Real-time 3D Face Fitting and Texture Fusion on In-the-wild Videos

Patrik Huber, William Christmas, Josef Kittler
Centre for Vision, Speech and Signal Processing
University of Surrey
Guildford, GU2 7XH, United Kingdom
Contact: http://www.patrikhuber.ch

Philipp Kopp, Matthias Rätsch
Image Understanding and Interactive Robotics
Reutlingen University
D-72762 Reutlingen, Germany

***Abstract*—We present a fully automatic approach to real-time 3D face reconstruction from monocular in-the-wild videos. With the use of a cascaded-regressor based face tracking and a 3D Morphable Face Model shape fitting, we obtain a semi-dense 3D face shape. We further use the texture information from multiple frames to build a holistic 3D face representation from the video footage. Our system is able to capture facial expressions and does not require any person-specific training. We demonstrate the robustness of our approach on the challenging *300 Videos in the Wild* (300-VW) dataset. Our real-time fitting framework is available as an open source library at http://4dface.org.**

I. Introduction

This paper addresses the problem of reconstructing a 3D face from monocular in-the-wild videos. While the problem has been studied in the past, existing algorithms rely either on depth (RGB-D) data or have not demonstrated their robustness on realistic in-the-wild videos.

From the algorithms working on monocular video sequences, the method of Garrido et al. [1] requires manual, subject-specific training and labelling. Moreover, it has only been evaluated on a limited set of HD quality videos under rather controlled conditions, with frontal poses. Ichim et al. [2] also require subject-specific training and manual labelling by an experienced labeler. This takes several minutes per subject, and their resulting model is person-specific. In addition to subject-specific manual training being a tedious step, creating a personalised face model *offline* is not possible where the subject can not be seen beforehand, e.g. for face recognition in the wild, customer tracking for behaviour analysis or various human-computer interaction scenarios. Jeni et al. [3] use rendered 3D meshes to train their algorithm, which do not contain the variations that occur in 2D in-the-wild images; for example, the meshes have to be rendered on random backgrounds. Furthermore, they only evaluate their method by cross-validation on the same 3D data their algorithm has been trained on. Cao et al. [4], [5] reconstruct only shape, without using texture, and do not perform evaluation on in-the-wild videos. Cao et al. [6] don't require user-specific training, but present only results in controlled conditions involving frontal pose and high image resolution and they require a GPU to achieve real-time performance.

In contrast to these approaches, we present an approach that requires no subject-specific training and evaluate it on a challenging 2D in-the-wild video data set. We are the first to carry out such an evaluation of a 3D face reconstruction algorithm on in-the-wild data with challenging pose and light variations as well as limited resolution and show the robustness of our algorithm. While many of the previous works focus on face re-enactment, we focus on a high-quality texture representation of the subject in front of the camera. Our approach runs in near real-time on a CPU.

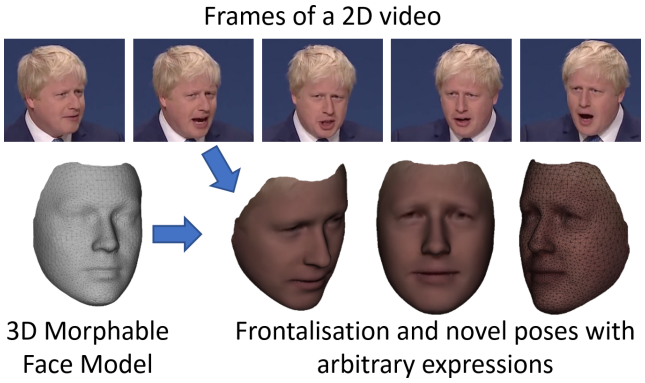

Fig. 1. Real-time 3D face reconstruction from a monocular in-the-wild video stream. Our method uses a 3D Morphable Face Model as face prior and fuses the information from multiple frames to create a holistic 3D face reconstruction, without requiring subject-specific training.

