3D FACIAL MODEL SYNTHESIS USING COUPLED DICTIONARIES

Swami Sankaranarayanan, Vishal M. Patel, Rama Chellappa

Department of Electrical and Computer Engineering and
Center for Automation Research, UMIACS, University of Maryland, College Park, MD 20742
{swamiviv, pvishalm, rama}@umiacs.umd.edu

ABSTRACT

In this work, we propose a generative way of modeling faces, where the 3D shape of a face is generated by a supervised learning procedure involving coupled sparse feature learning. To learn dictionaries using the proposed method, we use the USF-HUMAN ID database [1]. We provide as input to our training system, paired correspondences of 2D and 3D images of individuals and aim to learn the low-level patches both in 2D and 3D domains that describe the corresponding subspaces in a sparse manner. We demonstrate the efficacy of our method by quantitative results on the 3D database and qualitative results on images drawn from the internet.

Index Terms— 3D Model, Face Synthesis, Coupled Sparse Coding, Cross-modal Learning

1. INTRODUCTION

The problem of generating a 3D surface from a 2D image has been a greatly studied problem in Computer Vision. Since the seminal work of Horn *et al.* describing a variational solution to the Shape from Shading problem [2], there have been several approaches that have provided efficient solutions to learn the 3D geometry given an image of the surface. The common attributes of these approaches have been to exploit prior knowledge on reflectance and lighting conditions, thereby constraining the solution space. With the abundance of 3D sensors like kinect available today, there is a variety of approaches that try to "learn" these physical constraints from the training data. A popular example of such a method is the *make3D* [3] system that computes the depth of each pixel of a two-dimensional scene. This work is one such attempt of 3D model estimation of a particular class of objects, human faces. The overall objective of this work is to obtain a 3D surface that's underlying a given 2D face image. By 3D surface, we mean the depth map (range data) corresponding to the 2D image. One such 2D-3D pair is shown in Figure 1.

The problem of inferring the 3D model from a 2D image can be cast as learning a mapping between two domains: intensity and depth. The mapping is in most cases, highly non-linear. However, in cases where the domains under consideration have an intrinsic connection, such as when they represent the same classes of objects, this mapping could be approximately learned using shared sparse representations as shown in [4],[5] for a variety of problems such as compressive sensing, image super-resolution etc. In this work, we extend the idea of coupled feature learning using sparse dictionaries to the problem of obtaining a 3D model of a given 2D image. Due to the difficulty of the problem for general object classes, we restrict ourselves to human face images. Figure 1 presents an overview of the proposed method. Given an input image, the face is detected and warped to a 2D reference image chosen from a standard database. The trained dictionaries along with the reference 2D-3D pair are used to infer the 3D shape for the warped face. In order to visualize the 3D output better, the synthesized images from the 3D model estimate produced by our method are shown in the bottom row.

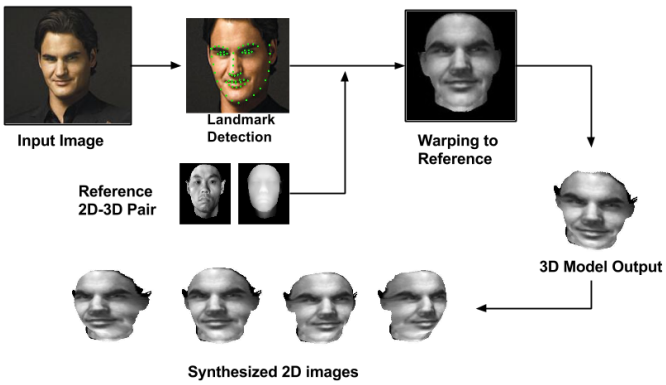

Fig. 1: Overview of the proposed method

2. RELATED WORK

The formulations used in this work can be related to two different classes of works: learning coupled feature representations; obtaining 3D model estimate from still images/video. Learning coupled feature space representations using dictionary-based methods was used in the work of single image super-resolution by Yang *et al.* [4]. In their work, the main objective was to generate a higher resolution image given the same image in a lower resolution. The idea of bilevel sparse coding was later proposed by the same authors for a more general category of problems involving coupled feature space representations. The major difference is that in bilevel sparse coding, the sparsity structure between the two domains under consideration can be tweaked using a parameter which trades off the reconstruction errors in each domain. Since then, there have been several approaches that extended these ideas to learn mappings between several domains of interest: Photo to Sketch [6], RGB-HOG [7] etc. This work uses the ideas of coupled feature learning to approximate the mapping between intensity and depth domains.

