Optimal Regions for Linear Model-Based 3D Face Reconstruction

Michaël De Smet and Luc Van Gool

K.U. Leuven ESAT-PSI/VISICS, Heverlee, Belgium
michael.desmet@esat.kuleuven.be

Abstract. In this paper, we explore 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 on 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. The generality of the segmentation approach is demonstrated on examples from the TOSCA database, and a cross-validation experiment on facial data shows that part-based models designed using the proposed algorithms are capable of outperforming alternative segmentations w.r.t. reconstruction accuracy.

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 paper 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.

Many authors have 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, 4, 5]. Similar results have also been found for 3D face recognition [6].

R. Kimmel, R. Klette, and A. Sugimoto (Eds.): ACCV 2010, Part III, LNCS 6494, pp. 276–289, 2011.

An important aspect of part-based representations is that they enable more accurate reconstructions of novel examples of the object class. Blanz and Vetter [7] 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. Similarly, Peyras et al. [8] used region-specific Active Appearance Models (AAMs) to enable accurate facial feature fitting on unseen faces. The same principle has been adopted by various authors [9,10,11,12,13,14] 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, but 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 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 gone into the development of automatic 3D mesh segmentation techniques [15]. The vast majority of these are based on geometric properties such as curvature or geodesic distances. While these methods tend to work well for articulated objects like full-body scans, they are less reliable for faces, where the parts are ill-defined from a geometrical standpoint. Indeed, we believe that in general a method for automatically subdividing an object class into meaningful parts should not be based solely on geometric properties, and could benefit greatly from deformation statistics. This is especially true when the available data is not geometrical in nature, which for example is the case for color images. In this paper, we demonstrate that good segmentations can be obtained using only deformation statistics.

For automatic blendshape segmentation in facial animation, Joshi et al. [16] 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 [17]. NMF and its relatives [18, 19] 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 [20] introduced the concept of Similarity Templates 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 facial data used in this paper 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 [21]. The resulting dataset consists of 187 3D face shapes, each composed of 60 436 vertices. Encouraged by the results of [22,23,24], 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 linear combinations of basis vectors. Better results may be possible when this linearity restriction is removed.
2. The reconstruction error is minimized in a least squares sense. Depending on the application, 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 in this paper. 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

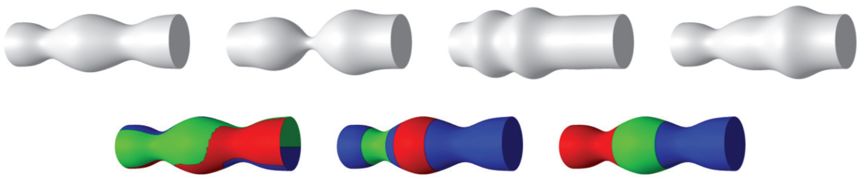

Fig. 1. Example of applying the segmentation technique to synthetic data. The data was generated by applying three independent overlapping gaussian deformations to a cylinder. *Top row:* four of the 20 available training examples. *Bottom row:* three-component segmentations based on (*left*) statistically normalized displacement vectors, (*center*) displacement magnitudes, (*right*) statistically normalized displacement magnitudes. Only the last type of feature vectors leads to the correct part subdivision.

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 [23] 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 [3]. 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, each sampled at N corresponding vertex locations. When subdividing this set of 3D surfaces into 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.

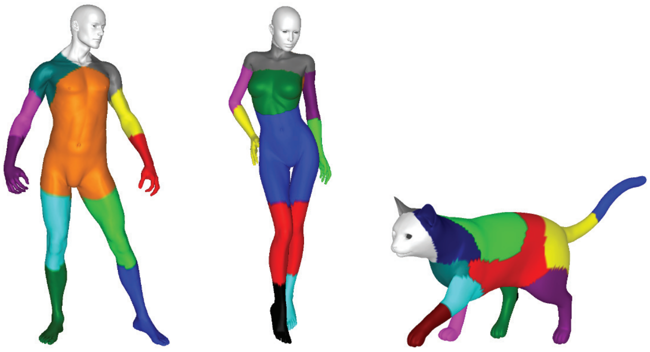

Fig. 2. Segmentations of models from the TOSCA high-resolution 3D database using weighted k-means clustering of the proposed feature vectors. Weighted k-means was used to compensate for large variations in local mesh density.

