnet as 2D encoder

Figure 6 shows the proposed framework which uses Alexnet pre-trained model as encoder component. This model, trained on more than one million images of about 1000 categories of images from ImageNet database [16] like human, chair, pencil and many other categories, is a well-known model for extracting rich features from images for classification of them into 1000 object categories. By

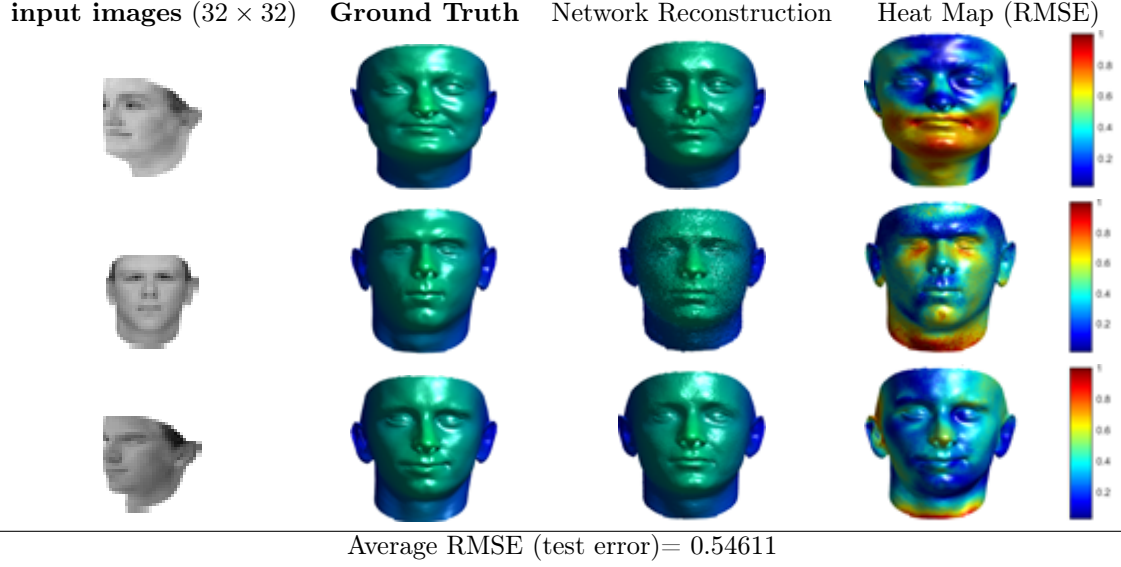


Figure 4: Results of using nonlinear feedforward network as mapping component on BESEL face dataset

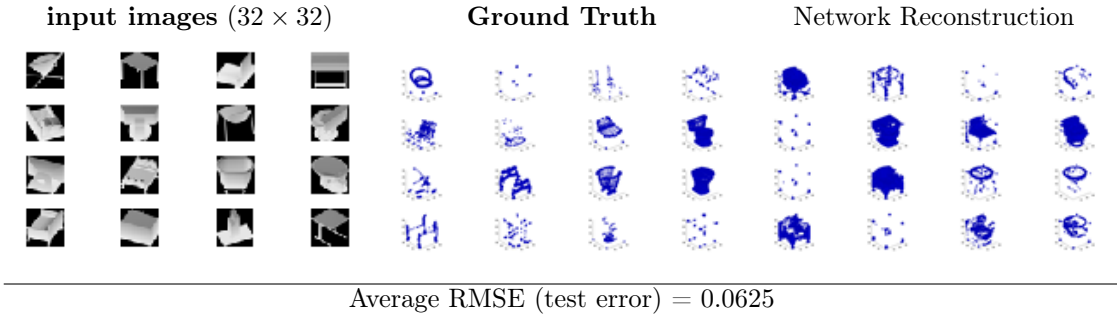


Figure 5: Results of using nonlinear feedforward network as mapping component on Modelnet dataset

using this model as encoder component, we first verify the possibility of using pre-trained components in our framework and second analyze the use of more complicated representations for solving inverse rendering problem in our proposed framework. We fine-tuned alexnet using a set of colored

