Conformal mapping of a 3D face representation onto a 2D image for CNN based face recognition

Josef Kittler, Paul Koppen, Philipp Kopp, Patrik Huber
Centre for Vision, Speech and Signal Processing
University of Surrey, Guildford, UK
{j.kittler, p.koppen, p.huber}@surrey.ac.uk, philipp@koppmaps.de

Matthias Rätsch
Department of Mechatronics
Hochschule Reutlingen, Reutlingen, Germany
matthias.raetsch@reutlingen-university.de

Abstract

Fitting 3D Morphable Face Models (3DMM) to a 2D face image allows the separation of face shape from skin texture, as well as correction for face expression. However, the recovered 3D face representation is not readily amenable to processing by convolutional neural networks (CNN). We propose a conformal mapping from a 3D mesh to a 2D image, which makes these machine learning tools accessible by 3D face data. Experiments with a CNN based face recognition system designed using the proposed representation have been carried out to validate the advocated approach. The results obtained on standard benchmarking data sets show its promise.

1. Introduction

During the last three years major advances towards unconstrained face recognition have been reported. The recent successes have been attributed to two main factors: the development and application of deep neural networks for the extraction of more powerful face representation to support face matching, and the availability of large face databases containing an order of million face images to serve as training sets for machine learning.

Although faces are 3D objects, and as such 3D face shape and skin texture provide richer information than their 2D projections, surprisingly the impressive progress in the field is limited to 2D face images only. There are two main reasons for this. In contrast to 2D images, which are naturally amenable to convolutional processing, 3D faces are commonly defined on a triangular mesh or an isomap, which preclude the application of convolution operators. There is also the data issue. There are no 3D face datasets of comparable size to those of 2D to achieve successful machine learning.

Potentially, the lack of 3D data can be rectified by fitting a 3D morphable face model to 2D face images. This approach would provide access to the same databases as those used for training 2D face recognition systems. In the past this solution was not computationally feasible because the fitting process required a prohibitive amount of time. Moreover, the 3D face reconstruction quality was rather limited due to the poor performance of facial landmarking algorithms. However, the recent progress in facial landmark detection and in algorithmic acceleration of the fitting process offers much enhanced quality of the 3D face reconstruction at realistic time scales and renders 3D face model to 2D face image fitting a realistic proposition.

The outcome of the fitting process is a 3D shape reconstruction of the input 2D image. We could also recover the skin texture model, but this invariably results in loss of information, and it is preferable to use the original face texture for further analysis.

The reconstructed 3D face is commonly represented as a mesh of vertices with associated 3D coordinates to represent 3D shape, and R, G, B values to convey skin texture information. The recovered pose and shape can be used for geometric normalisation of the 2D image and potentially for its illumination correction. Note that, in the case of 2D face image, the texture jointly captures face shape and skin properties. Thus 2D geometric normalisation methods do not destroy the discriminatory information conveyed by shape. In the case of 3D, the fitting of a registered 3D face model separates shape from surface texture. Consequently, the shape free texture image of a 3D face is a weaker source of in-

0-7695-6403-8/18/31.00 ©2018 IEEE
DOI 10.1109/ICB2018.2018.00029

formation for face recognition than 2D image texture. In addition, the surface texture is represented on a mesh, commonly unwrapped into a so called isomap. This irregular sampling structure inhibits the use of standard convolution operators to compute powerful DNN features as in the case of 2D images.

The aim of this paper is to address the issues relating to processing 3D face data. We propose two main contributions.

1. The loss of shape information from the recovered surface texture is compensated by augmenting the 3D face representation by the 3D shape data. Thus each vertex becomes 6 dimensional variable representing the x,y,z coordinates and the corresponding R,G,B values.

2. To facilitate the use of conventional DCNN for further processing we remap the isomap on a rectangular grid structure. The proposed approach decouples shape and texture properties of the input image, but retains all the discriminatory information about each subject.

The proposed scheme is evaluated on the IJB-A data set and compared to 2D processing results: We conduct recognition experiments using DCNN using texture only, shape only and combined shape and texture information. From the results it follows that there is discriminatory information in the recovered 3D shape, but it is less powerful than the shape free texture information. Not surprisingly, the shape-free texture information is a less powerful representation than the 2D texture. However, the combination of shape and texture in the decoupled form provided by the 3D model fitting appears to outperform the conventional 2D face approach.

