Unsupervised Performance Analysis of 3D Face Alignment with a Statistically Robust Confidence Test

Mostafa Sadeghi, Xavier Alameda-Pineda, Radu Horaud

Unsupervised Performance Analysis of 3D Face Alignment with a Statistically Robust Confidence Test*

Mostafa Sadeghi[a], Xavier Alameda-Pineda[b], Radu Horaud[b]

[a]*Centre Inria Université de Lorraine, France*
[b]*Centre Inria Université Grenoble Alpes, France*

Abstract

This paper addresses the problem of analysing the performance of 3D face alignment (3DFA), or facial landmark localization. This task is usually supervised, based on annotated datasets. Nevertheless, in the particular case of 3DFA, the annotation process is rarely error-free, which strongly biases the results. Alternatively, unsupervised performance analysis (UPA) is investigated. The core ingredient of the proposed methodology is the robust estimation of the rigid transformation between predicted landmarks and model landmarks. It is shown that the rigid mapping thus computed is affected neither by non-rigid facial deformations, due to variabilities in expression and in identity, nor by landmark localization errors, due to various perturbations. The guiding idea is to apply the estimated rotation, translation and scale to a set of predicted landmarks in order to map them onto a *mathematical home* for the shape embedded in these landmarks (including possible errors). UPA proceeds as follows: (i) 3D landmarks are extracted from a 2D face using the 3DFA method under investigation; (ii) these landmarks are rigidly mapped onto a canonical (frontal) pose, and (iii) a statistically-robust confidence score is computed for each landmark. This allows to assess whether the mapped landmarks lie inside (inliers) or outside (outliers) a *confidence volume*. An experimental evaluation protocol, that uses publicly available datasets and several 3DFA software packages associated with published articles, is described in detail. The results show that the proposed analysis is consistent with supervised metrics and that it can be used to measure the accuracy of both predicted landmarks and of automatically annotated 3DFA datasets, to detect errors and to eliminate them. Source code and supplemental materials for this paper are publicly available at https://team.inria.fr/robotlearn/upa3dfa/.

Keywords: Deep face alignment, 3D facial landmarks, Gaussian-uniform mixture, Student's t-distribution, robust statistical inference, rigid motion estimation, expectation-maximization algorithm, quaternion.

1. Introduction

The problem of face alignment is the problem of facial landmark localization from a single RGB image. It is an important research topic as it provides input to a variety of tasks, e.g. face tracking, face recognition, expression recognition, visual speech processing, facial animation, etc., [1, 2, 3]. 2D face alignment (2DFA) has been extensively studied for the last decades, yielding a plethora of methods and algorithms [4]. State of the art 2DFA is based on DNNs, e.g. [5], or it combines DNN with differentiable optical-flow and/or 3D-triangulation modules to supervise the location of 2D landmarks [6, 7].

In general, 2DFA yields poor performance in the presence of occlusions which occur in case of large poses induced by out-of-image-plane head rotations (self occlusions) as well as by the presence of various objects in the camera field of view, such as glasses, hair, hands or handheld objects. We note however, that more recently there have been successful attempts to develop 2DFA that can deal with extreme poses and partial occlusions, e.g. [8, 9]. In particular, a recent method based on transformers, namely reference heatmap transformers (RHT) yields impressive 2DFA results [10]. We also note that there is an increasing interest in 3DFA. 3D facial landmarks embed both head-pose with six degrees of freedom and rich non-rigid deformation information . However, the caveat is that 3DFA training comes with additional difficulties, notably due to the fact that annotation should be carried out automatically: Indeed, accurate manual annotation of the z-coordinate (depth) is impractical if not impossible.

This paper proposes Unsupervised Performance Analysis (UPA) for benchmarking 3DFA algorithms. At runtime, UPA-3DFA takes as input a set of 3D landmarks predicted from a 2D face and classifies these landmarks either as inliers or as outliers. The analysis can be applied to landmarks predicted by a trained DNN architecture, as well as to landmarks extracted with semi-automatic annotation techniques, e.g. [11, 12, 13].

The first contribution is the robust estimation of the rigid transformation (rotation, translation and scale) that maps the set of predicted 3D landmarks onto a frontally-viewed shape model. The guiding idea is to obtain a *natural mathematical home* for the shape embedded in the predicted landmarks,

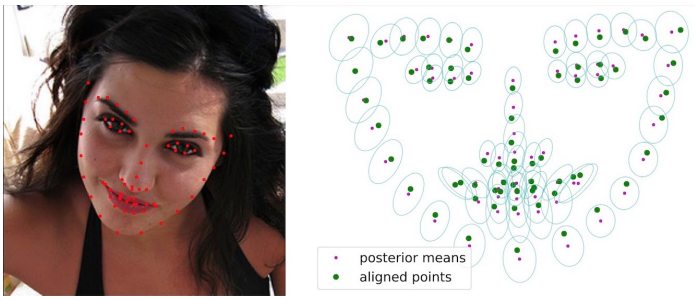
*This work is funded in part by the Multidisciplinary Institute in Artificial Intelligence (MIAI), Grenoble (ANR-19-P3IA-0003) and by the European Commission under the Horizon 2020 SPRING project (GA 871245).

i.e. [14]. Indeed, the intrinsic shape properties, such as non-rigid deformations, are preserved under rotation, translation and scale. The challenge addressed in this paper is to estimate the rigid transformation that brings the landmarks associated with all the faces in the same frontal pose (or in the same coordinate frame) such that they can be directly compared.

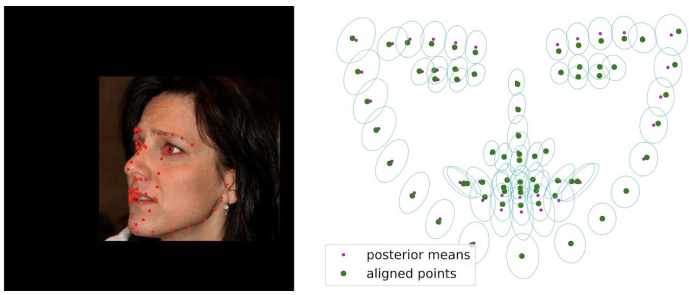
The second contribution is the construction of a *statistical shape atlas*, on the following grounds. Starting with 3D landmarks associated with a training dataset of face images with a large range of variabilities in pose, expression and identity, the proposed robust rigid-mapping estimator is used to compute a *statistical frontal landmark model* (SFL) and a *confidence score*. SFL consists of a shape atlas, namely, ellipsoidal volumes computed from the posterior means and posterior covariances of the mapped landmarks. The size of each one of these ellipsoids is estimated in such a way that its *inside* is an inlier volume with 99.7% confidence. *The built-in robust estimator downgrades the effect of large landmark errors that are inherently present in the training dataset, thus preventing the inlier volumes to grow exaggeratedly large.* The statistical shape atlas is thus conditioned by (i) the training dataset, (ii) the 3DFA method used to predict 3D landmarks, and (iii) the rigid-mapping parameters. In practice, UPA proceeds as follows. Firstly, 3D landmarks are predicted with the 3DFA architecture under investigation. Secondly, the predicted landmarks are rigidly mapped onto the shape atlas. Thirdly, a score is computed for each processed landmark, thus allowing to assess the percentage of landmarks that lie within the *confidence volumes*. The one feature of UPA-3DFA is that it can be applied to any kind of landmarks: either predicted by a 3DFA architecture/algorithm, or associated with an automatic or semi-automatic annotation process.

The third contribution is a thorough experimental evaluation that uses two publicly available datasets and five 3DFA software packages [15, 16, 17, 18, 19] associated with five peer-reviewed articles [20, 21, 12, 22, 23], respectively. The AFLW2000-3D dataset [12] and its semi-automatic annotations are used to compare UPA-3DFA with the supervised normalized mean error (NME). Correlation scores between UPA and NME are provided. This combined unsupervised-supervised analysis reveals the existence of annotation errors as well as a mechanism to disregard these errors. Altogether, this provides a novel methodological pipeline to evaluate the performance of 3DFA architectures as well as to analyse the quality of automatic and semi-automatic annotations.

The methodology is illustrated in Fig. 1 with two examples from the AFLW2000-3D dataset [12]. The statistical frontal landmark model (right) that is proposed consists of an ellipsoidal-shaped confidence volume centred at a posterior mean. Fig. 1(a): Landmarks extracted using [20] (left) are robustly mapped onto this frontal model (right). In this case, most of the landmarks lie inside their confidence volumes, thus assessing their correctness. Fig. 1(b): 3D landmarks obtained with a semi-automatic annotation process [12] are robustly mapped onto the frontal model (right). Note that several annotated landmarks fall outside the confidence volumes.



(a) 3D landmarks predicted with [20]



(b) Semi-automatic annotated 3D landmarks from [12]

Figure 1: Left: (a) landmarks predicted with the 3DFA method of [20]; (b) 3D landmarks obtained with the semi-automatic annotation method of [12]. Right: The mapped landmarks (big green dots) are overlapped onto a frontal landmark model, or a shape atlas, composed of ellipsoids centered at mean landmark locations (small red dots). This enables one to verify whether tested landmarks (left) lie within ellipsoidal-shaped volumes of confidence (right).

The remainder of this paper is organized as follows. Section 2 reviews the related work. Section 3 briefly reviews maximum likelihood estimation (MLE) for rigid mapping and describes two robust methods. Section 4 analyses the performance of the proposed rigid-mapping methods. Section 5 describes a pipeline for building a statistical face model and an associated parametric confidence metric. Section 6 presents extensive experimental results, and Section 7 draws some conclusions. The UPA-3DFA software is publicly available [24].

2. Related Work

Until recently, 3D face landmarks were extracted from 3D scans, e.g. based on rotation-invariant curvature analysis [25]. In contrast, recently proposed 3DFA methods take as input 2D images and the underlying models lie at the crossroads of deformable shape models, model-based image analysis and DNNs. DNN-based 3DFA methods use a variety of architectures in order to learn a regression function, e.g. [20, 21, 26, 22] as well as [27, 23, 28]. Given this variety, it is difficult to directly compare them and assess their merits based on the analysis of the underlying DNN concepts and methodologies. Alternatively, 3DFA algorithm performance could be measured empirically, as is often the case in deep learning.

To date, there has been a handful of 3DFA benchmarks and challenges, [29, 30, 13, 31]. The 3D face scans of the FRGC dataset [29] were used for benchmarking the rotation-invariant

curvature-based method of [25]. The authors manually annotated 12 facial landmarks. Note that this kind of annotation is possible because the 3D scans correspond to depth maps, and hence there is a depth value associated with each pixel location. Such a luxury is not possible with 2D images. In [30], four datasets were specifically gathered, annotated and prepared, and two performance metrics were used for this challenge. The BU-4DFE [32] and BP-4D-Spontaneous [33] datasets used a structured-light stereo sensor to capture textured 3D meshes of faces in controlled conditions and with various backgrounds. 2,295 meshes were selected from these datasets and manually annotated with 66 landmarks and with self-occlusion information. Then, 16,065 2D views were synthesized (seven views for each mesh) with yaw and pitch rotations ranging in the intervals $[-45°, +45°]$ and $[-30°, +30°]$, respectively. Additionally, there were 7,000 frames from the Multi-PIE [34] and 541 frames from the Time-Sliced datasets, respectively. Both these datasets contain RGB images gathered with multiple cameras from different viewpoints but with no 3D information. Therefore, a 3D face model is extracted for each image, using the model-based multi-view structure-from-motion technique of [35]. As above, each 3D face model was annotated with 66 landmarks and with self-occlusion information.

The Menpo challenge [13] is based on a dataset of 12,000 face images. In order to obtain 2D and 3D ground-truth landmarks, an automatic annotation process is proposed, which fits a 3D face model to each 2D image (see above). This fitting is carried out via non-linear minimization over the shape parameters (identity and expression), the rigid parameters (rotation and translation of the 3D model with respect to the camera), and the camera parameters.

The NoW benchmark [31] considers 3D reconstruction from a single monocular image of a face. The associated dataset contains 2,054 face images in frontal and profile views of 100 subjects and a 3D head scan for each subject. This dataset is similar in spirit with [36]. While the images contain four categories (neutral, expression, occlusion, and selfie) the 3D scans correspond to neutral faces. Therefore, the challenge is the reconstruction of a neutral 3D face from a non-neutral 2D face, implying that the latter undergoes disentanglement. Moreover, since the predicted 3D face is a mesh and the ground-truth is a 3D scan (point cloud), that lie in different coordinate frames, a rigid alignment is needed. The alignment method of [31] minimizes a scan-to-mesh distance over the scale, rotation, and translation parameters. This leads to a non-linear optimization problem. In contrast, it is proposed a rigid point-to-point alignment technique that is robust with respect to errors and that preserves the facial expressions embedded in the 3D landmarks, rather than eliminating them.