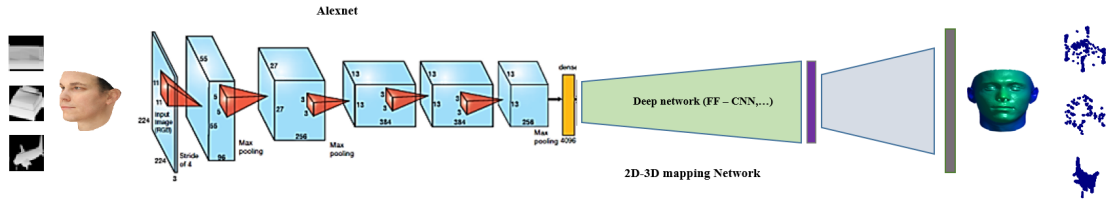


Figure 6: alexnet as encoder component in the proposed framework

2D input images of about 2000 samples generated by BESEL face dataset, and 10000 samples from Modelnet10 and the results are shown in Figures 7, 8 respectively. in the case of Modelnet10, Since there were no texture associated with the object to be rendered as a color image, we just represent the input images with the size $227 \times 227 \times 3$ to feed to the network.

From Figures 7, 8 it is observable that using this configuration and with computing more complex representations for 2D input images, the accuracy of reconstruction will be improved compared with the linear autoencoder. Actually, the objective of encoder component is to compute representations which are more suitable for being mapped to 3D representation and the results show that more complicated representations are appropriate for this aim.

In the case of decoder component, the aim is recovering sharp 3D shapes from 3D represen-

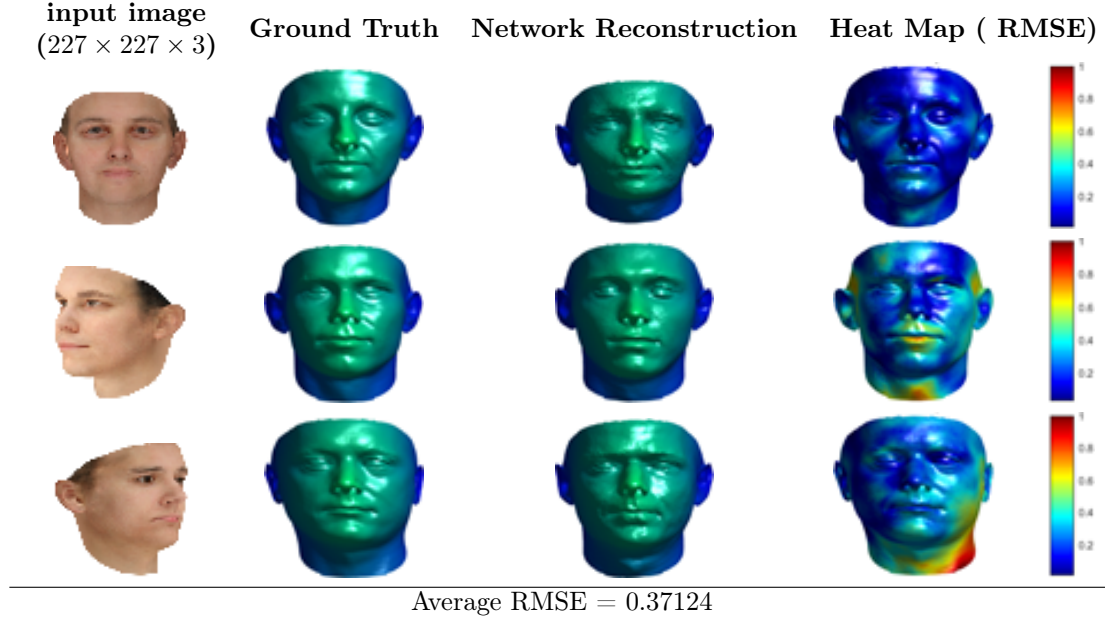


Figure 7: Visual and numerical results of using alexnet as encoder in proposed framework (BESEL face Dataset)

