Viewpoint-consistent 3D Face Alignment

Sergey Tulyakov, László A. Jeni, Jeffrey F. Cohn, and Nicu Sebe

Abstract—Most approaches to face alignment treat the face as a 2D object, which fails to represent depth variation and is vulnerable to loss of shape consistency when the face rotates along a 3D axis. Because faces commonly rotate three dimensionally, 2D approaches are vulnerable to significant error. 3D morphable models, employed as a second step in 2D+3D approaches are robust to face rotation but are computationally too expensive for many applications, yet their ability to maintain viewpoint consistency is unknown. We present an alternative approach that estimates 3D face landmarks in a single face image. The method uses a regression forest-based algorithm that adds a third dimension to the common cascade pipeline. 3D face landmarks are estimated directly, which avoids fitting a 3D morphable model.The proposed method achieves viewpoint consistency in a computationally efficient manner that is robust to 3D face rotation. To train and test our approach, we introduce the Multi-PIE Viewpoint Consistent database. In empirical tests, the proposed method achieved simple yet effective head pose estimation and viewpoint consistency on multiple measures relative to alternative approaches.

Index Terms—Face alignment, 3D face shape, morphable model

1 Introduction

OVER the last several years 2D face alignment has reached maturity making it possible to detect landmarks in the wild at very high frame rates [12], [27], [32], [44], [48]. These works can be grouped into three main categories [62]: Constrained Local Models (CLM) [4], [15], [48], [49], [73], Active Appearance Models (AAM) [14], [17], [19], [25], and more recent Cascaded Regression Methods (CRM) [27], [31], [55], [67], [78]. Up-to-date the notable achievements in the area are closely related to CRM. These methods are advantageous over the AAM and CLM based approaches in several respects: (i) running a sequence of regressors is faster than solving an optimization problem for every image, (ii) the offline training stage allows cascaded approaches to take advantage of the large available sets of training images, (iii) shape-invariant feature sampling makes these methods robust to rotations.

A typical face alignment task treats the face as a 2D object and is stated as follows: given an input image and the initial face location, determine the location of the main keypoints or landmarks of the face. Since 3D information about the face shape is lost, there are two main limitations of this formulation. Firstly, some of the landmarks can be hidden due to self-occlusions of the face. In the literature, this issue is tackled by detecting the *nearest visible landmarks*. This changes the natural face shape, since the landmarks no longer have semantic correspondence. The second limitation of the standard face alignment formulation arises in a multi-camera environment or in a face video (such as in Figure 1a), where several views/shots are available for the same scene. Ideally, the face keypoints detected for the same person using different cameras or for different frames in a video, must be consistent, since the underlying shape is the same. Classical approaches, however, fail to provide the desired consistency since they operate independently across frames/viewpoints. The latter limitation is due to the absence of publicly available 2D face databases annotated in a viewpoint consistent 3D fashion.

To some extent, these limitations have been tackled in the literature by 2D-3D works [12], [27]. These methods first detect 2D landmarks and as a second step fit a previously learned high resolution 3D face model to estimate a face shape. The resolution of the final 3D shape is comparable to depth-based methods [38], [56], [63], [64]. However, in many applications this high precision face shape estimation is not always required, while frame rates and low hardware requirements often become more critical [23], [57], [70]. Since these methods estimate a 3D shape of the face they naturally tackle the self occlusion problem at the price of a computationally expensive second step. Although 2D-3D methods estimate a person specific shape and keypoints for a face in the image, to the best of our knowledge, no study has been performed to evaluate their ability to preserve viewpoint consistency.

To overcome the first limitation and remove the redundant second step of 2D-3D works, we discuss a cascaded regressor-based method to estimate a set of 3D keypoints of a face from a single 2D image in a single step. Motivated by the recent success of sequential approaches for 2D face alignment, we discuss the framework that naturally detects 3D landmarks positions from a single image at state-of-the-art accuracy and processing speed. By starting with a set of mean 3D face keypoints, our method produces a sequence of 3D increments that move each landmark towards its desired location. In contrast with existing 2D-3D works, the method is capable of performing 3D face alignment in a single step. We perform standard evaluation of our method on a large corpus of 2D images rendered from the BU-4DFE [71]. Additionally, we provide a simple head pose estimation method, based on the predicted 3D shape, that outperforms available state-of-the-art systems.

The second limitation of the classical face alignment task is removed by introducing a new formulation of face alignment, i.e. by performing face alignment in a viewpoint-consistent manner. Given that every view of the face captures the same underlying shape, viewpoint-consistent face landmarks (Figure 1a) represent the same 3D structure with respect to the common coordinate

- *S. Tulyakov is with Snapchat Research.*
- *L. A. Jeni and J. F. Cohn are with the Robotics Institute at the Carnegie Mellon University and with the Affect Analysis Group at the University of Pittsburgh*
- *N. Sebe is with the Department of Information Engineering and Computer Science at the University of Trento, Italy.*

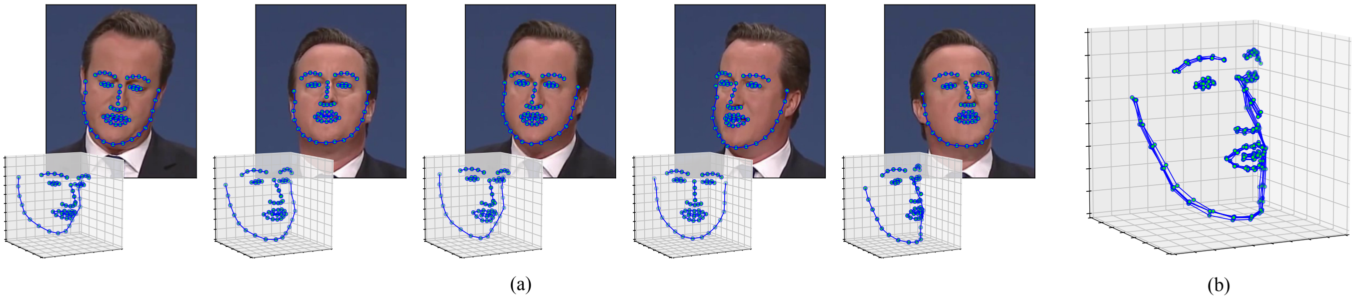

Fig. 1: Viewpoint-consistent 3D landmarks (a) represent the same underlying 3D structure for every frame (b)

system (Figure 1b). The lack of viewpoint-consistent alignment works in the literature can be explained by the absence of publicly available face datasets with viewpoint-consistent 3D annotation. To bridge this gap, we introduce the MultiPIE-VC (viewpoint-consistent) dataset that contains 3D annotations for 2D MultiPIE images. To evaluate viewpoint consistency of our method in comparison with the state-of-the-art 2D-3D methods we introduce several metrics, each highlighting different aspect of viewpoint consistency.

For the rest of the article vectors ($\mathbf{a}$) and matrices ($\mathbf{A}$) are denoted by bold letters. The Euclidean norm of vector $\mathbf{u} \in \mathbb{R}^d$ is $\|\mathbf{u}\|_2 = \sqrt{\sum_{i=1}^{d} u_i^2}$. The concatenation of matrices $\mathbf{A}_k \in \mathbb{R}^{d_k \times N}$ is denoted by $\mathbf{B} = [\mathbf{A}_1; \ldots; \mathbf{A}_K] \in \mathbb{R}^{(d_1+\ldots+d_K)\times N}$. Capital script letters ($\mathcal{A}$) denote sets.

2 Related work

Since there is no prior work on viewpoint-consistent face alignment, in the following we review the up-to-date standard face alignment approaches. We also compare the face databases most widely adopted by the community to provide the context for our MultiPIE-VC database.

2.1 Face alignment

Automatic face alignment is a highly researched area in computer vision. Since the first works on Active Shape Models (ASM) by Cootes et al. [13], [15] decades ago, face alignment has seen a significant improvement with a tremendous boost made recently. According to [62] face alignment methods can be categorized into three broad categories: Constrained Local Models (CLM), Active Appearance Models (AAM) and Cascaded Regression Methods (CRM). However, for the purpose of the current study, we use an alternative classification. We distinguish three groups of methods based on their final outcome. The first group contains 2D methods that given a face image produce 2D face landmarks. Methods of the second group provide a 3D face shape estimate, by first finding 2D pixel locations using a method from the first group. We, therefore, refer to these methods as 2D-3D. Methods of the third group provide 3D face keypoints for a face image directly avoiding the redundant second step.

2D methods. This avenue of research has started with ASM [13], [15] which belongs to the group of CLM methods, according to the classical classification. The basic idea is to represent an arbitrary 2D shape with a set of local appearance models. Then a global model is used to make a final decision about the shape. AAM methods [14], [19], [40] learn a global model of the shape and grayscale appearance jointly. Having this model, to perform face alignment one has to find a set of possible parameters that could have generated the query face. Despite their long history CLM and AAM are still highly used methods for face alignment in the literature (CLM [4], [48], [49], [73] and AAM [17], [25], [59]).

Modern face alignment is driven by CRM methods that initially originated in the medical image processing community [77]. In addition to dramatic performance gain [10], [58], [67], [68], [69], [78] these methods show impressive frame rates, being able to perform face alignment in one millisecond [32] or even less [44]. Importantly, CRM methods are able to naturally adopt large amounts of input data available on the Internet. In contrast, with optimization-based AAM and CLM methods, CRMs operate in a cascaded fashion. At every level of the cascade a regressor is learned to move the prediction closer to its desired location.

More recently, several CRM frameworks, based on training Deep Convolutional Neural Networks (CNN) have been introduced. In [74] a sequence of Deep Regression Networks is trained. To tackle the occlusion problem, a set of de-corrupt autoencoders is used. By combining the ability of a Recurrent Neural Network (RNN) to memorise previous events and using a CNN as a feature extraction function, the authors of [54] propose an end-to-end trainable method to localize facial landmarks. Although these works show state-of-the-art accuracy, they do not address viewpoint consistency of face shapes. Additionally, since these works are based on learning deep architectures, they require large training datasets and demand powerful hardware to achieve real-time frame rates. In the current study, we avoid using deep architectures to be able to maintain high frame rates and low power hardware requirements.

2D-3D methods. The output of the methods from the previous group is a set of 2D points in the image. Although, 2D-3D methods are typically based on 2D methods, their final goal is estimating a 3D face shape from a single image [6], [70], set of images [27], [46] or a video [27], [45]. Therefore, they can be referred to as two-step methods [55], performing 2D face alignment as the first step, and estimating a 3D shape as the second.

In this group, many works are done in the context of pose-invariant face recognition [6], [43], [52], [70]. The classical work of Blanz and Vetter [6] uses manual initialization as the first step, followed by fitting a 3D model. They achieve very accurate face models at a cost of low processing speed. In [70] a low resolution model is fit to 2D landmarks for determining feature sampling points. Although the final model is far from being perfect, the authors report improvement in face recognition results, while getting reasonable processing speed. The problem of estimating a face shape is tackled from a different perspective in [23]. This work presents a SIFTFlow-based [39] method to warp a depth-RGB image pair of a reference person to a single RGB image of a query person. Consequently, the method is rather slow and can estimate depth only for visible parts of the face. Again, the first step is performed by the 2D face alignment system of [79].

Cao et al. [9] tackles the problem of automated avatar animation. They propose to jointly estimate a parametric 3D face model together with 2D landmarks from a video of a human performer. The method uses the landmarks estimated for the previous frame to simultaneously regress the 3D and 2D shapes for the current frame. However, when applied to a single image, the previously estimated landmarks are not available, and the method requires 2D landmarks estimated using [10] as an initialization, which makes it a two-step method according to our classification. Their approach requires just a video stream to operate, reaching results close to modern methods assuming RGB-D as the input [24], [38]. An interesting application of face tracking and reconstruction technology is presented by Thies et al. in [53]. They present a real-time performance capture system for face reenactment using RGB videos only. The system features photo-realistic quality. Similarly to other methods of this group their pipeline is two step.

