The open University of Israel
Department of Mathematics and Computer Science

Effective Face Frontalization in Unconstrained Images

Final Paper submitted as partial fulfillment of the requirements
For an M.Sc. degree in Computer Science
The Open University of Israel
Computer Science Division

by

Shai Harel

Prepared under the supervision of Dr. Tal Hassner

December 2015

Abstract

"Frontalization" is the process of synthesizing frontal facing views of faces appearing in single unconstrained photos. Recent reports have suggested that this process may substantially boost the performance of face recognition systems. This, by transforming the challenging problem of recognizing faces viewed from unconstrained viewpoints to the easier problem of recognizing faces in constrained, forward facing poses. Previous frontalization methods did this by attempting to approximate 3D facial shapes for each query image. We observe that 3D face shape estimation from unconstrained photos may be a harder problem than frontalization and can potentially introduce facial misalignments. Instead, we explore the simpler approach of using a single, unmodified, 3D surface as an approximation to the shape of all input faces. We show that this leads to a straightforward, efficient and easy to implement method for frontalization. More importantly, it produces aesthetic new frontal views and is surprisingly effective when used for face recognition and gender estimation.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Face recognition performances, reported as far back as [8], have shown computer vision capabilities to surpass those of humans. Rather than signaling the end of face recognition research, these results have led to a redefinition of the problem, shifting attention from highly regulated, controlled image settings to faces captured in unconstrained conditions (a.k.a., "in the wild").

This change of focus, from constrained to unconstrained images, has toppled recognition rates (see, e.g., the original results [5] on the Labeled Faces in the Wild [9] benchmark, published in the same year as [8]). This drop was not surprising: Unconstrained photos of faces represented a myriad of new challenges, including changing expressions, occlusions, varying lighting, and non-frontal, often extreme poses. Yet in recent years recognition performance has gradually improved to the point where once again claims are being made for super-human face recognition capabilities (e.g., [10, 11, 4]).

Modern methods vary in how they address the many challenges of unconstrained face recognition. Facial pose variations in particular have often been considered by designing representations that pool information over large image regions, thereby accounting for possible misalignments due to pose changes (e.g., [10, 12, 13, 14]), by improving 2D face alignment accuracy [6, 5, 1], or by using massive face collections to learn pose-robust representations [15, 16, 17, 18].

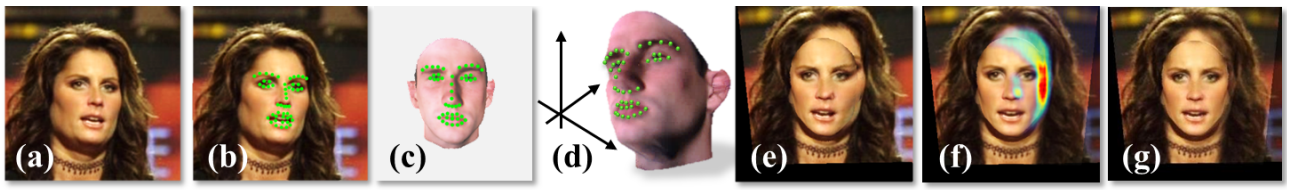

Figure 1.1: **Frontalization process overview.** (a) Query photo; (b) facial feature detections; (c) the same detector used to localize the same facial features in a reference face photo, produced by rendering a textured 3D computer graphics model (d); (e) from the 2D coordinates on the query and their corresponding 3D coordinates on the model we estimate a projection matrix which is then used to back-project query intensities to the reference coordinate system; (f) estimated visibility due to non-frontal poses, overlaid on the frontalized result. Warmer colors reflect less visible pixels. Facial appearance in these regions is produced by borrowing colors from corresponding symmetric parts of the face; (g) our final frontalized result.

Recently, some proposed to simplify unconstrained face recognition by reducing it, at least in terms of pose variations, to the simpler, constrained settings. This, by automatic synthesis of new, frontal facing views, or "frontalization" [3, 4]. To this end, they attempt to estimate a rough approximation for the 3D surface of the face and use this surface to generate the new views. Although appealing, this approach relies on accurate localization of facial feature points and does not guarantee that the same alignment (frontalization) will be applied to different images of the same face. Thus, different images of the same person may well be aligned differently, preventing their features from being accurately compared.

We propose the simple alternative approach of using a single, unmodified 3D reference for all query faces in order to produce frontalized views. Ignoring individual differences in facial shapes may be counter-intuitive – indeed, previous work has emphasized its importance [4] – however, qualitative examples throughout this paper show that any impact this has on facial appearances is typically negligible. In fact, faces remain easily recognizable despite this approximation. More importantly, our frontalized faces are aggressively aligned thereby improving performances over previous alignment methods. These claims are verified by showing elevated face verification results on the LFW benchmark and gender classification accuracy on the Adience benchmark, obtained using our frontalized faces.

Chapter 2

Related work

Generating novel views of a face viewed in a single image has been a longstanding challenge in computer vision, due in large part to the potential applications such methods have in face processing and recognition systems.

Previous methods for synthesizing new facial views typically did so by estimating the 3D surface of the face appearing in the photo with varying emphasis on reconstruction accuracy. Morphable-Models based methods [19, 20, 21, 22] attempt to learn the space of allowable facial geometries using many aligned 3D face models. These methods, however, typically require near-frontal views of clear, unoccluded faces, and so are not suitable for our purposes.

Shape from shading methods have been shown to produce outstanding facial details (e.g., [23]). Their sensitivity to occlusions and specularities (e.g., eyeglasses) and requirement for careful segmentation of faces from their backgrounds make them less suited for automatic, large scale application in face processing systems.

Facial symmetry was used in [24] to estimate 3D geometry. Like us, symmetry was used for replacing details in out-of-view facial regions in [25]. These methods have only been applied to controlled views due to their reliance on accurate segmentation.