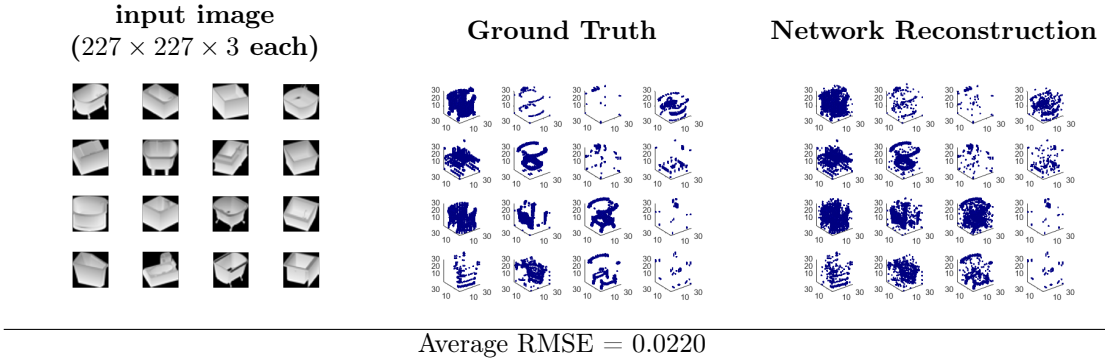


Figure 8: Visual and numerical results of using alexnet as encoder in proposed framework (Modelnet10 Dataset)

tations. therefore, the representation in this component should be so that the 3D shape can be inferred from. in the case of point cloud 3d shapes, where we directly deal with real 3D coordinate values the linear decoder gives the most promising result compared with nonlinear and convolutional structures with blur results. in the case of binary volume 3D shapes, where the output can be digitized to 0 and 1 using some threshold value, nonlinear structures show promising performances. figure 9 shows the digaram of RMSE vs epoch using different mentioned structures for different types of 3D shapes.

5.7 Pose invariant reconstruction

Figure 10 shows the power of proposed framework in reconstruction of a sample face input image captured from 3 different poses.

5.8 Reconstruction results on realistic datasets

We evaluated the final configuration of our framework (Alexnet encoder – nonlinear mapping – linear decoder) on Bosphorus Face dataset as a realistic 3D face dataset. Figure 8 shows the results and their heat map in terms of RMSE on Bosphorus dataset. We also compared the visual reconstruction results of proposed framework with [12] and [30] on some realistic images.

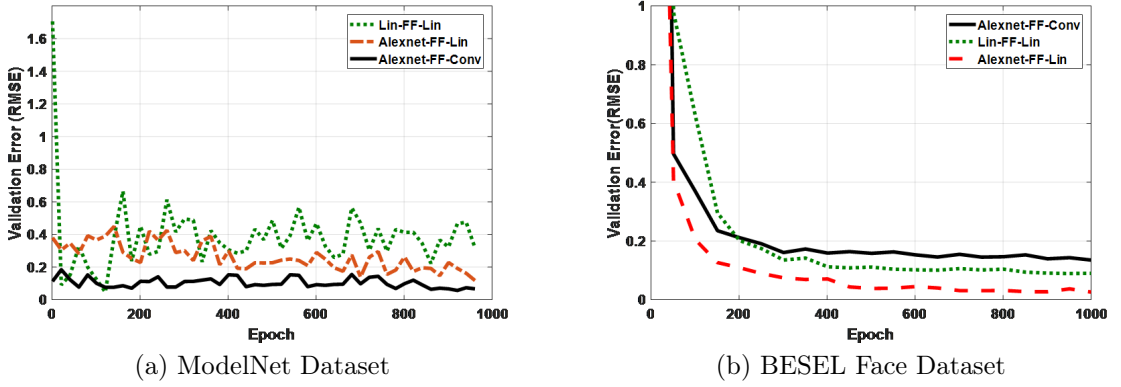


Figure 9: Validation error vs Epoch for Different structures used as components in proposed frameworks for (a) ModelNet dataset and (b) Face dataset where Lin: Linear encoder or decoder, FF: Nonlinear FeedForward network, Conv: Convolutional encoder or decoder