The major difference of this work with respect to the previously described two-step works is that our method is single-step and requires only a single image. A key advantage of our single-step method is that it is faster, since it does not require computationally expensive 3D model fitting, while is able to accurately estimate the third dimension of the interest points.

3D methods. This is a recently emerged group of methods with only several works available in the literature [31]. The idea is to avoid the sometimes unnecessary second step of the methods from the previous group. Clearly, the absence of 3D annotated 2D images is a critical problem for such methods. In [80] a 3D morphable model is used to annotate the available 2D datasets. These annotations are then used to train several CNNs in a cascaded manner. There is, however, a major problem with these annotations, since existing methods for single image face reconstruction on monocular cameras tend to suffer from only "hallucinating" a 3D shape from a 2D image. A single 2D image does not directly provide enough information for the reconstruction, i.e. 3D shapes can look the same when projected to 2D, while being different in the third dimension. As discussed in Section 2.2 only few databases contain multiple views for every subject. The method we discuss in this study belongs to this group.

Viewpoint-consistent 3D face alignment. The goal of this paper is to propose to the community a new research direction in face alignment. Consider a subject captured from several viewpoints (see Figure 2), or a video of a face where the viewpoint changes constantly (see Figure 1). Clearly, the underlying face shape remains the same, and therefore, it should be consistent between frames/viewpoints. However, 2D methods do not show this consistency, since they are only able to detect the visible points. Methods from 2D-3D group offer some form of viewpoint consistency at a price of employing the computationally expensive second step. Moreover, during the first step face alignment is performed in a classical inconsistent manner.

Methods from the 3D group, are the closest to our idea. However, there is no study available investigating if their outputs are consistent between the viewports. Importantly, to the best of our knowledge there exists no benchmark database containing 3D annotations for a 2D face database.

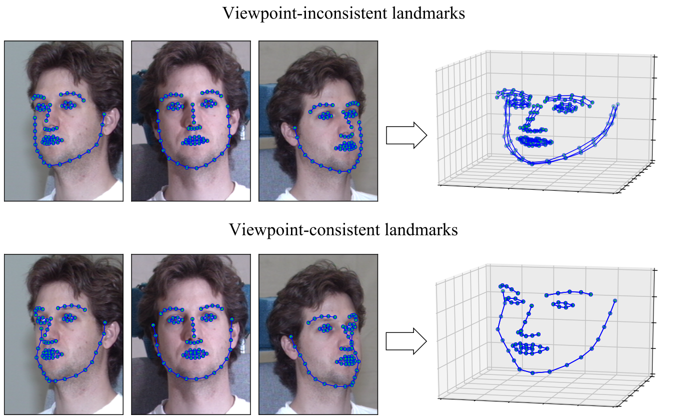
Viewpoint-inconsistent landmarks
Viewpoint-consistent landmarks

Fig. 2: Top: conventional viewpoint-inconsistent landmarks. Bottom: 3D landmarks annotated in a viewpoint-consistent manner. Right: landmarks from different viewpoints plotted in the common coordinate system. The third dimension of the inconsistent landmarks was estimated by fitting a 3D deformable model to the estimated 2D landmarks (Section 4). Bringing the landmarks to the common coordinate system is discussed in Section 5.3.

2.2 Face databases

Over the last decade a large number of face databases were made available to the community. Many of these were created for different purposes in the face analysis domain. First databases were created for face identification and verification (BioID [30], Texas3D [22]). Another group of databases was aimed at conducting facial expression recognition research (MultiPIE [20], BU-3DFE [72], BU-4DFE [71]) or Action Units recognition (BP4D-Spontaneous [75]). The third group was specifically created for face alignment tasks (AFLW [34], Helen [37], COFW [8]). Regardless of their initial purpose, all these databases contain annotation of a specific number of facial landmarks, making it possible to use them in face alignment research.

Table 1 shows a comparison of the databases used for face alignment research. Initially, the first face databases were created in controlled laboratory environment and recently the focus shifted to more unconstrained conditions. This is due to the saturation of standard face alignment performance on laboratory-controlled databases. The databases vary in the number of subjects, the total number of images, the number of available landmarks and the presence/absence of several shots/views for each subject, which is the most important aspect for our study.

Specifically, for viewpoint-consistent 3D face alignment the availability of several views for the same subject and the presence of viewpoint-consistent 3D landmarks are required. None of the outlined database in Table 1 possesses both requirements. A slight exception can be made for databases containing 3D meshes of faces (such as XM2VTS [41], BU-3DFE [72], BU-4DFE [71] and BP4D-Spontaneous [75]), since these meshes can be used to generate multiple random views of the subjects, allowing the algorithms to be trained and tested in a viewpoint-consistent manner [27], [55]. This, however, has some limitation, as the rendered meshes are no longer captured by a real camera, and the quality of the generated images depends on the models of the renderer. HPDB [2] is a recent database for head pose estimation and face alignment. It contains 120 face videos of 12 subjects (6 male and 6 female). Despite the database includes 3D points, its

TABLE 1: Comparison of available databases that were used for face alignment in the literature.

Database	Year	Modality	Images/Videos	Conditions	# subjects	# views	# images	# landmarks
XM2VTS [41]	1999	RGB + Depth	images	controlled	295	6	2360	68
BioID [30]	2001	RGB + Depth	images	in the wild	23	1	1521	20
BU-DepthFE [72]	2006	RGB + Depth	images	controlled	100	Depth[1]	2500	83
LFW [26]	2007	RGB	images	in the wild	5749	≥1[2]	13233	11
Multi-PIE [20]	2008	RGB	images	controlled	337	15	750K	68
BU-4DFE [71]	2008	RGB + Depth	videos	controlled	101	Depth[1]	60K	68[3]
Bosphorus [50]	2008	RGB + Depth	images	controlled	105	13	4666	24
Texas Depth [22]	2010	RGB + Depth	images	controlled	118	1	1149	25
MUCT [42]	2010	RGB	images	controlled	276	1	3755	76
AFLW [34]	2011	RGB	images	in the wild	n/a	1	25K	21
LFPW [5]	2011	RGB	images	in the wild	n/a	1	1400	39
AFW [79]	2012	RGB	images	in the wild	n/a	1	205	6
Helen [37]	2012	RGB	images	in the wild	n/a	1	2300	194
300-W [47]	2013	RGB	images	in the wild	n/a	1	6265[4]	68
COFW [8]	2013	RGB	images	in the wild	n/a	1	1007	29[5]
FaceWarehouse [11]	2014	RGB + Depth	images	controlled	150	20	3000	74
BP4D-Spontaneous [75]	2014	RGB + Depth	videos	controlled	31	Depth[1]	300K	84
HPDB [2]	2016	RGB	videos	controlled	12	1	36K	54

[1] For depth databases multiple views can be generated artificially
[2] For some subjects several views/shots are provided
[3] 68 landmarks for selected frames are provided in [55]
[4] Combines AFW, LFPW, Helen, XM2VTS
[5] Additionally for each landmark visible/occluded labels are given

use for viewpoint-consistent alignment is limited, as it provides only 12 subjects captured under a constrained laboratory setting. Moreover, the dataset employs an uncommon 54 points markup, making it impossible to use it for testing purposes, as the markup is not compatible with other face datasets.

Multi-view databases, such as MultiPIE [20] and multi-image databases (XM2VTS [41], LFW [26]) contain multiple images from different perspectives of the same subjects, which afford an attractive corpora for viewpoint-consistent alignment. However, these databases lack viewpoint-consistent landmarks and annotations of the third dimension, making it impossible to develop viewpoint-consistent methods using these databases.

To bridge this gap we introduce MultiPIE Viewpoint Consistent (MultiPIE-VC), a subset of the commonly adopted MultiPIE database, with the viewpoint-consistent 3D annotations of 66 facial landmarks. Up to the authors knowledge, this is the first benchmark suitable for viewpoint-consistent 3D face alignment.

3 Method

There are several major differences as compared to the standard cascaded regression works: (i) our shape estimates are 3D, (ii) we propose and compare three methods for 3D feature indexing, and (iii) we show a simple yet efficient head pose estimation method showing better or highly competitive scores on various benchmarks as we show in Section 5.2.

3.1 A framework of cascade regressors

A general cascade regression approach produces an estimate $\hat{\mathbf{S}}$ of facial landmarks $\mathbf{S}$ for an image of a face $\mathbf{I}$ by producing several increments $\Delta\mathbf{S}_t$ $(t = 1, ..., N)$ at every level t of the cascade in the following fashion:

$$\Delta\mathbf{S}_t = r_t(H_t(\mathbf{I}, \hat{\mathbf{S}}_{t-1})), \quad (1)$$

$$\hat{\mathbf{S}}_t = \hat{\mathbf{S}}_{t-1} + \Delta\mathbf{S}_t, \quad (2)$$

where H_t is a feature extraction function, r_t is a regressor function learned at the t^{th}-level of the cascade and N is the total number of levels in the cascade. A shape vector $\mathbf{S} = [\mathbf{x}_1; \mathbf{x}_2; ..., \mathbf{x}_n]$ represents a set of facial landmarks. We denote $\hat{\mathbf{S}} = \mathbf{r}(\mathbf{I}, \bar{\mathbf{S}})$ as the final estimate made by the cascade of regressors $\mathbf{r}(\cdot, \cdot)$ for an image $\mathbf{I}$ and the initial guess $\bar{\mathbf{S}}$.

In previous works, every point $\mathbf{x}_i$ of the face shape vector was represented either by x, y-coordinates in the image, or was augmented by an additional label m_i that represents a flag indicating whether a point is visible or occluded: $\mathbf{x}_i = [x_i; y_i; m_i$. Hereinafter we drop the index i and write $\mathbf{x}$ to denote a point of a shape to simplify the notation. Instead of adding an extra flag for every point we augment the usual x, y-coordinates of a point in the plane with the z-coordinate of the landmarks in the 3D space. Having a third dimension in the training set at every step of the cascade we learn the 3D shape increment $\Delta\mathbf{S}_t \in \mathbb{R}^{n\times 3}$.

The feature extraction function $H_t(\mathbf{I}, \hat{\mathbf{S}}_{t-1})$ in Eq. 1 depends not only on image $\mathbf{I}$ but also on the previous shape estimate $\hat{\mathbf{S}}_{t-1}$, this allows the cascade to extract shape independent features. We propose to extend a face shape with the third dimension so that $\Delta\,\mathbf{S}_t,\ \hat{\mathbf{S}}_t,\ \mathbf{S}\ \in\ \mathbb{R}^{n\times 3}$. Several models can be used as a regressor; we train a number of regression trees at each level of the cascade, since they have shown remarkable results in the literature [32], [44].

3.2 From world coordinates to 3D landmarks

To learn a face landmarks predictor one has to decide upon the landmarking scheme and perform annotation of the available training data. In our case, such annotation is hardly possible even for a human annotator due to the difficulty to estimate a z-coordinate by observing just a single 2D image. However, we propose the solution based on performing the 2D annotation as usual, and then augmenting the annotation of the z-coordinate estimated in a different way. To do so we use the available 2D+3D database BU-4DFE [71]. Manual annotation is performed on a frontal set of images provided in the database. Since 2D-3D correspondences are known, we map 2D coordinates of the point in a frontal RGB image to the corresponding 3D point on the mesh.