Related to our work is [3] and its extension for face recognition in [4]. Both attempt to adjust a 3D reference face, fitting it to the texture of the query face in order to preserve natural

appearances. This 3D estimation process, however, cannot guarantee that a similar shape would be produced for difference images of the same face. It further either relies on highly accurate facial feature localizations [4], which can be difficult to ensure in practice, or is computationally heavy, unsuited for mass processing [3].

Finally, [18] described a deep-learning based method for estimating canonical views of faces. Their method is unique in producing frontal views without estimating (or using) 3D information in the process. Besides requiring substantial training, their canonical views are not necessarily frontalized faces and are not guaranteed to be similar to the person appearing in the input image.

We propose to use a single 3D reference surface, unchanged, in order to produce front facing views for all query images. Despite the simplicity of this approach, we are unaware of previous reports of its use in unconstrained face photo alignment for face recognition. We explore the implications of our approach both qualitatively and empirically.

Chapter 3

Frontalization

We use the term "hard frontalization" to emphasize our use of a single, 3D, reference face geometry. This, in contrast to others who estimate or modify 3D facial geometry to fit facial appearances (Sec. 2). Our goal is to produce better aligned images which allow for accurate comparison of local facial features between different faces. As we next show, the use of a single 3D face results in a straightforward frontalization method which, despite its simplicity, is quite effective.

Our method is illustrated in Fig. 1.1. A face is detected using an off-the-shelf face detector viola2004robust and then cropped and rescaled to a standard coordinate system. The same dimensions and crop ratios previously used for Labeled Faces in the Wild (LFW) [9] images are used here in order to maintain parameter comparability with previous results.

Facial feature points are localized and used to align the photo with a textured, 3D model of a generic, reference face. A rendered, frontal view of this face provides a reference coordinate system. An initial frontalized face is obtained by back-projecting the appearance (colors) of the query photo to the reference coordinate system using the 3D surface as a proxy. A final result is produced by borrowing appearances from corresponding symmetric sides of the face wherever facial features are poorly visible due to the query's pose. These steps are details next.

3.1 Generating a frontalized view

We begin by computing a 3×4 projection matrix which approximates the one used to capture the query photo. To this end, we seek 2D-3D correspondences between points in the query photo (Fig 1.1 (a)) and points on the surface of our 3D face model (Fig 1.1 (d)). This, by matching query points to points on a rendered, frontal view of the model (Fig 1.1 (c)). Directly estimating correspondences between a real photo and a synthetic, rendered image can be exceedingly hard [26]. Instead, we use a robust facial feature detection method which seeks the same landmarks (e.g., corners of the eyes, mouth etc.) in both images.

Facial feature detection. Many highly effective methods were recently proposed for detecting facial features. In designing our system, we tested several state-of-the-art detectors, selecting the supervised descent method (SDM) of [27] as the one which balances both speed of detection with accuracy. Unlike others, the 49 facial features it localizes do not include points along the jawline (Fig 1.1 (b-c)). The features it detects are therefore all images of points lying close to the 3D plane at the front of the face. These and other concerns have been suggested in the past as reasons for preferring other approaches to pose estimation [28, 29]. As we later show, this did not appear to be a problem in our tests.

Pose estimation. Given a textured 3D model of a face, the synthetic, rendered view of this model is produced by specifying a reference projection matrix $\mathbf{C}_M = \mathbf{A}_M \quad [\mathbf{R}_M \quad \mathbf{t}_M]$, where $\mathbf{A}_M$ is the intrinsic matrix, and $[\mathbf{R}_M \quad \mathbf{t}_M]$ the extrinsic matrix consisting of rotation matrix $\mathbf{R}_M$ and translation vector $\mathbf{t}_M$. We select rotation and translation to produce a frontal view of the model (Fig. 1.1 (c)) which serves as our reference (frontalized) coordinate system.

When producing the reference view I_R we store for each of its pixels $\mathbf{p}'$ the 3D point coordinates $\mathbf{P} = (X, Y, Z)^T$ of the point located on the surface of the 3D model for which:

$$\mathbf{p}' \sim \mathbf{C}_M \quad \mathbf{P}. \tag{3.1}$$

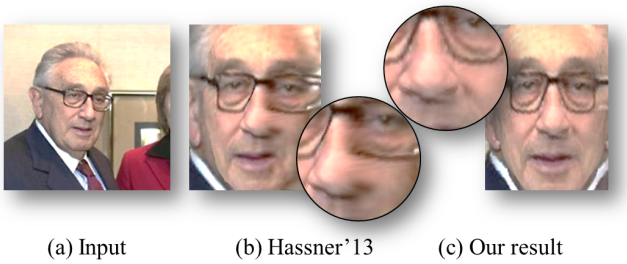

Figure 3.1: **Occlusion handling comparison.** (a) Input image. (b) Frontalization obtained by the method of [3], showing noticeable smearing artifacts wherever input facial features were occluded. (c) Our result, produced with occlusion detection and soft facial symmetry.

Let $\mathbf{p}_i = (x_i, y_i)^T$ be facial feature points detected in the query photo I_Q (Fig. 1.1 (b)), and $\mathbf{p}'_i = (x'_i, y'_i)^T$ be the same facial features, detected in the reference view (Fig. 1.1 (c)). From Eq. 3.1, we have the coordinates $\mathbf{P}_i = (X_i, Y_i, Z_i)^T$ of the point on the surface of the model, projected onto $\mathbf{p}'_i$ (Fig. 1.1 (d)). This provides the correspondences $(\mathbf{p}_i^T, \mathbf{P}_i^T) = (x_i, y_i, X_i, Y_i, Z_i)$ which allow estimating the projection matrix $\mathbf{M}_Q = \mathbf{A}_Q \ [\mathbf{R}_Q \ \ \mathbf{t}_Q]$, approximating the camera matrix used to capture the query photo I_Q [30]. Projection matrix estimation itself is performed using standard techniques (Sec. 4).

