Optimal Region-Based 3D Face Representations

Michaël De Smet, Luc Van Gool

Vision for Industry Communications and Services
(ESAT-PSI/VISICS), Faculty of Engineering,
Katholieke Universiteit Leuven,
Kasteelpark Arenberg 10, B-3001, Heverlee, Belgium

Abstract

In this report, we explore the domain of region-based 3D representations of the human face. We begin by noting that although they serve as a key ingredient in many state-of-the-art 3D face reconstruction algorithms, very little research has gone into devising strategies for optimally designing them. In fact, the great majority of such models encountered in the literature is based around manual segmentations of the face into subregions. We propose algorithms that are capable of automatically finding the optimal subdivision given a training set and the number of desired regions. In a cross-validation experiment, we show that part-based models designed using the proposed algorithms are easily capable of outperforming manually segmented models w.r.t. reconstruction accuracies.

1 Introduction

Many problems in computer vision deal with objects that can be subdivided into meaningful parts by a human observer. It is widely believed—both in psychology [1] and in computer science—that such a decomposition can enhance our understanding of an object. This may enable us for example to identify partially occluded or locally deformed objects, or to extrapolate from known examples of an object class. Because of its social relevance, one of the most frequently studied object classes in computer vision is the human face.

In this report we demonstrate that—taking faces as a good case in point—the optimal subdivision into parts does not follow the intuitive subdivisions that have been used so far. We derive a method to extract better parts and show their superiority in 3D face reconstruction experiments. This said, even intuitive subdivisions have proven better than using the face as a whole. Hence, there is a strong case for part-based representations of faces, and we improve on that strand of research.

Many authors have indeed already demonstrated the usefulness of intuitive part-based representations in automatic face recognition tasks. In one of the earliest works, Brunelli and Poggio [2] showed that a template matching scheme based on a combination of facial features such as the eyes, nose, and mouth provides better facial recognition rates than a similar technique based on the face as a whole. More recently, variations on this approach incorporating eigenfeatures have proven to be particularly useful when dealing with partial occlusions and facial expressions [3–5].

An important aspect of part-based representations is that they enable more accurate reconstructions of novel examples of the object class. Blanz and Vetter [6] augmented their 3D Morphable Model (3DMM) of the human face by manually partitioning the face into four regions. By independently adjusting the shape and texture parameters for these regions, and blending the results into a single face model, they were able to obtain more accurate 3D face reconstructions than with a holistic approach. The same principle has later been adopted by various authors [7–11] to enhance the performance of 3DMMs in 3D face modeling, 3D face reconstruction, and automatic face recognition tasks. The main difference between these contributions regarding the part-based representation lies in the way the parts are joined at the boundaries to form a complete face model.

The previously mentioned works have abundantly shown the merits of part-based representations, however they do not provide any automatic tools for obtaining the subdivision into parts, instead relying on manual segmentation of the regions. While this approach may be acceptable for familiar objects where the underlying regions are intuitively clear, other object classes may benefit from automatic partitioning techniques. Furthermore, we will demonstrate that even for familiar object classes like the human face, a manual segmentation is not necessarily optimal.

In the literature, a large amount of research has recently gone into the development of automatic 3D mesh segmentation techniques [12]. Like the pioneering work of Hoffman and Richards [13], the vast majority of these techniques are based on geometric properties such as curvature or geodesic distances. However, we believe that in general a method for automatically subdividing an object class into meaningful parts should not be based on

geometric properties, but rather on deformation statistics. This is especially true when the available data is not geometrical in nature, which for example is the case for color images.

For automatic blendshape segmentation in facial animation, Joshi *et al.* [14] proposed to apply a thresholding operation to a maximum deformation map, followed by some post-processing to clean up the resulting regions. A promising candidate for automatic object decomposition is the Nonnegative Matrix Factorization (NMF) framework, due to Lee and Seung [15]. NMF and its relatives [16, 17] have been applied to databases of facial images, resulting in a set of nonnegative basis images capable of reconstructing the original images with minimal error. By applying sparseness constraints, the basis images can be made to correspond more or less with facial features, but it is not entirely clear how to extract distinct facial regions.

While researching transform invariant models for pedestrian detection, Stauffer and Grimson [18] introduced the concept of Similarity Templates (ST) as a statistical model of pixel co-occurrences within images of the same object class. By applying hierarchical clustering to an aggregate Similarity Template of a database of aligned pedestrian images, they were able to automatically construct a region segmentation that corresponds well to meaningful parts of pedestrian images. Our approach is similar as it uses statistical information about the relationships between vertices in an aligned database of 3D face models, and applies clustering techniques to obtain a decomposition into regions of high correlation. Additionally, we present a technique to automatically determine optimal blending weights for recombining facial parts into a complete 3D face model.