To generate various head poses for training and testing purposes we render meshes under pitch and yaw rotations uniformly

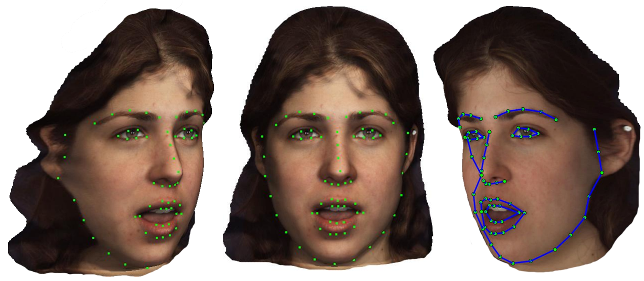

Fig. 3: An example of the actual landmark positions. Left image shows an annotated mesh with several landmarks occluded. Central image shows the landmarks on the frontal face. Right image shows the projections of the actual landmarks onto the image plane.

distributed in the range of $[-50, 50]$ degrees. Since the rendering parameters are known, we can get the locations of the points by using the pinhole camera model:

$$\lambda \mathbf{x}^c = \mathbf{A}\mathbf{R}\mathbf{x}^w + \mathbf{t}, \tag{3}$$

where $\mathbf{x}^w = [x^w; y^w; z^w]$ is the point in the world coordinate system, $\mathbf{x}^c = [x^c; y^c; 1]$ is the point in the camera coordinates, λ is the homogeneous scaling factor, $\mathbf{A}$ is the matrix of intrinsic parameters or the camera matrix, $\mathbf{R}$ and $\mathbf{t}$ are the rotation matrix and the translation vector correspondingly. We note here that the z-coordinate is still available after the transformation. We augment the point in the camera coordinates with this z-coordinate to form $\tilde{\mathbf{x}}^c = [x^c; y^c; \lambda]$. In this way every training example is formed by $\{\mathbf{I}(\text{yaw}, \text{pitch}), \tilde{\mathbf{S}}^c\}$, where $\tilde{\mathbf{S}}^c = [\tilde{\mathbf{x}}_1^c; \tilde{\mathbf{x}}_2^c; ...; \tilde{\mathbf{x}}_n^c]$. Although $\tilde{\mathbf{S}}^c$ is a 3D shape, its points are distorted by the camera matrix and therefore, proportions no longer correspond to the normal face proportions. This needs to be compensated. To this end we define a point $\mathbf{x}^{wR} = \mathbf{R}\mathbf{x}^w$ and a corresponding shape $\mathbf{S}^{wR}$ which is rotated according to the extrinsic rotation matrix, while being represented in the world coordinates. During testing a cascade of regressors produces a shape estimate $\hat{\mathbf{S}}^c = \mathbf{r}(\mathbf{I}, \tilde{\mathbf{S}}^c)$, where the shape $\hat{\mathbf{S}}^c$ is given by augmented points: $\hat{\mathbf{S}}^c = [\hat{\mathbf{x}}_1^c; \hat{\mathbf{x}}_2^c; ...; \hat{\mathbf{x}}_n^c]$. Then, if the camera matrix $\mathbf{A}$ is known, we can rewrite Eq. 3 to get $\hat{\mathbf{x}}^{wR}$:

$$\hat{\mathbf{x}}^{wR} = \mathbf{A}^{-1}(\lambda \hat{\mathbf{x}}^c - \mathbf{t}). \tag{4}$$

However, at testing time the matrix $\mathbf{A}$ is unknown, and therefore needs to be estimated. To get the estimate $\hat{\mathbf{A}}$ we perform camera calibration using $\bar{\mathbf{S}} \in \mathbb{R}^{n \times 3}$ as the coordinates in the world coordinate system and only x, y-values of the points in $\hat{\mathbf{S}}^c$ as the coordinates in the image plane. Finally, we substitute $\hat{\mathbf{A}}$ into Eq. 4 to get $\hat{\mathbf{x}}^{wR}$.

We train and test our model on the actual landmarks positions even if they are invisible because of face rotations. Figure 3 shows an example of this. In other works, the closest visible pixels to the invisible landmarks are usually used instead. For example, the boundary of the face is often considered as a jawline when the actual jawline is not visible. However, this operation changes the natural proportions of the estimated shape, which is acceptable for two-step systems, where the shape is regularized during the second step.

Our experiments show that it is possible to estimate the actual 3D positions of the invisible points. Moreover, since a recovered shape is unchanged and is represented in the world coordinates we can accurately determine the head pose (see Sections 3.4 and 5.2).

3.3 3D invariant features

At every level of the cascade, we build tree-based regressors to produce a shape increment. The decision function of a tree uses simple intensity difference features extracted at the points $\mathbf{u}$ and $\mathbf{v}$ indexed with respect to a mean shape. The points $\mathbf{u}$ and $\mathbf{v}$ are randomly generated during training. The goal of feature indexing is to have a way to compute $\mathbf{u}$ and $\mathbf{v}$ for every face geometrically close to their true locations, taking into account scaling, rotation and translation.

Indexing starts by defining an offset from $\mathbf{u}$ to the nearest point $\mathbf{x}_{k_\mathbf{u}}$ in the mean shape (we follow the notation in [32]):

$$\delta \mathbf{x}_\mathbf{u} = \mathbf{u} - \bar{\mathbf{x}}_{k_\mathbf{u}}, \tag{5}$$

where $\delta \mathbf{x}_\mathbf{u}$ is selected during training. To determine $\mathbf{u}'$, a point geometrically corresponding to the point $\mathbf{u}$, we first find the scaling and rotation transformations between the mean shape $\bar{\mathbf{S}}$ and the current shape estimate $\hat{\mathbf{S}}_t$:

$$\{s, \mathbf{R}, \mathbf{t}\} = \operatorname*{argmin}_{s, \mathbf{R}, \mathbf{t}} \sum_{i=1}^{n} \|\bar{\mathbf{x}}_i - (s\mathbf{R}\mathbf{x}_i + \mathbf{t})\|^2, \tag{6}$$

where $s, \mathbf{R}, \mathbf{t}$ represent scaling, rotation and translation correspondingly. Then $\mathbf{u}'$ is determined in the following way:

$$\mathbf{u}' = \mathbf{x}_{k_\mathbf{u}} + \frac{1}{s}\mathbf{R}^T \delta \mathbf{x}_\mathbf{u}. \tag{7}$$

If one considers the case when $\mathbf{S} \in \mathbb{R}^{2 \times n}$, then the rotation matrix $\mathbf{R} \in \mathbb{R}^{2 \times 2}$, which accounts for in-plane rotations, such as roll angle.

To address head rotation from a 3D perspective, for the current cascade level t we define a face basis $\mathbf{F}_t$. The basis is spanned by the normal $\vec{\mathbf{n}}_t$, the vector connecting the centers of the eyes $\vec{\mathbf{e}}_{1,t}$ and $\vec{\mathbf{e}}_{2,t} = \vec{\mathbf{n}}_t \times \vec{\mathbf{e}}_{1,t}$, where $\times$ is a cross product operation. The vector $\vec{\mathbf{n}}_t$ is determined as the eigenvector with the smallest eigenvalue of the following covariance matrix:

$$\mathbf{C}_t = \frac{1}{n} \sum_{i=1}^{n} (\mathbf{x}_{i,t-1} - \bar{\mathbf{x}}_{t-1})(\mathbf{x}_{i,t-1} - \bar{\mathbf{x}}_{t-1})^T, \tag{8}$$

where $\mathbf{x}_{i,t-1} \in \hat{\mathbf{S}}_{t-1}$, $\mathbf{C}_t \in \mathbb{R}^{3 \times 3}$. Since the direction of the normal vector $\vec{\mathbf{n}}_t$ can vary from iteration to iteration, depending on the face rotation, to obtain the normal consistently oriented with the observer direction $\vec{\mathbf{n}}_o$, we need to satisfy the following equation:

$$\vec{\mathbf{n}}_t \cdot \vec{\mathbf{n}}_o > 0, \tag{9}$$

where $\cdot$ is a dot product operation. We assume that $\vec{\mathbf{n}}_o$ is perpendicular to the image plane and directed to the observer. Having the basis $\mathbf{F}_t$ and the estimated scaling s we rewrite Eq. 7 in the following way:

$$\tilde{\mathbf{u}}' = \mathbf{x}_{k_\mathbf{u}} + \frac{1}{s}\mathbf{F}_t^T \delta \tilde{\mathbf{x}}_\mathbf{u}, \tag{10}$$

where $\delta \tilde{\mathbf{x}}_\mathbf{u} = [\delta \mathbf{x}_\mathbf{u}; 0]$, such that $\delta \tilde{\mathbf{x}}_\mathbf{u}, \tilde{\mathbf{u}}' \in \mathbb{R}^3$. After the transformation, the third dimension is truncated. In this way we find the coordinates of the offset vector in the face basis $\mathbf{F}_t$. Now we define three ways of indexing features:

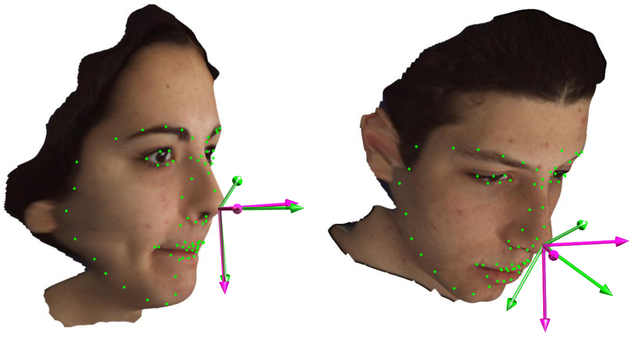

Fig. 4: Examples of face bases estimated using $\hat{\mathbf{S}}^{wR}$ (green) and $\hat{\mathbf{S}}^{c}$ (pink). Note that the bases estimated using shapes in the world coordinate system (green) are more consistent with the head rotation. The detected points are plotted in green. The background was removed for visualization purposes after detection.

- **Baseline** indexing is based on directly using Eq. 7. In this case the only difference with the original method [32] is that the learned shape is 3-dimensional.
- **3D transform** indexing. The difference with the baseline method is that minimization in Eq. 6 is performed in a 3D space, resulting in rotation matrix $\mathbf{R} \in \mathbb{R}^{3\times3}$.
- **Basis transform** indexing determines pixel sampling points by first estimating a basis $\mathbf{F}_t$ and then computing $\tilde{\mathbf{u}}'$ with Eq. 10.

The same analysis can be applied to get $\mathbf{v}'$. We report the comparison results of these methods in the experimental section.

3.4 Head pose analysis

All the components in our analysis are 3D. This is the advantage of our single-step approach that allows us to use a simple yet reliable method to estimate the head pose of a face. In the previous section we defined a face basis $\mathbf{F}_t$ that is associated with the direction of the face. Clearly, the directions of the basis vectors of $\mathbf{F}_t$ can reveal the head pose of the analyzed face. We exploit this fact to determine the head direction.

The final estimate $\hat{\mathbf{S}}^c = \mathbf{r}(\mathbf{I}, \bar{\mathbf{S}}^c)$ is represented in the camera coordinates. Although it is three-dimensional, its proportions no longer correspond to the actual face proportions, and therefore the estimated basis will not accurately correspond to the face direction or the head pose. To address this we apply the analysis detailed in Section 3.2. By using Eq. 4 and estimating the camera matrix $\mathbf{A}$ we transform every point of $\hat{\mathbf{S}}^c$ to the world coordinate system and obtain $\hat{\mathbf{S}}^{wR}$, for which the face proportions are preserved. We then analyze the angles of the basis vectors to estimate the head pose. This simple method offer highly competitive head pose estimation accuracy as shown in Section 5.2. Examples of bases estimated using $\hat{\mathbf{S}}^c$ and $\hat{\mathbf{S}}^{wR}$ are given in Figure 4.