The rest of the paper is organised as follows. In the next section we review the related work. Section 3 describes the 3D fitting method used in our work. The proposed shape and texture representation of the fitted 3D face is presented in Section 4. Section 5 describes the face recognition engine based on deep neural network features extracted from the advocated representation. The proposed approach is evaluated in Section 6. The paper is drawn to conclusion in Section 7.

2. Related work

The idea of tackling the challenges of 2D face recognition with the help of prior knowledge in the form of a 3D morphable face model has been a moot point since the publication of the seminal paper of Blanz and Vetter [1]. The prospect of principled handling of the challenges posed by illumination, expression and pose with the help of 3D morphable face models has attracted increasing attention. The early attempts, discussed in [2] reviewing 3D imaging, 3D face modelling and recognition approaches, include [1] and [3]. The benefit of 2D face image frontalisation using 3DMM has been demonstrated in [4] in the context of the Face Recognition Grand Challenge organised by NIST in 2004-5.

More recent attempts include Niinuma et al.'s use of 3DMM for multiview 2D face recognition [5]. In their approach, a 3DMM model is fitted to each frontal gallery face image and used to augment the gallery set by images of different poses. For an input 2D face image, its pose is estimated and a subset of gallery images of similar pose used for matching.

As 3DMM face fitting using conventional algorithms is time consuming, Taigman et al. [6] fit only the mean 3D face shape, but correct the estimated 3D shape coordinates by vertex specific x, y fitting error residuals before rendering a frontalised face image for deep neural network analysis. The use of a generic 3D face shape model is also advocated in [7] and by Ding et al [8], who use the frontalised 2D face image for a patch based self-occlusion aware face matching.

In [9], Hassner et al. argue that fitting a generic 3D shape face model is quite sufficient for recognition. However, for handling a wider range of variations, and especially expressions, 3DMM is the model of choice [10], [11]. The computational complexity of the 3DMM fitting has recently been addressed in [12] and in [13].

Most of the reviewed papers use a 3D model for 2D face image frontalisation [4], [6], [8]–[11] or other forms of pose regularisation [5]. A few papers attempt to make use of the estimated 3D shape and texture information for face matching. The exceptions include [12], [14], [15] where the recovered 3D face shape and texture parameters constitute the features for pose and illumination invariant face recognition. However, these features are just PCA coefficients, whose discriminatory power is not comparable to deep learning features used in 2D face recognition.

The aim of this paper is to investigate the possibility of using 3D reconstructed 2D faces for face recognition. This requires a more powerful representation that retains both 3D shape and texture information, as well as the ability to extract powerful features using convolutional neural networks. This challenge is addressed in the following sections.

3. 3D face model fitting

3.1. 3D Morphable Models

3D Morphable Face Models (3DMM) are statistical models constructed on 3D meshes of vertices conveying x, y, z coordinates of the 3D face surface and the corresponding R, G, B values of the surface texture. First introduced by Blanz & Vetter [1], they are built from real 3D scans of people. After scanning, the meshes are brought into dense correspondence. This means the vertices are as-

signed a specific semantic position, for example at the nose tip, over all scans. Also all scans now have the same number of vertices N. The model is separated in shape and colour, although for the approaches presented in this paper, only the shape model is used. With a principle component analysis the meshes can be split into the mean $\bar{\mathbf{e}} \in \mathbb{R}^{3N}$ and the principle components $\mathbf{E} = [\mathbf{e}_1, \ldots, \mathbf{e}_M] \in \mathbb{R}^{3N \times M}$, capturing the statistical variations in the data. M is the number of principal components. Additionally, to cope with different expressions, a set of expression blendshapes $\mathbf{B}$ of size K has been added to the shape model. A face shape $\mathbf{S}$ can then be represented as:

$$\mathbf{S} = \bar{\mathbf{e}} + \sum_{i}^{M} \boldsymbol{\alpha}_i \mathbf{e}_i + \sum_{j}^{K} \psi_j \mathbf{B}_j, \qquad (1)$$

where $\boldsymbol{\alpha}$ and ψ are the coefficients of the shape components and the blendshapes and i and j the indices, respectively.

3.2. Linear Landmark Fitting

In order to use a 3DMM for face analysis or as a prepossessing step for face recognition it first has to be fitted. Fitting means finding the shape, blendshape and camera parameters that describe the person's face in a specific image. For this purpose we use the fitting algorithm introduced by Huber et al. [13], [16].

It uses given face landmarks for a real time shape fitting. With the 2D landmark locations and their known correspondences in the 3D Morphable Model, the pose of the camera is estimated.