This paper presents the following contributions. By combining cascaded regression with 3D Morphable Face Model fitting, we obtain real-time face tracking and semi-dense 3D shape estimates from low-quality 2D webcam videos. We present an approach to fuse the face texture from multiple video frames to yield a holistic textured face model. We demonstrate the applicability of our method to in-the-wild videos on the challenging 300-VW database [7] that includes scenarios such as speeches and TV shows. In addition, we propose a linear method to fit both shape identity and expressions by extending an existing shape fitting method. Finally, our method is available as open-source software on GitHub.

II. Method

In general, reconstructing a 3D face from 2D data is an ill-posed problem. To make this task feasible, our approach incorporates a 3D Morphable Face Model (3DMM) to provide

prior knowledge about faces. In this section, we first briefly introduce the 3D Morphable Face Model we use. We then present our 3D face reconstruction approach and the texture fusion.

A. 3D Morphable Face Model

A 3D Morphable Model (3DMM) is based on three-dimensional meshes of faces that have been registered to a reference mesh, i.e. are in dense correspondence. A face is represented by a vector $\mathbf{S} \in \mathbb{R}^{3N}$, containing the x, y and z components of the shape, and a vector $\mathbf{T} \in \mathbb{R}^{3N}$, containing the per-vertex RGB colour information. N is the number of mesh vertices. The 3DMM consists of two PCA models, one for the shape and one for the colour information, of which we only use the shape model in this paper. Each PCA model consists of the mean of the model $\bar{\mathbf{v}} \in \mathbb{R}^{3N}$, a set of principal components $\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_M] \in \mathbb{R}^{3N \times M}$, and a vector of standard deviations $\boldsymbol{\sigma} \in \mathbb{R}^M$. M is the number of principal components, capturing 95% of the variance. Novel faces can be generated by calculating

$$\mathbf{S} = \bar{\mathbf{v}} + \sum_i^M \alpha_i \sigma_i \mathbf{v}_i \tag{1}$$

for the shape, where the vector $\boldsymbol{\alpha} \in \mathbb{R}^M$ conveys a set of 3D face instance coordinates in the shape PCA space.

Throughout this paper, we use the *Surrey Face Model* [8] with $N = 3448$ for our experiments, which provided adequate representation of shape.

B. Face Tracking

To track the face in each frame of a video, we use a cascaded-regression-based approach, similar to Feng et al. [9], to regress a set of sparse facial landmark points. The goal is to find a regressor $R : \mathbf{f}(\mathbf{I}, \boldsymbol{\theta}) \rightarrow \delta\boldsymbol{\theta}$, where $\mathbf{f}(\mathbf{I}, \boldsymbol{\theta})$ is a vector of image features extracted from the input image, given the current model parameters $\boldsymbol{\theta}$, and $\delta\boldsymbol{\theta}$ is the predicted model parameter update. This mapping is learned from a training dataset using a series of linear regressors $\{R_n\}$, where

$$R_n : \delta\boldsymbol{\theta} = \mathbf{A}_n \mathbf{f}(\mathbf{I}, \boldsymbol{\theta}) + \mathbf{b}_n, \tag{2}$$

and $\mathbf{A}_n$ is the projection matrix, $\mathbf{b}_n$ is the offset (bias) of the n-th regressor, and $\mathbf{f}(\mathbf{I}, \boldsymbol{\theta})$ extracts HOG (histogram of oriented gradients) features from the image.

When run on a video stream, the regression is initialised at the location from the previous frame but with the model's mean landmarks, which acts as a regularisation.

C. 3D Model Fitting

In subsequent steps, the 3D Morphable Model is fitted to the subject in a frame. This section describes our camera model, the PCA shape fitting, and subsequent refinement using facial expressions and contour landmarks.

Camera model From the 2D landmark locations and their known correspondences in the 3D Morphable Model, we estimate the pose of the camera. We assume an affine camera model and implement the *Gold Standard Algorithm* of Hartley & Zisserman [10], which finds a least-squares approximation of a camera matrix $\mathbf{C} \in \mathbb{R}^{3\times4}$ given the 2D - 3D point pairs.