In the context of facial models, with sophisticated imaging sensors, the seminal work of Blanz and Vetter [8] provided a boost to the area of 3D modeling. Several approaches for solving this ill-constrained problem have been proposed over the years [9], [3], [10], [11], [12]. The solution to 3D model estimation has been dealt

This research is based upon work supported by the Office of the Director of National Intelligence (ODNI), Intelligence Advanced Research Projects Activity (IARPA),via IARPA R&D Contract No. 2014-14071600012. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either expressed or implied, of the ODNI, IARPA, or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright annotation thereon.

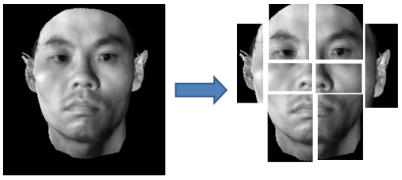

Fig. 2: Region masks overlaid over the reference image.

with in two ways: learning a mapping between the image and depth domain by using hand-crafted features as in [3] or molding a prior 3D shape to fit the appearance of the test image. In this work, we combine the two approaches by learning the mapping using features learnt from the data and using a prior 3D shape as a global constraint. We provide both visual results and a quantitative comparison of our method with the Depth transfer method [12].

3. PREPROCESSING - WARPING AND REGION LABELING

For training, we use a database of 2D-3D image pairs, an example of which is shown in Fig.1. In this work, the USF-HUMAN ID database [1] is used for experiments. In the training stage, since we are given the 3D models that belong to the input 2D images, a straight forward rendering scheme is used to render a frontal view. During inference, we are given a random 2D input whose 3D model is unknown and hence a more sophisticated warping procedure is followed. Given an input image, we use the publicly available implementation of Discriminative response map fitting (DRMF) method [13] to extract facial landmarks. The input image is warped to a reference image chosen from [1], whose 3D model is available. The warping procedure takes as input the facial landmarks from the input image, reference image and the 2D-3D correspondences to estimate the intrinsic and extrinsic camera parameters and hence the projection matrices. Using these projection matrices, a frontal warp of the input image is generated. The whole procedure is performed using our implementation of the method described in the DeepFace work [14].

We split the frontal face into eight overlapping regions as shown in Figure 2 and learn the dictionaries $\{\mathbf{D_x}, \mathbf{D_z}\}$ for each region separately. Since any input image is warped to a given reference image the regions are marked off by manually annotated masks on the reference image. Thus, when the patches are generated, each patch carries a region-id signifying the region it belongs to. In the inference stage, a patch extracted from region r is processed using the dictionaries learnt corresponding to region r. This step is a simple way to affirming that the semantic regions of the face are represented in a consistent manner. These regions are chosen so that each region models at most one semantic part of the face (eyes/nose/mouth/ear) and the surface orientation encoded by different regions are different.

4. LEARNING WITH PATCHES

After the pre-processing stage, we are left with a training database of *corresponding* 2D-3D image patch pairs $\{\mathbf{X}, \mathbf{Z}\} = \{\mathbf{x^n}, \mathbf{z^n}\}_{n=1}^{N}$, where N is the total number of patches. In our experiments, we extract roughly 10000 overlapping patches per image. These patches are then split into $R = 8$ regions as explained in the Section 3. The objective of the training phase is to learn coupled dictionaries $\{\mathbf{D_x}, \mathbf{D_z}\}$ for each region separately such that the sparse representation of each 2D-3D pair are shared. We also require that the dictionary $\mathbf{D_{x(z)}}$ is able to sparsely represent the domain $\mathbf{X(Z)}$. In this section, we present the two formulations used in this work in terms of joint/coupled dictionary learning and how we use it to perform inference.