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

$$\mathbb{E} = [(\mathbf{y}_p(\boldsymbol{\alpha}) - \mathbf{y}]^T \Omega^{-1} [(\mathbf{y}_p(\boldsymbol{\alpha}) - \mathbf{y}] + \lambda \|\boldsymbol{\alpha}\|_2^2 \qquad (2)$$

where $\mathbf{y}$ is a stacked vector of 2D landmarks, Ω is a diagonal matrix of variances of the landmark points, and $\mathbf{y}_p(\boldsymbol{\alpha})$ is a stacked vector of the 3D Morphable Model shape points that correspond to the respective 2D landmarks, projected to 2D using the estimated camera parameters.

Similar to the shape coefficients $\boldsymbol{\alpha}$, the blendshape coefficients $\boldsymbol{\psi}$ are found with a standard least-squares formulation. Instead of using the mean shape $\bar{\mathbf{v}}$ in the previous formulation, it is substituted with a face instance $\mathbf{S}(\boldsymbol{\alpha})$, generated with the currently estimated $\boldsymbol{\alpha}$.

The outer face contours are very important for an accurate face reconstruction, as they define the border between the face and the background. However, the outer face contours present in the 2D image do not correspond to unique contours on the 3D model, thus fixed correspondences between landmarks and vertices in the mesh cannot be used. Additionally the front and the back-facing occluded contour have to be handled differently as the back-facing contour strongly depends on the pose.

All the previous steps are processed iteratively. The fitting is initialised by computing a rough pose estimate using only the inner face landmarks (i.e. excluding all contour landmarks), followed by an initial estimate of the expressions. Then, the components of the contour fitting process are applied to get additional correspondences for both the front-facing as well as the occluded face contour. Subsequently, the pose is re-estimated using all landmarks, including the contour landmarks, followed by solving for PCA shape identity coefficients and blendshapes. These four main components are iterated towards convergence.

When more than one frame, or image, of the same subject are available, as in video, all the frames are used jointly in the fitting process. This leads to the recovery of a single set of shape parameters, given all images.

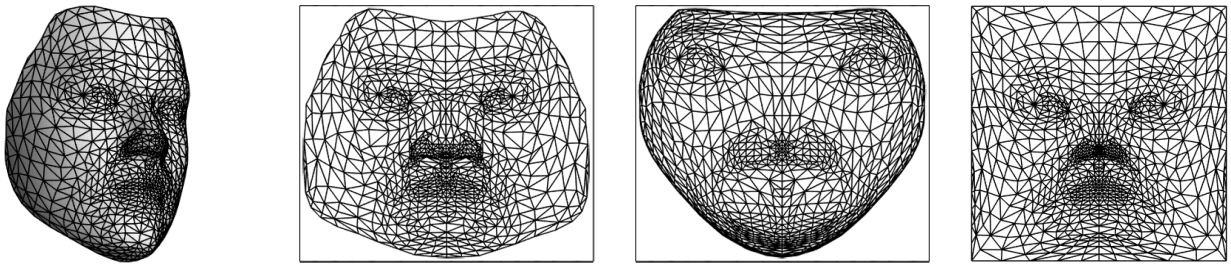
3.3. 3D Shape and Texture Representation

The reconstructed 3D shape obtained by fitting 3DMM to an input image is defined in terms of the shape parameter vector $\boldsymbol{\alpha}$. Although the shape vector provides a concise representation of the 3D shape of the 2D face, in this paper we argue that in certain circumstances it is convenient to work with a long hand alternative, defined in terms of actual 3D surface points. In particular, we can represent the surface at each vertex of the mesh by the x, y, z coordinates of the corresponding point on the face.

The linear landmark fitting uses the shape model only and does not fit any texture. Nevertheless it is possible to extract the texture information from the original image by sampling it at the mesh vertex points projected into the 2D image. This sampling provides the R, G, B values of the surface texture. In summary, the fitted 3D mesh produces a joint shape and texture representation of the input 2D face conveyed in terms of x, y, z coordinates of the shape and the associated R, G, B values of the texture for every vertex of the mesh.