Shape fitting Given the estimated camera pose, the 3D shape model is fitted to the sparse set of 2D landmarks to produce an identity-specific semi-dense 3D shape. We find the most likely vector of PCA shape coefficients $\boldsymbol{\alpha}$ by minimising the following cost function:

$$\mathbb{E} = \sum_{i=1}^{3L} \frac{(y_{p,i} - y_i)^2}{2\sigma_{2D,i}^2} + \|\boldsymbol{\alpha}\|_2^2, \tag{3}$$

where L is the number of landmarks, $\mathbf{y}$ is a stacked vector of detected or labelled 2D landmarks in homogeneous coordinates, $\boldsymbol{\sigma}_{2D}^2$ are the variances of these landmark points, and $\mathbf{y_p}$ is a stacked vector of the 3D Morphable Model shape points that correspond to the respective 2D landmarks, projected to 2D using the estimated camera matrix. More specifically, $\mathbf{y_p} = \mathbf{P} \cdot (\hat{\mathbf{V}}_h \boldsymbol{\alpha} + \bar{\mathbf{v}})$, where $\mathbf{P}$ is a matrix that has copies of the camera matrix $\mathbf{C}$ on its diagonal, and $\hat{\mathbf{V}}_h$ is a modified PCA basis matrix that consists of the rows that correspond to the landmark points that the shape is fitted to. The basis vectors are multiplied with the square root of their respective eigenvalue, and, because the derivation is expressed in homogeneous coordinates, a row of zeros is inserted after every third row. With this formulation, the cost function in Eq. (3) can be expressed in terms of a regularised quadratic form, which has a closed form solution (derived in [11]): $\boldsymbol{\alpha} = -(\hat{\mathbf{V}}_h^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \boldsymbol{\Omega} \mathbf{P} \hat{\mathbf{V}}_h + \lambda \mathbf{I})^{-1} (\hat{\mathbf{V}}_h^{\mathrm{T}} \mathbf{P}^{\mathrm{T}} \boldsymbol{\Omega}^{\mathrm{T}} (\mathbf{P}\bar{\mathbf{v}} - \mathbf{y}))$, where $\boldsymbol{\Omega} = \mathrm{diag}(\boldsymbol{\sigma}_{2D}^{-2})$.

D. Expression Fitting

To model expressions, we use a set of additive expression blendshapes $\mathbf{B}$ that have been computed from 3D expression scans. A linear combination of these blendshapes is added to the PCA model, so a face shape is represented as:

$$\mathbf{S} = \bar{\mathbf{v}} + \sum_i^M \alpha_i \sigma_i \mathbf{v}_i + \sum_j^K \psi_j \mathbf{B}_j, \tag{4}$$

where $\mathbf{B}_j$ is the j-th column of $\mathbf{B}$ (the j-th blendshape) and ψ_j the corresponding blendshape coefficient.

To find the blendshape coefficients, we propose an extension of [11] that fits expression blendshapes, and then alternates to fit both identity and expression. We use a standard least-squares formulation, similar to Section II-C, but instead of using the mean shape $\bar{\mathbf{v}}$, we substitute it with a face instance $\mathbf{S}^{\boldsymbol{\alpha}}$, generated with the currently estimated $\boldsymbol{\alpha}$. $\hat{\mathbf{V}}_h$ is replaced with $\hat{\mathbf{B}}_h$, where $\hat{\mathbf{B}}_h$ is modified from $\mathbf{B}$ in the same way as $\hat{\mathbf{V}}_h$, and we set: $\boldsymbol{\psi} = -(\mathbf{P}\hat{\mathbf{B}}_h)^{-1}(\mathbf{P}\mathbf{S}^{\boldsymbol{\alpha}} - \mathbf{y})$. We solve this system of equations with a Non-Negative Least Squares algorithm [12].

Once an estimate of the blendshape coefficients $\boldsymbol{\psi}$ is computed, we generate a shape instance $\mathbf{S}^{\boldsymbol{\psi}}$ using these estimated coefficients, and use this face instance in the identity fitting instead of the mean face $\bar{\mathbf{v}}$. This process of shape identity and expression blendshape fitting is alternated, and usually converges within ten iterations.

The result of the fitting is the identity-specific shape coefficients $\boldsymbol{\alpha}$ and expression blendshape coefficients $\boldsymbol{\psi}$. Besides modelling the subject's expressions, blendshape fitting can be used to remove a facial expression from a subject, or to re-render it with a different expression. Figure 2 shows a frame with a strong expression, the expression-neutralised face, and a re-rendering with a synthesised expression.