4.1. Joint Dictionary Learning (JDL)

This formulation is derived from the Joint Feature learning framework of [4] for single image-based super resolution. Here the sparsity structure between the two domains is constrained to be the same. The objective function should reflect that the reconstruction errors in each domain using the dictionary atoms belonging to that domain should be minimized, also making sure that a sparse combination of the dictionary atoms is used. Equation 1 implements this objective. λ offers a trade-off between the domain reconstruction error and the sparsity of $\boldsymbol{\alpha}$. In our experiments, we set $\lambda = 0.5$.

$$\min_{\mathbf{D_x}, \mathbf{D_z}, \boldsymbol{\alpha}} ||\mathbf{X} - \mathbf{D_x}\boldsymbol{\alpha}||_2^2 + ||\mathbf{Z} - \mathbf{D_z}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1 \qquad (1)$$

4.2. Optimizing (1)

It can be seen that (1) can be compactly written in the following form:

$$\min_{\tilde{\mathbf{D}}, \boldsymbol{\alpha}} ||\tilde{\mathbf{X}} - \tilde{\mathbf{D}}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1 \qquad (2)$$

where, $\tilde{\mathbf{X}} = [\mathbf{X}, \mathbf{Z}]^T$ and $\tilde{\mathbf{D}} = [\mathbf{D_x}, \mathbf{D_z}]^T$. Equation (2) is in the form of the traditional sparse coding problem and we use the publicly available SPAMS toolbox [15] for its solution.

4.3. Coupled Dictionary Learning (CDL)

The bi-level sparse coding approach from [5] offers a slightly different formulation of the same problem. In JDL, the sparsity structure is constrained to be the same across the two domains. Since this need not be the case, we can add an additional term to the objective in (1) where the reconstruction errors are minimized while the sparse codes across the domains are "similar" in a L2-norm sense. This gives rise to the objective function shown in (3).

$$\begin{aligned}
&\mathcal{J}(\mathbf{D_x}, \mathbf{D_z}, \boldsymbol{\alpha_x}, \boldsymbol{\alpha_z}) = \\
&\min_{\mathbf{D_x}, \mathbf{D_z}, \boldsymbol{\alpha_x}, \boldsymbol{\alpha_z}} ||\mathbf{X} - \mathbf{D_x}\boldsymbol{\alpha_x}||_2^2 + ||\mathbf{Z} - \mathbf{D_z}\boldsymbol{\alpha_z}||_2^2 \\
&+ \gamma||\boldsymbol{\alpha_x} - \boldsymbol{\alpha_z}||_2^2 \\
&\text{s.t} \quad \boldsymbol{\alpha_x} = \min_{\boldsymbol{\alpha}} ||\mathbf{X} - \mathbf{D_x}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1 \\
&\boldsymbol{\alpha_z} = \min_{\boldsymbol{\alpha}} ||\mathbf{Z} - \mathbf{D_z}\boldsymbol{\alpha}||_2^2 + \lambda||\boldsymbol{\alpha}||_1 \qquad (3)
\end{aligned}$$

In (3), the parameter λ serves the same purpose as in (1). The parameter γ trades off the similarity of the structure of the sparse codes belonging to each domain with the corresponding reconstruction errors. We set $\gamma = 1$.

4.4. Optimizing (3)

The coupled dictionary learning formulation is a bilevel optimization problem and highly non-convex. Hence, we use the stochastic gradient descent algorithm to solve the problem due to the large training size and the difficulty in gradient computation for each step. The optimization is done in two levels : Given initial estimates of $\{\mathbf{D_x}, \mathbf{D_z}\}$ the lower level (constraint level) is solved to give a feasible solution, which consists of the $\{\boldsymbol{\alpha_x}, \boldsymbol{\alpha_z}\}$ pair, using the SPAMS toolbox [15]. In the next step, $\{\boldsymbol{\alpha_x}, \boldsymbol{\alpha_z}\}$ are held fixed and the upper level optimization or the objective function $\mathcal{J}(\mathbf{D_x}, \mathbf{D_z})$, is solved by computing its gradient with respect to the variables $\mathbf{D_x}$ and $\mathbf{D_z}$. These two steps are performed iteratively until convergence. The convergence is determined by computing the objective value at each iteration over a validation set. In our experiments, we