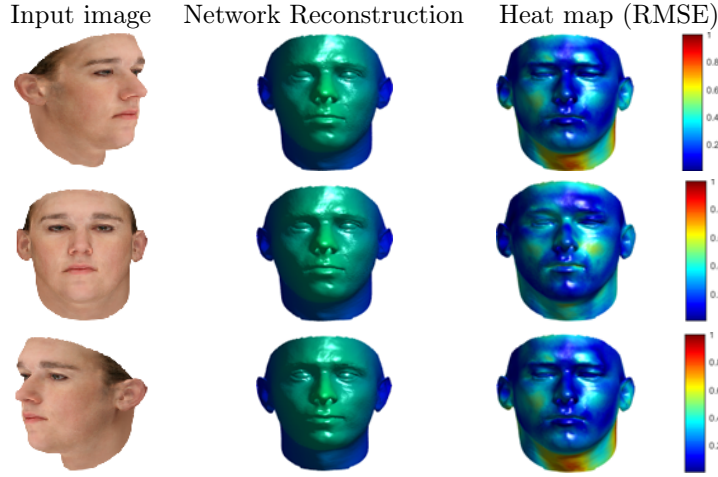


Figure 10: Result of proposed framework on Bosphorus face dataset

In the case of realistic images we first manually cropped the face from input image and then aligned the intensity of the image with a sample image from training samples of our framework using Procrustes analysis used in [29] and then use them as input image to our framework. Note that in order to compute RMSE for heat map we computed the minimum RMSE from Ground truth to network reconstruction.

The results in Figure 11 show that the network could successfully recover the main features in each face. Note that using the linear decoder in the case of face datasets is equivalent to performing PCA on all of training samples (about 2000) and this causes achieving better results compared with a predefined 3DMM using small set of training data. Figure 9 shows some visual reconstruction results, using some realistic images, obtained by proposed framework compared with two similar recent reconstruction methods [30] and [12] which were described in section 3. We can see that the results obtained by proposed framework are comparable with unsupervised training methods and can detect more shape details compared with the binary volume representation for 3D shapes. We believe this is because of dealing directly with point clouds and extracting basis vectors from a large set of training data.

6 Discussion

As we observed in the experiments, using a convolutional network as 2D encoder and mapping component in our framework significantly improves its performance. However in the case of 3D decoder using a non-convolutional network (linear or non-linear) results in better reconstruction results. We believe this is because in convolutional Autoencoder, reconstruction of the representation

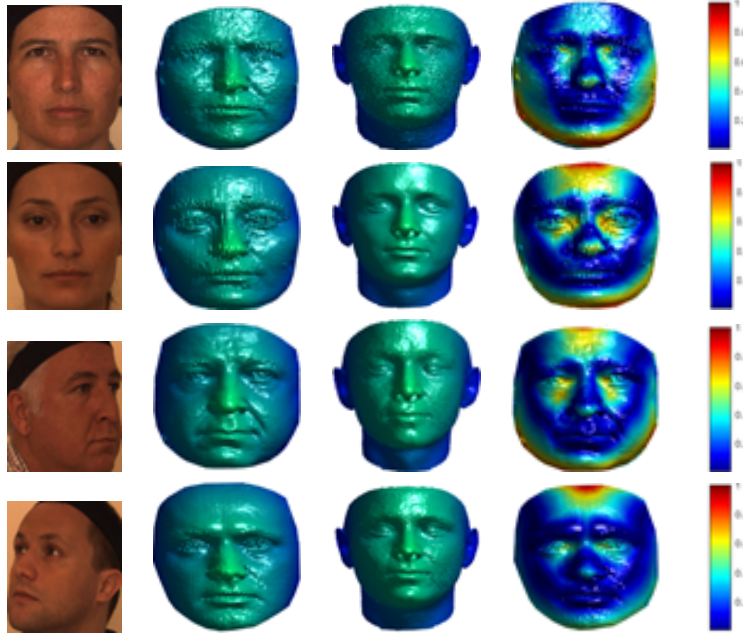


Figure 11: Result of proposed framework on Bosphorus face dataset

obtained from data may be inaccurate because of the information loss resulted by convolution and pooling units.