The dataset of 3D surfaces that we wish to segment consists of M surfaces, each composed of N vertices $\mathbf{s}_{ij}$, $i \in \{1, \ldots, N\}$, $j \in \{1, \ldots, M\}$. Denote by $\boldsymbol{\mu}_i = \frac{1}{M} \sum_{j=1}^{M} \mathbf{s}_{ij}$ the mean position of the i-th vertex, averaged over all surfaces. Then $d_{ij} = ||\mathbf{s}_{ij} - \boldsymbol{\mu}_i||_2$ 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 different displacement vectors. Since such a scaling operation could be represented by a single basis vector in the eigenfeatures approach, it is more efficient to assign both vertices to the same region. Based on similar reasoning, we choose not to work with the distances directly, but rather with the normalized distance values

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

where the normalization is performed w.r.t. the entire range of displacements the vertex undergoes throughout the training set. The normalized vectors $\mathbf{t}_i = [t_{i1}, \ldots, t_{iM}]^{\mathrm{T}}, i \in \{1, \ldots, N\}$ can now be used as feature vectors for determining similarities between vertices across the training set. By applying a suitable clustering algorithm to the feature vectors, a segmentation into regions of maximum deformation similarity can be obtained. In our experiments, we used the *k-means++* algorithm [25] with 1000 random restarts. The effect of using different types of feature vectors is illustrated in Fig. 1 for an artificially generated dataset. Results on models from the TOSCA high-resolution 3D database [26] are shown in Fig. 2. For the face dataset described in Section 2.1, we obtained the facial components shown in Fig. 3.

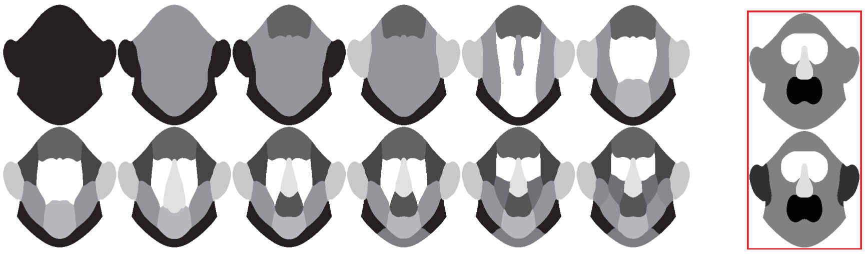

Fig. 3. Facial components found by applying the k-means++ clustering algorithm to the similarity features described in Section 3. The features were computed on the 3D shape of a dataset of 187 registered laser scans of the human face. The red box (*right*) shows manual segmentations of the face into four and five regions as used in our experiments (Section 5.1). The manual subdivisions correspond well with those commonly found in the literature.

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 [7, 11, 13, 14], this is resolved by 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 issues, 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., 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 assume that each vertex can be represented by a D-dimensional vector. Furthermore, without loss of generality we assume that each object is vectorized by vertically concatenating its vertices, and mean-normalized by subtracting the corresponding mean from each vertex. I.e., the j-th object in the training set is represented by the DN-dimensional vector

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

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

$$\mathbf{S} = [\mathbf{s}_1, \ldots, \mathbf{s}_M] \tag{3}$$

One of the 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

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

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

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

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

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

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

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

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

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,

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

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 all 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 is used for building the region-based subspaces, while the other part serves as a source for generating out-of-training-set examples. To make maximum use of the available data, and to reduce the danger of overfitting, we propose to randomly reassign vectors to both sets in 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 used 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

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

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]

[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 $M_Y \geq K$.

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 [27], 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 paper (Fig. 4) can be stated as follows:

```
Input:
    – Mean-normalized training set $\mathbf{S}$.
    – Number of desired regions $K$.
Initialization:
    – Compute pyramid representation of $\mathbf{S}$.
    – 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).
            ∗ Compute weights $\mathbf{W}$ (Step 2).
            ∗ Set negative entries in $\mathbf{W}$ to zero.
            ∗ Force $\mathbf{W}$ to sum to one for each vertex.
        • Upsample $\mathbf{W}$ to the next pyramid level.
Output: $\mathbf{W}$
```

5 Evaluation

Since the objective of the proposed algorithms is to minimize the reconstruction error for faces 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 the reconstruction accuracy on a test set. An important aspect is to compare the results of using different subdivisions.

