# 3D Morphable Models as Spatial Transformer Networks

Anil Bas*, Patrik Huber†, William A. P. Smith*, Muhammad Awais†, Josef Kittler†
*Department of Computer Science, University of York †Centre for Vision, Speech and Signal Processing, University of Surrey

ICCV 17

## Summary

This study shows how to use a 3D morphable model as a spatial transformer within a convolutional neural network. It is an extension of the original spatial transformer network [1] in that we are able to interpret and normalise 3D pose changes and self-occlusions. The network (specifically, the localiser part of the network) learns to fit a 3D morphable model to a single 2D image without needing labelled examples of fitted models.

## Contributions

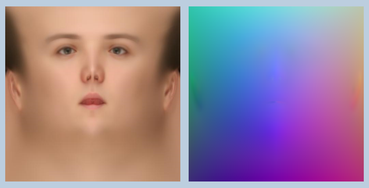
The proposed architecture is based on a purely geometric approach in which only the shape component of a 3DMM is used to geometrically normalise an image.

The method can be trained in an unsupervised fashion, and thus does not depend on synthetic training data or the fitting results of an existing algorithm.

In contrast to all previous 3DMM fitting networks, the output of our 3DMM-STN is a 2D resampling of the original image which contains all of the high frequency, discriminating detail in a face rather than a model-based reconstruction which only captures the gross, low frequency aspects of appearance that can be explained by a 3DMM.

## Sampling

**The output grid** is a flattened 3DMM mesh in 2D texture space in which the images are in dense, pixel-wise correspondence.



A Tutte embedding and a geometry of the mean shape

**Self-occlusions** The occluded vertices can be computed exactly using ray-tracing or z-buffering or they can be precomputed and stored in a lookup table. For efficiency, we approximate occlusion by only computing which vertices have backward facing normals.

**Masking layer** combines the sampled image and the visibility map via pixel-wise products.

## Source Code

The source code (MatConvNet implementation) is available at `https://github.com/anilbas/3DMMasSTN`

## Geometric Loss Functions

**Landmark loss** minimises the Euclidean distance between observed and predicted 2D points. Given $L$ landmark locations $\mathbf{l}_1, \ldots, \mathbf{l}_L$ and associated detection confidence values $c_1, \ldots, c_L$, we computed a weighted Euclidean loss:

$$\ell_{\text{landmark}} = \sum_{i=1}^{L} c_i \|\mathbf{L}_i - \mathbf{l}_i\|^2. \quad (1)$$

**Bilateral symmetry loss** measures asymmetry of the sampled face texture over visible pixels.

$$\ell_{\text{sym}} = \sum_{i=1}^{N} \sum_{c=1}^{3} M_{x_i^t, y_i^t} M_{x^t_{\text{sym}(i)}, y^t_{\text{sym}(i)}} (V_i^c - V^c_{\text{sym}(i)})^2. \quad (2)$$

**Siamese multi-view fitting loss** penalises differences between multiple images of the same face in different poses. Siamese training would perform where a pair of images in different poses were sampled into images $V_i^c$ and $W_i^c$ with visibility masks $\mathbf{M}$ and $\mathbf{N}$ giving a loss:

$$\ell_{\text{multiview}} = \sum_{i=1}^{N} \sum_{c=1}^{3} M_{x_i^t, y_i^t} N_{x_i^t, y_i^t} (V_i^c - W_i^c)^2. \quad (3)$$

**Statistical prior loss** minimises an appearance error, regularising the statistical shape prior which can be encoded by the following loss function:

$$\ell_{\text{prior}} = \|\boldsymbol{\alpha}\|^2. \quad (4)$$

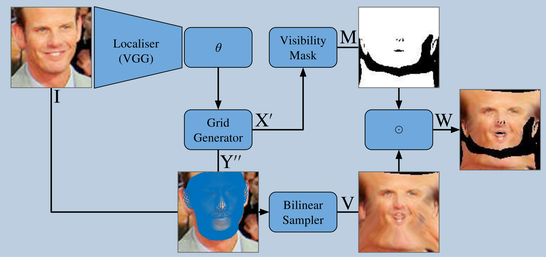
## References

[1] M. Jaderberg, K. Simonyan, A. Zisserman and K. Kavukcuoglu. Spatial Transformer Networks. In Proc. *NIPS'15*, 2015.