3.5 Learning

Our learning framework is similar to one presented in Kazemi et al. [32]. We train N levels of the cascade, where each level contains K regression trees. A node split is performed with the following split function:

$$h(I, \hat{S}_t, \theta) = \begin{cases} 1 & \mathbf{I}(\mathbf{u}') - \mathbf{I}(\mathbf{v}') > \tau \\ 0 & \text{otherwise,} \end{cases} \tag{11}$$

where $\theta = (\mathbf{u}', \mathbf{v}', \tau)$, $\mathbf{u}'$ are $\mathbf{v}'$ are obtained by using Eq. 7 or 10 depending on the indexing strategy. The split parameters in θ are randomly generated at each split node and a tree is trained with a gradient boosting algorithm that minimizes the sum of squared error.

3.6 Running time analysis

At every stage of the cascade $t = 1, ..., N$ we need to propagate the trees $O(KF)$ and compute a face basis $O(n^2p+p^3)$, where K is the number of weak regressors, F is the number of trees, n - the number of landmarks and p - the dimensionality of each landmark. Therefore, for a single image the running time complexity of our algorithm is constant $O(N(KF + n^2p + p^3))$. For a case $\{N = 10, K = 500, F = 5, n = 68, p = 3\}$ the method takes on average 9 ms to process an image on an Intel Core i7-4702HQ processor

4 3D DEFORMABLE MODEL FITTING

In order to compare the proposed single-step 3D method with the available 2D approaches, we augment the standard 2D methods with the 3D deformable model fitting step detailed in this section. We follow [27] and define the shape model using a 3D mesh. Consider the 3D shape as the coordinates of N 3D vertices that make up the mesh:

$$\mathbf{S} = [\mathbf{x}_1; \ldots; \mathbf{x}_N], \mathbf{x}_i = [x_i; y_i; z_i]. \tag{12}$$

The tracker uses a 3D deformable model describing non-rigid shape variations linearly and composes it with a global rigid transformation, placing the shape in the image frame:

$$\mathbf{x}_i = \mathbf{x}_i(\mathcal{P}, \mathbf{q}) = s\mathbf{R}(\bar{\mathbf{x}}_i + \mathbf{\Phi}_i\mathbf{q}) + \mathbf{t} \quad (i = 1, \ldots, N), \tag{13}$$

where $\mathbf{x}_i(\mathcal{P}, \mathbf{q})$ denotes the 3D location of the i^{th} landmark and $\mathcal{P} = \{s, \alpha, \beta, \gamma, \mathbf{t}\}$ denotes the rigid parameters of the model, which consist of a global scaling s, the angles of rotation in three dimensions $\mathbf{R} = \mathbf{R}_1(\alpha)\mathbf{R}_2(\beta)\mathbf{R}_3(\gamma)$, and the translation $\mathbf{t}$. The non-rigid transformation is denoted with $\mathbf{q}$. Here $\bar{\mathbf{x}}_i$ denotes the mean location of the i^{th} landmark (i.e. $\bar{\mathbf{x}}_i = [\bar{x}_i; \bar{y}_i; \bar{z}_i]$ and $\bar{\mathbf{x}} = [\bar{\mathbf{x}}_1; \ldots; \bar{\mathbf{x}}_N]$). The d pieces of $3N$ dimensional basis vectors are denoted with $\mathbf{\Phi} = [\mathbf{\Phi}_1; \ldots; \mathbf{\Phi}_N] \in \mathbb{R}^{3N\times d}$. Vector $\mathbf{q}$ represents the 3D distortion of the face in the $3N \times d$ dimensional linear subspace.

To reconstruct the 3D shape of the face, we first estimate a set of 2D landmarks (2D shape, $\mathbf{z}$) and then minimize the reconstruction error using Eq. 13:

$$\underset{\mathcal{P},\mathbf{q}}{\operatorname{argmin}} \sum_{i=1}^{N} \|\mathbf{P}\mathbf{x}_i(\mathcal{P}, \mathbf{q}) - \mathbf{z}_i\|_2^2 \tag{14}$$

Here $\mathbf{P}$ denotes the projection matrix to 2D, and $\mathbf{z}$ is the target 2D shape. An alternating, iterative least squares method is used to register the 3D model on the 2D landmarks. The algorithm iteratively refines the 3D shape and 3D pose until convergence, and estimates the rigid ($\mathcal{P} = \{s, \alpha, \beta, \gamma, \mathbf{t}\}$) and non-rigid transformations ($\mathbf{q}$).

Applying Eq. 14 on a single image frame has the drawback of simply ”hallucinating” a 3D representation from 2D. From a single viewpoint there are multiple solutions that satisfy Eq. 14. To avoid the problem of single frame 2D-3D hallucination, one can apply the method simultaneously across multiple image-frames.

Assuming that we have access to time-synchronized 2D measurements $\mathbf{z}^{(1)}, \ldots, \mathbf{z}^{(C)}$ from a multi-view setup consisting of C cameras. The exact camera locations and camera calibration matrices are unknown. In this case all C measurements represent the same 3D face, but from a different point of view. We can extend Eq. 14 to this scenario by constraining the reconstruction to all the measurements:

$$\underset{\substack{\mathcal{P}^{(1)},\ldots,\mathcal{P}^{(C)},\\ \mathbf{q}}}{\operatorname{argmin}} \sum_{k=1}^{C}\sum_{i=1}^{N} \left\| \mathbf{P}\mathbf{x}_i(\mathcal{P}^{(k)}, \mathbf{q}) - \mathbf{z}_i^{(k)} \right\|_2^2 \qquad (15)$$

where superscripts (k) denote the k^{th} measurement. Note that in this case the shape parameters $\mathbf{q}$ are consistent for all measurements, since we are observing the same face, but from different views.

5 Experiments

We perform quantitative evaluation using two experimental protocols. The first one is a standard face alignment evaluation scheme, i.e. comparison of the predictions with the ground truth locations of the landmarks. The second experimental protocol is aimed at quantifying viewpoint consistency of the compared methods. We report results on synthetic data rendered using the BU-4DFE dataset and on in-the-wild data.

5.1 On the difficulty of comparing on the standard benchmarks

Viewpoint-consistent 3D landmarks represent a set of 3D points that preserve semantic correspondence between viewports of the same subject. In contrast, inconsistent 2D landmarks contain only 2D locations of face points that can be marked by a human annotator. Due to this only the visible pixels can be annotated. This difference becomes particularity significant under non-frontal head poses. Two examples of this are given in Figure 5. We highlighted in yellow several incompatibilities of consistent and inconsistent landmarks. Note how they become larger as the head pose reaches side face views. Clearly, one cannot directly compare these two types of landmarks. Selecting a subset of landmarks that correspond to each other or limiting head pose ranges so that all face pixels are visible is suboptimal and does not provide the necessary basis for comparison. We therefore argue, that 2D inconsistent landmarks can be compared with the consistent landmarks by restoring the third dimension using 2D-3D methods (see Section 5.4). Comparison with the classical alignment methods can be performed by retraining them on viewpoint-consistent landmarks (see Sections 5.2 and 5.5).

5.2 Standard evaluation

We show that 3D information embedded into the regression pipeline is essential and provides improvement over purely 2D methods. Since most of the works for face alignment estimate only 2D landmarks from an RGB image and invisible landmarks are either skipped from the estimation or their nearest visible neighbors are predicted, direct comparison on publicly available benchmarks is not possible. To this end, in order to perform standard (non-viewpoint-consistent) evaluation we generate a large set of training and testing images and perform semi-automatic annotation of this set. For comparison purposes we train the method presented in [32] on x, y-coordinates of our 3D annotations, keeping their default parameters unchanged. We use the open-source implementation of [32] made available by [33].

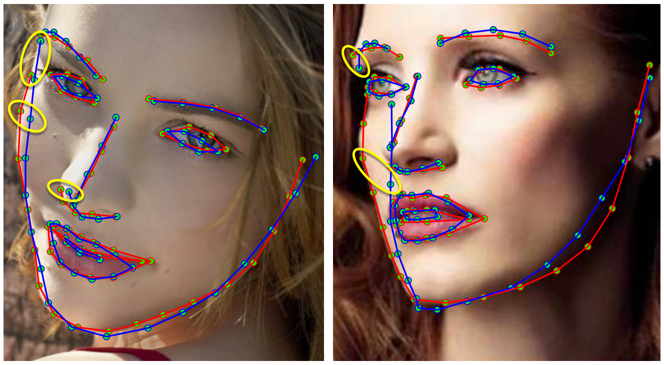

Fig. 5: Comparisons of viewpoint-consistent 3D landmarks (blue) and inconsistent 2D landmarks (red). In yellow we highlight several major differences located on the jaw-line, eyebrows, nose. The viewpoint-consistent landmarks were obtained using the method discussed in this paper. The inconsistent landmarks are taken from [47].

Data. For the standard evaluation we build our training and testing set by using the BU-4DFE [71] database. This database contains 2D and 3D videos for six posed prototypical facial expressions (anger, disgust, fear, happy, sad, surprise) for 101 ethnically diverse subjects (58 female and 43 male). The database contains more than 60K 2D-3D pairs. Since BU-4DFE does not contain facial landmarks annotations, we performed manual annotation. We followed the widely accepted MultiPIE [21] 68-landmarks scheme. The 60K samples of the database were uniformly sampled to obtain 3000 face images with the corresponding 3D meshes. Manual annotation was performed on these 2D images, and the annotations were augmented with the third coordinate by finding the reference points on the mesh. As a result, we have 3000 images of faces annotated with the 3D landmarks positions. To generate images of faces with various head poses we rendered the meshes under uniformly distributed face rotations taken from the range $[-50, 50]$ degrees for yaw and pitch angles. In total we have 120K images. To add variability to this generated set we used images from the SUN database [65] as backgrounds, removing images annotated as containing a person. The selected BU-4DFE recordings ids as well as the 3D annotations will be made available to the research community.

3D landmarks localization. To test the accuracy of our method we randomly split the rendered images into folds and perform 6-fold cross-validation. We report the averaged results for all the folds. We use the commonly accepted metric that measures the distance from a landmark to its ground truth position normalized by dividing it by the interocular distance for each image. Table 2 shows the results. We perform a separate comparison for 2D and 3D. For 2D only the first two coordinates (x, y) were used.

Table 2 shows that learning 3D face landmarks improves the accuracy even if we are only interested in 2D points in the image plane. Basis transform indexing shows a slightly better performance for 3D case than the other methods (not statistically significant). The intuition for this effect is that a face is inherently a 3D object, and therefore three-dimensional indexing is able to more reliably estimate the corresponding sampling points. The values in Table 2 are close to those reported in the literature for

Method	2D	3D
Kazemi et al. [32]	0.0522	-
Baseline indexing	**0.0515**	0.0610
3D Transform	**0.0515**	0.0607
Basis Transform	0.0518	**0.0592**

TABLE 2: Landmark localization errors. The numbers represent the average distance from an estimated landmark to its ground truth location normalized by the interocular distance.

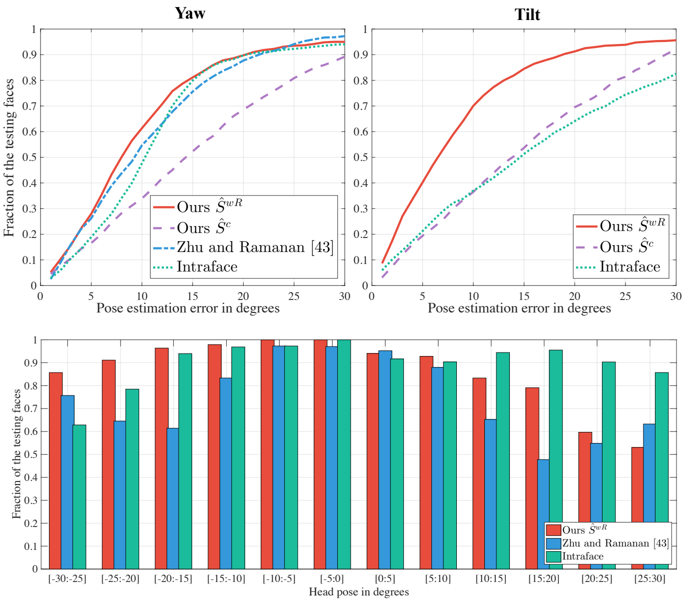

Fig. 6: Top: cumulative error distribution rates for head pose estimation for yaw and pitch angles. We do not report results of [79] for pitch since the method provides only yaw estimates. To initialize our method we used the face detector in [33]. The benchmark systems use their internal face detectors to find faces. Bottom: The distribution of the fraction of correctly recognized images within ±15° error tolerance over the yaw angle.