The evaluation metrics used in these benchmarks require either annotated 3D landmarks or 3D scans. As already argued, manual annotation is infeasible. Automatic annotation is based on complex non-linear minimization methods that are prone to errors and may not be reliable in the presence of profile views, of extreme expressions, and of occlusions. Localization noise is inherent. Moreover, these evaluation metrics are limited in scope since they cannot distinguish between landmark localization noise (inlying data) and large localization errors (outlying data).

In contrast, the proposed methodology doesn't require annotations of any kind. Robust rigid alignment (analyzed in detail below) is used to build a frontal landmark model, i.e. a shape atlas, in a completely unsupervised way. A statistical characterization of each landmark is provided by measuring the discrepancy between the predicted landmarks and the corresponding model landmarks. This is particularly useful to check whether the predicted 3D landmarks could be used any further, e.g. for facial expression recognition, lip reading, or head-pose estimation. In addition, the proposed analysis may well be used to remove badly located landmarks from an automatically annotated dataset.

A fundamental building block of the proposed method is a robust rigid transformation estimator. Robustness refers to the capacity of an estimator to be unaffected by large errors, i.e. outliers. For that purpose, the choice of a probability distribution function is crucial. We opt for two choices: (i) a mixture model formed by a Gaussian component and a uniform component (GUM), and (ii) the generalized Student's t-distribution (GStudent) [37, 38, 39, 40]. Robust mixture models that use a Gaussian mixture with an additional uniform component have been used for several decades in model based clustering, in order to downgrade the influence of outliers [41]. They have also been used for robust factorization [42], for point-set registration [43] and, more recently, for robust deep regression [44]. The EM algorithm proposed in [43] alternates between point registration (matching) and rigid alignment, while the uniform component of the mixture is used to disregard points in one set that don't have a match in the other set. Our use of GUM is different. Since the points (landmarks) are already registered, the residuals present in (1) (please refer to the next Section) are drawn from a GUM distribution in order to model shape-, deformation- and localization errors.

GUM and GStudent treat the above errors quite differently. GUM evaluates the posterior probability of a data point to be either an inlier or an outlier, i.e. (9) and (11). GStudent evaluates a weight w for each data point. The weights are treated as random variables drawn from a gamma distribution, i.e. (17) and they can be interpreted as precisions (the inverse of the variance): higher is a weight, more reliable is the corresponding data point. While the GUM posteriors can only vary in the range $[0; 1]$, the weight realizations vary from zero to a very large positive value. Both GUM and GStudent provide a statistically well-founded mechanism to associate a figure of merit to each data point and hence to downgrade the influence of large localization errors. This is of paramount importance, not only to properly estimate rigid parameters from 3D landmarks (Section 3), but also to build a robust confidence test (Section 5).

3. Robust Rigid Mapping

Let us consider the mapping between two sets of 3D facial landmarks, a predicted set, $\boldsymbol{x}_{1:N} = (\boldsymbol{x}_1 \dots \boldsymbol{x}_N) \in \mathbb{R}^{3\times N}$, and a model set, $\boldsymbol{y}_{1:N} = (\boldsymbol{y}_1 \dots \boldsymbol{y}_N) \in \mathbb{R}^{3\times N}$. The predicted set corresponds to a face with arbitrary and unknown pose, identity, expression and occlusion. Without loss of generality, the model set corresponds to a frontally viewed neutral face. The mapping writes:

$$\boldsymbol{y}_n = s\mathbf{R}\boldsymbol{x}_n + \boldsymbol{t} + \boldsymbol{r}_n, \forall n \in \{1, \dots, N\}, \tag{1}$$

where the rigid transformation is parameterized by a scale factor $s \in \mathbb{R}$, a rotation matrix $\mathbf{R} \in SO(3) \subset \mathbb{R}^{3\times 3}$ and a translation vector $\boldsymbol{t} \in \mathbb{R}^3$, while the non-rigid deformations and possible errors are modeled by the residuals $\boldsymbol{r}_{1:N}$. Let's cast the rigid-mapping estimation problem into the framework of maximum-likelihood estimation (MLE) with a robust pdf. It is assumed that the residuals $\boldsymbol{r}_{1:N}$ are independent and identically distributed (i.i.d). Then, the problem of estimating the rigid transformation could be solved via MLE:

$$\mathcal{L}(\boldsymbol{\theta}|(\boldsymbol{x}, \boldsymbol{y})_{1:N}) = -\sum_{n=1}^{N} \log P(\boldsymbol{r}_n; \boldsymbol{\theta}), \tag{2}$$

where $P(\boldsymbol{r}; \boldsymbol{\theta})$ is the pdf of $\boldsymbol{r}$ parameterized by $\boldsymbol{\theta}$.

3.1. Gaussian Model

The simplest statistical model is to assume that the residuals follow a zero-centered Gaussian distribution with covariance matrix $\boldsymbol{\Sigma} \in \mathbb{R}^{3\times 3}$, namely $P(\boldsymbol{r}; \boldsymbol{\theta}) = \mathcal{N}(\boldsymbol{r}; \mathbf{0}, \boldsymbol{\Sigma})$. Equation (2) yields:

$$\mathcal{L}(\boldsymbol{\theta}|(\boldsymbol{x}, \boldsymbol{y})_{1:N}) = \sum_{n=1}^{N} (\|\boldsymbol{y}_n - s\mathbf{R}\boldsymbol{x}_n - \boldsymbol{t}\|^2_{\boldsymbol{\Sigma}} + \log|\boldsymbol{\Sigma}|), \tag{3}$$

where $\|\boldsymbol{a}\|^2_{\boldsymbol{\Sigma}} = \boldsymbol{a}^\top \boldsymbol{\Sigma}^{-1} \boldsymbol{a}$ is the squared Mahalanobis norm of $\boldsymbol{a} \in \mathbb{R}^3$ and $|\cdot|$ is the determinant operator. The minimization of (3) over $\boldsymbol{t}$ yields:

$$\boldsymbol{t}^\star = \overline{\boldsymbol{y}} - s^\star \mathbf{R}^\star \overline{\boldsymbol{x}}, \tag{4}$$

where $\boldsymbol{p}^\star$ is the optimal value of a parameter $\boldsymbol{p}$, and with:

$$\overline{\boldsymbol{x}} = 1/N \sum_{n=1}^{N} \boldsymbol{x}_n, \quad \overline{\boldsymbol{y}} = 1/N \sum_{n=1}^{N} \boldsymbol{y}_n. \tag{5}$$

By substituting (4) into (3) and by using centered coordinates, i.e. $\boldsymbol{x}'_n = \boldsymbol{x}_n - \overline{\boldsymbol{x}}$, $\boldsymbol{y}'_n = \boldsymbol{y}_n - \overline{\boldsymbol{y}}$, one obtains:

$$\mathcal{L}(\boldsymbol{\theta}|(\boldsymbol{x}', \boldsymbol{y}')_{1:N}) = \sum_{n=1}^{N} (\|\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n\|^2_{\boldsymbol{\Sigma}} + \log|\boldsymbol{\Sigma}|). \tag{6}$$

Standard approaches assume an isotropic covariance, $\boldsymbol{\Sigma} = \sigma\mathbf{I}_3$, yielding a closed-form solution, e.g. [45, 46] and Appendix A. Indeed, one may easily verify that in this particular case the $s^2\boldsymbol{x}'_n\mathbf{R}\boldsymbol{\Sigma}^{-1}\mathbf{R}^\top\boldsymbol{x}'^\top_n$ term in the development of $\|\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n\|^2_{\boldsymbol{\Sigma}}$ that is present in (6) is equal to $s\sigma^{-1}\boldsymbol{x}'_n\boldsymbol{x}'^\top_n$. Nevertheless, the isotropic-covariance assumption is barely valid in practice. In the case of a full covariance, the minimization of (6) with respect to the rotation matrix yields:

$$\mathbf{R}^\star = \operatorname{argmin}_{\mathbf{R}} \operatorname{tr}(\boldsymbol{\Sigma}^{-1}(s^2\mathbf{R}\mathbf{A}\mathbf{R}^\top - 2s\mathbf{R}\mathbf{B})), \tag{7}$$

where $\operatorname{tr}(\cdot)$ is the trace operator and with the notations $\mathbf{A} = \sum_{n=1}^{N} \boldsymbol{x}'_n\boldsymbol{x}'^\top_n$, $\mathbf{B} = \sum_{n=1}^{N} \boldsymbol{x}'_n\boldsymbol{y}'^\top_n$. A rotation matrix must satisfy $\mathbf{R}\mathbf{R}^\top = \mathbf{I}_3$ and $|\mathbf{R}| = +1$. This yields a constrained non-linear optimization problem. An elegant formulation consists of parameterizing the rotation with a unit quaternion, thus reducing the number of parameters from 9 to 4, and the number of constraints from 7 to 1. Let $\mathbf{R}(\boldsymbol{q})$, where the parameter vector $\boldsymbol{q}$ is a unit quaternion, i.e. Appendix A. The constrained non-linear optimization problem, i.e. Appendix B, writes:

$$\begin{cases} \min_{\boldsymbol{q}} & \operatorname{tr}(\boldsymbol{\Sigma}^{-1}(s^2\mathbf{R}(\boldsymbol{q})\mathbf{A}\mathbf{R}(\boldsymbol{q})^\top - 2s\mathbf{R}(\boldsymbol{q})\mathbf{B})) \\ \text{s.t.} & \boldsymbol{q}^\top\boldsymbol{q} = 1 \end{cases} \tag{8}$$

3.2. Gaussian-uniform Mixture Model

Unfortunately, the above statistical model does not behave well in the presence of large residuals, or outliers. A discrete hidden random variable Z_n is associated with each residual $\boldsymbol{r}_n$, and let z be a realization of Z. $\boldsymbol{r}$ is drawn either from a zero-centered Gaussian distribution or from a multivariate uniform distribution:

$$P(\boldsymbol{r}|Z = z) = \begin{cases} \mathcal{N}(\boldsymbol{r}; \mathbf{0}, \boldsymbol{\Sigma}) & \text{if } z = \text{inlier} \\ \mathcal{U}(\boldsymbol{r}; 0, \gamma) & \text{if } z = \text{outlier,} \end{cases} \tag{9}$$

where γ is the volume of the distribution. This yields a two-component mixture model, an inlier component with prior p, and an outlier component with prior $1 - p$. Formally, this leads to solving the problem via expectation-maximization (EM) which alternates betwen (i) evaluating the posterior probabilities of the residuals, and (ii) minimizing the *expected complete-data negative log-likelihood*, $\mathrm{E}_Z[-\log P(\boldsymbol{r}_{1:N}, Z_{1:N}|\boldsymbol{r}_{1:N}; \boldsymbol{\theta})]$, where the expectation is taken over the realizations of Z, with $\boldsymbol{\theta} = \{s, \mathbf{R}, p, \boldsymbol{\Sigma}\}$.[1] This yields the minimization of:

$$\mathcal{E}(\boldsymbol{\theta}|(\boldsymbol{x}', \boldsymbol{y}')_{1:N}) = \sum_{n=1}^{N} \alpha_n(\|\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n\|^2_{\boldsymbol{\Sigma}} + \log|\boldsymbol{\Sigma}|) \tag{10}$$

where the inlier posterior $\alpha_n = P(Z = \text{inlier}|\boldsymbol{r}_n)$, is :

$$\alpha_n = \frac{p\,\mathcal{N}(\boldsymbol{r}_n; \mathbf{0}, \boldsymbol{\Sigma})}{p\,\mathcal{N}(\boldsymbol{r}_n; \mathbf{0}, \boldsymbol{\Sigma}) + (1 - p)\,\gamma^{-1}}, \tag{11}$$

and the outlier posterior is $1 - \alpha_n$. The presence of $\alpha_{1:N}$ in (10) replaces (5) with:

$$\overline{\boldsymbol{x}} = \sum_{n=1}^{N} \alpha_n\boldsymbol{x}_n / \sum_{n=1}^{N} \alpha_n, \quad \overline{\boldsymbol{y}} = \sum_{n=1}^{N} \alpha_n\boldsymbol{y}_n / \sum_{n=1}^{N} \alpha_n, \tag{12}$$

[1] Note that the translation vector $\boldsymbol{t}$ is evaluated with (4).