2 Preliminaries

Before introducing our methods for automatically subdividing an object, it is worth mentioning the problem we set out to solve. Although the data used in our experiments was derived from a set of laser scanned faces, the formulation and algorithms are general enough to be applied to any object class that can be modeled as a linear combination of eigenfeatures.

2.1 Facial data

The data used throughout this report is based on the USF DARPA HumanID 3D Face Database of laser scanned faces in a neutral pose. A subset of 187 laser scans has been selected from the original database so that each person's face is present only once in the dataset. The laser scans have been

brought into dense correspondence using a regularized non-rigid registration procedure derived from [19]. The resulting dataset consists of 187 3D face shapes, each composed of 60 436 vertices. Encouraged by the results of [20–23], we decided to exploit the mirror symmetry properties of the human face space by extending the dataset with a mirror image of each face.

2.2 Linear subspaces for reconstruction

Given a set of data vectors, the optimal set of basis vectors for linearly reconstructing the original set in the least squares sense is given by Principal Component Analysis (PCA). This inherently entails some restrictions.

1. The reconstructions are designed as linear combinations of basis vectors. Better results may be possible when this linearity restriction is removed, as for example in kernel PCA.

2. The reconstruction error is minimized in a least squares sense. For some applications this may not be the best error measure.

3. Minimal squared reconstruction error is only guaranteed when reconstructing vectors from the training set. When a dataset is split into a training set and a test set, the basis vectors obtained by applying PCA to the training set may not optimally reconstruct the vectors in the test set.

The third issue is what we are trying to address here. It occurs when the training set is not large enough for PCA to reliably estimate all the modes of variation in the population, which is normally always the case when dealing with non-synthetic data. One way of improving the reconstruction quality for vectors outside the training set, is by incorporating prior knowledge about certain regularities in the population. For example, the mirror symmetry of human faces can be exploited by adding mirrored examples of the original faces to a training set. When the number of training vectors is less than the size of the vectors, this trivially boosts the reconstruction quality by increasing the number of degrees of freedom in the PCA model. Much more importantly, as shown in [21] for grayscale images of faces, this also improves the quality of the computed basis vectors, resulting in increased signal-to-noise ratios for reconstructions even when using a fixed number of basis vectors. This effect was shown to persist even with large training sets of 5627 images of 64×60 pixels. Another popular approach is to subdivide the original training vectors into separate regions, preferably corresponding to localized features in the object class, and train a PCA model on each of

those regions. This is known as the *eigenfeatures* approach. By doing so, the number of available basis vectors is multiplied by the number of subregions, resulting in greater representational power, while retaining the ability to perfectly reconstruct the original training vectors. In principle, one could keep subdividing the training vectors into more and more regions until the desired reconstruction accuracy is achieved. However, in most applications it is desirable to keep the number of basis vectors as low as possible. Therefore, our objective is—given a limited number of regions—to automatically find those regions that minimize the reconstruction error outside the training set. It is expected that these regions will correspond to meaningful parts of the object class.

3 Automatic segmentation of facial regions

Suppose we have a training database of M objects (i.c. faces), each sampled at N corresponding vertex locations. When subdividing this set of 3D surfaces into meaningful regions, we wish to cluster vertices together according to some measure of similarity. In this section, we will design a similarity measure suitable for this context.

The dataset of 3D surfaces that we wish to segment consists of M surfaces, each composed of N vertices $\mathbf{x}_{ij}$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$. Denote by $\boldsymbol{\mu}_i = \frac{1}{M} \sum_{j=1}^{M} \mathbf{x}_{ij}$ the mean position of the i-th vertex, averaged over all surfaces. Then $d_{ij} = ||\mathbf{x}_{ij} - \boldsymbol{\mu}_i||_2$ is the euclidean distance from the i-th vertex of the j-th surface to its mean position. The reason why we prefer to work with distance values rather than displacement vectors is because we want vertices to be clustered together even if they move in different directions w.r.t. their mean positions. For example, consider two vertices located on opposite sides of the nose. If the training set contains faces with noses of varying sizes, then these vertices will move further apart or closer together, causing them to have almost opposite displacement vectors. Since such a scaling operation could easily be represented by a single basis vector, it is more efficient to assign both vertices to the same region. Based on similar reasoning, we choose not to work with the distance values directly, but rather with the normalized distance values

$$y_{ij} = \frac{d_{ij}}{\sqrt{\sum_{l=1}^{M} d_{il}^2}} \,, \tag{1}$$

where the normalization is performed w.r.t. the entire range of displacements the vertex undergoes throughout the training set. The normalized vectors



Figure 1: Correlations between a selected vertex and all other vertices of the face, based on the similarity features described in Section 3. The selected vertex is indicated with a red cross.