Another Property of our proposed framework is the use of autoencoder to directly reconstruct the 3D shape structure by learning from large set of examples. this structure Uses the training data itself to extract bases of the shape space. We believe and this characteristic leads to have unbiased reconstructions rather than using a predefined 3D model which is biased toward mean shape of the training samples. On the other hand, working directly with point cloud representation for representing the 3D shape helps the framework to achieve more detailed results.

As a disadvantage of regressing process in our framework, we can mention the existing loss functions. in the case of situation which the output of the network should be some exact values for regression, there is need to define a suitable loss function to guide the learning process toward finding desired solution. If we can turn the problem of regression into the form of some classification problem using losses like cross entropy and so on, we will achieve more desirable solutions like the results obtained by Generator Adversarial Netwok (GAN) frameworks.

7 Conclusions and future works

In this paper, a semi-supervised interpretable framework is proposed for pose invariant 3D shape inverse rendering from a single 2D input image. Using autoencoder based on mapping suitable low dimensional representations computed from 2D and 3D data components for extracting representations and reconstruction of final 3D shapes in the proposed framework, on one hand, makes it possible to use unlabeled data for unsupervised pre-training and therefore reducing the need to large sets of labeled data for training the framework and on the other hand, helps to achieve more promising and detailed reconstructions because of extraction of bases vector of 3D shape space from the large set of training data compared with predefined 3D models.

in this proposed framework two types of 3D shape representations were used for reconstruction: point cloud and binary volumes. The obtained results show that binary volumes are more suitable representations for more complicated deep networks. However they express blur shapes with high dimensional representation and can express less details compared with point cloud representations. on the other hand, working with exact 3D coordinates of point clouds is not too easy for deep networks and linear structures are preferred for such shape reconstructions.

In the future attempts one can unify the Inverse rendering framework for working with both point cloud and volume shape structures for different class of objects.

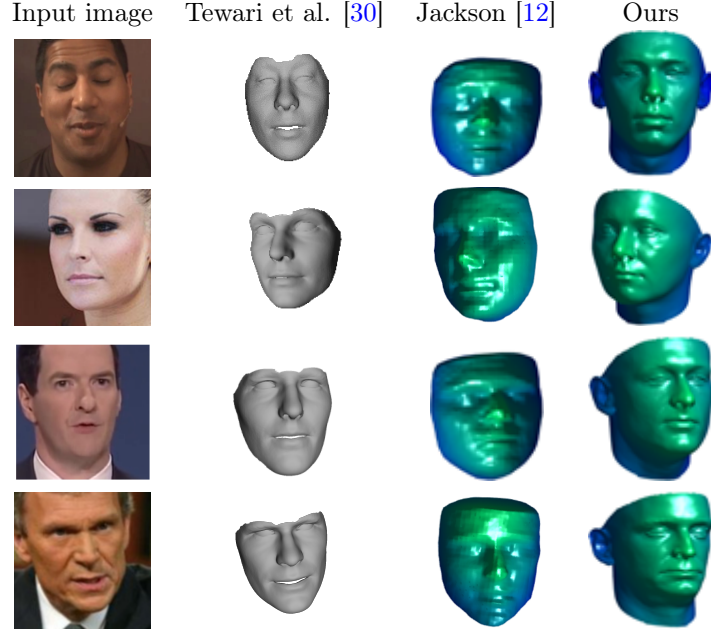


Figure 12: Comparing visual reconstruction results with [12, 30]