4. A Structured Mesh Representation for Convolution

We use the following notation. A surface mesh $\boldsymbol{M} = (V, F)$ describes the face shape explicitly over a set of vertices V, and implicitly in between the vertices over the triangles $\mathbf{F}$, with $\mathbf{f}_i = (j, k, l)$ a triangulation between vertices j, k and l. Every vertex stores a spatial coordinate, $\mathbf{x}_i = (x_i, y_i, z_i)$, as well as a texture coordinate, $\mathbf{u}_i = (u_i, v_i)$, with $0 \leq u_i, v_i \leq 1$. While the spatial coordinates are usually directly interpolated, texture colour val-



(a) 3D face shape (b) Isomap (c) Laplacian Eigenmap (d) Constrained Laplacian

Figure 1: Different mesh parameterisation methods result in different 2D embeddings. Isomap computes a quasi-isometric mapping that preserves pairwise distances. The Laplacian Eigenmap is conformal, preserving angles and area. It is possible to pose constraints on the boundary vertices. This causes the internal vertices to map to the interior of its convex hull. In the example on the right, the boundary vertices were constrained to a square. The mapping is still conformal.

ues are looked up in the texture image at interpolated UV coordinates.

In essence, the UV coordinates form a 2D embedding of the 3D vertex coordinates, which allows us to store other modalities, such as texture, more efficiently in an image form. But it is not limited to that. In fact, just as the texture is stored as an image, we could also store 3D spatial data (x, y, z), instead of (r, g, b), as a "shape image". Under certain conditions, which we will come to below, this would bring image processing techniques, such as deep convolutional networks, to the 3D mesh.

The face mesh, however, is incompatible with a regular image in three respects: 1) the facial surface is three-dimensional (although it is intrinsically a 2D structure). 2) the face surface is not rectangular. 3) the mesh connectivity is unstructured: some vertices have more neighbours than others, whereas standard images are regular with pixels in a fixed grid.

We will now describe a procedure that takes an irregular 3D surface mesh and conforms it to a regular rectangular grid. Although we will use the shape data as a guideline, it may be interesting to note that the method can equally applied to *any* function over the mesh, such as surface normals, curvature, etc.

4.1. 2D mesh parameterisation

The goal is to find a 2D embedding (parameterisation) of the vertices that preserves most of the surface structure. For example, in the 2D space, triangles cannot overlap. Well known algorithms for this are the Isomap [17] and Laplacian Eigenmaps [18]. In particular the latter method and variations based on harmonics of the surface have been studied extensively in the field of differential geometry [19]–[23]. An example of the embeddings is shown in Figure 1. The last figure shows how boundary constraints can be defined on the Laplacian so that the result is a rectangular planar mesh. We now describe its derivation.

Let us denote by E the set of all edges (pairs of vertices (i, j) that share at least one triangle). Then the set of nodes adjacent to vertex i form its 1-neighbourhood, $N(i) = \{j|(i, j) \in E\}$. In matrix-form this is the adjacency graph

$$A_{ij} = \begin{cases} 1 & (i,j) \in E \\ 0 & \text{otherwise,} \end{cases} \tag{3}$$

which is a sparse symmetric matrix of size $N \times N$. When we further define D to be the diagonal matrix with entries $D_{ii} = |N(i)|$, then

$$L = D - A \tag{4}$$

is the (discrete) graph Laplacian of mesh $\boldsymbol{M}$.

Note that L is rank deficient: the diagonal elements of D are the row sums of A and so $\text{rank}(L) = N - k$, where k is the number of connected components in the mesh. As we are dealing with a single surface mesh, here $k = 1$.

The Laplacian Eigenmap is obtained from the solutions of the linear system $L\mathbf{v} = \lambda\mathbf{v}$. Ordering the eigenvalues of the solutions such that $\lambda_0 \leq \lambda_1 \leq \lambda_2 \leq \ldots$, and noting that $\lambda_0 = 0$, then the 2D embedding coordinates $\mathbf{u}$ of the 3D vertex positions $\mathbf{x}$ are found directly as $\mathbf{u} = [\mathbf{v}_1, \mathbf{v}_2]$.

As L depends solely on the mesh connectivity, and not on the 3D vertex positions, the embedding is geometry agnostic. A generally better embedding is obtained when using informed edge weights in matrix A. In particular the *cotangent weights* [19], [24] have been proven to approximate the continuous Laplace-Beltrami operator on smooth surfaces. Here $A_{ij} = \cot(\alpha_{ij}) + \cot(\beta_{ij})$ where α_{ij} and β_{ij} denote the two angles opposite of edge (i, j). We should also update $D_{ii} = \sum_j A_{ij}$. Figure 1c shows an example of the result.