2D face alignment. This proves the difficulty of rendered testing set compared to the commonly used benchmarks such as [37].

Head pose estimation. We compare our method with the available state-of-the-art methods of Zhu and Ramanan [79] and Intraface[1] by Xiong and De la Torre [67]. For comparison, we use a subset of 1300 images of faces from our rendered set. The head pose is uniformly distributed within $[-50, 50]$ degrees. The images were taken from the testing folds of the trained models. The advantage of such generated dataset is that it uniformly covers all head poses in a range, and, more importantly, requires no manual annotation, because head poses are known exactly. If the competing systems were not able to detect the face in an image, we removed the image from the testing set. In total 1123 images were left. We report the results of our model that uses basis transform as the indexing method.

Table 3 shows the fraction of correctly classified images within the $\pm 15°$ error tolerance, which is the commonly accepted metric in head pose analysis literature (also used in [79]). The table shows that our method based on analyzing the normal vector to $\hat{\mathbf{S}}^{wR}$ scores the best. The method based on $\hat{\mathbf{S}}^{c}$ still shows reasonable performance for pitch, but these results prove that 3D information contained in $\hat{\mathbf{S}}^{c}$ is not sufficient for head pose estimation, while the analysis in Section 3.2 is a tool to restore the shape of the face. In Figure 6 we plot the dependency of the fraction of the correctly labeled testing faces on the error tolerance value. In addition we report the fraction of correctly classified images as a function of the yaw angle for error smaller than 15° of our best method versus [79] and Intraface [67].

Method	Yaw	Pitch
Ours $\hat{\mathbf{S}}^{c}$	0.52	0.54
Zhu and Ramanan [79]	0.76	-
Intraface	0.80	0.51
Ours $\hat{\mathbf{S}}^{wR}$	**0.81**	**0.85**

TABLE 3: Head pose estimation results. The numbers show the fraction of faces correctly labeled within ±15° error tolerance.

Method	Yaw	Pitch	Mean
An and Chung [1]	5.33	7.22	6.28
Valenti et al. [60]	6.10	5.26	5.68
Kumano et al. [35]	7.10	4.20	5.65
Sung et al. [51]	5.40	5.60	5.50
Vincente et al. [61]	4.30	6.20	5.25
Saragih et al. [49]	5.20	4.50	4.85
La Cascia et al. [36]	3.30	6.10	4.70
Asteriadis et al. [3]	4.56	3.82	4.19
This work	5.32	3.03	4.18
Xiao et al. [66]	3.80	3.20	3.50
Jeni et al. [28]	3.93	2.66	3.30

TABLE 4: Head pose estimation results obtained on the Boston University head tracking dataset, presented in the mean absolute angular error in degrees. The accuracy of the methods except ours are taken from [28]. The results are sorted by the average error on both angles in a descending order.

To further study the performance of our method on the head pose estimation task we report results obtained on two additional benchmark databases. The first is the Boston University head tracking database [36]. To compare with the previous works we used 45 videos of diverse subjects captured under uniform lighting. The videos contain various head movements and facial expressions. The ground truth was captures by the Flock of Birds tracker fixed on the subject's head in every sequence. The second benchmark is the recently introduced HBPD database [2]. It includes 120 videos of 10 subjects (12 videos per subjects) annotated with the head pose orientation.

Table 4 presents the evaluation results in comparison with other methods available in the literature. Given 3D landmarks estimated by our method, a simple head pose estimation method, based on understanding the direction of the face scores among the best methods. Table 5 shows the results obtained on the HBPD database, on which our method shows the second best performance.

5.3 Validating viewpoint consistency

A commonly accepted metric for analyzing the performance of a face alignment method represents the distance from the predicted landmarks $\hat{\mathbf{S}} = [\mathbf{x}_1; \mathbf{x}_2; \ldots; \mathbf{x}_N]$ to their ground truth location

[1] http://www.humansensing.cs.cmu.edu/intraface/

Method	Yaw	Pitch	Mean
Posit + AAM 2D + Cylindrical Model	3.68	8.83	6.26
Posit + ASM 2D + Cylindrical Model	3.56	5.52	4.54
Posit + AAM 2D + BFM Model	2.30	6.01	4.16
This work	4.33	3.41	3.87
Posit + ASM 2D + BFM Model	2.97	4.04	3.51

TABLE 5: Head pose estimation results obtained on the HPDB database, present in the mean absolute angular error in degrees. The accuracy of the methods except ours are taken from [2]. The results are sorted by the average error on both angles in a descending order.

$\mathbf{S}^{gt} = [\mathbf{y}_1; \mathbf{y}_2; \ldots; \mathbf{y}_N]$ divided by the interocular distance d_i:

$$E(\hat{\mathbf{S}}, \mathbf{S}^{gt}) = \frac{1}{N} \sum_{k=1}^{N} \frac{\|\mathbf{x}_k - \mathbf{y}_k\|_2}{d_i} \tag{16}$$

While providing a quantitative measure of accuracy for conventional face alignment tasks, this measure does not take viewpoint-consistency into account. For every single image, there exist many 3D shapes that share their x, y coordinates, while having different z-dimension. Therefore, Eq. 16 alone is not sufficient for validating the viewpoint consistency. To this end, we extend Eq. 16 and define E_{vc} as a measure of shape consistency that first transforms the shape $\mathbf{S_1}$ into the coordinate system of the shape $\mathbf{S_2}$ and computes the error in the following manner:

$$E_{vc}(\mathbf{S}_1, \mathbf{S}_2, \mathcal{P}) = \frac{1}{N} \sum_{k=1}^{N} \frac{\|(s\mathbf{R}\mathbf{x}_k + \mathbf{t}) - \mathbf{y}_k\|_2}{d_i} \tag{17}$$

where $\mathcal{P} = \{s, \mathbf{R}, \mathbf{t}\}$ defines the transformation between the coordinate systems. There are several ways to get $\mathcal{P}$. If the cameras for every viewpoint have joint calibration information, then the transformation parameters can be obtained by performing pair-wise calibration. Alternatively, if no calibration information is available, since the correspondence between the shapes is known, these parameters can be estimated by solving Eq. 6. In this way, Eq. 17 measures the differences in shapes only, regardless of their initial location in space.

5.4 Viewpoint-consistent evaluation

The majority of existing works in the literature predict only 2D facial landmarks. For comparison purposes we add the second step using the analysis discussed in Section 4. To quantify viewpoint-consistency in this section report experimental evaluation on the proposed MultiPIE-VC dataset. In Section 5.5 we report our results on the first 3D Face Alignment in the Wild Challenge.

Data. In order to evaluate under the viewpoint-consistent setting we extended the standard MultiPIE and introduce the MultiPIE Viewpoint Consistent (MultiPIE-VC) database. It contains 2169 images of 337 subjects annotated in a viewpoint-consistent manner. To perform the annotation we have selected 5 views of the original MultiPIE: from left half-profile to right half-profile. These views are in the consistent range of the ZFace tracker [27]. Figure 7 shows the selected views. Note that ZFace provides 1024 3D landmarks (see Figure 7, bottom), where the first 66 markers correspond to the main fiducial points (eg. jawline, eyes, mouth, etc). We manually inspected all the images to ensure that face alignment succeeded. For every image the annotation consists of 66 3D face landmarks in the camera coordinates. Additionally, we provide pairwise camera calibration parameters for cross-view experiments. In total there are 20 pairs of views.

Setting. We compare methods from the 2D-3D group and the 3D group on MultiPIE-VC. Since the works of Cao et al. [10][2], Kazemi et al. [32][3] and Zhu et al. [78][4] are 2D by their nature, we add the second step by fitting the deformable model to their 2D outputs, as discussed in Section 4, and take the estimated shape for comparison purposes. The methods of Saragih et al. [49][5] and Jeni et al. [27] are initially 2D-3D. We have reimplemented the 3D method of Tulyakov et al. [55], 2D-3D method of Jeni et al. [27] and we have reimplemented and extended the originally 2D method of Xiong et al. [67] to predict the third dimension as discussed in Section 3.2, making it belong to the 3D group of methods. The methods of Cao et al. [10], Kazemi et al. [32] and Zhu et al. [78] were trained on the publicly available 300-W database [47] using the provided landmarks. To train/test the 3D methods of Tulyakov et al. [55], Xiong et al. [67], Saragih et al. [49] and Jeni et al. [27] we perform five fold cross-validation on MultiPIE-VC.

Some of these methods were trained on the 68-landmarks annotation provided with the Multi-PIE and the 300-W datasets. We note, that the only difference with our 66-landmarks annotation are the two extra inner mouth-corner landmarks. We removed these extra points where they were present to make the annotation consistent with the 66-landmarks configuration. Additionally, we note that each of the compared methods assumes only a single image to produce the output.

Consistency measures. We define the ground truth landmarks and the prediction for the view c_k as $\mathbf{S}^{gt_k}$ and $\hat{\mathbf{S}}^{c_k}$ correspondingly. To extensively compare viewpoint consistency of different methods we define four validation metrics:

- **Ground Truth Consistency error I (GTC-I)**: a standard error measure adopted by previous works. It is computed using Eq. 16 as $E(\hat{\mathbf{S}}^{c_k}, \mathbf{S}^{gt_k})$. GTC-I penalizes for both differences: in shape and its location in space.
- **Ground Truth Consistency error II (GTC-II)**: defined using Eq. 17 as $E_{vc}(\hat{\mathbf{S}}^{c_k}, \mathbf{S}^{gt_k}, \mathcal{P})$, with the purpose to encompass only the differences in shape measured in the common coordinate system.
- **Cross-View Prediction Consistency error (CVPC)**. Having two independent estimates $\hat{\mathbf{S}}^{c_p}$ and $\hat{\mathbf{S}}^{c_l}$ for different views c_p and c_l of the same subject, we check for prediction consistency by computing $E_{vc}(\hat{\mathbf{S}}^{c_p}, \hat{\mathbf{S}}^{c_l}, \mathcal{P})$.
- **Cross-View Ground Truth Consistency error (CVGTC)**. Instead of comparing the two predictions, this measure analyses the correctness of the estimate $\hat{\mathbf{S}}^{c_p}$ by comparing it with the ground truth for another view $\mathbf{S}^{gt_l}$ in the following way: $E_{vc}(\hat{\mathbf{S}}^{c_p}, \mathbf{S}^{gt_l}, \mathcal{P})$, where $p \neq l$.

For consistency measures, requiring the estimated transformation parameters $\mathcal{P}$, we report two types of experimental results. For the first type we estimated $\mathcal{P}$ using the Eq. 6. The transformation parameters for the second type of results were obtained using the camera calibration information provided in MultiPIE. We, therefore, refer to these two types of results as estimation and calibration correspondingly.