Data: Centered point coordinates, i.e. (5).
Normalization parameter γ ;
Initialization of $\boldsymbol{\theta}^{\text{old}}$: Use the closed-form solution [45] to evaluate s^{old} and $\mathbf{R}^{\text{old}}$ and then use these parameter values to evaluate $\boldsymbol{\Sigma}^{\text{old}}$ and set $p^{\text{old}} = 0.8$;
while $\|\boldsymbol{\theta}^{\text{new}} - \boldsymbol{\theta}^{\text{old}}\| > \epsilon$ **do**
  E-step: Evaluate the posteriors $\boldsymbol{\alpha}_{1:N}$ using (11) with $\boldsymbol{\theta}^{\text{old}}$;
  Update the centered coordinates using (12) ;
  M-scale-step: Evaluate s^{new} using (14);
  M-rotation-step: Estimate $\mathbf{R}^{\text{new}}$ via constrained non-linear optimization of (8) using (13) ;
  M-covariance-step: Evaluate $\boldsymbol{\Sigma}^{\text{new}}$ using (16);
  M-prior-step: Evaluate p^{new} using (15);
  $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$;
end
Result: Estimated scale $s^\star$, rotation $\mathbf{R}^\star$, translation $\boldsymbol{t}^\star$ (4), prior $p^\star$, covariance $\boldsymbol{\Sigma}^\star$, and posterior probabilities of landmarks $\boldsymbol{\alpha}_{1:N}$.

Algorithm 1: GUM Expectation-Maximization.

as well as $\mathbf{A}$ and $\mathbf{B}$ from (8) with

$$\mathbf{A} = \sum_{n=1}^{N} \alpha_n \boldsymbol{x}'_n \boldsymbol{x}'^{\top}_n, \quad \mathbf{B} = \sum_{n=1}^{N} \alpha_n \boldsymbol{x}'_n \boldsymbol{y}'^{\top}_n. \tag{13}$$

Hence, (8) can be used to estimate the optimal rotation while the optimal scale is estimated with:

$$s^\star = \left(\frac{\sum_{n=1}^{N} \alpha_n \boldsymbol{y}'^{\top}_n \boldsymbol{\Sigma}^{-1} \boldsymbol{y}'_n}{\sum_{n=1}^{N} \alpha_n (\mathbf{R}^\star \boldsymbol{x}'_n)^{\top} \boldsymbol{\Sigma}^{-1} \mathbf{R}^\star \boldsymbol{x}'_n} \right)^{1/2}. \tag{14}$$

The prior p and covariance $\boldsymbol{\Sigma}$ are estimated with:

$$p = \frac{1}{N} \sum_{n=1}^{N} \alpha_n, \tag{15}$$

$$\boldsymbol{\Sigma}^\star = \frac{\sum_{n=1}^{N} \alpha_n (\boldsymbol{y}'_n - s^\star \mathbf{R}^\star \boldsymbol{x}'_n)(\boldsymbol{y}'_n - s^\star \mathbf{R}^\star \boldsymbol{x}'_n)^{\top}}{\sum_{n=1}^{N} \alpha_n}. \tag{16}$$

This model is referred to as the *Gaussian-uniform mixture* (GUM) and the associated EM is summarized in Algorithm 1.

3.3. Generalized Student Model

Another way to enforce robustness is to use the *generalized Student's t-distribution*, also known as the Pearson type VII distribution [38]:

$$\begin{aligned} P(\boldsymbol{r}; \boldsymbol{\Sigma}, \mu, \nu) &= \int_0^\infty \mathcal{N}(\boldsymbol{r}; 0, w^{-1}\boldsymbol{\Sigma}) \mathcal{G}(w, \mu, \nu) dw \\ &= \frac{\Gamma(\mu + \frac{3}{2})}{|\boldsymbol{\Sigma}|^{\frac{1}{2}} \Gamma(\mu) (2\pi\nu)^{\frac{3}{2}}} \left(1 + \frac{\|\boldsymbol{r}\|^2_{\mathcal{M}}}{2\nu}\right)^{-(\mu+\frac{3}{2})}, \end{aligned} \tag{17}$$

where $\Gamma(\cdot)$ is the gamma function. Note that the *latent* weight variables $w_{1:N}$ are drawn from a prior gamma distribution with parameters μ and ν. In (17) ν and $\boldsymbol{\Sigma}$ appear only through their product, which means that an additional constraint is required to make the parameterization unique. Let $\nu = 1$ as in [39]. The posterior distribution of w_n is also a gamma distribution, namely the *posterior gamma distribution*:

$$\begin{aligned} P(w_n | \boldsymbol{r}_n; \boldsymbol{\Sigma}, \mu, \nu) &= \mathcal{N}(\boldsymbol{r}_n; 0, w_n^{-1}\boldsymbol{\Sigma}) \mathcal{G}(w_n, \mu, \nu) \\ &= \mathcal{G}(w_n; a, b_n), \end{aligned} \tag{18}$$

with parameters:

$$a = \mu + 3/2, \quad b_n = 1 + \|\boldsymbol{r}_n\|^2_{\boldsymbol{\Sigma}}/2. \tag{19}$$

The posterior mean of the weight variable is:

$$\overline{w}_n = \mathrm{E}[w_n | \boldsymbol{r}_n] = a/b_n. \tag{20}$$

As above, one needs to minimize the expected complete-data negative log-likelihood, $\mathrm{E}_W[-\log P(\boldsymbol{r}_{1:N}, W_{1:N} | \boldsymbol{r}_{1:N}; \boldsymbol{\theta})]$ with $\boldsymbol{\theta} = \{s, \mathbf{R}, \boldsymbol{\Sigma}, \mu\}$, yielding the minimization:

$$\mathcal{Q}(\boldsymbol{\theta} | (\boldsymbol{x}', \boldsymbol{y}')_{1:N}) = \sum_{n=1}^{N} (\overline{w}_n \|\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n\|^2_{\boldsymbol{\Sigma}} + \log|\boldsymbol{\Sigma}|), \tag{21}$$

thus replacing $\alpha_{1:N}$ with $w_{1:N}$ in (12) and (13) to estimate the optimal rotation (8) and scale (14). The covariance matrix is estimated with:

$$\boldsymbol{\Sigma} = \sum_{n=1}^{N} \overline{w}_n (\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n)(\boldsymbol{y}'_n - s\mathbf{R}\boldsymbol{x}'_n)^{\top}/N \tag{22}$$

The parameter μ is updated by solving the following equation, where $\Psi(\cdot)$ is the digamma function:

$$\mu = \Psi^{-1}\left(\Psi(a) - \frac{1}{n}\sum_{n=1}^{N} \log b_n\right). \tag{23}$$

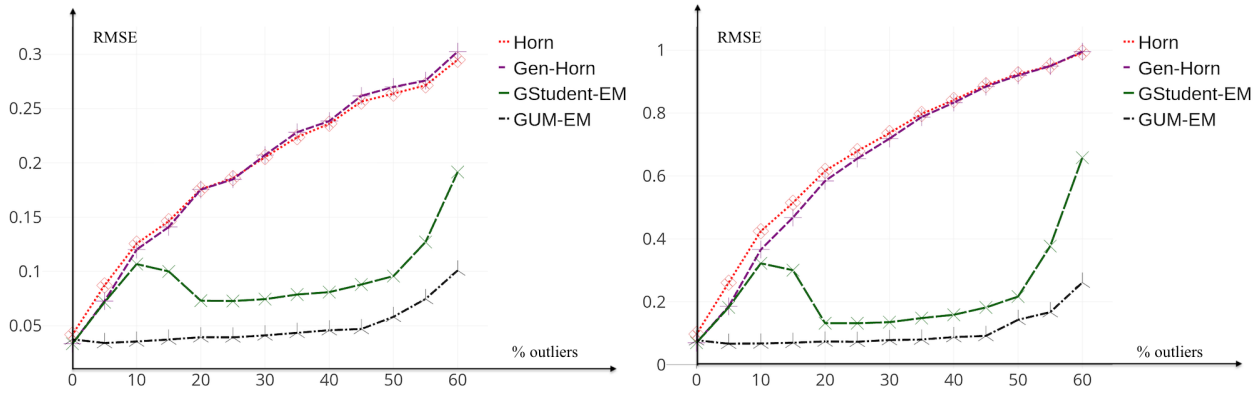
This model is referred to as the *generalized Student* (GStudent) and the associated EM algorithm is summarized in Algorithm 2.

4. Analyzing the Robustness of Rigid Mapping

In order to quantify the performance of the proposed robust rigid-mapping algorithms, An experimental protocol is devised on the following grounds. As above:

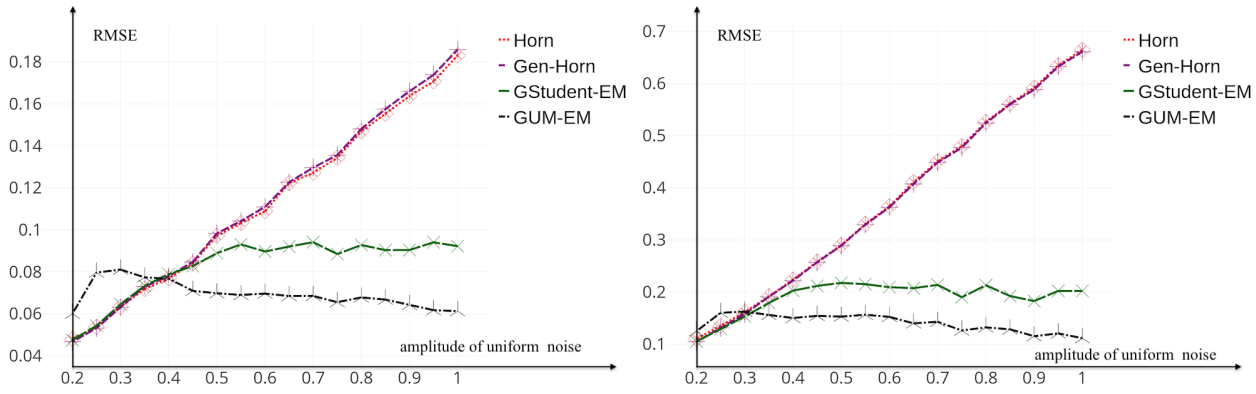
$$\boldsymbol{y}^m_n(b) = s^m \mathbf{R}^m \boldsymbol{x}_n + \boldsymbol{t}^m + \boldsymbol{r}^m_n(b), \forall n \in \{1, \ldots, N\}, \tag{24}$$

where $b > 0$ is a scalar that controls the level of noise and m is the trial index. As described in detail below, the noise level, b can be the total variance of Gaussian anisotropic noise, or the volume of uniformly distributed noise. The image coordinates are normalized such that $\forall n, \boldsymbol{x}_n \in [0, 1]^3$. For each noise level, M trials are randomly generated, namely M rigid mappings and M sets of N residuals $\boldsymbol{r}_{1:N} = \{\boldsymbol{r}_n\}_{n=1}^{n=N}$. For each trial m the rigid mapping parameters are estimated, s^m, $\mathbf{R}^m$, $\boldsymbol{t}^m$, and the *root*



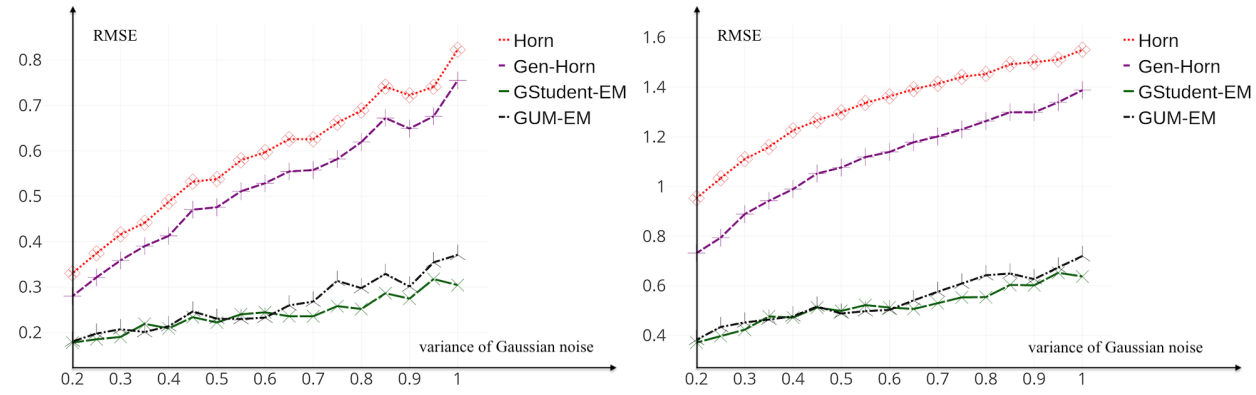
(a) Rotation error. (b) Translation error.

Figure 2: RMSE as a function of the percentage of outliers (0% to 60%): inliers are affected by Gaussian noise with $\lambda = 0.0025$ while outliers are affected by uniform noise with amplitude $a = 1.5$.



(a) Rotation error. (b) Translation error.