Frontal pose synthesis. An initial frontalized view I_F is produced by projecting query facial features back onto the reference coordinate system using the geometry of the 3D model. For every pixel coordinate $\mathbf{q}' = (x', y')^T$ in the reference view, from Eq. 3.1 we have the 3D location $\mathbf{P} = (X, Y, Z)^T$ on the surface of the reference which was projected onto $\mathbf{q}'$ by $\mathbf{C}_M$. We use the expression

$$\mathbf{p} \sim \mathbf{C_Q} \ \ \mathbf{P} \tag{3.2}$$

to provide an estimate for the location $\mathbf{p} = (x, y)^T$ in I_Q of that same facial feature. Bi-linear interpolation is used to sample the intensities of the query photo at $\mathbf{p}$. The sampled color is then assigned to pixel coordinates $\mathbf{q}'$ in the new, frontalized view (Fig. 1.1 (e)).

3.2 Soft symmetry for self-occlusions

Out-of-plane rotation of the head can cause some facial features to be less visible than others, particularly those on the sides of the nose and head. In [3] a depth map was used to generate new views. This has the effect of over-sampling, or "smearing" textures whenever they were occluded in the original photo (Fig. 3.1 (b). In [4], the authors suggest using mesh triangle visibility, presumably using 3D surface normals computed on their low resolution 3D shape estimate (Fig. 3.6 (top right)), though it is not clear if they employed this approach in practice. In doing so they rely on the accuracy of their facial feature detector to define the exact positions of their 3D triangles. In addition, the coarse triangulation used to represent their 3D model may not provide accurate enough, per-pixel visibility estimates.

Estimating visibility. We estimate visibility using an approach similar to the one used by multi-view 3D reconstruction methods (e.g., [31, 32]). Rather than using two or more views to estimate 3D geometry we use an approximation to the 3D geometry (the reference face) and a single view (I_R) to estimate visibility in a second image (I_Q).

We evaluate visibility by counting the number of times query pixels are accessed when generating the new view: As a face rotates away from the camera, the angle between its less visible features and the camera plane increases, consequently increasing the number of surface points projected onto the same pixel in the photo (Fig. 3.2). This translates to the following sampling-rate measure of visibility as follows.

Returning to Eq. 3.2, for each pixel $\mathbf{q}'$ in the reference view I_R, we store the location in the query photo of its corresponding pixel $\mathbf{q}$ (in practice, a quantized, integer value reflecting the nearest neighboring pixel for the non-integer value of $\mathbf{q}$). A visibility score is then determined for each pixel in the frontalized $\mathbf{q}'$ view by:

$$v(\mathbf{q}') = 1 - \exp(-\#\mathbf{q}). \tag{3.3}$$

Where $\#\mathbf{q}$ is the number of times query pixel $\mathbf{q}$ corresponded with *any* frontalized pixel $\mathbf{p}'$.

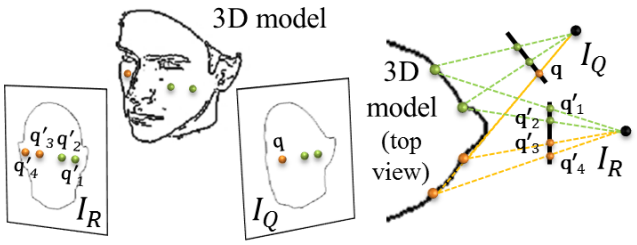

Figure 3.2: **Visibility estimation.** Pixels $\mathbf{q}'_3$ and $\mathbf{q}'_4$ in the reference (frontalized) coordinate system I_R, both map to the same pixel $\mathbf{q}$ in the query photo I_Q, and so would both be considered less visible. Their corresponding symmetric pixels $\mathbf{q}'_1$ and $\mathbf{q}'_2$ are used to predict their appearance in the final frontalized view.

Figure 3.3: **Visibility estimation for an extreme out-of-plane pose.** (a) Input image. (b) Visibility estimates overlaid on the initial frontalized image. Warmer colors reflect less visibility of these features in the original input image (a). (c) Frontalization with soft-symmetry.

Fig. 1.1 (f) and Fig. 3.3 (b) both visualize the estimated visibility rates for two faces, overlaid on the initial frontalized results. In both cases, facial features turned away from the camera are correctly highlighted.

We note that an alternative method of projecting the surface normals of the 3D model down to the query photo and using their values to determine visibility can also be employed. We found the approach described above faster and both methods provided similar results in practice.

Intensities of poorly visible pixels (low visibility scores in Eq. 3.3) are replaced by a mean of their intensities and the intensities of their corresponding symmetric pixels, weighted by the visibility scores. We note that this weighing of symmetric parts of the face can produce artifacts, especially when the head is at non-frontal poses and lighting on both parts of the face are different (e.g., the example in Fig. 3.3). Although more elaborate methods of blending the two parts of the face can be employed, descriptors commonly used for face recognition are typically designed to overcome noise and minor artifacts such as these, and so other blending

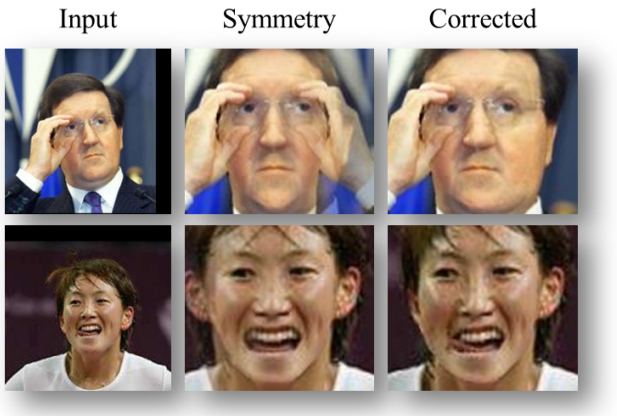

Figure 3.4: **Corrected soft-symmetry examples.** Left: Input image; Mid: results following soft symmetry. Right: Non-symmetric results, automatically selected due to detected symmetry errors. In the top row, symmetry replicated an occluding hand, in the bottom an unnatural expression was produced by transferring an asymmetric expression.

methods were not used here.