[2] https://github.com/ming81/FaceAlignment
[3] https://github.com/davisking/dlib
[4] We used the implementation provided by the authors
[5] https://github.com/kylemcdonald/FaceTracker

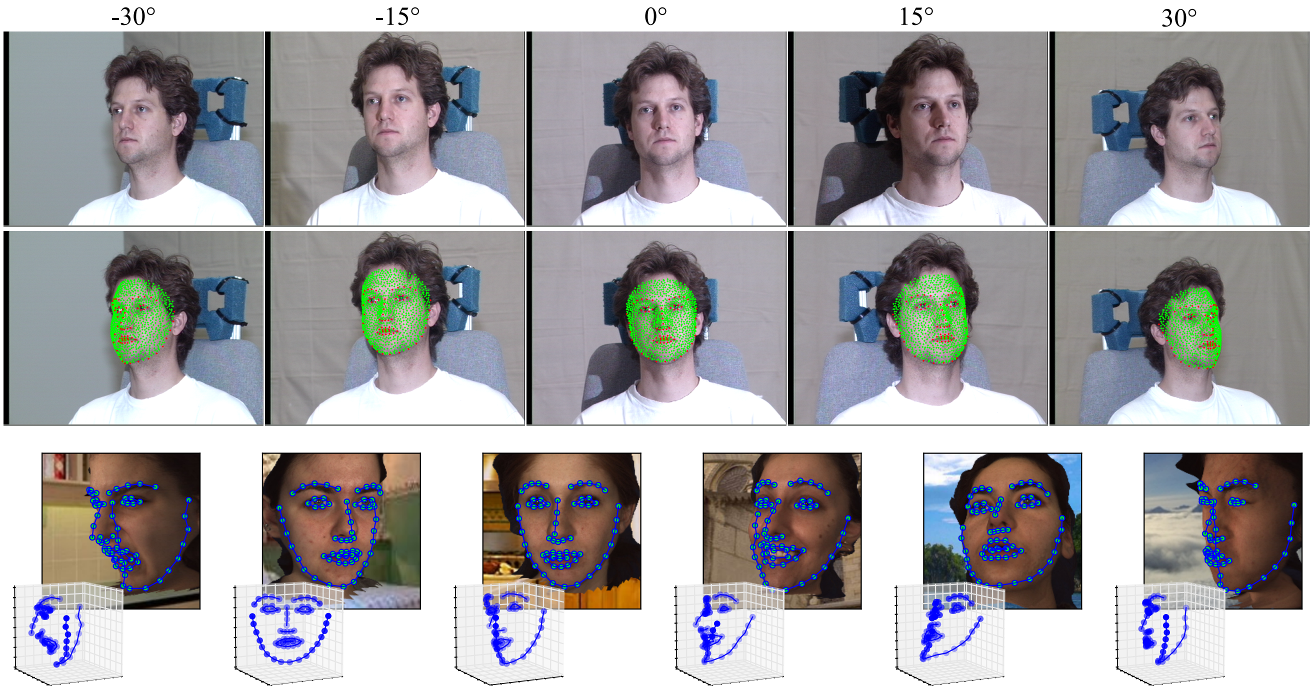

Fig. 7: Selected examples from the benchmark datasets. Selected views from the MultiPIE-VC (a), automatic 3D annotation performed using ZFace tracker (b). Rendered images of the BU-4DFE dataset with 3D ground-truth.

TABLE 6: viewpoint-consistent landmarks detection accuracy among different methods on MultiPIE-VC database (best performance in bold). For cross-view metrics two types of results are reported, based on the method for estimating the transformation parameters.

	Method	GTC-I	GTC-II	CVPC		CVGTC	
				Estimation	Calibration	Estimation	Calibration
2D-3D	Cao et al. [12]	0.266	0.152	0.057	0.169	0.156	0.224
	Kazemi et al. [32]	0.244	0.149	0.062	0.127	0.155	0.212
	Saragih et al. [49]	0.295	0.146	0.050	0.207	0.149	0.240
	Zhu et al. [78]	0.282	0.153	0.070	0.156	0.157	0.226
	Jeni et al. [27]	0.075	0.050	0.075	0.141	0.066	0.133
3D	Xiong et al. [67]	0.106	0.053	0.059	0.119	0.066	0.128
	This work	**0.074**	**0.044**	**0.048**	**0.104**	**0.059**	**0.120**

In addition to multiple views, the original MultiPIE database contains different expressions and illuminations per subject. Clearly, facial expressions change the shape of the face. Therefore, to compare methods using the proposed cross-view consistency measures, we group the subjects by their id and expression, and perform cross-view comparison for each group, averaging the group results to obtain the value of the particular measure for the whole database.

3D landmarks localization. Ideally, a viewpoint-consistent method should perform well under all consistency measures. Table 6 shows the results for two groups of methods obtained on the MultiPIE-VC database. GTC-I, GTC-II, CVGTC show the consistency of the prediction to the ground truth of either the same view or a different one. Methods performing poorly on these measures fail to provide an estimate that preserves the face shape (i.e. consistency with the ground truth) across the tested viewpoints.

Due to fitting a deformable model during their second step the 2D-3D methods show CVPC close to 3D methods. However, their output shapes are inconsistent with the ground truth. The low CVPC value indicates that cross-view estimates of the method have a similar shape. Methods trained using viewpoint-consistent annotation perform similarly on all four consistency measures. This supports the value of predicting viewpoint-consistent 3D landmarks. Note that CVPC and CVGTC computed with the transformation parameters estimated using Eq. 6 are smaller compared to the case when using the camera calibration parameters. This is due to usually nonzero reprojection errors obtained when performing pairwise calibration of two cameras.

To show pairwise consistency, we report errors computed against each pair of views. Figure 8 shows CVPC (top) and CVGTC (bottom) for every method. The transformation parameters $\mathcal{P}$ were estimated using Eq. 6. Note how the methods find the most distant views as the most difficult ones. For example, for the opposite views (e.g. $-30°$ and $30°$ or $-15°$ and $30°$), many of the methods show the worst consistency results. Methods trained on the consistent landmarks outperform others by a large margin for every view, emphasizing the importance of shifting to the viewpoint-consistent formulation of the face alignment problem. Figure 9 shows qualitative results of the method in [55] (top) and the method in [27] (bottom) trained on MultiPIE-VC and applied to images from 300-W [47].

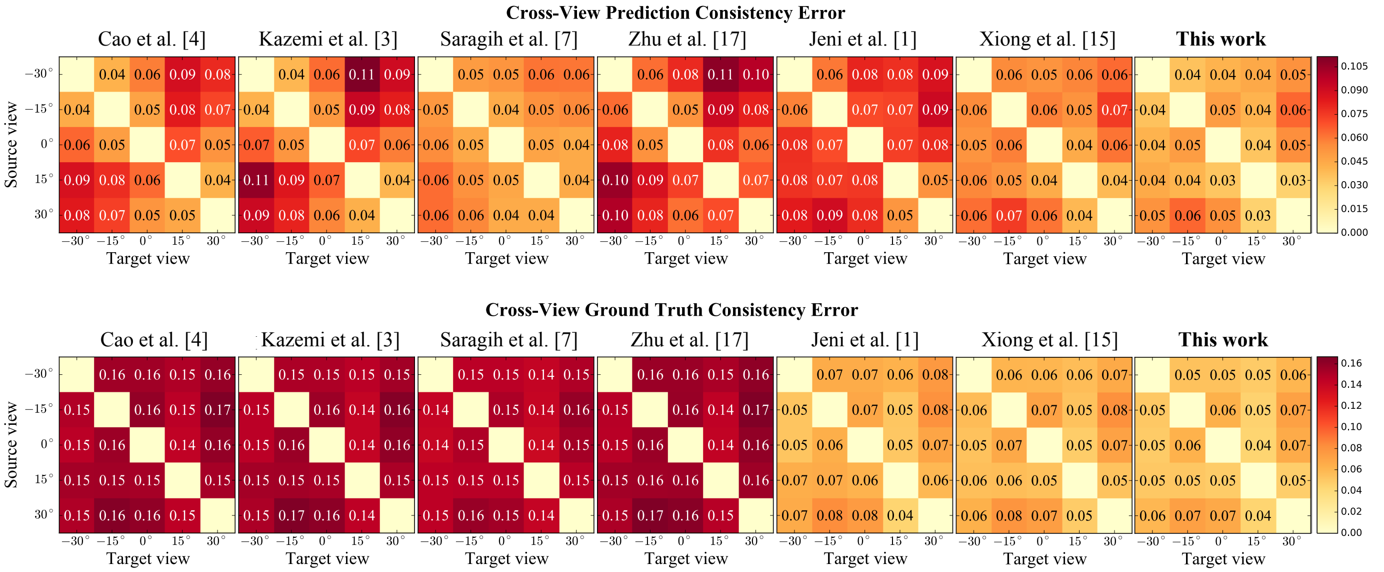

Fig. 8: Pairwise view consistencies for every method are reported. Top: CVPC. Bottom: CVGTC. Cross-view transformation parameters were estimated using Eq. 6

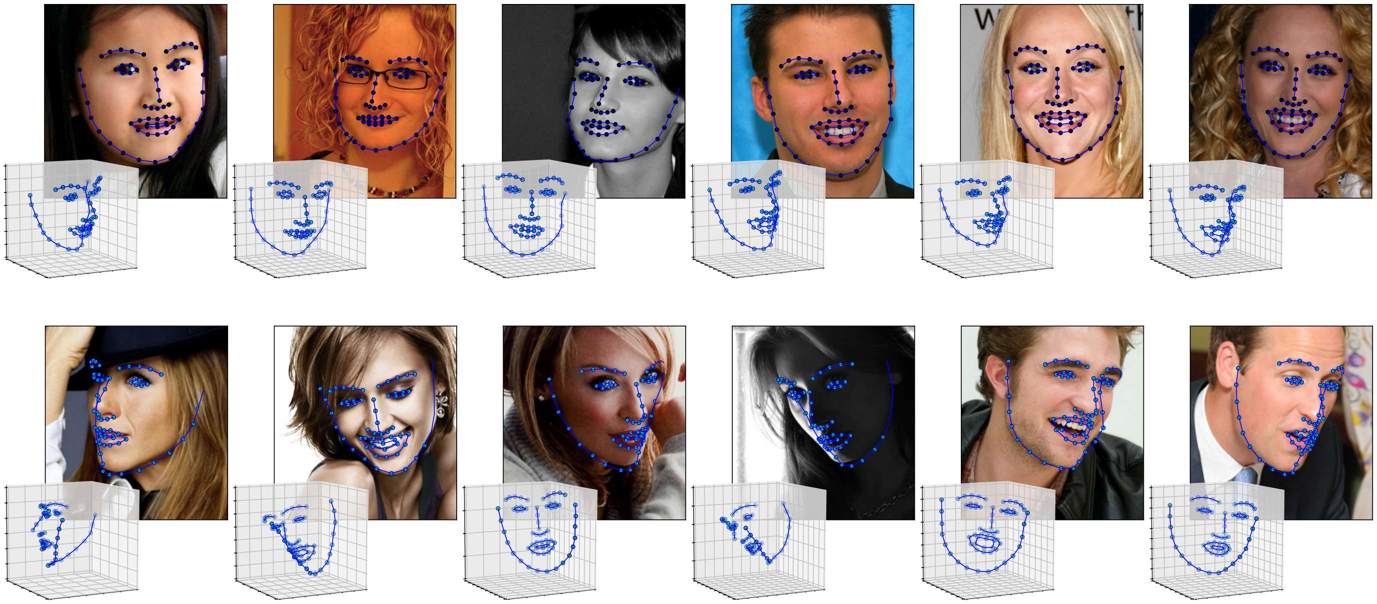

Fig. 9: Selected images from the 300-W [47] challenge are plotted. Each example contains the original image with the plotted 66 landmarks and the landmarks in 3D space.