Figure 3: RMSE as a function of uniform noise affecting a fixed number of outliers: inliers (50%) are affected by Gaussian noise with $\lambda = 0.0025$ while outliers (50%) are affected by uniform noise of increasing amplitude $a \in [0.2, 1.0]$.



(a) Rotation error. (b) Translation error.

Figure 4: RMSE as a function of Gaussian noise affecting the outliers: inliers (50%) are affected by Gaussian noise with $\lambda = 0.0025$ while outliers (50%) are affected by Gaussian noise with $\lambda \in [0.2, 1.0]$.

Data: Centered point coordinates, i.e. (5). ;
Initialization of $\boldsymbol{\theta}^{\text{old}}$: Use the closed-form solution [45] to evaluate s^{old} and $\mathbf{R}^{\text{old}}$; evaluate $\boldsymbol{\Sigma}^{\text{old}}$. Provide μ^{old} ;
while $\|\boldsymbol{\theta}^{\text{new}} - \boldsymbol{\theta}^{\text{old}}\| > \epsilon$ **do**
  E-step: evaluate a^{new} and $\boldsymbol{b}_{1:N}^{\text{new}}$ using (19) with $\boldsymbol{\theta}^{\text{old}}$, then evaluate $\overline{\boldsymbol{w}}_{1:N}^{\text{new}}$ using (20) ;
  Update the centered coordinates using (12), where $\alpha_{1:N}$ are replaced with $\overline{w}_{1:N}$;
  M-scale-step: Evaluate s^{new} using (14);
  M-rotation-step: Estimate $\mathbf{R}^{\text{new}}$ with (8), (13) ;
  M-covariance-step: Evaluate $\boldsymbol{\Sigma}^{\text{new}}$ using (22) ;
  M-mu-step: Evaluate μ^{new} using (23) ;
  $\boldsymbol{\theta}^{\text{old}} \leftarrow \boldsymbol{\theta}^{\text{new}}$;
end
Result: Estimated scale $s^{\star}$, rotation $\mathbf{R}^{\star}$, translation $\boldsymbol{t}^{\star}$ (4), covariance $\boldsymbol{\Sigma}^{\star}$, and landmark weights $w_{1:N}$.

Algorithm 2: GStudent Expectation-Maximization.

mean square error, (RMSE) between these estimated parameters and the ground-truth parameters $\tilde{s}^m$, $\tilde{\mathbf{R}}^m$, $\tilde{\boldsymbol{t}}^m$, is estimated, namely:

$$E_s = \left(1/M \sum_{m=1}^{M} |s^m - \tilde{s}^m|^2\right)^{1/2}, \tag{25}$$

$$E_{\boldsymbol{t}} = \left(1/M \sum_{m=1}^{M} \|\boldsymbol{t}^m - \tilde{\boldsymbol{t}}^m\|^2\right)^{1/2}, \tag{26}$$

$$E_{\mathbf{R}} = \left(1/M \sum_{m=1}^{M} \|\mathbf{R}^m - \tilde{\mathbf{R}}^m\|^2\right)^{1/2}, \tag{27}$$

The ground-truth rigid-mapping parameters are generated in the following way. For each trial m, the scale and the translation vector are generated from uniform distributions, namely $s^m \sim \mathcal{U}(0.5, 2)$ and $\boldsymbol{t}^m \sim \mathcal{U}(0.5, 5)^3$. The rotation matrix is parameterized by the pan, tilt and yaw angles, namely, $\mathbf{R} = \mathbf{R}_\gamma \mathbf{R}_\phi \mathbf{R}_\psi$. A rotation matrix is obtained by randomly generating the pan, tilt and yaw angles, γ^m, ϕ^m, ψ^m, from a uniform distribution, $\mathcal{U}(-90°, +90°)$.

In order to generate residuals, $\boldsymbol{r}_{1:N}$, Gaussian and uniform noise are simulated, namely $\boldsymbol{r} \sim \mathcal{N}(\mathbf{0}, \boldsymbol{\Sigma})$ and $\boldsymbol{r} \sim \mathcal{U}(-a/2, a/2)^3$. In the Gaussian case, a covariance matrix must be randomly generated for each trial. This is done in the following way. Let $\boldsymbol{\Sigma} = \mathbf{Q}\boldsymbol{\Lambda}\mathbf{Q}^\top$, with $\mathbf{Q} \in O(3)$ and with $\boldsymbol{\Lambda} = \text{Diag}(\lambda_1, \lambda_2, \lambda_3)$. Let $\lambda = \lambda_1 + \lambda_2 + \lambda_3$, i.e. the total variance. A sample covariance $\boldsymbol{\Sigma}$ is simulated by randomly generating an orthogonal matrix $\mathbf{Q}$ as well as three eigenvalues from a uniform distribution, $\mathcal{U}(0, 1)$.

The following rigid mapping models and associated algorithms were tested:

- *Horn*: Gaussian distribution with isotropic covariance, [45] and Appendix A;
- *Gen-Horn*: Gaussian distribution with anisotropic covariance, Section 3.1;
- *GUM-EM*: Gaussian-uniform mixture distribution, Algorithm 1, and
- *GStudent-EM*: Generalized Student's t-distribution, Algorithm 2.

The experiments were conducted in the following way. For each noise level, $M = 500$ trials were simulated and the RMSEs were computed, namely eqs. (25), (26), and (27). For each trial m the landmarks were split into an inlier set and an outlier set and the $N = 68$ landmarks are randomly assigned to one of these sets. The first experiment determines the percentage of outliers that can be handled by the robust algorithms, Figure 2. For this purpose, the percentage of outliers is increased from 10% to 60%. The inlier noise is drawn from an anisotropic Gaussian distribution with a total variance $\lambda = 0.0025$. The outlier noise is drawn from a uniform distribution with amplitude $a = 1.5$ (remember that the landmark coordinates are normalized to lie in the interval $[0, 1]$). The curves plotted in Figure 2 show that the RMSE associated with non robust methods, i.e. Horn and Gen-Horn increase monotonically. On the contrary, the robust algorithms, GUM-EM and GStudent-EM, have a radically different behavior. After a short increase, the RMSE remains constant, and then it increases again.

In the remaining experiments, the number of inliers was set to be equal to the number of outliers. Figure 3 shows the RMSEs for the case when inlier noise is drawn from an anisotropic Gaussian distribution with total variance $\lambda = 0.0025$, while outlier noise is drawn from a uniform distribution whose volume is increased from $a = 0.2$ to $a = 1.0$. Finally, Figure 4 shows the RMSEs when inlier noise is drawn from an anisotropic Gaussian distribution with total variance $\lambda = 0.0025$, while outlier noise is drawn from an anisotropic Gaussian distribution with total variance varying from $\lambda = 0.2$ to $\lambda = 1.0$.

These experiments clearly show that GUM and GStudent can deal with up to 50% landmarks affected by a substantial noise level (1.5 times the size of the image). The posteriors (GUM) and the weights (GStudent), estimated by Algorithms 1 and 2, respectively, characterize the observed landmarks and reduce the importance of those landmarks that have large errors in localization. As can be seen from (11), the GUM posteriors are in the range $0 < \alpha < 1$ and consequently their values are a relative measure of the importance of the landmarks. In contrast, the GStudent weights (20) are in the range $0 < w < \infty$, hence they constitute an absolute measure. Altogether, this offers a valuable framework for building a statistical landmark model from a training dataset.

5. 3DFA Performance Analysis

In this section we describe an unsupervised methodology for quantitatively assessing the performance of 3DFA algorithms. The idea is to apply 3DFA to a dataset of face images in order to extract 3D landmarks, to robustly estimate the rigid transformation that maps these facial landmarks onto a 3D landmark model, and to analyze the discrepancy between the extracted-and-mapped 3D landmarks and the model. Based on a confidence score, it is then possible to decide whether a landmark

is correctly localized. This allows to assess the overall performance of a 3DFA algorithm as well as its behavior with respect to various sources of perturbations.

5.1. Neutral Frontal Landmark Model

Let's start by computing a *neutral frontal landmark* (NFL) model, $\boldsymbol{y}_{1:N}$, in the following way. A dataset $\mathcal{D}_1$ of K images of neutral faces (frontal viewing, no expression and no interfering object causing occlusion) is collected and N landmarks are extracted from each one of these K faces, $\{\boldsymbol{y}_{1:N,k}\}_{k=1}^{k=K}$. The directions of maximum variance are then computed for each face as to align the faces using these directions. Next, mean coordinates for each landmark n are computed, namely:

$$\boldsymbol{y}_n = \frac{1}{K}\sum_{k=1}^{K}\boldsymbol{y}_{n,k}, \forall n, 1 \leq n \leq N. \tag{28}$$

The NFL model was created in-the-wild by harvesting web images and using the face detector of [47] and the head-pose estimator of [48] in order to select frontal faces. These images were then visually inspected one-by-one to guarantee shape and aspect variabilities as well as neutral facial expressions.This process yields a dataset $\mathcal{D}_1$ composed of $K = 1,000$ images. The 3DFA method of [20] was used to extract landmarks from each face in the dataset. Once aligned, a mean for each landmark is computed, using (28). Figure 5 shows a few examples of images from the $\mathcal{D}_1$ dataset, the detected landmarks using [21], and a 3D view of the neutral frontal landmark model.

5.2. Statistical Frontal Landmark Model

Now it is explained how a *statistical frontal landmark* (SFL) model is built, namely $\{\boldsymbol{p}_{1:N}, \mathbf{C}_{1:N}\}$, where $\boldsymbol{p}_{1:N}$ are posterior means and $\mathbf{C}_{1:N}$ are posterior covariance matrices associated with this model. The means and covariances are estimated in the following way. A second dataset $\mathcal{D}_2$ is considered, namely the YawDD dataset [49]. The faces in this dataset have large variabilities in terms of face shapes, face aspects, head poses and facial expressions, and with no external sources of perturbation such as the presence of interfering objects that may cause occlusions. First, 3D landmarks are extracted from these face images (see below), namely $\{\boldsymbol{x}_{1:N,l}\}_{l=1}^{l=L}$, using either GUM-EM (Algorithm 1) or GStudent-EM (Algorithm 2) to robustly estimate the rigid transformations between each landmark-set $\boldsymbol{x}_{1:N,l}$ and the the NFL model $\boldsymbol{y}_{1:N}$. Based on this, L mapping parameters are obtained (one for each face l): L scale factors, L rotations and L translations: $\{s_l^{\text{Alg}}, \mathbf{R}_l^{\text{Alg}}, \boldsymbol{t}_l^{\text{Alg}}\}_{l=1}^{L}$, where the overscript Alg denotes a robust algorithm, namely either GUM-EM or GStudent-EM. It is reminded that both algorithms provide a figure of merit characterizing each landmark: posterior probabilities $\{\alpha_{n,l}\}_{n=1}^{n=N}$ in the case of GUM-EM, i.e. (11), and posterior weight means $\{\overline{w}_{n,l}\}_{n=1}^{n=N}$ in the case of GSudent-EM, i.e. (20). Applying one of these robust rigid-alignment methods provides frontal landmarks, $\{\widetilde{\boldsymbol{x}}_{1:N,l}^{\text{Alg}}\}_{l=1}^{l=L}$, namely:

$$\widetilde{\boldsymbol{x}}_{n,l}^{\text{Alg}} = s_l^{\text{Alg}}\mathbf{R}_l^{\text{Alg}}\boldsymbol{x}_{n,l} + \boldsymbol{t}_l^{\text{Alg}}. \tag{29}$$

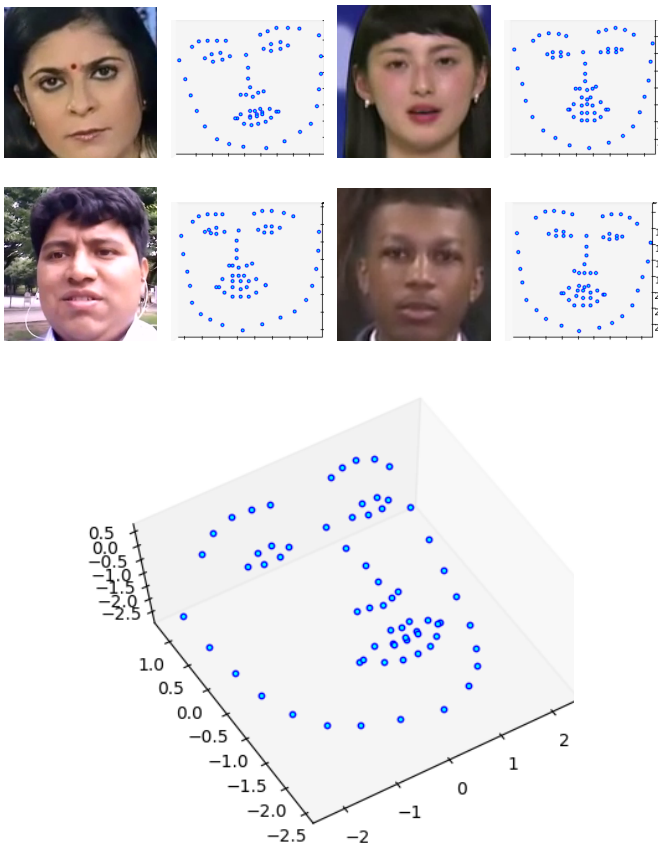


Figure 5: Examples of faces and corresponding 3D landmarks (top) used to compute a neutral frontal landmark model (bottom).