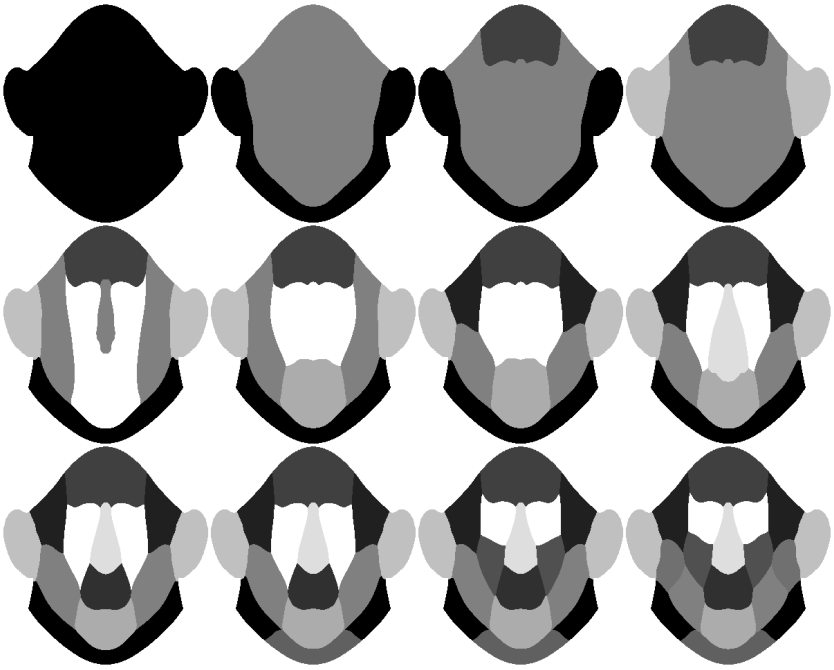

Figure 2: Facial components found by applying the k-means++ clustering algorithm to the similarity features described in Section 3. The features were computed on the 3D shape of a dataset of 187 registered laser scans of the human face.

$\mathbf{y}_i = [y_{i1}, \ldots, y_{iM}]^{\mathrm{T}}, i \in \{1, \ldots, N\}$ can now be used as feature vectors for determining similarities between vertices across the training set (Figure 1). By applying a suitable clustering algorithm to the feature vectors, a segmentation into regions of maximum similarity can be obtained. In our experiments, we used the *k-means++* algorithm [24] with 1000 random restarts. The resulting facial components are shown in Figure 2.

4 Optimal region blending

While a subdivision of an object class into disjoint parts can be useful in itself, it is not enough for optimal part-based reconstructions of a particular object. Consider a part-based 3D shape model of an object. If one of the model parts is allowed to change shape while the rest of the model remains constant, discontinuities are likely to occur at the boundaries between the morphing part and the rest of the model. This is counterproductive in at least two ways:

1. If not properly taken care of, such discontinuities will show up as visible artifacts in the reconstructed object shape. Traditionally [6, 8, 10, 11], this is resolved by applying some kind of blending at the boundaries in a post-processing step.

2. When a part-based model is used in an automatic fitting algorithm, discontinuity errors at the boundaries will be taken into account in the objective function. This may steer the optimization away from the optimal solution, towards a solution that provides a better fit near the boundaries. This issue is not solved by post-processing.

To address both of these issues at the same time, the basis vectors of the part-based model need to be continuous across the entire object. An easy way to achieve this is by training the model on smoothly overlapping training examples, rather than examples that contain all the information of a single region, and are abruptly cut off beyond the region boundaries. The question that remains is how to design the regions of overlap.

4.1 Algorithm derivation

In this section, we present an algorithm for automatically determining the optimal regions of overlap for a linear part-based model. Recall from Section 3 that we have a training database of M objects, each sampled at N corresponding vertex locations. Ideally, the vertices should be organized such that vertices with the same index retain the same physical meaning across all objects. *E.g.* for faces, the vertex located at the tip of the nose should have the same index in all face models of the dataset, and similarly for all other points of the face. To keep the derivation as general as possible, we will assume that each vertex can be represented by a D-dimensional vector. Furthermore, without loss of generality we assume that each sampled object is vectorized by vertically concatenating its vertices, and mean-normalized

by subtracting the corresponding mean from each vetex. *I.e.*, the j-th object in the training set is represented by the DN-dimensional vector

$$\mathbf{x}_j = [\mathbf{x}_{1j}^{\mathrm{T}}, \ldots, \mathbf{x}_{Nj}^{\mathrm{T}}]^{\mathrm{T}} - [\boldsymbol{\mu}_1^{\mathrm{T}}, \ldots, \boldsymbol{\mu}_N^{\mathrm{T}}]^{\mathrm{T}} \ . \tag{2}$$