Fig. 2. Frame with strong expression and expression-neutralised image. *(left)*: Input frame. *(middle)*: Expression-neutralised 3D model. *(right)*: Face with artificially added smile expression.

E. Contour Refinement

In general, the outer face contours present in the 2D image do not correspond to unique contours on the 3D model. At the same time, these contours are important for an accurate face reconstruction, as they define the boundary region of the face. This problem has had limited attention in the research community, but for example Bas et al. [13] recently provided an excellent overview describing the problem in more detail.

To deal with this problem of contour correspondences, we introduce a simple contour fitting that fits the front-facing face contour given semi-fixed 2D-3D correspondences. We assume that the front-facing contour (that is, the half of the contour closer to the camera, for example the right face contour when a subject looks to the left) corresponds to the outline of the model. We thus define a set of vertices V along the outline of the 3D face model, and then, given an initial fit, search for the closest vertex in that list for each detected 2D contour point. Given a 2D contour landmark y, the optimal corresponding 3D vertex $\hat{v}$ is chosen as:

$$\hat{v} = \underset{v \in V}{\arg\min} \left\| Pv - y \right\|^2, \tag{5}$$

where P is the currently estimated projection matrix from 3D to 2D. V is small, and we find $\hat{v}$ by computing all distances.

Using a whole set of potential 3D contour vertices makes the method robust against varying roll and pitch angles, as well as against vertical inaccuracies of the contour from the landmark regressor. Once found, these contour correspondences are then used as additional corresponding points in the algorithm described in II-C and II-D.

F. Texture Reconstruction

Once an accurate model fit is obtained, we remap the image texture from a frame to an isomap that puts each pixel into a globally registered representation. The isomap is a texture map, created by projecting the 3D model's triangles to 2D while preserving the geodesic distances between vertices ([14], [15]). The mapping is computed only once, so the isomaps of all of the frames are in dense correspondence with each other. Note that the texture map resolution is independent of the number of vertices N of the shape model.

Inspired by [16], we compute a weighting ω for each point in the isomap that is given by the angle of the camera viewing direction $\mathbf{d}$ and the normal $\mathbf{n}$ of the 3D mesh's triangle that corresponds to the point: $\omega = \langle \mathbf{d}, \mathbf{n} \rangle$. Thus, vertices that are facing away from the camera receive a lower weighting, and self-occluded regions are discarded. In contrast to [16], our approach does not depend on the colour model or an illumination model fitting. Figure 3 shows an example.

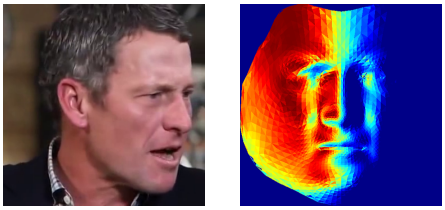

Fig. 3. View visibility information (including regions of self-occlusions) from the 3D face model. *(left)*: Input frame. *(right)*: red = 0°(facing the camera), blue = 90°or facing away. JET colourmap.

To reconstruct the texture value $\hat{c}_{x,y}$ at each pixel location (x, y), we calculate a weighted average of all frames up to the current frame n, each pixel weighed by its triangle's computed ω of a particular frame:

$$\hat{c}^n_{x,y} = \frac{1}{n} \sum_{i=1}^{n} \omega^i_{x,y} c^i_{x,y}, \tag{6}$$

where $c^i_{x,y}$ is the colour of frame i at location (x, y).

In practice, this average can be computed very efficiently, i.e. by adding the values of the current frame to the previous average and normalising accordingly, without having to recompute the values for all previous frames. Naturally, there is a trade-off between coverage and blurring with respect to the number of frames - to address this is the subject of future work. While more complex fusion techniques could be applied, our method is particularly suited for real-time application and in that it allows the computation of an incremental texture model on a video stream, without having knowledge of the whole video in advance.