One can now build first- and second-order statistics with the use of posterior probabilities (GUM-EM) and posterior weights (GStudent-EM), respectively:

$$\boldsymbol{p}_n^{\text{GUM}} = \sum_{l=1}^{L}\alpha_{n,l}\widetilde{\boldsymbol{x}}_{n,l}^{\text{GUM}} / \sum_{l=1}^{L}\alpha_{n,l}, \tag{30}$$

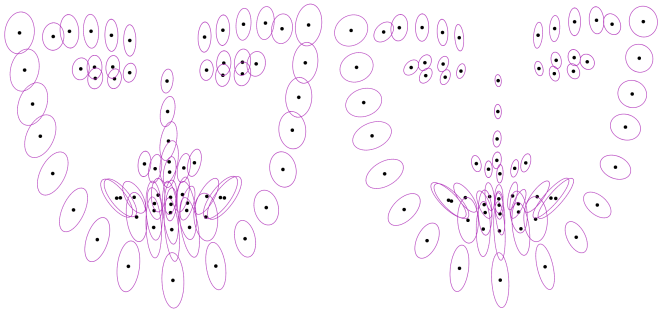
$$\mathbf{C}_n^{\text{GUM}} = \frac{\sum_{l=1}^{L}\alpha_{n,l}(\widetilde{\boldsymbol{x}}_{n,l}^{\text{GUM}} - \boldsymbol{p}_n^{\text{GUM}})(\widetilde{\boldsymbol{x}}_{n,l}^{\text{GUM}} - \boldsymbol{p}_n^{\text{GUM}})^\top}{\sum_{l=1}^{L}\alpha_{n,l}}, \tag{31}$$

and

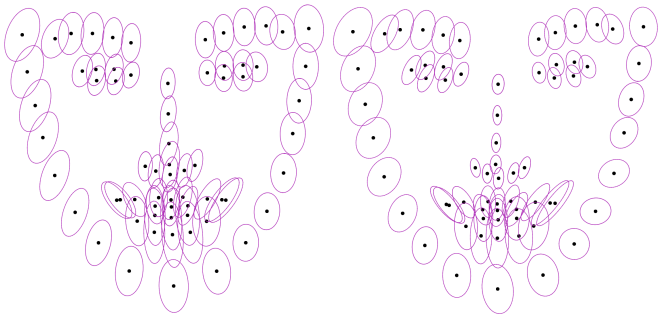
$$\boldsymbol{p}_n^{\text{GSt}} = \sum_{l=1}^{L}\overline{w}_{n,l}\widetilde{\boldsymbol{x}}_{n,l}^{\text{GSt}} / \sum_{l=1}^{L}\overline{w}_{n,l}, \tag{32}$$

$$\mathbf{C}_n^{\text{GSt}} = \frac{\sum_{l=1}^{L}\overline{w}_{n,l}(\widetilde{\boldsymbol{x}}_{n,l}^{\text{GSt}} - \boldsymbol{p}_n^{\text{GSt}})(\widetilde{\boldsymbol{x}}_{n,l}^{\text{GSt}} - \boldsymbol{p}_n^{\text{GSt}})^\top}{\sum_{l=1}^{L}\overline{w}_{n,l}}. \tag{33}$$

The YawDD dataset [49] contains 342 videos of 107 participants. The videos were recorded at 30 FPS and each video lasts between 15 and 40 seconds, which is equivalent to $L = 300,000$ images. All the images were processed with no human intervention, namely: face detection [47], 3D face alignment, and



(a) 3DFA1/GUM-EM (b) 3DFA2/GUM-EM



(c) 3DFA1/GStudent-EM (d) 3DFA2/GStudent-EM

Figure 6: The four SFL models obtained with two 3DFA methods and with the proposed robust algorithms. The figure shows the image projections of these 3D models.

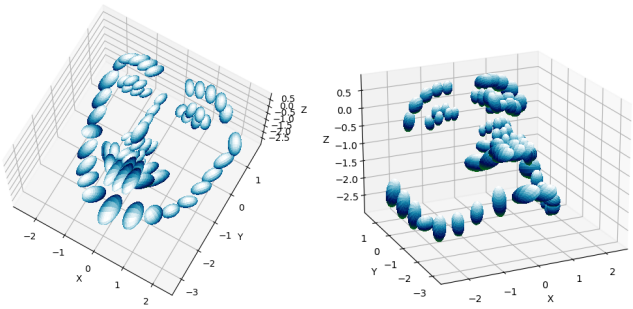


Figure 7: Two 3D views of the 3DFA1/GStudent-EM SFL model, i.e. displayed in Fig. 6(c).

robust rigid alignment with the NFL model just described. This yields the SFL model described above. For that purpose two 3DFA methods were used and the two robust alignment algorithms described in this paper. Hence, there are four possible 3DFA and robust alignment combinations used to train four different models:

1. 3DFA1/GUM-EM: [20] and Algorithm 1,
2. 3DFA2/GUM-EM: [21] and Algorithm 1,
3. 3DFA1/GStudent-EM: [20] and Algorithm 2, and
4. 3DFA2/GStudent-EM: [21] and Algorithm 2.

Figure 6 shows the SFL models obtained with these four combinations. In this figure, the dots correspond to the posterior means, i.e. (30) and (32), while the ellipses correspond to image projections of the ellipsoids defined by (36). Figure 7 shows the 3D ellipsoids associated with the 3DFA1/GStudent-EM model. Each one of these models may well be viewed as a shape atlas.

5.3. Unsupervised Confidence Test

Let us now develop an unsupervised (statistical) confidence test for assessing whether the accuracy of a landmark, i.e. its 3D coordinates, is within (inlier) or outside (outlier) an expected range [50, 51]. Let us drop the algorithm over-script and let $\mathbf{C}_n = \mathbf{Q}_n \mathbf{\Lambda}_n \mathbf{Q}_n^\top$ be the eigen factorization of $\mathbf{C}_n$, where $\mathbf{Q}_n$ is an orthonormal matrix and $\mathbf{\Lambda}_n$ is a diagonal matrix containing the eigenvalues. One can now project each landmark (n, l) onto the space spanned by the three eigenvectors of this matrix:

$$\widetilde{\boldsymbol{z}}_{n,l} = \mathbf{Q}_n^\top (\widetilde{\boldsymbol{x}}_{n,l} - \boldsymbol{p}_n). \tag{34}$$

Landmark (n, l) is an inlier with 99.7% confidence if $\widetilde{\boldsymbol{z}}_{n,l}$ lies inside the ellipsoid whose half-axes are three times the standard deviations, or $3\sqrt{\lambda_n^1}$, $3\sqrt{\lambda_n^2}$, $3\sqrt{\lambda_n^3}$, where $\{\lambda_n^1, \lambda_n^2, \lambda_n^3\}$ are the eigenvalues of $\mathbf{C}_n$, or

$$\widetilde{\boldsymbol{z}}_{n,l}{}^\top \widetilde{\mathbf{\Lambda}}_n^{-1} \widetilde{\boldsymbol{z}}_{n,l} \leq 1, \tag{35}$$

where $\widetilde{\mathbf{\Lambda}}_n = 9\mathbf{\Lambda}_n$. Combining (34) and (35), yields $(\widetilde{\boldsymbol{x}}_{n,l} - \boldsymbol{p}_n)^\top \mathbf{Q}_n \widetilde{\mathbf{\Lambda}}_n^{-1} \mathbf{Q}_n^\top (\widetilde{\boldsymbol{x}}_{n,l} - \boldsymbol{p}_n) \leq 1$. With the notation

$$\widetilde{\mathbf{C}}_n = \mathbf{Q}_n \widetilde{\mathbf{\Lambda}}_n \mathbf{Q}_n^\top. \tag{36}$$

The 99.7% confidence test writes:

$$\begin{cases} \text{if} \quad \|\widetilde{\boldsymbol{x}}_{n,m} - \boldsymbol{p}_n\|_{\widetilde{\mathbf{C}}_n} \leq 1 & (n,m) = \text{inlier} \\ \text{otherwise} & (n,m) = \text{outlier}, \end{cases} \tag{37}$$

Based on this confidence test, one can now build an unsupervised *confidence score* (the higher the better) associated with a sample face m, namely:

$$\mathrm{u}(m) = \frac{1}{N} \sum_{n=1}^{N} \mathcal{I}(\|\widetilde{\boldsymbol{x}}_{n,l} - \boldsymbol{p}_n\|_{\widetilde{\mathbf{C}}_n} \leq 1), \tag{38}$$

where $\mathcal{I}(\cdot)$ denotes the indicator function. Notice that (38) corresponds to the percentage of inliers, i.e. landmarks that, once scaled, rotated and translated, lie inside the confidence volume. Therefore, (38) can be used to assess whether the pose has been correctly estimated, namely $u \leq 0.5$, or not. Indeed, Section 4 empirically shows that both the GStudent-EM and GUM-EM yield accurate rigid parameters in the presence of up to 50% outliers. One may use the value of the Mahalanobis distance to quantify the degree of confidence, or to increase (or decrease) the size of the ellipsoid in order to provide looser (or stricter) inlier/outlier decisions. For a test dataset $\mathcal{D}_3$ composed of M samples, one can then compute the unsupervised *mean confidence score* (MCS) over the entire dataset:

$$\mathrm{U} = \frac{1}{M} \sum_{m=1}^{M} \frac{1}{N} \sum_{n=1}^{N} \mathcal{I}(\|\widetilde{\boldsymbol{x}}_{n,l} - \boldsymbol{p}_n\|_{\widetilde{\mathbf{C}}_n} \leq 1). \tag{39}$$

It should be noted that this unsupervised confidence test is based on the covariance (36) associated with the SFL model. The covariance thus computed characterizes small-amplitude noise (inliers) generated by several processes, such as shape variabilities, non-rigid deformations and landmark localization noise. These are indistinguishable by our model. Nevertheless, the proposed performance-analysis method is able to detect large localization errors with 99.7% confidence. Therefore, it is useful to detect outliers associated with 3D face alignment architectures.

5.4. Supervised Metrics

Whenever a face dataset comes with annotations, one can use the *normalized mean error* (NME) between the detected landmarks and the annotations, as a standard supervised performance measure [52]. The NME associated with face m is defined as follows:

$$\mathrm{NME}(m) = \frac{1}{N}\sum_{n=1}^{N} \|\boldsymbol{x}_{n,m} - \hat{\boldsymbol{x}}_{n,m}\|/d_m, \tag{40}$$

where d_m is a normalization factor, which could be the 2D or 3D distance between the eye centers, the size of the face- or of the landmark bounding-box. For a set of M faces, one defines the cumulated error distribution (CED) as a function of ε (a user-defined parameter):

$$\mathrm{CED}(\varepsilon) = \frac{1}{M}\sum_{m=1}^{M} \mathcal{I}\Big(\mathrm{NME}(m) \leq \varepsilon\Big). \tag{41}$$

For supervised performance analysis, it is common to use the area under curve (AUC) of CED up to a value of ε. Analogous to the unsupervised metric (38), one now defines supervised metrics that count the proportion of inliers (higher the better), for a face m and for M faces:

$$\mathrm{s}(m) = \frac{1}{N}\sum_{n=1}^{N} \mathcal{I}(\|\boldsymbol{x}_{n,m} - \hat{\boldsymbol{x}}_{n,m}\|/d_m \leq \varepsilon), \tag{42}$$

$$\mathrm{S} = \frac{1}{M}\sum_{m=1}^{M} s(m). \tag{43}$$

5.5. Unsupervised-Supervised Correlation

Another interesting metric is the correlation between the unsupervised and supervised metrics. The proposed pipeline is used to reject erroneous annotations:

$$\mathcal{M}_\tau = \{m \mid \hat{u}(m) \geq \tau\}, \quad \mathcal{M}_\tau \subset \mathcal{D}_3, \tag{44}$$

where $\hat{u}(m)$ is the score (38) for the annotations of face m. The percentage of annotation inliers present in $\mathcal{M}_\tau$ increases with τ, at the price of drastically decreasing the number of inlier samples, which in turn lowers down the statistical significance of the resulting scores. The correlation is:

$$\mathrm{Cor}(\tau) = \frac{\sum\limits_{m\in\mathcal{M}_\tau} (\mathrm{u}(m) - \mathrm{U})(\mathrm{s}(m) - \mathrm{S})}{\Big(\sum\limits_{m\in\mathcal{M}_\tau} (\mathrm{u}(m) - \mathrm{U})^2 \sum\limits_{m\in\mathcal{M}_\tau} (\mathrm{s}(m) - \mathrm{S})^2\Big)^{\frac{1}{2}}} \tag{45}$$

6. Experimental Results

Once the NFL and SFL models are computed using datasets $\mathcal{D}_1$ and $\mathcal{D}_2$, respectively, we use a third dataset, $\mathcal{D}_3$, to empirically assess the performance of five 3DFA algorithms based on the supervised, unsupervised and correlation metrics just described. Table 1 lists the methods that are analysed.