References

- [1] ALDRIAN, O., AND SMITH, W. A. Inverse rendering of faces with a 3d morphable model. *IEEE transactions on pattern analysis and machine intelligence* 35, 5 (2013), 1080–1093.
- [2] BALDI, P., AND HORNIK, K. Neural networks and principal component analysis: Learning from examples without local minima. *Neural networks* 2, 1 (1989), 53–58.
- [3] BLANZ, V., AND VETTER, T. A morphable model for the synthesis of 3d faces. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques* (1999), ACM Press/Addison-Wesley Publishing Co., pp. 187–194.
- [4] CHANG, A. X., FUNKHOUSER, T., GUIBAS, L., HANRAHAN, P., HUANG, Q., LI, Z., SAVARESE, S., SAVVA, M., SONG, S., SU, H., ET AL. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012* (2015).
- [5] DENG, J., DONG, W., SOCHER, R., LI, L.-J., LI, K., AND FEI-FEI, L. Imagenet: A large-scale hierarchical image database. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on* (2009), IEEE, pp. 248–255.
- [6] DOU, P., SHAH, S. K., AND KAKADIARIS, I. A. End-to-end 3d face reconstruction with deep neural networks. *arXiv preprint arXiv:1704.05020* (2017).
- [7] FOUHEY, D. F., GUPTA, A., AND ZISSERMAN, A. Understanding higher-order shape via 3d shape attributes. *arXiv preprint arXiv:1612.06836* (2016).
- [8] GADELHA, M., MAJI, S., AND WANG, R. 3d shape induction from 2d views of multiple objects. *arXiv preprint arXiv:1612.05872* (2016).
- [9] GAO, Y., AND YUILLE, A. L. Exploiting symmetry and/or manhattan properties for 3d object structure estimation from single and multiple images. *arXiv preprint arXiv:1607.07129* (2016).
- [10] GARRIDO, P., ZOLLHÖFER, M., CASAS, D., VALGAERTS, L., VARANASI, K., PÉREZ, P., AND THEOBALT, C. Reconstruction of personalized 3d face rigs from monocular video. *ACM Transactions on Graphics (TOG)* 35, 3 (2016), 28.
- [11] HUANG, G. B., RAMESH, M., BERG, T., AND LEARNED-MILLER, E. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. Tech. Rep. 07-49, University of Massachusetts, Amherst, October 2007.

- [12] JACKSON, A. S., BULAT, A., ARGYRIOU, V., AND TZIMIROPOULOS, G. Large pose 3d face reconstruction from a single image via direct volumetric cnn regression. *arXiv preprint arXiv:1703.07834* (2017).
- [13] JIANG, L., ZHANG, J., DENG, B., LI, H., AND LIU, L. 3d face reconstruction with geometry details from a single image. *arXiv preprint arXiv:1702.05619* (2017).
- [14] KIM, H., ZOLLHÖFER, M., TEWARI, A., THIES, J., RICHARDT, C., AND THEOBALT, C. Inversefacenet: Deep single-shot inverse face rendering from a single image. *arXiv preprint arXiv:1703.10956* (2017).
- [15] KIM, K., TORII, A., AND OKUTOMI, M. Multi-view inverse rendering under arbitrary illumination and albedo. In *European Conference on Computer Vision* (2016), Springer, pp. 750–767.
- [16] KRIZHEVSKY, A., SUTSKEVER, I., AND HINTON, G. E. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems* (2012), pp. 1097–1105.
- [17] LIU, J., YU, F., AND FUNKHOUSER, T. Interactive 3d modeling with a generative adversarial network. *arXiv preprint arXiv:1706.05170* (2017).
- [18] LUN, Z., GADELHA, M., KALOGERAKIS, E., MAJI, S., AND WANG, R. 3d shape reconstruction from sketches via multi-view convolutional networks. *arXiv preprint arXiv:1707.06375* (2017).
- [19] MARSCHNER, S. R., AND GREENBERG, D. P. *Inverse rendering for computer graphics*. Cornell University, 1998.
- [20] PARKHI, O. M., VEDALDI, A., ZISSERMAN, A., ET AL. Deep face recognition. In *BMVC* (2015), vol. 1, p. 6.
- [21] PATOW, G., AND PUEYO, X. A survey of inverse rendering problems. In *Computer graphics forum* (2003), vol. 22, Wiley Online Library, pp. 663–687.
- [22] PAYSAN, P., KNOTHE, R., AMBERG, B., ROMDHANI, S., AND VETTER, T. A 3d face model for pose and illumination invariant face recognition. In *Advanced Video and Signal Based Surveillance, 2009. AVSS'09. Sixth IEEE International Conference on* (2009), Ieee, pp. 296–301.
- [23] PIOTRASCHKE, M., AND BLANZ, V. Automated 3d face reconstruction from multiple images using quality measures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 3418–3427.
- [24] POGGIO, T., AND KOCH, C. Ill-posed problems in early vision: from computational theory to analogue networks. *Proceedings of the Royal Society of London B: Biological Sciences* 226, 1244 (1985), 303–323.
- [25] REZENDE, D. J., ESLAMI, S. A., MOHAMED, S., BATTAGLIA, P., JADERBERG, M., AND HEES, N. Unsupervised learning of 3d structure from images. In *Advances in Neural Information Processing Systems* (2016), pp. 4996–5004.
- [26] RICHARDSON, E., SELA, M., AND KIMMEL, R. 3d face reconstruction by learning from synthetic data. In *3D Vision (3DV), 2016 Fourth International Conference on* (2016), IEEE, pp. 460–469.
- [27] RICHARDSON, E., SELA, M., OR-EL, R., AND KIMMEL, R. Learning detailed face reconstruction from a single image. *arXiv preprint arXiv:1611.05053* (2016).
- [28] SAVRAN, A., ALYÜZ, N., DIBEKLIOĞLU, H., ÇELIKTUTAN, O., GÖKBERK, B., SANKUR, B., AND AKARUN, L. Bosphorus database for 3d face analysis. *Biometrics and identity management* (2008), 47–56.
- [29] SKOGLUND, K. Three-dimensional face modelling and analysis. Master’s thesis, Technical University of Denmark, DTU, DK-2800 Kgs. Lyngby, Denmark, 2003.