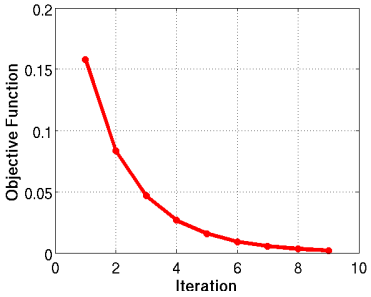

Fig. 3: Convergence plot of inference phase

found that using a batch size of 10 for the stochastic descent, the problem converges in a single pass over the training data. To get the update rule to perform the descent step, the gradients for (3) should be computed. The gradients for the p^{th} sample with respect to the variables of the problem is given as follows by applying the chain rule:

$$\frac{d\mathcal{J}_p}{d\mathbf{D_k}} = \frac{\partial \mathcal{J}_p}{\partial \mathbf{D_k}} + \frac{\partial \mathcal{J}_p}{\partial \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{p}}} \frac{\partial \boldsymbol{\alpha}_{\boldsymbol{k}}^{\boldsymbol{p}}}{\partial \mathbf{D_k}}, \tag{4}$$

where $\mathbf{k} = \{\mathbf{x}, \mathbf{z}\}$. The first two partial derivatives are computed as:

$$\frac{\partial \mathcal{J}_p}{\partial \mathbf{D_x}} = -2(\mathbf{x^p} - \mathbf{D_x}\boldsymbol{\alpha}_{\boldsymbol{x}}^{\boldsymbol{p}})\boldsymbol{\alpha}_{\boldsymbol{x}}^{\boldsymbol{pT}}$$
$$\frac{\partial \mathcal{J}}{\partial \alpha_{\mathbf{x}}} = -2\mathbf{D_x}^T(\mathbf{x^p} - \mathbf{D_x}\boldsymbol{\alpha}_{\boldsymbol{x}}^{\boldsymbol{p}}) + 2\gamma(\boldsymbol{\alpha}_{\boldsymbol{x}} - \boldsymbol{\alpha}_{\boldsymbol{z}})$$

and similarly for $\mathbf{z}$. The existence of the derivatives depends on the existence of the partial derivatives of the sparse code $\boldsymbol{\alpha}_{\boldsymbol{x(z)}}$ with respect to the corresponding dictionary $\mathbf{D}_{\mathbf{x(z)}}$. As pointed out in [5], there is no direct link between them and hence they have to be evaluated using implicit differentiation. The result of the analysis from [5] is being used here to compute the derivatives as follows:

$$\frac{\partial \boldsymbol{\alpha}_{\boldsymbol{x}}^{\boldsymbol{p}}}{\partial \mathbf{D_x}} = (\mathbf{D_x}^T\mathbf{D_x}^{-1})\left(\frac{\partial \mathbf{D_x}^T\mathbf{x^p}}{\partial \mathbf{D_x}} - \frac{\partial \mathbf{D_x}^T\mathbf{D_x}}{\partial \mathbf{D_x}}\boldsymbol{\alpha}_{\boldsymbol{x}}^{\boldsymbol{p}}\right)$$

Thus, the update rule for the Stochastic descent algorithm for solving the upper-level subproblem can be given as:

$$\mathbf{D_x}^{t+1} = \mathbf{D_x} - \frac{\eta^t}{B}\sum_{b=1}^{B}\frac{(\nabla\mathcal{J}_b)_{\mathbf{D_x}}}{||(\nabla\mathcal{J}_b)_{\mathbf{D_x}}||_2}$$

where η is chosen as $\frac{0.1}{\sqrt{t}}$ for the t^{th} iteration; $(\nabla\mathcal{J}_b)_{\mathbf{D_x}} \equiv \frac{d\mathcal{J}_b}{d\mathbf{D_x}}$ and B is the batch size. In both the optimizations above (4.1,4.3), the dictionary pair $\{\mathbf{D_x}, \mathbf{D_z}\}$ are initialized individually using the K-SVD method [16] and the dictionary atoms are normalized to have unit norm after each update step.