The entire training set can then be written in matrix form as

$$\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_M] \ . \tag{3}$$

One of the underlying assumptions of the traditional PCA approach is that a particular instance of an object class can be approximated by a linear combination of training examples. Formally, if $\mathbf{y}$ is a (mean-normalized) vector representing a particular instance of the object class, then $\mathbf{y}$ can be approximated as

$$\mathbf{y} \approx \sum_{i=1}^{M} c_i \mathbf{x}_i \ . \tag{4}$$

This principle can be extended to part-based models by introducing a N-dimensional per-vertex weighting vector $\mathbf{w}_j$ for each region $j \in \{1, \ldots, K\}$. After extending the weighting vectors to the full dimensionality of the training vectors by replicating each element D times (which we shall write as $\boldsymbol{\omega}_j$), we obtain

$$\mathbf{y} \approx \sum_{j=1}^{K} \mathrm{diag}(\boldsymbol{\omega}_j)\mathbf{X}\mathbf{c}_j \ , \tag{5}$$

where $\mathbf{c}_j$ is the vector of linear coefficients corresponding to the j-th part of the object. The weighting vectors $\mathbf{w}_j$ contain per-vertex weights specifying the influence that each of the K regions has on the final position (or value) of each vertex. Note that given the weighting vectors, the coefficient vectors that minimize the reconstruction error in the least squares sense are found as

$$[\mathbf{c}_1^{\mathrm{T}}, \ldots, \mathbf{c}_K^{\mathrm{T}}]^{\mathrm{T}} = [\mathrm{diag}(\boldsymbol{\omega}_1)\mathbf{X}, \ldots, \mathrm{diag}(\boldsymbol{\omega}_K)\mathbf{X}]^{+}\mathbf{y}, \tag{6}$$

where the superscript $(^+)$ denotes the Moore-Penrose pseudoinverse. In the interest of brevity, we introduce the notations

$$\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_K] \ , \tag{7a}$$

$$\mathbf{X}_{\mathbf{W}} = [\mathrm{diag}(\boldsymbol{\omega}_1)\mathbf{X}, \ldots, \mathrm{diag}(\boldsymbol{\omega}_K)\mathbf{X}] \tag{7b}$$

for the matrix containing the region weights and the matrix formed by horizontally concatenating the weighted training vectors.

The objective is now to find the weights that allow us to minimize the expected reconstruction error, given the training set. Formally,

$$\mathbf{W}_{\mathrm{opt}} = \arg\min_{\mathbf{W}} \mathrm{E}_{\mathbf{y}}\left[||\mathbf{y} - \mathbf{X}_{\mathbf{W}}(\mathbf{X}_{\mathbf{W}})^{+}\mathbf{y}||_2^2\right] \ , \tag{8}$$

where, in theory, the expectation should be taken w.r.t. the entire population of possible test vectors. Obviously, we don't have access to the full population of possible test vectors. If we did, the optimal basis vectors would be given by a straightforward PCA and the problem would be solved. Here, we only have the training set available and we will have to base the expectation on what's available in there. The solution we propose is to split the training set in two disjoint parts. One part will be used for building the region-based subspaces, while the other part will serve as a source for generating out-of-training-set examples. In order to make maximum use of the available data, and to reduce the danger of overfitting, we propose to randomly reassign vectors to both sets after each iteration of the algorithm. The optimal weights can be iteratively estimated with the following alternating least squares algorithm.

Step 1 Given a $DN \times M_X$ matrix of training vectors $\mathbf{X}$, a $DN \times M_Y$ matrix of test examples $\mathbf{Y}$, and a $N \times K$ matrix of region weights $\mathbf{W}$, we need to find the coefficient matrix $\mathbf{C}$ of dimensions $KM_X \times M_Y$ that optimizes

$$\mathbf{C}^* = \arg\min_{\mathbf{C}} \|\mathbf{Y} - \mathbf{X}_\mathbf{W}\mathbf{C}\|_\mathrm{F}^2 \ , \tag{9}$$

where we have taken advantage of the notations in Eqs. (7a) and (7b). The subscript F indicates the Frobenius norm. Similar to Eq. (6), the solution is given by

$$\mathbf{C}^* = (\mathbf{X}_\mathbf{W})^+\mathbf{Y} \ . \tag{10}$$

Step 2 In the next step, we search for the weight matrix $\mathbf{W}^*$ that optimizes

$$\mathbf{W}^* = \arg\min_{\mathbf{W}} \|\mathbf{Y} - \mathbf{X}_\mathbf{W}\mathbf{C}^*\|_\mathrm{F}^2 \ , \tag{11}$$