As mentioned above, the objective is an embedding that covers precisely a square. This is achieved by constraining

the solution for vertices on the mesh boundary as follows. Without loss of generality we assume that the first m vertices lie on the boundary and vertices $i = (m+1) \dots N$ are internal. We now build a new system of equations with the first m rows in L replaced by the identity matrix:

$$\left(\frac{I_{m\times m} \mid \mathbf{0}}{L_{(m+1)\dots N}}\right)\mathbf{u} = \begin{pmatrix}\mathbf{u}_{1\dots m}\\ \mathbf{0}\end{pmatrix} \tag{5}$$

The right-hand side provides the boundary constraints in the first m rows followed by $N - m$ rows of zeros. The least-squares solution for $\mathbf{u}$ gives the required embedding. An example embedding is shown in Figure 1d.

4.2. Regular sampling

An important result of the previous section is that the UV space forms the domain for *any* function over the mesh. In particular the mesh shape is now a (vector-valued) function, $\mathbf{x} : f(\mathbf{u}) \subset \mathbb{R}^3$. This function is known at the given points (u_i, v_i). Transforming the irregular connectivity of the mesh to a regular grid connectivity (as pixels in images) thus consists of yielding values at regularly spaced points (u, v) on the unit square.

By default, meshes are treated as piecewise linear, yielding linearly interpolated values between the three vertices of an enveloping triangle. With a dense sample of vertices on the facial surface this may be sufficiently accurate. Other interpolation methods have nicer properties, however, and may give better results on areas with fewer given points. Examples include inverse distance weighting and natural neighbour interpolation which are both once differentiable, and spline interpolation which has a continuous second derivative. Also Gaussian Processes can be used.

Figure 2b shows a regularly sampled mesh extracted from an input image. The mesh is composed of two independent maps: the texture and the shape image. Vertex connectivity is now implicit in the grid structure.

In view of the remainder of this paper, we further note that all face meshes are registered. This means that the UV coordinates have a *consistent* interpretation. If, e.g. coordinate $(u, v) = (0.5, 0.5)$ is the tip of the nose, then it is so across all images. This makes the texture map especially suitable for processing with further algorithms.

5. Recognition Engine

Based on the texture map and the shape map introduced in the previous section we wish to extract an identity specific feature vector for face recognition. We use CNNs inspired by the progress made in recent years.

The shape and texture maps can either be used as sole inputs, or be stacked on top of each other. In the latter case, adopted here, the filters in the first convolutional layer have a depth of 6: three for (r, g, b) texture values and another three for (x, y, z) shape values. In either case, the mean input over the entire training set is subtracted.

During training the last fully connected layer has the length equal to the number of identities in the training set. A L2 or Softmax-Loss is then calculated and the error propagated through the network to update the weights.

At testing stage, the second to last layer is used as a feature vector for decision making in the last layer. If a template for more than one image is needed, a feature vector is extracted with a CNN for each image separately and then averaged.

For matching two feature vectors we use the cosine angle between the two feature vectors $\boldsymbol{\gamma}_1$ and $\boldsymbol{\gamma}_2$,

$$d = \frac{\langle \boldsymbol{\gamma}_1, \boldsymbol{\gamma}_2 \rangle}{\|\boldsymbol{\gamma}_1\| \cdot \|\boldsymbol{\gamma}_2\|}. \tag{6}$$

6. Experiments

We conduct face recognition experiments to validate the proposed approach. We put special focus on shape and texture representations and how these relate to one another. The CNN architecture described in Section 5 was used with an input resolution of 256×256 pixels for both texture and shape.

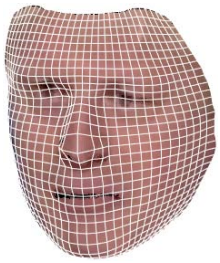
6.1. Datasets

For training and evaluation of the CNNs and the testing of the entire approach the following datasets were used. For landmarking we use a Random Cascaded-Regression introduced by Feng et al. [25].

Casia Webface [26] (training)
With 494,414 images of 10,575 subjects, Casia Webface is among the biggest publicly available datasets for face recognition. The images are all gathered from the Internet Movie Database website (IMDb) and highly unconstrained in lighting and pose although faces are automatically detected. To obtain the shape information we apply the multi-image fitting using all images of the subject.