3.3 Conditional soft-symmetry

Although transferring appearances from one side of the face to another may correct pose related visibility issues, it can also introduce problems whenever one side of the face is occluded by anything other than the face itself: symmetry can replicate the occlusion, leaving the final result unrecognizable. Asymmetric facial expressions, lighting, and facial features may also cause frontalization errors. Two such examples are presented in Fig. 3.4 (mid).

In order to detect these failures, we take advantage of the aggressive alignment of the frontalized images. By using the same 3D reference, features on frontalized faces appear in the same image locations regardless of the actual shape of the face. For example, all right corners of all mouths will appear in the same image region on the frontalized faces. These local appearances, following such alignment, can be easily verified using a standard robust representation and a classifier trained on example patches extracted at the same facial locations.

In our implementation, we manually specified eight location on the reference face, corresponding to the sides of the mouth, nose and eyes. We then trained eight linear SVM classifiers, one

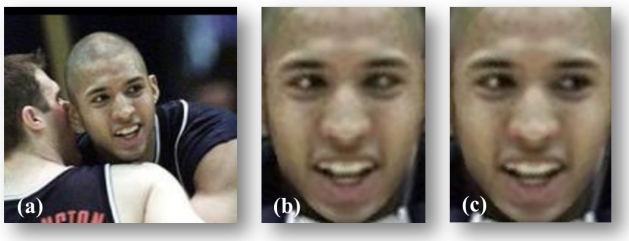

Figure 3.5: **Eye correction.** (a) Input image. (b) Frontalization with soft-symmetry. (c) Frontalization with eye-excluded soft-symmetry.

for each point, to recognize local appearances at each point, represented as LBP code [33, 34]. Training examples were generated from frontalized images (in practice, LFW [9] images not included in the benchmark tests) using LBP code patches extracted at these eight locations from all training images.

Given a new frontalized face, we classify its patches, extracted from the same eight locations. A frontalized face with soft symmetry is rejected in favor of the non-symmetric frontalization if more of the latter's points were correctly identified by their classifiers. Fig. 3.4 shows two examples with (erroneous) soft symmetry and the automatically selected, non-symmetric frontalized result.

Finally, eyes are ignored when symmetry is applied; their appearance is unchanged from the initial frontalized view regardless of their visibility. This is done for aesthetic reasons: As demonstrated in Fig. 3.5, simply using symmetry can result in unnaturally looking, cross-eyed faces, though this exclusion of the eyes did not seem to affect our face recognition performance one way or another. To exclude the eyes from the symmetry, we again exploit the strong alignment: Eye locations are selected once, in the reference coordinate system, and the same pixel coordinates were always excluded from the soft symmetry process.

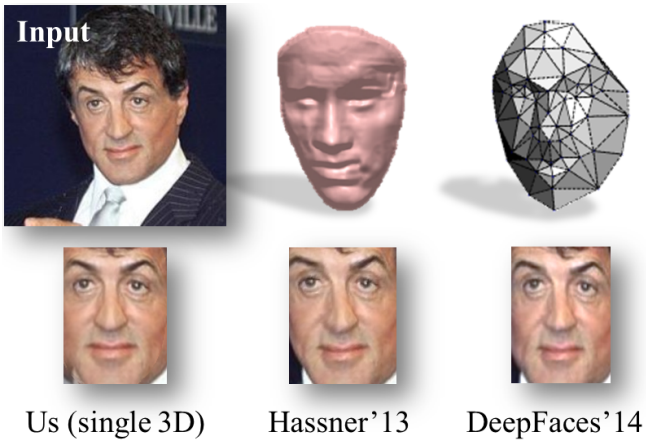

Figure 3.6: **Visualization of estimated 3D surfaces.** Top: Surfaces estimated for the same input image (left) by Hassner [3] (mid) and DeepFaces [4] (right). Bottom: Frontalized faces using our single-3D approach (left), Hassner (mid) and DeepFaces (right). Evidently, both surfaces are very rough approximations to the shape of the face. Moreover, despite the different surfaces, all three results seem qualitatively similar. This calls to question the need for shape estimation or fitting when performing frontalization.

3.4 Discussion: Soft vs. hard frontalization

Unlike previous methods we do not try to tailor a 3D surface to match the appearance of each query face. Ostensibly, doing so allowed previous methods to better preserve facial appearances in the new, synthesized views. We claim that this may actually be unnecessary and possibly even counterproductive; *damaging* rather than improving face recognition performance.

In [4], 3D facial geometry was altered by using the coordinates of detected facial feature points to modify a 3D surface, matching it to the query face. This surface, however, is a rough approximation of the true facial geometry, which preserves little if any identifying features (Fig. 3.6 (top-right)). Furthermore, there is no guarantee that local feature detections will be repeatedly detected in the same exact positions in different views of the same face. Thus, different 3D shapes could be estimated for different views of the same face, resulting in misaligned features and possible noise.

Although the problem of accurately detecting facial feature points is somewhat ameliorated in [3] by using dense correspondences rather than sparse image detections, they too produce only a rough approximation of the subject's face (Fig. 3.6 (top-mid)) and similarly cannot guarantee alignment of the same facial features across different images.

Of course, face shape differences may provide important cues for recognition. This is supported by many previous reports [35] which have found significant age, gender and ethnicity based differences in facial shapes. However, previous frontalization methods do not guarantee these differences will actually be preserved, implicitly relying on texture rather than shape for recognition. This is evident in Fig. 3.6 (bottom), where frontalizations for these two methods and our own appear qualitatively comparable.

Observing that for frontalization, one rough approximation to the 3D facial shape seems as good as another, we propose using the same 3D reference, unmodified with all faces. In doing so, we abandon attempts to preserve individual 3D facial structures in favor of gaining highly aligned faces. This is demonstrate in Fig. 3.7, showing average faces from our frontalized LFW set ("LFW3D"), as well as funneled [5], LFW-a [1] (aligned using a commercial system), and deep-funneled [6] versions of LFW. Our results all have slightly elongated faces, reflecting the shape of the reference face (Fig. 1.1 (c)), yet are all clearly identifiable. Moreover, even when averaging the 530 LFW3D images of George W. Bush, our result retains crisp details and sharp edges despite the extreme variability of the original images, testifying to their aggressive alignment.