given $\mathbf{X}$, $\mathbf{Y}$, and $\mathbf{C}^*$. First, note that the difference can be rewritten as

$$\begin{aligned} \mathbf{Y} - \mathbf{X}_\mathbf{W}\mathbf{C}^* = \ & [\mathbf{y}_1, \ldots, \mathbf{y}_{M_Y}] \\ & - \sum_{i=1}^{K} \mathrm{diag}(\boldsymbol{\omega}_i)[\mathbf{v}_{i1}, \ldots, \mathbf{v}_{iM_Y}] \ , \end{aligned} \tag{12}$$

by forming the example vectors $\mathbf{v}_{ij}$, $i \in \{1, \ldots, K\}$, $j \in \{1, \ldots, M_Y\}$ as linear combinations of the training vectors in $\mathbf{X}$, based on the coefficients in $\mathbf{C}^*$. By examining eq. (12), it becomes clear that the least squares solution to eq. (11) can be found by solving N independent linear systems (one system of DM_Y equations and K unknowns per vertex). Therefore, at least $M_Y \geq K/D$ test

vectors are needed to find a solution.[1]

The following additional notes complete the algorithm:

1. The weights computed in Step 2 of the iteration may include negative values, which is not physically meaningful. As is standard practice in the NMF framework [25], a valid weight matrix can be found by setting the negative values to zero after each iteration.

2. By additionally constraining the weights to sum to one for each vertex of the model, we ensure that the resulting set of basis vectors retains the ability to perfectly reconstruct the original training vectors.

3. For high dimensional data—such as high-resolution 3D scans—the algorithm converges rather slowly. In our implementation, this is resolved by using a coarse-to-fine approach with four pyramid levels.

4.2 Final algorithm

To conclude, the final algorithm as used to generate the results in this report can be stated as follows:

Input:

- Mean-normalized training set $\mathbf{X}$.
- Number of desired regions K.

Initialization:

- Compute pyramid representation of $\mathbf{X}$.
- Compute initial hard segmentation by applying the method described in Section 3.
- Set initial weights $\mathbf{W}$ according to the hard segmentation.

Iteration:

- For each level of the pyramid:
  - Iterate until convergence or desired number of iterations reached:
    - Randomly split training set into $\mathbf{X}$ and $\mathbf{Y}$.
    - Compute coefficients $\mathbf{C}$ (Step 1).

[1] In some applications, particularly where multimodal data is involved, it might be beneficial to use different weights for each dimension of the vertices. In that case, the requirement becomes $M_Y \geq K$.

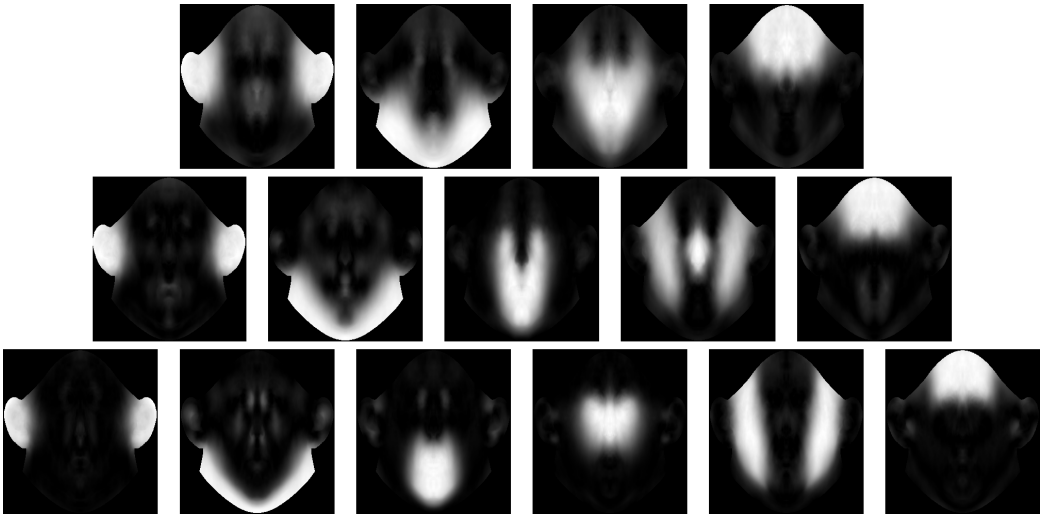

Figure 3: Optimal weights computed with the algorithm described in Section 4 on a dataset of 187 laser scanned faces, for four (top), five (center), and six (bottom) regions.

    - ∗ Compute weights **W** (Step 2).
    - ∗ Set negative entries in **W** to zero.
    - ∗ Force **W** to sum to one for each vertex.
  - Upsample **W** to next pyramid level.