5.5 3D Face Alignment in the Wild Challenge

To improve progress on viewpoint-consistent 3D face alignment we have organized the First 3D Face Alignment in the Wild Challenge (3DFAW)[6]. The challenge offered consistently annotated face images coming from four different sources. The first two sources were rendered using the BP4D-Spontaneous [75] and the BU-4DFE [71] database, the third source was the proposed MultiPIE-VC database. The forth source included time-sliced videos downloaded from the internet. Therefore the 3DFAW data included images taken in the lab and uncontrolled in-the-wild images. All the four sources were consistently annotated in 3D. The challenge consisted of 3 phases. During the first phase the participants were provided with access to the training set of the images, their ground truth 3D landmarks and face bounding boxes. During the second phase we released the validation set with the ground truth information. The last phase provided testing images with face bounding boxes only. The participants uploaded their predictions via the CodaLab platform[7]. Table 7 shows the distributions of the number of images provided at every phase. Evaluation was performed as discussed in Section 5.3. More details about the challenge are given in [29]. Currently, the post-challenge phase is open.

[6] http://mhug.disi.unitn.it/workshop/3dfaw/

[7] https://competitions.codalab.org/competitions/10261

TABLE 7: Distribution of the different sets.

	Training	Validation	Test	Total
BP-4DFE	5677	1960	1918	9555
BP-4D-Spontaneous	3794	1365	1351	6510
MultiPIE	4200	1400	1400	7000
TimeSliced	298		243	541

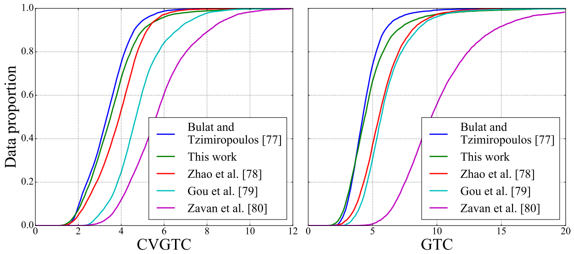

Fig. 10: Cumulative error distributions for CVGTC and GTC for different methods.

We have used the training and the validation sets to train the method. The test set was used to evaluate the performance of the method. Table 8 gives the final ranking of the methods on the test set. The numbers for all the method except ours are taken from the CodaLab ranking dashboard. Finally, Figure 10 plots the cumulative distribution errors for all the methods.

TABLE 8: Prediction Consistency Error (CVGTC) and Ground Truth Consistency (GTC) of the different methods on the Test set.

Rank	Method	CVGTC	GTC
1	Bulat and Tzimiropoulos [7]	0.034	0.045
2	This work	0.038	0.051
3	Zhao et al. [76]	0.039	0.058
4	Gou et al. [18]	0.049	0.062
5	Zavan et al. [16]	0.059	0.108

The method discussed in this study shows the second best scores on both CVGTC and GTC. We note that the works of Bulat and Tzimiropoulos [7] and Zhao et al. [76] are based on training deep neural networks, requiring mutlicore GPUs for time-efficient inference, while the work presented in this study proposes a highly competitive solution and has low hardware requirements for real-time operation.

6 Conclusion

Viewpoint-consistent 3D Face Alignment offers an important means for detecting 3D face landmarks, i.e., the estimated key-points are consistent across views when a subject is captured in a multi-camera setup. Such behavior is achieved by training the models on viewpoint-consistent data, introduced in this study.

Previous works attained a similar form of consistency by employing a computationally extensive second step of fitting a morphable model. In this study we showed that the second step can be avoided, making the models capable of reaching impressive framerates. Additionally, we showed that although two-step methods have prediction consistency between pairs of views, they are not able to provide estimates consistent with the ground for the same pairs of views, showing high CVGTC errors. Moreover, when a pair of views under consideration includes distant views, 2D-3D works show higher error rates as compared to the methods trained in a viewpoint-consistently.

In addition to viewpoint-consistency, methods trained on the proposed data detect 3D landmarks and preserve semantic correspondence of the landmarks — a feature not available for the standard methods, as they detect only visible points, while the face can be severely occluded. In such a setting, viewpoint-consistent methods detect the true 3D location of a 3D point. Such behavior offers an additional advantage, enabling a simple direction-based head pose estimation method. We showed that the method outperforms or reaches highly competitive accuracy scores on a range of benchmarks.

Given the incompatibility of the standard landmarks and our proposed viewpoint-consistent landmarks, we proposed a means of comparing methods trained on different sources of data. We argued that the standard methods trained on inconsistent landmarks can be compared by adding the third dimension via the second step of 3D morphable model fitting, typically done in 2D-3D works. Further comparisons can be made by training all the methods on the viewpoint-consistent data directly. To assess different aspects of viewpoint-consistency we proposed four different consistency metrics and showed that the best methods give low errors under all four scores, while 2D-3D methods are unable to operate uniformly well in this case.

We have discussed a regression forest-based method that features viewpoint-consistency by adding the third dimension to the cascaded pipeline. The method showed best scores on the standard evaluation, when the only difference is presence/absence of the third dimension during training. This supports the idea that adding the third dimension improves even the 2D landmark localization accuracy. We further showed in this study, that the method offers the best accuracy under viewpoint-consistent setting, showing the best scores under all consistency measures compared to six competing methods. Since the the third dimension is naturally incorporated into the pipeline, the method provides head pose estimates for free without any extra computation. In this study we showed, that these estimates are highly competitive obtaining the first, the second or third-best scores when compared on three different baselines.

In order for the community to adopt the proposed research direction, we organized the 3DFAW challenge, in which each participant was provided with training data coming from different sources and consistently annotated in 3D. The testing data included only images, without providing the ground-truth labels. The method discussed in this study showed the second performance, giving the first place to the method exploiting deep neural architectures, requiring multicore GPUs for time-efficient inference. Our method shows impressive frame rates, having moderate hardware requirements.

We believe that viewpoint-consistent 3D face alignment is a promising research direction, with multiple future contributions to come. For example, in this study we considered sparse face shape estimation, where only a small set of 3D face points is determined. There is nothing, however, that restricts viewpoint-consistent methods to perform dense shape regression, estimating the 3D mesh instead of 3D outline.

References

[1] K. H. An and M. J. Chung, "3d head tracking and pose-robust 2d texture map-based face recognition using a simple ellipsoid model," in *Intel-*

ligent Robots and Systems, 2008. IROS 2008. IEEE/RSJ International Conference on, 2008, pp. 307–312.

[2] M. Ariz, J. J. Bengoechea, A. Villanueva, and R. Cabeza, "A novel 2d/3d database with automatic face annotation for head tracking and pose estimation," *Computer Vision and Image Understanding*, vol. 148, pp. 201–210, 2016.

[3] S. Asteriadis, D. Soufleros, K. Karpouzis, and S. Kollias, "A natural head pose and eye gaze dataset," in *International Workshop on Affective-Aware Virtual Agents and Social Robots*, 2009, p. 1.

[4] T. Baltrusaitis, P. Robinson, and L. P. Morency, "3D Constrained Local Model for rigid and non-rigid facial tracking," in *CVPR*, 2012, pp. 2610–2617.

[5] P. N. Belhumeur, D. W. Jacobs, D. J. Kriegman, and N. Kumar, "Localizing parts of faces using a consensus of exemplars," in *CVPR*, 2011.

[6] V. Blanz and T. Vetter, "Face recognition based on fitting a 3D morphable model," *PAMI*, vol. 25, no. 9, pp. 1063–1074, 2003.

[7] A. Bulat and G. Tzimiropoulos, "Two-stage convolutional part heatmap regression for the 1st 3d face alignment in the wild (3DFAW) challenge," in *ECCVW*, 2016.

[8] X. P. Burgos-Artizzu, P. Perona, and P. Dollar, "Robust face landmark estimation under occlusion," in *ICCV*, 2013, pp. 1513–1520.

[9] C. Cao, Q. Hou, and K. Zhou, "Displaced dynamic expression regression for real-time facial tracking and animation," in *SIGGRAPH*, vol. 33, no. 4, 2014.

[10] C. Cao, Y. Weng, S. Lin, and K. Zhou, "3D Shape Regression for Real-time Facial Animation," in *SIGGRAPH*, vol. 32, no. 4, 2013.

[11] C. Cao, Y. Weng, S. Zhou, Y. Tong, and K. Zhou, "FaceWarehouse: A 3D facial expression database for visual computing," in *Trans. of Visualization and Computer Graphics*, vol. 20, no. 3, 2014, pp. 413–425.

[12] X. Cao, "Face alignment by Explicit Shape Regression," in *CVPR*, 2012, pp. 2887–2894.

[13] T. F. Cootes and C. J. Taylor, "Active Shape Models - 'Smart Snakes'," in *BMVC*, 1992, pp. 266–275.

[14] T. Cootes, G. Edwards, and C. Taylor, "Active appearance models," in *ECCV*, 1998.

[15] T. F. Cootes and C. J. Taylor, "Active Shape Model Search using Local Grey-Level Models: A Quantitative Evaluation," in *BMVC*, 1993.

[16] F. H. de B. Zavan, A. C. P. Nascimento, L. P. e Silva, O. R. P. Bellon, and L. Silva, "3d face alignment in the wild: A landmark-free, nose-based approach," in *ECCVW*, 2016.

[17] G. Fanelli, M. Dantone, and L. Van Gool, "Real time 3D face alignment with Random Forests-based Active Appearance Models," in *FG*, 2013.

[18] C. Gou, Y. Wu, F.-Y. Wang, and Q. Ji, "Shape augmented regression for 3d face alignment," in *ECCVW*, 2016.

[19] R. Gross, R. Gross, I. Matthews, I. Matthews, S. Baker, and S. Baker, "Generic vs. Person Specific Active Appearance Models," *IVC*, 2005.

[20] R. Gross, I. Matthews, J. Cohn, and T. Kanade, "Multi-PIE," in *FG*, 2008.

[21] ——, "Multi-PIE," in *FG*, 2008, pp. 1 – 8.

[22] S. Gupta, K. R. Castleman, M. K. Markey, and A. C. Bovik, "Texas 3D Face Recognition Database," in *Southwest Symposium on Image Analysis and Interpretation*, 2010, pp. 97–100.

[23] T. Hassner, "Viewing Real-World Faces in 3D," in *ICCV*, 2013, pp. 3607–3614.

[24] P.-L. Hsieh, C. Ma, J. Yu, and H. Li, "Unconstrained realtime facial performance capture," in *CVPR*, 2015.

[25] C. Huang, X. Ding, and C. Fang, "Pose robust face tracking by combining view-based AAMs and temporal filters," *CVIU*, vol. 116, no. 7, pp. 777–792, 2012.

[26] G. B. Huang, M. Ramesh, T. Berg, and E. Learned-Miller, "Labeled faces in the wild: A database for studying face recognition in unconstrained environments," University of Massachusetts, Amherst, Tech. Rep. 07-49, October 2007.

[27] L. A. Jeni, J. F. 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. IEEE, 2015, pp. 1–8.

[28] ——, "Dense 3d face alignment from 2d video for real-time use," *Image and Vision Computing*, vol. 58, pp. 13–24, 2017.

[29] L. A. Jeni, S. Tulyakov, L. Yin, N. Sebe, and J. F. Cohn, "The first 3d face alignment in the wild (3dfaw) challenge," in *ECCV*, 2016, pp. 511–520.

[30] O. Jesorsky, K. J. Kirchberg, and R. Frischholz, "Robust face detection using the hausdorff distance," in *Intl. Conf. on Audio- and Video-Based Biometric Person Authentication*, 2001, pp. 90–95.

[31] A. Jourabloo and X. Liu, "Pose-Invariant 3D Face Alignment," in *ICCV*, 2015.

[32] V. Kazemi and S. Josephine, "One Millisecond Face Alignment with an Ensemble of Regression Trees," in *CVPR*, 2014, pp. 1867–1874.

[33] D. E. King, "Dlib-ml: A machine learning toolkit," *Journal of Machine Learning Research*, vol. 10, pp. 1755–1758, 2009.

[34] M. Köstinger, P. Wohlhart, P. M. Roth, and H. Bischof, "Annotated facial landmarks in the wild: A large-scale, real-world database for facial landmark localization," in *ICCV Worshops*, 2011, pp. 2144–2151.