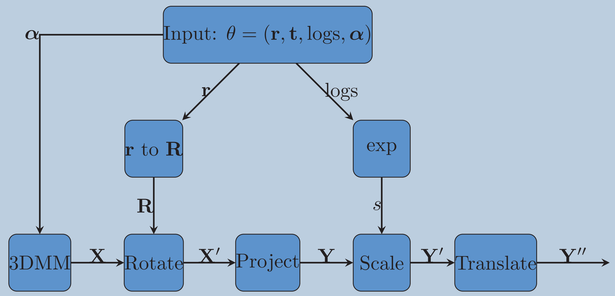
[2] O. M. Parkhi, A. Vedaldi and A. Zisserman. Deep Face Recognition. In Proc. *BMVC'15*, 2015.

[3] A. T. Trãn, T. Hassner, I. Masi and G. Medioni. Regressing Robust and Discriminative 3D Morphable Models with a very Deep Neural Network. In Proc. *CVPR*, 2017.

## CNN Architecture




Overview of the 3DMM-STN




The grid generator network within a 3DMM-STN

**Localiser network** is a CNN that takes an image as input and regresses the pose and shape parameters, $\theta$, of the face in the image. Specifically, we predict the following vector of parameters:

$$\theta = (\underbrace{\mathbf{r}, \mathbf{t}, \text{logs}}_{\text{pose}}, \underbrace{\boldsymbol{\alpha}}_{\text{shape}}). \quad (5)$$

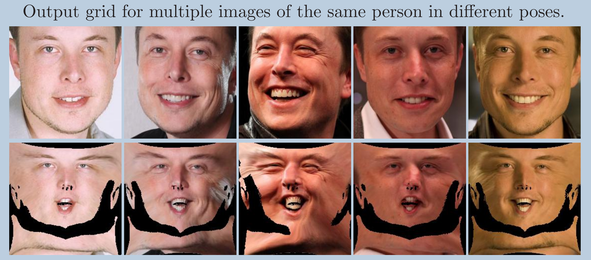
Here, $\mathbf{t} \in \mathbb{R}^2$ is a 2D translation, $\mathbf{r} \in \mathbb{R}^3$ is an axis-angle representation of a 3D rotation with rotation angle $\|\mathbf{r}\|$ and axis $\mathbf{r}/\|\mathbf{r}\|$. Since scale must be positive, we estimate log scale and later pass this through an exponentiation layer, ensuring that the estimated scale is positive. The shape parameters $\boldsymbol{\alpha} \in \mathbb{R}^D$ are the principal component weights used to reconstruct the shape.

For our localiser network, we use the pretrained VGG-Faces [2] architecture, delete the classification layer and add a new fully connected layer with $6 + D$ outputs.

**Grid generator network** combines a linear statistical model with a scaled orthographic projection. We apply a 3D transformation and projection to a 3D mesh that comes from the morphable model. The intensities sampled from the source image are then assigned to the corresponding points in a flattened 2D grid.

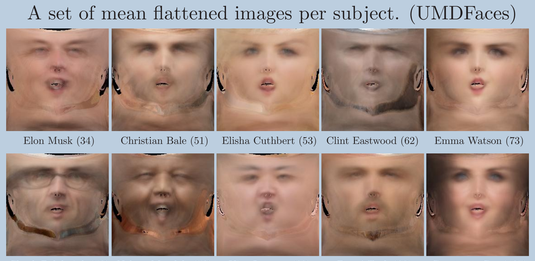
The sample points in our grid generator are determined by the transformation parameters $\theta$ estimated by the localiser network.

Output grid for multiple images of the same person in different poses.

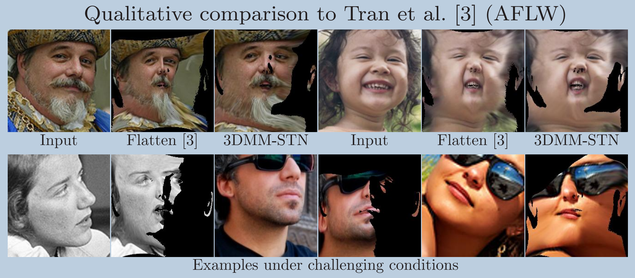


## Experiments

A set of mean flattened images per subject. (UMDFaces)




Qualitative comparison to Tran et al. [3] (AFLW)




Examples under challenging conditions

## Acknowledgements

We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan X Pascal GPU used for this research. We also thank BMVA for kindly providing a travel bursary.