Output: W

5 Evaluation

Since the objective of the proposed algorithms is to minimize the reconstruction error for objects that are not present in the training set, the best way to evaluate their performance is by building region-based models using the output **W**, and experimentally testing their reconstruction performance on a test set. An important aspect of this is to compare the results against the performance of similar models based on manually defined regions.

First, we need to specify how to build a part-based model starting from a set of regions and a training set. The naive way to do this would be to compute a global PCA model from the training data, and then to subdivide the resulting eigenvectors according to the regions. The disadvantages of this approach are threefold.

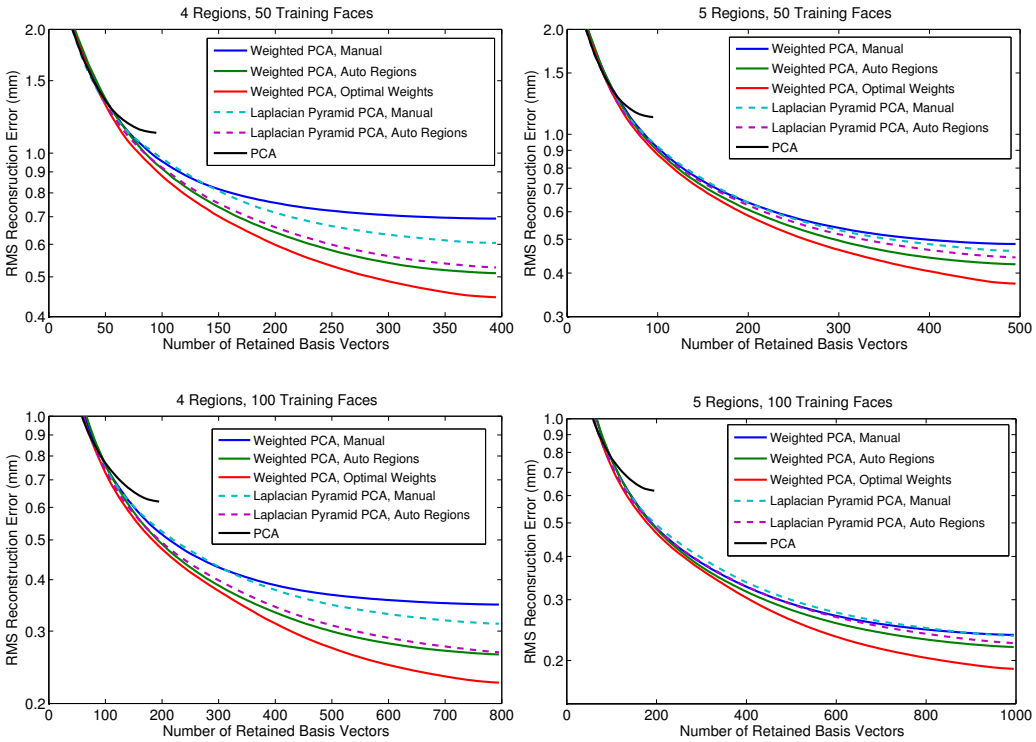

Figure 4: Average reconstruction errors for the 10-fold cross-validation experiments described in Section 5.1. Note that the approach using the optimal weights computed with the proposed algorithm (red solid line) clearly outperforms the approach using manually segmented regions (blue solid line).

1. While the global PCA basis vectors are designed to be optimal for reconstructing the training vectors, there is absolutely no reason why this would still be the case after subdividing them.

2. Subdividing the basis vectors does not retain orthogonality, which complicates the reconstruction task and introduces redundancy in the model.

3. While the basis vectors returned by PCA are sorted according to their ability to explain the training data, this is no longer the case after subdivision. Hence, truncating the model by retaining the first few basis vectors may arbitrarily worsen the quality of reconstructions.

A better technique would be to apply the subdivision directly to the training vectors, and then to train separate PCA models for all of the regions. In this way, each of the region-specific PCA models is guaranteed to be optimal w.r.t. the reconstruction errors on the training set within its own region. However, problems arise when the models are combined for reconstructing an entire object. Indeed, while the ordering of the basis vectors is optimal within each region, this need not be the case *across* regions. As a compromise, one might try to re-sort the basis vectors of the combined model according to their relevance for reconstructing the global training vectors. Furthermore, if the regions have any form of overlap, which is necessary for avoiding discontinuities (see Section 4), the basis vectors belonging to neighbouring regions are not orthogonal w.r.t. eachother.