III. Experiments

A. Landmark Accuracy

First, we evaluate the proposed approach on the ibug-Helen test set [17] to be able to compare the landmark accuracy to other approaches in the literature. We train a model using the algorithm of Section II-B, using F-HOG features and 5 cascaded linear regressors in series. On the official ibug-68 landmarks set, we achieve a mean error of 0.049, measured in percent of the distance between the outer eye corners, as defined by the official ibug protocol (which they refer to as inter-eye distance, *IED*). The algorithm was initialised with bounding boxes given by the ibug face detector. Table I shows a comparison with recent state-of-the-art methods.

Fig. 4. *(Top row)*: Frame from the original video. *(Second row)*: Reconstructed face texture using our real-time method. *(Third row)*: Ground truth face texture. *(bottom row)*: Rendering of a novel pose.

TABLE I
LANDMARK ERROR (IN % OF IED)

	SDM [18]	**ERT** [19]	**Ours**
HELEN	0.059	0.049	0.049
300-VW	-	-	0.047

To evaluate the accuracy of our tracking and the landmarks used for the shape reconstruction on in-the-wild videos, we evaluate the proposed approach on the public part of the 300-VW dataset [7]. For the amount of subject and camera movements present in 300-VW, the tracking did not require reinitialisation. Across all videos, our tracking achieves an average error of 0.047. Figure 5 shows the accuracy of each individual landmark. Our approach achieves competitive results even on challenging video sequences. Given that all 300-VW data is annotated semi-automatically, and the ground truth contour landmarks (1-8, 10-17) are not well-defined and vary largely along the face contour, we believe this to be very close to the optimum achievable accuracy.

All the results in this paper were achieved by training on databases from sources other than 300-VW.

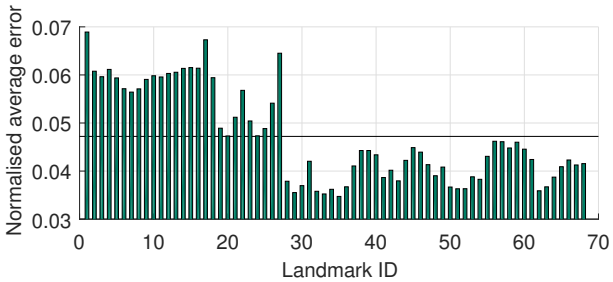

Fig. 5. Accuracy of each individual landmark on the 300-VW videos. 1-8 and 10-17 are contour landmarks, with significantly higher error. The horizontal bar depicts the average error.

B. Face Reconstruction

Our main experiment is concerned with reconstructing the 3D face and texture from in-the-wild video sequences. Since for such video sequences, no 3D ground truth is available, we evaluate the reconstruction quality on the texture map, since it accounts for shape as well as texture reconstruction accuracy. We create a ground truth isomap for ten 300-VW videos, by manually merging a left, frontal and right view, generated from accurate manual landmarks. We then compare our fully automatic reconstruction with these reference isomaps.

Figure 4 shows results of ibug 300-VW reconstructions. Our pipeline copes well with changing background, challenging poses, and, to some degree, varying illumination. The weighted fusion works well in these challenging conditions and results in a holistic, visually appealing reconstruction of the full face. The averaging results in slight blurring, but produces consistent results.

IV. CONCLUSION

We presented an approach for real-time 3D face reconstruction from monocular in-the-wild videos. The algorithm is competitive in landmark tracking and succeeds at reconstructing a shape and textural face representation, fusing different frames and view-angles. In comparison with existing work, the proposed algorithm requires no subject-specific or manual training, reconstructs texture as well as a semi-dense shape, and it is evaluated on a true in-the-wild video database.

Furthermore, the 3D face model and the fitting library are available at https://github.com/patrikhuber/eos. In future work, we plan to improve the real-time texture fusion with a method that reduces the blur caused by averaging, while still being robust to in-the-wild conditions.

ACKNOWLEDGMENTS

This work is in part supported by the Centre for Vision, Speech and Signal Processing of the University of Surrey, UK. Partial support from the EPSRC Programme Grant EP/N007743/1 is gratefully acknowledged.