For this purpose AFLW2000-3D [12] is used as $\mathcal{D}_3$, consisting of 2000 images with large pose variations: the yaw angles (vertical axis of rotation) are in the interval $[0°, \pm 30°]$ for 1306 faces, in $[\pm 30°, \pm 60°]$ for 462 faces and in $[\pm 60°, \pm 90°]$ for 232 faces. The dataset contains a large variety of identities, expressions and illumination conditions. Moreover, there are many faces with partial occlusions caused by the presence of hair, hands, glasses, etc. Notice that large poses induce self occlusions.

Each image in AFLW2000-3D is annotated with 68 3D landmarks. The annotation is performed by fitting a 3DMM to 2D facial landmarks [12]. Failures of this automatic annotation process are manually corrected, hence it is a semi-automatic annotation.[2] As noted in [52], in spite of manual correction, annotation errors are still present, especially in the case of profile views. Hence, performance evaluation based on supervised metrics is likely to be biased by these annotation errors. In [52] it is qualitatively (visually) shown that in these extreme poses their 3DFA method yields more precise landmark localization results than the semi-automatic annotations. The proposed unsupervised metric is one possible way to quantify the results of [52]. In an attempt to analyse the semi-automatic annotation process itself, the proposed unsupervised metrics was applied to the annotated landmarks (ANN) of [12], yielding two combinations: ANN/GUM and ANN/GSt.

The results that were obtained based on computing MCS, i.e. (39) are summarized in Table 2. It is reminded that $\mathcal{D}_1$ and $\mathcal{D}_2$ were used to compute the NFL model and to train the SFL model, respectively, and $\mathcal{D}_3$ to test their performance. The average scores obtained with the annotated landmarks (the last two rows and last column of Table 2), are equal to 0.70 and to 0.65, respectively, which seems to confirm that the semi-automatic

[2] Please consult https://openaccess.thecvf.com/content_cvpr_2016/supplemental/Zhu_Face_Alignment_Across_2016_CVPR_supplemental.pdf.

Table 1: 3D face alignment methods that are analysed, the corresponding citations, and the website of the associated software packages that are publicly available.

Method	References	Code/year
3DFA1	[52] (ICCV'17)	[15]/2020
3DFA2	[21] (ECCV'18)	[16]/2018
3DFA3	[12] (CVPR'18), [26] (PAMI'19)	[17]/2019
3DFA4	[22] (IEEE TMM'21)	[18]/2019
3DFA5	[23] (ECCV'20)	[19]/2021

Table 2: Performance analysis based on MCS computed with (39). The numbers correspond to the proportion of inliers (the higher the better) over the AFLW2000-3D dataset, that contains 2,000 face images and 68 landmarks per face. The last two rows show the results of applying the proposed unsupervised metric to the 3D annotations obtained via a semi-automatic process described in [12].

Test using $\mathcal{D}_3$	Neutral/statistical models computed/trained with $\mathcal{D}_1/\mathcal{D}_2$ datasets:				
3DFA#/Method:	[52]/GUM	[21]/GUM	[52]/GStudent	[21]/GStudent	Mean
3DFA1/GUM	0.89	0.65	0.93	0.80	0.82
3DFA2/GUM	0.93	0.88	0.95	0.93	0.92
3DFA3/GUM	**0.98**	**0.96**	**0.98**	**0.98**	**0.98**
3DFA4/GUM	0.11	0.06	0.16	0.12	0.11
3DFA5/GUM	**0.98**	0.95	**0.98**	0.97	0.97
3DFA1/GStudent	0.80	0.57	0.88	0.74	0.75
3DFA2/GStudent	0.84	0.76	0.90	0.88	0.84
3DFA3/GStudent	0.93	0.85	0.96	0.94	0.92
3DFA4/GStudent	0.18	0.12	0.23	0.18	0.18
3DFA5/GStudent	0.94	0.86	0.97	0.95	0.93
ANN/GUM	0.73	0.54	0.82	0.71	0.70
ANN/GStudent	0.67	0.48	0.78	0.66	0.65

annotations available with AFLW2000-3D contain a substantial amount of errors and that the 3DFA methods [20] and [21], used for training, predict landmark locations that are more accurate than the annotated locations themselves.

The performance of both 2D and 3D face alignments are analyzed in terms of the supervised metric (43), where for the 2D case we simply discard depth. In order to remove any bias possibly due to the prediction of the z coordinate, the performance is also compared by centering this coordinate for all the landmarks of each face. And finally, several NME normalizations are considered (see above). The results are presented in Figure 8. The AUC, NME mean and NME standard deviation, corresponding to Figure 8(d)&(f), are reported in Table 3 and Table 4, respectively.

In addition, the relation between the supervised metric, i.e. (42) (with the settings of Figure 8(f)), and the unsupervised metric are analyzed: the correlation coefficients and the corresponding p-values are plotted in Figure 9. The statistical model has been obtained using 3DFA1/GUM. Each figure corresponds to a different value for the threshold ε in (42). As can be seen, increasing the threshold leads to a higher correlation between the supervised and unsupervised metrics.

Figure 10 illustrates the proposed method with two examples from AFLW2000-3D. In all these examples 3DFA1/GStudent-EM was used for training and GStudent-EM was used for testing. One may notice that all the 3DFA methods partially fail on the right hand-side example. Because the predicted results correspond to valid facial landmarks, the proposed pipeline incorrectly classifies many landmarks as inliers. It is worth noticing that the incorrect landmarks, associated with the automatic annotation process, are correctly classified as outliers. This example shows the limitations of both the unsupervised (proposed) and supervised metrics because they are unable to assess whether the predicted landmarks are coherent with the RGB information.

Table 3: AUC, NME mean and NME standard deviation associated with Figure 8(d) up to NME = 30%.

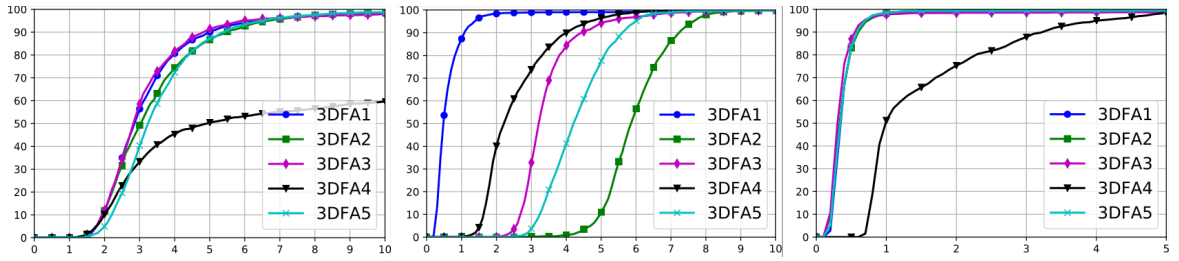
Method	AUC ↑ (%)	Mean NME ↓ (%)	Std. NME ↓ (%)
3DFA1	56.50	14.87	1.67
3DFA2	57.48	**9.22**	**0.64**
3DFA3	**58.24**	9.84	0.95
3DFA4	35.84	31.24	1.07
3DFA5	55.73	10.18	0.76

Table 4: AUC, NME mean and NME standard deviation associated with Figure 8(f) up to NME = 30%.

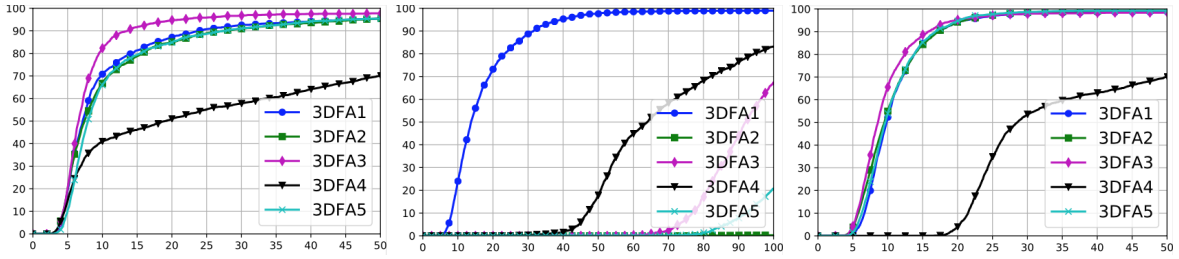
Method	AUC ↑ (%)	Mean NME ↓ (%)	Std. NME ↓ (%)
3DFA1	49.48	11.29	0.46
3DFA2	**52.01**	**10.20**	**0.34**
3DFA3	51.08	12.72	0.67
3DFA4	6.00	41.67	0.71
3DFA5	51.10	10.79	0.43

7. Discussion and Conclusions

This paper proposes to analyse the performance of DNN-based 3DFA in an unsupervised way. The rationale of the approach is to build a statistical shape model that encodes variabilities due to identities and to non-rigid facial deformations. The performance analysis per-se takes as input a set of predicted 3D landmarks and then maps this set onto a frontal pose. A statistical confidence test is used to classify each landmark as either an inlier or an outlier. The inlier/outlier decision is based on simply verifying whether a landmark lies inside or outside an ellipsoidal-shaped volume of confidence. Each ellipsoid has a posterior mean as a center, and corresponds to a posterior covariance. The use of robust probability distribution functions (Gaussian-uniform mixture or generalized Student t-

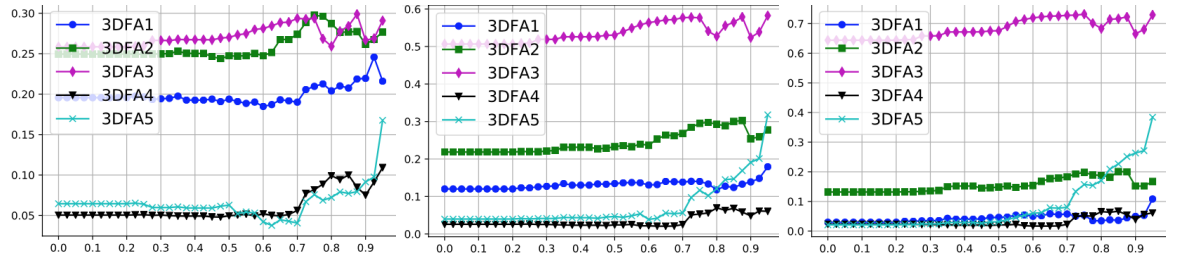


(a) 2D landmarks (bounding-box) (b) 3D landmarks (bounding-box) (c) Centered 3D (bounding-box)

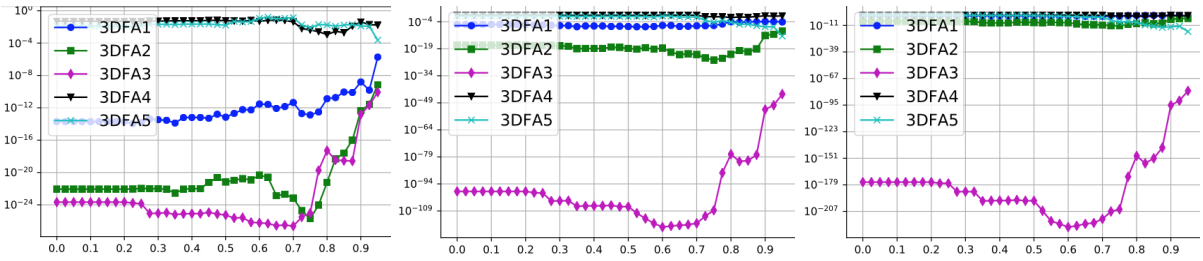


(d) 2D landmarks (eye-distance) (e) 3D landmarks (eye-distance) (f) Centered 3D landmarks (eye-distance)

Figure 8: Cumulative error distribution (CED) curves computed with (43) for 2D and 3D landmark coordinates, for bounding-box and eye-distance normalizations.