- [30] TEWARI, A., ZOLLHÖFER, M., KIM, H., GARRIDO, P., BERNARD, F., PÉREZ, P., AND THEOBALT, C. Mofa: Model-based deep convolutional face autoencoder for unsupervised monocular reconstruction. *arXiv preprint arXiv:1703.10580* (2017).
- [31] THIES, J., ZOLLHOFER, M., STAMMINGER, M., THEOBALT, C., AND NIESSNER, M. Face2face: Real-time face capture and reenactment of rgb videos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 2387–2395.
- [32] TRAN, A. T., HASSNER, T., MASI, I., AND MEDIONI, G. Regressing robust and discriminative 3d morphable models with a very deep neural network. *arXiv preprint arXiv:1612.04904* (2016).
- [33] WANG, J., CHE, C., GALEOTTI, J., HORVATH, S., GORANTLA, V., AND STETTEN, G. Ultrasound tracking using probesight: Camera pose estimation relative to external anatomy by inverse rendering of a prior high-resolution 3d surface map. In *Applications of Computer Vision (WACV), 2017 IEEE Winter Conference on* (2017), IEEE, pp. 825–833.
- [34] WOLPERT, D. H., AND MACREADY, W. G. No free lunch theorems for optimization. *IEEE transactions on evolutionary computation* 1, 1 (1997), 67–82.
- [35] WOOD, E., BALTRUŠAITIS, T., MORENCY, L.-P., ROBINSON, P., AND BULLING, A. A 3d morphable eye region model for gaze estimation. In *European Conference on Computer Vision* (2016), Springer, pp. 297–313.
- [36] WU, Z., SONG, S., KHOSLA, A., YU, F., ZHANG, L., TANG, X., AND XIAO, J. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2015), pp. 1912–1920.
- [37] YU, D., AND DENG, L. *Automatic speech recognition: A deep learning approach*. Springer, 2014.
- [38] ZHU, X., LEI, Z., LIU, X., SHI, H., AND LI, S. Z. Face alignment across large poses: A 3d solution. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (2016), pp. 146–155.