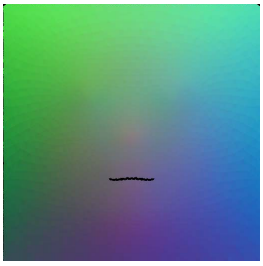
PaSC – Point and Shoot Challenge [27] (training and evaluation)
This dataset consists of two parts. First a set of 9,376 still images of 293 subjects and second 255,100 frames in 2,802 videos of 265 subjects. The subjects in the videos are a subset of the group shown in the still images. The data was recorded in 9 different settings with low resolution and shaky hand-held consumer cameras as well as high-resolution cameras on a tripod. Faces are detected automatically. Because of the videos, PaSC comes with a much higher number of frames and images per subject in comparison to Casia Webface. For the multi-image fitting we use all frames in a video. As there are several videos for each subject we now have more than one shape estimation

(a) Input image



(b) Extracted mesh (regularised)

(c) Texture map



(d) Shape map

Figure 2: Example of shape and texture extraction onto a regularised mesh. The 3D model if fitted onto the input image *(a)* and regularised. The resulting mesh is rendered in *(b)*, with the independent texture and shape maps in *(c)* and *(d)*. (In the last visualisation, (x, y, z) values were mapped onto the image (r, g, b) channels, thus green means high y and low x and z.)

per subject and the recognition network is able to learn a intra-class variance. We use all still images and 80% of the videos for training. The remaining 20% of the videos are used to evaluate CNN performance while training.

IJB-A – IARPA Janus Benchmark A [28] (testing)
The IJB-A dataset is one of the most challenging datasets for face recognition to date. 46,962 images of 500 different IDs were labelled manually. Annotation mainly included the ID and a bounding box. This means this dataset also includes faces with extreme poses that would not be detected by face detectors. IJB-A focuses on so called templates which can consist of several images. Thus the difficulty lays in the construction of a feature vector that uses information from more than one image.

6.2. Networks

The networks chosen for this task are standard networks. The aim was not to focus on bigger and therefore more powerful networks but rather well understood networks. The first network chosen is an alexNet [29], which was first introduced in 2012, so is rather old but well understood and fast in training. The second network used for the face recognition experiments is called DCNN published with the CASIA-Webface dataset [26]. We denote the original results by the authors as DCNN and our results using the same network structure as dcnn. Table 1 compares alexNet and DCNN.

6.3. Experimental Results

The approaches in this paper are tested on the IJB-A verification challenge [28]. We compare to two baselines, first OpenBR [30], which is an open-source face recognition system and secondly a GOTS face recognition system.

In Figure 3 several different inputs were compared. First we compare two shape representations without using any texture. We match the shape coefficients $\boldsymbol{\alpha}$ of the fitted 3D Morphable Model directly using the cosine angle. No training is needed. As expected, the performance is rather poor due to the lack of texture information.

	alexNet	DCNN
# convolutional layers	5	10
# fully connected layers	3	1
# parameters	60 Mio	5 Mio
length of feature vector	512	320

Table 1: Comparison of alexNet [29] and DCNN [26], the networks used for the face recognition experiments.

As a second shape representation we test the *xyz* shape image, as introduced in Section 4. For this we train an alexNet on the given training set. On the test set we fit the 3DMM, extract the shape image, apply the trained CNN on this and extract a feature vector in the second to last layer. Matching is again conducted based on the cosine angle. Although only the *representation* of the shape information has changed, the performance improves significantly.

Focusing on texture representations, we compare an alexNet trained on the shape independent texture map and an alexNet trained on a facebox aligned by the landmarks of the eyes and the mouth. The texture map outperforms the facebox by a significant margin. With both texture representations the alexNets are better than OpenBR. Using the square texture map we beat the GOTS baseline of the IJB-A paper.

Although both recognition experiments on shape only representations in Figure 3 do not reach any baseline they are better than random. Thus the shape representations contain valuable information that is yet not used in the texture only experiment. Figure 4 shows performance for alexNets with the alpha vector as additional input and a shape map as additional input in comparison with the texture map only.

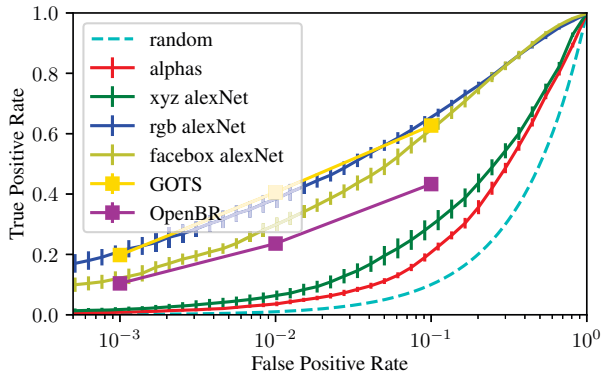


Figure 3: IJB-A verification performance, trained on texture map only, shape map only, or alphas directly.