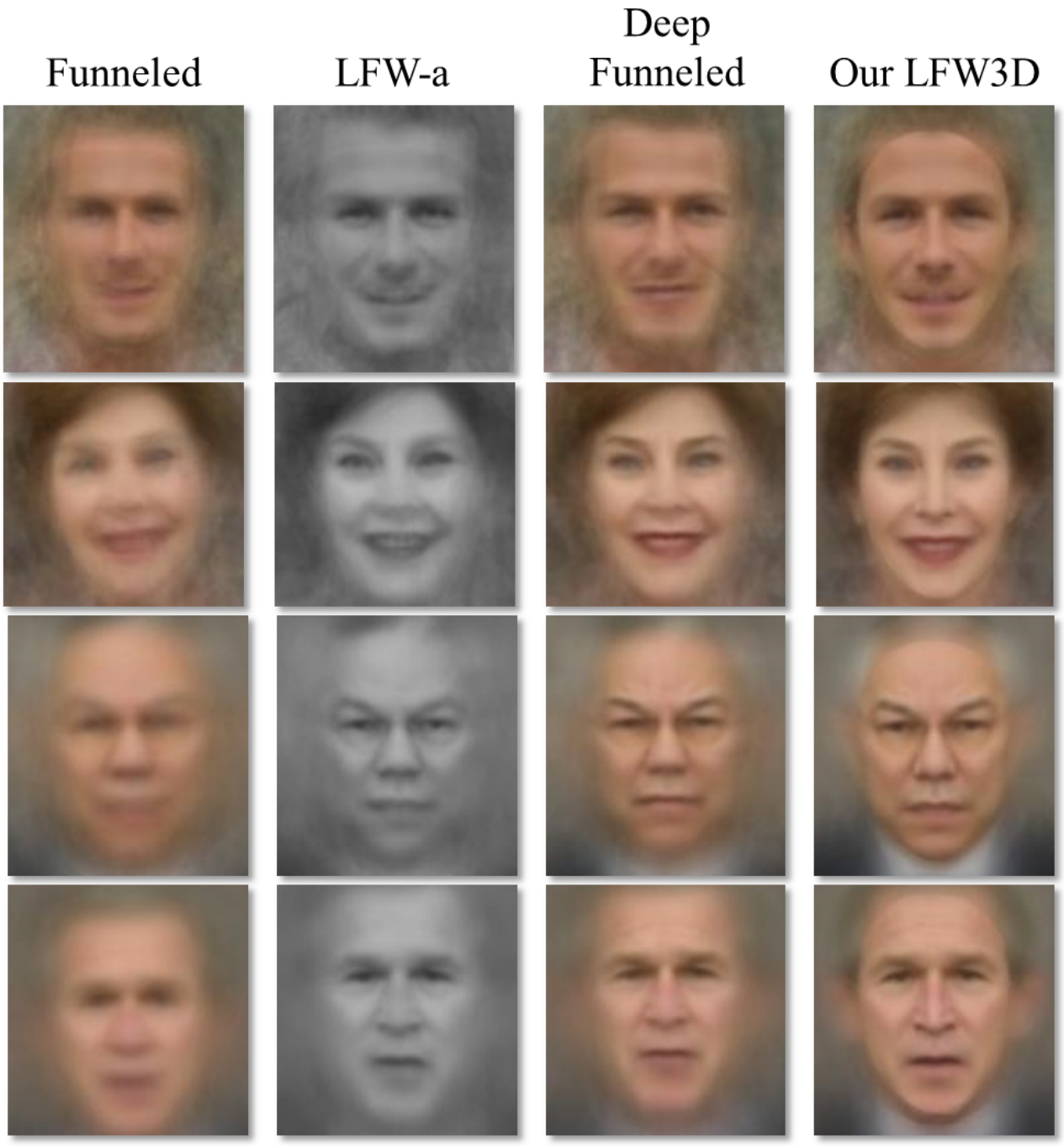

Figure 3.7: **Mean faces with different alignment methods.** Average faces from the 31 David Beckham, 41 Laura Bush, 236 Colin Powell, and 530 George W. Bush images in the LFW set. From left to right, columns represent different alignments: The Funneling of [5], the LFW-a images (available only in grayscale) [1], the deep-funneled images of [6] and our own frontalized faces. Wrinkles on the forehead of George W Bush in our result are faintly visible. These were preserved despite having been averaged from 530 images captured under extremely varying conditions.

Chapter 4

Experiments

Our method was implemented entirely in MATLAB, using the "renderer" function to render a reference view and produce the 2D-3D correspondences of Eq. 3.1 and the "calib" function to estimate the projection matrix $\mathbf{C_Q}$, both functions available from [3]. In all our experiments, we used the 3D face geometry used by [3], taken from the USF Human-ID database collection [36]. Facial feature detection was performed using the SDM method [27], with their own implementation out-of-the-box. Its running time is approximately .04 seconds. Following detection, frontalization (including pose estimation) took an additional $\sim$0.1 seconds on 250×250 pixel color images. These times measured on a standard Windows machine with an Intel i5 core processor and 8Gb RAM.

4.1 Qualitative results

Front-facing new views of Labeled Faces in the Wild images are provided throughout this paper. These were selected to show how our frontalization affects faces of varying age, gender, and ethnic backgrounds, as well as varying poses, occlusions, and more. We additionally compare our results with the two most relevant previous methods. Fig. 3.1 and 3.6 present results obtained using the code from [3]. It was not designed specifically for frontalization, and so front facing views were manually produced. Fig. 3.6 additionally provides a comparison with [4].