First, we need to specify how to build a part-based model starting from a set of regions and a training set. The models used in our experiments were constructed according to the following approach:

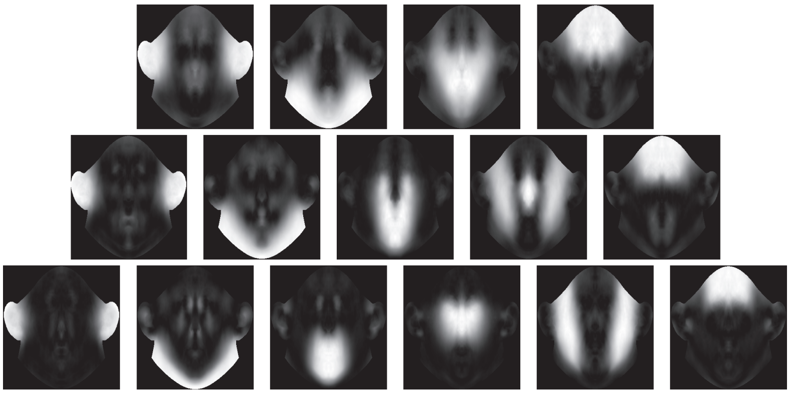

Fig. 4. Optimal weights computed with the algorithm described in Section 4 on a dataset of 187 laser scanned faces, for four (*top*), five (*center*), and six (*bottom*) regions

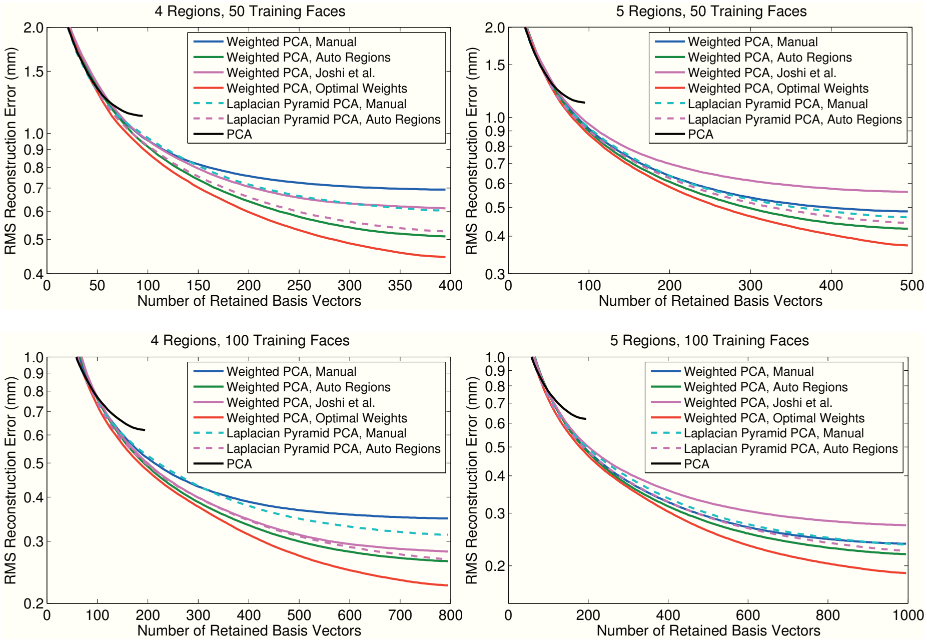

Fig. 5. Average reconstruction errors for the 10-fold cross-validation experiments described in Section 5.1. The standard deviation of the curves is less than 5 μm, and roughly corresponds to the line thickness. Note that the approach using the optimal weights computed with the proposed algorithm (*red solid line*) clearly outperforms the approach using manually segmented regions (*blue solid line*).

1. Create a global training set by subtracting the mean face from the set of training examples.
2. Create region-specific training sets by applying the per-vertex region weights to the global training set (Eq. (7b)).
3. Gather the global and region-specific training sets into a single set and compute basis vectors by applying PCA.

This procedure combines the desirable properties of PCA-based models with the advantages of part-based models. Specifically, the models have the following properties:

- The basis vectors are orthogonal, which avoids redundancy in the model and facilitates the reconstruction task.
- The basis vectors are sorted according to their ability to explain the training data. This ensures that truncating the model by removing some of the least significant basis vectors does not greatly impair the reconstruction quality.
- The model was trained on the set of complete faces as well as their parts, and therefore has knowledge of both global and local deformation statistics. This is in contrast to models trained on only the parts, which would lack any knowledge of statistical relations between the parts.