An easy way to address all of the issues raised in the previous paragraph is the following. Apply the parts subdivision directly to the training data, resulting in K region-specific training sets, and train a single PCA model on the union of these sets. The resulting basis vectors will be strictly orthogonal, optimal for reconstructing the unified training set, and sorted according to their relevance on the training data. One flaw still remains. The model has no knowledge about statistical interdependencies between regions, because it has only seen data containing isolated regions. To address this, we propose to include the original global training vectors in the combined training set. This is the approach taken in our following experiments.

One approach to the region blending problem that deserves special attention is the method used by Blanz and Vetter [6] for their 3DMM. Instead of simply blending the surface patches at the boundaries according to some weighting factor, they employ a blending technique based on Laplacian pyramids. By simultaneously blending the patches at multiple pyramid levels, a wavelength-dependent overlap size is obtained, which provides discontinuity-free blending, while retaining high-frequency detail.

5.1 Experiments

In our experiments, we compare the reconstruction accuraccy on a test set for various part-based models of the 3D shape of human faces. Our experimental setup is as follows. As mentioned in Section 2.1, the dataset consists of 187 laser scanned faces and their mirror images. We test two scenarios: one where 50 faces (*i.e.*, 100 scans including the mirror images) are available for training, and one with 100 training faces (200 scans). Ten cross-validation tests are performed. In each test the dataset is randomly partitioned in a training set and a test set (taking care to always assign mirrored scans to the same set as the original ones). First, a global PCA model is trained, and its reconstruction accuracies are evaluated on the test set for a wide range of model truncations. (By model truncation, we mean the operation of retaining the first few basis vectors of a model, while discarding the rest.) The process is repeated for region-based models, where models with two, up to a maximum of seven parts are considered. As a baseline, we manually define segmentations of four components (eyes, nose, mouth, rest) and five components (eyes, nose, mouth, ears, rest), and compare models based on these segmentations against the automatically generated ones. To avoid discontinuity artifacts at the boundaries for those region-based PCA models that are purely based on a hard segmentation (either manually defined or automatically clustered as in Section 3), a smooth overlap was created by convolving the hard partition masks with a Gaussian filter having a standard deviation of approximately 10 mm. In the following, we will use the term *Weighted PCA* for all region-based PCA models that use a single matrix $\mathbf{W}$ to define the regions. This in contrast to *Laplacian Pyramid PCA* models, which have different regions of overlap—or equivalently, different weights—for different levels of detail.

The results for four and five components are presented in Figure 4. Perhaps surprisingly, weighted PCA with manually defined regions was the worst performer of the bunch in all test cases. As expected, the best results were invariably obtained with the weighted PCA approach based on the optimal weights computed with the proposed algorithm from Section 4.2. The second-best results were obtained with a weighted PCA model based purely on the automatically clustered regions from Section 3. When considering only four regions, the Laplacian pyramid PCA models exhibit a somewhat unexpected behavior. While this technique provides a marked improvement over weighted PCA for manually defined regions, it actually deteriorates the results when the regions are based on the automatic segmentation. This might indicate that the automatically determined boundaries are located in regions where a frequency-dependent overlap is less relevant. Also notewor-

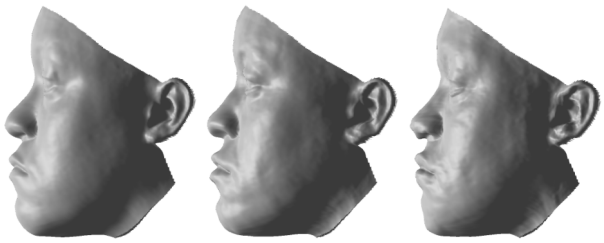

Figure 5: Example of a face reconstruction based on 50 training faces. Left: Original face. Center: Reconstructed with a region-based model using optimal region blending (Section 4). Right: Reconstruction based on manual regions.

thy is the fact that all region-based PCA models in our tests succeeded in outperforming global PCA from about halfway its number of available basis vectors. Without constraining the number of basis vectors, it is easily feasible to cut the reconstruction error of global PCA in half, or even in three.

Figure 5 shows a typical example of the quality improvements that can be expected when upgrading from a manually segmented model (right-hand side) to a model with optimal region blending (center). Note the overall improved signal-to-noise ratio, and the improved reconstruction quality of facial features like the nose and the chin.

The take-home message of these experiments is that part-based models can seriously boost the performance of linear eigenspace-based reconstruction methods, and that there are better alternatives than segmenting the relevant parts by hand. Of course, the improvements generally don't come for free. The increased number of basis vectors can be quite taxing on computing resources, especially when dealing with high-resolution models. On the other hand, it is certainly worth mentioning that the proposed technique *does* provide a significant improvement w.r.t. others that is essentially for free, since the number of basis vectors remains the same. Even better, when selecting a number of basis vectors to achieve a certain reconstruction quality, the optimally weighted regions can often get away with less than half the number of basis vectors needed by the manual parts.