(a) Correlation ($\varepsilon = 0.10$) (b) Correlation ($\varepsilon = 0.20$) (c) Correlation ($\varepsilon = 0.30$)



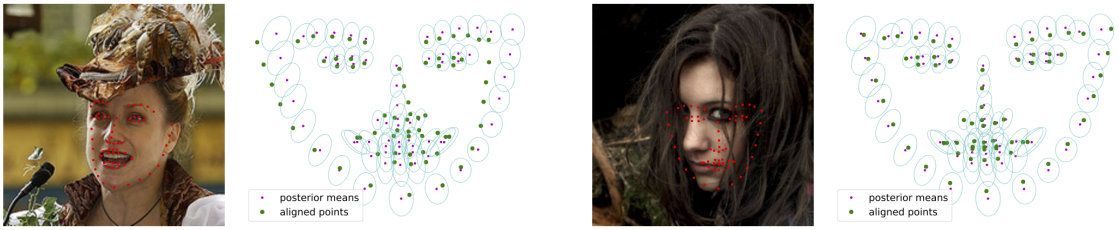
(d) P-value ($\varepsilon = 0.10$) (e) P-value ($\varepsilon = 0.20$) (f) P-value ($\varepsilon = 0.30$)

Figure 9: Correlation (top row) and the corresponding p-value (bottom row) between the supervised and unsupervised metrics for three different values of ε defined in (42).

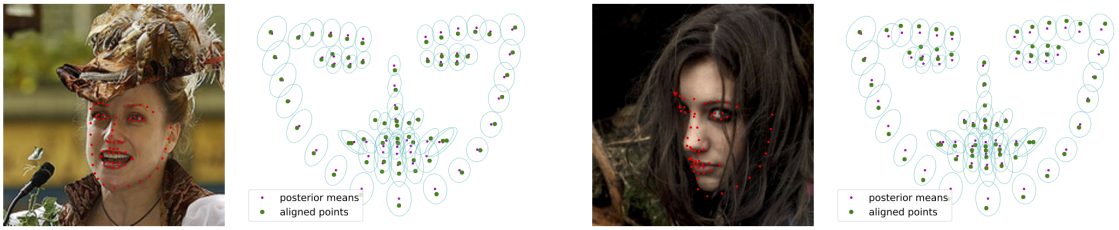
(a) Results obtained with 3DFA1 [20]

(b) Results obtained with 3DFA2 [21]

(c) Results obtained with 3DFA3 [12]



(d) Results obtained with 3DFA4 [22]



(e) Results obtained with 3DFA5 [23]

(f) Results obtained with the semi-automatic annotations [12].

Figure 10: Two examples from the AFLW2000-3D dataset [12]. Left column: 3D landmarks predicted with five 3DFA architectures and with the semi-automatic annotation method of [12] (last row). Right column: results obtained by robustly mapping the predicted landmarks onto the statistical frontal landmark model.

distribution) enable the proposed method to disregard large errors from the shape model. Indeed, there is a score (or a weight) associated with each landmark, which prevents the confidence volume to grow exaggeratedly large, i.e. (31) and (33).

The foundational principle of the proposed method is to rotate, translate and scale the landmarks in order to obtain a normalized representation (or a natural mathematical home) for the shape embedded in the facial landmarks. Key to the success of the proposed pipeline is the robust estimation of the rigid parameters: both GUM-EM and GStudent-EM yield accurate rigid parameters in the presence of non-rigid facial deformations and of (up to) 50% large errors. This reveals, as expected, that landmarks associated with deformable regions of the face, e.g. the lips and the lower jaw, have larger confidence volumes than landmarks associated with rigid regions, e.g. the nose, the eyes and the upper jaw, e.g. Figure 6. The proposed method could also be seen as a procedure to separate rigid head motions from non-rigid facial deformations and, hence, to enable the analysis of facial expressions in realistic settings, i.e. in the presence of head movements. The statistical model could be used as a prior for the dynamic analysis of facial expressions. The method could be easily adapted to the task of evaluating 3D landmarks predicted from 3D face scans, e.g. [53, 54].

A limitation of the proposed methodology is linked to the discriminative nature of 3DFA training and its inherent use of semi-automatic annotations, which is likely to induce errors. The negative effects due to the propagation of these errors are mitigated by the use of computationally tractable robust estimators. Nevertheless, small annotation errors are indistinguishable from variabilities due to identity and expressions.

The proposed unsupervised analysis empirically reveals that neither 3DFA predictions nor automatic annotations could count as ground-truth landmark coordinates. A promising follow-up is to consider several "experts" as proposed in [55]. Here an expert is a 3DFA architecture combined with a robust rigid-mapping estimator. Each such expert could infer a statistical frontal landmark model which may well be viewed as an observation of the unknown ground truth. Then, one may consider several experts simultaneously and cast the problem of estimating the unknown ground-truth into the problem of maximization of the joint distribution of the complete data, namely the observed data (provided by the experts) and the hidden data (the unknown annotations).

Appendix A. Closed-Form Solution Using Unit Quaternions

Consider (10) with $\boldsymbol{\Sigma} = \sigma\mathbf{I}$. We immediately obtain the following formulas for the model parameters:

$$s^\star = \left(\frac{\sum_{n=1}^N \alpha_n \hat{\mathbf{y}}_n^\top \hat{\mathbf{y}}_n}{\sum_{n=1}^N \alpha_n \hat{\mathbf{x}}_n^\top \hat{\mathbf{x}}_n}\right)^{1/2}. \tag{A.1}$$

$$\mathbf{R}^\star = \underset{\mathbf{R}}{\operatorname{argmin}} \frac{1}{2}\sum_{n=1}^N \alpha_n \|\hat{\mathbf{y}}_n - s^\star \mathbf{R}\hat{\mathbf{x}}_n\|^2, \tag{A.2}$$

$$\sigma^\star = \frac{1}{3\sum_{n=1}^N \alpha_n}\sum_{n=1}^N \alpha_n \|\hat{\mathbf{y}}_n - s^\star \mathbf{R}^\star \hat{\mathbf{x}}_n\|^2, \tag{A.3}$$

The formula for the posteriors becomes:

$$\alpha_n = \frac{p(2\pi\sigma)^{-3/2}\exp(-\|\boldsymbol{x}_n\|^2/2\sigma)}{p(2\pi\sigma)^{-3/2}\exp(-\|\boldsymbol{x}_n\|^2/2\sigma) + (1-p)\gamma^{-1}} \tag{A.4}$$

It is well known that a rotation matrix can be parameterized by a unit quaternion [45]. Let $\mathbf{R}$ be parameterized by its axis and angle of rotation, $\boldsymbol{n} = (n_1\ n_2\ n_3)^\top$, $\|\boldsymbol{n}\| = 1$ and ϕ. The unit quaternion parameterizing the rotation is:

$$\begin{aligned} q &= \cos\frac{\phi}{2} + \sin\frac{\phi}{2}(in_1 + jn_2 + kn_3) \\ &= q_0 + iq_1 + jq_2 + kq_3, \end{aligned} \tag{A.5}$$

with $i^2 = j^2 = k^2 = ijk = -1$, $\boldsymbol{q} = (q_0\ q_1\ q_2\ q_3)^\top \in \mathbb{R}^4$ by abuse of notation, and $\boldsymbol{q}\boldsymbol{q}^\top = 1$. A vector $\boldsymbol{a} \in \mathbb{R}^3$ can be represented as a purely imaginary quaternion, namely $\widetilde{\boldsymbol{a}} = (0\ a_1\ a_2\ a_3)^\top \in \mathbb{R}^4$. The action of a rotation onto $\widetilde{\boldsymbol{a}}$ can be written as $\boldsymbol{q} * \widetilde{\boldsymbol{a}} * \overline{\boldsymbol{q}}$, where the symbol $*$ corresponds to the quaternion product and $\overline{\boldsymbol{q}}$ is the conjugate of $\boldsymbol{q}$, namely $\overline{q} = q_0 - iq_1 - jq_2 - kq_3$. Making use of the properties $\|\boldsymbol{q}_1 * \boldsymbol{q}_2\|^2 = \|\boldsymbol{q}_1\|^2\|\boldsymbol{q}_2\|^2$ and $\overline{\boldsymbol{q}} * \boldsymbol{q} = \|\boldsymbol{q}\|^2 = 1$, the squared Euclidean norm in (A.2) can be successively written as:

$$\begin{aligned} \|\hat{\mathbf{y}}_n - s\mathbf{R}\hat{\mathbf{x}}_n\|^2 &= \|\widetilde{\mathbf{y}}_n - s\boldsymbol{q} * \widetilde{\mathbf{x}}_n * \overline{\boldsymbol{q}}\|^2\|\boldsymbol{q}\|^2 \\ &= \|\widetilde{\mathbf{y}}_n * \boldsymbol{q} - s\boldsymbol{q} * \widetilde{\mathbf{x}}_n * \overline{\boldsymbol{q}} * \boldsymbol{q}\|^2 \\ &= \|\widetilde{\mathbf{y}}_n * \boldsymbol{q} - s\boldsymbol{q} * \widetilde{\mathbf{x}}_n\|^2 \\ &= \|Q(\widetilde{\mathbf{y}}_n)\boldsymbol{q} - sW(\widetilde{\mathbf{x}}_n)\boldsymbol{q}\|^2 \\ &= \boldsymbol{q}^\top \mathbf{M}_n \boldsymbol{q}, \end{aligned} \tag{A.6}$$

with:

$$\mathbf{M}_n = (Q(\widetilde{\mathbf{y}}_n) - sW(\widetilde{\mathbf{x}}_n))^\top (Q(\widetilde{\mathbf{y}}_n) - sW(\widetilde{\mathbf{x}}_n)), \tag{A.7}$$

and where we replaced the quaternion products $\widetilde{\boldsymbol{a}} * \boldsymbol{q}$ and $\boldsymbol{q} * \widetilde{\boldsymbol{a}}$ with matrix-vector products $\mathbf{Q}(\widetilde{\boldsymbol{a}})\boldsymbol{q}$ and $\mathbf{W}(\widetilde{\boldsymbol{a}})\boldsymbol{q}$, with:

$$\mathbf{Q}(\widetilde{\boldsymbol{a}}) = \begin{pmatrix} 0 & -a_1 & -a_2 & -a_3 \\ a_1 & 0 & -a_3 & a_2 \\ a_2 & a_3 & 0 & -a_1 \\ a_3 & -a_2 & a_1 & 0 \end{pmatrix} \tag{A.8}$$

$$\mathbf{W}(\widetilde{\boldsymbol{a}}) = \begin{pmatrix} 0 & -a_1 & -a_2 & -a_3 \\ a_1 & 0 & a_3 & -a_2 \\ a_2 & -a_3 & 0 & a_1 \\ a_3 & a_2 & -a_1 & 0 \end{pmatrix} \tag{A.9}$$

Consequently, the right-hand side of (A.2) writes

$$\sum_{n=1}^{N} (\boldsymbol{q}^\top \alpha_n \mathbf{M}_n \boldsymbol{q}) = \boldsymbol{q}^\top (\sum_{n=1}^{N} \alpha_n \mathbf{M}_n) \boldsymbol{q} = \boldsymbol{q}^\top \mathbf{M} \boldsymbol{q},$$

where $\alpha_n \geq 0$ and $\mathbf{M}_n \in \mathbb{R}^{4\times4}$ is semi-definite positive symmetric, i.e. (A.7), hence so is $\mathbf{M}$. By constraining the minimizer to be a unit quaternion, we obtain the following minimization problem:

$$\min_{\boldsymbol{q}} Q(\boldsymbol{q}) = \min_{\boldsymbol{q}} (\boldsymbol{q}^\top \mathbf{M} \boldsymbol{q} + \lambda(1 - \boldsymbol{q}^\top \boldsymbol{q})). \tag{A.10}$$

From $dQ/d\boldsymbol{q} = 0$ we obtain $\mathbf{M}\boldsymbol{q} = \lambda \boldsymbol{q}$ and by substitution in (A.10) we obtain $Q(\boldsymbol{q}) = \lambda$. Therefore, the minimization problem (A.10) is equivalent to estimating the smallest eigenvalue-eigenvector pair $(\lambda^\star, \boldsymbol{q}^\star)$ of $\mathbf{M}$.