	Funneled		LFW-a		LFW3D	
Method	Values	Values Sqrt	Values	Values Sqrt	Values	Values Sqrt
LBP	0.6767	0.6782	0.6824	0.6790	0.7465 ± 0.0053 (0.80)	0.7322 ± 0.0061 (0.79)
TPLBP	0.6875	0.6890	0.6926	0.6897	0.7502 ± 0.0055 (0.81)	0.6723 ± 0.0323 (0.72)
FPLBP	0.6865	0.6820	0.6818	0.6746	0.7265 ± 0.0143 (0.80)	0.7345 ± 0.0061 (0.81)
OSS LBP	0.7343	0.7463	0.7663	0.7820	0.8088 ± 0.0123 (0.87)	0.8052 ± 0.0106 (0.87)
OSS TPLBP	0.7163	0.7226	0.7453	0.7514	0.8022 ± 0.0054 (0.87)	0.7983 ± 0.0066 (0.87)
OSS FPLBP	0.7175	0.7145	0.7466	0.7430	0.7852 ± 0.0057 (0.86)	0.7822 ± 0.0049 (0.85)
Hybrid*	0.7847 ± 0.0051		0.8255 ± 0.0031		0.8563 ± 0.0053 (0.92)	
Sub-SML [7]	0.8973 ± 0.0038					
Sub-SML + Hybrid	0.9165 ± 0.0104 (0.92)					

Table 4.1: **Hybrid method verification results on the LFW benchmark.** Accuracy ± standard errors (SE) as well as area under the ROC curve (AUC) reported on the LFW View-2, restricted benchmark. Results for funneled and LFW-a images were taken from [1]. No SE were reported for these methods. "Value" denotes the use of descriptor values directly (i.e., L2 or OSS distances); "Values Sqrt" represents Hellinger and Sqrt-OSS. * Results reported for funneled and LFW-a images were obtained using four representations and 16 similarity scores to our three and 12.

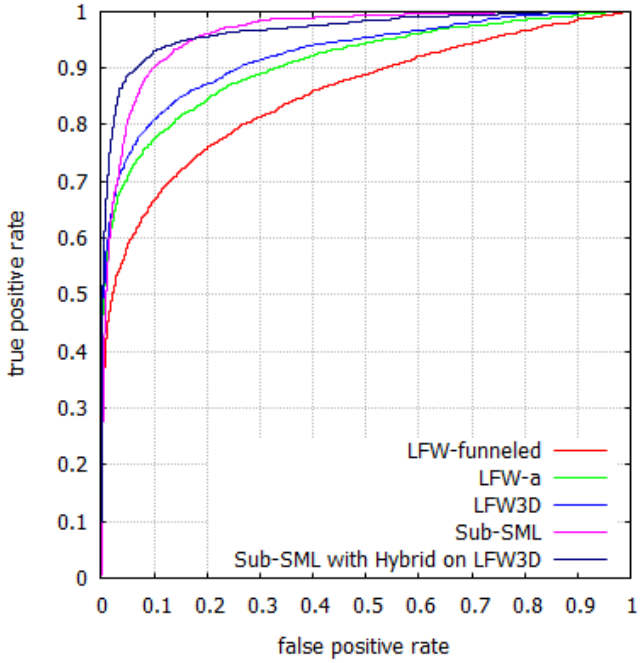

Figure 4.1: **ROC curves for LFW verification results.** Comparing the performance of the Hybrid method [1] on Funneled LFW images, LFW-a and our own LFW3D, as well as the performance reported in Sub-SML [7], and the accuracy obtained by combining both Sub-SML and Hybrid on LFW3D images.

4.2 Face verification on the LFW benchmark

We perform face verification tests on the Labeled Faces in the Wild (LFW) benchmark [9]. Its View-2 test protocol provides ten sets of 600 image pairs. Each one with 300 same person image pairs and 300 not-same pairs. Ten-fold cross validation tests are used taking each set in

turn for testing and the rest for training, along with their ground truth same/not-same labels. Mean ± standard error (SE) over these ten folds are reported as well as area under the ROC curve (AUC). Our tests follow the "Image-Restricted, Label-Free Outside Data" protocol [37]; outside data only used to train the facial feature detector.

We aim to see how much is face recognition performance improved with our frontalized faces. Thus, rather than using recent state-of-the-art methods which may mask the contribution of frontalization, we use the "Hybrid" method [13], one of the first methods developed and successfully tested on LFW. Since then, newer, more modern methods have out-performed it with increasing margins by using better representations and learning techniques. We test how much of this performance gain can be reproduced by simply using better aligned images.

Our implementation uses these three representations: LBP [33, 34], TPLBP and FPLBP [13]. Image descriptors are compared using L2 distance, Hellinger distance [13] (L2 between descriptor element-wise square roots), One-Shot Similarity (OSS) [38] and OSS applied to the descriptors' element-wise square root. In total, we use 3 descriptors × 4 similarities = 12D vector of similarity values, classified by stacking [39] linear SVM classifiers. [40].

We frontalized LFW images as described in Sec. 3. Conditional symmetry (Sec. 3.3) was used here to also reject failed frontalizations: whenever six or more of the eight detectors failed on frontalized images, with and without soft-symmetry, the system defaulted to a planar alignment of the photo, in our case the corresponding deep-funneled images [6]. Of the 13,233 LFW images, ∼2.5% were thus rejected, though more undetected failures exist. In most cases, these were due to occluded or extreme profile faces. In all cases, failures were the result of badly localized facial features; better facial feature detectors would therefore allow for better frontalization.

Results are compared to those reported on the LFW-a collection using a similar system [1]. To our knowledge, these are the best results reported for the same face verification pipeline with an alternative alignment method, presumably optimized for best results. Alternatively, Deep-funneled images can be used instead of LFW-a but its performance gain over LFW-a are small [6].

Table 4.1 lists our results and Fig. 4.1 provides ROC curves. Evidently, our frontalized faces provide a performance boost of over 3% (possibly more, as the original Hybrid method included C1-Gabor descriptors in addition to the three we used). More importantly, these results show that rather than losing information when correcting pose using a single reference model, faces aligned this way are easier to classify than by using appearance preserving, in-plane alignment methods.