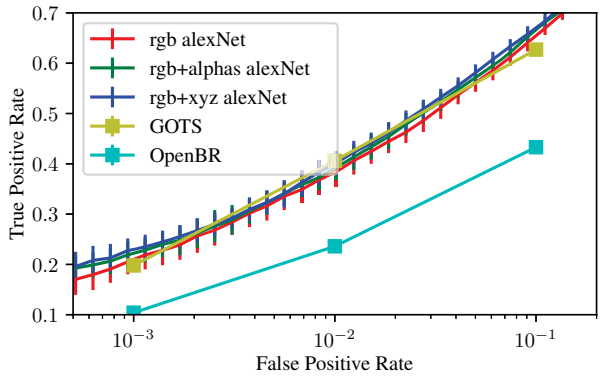


Figure 4: IJB-A verification performance with alexNets on different texture and shape inputs.

If the shape coefficients alphas are used they are fed into the alexNet in the first fully connected layer, that is one layer before the features are extracted during testing. The network is therefore capable of weighting between texture and shape features and also among the shape coefficients.

When we use the shape map, we stack it with the texture map and feed them both to the network as one input. As a result, all CNN layers are used to extract high level features from both texture and shape. Although both approaches using the shape information outperform the texture only network, the experiments show that the shape map encoding is better than using the alphas directly.

The alexNet used in the previous experiments is rather old and not specifically designed for face recognition tasks. Therefore we validate our findings with a newer network especially designed for face recognition, called DCNN [26]. Figure 5 compares two dcnns, one trained on the texture map and the other on texture and shape maps. This Figure also shows the performance of the DCNN as published by the authors without any finetuning and metric learning, trained on Casia Webface. As expected the deeper dcnn greatly outperforms the small alexNet. We are able to achieve comparable results to the original DCNN with both texture only and the combination of texture and shape.

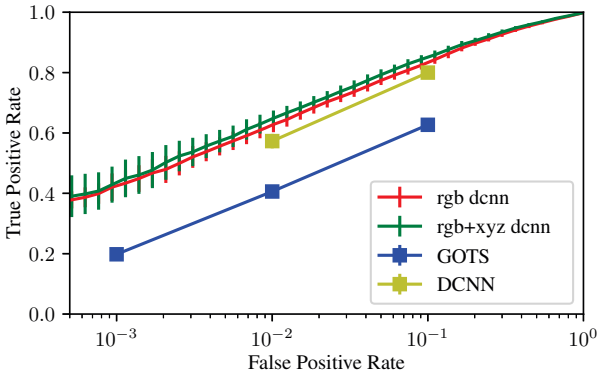


Figure 5: IJB-A verification performance with DCNN on texture only and texture+shape.

7. Conclusion and future work

2D face recognition has made an amazing progress during the last three years thanks to deep neural networks. As 2D face appearance mixes together face shape with skin texture and illumination, it is pertinent to ask whether 3D reconstruction of 2D images obtained by fitting a 3D face model would enable even better performance by virtue of separating shape, texture, expression and lighting. As modern machine learning tools require a huge amount of data for training, the route to extensive 2D face databases via 3D face model fitting could be a viable proposition which 3D assisted approaches could beneficially exploit. However, the conventional 3D face representation in the form of x, y, z and R, G, B measurements on a mesh of vertices is not amenable to CNN processing.

To addressed this problem, we proposed a conformal mapping from a 3D mesh to a 2D image, which makes these machine learning tools accessible by conventional 3D face data representations. A CNN based face recognition system using the proposed representation has been designed and trained on the CASIA and PaSC databases. Its performance was evaluated on the IJB-A database using the standard protocol. The results show the relative merits of shape and texture information. Most importantly, the combined use of shape and texture delivers performance comparable to the more mature 2D based face matching engines exemplified by [26] (without metric learning). This validates the proposed mapping, as well as the advocated 3D assisted approach it enables. The latter is promising, as it offers a better scope for dealing with nuisance factors and data augmentation. These issues will be adressed in future work.

Acknowledgements

The financial support from EPSRC Programme Grant "FACER2VM", reference EP/N007743/1 is gratefully acknowledged.

References

[1] V. Blanz and T. Vetter, "A morphable model for the synthesis of 3d faces," in *SIGGRAPH*, 1999, pp. 187–194.

[2] J. Kittler, A. Hilton, *et al.*, "3d assisted face recognition: A survey of 3d imaging, modelling and recognition approachest," in *CVPR*, 2005, p. 114.

[3] L. Zhang and D. Samaras, "Pose invariant face recognition under arbitrary unknown lighting using spherical harmonics," in *BioAW, ECCV Workshop*, 2004, pp. 10–23.