To avoid discontinuity artifacts at the boundaries for those region-based PCA models that are based on a hard segmentation (either manually or automatically determined), a smooth overlap between regions was created by convolving the hard partition masks with a Gaussian filter having a standard deviation of approximately 10 mm. An alternative approach to the region blending problem that deserves special attention is the method used by Blanz and Vetter [7] 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 preserving high-frequency detail. To discriminate between the two blending schemes, we will use the term *Weighted PCA* for all region-based PCA models that use a single matrix $\mathbf{W}$ to define the regions, and *Laplacian Pyramid PCA* for models using different regions of overlap—or equivalently, different weight matrices—for different spatial freqencies.

5.1 Experiments

In our experiments, we compare the reconstruction accuracy 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 (100 scans when 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 into a training set and a test set (taking care to always assign mirrored scans to the same set as the original ones).

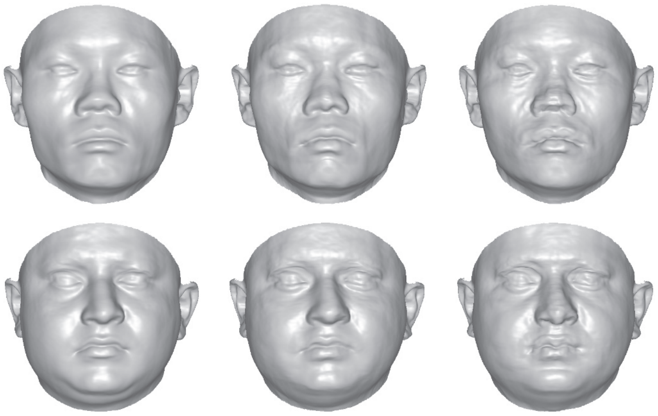

Fig. 6. Examples of face reconstructions based on 50 training faces using four facial regions. *Left:* Original face. *Center:* Reconstructed with a region-based model using optimal region blending (Section 4). *Right:* Reconstruction based on manual regions.

First, a global PCA model is trained, and its reconstruction accuracy is 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. As a baseline, we manually define segmentations into four components (eyes, nose, mouth, rest) and five components (eyes, nose, mouth, ears, rest) (Fig. 3), and compare models based on these segmentations against the automatically generated ones.

The results are presented in Fig. 5. As expected, the best results were obtained with a weighted PCA approach based on 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. Given four regions, part-based models based on the method by Joshi et al. [16] still managed to outperform the manual segmentations, while for five regions this method came out last. Our results also show that the performance of Laplacian pyramid PCA models can be improved by using our automatic segmentations. All region-based PCA models in our tests succeeded in outperforming the baseline 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.

Typical examples of the quality improvements that can be expected when upgrading from a manually segmented model to a model with optimal region blending are shown in Fig. 6. Note the overall improved signal-to-noise ratio, and the improved reconstruction quality of facial features (most visible in the nose, mouth, and chin regions).

The main conclusion 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. When compared to other part-based models, the proposed technique provides a significant improvement that is essentially for free, since for a given number of basis vectors the quality improves. Even better: for a given reconstruction accuracy, models using optimally blended regions often get away with less than half the number of basis vectors needed by others.

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 have noticed that although the advantages of part-based 3D face representations are widely accepted, the mechanics behind them are only superficially understood. Many state-of-the-art approaches rely on them for enabling lifelike 3D reconstructions, yet none of them seem to have pursued optimality in the design of the regions. Most of the methods are based on manual segmentations, and blending at the boundaries is usually done as a post-processing step. In this paper, we have presented two complementary methods for automatically finding the underlying parts in vectors representing objects of the same class. The first method uses suitable features for finding disjoint regions of maximum deformation 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. In our experiments, the resulting part-based models have been shown to outperform models based on other segmentations.

In future work, it would be interesting to see how these techniques perform on datasets representing other object classes, or the same object class (i.e. faces) seen through different modalities. Also, we intend to check whether the improved reconstruction quality translates to better face recognition results in reconstruction-based approaches.

Acknowledgements. This study was supported by IWT-Flanders through the AMASS++ project (IWT/SBO 060051), and through the project "Forensic Biometric Authentication" (IWT/SBO 60851).