Appendix B. Implementation Details

Algorithm 1 and Algorithm 2 are expectation maximization (EM) procedures and it is well known that they have good convergence properties. One should notice that all the computations inside these algorithms are in closed-form, with the notable exception of the estimation of the rotation matrix. The latter is parameterized with a unit quaternion and it is estimated via optimization of (8). The unit-quaternion parameterization of rotations, i.e. Appendix Appendix A, has several advantages: (i) the number of parameters to be estimated is reduced from nine to four, (ii) the number of nonlinear constraints is reduced from seven constraints (six quadratic constraints, i.e. $\mathbf{R}^\top\mathbf{R} = \mathbf{I}$, and one quartic constraint, i.e. $|\mathbf{R}| = 1$) to one quadratic constraint ($\boldsymbol{q}^\top\boldsymbol{q} = 1$), (iii) the initialization is performed with the closed-form solution of [45] that uses a unit quaternion as well.

In practice, the constrained nonlinear optimization problem (8) is solved using the sequential quadratic programming method [56], more precisely a sequential least squares programming (SLSQP) solver[3] is used in combination with a root-finding software package [57]. The SLSQP minimizer found at the previous EM iteration is used as an initial estimate at the current EM iteration. The closed-form method of Appendix A is used to initialize the unit-quaternion, and hence the rotation matrix, at the start of the EM algorithm.

References

[1] S. Escalera, X. Baro, I. Guyon, H. J. Escalante, G. Tzimiropoulos, M. Valstar, M. Pantic, J. Cohn, and T. Kanade, "Special issue on the computational face," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 40, no. 11, pp. 2541–2545, Nov 2018. 1

[2] C. C. Loy, X. Liu, T.-K. Kim, F. De la Torre, and R. Chellappa, "Special issue on deep learning for face analysis," *International Journal of Computer Vision*, vol. 127, no. 6, pp. 533–536, June 2019. 1

[3] M. Wang and W. Deng, "Deep face recognition: A survey," *Neurocomputing*, vol. 429, pp. 215–244, 2021. 1

[4] Y. Wu and Q. Ji, "Facial landmark detection: A literature survey," *International Journal of Computer Vision*, vol. 127, no. 2, pp. 115–142, 2019. 1

[5] J. Deng, G. Trigeorgis, Y. Zhou, and S. Zafeiriou, "Joint multi-view face alignment in the wild," *IEEE Transactions on Image Processing*, vol. 28, no. 7, pp. 3636–3648, 2019. 1

[6] X. Dong, S.-I. Yu, X. Weng, S.-E. Wei, Y. Yang, and Y. Sheikh, "Supervision-by-registration: An unsupervised approach to improve the precision of facial landmark detectors," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 360–368. 1

[7] X. Dong, Y. Yang, S.-E. Wei, X. Weng, Y. Sheikh, and S.-I. Yu, "Supervision by registration and triangulation for landmark detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 43, no. 10, pp. 3681–3694, 2020. 1

[8] J. Wan, Z. Lai, J. Li, J. Zhou, and C. Gao, "Robust facial landmark detection by multiorder multiconstraint deep networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 5, pp. 2181–2194, 2021. 1

[9] J. Wan, Z. Lai, J. Liu, J. Zhou, and C. Gao, "Robust face alignment by multi-order high-precision hourglass network," *IEEE Transactions on Image Processing*, vol. 30, pp. 121–133, 2021. 1

[10] J. Wan, J. Liu, J. Zhou, Z. Lai, L. Shen, H. Sun, P. Xiong, and W. Min, "Precise facial landmark detection by reference heatmap transformer," *IEEE Transactions on Image Processing*, vol. 32, pp. 1966–1977, 2023. 1

[11] C. Gou, Y. Wu, F.-Y. Wang, and Q. Ji, "Shape augmented regression for 3D face alignment," in *European Conference on Computer Vision*, 2016, pp. 604–615. 1

[12] X. Zhu, Z. Lei, X. Liu, H. Shi, and S. Z. Li, "Face alignment across large poses: a 3D solution," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 146–155. 1, 2, 10, 11, 13

[13] J. Deng, A. Roussos, G. Chrysos, E. Ververas, I. Kotsia, J. Shen, and S. Zafeiriou, "The Menpo benchmark for multi-pose 2D and 3D facial landmark localisation and tracking," *International Journal of Computer Vision*, vol. 127, no. 6-7, pp. 599–624, 2019. 1, 2, 3

[14] D. G. Kendall, D. Barden, T. K. Carne, and H. Le, *Shape and shape theory*. John Wiley & Sons, 2009. 2

[15] A. Bulat, "3D-FAN (V1.1.1)," https://github.com/1adrianb/face-alignment, 2020. 2, 10

[16] F. Yao, "PRNet," https://github.com/YadiraF/PRNet, 2018. 2, 10

[17] J. Guo, "3DDFA," https://github.com/cleardusk/3DDFA, 2019. 2, 10

[18] X. Tu and Y. Luo, "2DASL," https://github.com/XgTu/2DASL, 2019. 2, 10

[19] J. Guo, "3DDFA-V2," https://github.com/cleardusk/3DDFA_V2, 2021. 2, 10

[20] A. Bulat and G. Tzimiropoulos, "Two-stage convolutional part heatmap regression for the 1st 3D face alignment in the wild (3DFAW) challenge," in *European Conference on Computer Vision Workshops*. Springer, 2016, pp. 616–624. 2, 8, 9, 11, 13

[21] Y. Feng, F. Wu, X. Shao, Y. Wang, and X. Zhou, "Joint 3D face reconstruction and dense alignment with position map regression network," in *European Conference on Computer Vision*, 2018, pp. 534–551. 2, 8, 9, 10, 11, 13

[22] X. Tu, J. Zhao, Z. Jiang, Y. Luo, M. Xie, Y. Zhao, L. He, Z. Ma, and J. Feng, "3D face reconstruction from a single image assisted by 2D face images in the wild," *IEEE Transactions on Multimedia*, vol. 23, pp. 1160–1172, May 2021. 2, 10, 13

[23] J. Guo, X. Zhu, Y. Yang, F. Yang, Z. Lei, and S. Z. Li, "Towards fast, accurate and stable 3D dense face alignment," in *European Conference on Computer Vision*. Springer, 2020, pp. 152–168. 2, 10, 13

[24] S. Mostafa, "UPA-3DFA," https://gitlab.inria.fr/smostafa/upa3dfa, 2023. 2

[25] P. Szeptycki, M. Ardabilian, and L. Chen, "A coarse-to-fine curvature analysis-based rotation invariant 3D face landmarking," in *International Conference on Biometrics: Theory, Applications, and Systems*. IEEE, 2009, pp. 1–6. 2, 3

[26] X. Zhu, X. Liu, Z. Lei, and S. Z. Li, "Face alignment in full pose range: A 3d total solution," *IEEE Transactions on Pattern Analysis and Machine*

[3] https://docs.scipy.org/doc/scipy/reference/optimize.html

Intelligence, vol. 41, no. 1, pp. 78–92, 2019. 2, 10
[27] X. Ning, P. Duan, W. Li, and S. Zhang, "Real-time 3D face alignment using an encoder-decoder network with an efficient deconvolution layer," *IEEE Signal Processing Letters*, vol. 27, pp. 1944–1948, 2020. 2
[28] V.-T. Hoang, D.-S. Huang, and K.-H. Jo, "3-D facial landmarks detection for intelligent video systems," *IEEE Transactions on Industrial Informatics*, vol. 17, no. 1, pp. 578–586, 2021. 2
[29] P. J. Phillips, P. J. Flynn, T. Scruggs, K. W. Bowyer, J. Chang, K. Hoffman, J. Marques, J. Min, and W. Worek, "Overview of the face recognition grand challenge," in *Conference on Computer Vision and Pattern Recognition*, vol. 1. IEEE, 2005, pp. 947–954. 2
[30] L. A. Jeni, S. Tulyakov, L. Yin, N. Sebe, and J. F. Cohn, "The first 3D face alignment in the wild (3DFAW) challenge," in *European Conference on Computer Vision*. Springer, 2016, pp. 511–520. 2, 3
[31] S. Sanyal, T. Bolkart, H. Feng, and M. J. Black, "Learning to regress 3D face shape and expression from an image without 3D supervision," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 7763–7772. 2, 3
[32] L. Yin, X. C. Y. Sun, T. Worm, and M. Reale, "A high-resolution 3D dynamic facial expression database," in *IEEE International Conference on Automatic Face and Gesture Recognition*, 2008. 3
[33] 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. 3
[34] R. Gross, I. Matthews, J. Cohn, T. Kanade, and S. Baker, "Multi-PIE," *Image and Vision Computing*, vol. 28, no. 5, pp. 807–813, 2010. 3
[35] L. A. Jeni, J. F. Cohn, and T. Kanade, "Dense 3D face alignment from 2D video for real-time use," *Image and Vision Computing*, vol. 58, pp. 13–24, 2017. 3
[36] A. D. Bagdanov, A. Del Bimbo, and I. Masi, "The Florence 2D/3D hybrid face dataset," in *Joint ACM Workshop on Human Gesture and Behavior Understanding*, 2011, pp. 79–80. 3
[37] G. McLachlan and D. Peel, "Robust mixture modelling using the t distribution," *Statistics and Computing*, vol. 10, no. 4, pp. 339–348, 2000. 3
[38] J. Sun, A. Kabán, and J. M. Garibaldi, "Robust mixture clustering using Pearson type VII distribution," *Pattern Recognition Letters*, vol. 31, no. 16, pp. 2447–2454, 2010. 3, 5
[39] F. Forbes and D. Wraith, "A new family of multivariate heavy-tailed distributions with variable marginal amounts of tailweight: application to robust clustering," *Statistics and computing*, vol. 24, no. 6, pp. 971–984, 2014. 3, 5
[40] F. Chamroukhi, "Skew t mixture of experts," *Neurocomputing*, vol. 266, pp. 390–408, 2017. 3
[41] J. D. Banfield and A. E. Raftery, "Model-based gaussian and non-gaussian clustering," *Biometrics*, pp. 803–821, 1993. 3
[42] A. Zaharescu and R. Horaud, "Robust factorization methods using a Gaussian/uniform mixture model," *International Journal of Computer Vision*, vol. 81, no. 3, pp. 240–258, 2009. 3
[43] A. Myronenko and X. Song, "Point set registration: Coherent point drift," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 12, pp. 2262–2275, 2010. 3
[44] S. Lathuilière, P. Mesejo, X. Alameda-Pineda, and R. Horaud, "DeepGUM: Learning deep robust regression with a Gaussian-uniform mixture model," in *European Conference on Computer Vision*, 2018, pp. 202–217. 3
[45] B. K. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America A*, vol. 4, no. 4, pp. 629–642, 1987. 4, 5, 7, 14, 15
[46] S. Umeyama, "Least-squares estimation of transformation parameters between two point patterns," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 4, pp. 376–380, April 1991. 4
[47] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Computer Vision and Pattern Recognition*, vol. 1. Ieee, 2001. 8
[48] V. Drouard, R. Horaud, A. Deleforge, S. Ba, and G. Evangelidis, "Robust head-pose estimation based on partially-latent mixture of linear regressions," *IEEE Transactions on Image Processing*, vol. 26, no. 3, pp. 1428–1440, 2017. 8
[49] S. Abtahi, M. Omidyeganeh, S. Shirmohammadi, and B. Hariri, "YawDD: a yawning detection dataset," in *ACM Multimedia Systems Conference*, 2014, pp. 24–28. 8
[50] L. J. Savage, *The Foundations of Statistics*. Dover, 1972. 9
[51] F. Huber, *A Logical Introduction to Probability and Induction*. Oxford University Press, 2018. 9
[52] A. Bulat and G. Tzimiropoulos, "How far are we from solving the 2D & 3D face alignment problem? (and a dataset of 230,000 3D facial landmarks)," in *IEEE International Conference on Computer Vision*, 2017, pp. 1021–1030. 10, 11
[53] J. Zhang, K. Gao, K. Fu, and P. Cheng, "Deep 3D facial landmark localization on position maps," *Neurocomputing*, vol. 406, pp. 89–98, 2020. 14
[54] Y. Wang, M. Cao, Z. Fan, and S. Peng, "Learning to detect 3D facial landmarks via heatmap regression with graph convolutional network," in *The 36th AAAI Conference on Artificial Intelligence*, 2022. 14
[55] A. Akhondi-Asl, L. Hoyte, M. E. Lockhart, and S. K. Warfield, "A logarithmic opinion pool based STAPLE algorithm for the fusion of segmentations with associated reliability weights," *IEEE Transactions on Medical Imaging*, vol. 33, no. 10, pp. 1997–2009, 2014. 14
[56] J.-F. Bonnans, J. C. Gilbert, C. Lemaréchal, and C. A. Sagastizábal, *Numerical optimization: theoretical and practical aspects*. Springer Science & Business Media, 2006. 15
[57] D. Kraft, "A software package for sequential quadratic programming," DLR German Aerospace Center – Institute for Flight Mechanics, Koln, Germany, Tech. Rep. DFVLR-FB 88-28, 1988. 15