4.5. Inference

In the testing phase, we use a reference intensity-depth pair to guide our optimization process. Given an input $\hat{\mathbf{X}}$, we obtain the sparse codes using the intensity domain dictionary($\mathbf{D_x}$) and use them along with the depth dictionary ($\mathbf{D_z}$) to reconstruct the depth map corresponding to $\hat{\mathbf{X}}$. To impose a global shape constraint on the output depth map, we use the reference depth-map $\mathbf{Z_r}$ as a prior. Furthermore, to compute the depth for a pixel which belongs to a region r, we use the dictionary pair $\{\mathbf{D_x^r}, \mathbf{D_z^r}\}$. Since the test image is warped to the reference image, this region information is readily available along with the reference, as shown in Figure 3. The following objective function trades off between the reconstruction error and the similarity to the prior shape. We found that

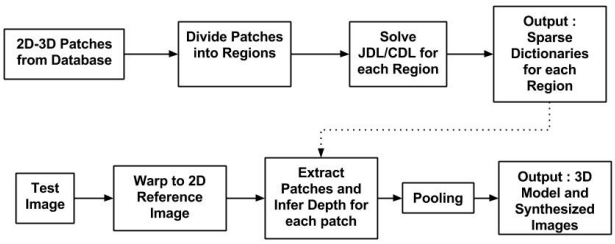

Fig. 4: Stepwise procedure for training (top row) and inference (bottom row)

$\beta = 0.2$ gives a better trade off on the validation set. We start with initializing $\hat{\alpha}_z$ as the solution of the sparse coding problem with image $\hat{\mathbf{X}}$ and dictionary $\mathbf{D_x}$. Then, (5) is solved to get the updated model estimate.

$$\min_{\hat{\mathbf{Z}}} ||\hat{\mathbf{Z}} - \mathbf{D_z}\hat{\boldsymbol{\alpha}}_{\boldsymbol{z}}||_F^2 + \beta||\hat{\mathbf{Z}} - \mathbf{Z_r}||_F^2 \tag{5}$$

The solution to (5) can be calculated in a straight forward manner to be: $\hat{\mathbf{Z}} = \frac{\mathbf{D_z}\hat{\boldsymbol{\alpha}}_{\boldsymbol{z}} + \beta\mathbf{Z_r}}{1+\beta}$. Then, for the JDL method, $\hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}$ is recomputed as the solution to Eq.(1) with $\mathbf{X} \equiv \hat{\mathbf{X}}, \mathbf{Z} \equiv \hat{\mathbf{Z}}$. $\mathbf{D_z}$. For the CDL method, $\hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}$ is recomputed as the solution to Eq. 6 using the Elastic-Net formulation in the SPAMS toolbox [15].

$$\min_{\hat{\boldsymbol{\alpha}_{\boldsymbol{x}}}, \hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}} ||\hat{\mathbf{X}} - \mathbf{D_x}\hat{\boldsymbol{\alpha}_{\boldsymbol{x}}}||_F^2 + ||\hat{\mathbf{Z}} - \mathbf{D_z}\hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}||_F^2$$
$$+ \lambda(||\hat{\boldsymbol{\alpha}_{\boldsymbol{x}}}||_1 + ||\hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}||_1) + \gamma||\hat{\boldsymbol{\alpha}_{\boldsymbol{x}}} - \hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}||_2^2 \tag{6}$$