[4] V. Blanz, P. Grother, *et al.*, "Face recognition based on frontal views generated from non-frontal images," in *CVPR*, 2005, pp. 454–461.

[5] K. Niinuma, H. Han, *et al.*, "Automatic multi-view face recognition via 3d model based pose regularization," in *BTAS*, 2013.

[6] Y. Taigman, M. Yang, *et al.*, "Deepface: Closing the gap to human-level performance in face verification," in *CVPR*, 2014, pp. 1701–1708.

[7] A. Asthana, M. J. Jones, *et al.*, "Pose normalization via learned 2d warping for fully automatic face recognition," in *BMVC*, 2011, pp. 1–11.

[8] C. Ding, C. Xu, *et al.*, "Multi-task pose-invariant face recognition," *IEEE TIP*, pp. 980–993, 2015.

[9] T. Hassner, S. Harel, *et al.*, "Effective face frontalization in unconstrained images," in *CVPR*, 2015, pp. 4295–4304.

[10] B. Chu, S. Romdhani, *et al.*, "3d-aided face recognition robust to expression and pose variations," in *CVPR*, 2014, pp. 1907–1914.

[11] X. Zhu, Z. Lei, *et al.*, "High-fidelity pose and expression normalization for face recognition in the wild," in *CVPR*, 2015, pp. 787–796.

[12] G. Hu, F. Yan, *et al.*, "Efficient 3d morphable face model fitting," *Pattern Recognit.*, pp. 366–379, 2017.

[13] P. Huber, P. Kopp, *et al.*, "Real-Time 3d Face Fitting and Texture Fusion on In-the-Wild Videos," *IEEE Signal Process. Lett.*, pp. 437–441, 2017.

[14] V. Blanz and T. Vetter, "Face recognition based on fitting a 3d morphable model," *IEEE TPAMI*, pp. 1063–1074, 2003.

[15] G. Hu, C. Chan, *et al.*, "Robust face recognition by an albedo based 3d morphable model," in *IJCB*, 2014.

[16] P. Huber, G. Hu, *et al.*, "A Multiresolution 3d Morphable Face Model and Fitting Framework," in *VISAPP*, 2015.

[17] J. B. Tenenbaum, V. d. Silva, *et al.*, "A global geometric framework for nonlinear dimensionality reduction," *Science*, pp. 2319–2323, 2000.

[18] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, pp. 1373–1396, 2003.

[19] U. Pinkall and K. Polthier, "Computing discrete minimal surfaces and their conjugates," *Exp. Math.*, pp. 15–36, 1993.

[20] M. Desbrun, M. Meyer, *et al.*, "Intrinsic Parameterizations of Surface Meshes," *Comput. Graph. Forum*, pp. 209–218, 2002.

[21] B. Lévy, S. Petitjean, *et al.*, "Least squares conformal maps for automatic texture atlas generation," *ACM TOG*, pp. 262–371, 2002.

[22] O. Sorkine, "Laplacian Mesh Processing," *Eurographics - State Art Reports*, pp. 53–70, 2005.

[23] R. Zayer, B. Lévy, *et al.*, "Linear angle based parameterization," *Eurographics Symp. Geom. Process.*, pp. 135–141, 2007.

[24] M. Meyer, M. Desbrun, *et al.*, "Discrete Differential-Geometry Operators for Triangulated 2-Manifolds," in *Vis. Math. III*, 2003, ch. I-2, pp. 35–57.

[25] Z.-H. Feng, G. Hu, *et al.*, "Cascaded collaborative regression for robust facial landmark detection trained using a mixture of synthetic and real images with dynamic weighting," *IEEE TIP*, pp. 3425–3440, 2015.

[26] D. Yi, Z. Lei, *et al.*, "Learning face representation from scratch," *preprint arXiv:1411.7923*, 2014.

[27] J. R. Beveridge, P. J. Phillips, *et al.*, "The challenge of face recognition from digital point-and-shoot cameras," in *BTAS*, 2013.

[28] B. F. Klare, B. Klein, *et al.*, "Pushing the frontiers of unconstrained face detection and recognition: IARPA Janus Benchmark A," in *CVPR*, 2015, pp. 1931–1939.

[29] A. Krizhevsky, I. Sutskever, *et al.*, "Imagenet classification with deep convolutional neural networks," in *NIPS*, 2012, pp. 1097–1105.

[30] L. Best-Rowden, S. Bisht, *et al.*, "Unconstrained face recognition: Establishing baseline human performance via crowdsourcing," in *IJCB*, 2014.