Given the generality of the derived method, it would be interesting to test it on different modalities, like facial textures, or thermal infrared data. It seems likely that the optimal regions would differ from those obtained for 3D shapes.

6 Conclusion

We began our investigation into the world of part-based 3D face representations by noticing that although their advantages are widely accepted, the mechanics behind them are only superficially understood. Many state-of-the-art approaches rely on them as a key component for enabling lifelike 3D reconstructions, yet practically none of them seem to have given much thought as to how to optimally design them. Most of the methods are based on manual segmentations and often, the concept of blending at the boundaries seems to have been added more as an afterthought. By delving deeper into the subject, we were able to come up with two complimentary methods for automatically finding the underlying parts in vectors representing objects of the same class. The first method harnesses suitable features for finding strictly disjoint regions of maximum similarity, while the second method relaxes the constraints in favor of finding optimal per-vertex weights that minimize the expected reconstruction error on objects outside the training set. The resulting part-based models have been shown to outperform similar representations based on manual segmentations.

In future work, it would be interesting to see the developed machinery deployed on different datasets representing other object classes, or even the same object class (*i.e.* faces) seen through different modalities.

References

[1] I. Biederman, "Recognition-by-components: A theory of human image understanding," *Psychological Review*, vol. 94, no. 2, pp. 115–147, 1987.

[2] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," *PAMI*, vol. 15, pp. 1042–1052, Oct 1993.

[3] A. Pentland, B. Moghaddam, and T. Starner, "View-based and modular eigenspaces for face recognition," in *CVPR*, pp. 84–91, Jun 1994.

[4] A. M. Martínez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *PAMI*, vol. 24, no. 6, pp. 748–763, 2002.

[5] F. Tarrés, A. Rama, and L. Torres, "A novel method for face recognition under partial occlusion or facial expression variations," in *ELMAR*, pp. 163–166, 2005.

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

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

[8] C. Basso and A. Verri, "Fitting 3D morphable models using implicit representations," in *GRAPP*, pp. 45–52, 2007.

[9] I. A. Kakadiaris, G. Passalis, G. Toderici, M. N. Murtuza, Y. Lu, N. Karampatziakis, and T. Theoharis, "Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach," *PAMI*, vol. 29, no. 4, pp. 640–649, 2007.

[10] Y. Zhang and S. Xu, "Data-driven feature-based 3D face synthesis," *3DIM*, vol. 0, pp. 39–46, 2007.

[11] F. B. ter Haar and R. C. Veltkamp, "3D face model fitting for recognition," in *ECCV (4)*, pp. 652–664, 2008.

[12] A. Shamir, "A survey on mesh segmentation techniques," *Comput. Graph. Forum*, vol. 27, no. 6, pp. 1539–1556, 2008.

[13] D. D. Hoffman and W. A. Richards, "Parts of recognition," *Cognition*, vol. 18, pp. 65–96, 1984.

[14] P. Joshi, W. C. Tien, M. Desbrun, and F. H. Pighin, "Learning controls for blend shape based realistic facial animation," in *SIGGRAPH*, 2003.

[15] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, 1999.

[16] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *JMLR*, vol. 5, pp. 1457–1469, 2004.

[17] R. Zass and A. Shashua, "Nonnegative sparse PCA," in *NIPS*, pp. 1561–1568, 2006.

[18] C. Stauffer and W. L. Grimson, "Similarity templates for detection and recognition," in *CVPR*, pp. 221–230, 2001.

[19] C. Basso, P. Paysan, and T. Vetter, "Registration of expressions data using a 3D morphable model," in *FG*, pp. 205–210, 2006.

[20] M. Kirby and L. Sirovich, "Application of the karhunen-loeve procedure for the characterization of human faces," *PAMI*, vol. 12, no. 1, pp. 103–108, 1990.

[21] P. S. Penev and L. Sirovich, "The global dimensionality of face space," in *FG*, pp. 264–270, 2000.

[22] Q. Yang and X. Ding, "Symmetrical PCA in face recognition," in *ICIP (2)*, pp. 97–100, 2002.

[23] M. I. Shah and D. C. Sorensen, "A symmetry preserving singular value decomposition," *JMAA*, vol. 28, no. 3, pp. 749–769, 2006.

[24] D. Arthur and S. Vassilvitskii, "k-means++: The advantages of careful seeding," in *SODA*, pp. 1027–1035, 2007.

[25] M. W. Berry, M. Browne, A. N. Langville, V. P. Pauca, and R. J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics & Data Analysis*, vol. 52, no. 1, pp. 155–173, 2007.