We additionally report the performance obtained by combining the Sub-SML method of [7], using their own implementation, with our Hybrid method, computed on frontalized LFW3D images. Sub-SML and Hybrid methods were combined by adding the Sub-SML image-pair similarity scores to the stacking SVM for a total of 13 values used for classification. For comparison, the performance originally reported by [7] is also provided. Adding the Hybrid method with LFW3D provides a 2% accuracy boost, raising the final performance to 0.9165 ± 1.04. To date, this is the highest score reported on the LFW challenge in the "Image-Restricted, Label-Free Outside Data" category.

4.3 Gender estimation on the Adience benchmark

The recently introduced Adience benchmark for gender estimation [2] has been shown to be the most challenging of its kind. It includes 26,580 photos of 2,284 subjects, downloaded from Flickr albums. Unlike LFW images, these images were automatically uploaded to Flickr from iPhone devices without manual filtering. They are thus far less constrained than LFW images. We use the *non*-frontal, version of this benchmark, which includes images of faces in $\pm 45^\circ$ yaw poses. The test protocol defined for these images is 5-fold cross validation tests with album/subject-exclusive splits (images from the same subject or Flickr album appear in only one split). Performance is reported using mean classification accuracy $\pm$ standard errors (SE).

We compare results obtained by the best performing method in [2] on Adience images aligned with their proposed method with our implementation of the same method applied to frontalized Adience images ("Adience3D"). We again use LBP and FPLBP as image representations

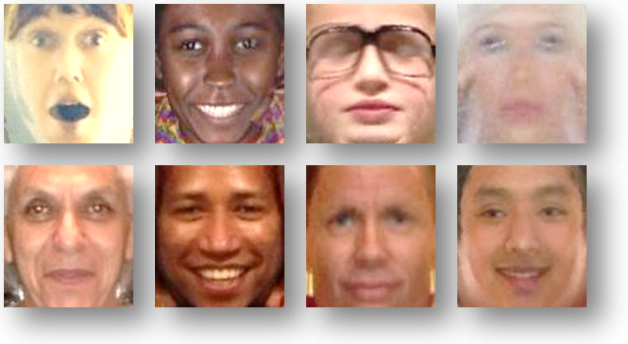

Figure 4.2: **Adience3D gender mis-classifications.** Top: Females classified as males; bottom: males classified as females. Errors result from the absence of clear gender related features or severely degraded images (e.g., top right).

Method	Addience-aligned	Adience3D
LBP	0.734 ± 0.007	0.800 ± 0.007
FPLBP	0.726 ± 0.009	0.753 ± 0.010
LBP+FPLBP+Dropout 0.5	0.761 ± 0.009	0.793 ± 0.008

Table 4.2: **Gender estimation on the Adience benchmark.** Mean accuracy (± standard errors) reported on aligned Adience images [2] and our frontalized Adience3D images.

(results for TPLBP were not reported in [2]). Linear SVM were trained to classify descriptor vectors as belonging to either "male" or "female" using images in the training splits. We also tested training performed using "dropout-SVM" [2] with a dropout rate of 0.5.

Gender estimation results are listed in Table 4.2. Remarkably, frontalization advanced state-of-the-art performance by $\sim 4\%$. Some classification errors are additionally provided in Fig. 4.2. These demonstrate the elevated challenge of the Adience images along with successful frontalizations even with these challenging images.

Chapter 5

Conclusion

Computer vision systems have long since sought effective means of overcoming the many challenges of face recognition in unconstrained conditions. One of the key aspects of this problem is the variability of facial poses. Recently, an attractive, intuitive solution to this has been to artificially change the poses of faces appearing in photos, by generating novel, frontal facing views. This better aligns their features and reduces the variability that face recognition systems must address.

We propose producing such frontalized faces using a simple yet, as far as we know, previously untested approach of employing a single 3D shape, unchanged, with all query photos. We show that despite the use of a face shape which can be very different from the true shapes, the resulting frontalizations lose little of their identifiable features. Furthermore, they are highly aligned, allowing for appearances to be easily compared across faces, despite possibly extreme pose differences in the input images.

Beyond providing a simple and effective means for face frontalization, our work relates to a longstanding debate in computer vision on the role of appearances vs. 3D shape in face recognition. Our results seem to suggest that 3D information, when it is estimated directly from the query photo rather than provided by other means (e.g., stereo or active sensing systems), may potentially damage recognition performance instead of improving it. In the settings explored here, it may therefore be facial texture, rather than shape, that is key to

effective face recognition.

Bibliography

[1] L. Wolf, T. Hassner, and Y. Taigman. Similarity scores based on background samples. In *Asian Conf. Comput. Vision*, 2009.

[2] Eran Eidinger, Roee Enbar, and Tal Hassner. Age and gender estimation of unfiltered faces. *Trans. on Inform. Forensics and Security*, 9(12):2170 – 2179, 2014. Available: `www.openu.ac.il/home/hassner/Adience/data.html`.

[3] Tal Hassner. Viewing real-world faces in 3D. In *Proc. Int. Conf. Comput. Vision*, pages 3607–3614. IEEE, 2013. Available: `www.openu.ac.il/home/hassner/projects/poses`.

[4] Yaniv Taigman, Ming Yang, Marc'Aurelio Ranzato, and Lior Wolf. Deepface: Closing the gap to human-level performance in face verification. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1701–1708, 2013.

[5] Gary B Huang, Vidit Jain, and Erik Learned-Miller. Unsupervised joint alignment of complex images. In *Proc. Int. Conf. Comput. Vision*. IEEE, 2007.

[6] Gary Huang, Marwan Mattar, Honglak Lee, and Erik G Learned-Miller. Learning to align from scratch. In *Neural Inform. Process. Syst.*, pages 764–772, 2012.

[7] Qiong Cao, Yiming Ying, and Peng Li. Similarity metric learning for face recognition. In *Proc. Int. Conf. Comput. Vision*, pages 2408–2415. IEEE, 2013. Available: `empslocal.ex.ac.uk/people/staff/yy267/software.html`.

[8] P Jonathon Phillips, W Todd Scruggs, Alice J OToole, Patrick J Flynn, Kevin W Bowyer, Cathy L Schott, and Matthew Sharpe. FRVT 2006 and ICE 2006 large-scale results. *National Institute of Standards and Technology, NISTIR*, 7408, 2007.