[35] S. Kumano, K. Otsuka, J. Yamato, E. Maeda, and Y. Sato, "Pose-invariant facial expression recognition using variable-intensity templates," *IJVC*, vol. 83, no. 2, pp. 178–194, 2009.

[36] M. La Cascia, S. Sclaroff, and V. Athitsos, "Fast, reliable head tracking under varying illumination: An approach based on registration of texture-mapped 3d models," *PAMI*, vol. 22, no. 4, pp. 322–336, 2000.

[37] V. Le, J. Brandt, Z. Lin, L. Bourdev, and T. S. Huang, "Interactive facial feature localization," in *ECCV*, 2012, pp. 679–692.

[38] H. Li, J. Yu, Y. Ye, and C. Bregler, "Realtime facial animation with on-the-fly correctives," in *SIGGRAPH*, vol. 32, 2013, p. 1.

[39] C. Liu, J. Yuen, S. Member, and A. Torralba, "SIFT flow: dense correspondence across difference scenes," *PAMI*, vol. 33, no. 5, pp. 978 – 994, 2011.

[40] I. Matthews and S. Baker, "Active Appearance Models Revisited," *IJCV*, vol. 60, no. 2, pp. 135–164, 2004.

[41] K. Messer, J. Matas, J. Kittler, and K. Jonsson, "Xm2vtsdb: The extended m2vts database," in *Intl. Conf. on Audio and Video-based Biometric Person Authentication*, 1999, pp. 72–77.

[42] S. Milborrow, J. Morkel, and F. Nicolls, "The MUCT Landmarked Face Database," *Pattern Recognition Association of South Africa*, 2010.

[43] P. Paysan, R. Knothe, B. Amberg, S. Romdhani, and T. Vetter, "A 3D Face Model for Pose and Illumination Invariant Face Recognition," in *International Conference on Advanced Video and Signal Based Surveillance*, 2009, pp. 296 – 301.

[44] S. Ren, X. Cao, Y. Wei, and J. Sun, "Face Alignment at 3000 FPS via Regressing Local Binary Features," in *CVPR*, 2014, pp. 1685 – 1692.

[45] J. Roth, Y. Tong, and X. Liu, "Unconstrained 3d face reconstruction," in *CVPR*, 2015.

[46] ——, "Adaptive 3d face reconstruction from unconstrained photo collections," in *CVPR*, June 2016.

[47] C. Sagonas, G. Tzimiropoulos, S. Zafeiriou, and M. Pantic, "300 faces in-the-wild challenge: The first facial landmark Localization Challenge," in *ICCV Worshops*, 2013, pp. 397–403.

[48] E. Sangineto, "Pose and expression independent facial landmark localization using dense-surf and the hausdorff distance," *PAMI*, vol. 35, no. 3, pp. 624–638, 2013.

[49] J. M. Saragih, S. Lucey, and J. F. Cohn, "Deformable model fitting by regularized landmark mean-shift," *IJCV*, vol. 91, no. 2, pp. 200–215, 2011.

[50] A. Savran, N. Alyüz, H. Dibeklioğlu, O. Çeliktutan, B. Gökberk, B. Sankur, and L. Akarun, *Bosphorus Database for 3D Face Analysis*, 2008, pp. 47–56.

[51] J. Sung, T. Kanade, and D. Kim, "Pose robust face tracking by combining active appearance models and cylinder head models," *IJVC*, vol. 80, no. 2, pp. 260–274, 2008.

[52] Y. Taigman, M. Yang, M. Ranzato, and L. Wolf, "DeepFace: Closing the Gap to Human-Level Performance in Face Verification," in *CVPR*, 2014, pp. 1701 – 1708.

[53] J. Thies, M. Zollhofer, M. Stamminger, C. Theobalt, and M. Niessner, "Face2face: Real-time face capture and reenactment of rgb videos," in *CVPR*, June 2016.

[54] G. Trigeorgis, P. Snape, M. A. Nicolaou, E. Antonakos, and S. Zafeiriou, "Mnemonic descent method: A recurrent process applied for end-to-end face alignment," in *CVPR*, June 2016.

[55] S. Tulyakov and N. Sebe, "Regressing a 3d face shape from a single image," in *ICCV*, 2015.

[56] S. Tulyakov, R. L. Vieriu, S. Semeniuta, and N. Sebe, "Robust Real-Time Extreme Head Pose Estimation," in *ICPR*, 2014.

[57] S. Tulyakov, X. Alameda-Pineda, E. Ricci, L. Yin, J. F. Cohn, and N. Sebe, "Self-adaptive matrix completion for heart rate estimation from face videos under realistic conditions," in *CVPR*, 2016.

[58] G. Tzimiropoulos, "Project-Out Cascaded Regression with an application to Face Alignment," in *CVPR*, 2015.

[59] G. Tzimiropoulos and M. Pantic, "Optimization problems for fast AAM fitting in-the-wild," in *ICCV*, 2013, pp. 593–600.

[60] R. Valenti, N. Sebe, and T. Gevers, "Combining head pose and eye location information for gaze estimation," *TIP*, vol. 21, no. 2, pp. 802–815, 2012.

[61] F. Vicente, Z. Huang, X. Xiong, F. De la Torre, W. Zhang, and D. Levi, "Driver gaze tracking and eyes off the road detection system," *IEEE Transactions on Intelligent Transportation Systems*, vol. 16, no. 4, pp. 2014–2027, 2015.

[62] N. Wang, X. Gao, D. Tao, and X. Li, "Facial Feature Point Detection: A Comprehensive Survey," *arXiv*, 2014. [Online]. Available: http://arxiv.org/abs/1410.1037

[63] T. Weise, S. Bouaziz, H. Li, and M. Pauly, "Realtime performance-based facial animation," in *SIGGRAPH*, 2011.

[64] T. Weise, H. Li, L. Van Gool, and M. Pauly, "Face/off: Live facial puppetry," in *SIGGRAPH*, 2009, pp. 7–16.

[65] J. Xiao, J. Hays, K. a. Ehinger, A. Oliva, and A. Torralba, "SUN database: Large-scale scene recognition from abbey to zoo," in *CVPR*, 2010, pp. 3485–3492.

[66] J. Xiao, T. Moriyama, T. Kanade, and J. F. Cohn, "Robust full-motion recovery of head by dynamic templates and re-registration techniques," *International Journal of Imaging Systems and Technology*, vol. 13, no. 1, pp. 85–94, 2003.

[67] X. Xiong and F. De La Torre, "Supervised descent method and its applications to face alignment," in *CVPR*, 2013, pp. 532–539.

[68] X. Xiong and F. D. Torre, "Global Supervised Descent Method," in *CVPR*, 2015.

[69] H. Yang and I. Patras, "Sieving Regression Forest Votes for Facial Feature Detection in the Wild," in *ICCV*, 2013, pp. 1936–1943.

[70] D. Yi, Z. Lei, and S. Z. Li, "Towards Pose Robust Face Recognition," in *CVPR*, 2013, pp. 3539–3545.

[71] L. Yin, X. Chen, Y. Sun, T. Worm, and M. Reale, "A high-resolution 3d dynamic facial expression database," in *FG*, 2008.

[72] L. Yin, X. Wei, Y. Sun, J. Wang, and M. J. Rosato, "A 3d facial expression database for facial behavior research," in *FG*, 2006.

[73] X. Yu, J. Huang, S. Zhang, W. Yan, and D. N. Metaxas, "Pose-Free Facial Landmark Fitting via Optimized Part Mixtures and Cascaded Deformable Shape Model," in *ICCV*, 2013, pp. 1944–1951.

[74] J. Zhang, M. Kan, S. Shan, and X. Chen, "Occlusion-free face alignment: Deep regression networks coupled with de-corrupt autoencoders," in *CVPR*, June 2016.

[75] X. Zhang, L. Yin, J. F. Cohn, S. Canavan, M. Reale, A. Horowitz, P. Liu, and J. M. Girard, "BP4D-Spontaneous: a high-resolution spontaneous 3D dynamic facial expression database," *Image and Vision Computing*, vol. 32, no. 10, pp. 692–706, 2014.

[76] R. Zhao, Y. Wang, C. F. Benitez-Quiroz, Y. Liu, and A. M. Martinez, "Fast & precise face alignment and 3d shape recovery from a single image," in *ECCVW*, 2016.

[77] S. K. Zhou and D. Comaniciu, "Shape regression machine," in *Medical Imaging*, vol. 20, 2007.

[78] S. Zhu, C. Li, C. Change, and X. Tang, "Face Alignment by Coarse-to-Fine Shape Searching," in *CVPR*, 2015.

[79] X. Zhu and D. Ramanan, "Face detection, pose estimation, and landmark estimation in the wild." in *CVPR*, 2012, pp. 2879 – 2886.

[80] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: A 3d solution," in *CVPR*, June 2016.

Sergey Tulyakov, PhD., is a Senior Research Engineer at Shapchat Research. His research focuses on computer vision, machine learning and face analysis, including 2D and 3D detection, tracking, pose estimation, heart rate estimation from videos, 3D face alignment techniques, with particular emphasis on realistic capturing conditions. He has co-organized the 3D Face Alignment in the Wild workshop held in conjunction with ECCV in 2016. He received his PhD degree from the University of Trento, Italy.

László A. Jeni, PhD., is Project Scientist at the Carnegie Mellon University, Pittsburgh, PA, USA. He received his M.S. degree in Computer Science from the Eotvos Lorand University, Hungary, and his Ph.D. degree in Electrical Engineering and Information Systems from the University of Tokyo, Japan. His research interests are in the fields of Computer Vision and Machine Learning. He develops advanced methods of 2D and 3D automatic analysis and synthesis of facial expressions; and applies those tools to research in human emotion, non-verbal communication, and assistive technology. He has co-organized the 3D Face Alignment workshop in 2016, the third Facial Expression Recognition and Analysis Challenge in 2017, and he is an Area Chair at IEEE FG 2018. His honors include best paper awards at IEEE HSI 2011 and at IEEE FG 2015 conferences. He is a member of the Affect Analysis Group at the University of Pittsburgh, USA, a member of the NAIST International Collaborative Laboratory for Robotics Vision, Japan, and a Founding member of the Section of Robotics, John von Neumann Computer Society, Hungary.

Jeffrey F. Cohn, PhD., is Professor of Psychology and Psychiatry at the University of Pittsburgh and Adjunct Professor of Computer Science at the Robotics Institute at CMU. He has led interdisciplinary and inter-institutional efforts to develop advanced methods of automatic analysis and synthesis of facial expression and applied them to research in human emotion, social interaction, pain, and psychopathology. Along with his collaborators, they have developed leading approaches to cylindrical head tracking, multi-view face tracking, deformable face models and nonrigid face tracking, and most recently dense 3D registration from 2D video. They have created widely-used benchmark databases, including Cohn-Kanade and CK+, CMU MultiPIE, and BP4D- Spontaneous. Dr. Cohn has served as Co-Chair of the 2008 and 2015 IEEE International Conference on Automatic Face and Gesture Recognition (FG2008) (FG2015), the 2009 International Conference on Affective Computing and Intelligent Interaction (ACII2009), the Steering Committee for IEEE International Conference on Automatic Face and Gesture Recognition, and the 2014 International Conference on Multimodal Interfaces (ACM 2014). He has co-edited special issues of the Journal of Image and Vision Computing and is a Co-Editor of IEEE Transactions on Affective Computing (TAC).

Nicu Sebe, PhD., is a professor in the University of Trento, Italy, leading the research in the areas of multimedia information retrieval and human behavior understanding. He was General Co-chair of the IEEE FG Conference 2008 and ACM Multimedia 2013, and Program Chair of the International Conference on Image and Video Retrieval in 2007 and 2010 and ACM Multimedia 2007 and 2011. He was a Program Chair of ECCV 2016 and ICCV 2017. He is a fellow of IAPR.