Using the recomputed $\hat{\boldsymbol{\alpha}_{\boldsymbol{z}}}$, $\hat{\mathbf{Z}}$ is computed again and this procedure is performed until convergence, that is, until there is no change in the model output. We found that it takes typically 6-7 iterations to converge as shown in Figure 3. The JDL method converges faster but the CDL method produces slightly better results visually. Both the procedures converge in 60-70s using a naively implemented MATLAB code. In practice, (5)-(6) are solved on overlapping patches and hence their values should be pooled together to give a single depth estimate at any given pixel. In this step, we perform a distance based weighted pooling which is explained as follows: Any given pixel is covered by multiple overlapping patches and hence if a pixel is at the center of a patch then its estimate of the depth value should be weighted more as compared to that of a patch where the pixel appears in a corner. Thus, we perform a weighted pooling where the closer the pixel is to the center of the patch, the higher the depth estimate of that patch is weighted. We found that this method gives much better visual results as compared to max or average pooling which give very noisy or very smooth results respectively. We choose the β value such that the resulting 3D shape does not overfit the prior. Based on experiments over the validation set, we fix $\beta = 0.2$. Figure 4 provides a stepwise procedure for training and inference stages.

5. EXPERIMENTS AND RESULTS

We show the effectiveness of our modeling paradigm by conducting both qualitative and quantitative experiments on the 3D Face dataset: USF-HUMAN ID [1]. To generate training data, we use the 3D model files provided to render frontal views of 100 individuals with neutral expressions. For training and inference, we set the patch size as 16 and the number of dictionary atoms to 512. These were chosen as a trade-off between faster computation and modeling accuracy. We show the synthesised 2D non-frontal images from the resultant 3D models in Fig. 5. In order to measure the accuracy

Fig. 5: Synthesized faces using the 3D model output. Input images are shown in the first column and the synthesized images are shown in columns 2 to 7.

Fig. 6: Left row shows the 2D input images; Right row shows the 3D model produced by the proposed method, texture mapped with the input image.

of the modeling procedure, we use the following metrics (z is the Ground Truth (GT) depth image and $\hat{z}$ is the model estimate, both of dimension N): Root mean Square error(RMSE): $\sqrt{\frac{||z-\hat{z}||_2^2}{N}}$, log10 error: $\frac{1}{N}\sum_{i=1}^{N}|log10(z)-log10(\hat{z})|$. Aside from these commonly used metrics, we use two new metrics which are independent of the range of the output values and do not strictly penalize pixel-wise inaccuracies, like the previous metrics do. The mean normal error (MNE) penalizes the difference in the surface orientation between the GT and the model estimate. Given a depth map z, the surface normals $\vec{n}$ of the underlying 3D point cloud is calculated using standard methods, for the GT and the model estimate. MNE is the computed by measuring the difference between the normal orientations: $\frac{1}{N}\sum_{i=1}^{N} cos^{-1}(n_i \cdot \hat{n}_i)$, where the normals are normalized to unit norm. Thus, MNE penalizes the difference in the 3D surface orientation rather than the actual depth values. MNE-V refers to the average error for per vertex normals and MNE-F refers to the per face normals, where the vertex and faces are from the triangulated 3D model output. The reported values for MNE are in radians.

	Error	Baseline 1	DT [12]	JDL	CDL
USF	log10	0.7386	**0.2882**	0.3181	0.31
	RMSE	0.1323	**0.0837**	0.1174	0.1028
	MNE-V	0.4425	0.4538	**0.3355**	0.3436
	MNE-F	0.4915	0.4834	**0.3766**	0.384

Table 1: Quantitative results on USF dataset.

5.1. 3D Model Synthesis Results

The dataset consists of 100 pairs of 2D intensity and range images each in neutral expression and frontal pose. We split the dataset as 20 pairs for training, 10 for validation and 70 for testing such that each image gets to be in the training once and hence resulting in 5 runs. The reported results are averaged across these runs. As a baseline we use two comparisons. For baseline 1, we take the reference depth image and consider that as the model estimate. For Baseline 2, we use the Depth Transfer (DT) approach. It should be noted that there is no training required for DT but it requires the existence of the 2D-3D pairs during inference. To make a fair comparison, we use the same 20 training pairs for each run for DT and allow one reference image. It should be noted that the DT code was not optimized for any parameters and default settings were used. While both the proposed formulations give very close numerical results in all the metrics used, we found that the CDL method gives visually better results in slightly more cases in the validation set as compared to the JDL method. Hence the qualitative results shown here are using the CDL method. Figure 5 shows the synthesized faces using the 3D model outputs for the input images shown in the first column. More results are shown in Figure 6.