REFERENCES

[1] P. Garrido, L. Valgaert, C. Wu, and C. Theobalt, "Reconstructing detailed dynamic face geometry from monocular video," *ACM Trans. Graph.*, vol. 32, no. 6, pp. 158:1–158:10, Nov. 2013. [Online]. Available: http://doi.acm.org/10.1145/2508363.2508380

[2] A. E. Ichim, S. Bouaziz, and M. Pauly, "Dynamic 3D avatar creation from hand-held video input," *ACM Trans. Graph.*, vol. 34, no. 4, pp. 45:1–45:14, Jul. 2015.

[3] L. Jeni, J. Cohn, and T. Kanade, "Dense 3D face alignment from 2D videos in real-time," in *Automatic Face and Gesture Recognition (FG), 2015 11th IEEE International Conference and Workshops on*, vol. 1, May 2015, pp. 1–8.

[4] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D shape regression for real-time facial animation," *ACM Trans. Graph.*, vol. 32, no. 4, pp. 41:1–41:10, Jul. 2013. [Online]. Available: http://doi.acm.org/10.1145/2461912.2462012

[5] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," *ACM Trans. Graph.*, vol. 33, no. 4, pp. 43:1–43:10, Jul. 2014. [Online]. Available: http://doi.acm.org/10.1145/2601097.2601204

[6] C. Cao, D. Bradley, K. Zhou, and T. Beeler, "Real-time high-fidelity facial performance capture," *ACM Trans. Graph.*, vol. 34, no. 4, p. 46, 2015. [Online]. Available: http://doi.acm.org/10.1145/2766943

[7] J. Shen, S. Zafeiriou, G. G. Chrysos, J. Kossaifi, G. Tzimiropoulos, and M. Pantic, "The first facial landmark tracking in-the-wild challenge: Benchmark and results," in *2015 IEEE International Conference on Computer Vision Workshop, ICCV Workshops 2015, Santiago, Chile, December 7-13, 2015*. IEEE, 2015, pp. 1003–1011. [Online]. Available: http://dx.doi.org/10.1109/ICCVW.2015.132

[8] P. Huber, G. Hu, R. Tena, P. Mortazavian, W. P. Koppen, W. Christmas, M. Rätsch, and J. Kittler, "A multiresolution 3D Morphable Face Model and fitting framework," in *International Conference on Computer Vision Theory and Applications (VISAPP)*, 2016. [Online]. Available: http://dx.doi.org/10.5220/0005669500790086

[9] Z.-H. Feng, P. Huber, J. Kittler, W. Christmas, and X.-J. Wu, "Random cascaded-regression copse for robust facial landmark detection," *IEEE Signal Processing Letters*, vol. 22, no. 1, pp. 76–80, Jan 2015. [Online]. Available: http://dx.doi.org/10.1109/LSP.2014.2347011

[10] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, 2004.

[11] O. Aldrian and W. A. P. Smith, "Inverse rendering of faces with a 3D Morphable Model," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 5, pp. 1080–1093, 2013.

[12] C. Lawson and R. Hanson, *Solving Least Squares Problems*. Society for Industrial and Applied Mathematics, 1995. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611971217

[13] A. Bas, W. A. P. Smith, T. Bolkart, and S. Wuhrer, "Fitting a 3D morphable model to edges: A comparison between hard and soft correspondences," *CoRR*, vol. abs/1602.01125, 2016. [Online]. Available: http://arxiv.org/abs/1602.01125

[14] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319–2323, 2000.

[15] J. R. T. Rodríguez, "3D face modelling for 2D+3D face recognition," Ph.D. dissertation, University of Surrey, 2007.

[16] R. T. A. van Rootseler, L. J. Spreeuwers, and R. N. J. Veldhuis, "Using 3D Morphable Models for face recognition in video," in *Proceedings of the 33rd WIC Symposium on Information Theory in the Benelux*, 2012.

[17] C. Sagonas, E. Antonakos, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: Database and results," *Image and Vision Computing*, pp. –, 2016. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0262885616000147

[18] X. Xiong and F. De la Torre, "Supervised descent method and its applications to face alignment," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2013, pp. 532–539.

[19] V. Kazemi and J. Sullivan, "One millisecond face alignment with an ensemble of regression trees," in *2014 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014, Columbus, OH, USA, June 23-28, 2014*. IEEE, 2014, pp. 1867–1874. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2014.241