[9] Gary B. Huang, Manu Ramesh, Tamara Berg, and Erik Learned-Miller. Labeled faces in the wild: A database for studying face recognition in unconstrained environments. University of Massachusetts, Amherst, TR 07-49, 2007.

[10] Chaochao Lu and Xiaoou Tang. Surpassing human-level face verification performance on LFW with gaussianface. *arXiv preprint arXiv:1404.3840*, 2014.

[11] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation from predicting 10,000 classes. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1891–1898. IEEE, 2014.

[12] Karen Simonyan, Omkar M Parkhi, Andrea Vedaldi, and Andrew Zisserman. Fisher vector faces in the wild. In *Proc. British Mach. Vision Conf.*, page 7, 2013.

[13] Lior Wolf, Tal Hassner, and Yaniv Taigman. Descriptor based methods in the wild. In *post-ECCV Faces in Real-Life Images Workshop*, 2008.

[14] Lior Wolf, Tal Hassner, and Yaniv Taigman. Effective unconstrained face recognition by combining multiple descriptors and learned background statistics. *Trans. Pattern Anal. Mach. Intell.*, 33(10):1978–1990, 2011.

[15] Junlin Hu, Jiwen Lu, and Yap-Peng Tan. Discriminative deep metric learning for face verification in the wild. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 1875–1882. IEEE, 2014.

[16] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Hybrid deep learning for face verification. In *Proc. Int. Conf. Comput. Vision*, pages 1489–1496. IEEE, 2013.

[17] Yi Sun, Xiaogang Wang, and Xiaoou Tang. Deep learning face representation by joint identification-verification. *arXiv preprint arXiv:1406.4773*, 2014.

[18] Zhenyao Zhu, Ping Luo, Xiaogang Wang, and Xiaoou Tang. Recover canonical-view faces in the wild with deep neural networks. *arXiv preprint arXiv:1404.3543*, 2014.

[19] V. Blanz, K. Scherbaum, T. Vetter, and H.P. Seidel. Exchanging faces in images. *Comput. Graphics Forum*, 23(3):669–676, 2004.

[20] V. Blanz and T. Vetter. Morphable model for the synthesis of 3D faces. In *Proc. ACM SIGGRAPH Conf. Comput. Graphics*, pages 187–194, 1999.

[21] Hao Tang, Yuxiao Hu, Yun Fu, Mark Hasegawa-Johnson, and Thomas S Huang. Real-time conversion from a single 2d face image to a 3D text-driven emotive audio-visual avatar. In *Int. Conf. on Multimedia and Expo*, pages 1205–1208. IEEE, 2008.

[22] Fei Yang, Jue Wang, Eli Shechtman, Lubomir Bourdev, and Dimitri Metaxas. Expression flow for 3D-aware face component transfer. *ACM Trans. on Graphics*, 30(4):60, 2011.

[23] I. Kemelmacher-Shlizerman and R. Basri. 3D face reconstruction from a single image using a single reference face shape. *Trans. Pattern Anal. Mach. Intell.*, 33(2):394–405, 2011.

[24] R. Dovgard and R. Basri. Statistical symmetric shape from shading for 3D structure recovery of faces. *European Conf. Comput. Vision*, pages 99–113, 2004.

[25] Daniel González-Jiménez and José Luis Alba-Castro. Symmetry-aided frontal view synthesis for pose-robust face recognition. In *Int. Conf. on Acoustics, Speech and Signal Processing*, volume 2. IEEE, 2007.

[26] Tal Hassner, Liav Assif, and Lior Wolf. When standard RANSAC is not enough: cross-media visual matching with hypothesis relevancy. *Machine Vision and Applications*, 25(4):971–983, 2014.

[27] Xuehan Xiong and Fernando De la Torre. Supervised descent method and its applications to face alignment. In *Proc. Conf. Comput. Vision Pattern Recognition*, pages 532–539. IEEE, 2013. Available: `www.humansensing.cs.cmu.edu/intraface/index.php`.

[28] Chen Cao, Qiming Hou, and Kun Zhou. Displaced dynamic expression regression for real-time facial tracking and animation. *ACM Trans. on Graphics*, 33(4), 2014.

[29] Jason M Saragih, Simon Lucey, and Jeffrey F Cohn. Real-time avatar animation from a single image. In *Int. Conf. on Automatic Face and Gesture Recognition*, pages 117–124. IEEE, 2011.

[30] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[31] Kiriakos N Kutulakos and Steven M Seitz. A theory of shape by space carving. *Int. J. Comput. Vision*, 38(3):199–218, 2000.

[32] Gang Zeng, Sylvain Paris, Long Quan, and François Sillion. Progressive surface reconstruction from images using a local prior. In *Proc. Conf. Comput. Vision Pattern Recognition*, volume 2, pages 1230–1237. IEEE, 2005.

[33] Timo Ojala, Matti Pietikäinen, and Topi Mäenpää. A generalized local binary pattern operator for multiresolution gray scale and rotation invariant texture classification. In *ICAPR*, 2001.

[34] Timo Ojala, Matti Pietikainen, and Topi Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *Trans. Pattern Anal. Mach. Intell.*, 24(7):971–987, 2002.

[35] Leslie G Farkas. *Anthropometry of the head and face in medicine*. Elsevier New York, 1981.

[36] USF. DARPA Human-ID 3D Face Database:. Courtesy of Prof. Sudeep Sarkar, University of South Florida, Tampa, FL.

[37] Gary B Huang and Erik Learned-Miller. Labeled faces in the wild: Updates and new reporting procedures. University of Massachusetts, Amherst, UM-CS-2014-003, 2014.

[38] Lior Wolf, Tal Hassner, and Yaniv Taigman. The one-shot similarity kernel. In *Proc. Int. Conf. Comput. Vision*, pages 897–902. IEEE, 2009.

[39] David H Wolpert. Stacked generalization. *Neural networks*, 5(2):241–259, 1992.

[40] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995.