References

1. Biederman, I.: Recognition-by-components: A theory of human image understanding. Psychological Review 94, 115–147 (1987)
2. Brunelli, R., Poggio, T.: Face recognition: Features versus templates. IEEE T. Pattern Anal. 15, 1042–1052 (1993)
3. Pentland, A., Moghaddam, B., Starner, T.: View-based and modular eigenspaces for face recognition. In: Proc. CVPR, pp. 84–91 (1994)
4. Martínez, A.M.: Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class. IEEE T. Pattern Anal. 24, 748–763 (2002)

5. Tarrés, F., Rama, A., Torres, L.: A novel method for face recognition under partial occlusion or facial expression variations. In: Proc. ELMAR, pp. 163–166 (2005)
6. Faltemier, T.C., Bowyer, K.W., Flynn, P.J.: A region ensemble for 3-D face recognition. IEEE T. Inf. Foren. Sec. 3, 62–73 (2008)
7. Blanz, V., Vetter, T.: A morphable model for the synthesis of 3D faces. In: Proc. SIGGRAPH, pp. 187–194 (1999)
8. Peyras, J., Bartoli, A., Mercier, H., Dalle, P.: Segmented AAMs improve person-independent face fitting. In: Proc. BMVC (2007)
9. Blanz, V., Vetter, T.: Face recognition based on fitting a 3D morphable model. IEEE T. Pattern Anal. 25, 1063–1074 (2003)
10. Mian, A., Bennamoun, M., Owens, R.: Region-based matching for robust 3D face recognition. In: Proc. BMVC, vol. 1, pp. 199–208 (2005)
11. Basso, C., Verri, A.: Fitting 3D morphable models using implicit representations. In: Proc. GRAPP, pp. 45–52 (2007)
12. Kakadiaris, I.A., Passalis, G., Toderici, G., Murtuza, M.N., Lu, Y., Karampatziakis, N., Theoharis, T.: Three-dimensional face recognition in the presence of facial expressions: An annotated deformable model approach. IEEE T. Pattern Anal. 29, 640–649 (2007)
13. Zhang, Y., Xu, S.: Data-driven feature-based 3D face synthesis. In: Proc. 3DIM, pp. 39–46 (2007)
14. ter Haar, F.B., Veltkamp, R.C.: 3D face model fitting for recognition. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part IV. LNCS, vol. 5305, pp. 652–664. Springer, Heidelberg (2008)
15. Shamir, A.: A survey on mesh segmentation techniques. Comput. Graph. Forum 27, 1539–1556 (2008)
16. Joshi, P., Tien, W.C., Desbrun, M., Pighin, F.H.: Learning controls for blend shape based realistic facial animation. In: Proc. SIGGRAPH (2003)
17. Lee, D.D., Seung, H.S.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
18. Hoyer, P.O.: Non-negative matrix factorization with sparseness constraints. J. Mach. Learn. Res. 5, 1457–1469 (2004)
19. Zass, R., Shashua, A.: Nonnegative sparse PCA. In: NIPS, pp. 1561–1568 (2006)
20. Stauffer, C., Grimson, W.L.: Similarity templates for detection and recognition. In: Proc. CVPR, pp. 221–230 (2001)
21. Basso, C., Paysan, P., Vetter, T.: Registration of expressions data using a 3D morphable model. In: Proc. FG, pp. 205–210 (2006)
22. Kirby, M., Sirovich, L.: Application of the karhunen-loeve procedure for the characterization of human faces. IEEE T. Pattern Anal. 12, 103–108 (1990)
23. Penev, P.S., Sirovich, L.: The global dimensionality of face space. In: Proc. FG, pp. 264–270 (2000)
24. Yang, Q., Ding, X.: Symmetrical PCA in face recognition. In: Proc. ICIP, vol. 2, pp. 97–100 (2002)
25. Arthur, D., Vassilvitskii, S.: k-means++: The advantages of careful seeding. In: Proc. SODA, pp. 1027–1035 (2007)
26. Bronstein, A., Bronstein, M., Kimmel, R.: Numerical Geometry of Non-Rigid Shapes. Springer Publishing Company, Incorporated, Heidelberg (2008)
27. Berry, M.W., Browne, M., Langville, A.N., Pauca, V.P., Plemmons, R.J.: Algorithms and applications for approximate nonnegative matrix factorization. Comput. Stat. Data An. 52, 155–173 (2007)