6. CONCLUSION AND FUTURE WORK

In this work, we proposed a supervised learning approach for 3D model estimation approach with applications in synthesizing novel views of faces. Due to the flexibility of the learning formulation, the same methodology with simple changes can be applied to any class of images. In future, we plan to formulate the model estimation problem as a convolutional sparse coding problem. This can be made to work in a feed forward way and can be recast as a deep learning problem thus making it significantly faster and more accurate. Another natural extension of this formulation that is being worked on is to derive 3D models of faces appearing over a video sequence, performing model updates in an online manner.

7. REFERENCES

[1] "USF DARPA HumanID 3D Face Database. Courtesy of Professor Sudeep Sarkar, University of South Florida.," Dec.

[2] Berthold K. P. Horn and Michael J. Brooks, "The variational approach to shape from shading," *Computer Vision, Graphics, and Image Processing*, vol. 33, no. 2, pp. 174–208, 1986.

[3] A. Saxena, Min Sun, and A.Y. Ng, "Make3d: Learning 3d scene structure from a single still image," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 31, no. 5, pp. 824–840, May 2009.

[4] Jianchao Yang, John Wright, Thomas S Huang, and Yi Ma, "Image super-resolution via sparse representation.," *IEEE transactions on image processing : a publication of the IEEE Signal Processing Society*, vol. 19, no. 11, pp. 2861–73, Dec. 2010.

[5] Jianchao Yang, Zhaowen Wang, Zhe Lin, Xianbiao Shu, and T. Huang, "Bilevel sparse coding for coupled feature spaces," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 2360–2367.

[6] Shenlong Wang, D. Zhang, Yan Liang, and Quan Pan, "Semi-coupled dictionary learning with applications to image super-resolution and photo-sketch synthesis," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June 2012, pp. 2216–2223.

[7] C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba, "Hoggles: Visualizing object detection features," in *Computer Vision (ICCV), 2013 IEEE International Conference on*, Dec 2013, pp. 1–8.

[8] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1063–1074, Sept. 2003.

[9] Oswald Aldrian and William Smith, "A Linear Approach to Face Shape and Texture Recovery using a 3D Morphable Model," *Procedings of the British Machine Vision Conference 2010*, pp. 75.1–75.10, 2010.

[10] 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*, vol. 33, no. 2, pp. 394–405, Mar. 2011.

[11] Tal Hassner, "Viewing Real-World Faces in 3D," *2013 IEEE International Conference on Computer Vision*, pp. 3607–3614, Dec. 2013.

[12] K. Karsch, Ce Liu, and Sing Bing Kang, "Depth transfer: Depth extraction from video using non-parametric sampling," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 36, no. 11, pp. 2144–2158, Nov 2014.

[13] A. Asthana, S. Zafeiriou, Shiyang Cheng, and M. Pantic, "Robust discriminative response map fitting with constrained local models," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013, pp. 3444–3451.

[14] Y. Taigman, Ming Yang, M. Ranzato, and L. Wolf, "Deepface: Closing the gap to human-level performance in face verification," in *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, June 2014, pp. 1701–1708.

[15] Julien Mairal, Francis Bach, Jean Ponce, and Guillermo Sapiro, "Online dictionary learning for sparse coding," in *Proceedings of the 26th Annual International Conference on Machine Learning*, New York, NY, USA, 2009, ICML '09, pp. 689–696, ACM.

[16] M. Aharon, M. Elad, and A. Bruckstein, "k -svd: An algorithm for designing overcomplete dictionaries for sparse representation," *Signal Processing, IEEE Transactions on*, vol. 54, no. 11, pp. 4311–4322, Nov 2006.