Emergent Graphs with PCA-features for Improved Face Recognition

Alexander Heinrichs, Marco K. Müller, Andreas H. J. Tewes and Rolf P. Würtz

Institut für Neuroinformatik, Ruhr-Universität, D–44780 Bochum, Germany

Abstract. Built on the principles of "Learning from Nature" and "Self-organization" Elastic Bunch Graph Matching for face recognition is a defining example for Organic Computing methodology. Here, we follow these principles further to advance the method in two respects. First, the requirement for manual annotation of landmarks is reduced to one single face, from which a self-organizing selection process gradually builds up the bunches by adding the most similar face to the bunch graph and then recalculating the matching. Second, the resulting bunches are replaced by the principal components of the nodes of all persons in the database. The similarity function is restricted to a suitable subset of these components. The additional self-organizing processes lead to improved precision of landmark localization and recognition rates. Altogether, an improved data structure for face storage has emerged from the simple presentation of examples in a minimally supervised way.

Keywords: Face Recognition, Landmark Finding, PCA
PACS: 42.30.Tz, 07.05.Pj, 42.30.Sy

INTRODUCTION

Automatic face recognition has become an important part of biometric identification systems. It is also a computer vision problem, which has been treated successfully with methods of Organic Computing. The principles of "learning from nature" and "self-organization" have been used to develop the methods of *Elastic Graph Matching* (EGM) [2] and *Elastic Bunch Graph Matching* (BGM) [8]. The Organic Computing methodology behind them is described in [9], where it is shown how the hierarchical organization of elementary feature detectors to more and more complex structures can be made useful for a computer system.

In this paper we take two further steps. First, we study the self-organized creation of a bunch graph starting from one manually labeled face. In earlier versions the arduous task of hand-labeling was required for about 50 faces in order to construct a useful bunch graph. The remaining manual interaction is required in order to provide the system with a basic definition of the concept of a face.

In a second organization step the bunches of collected jets are replaced by a subset of their principal components, which can also be seen as a result of a self-organizing neural network [1] and be computed very efficiently and incrementally [6].

The new method is tested on computer-graphics generated images of faces and a large data set of real face images. In both cases it provides both better recognition results and better performance in terms of computation time. For the artificial faces the positioning precision of landmarks was highly improved.

CP860, *Information Optics: 5th International Workshop,*
edited by G. Cristóbal, B. Javidi, and S. Vallmitjana

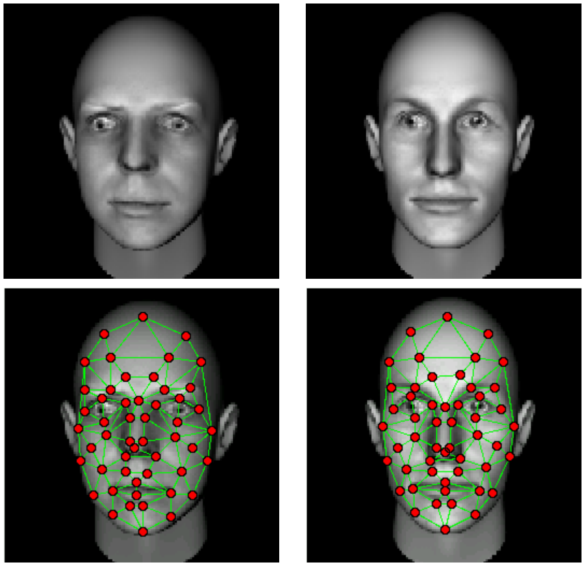

FIGURE 1. Examples of FaceGen generated face images with matched graphs.

ELASTIC GRAPH MATCHING

Elastic graph matching was developed for face recognition in [2] and extended to bunch graph matching in [8]. A detailed description of this approach can be found there. In 1996 bunch graph matching showed very strong performance in the FERET test [5], and in 2004 basically the same method scored second in the face authentication test at the 17th International Conference on Pattern Recognition [4].

Faces in this method are represented by elastic graphs, so called model graphs. Nodes are placed on characteristic points on the face, which are called landmarks. Examples of model graphs are shown in figure 1. At these landmarks local information about the texture is stored by calculating the scalar products with a set of Gabor wavelet filters, the resulting vector is called a *jet*. For the construction of a bunch graph the model graphs of many faces are combined by linking the jets of the single graphs nodewise. It is crucial for the success of the recognition that the nodes of the individual graphs are placed as precisely as possible on the same landmarks in the face, so that the information at the feature sets presents the same area of the faces. The concept of the bunch graph is illustrated in figure 2.

To find the position of a face in an unknown image, the features of the bunch graph are compared with features of points in the image in several steps, so called *moves*. In the first move, called Global Move, the graph is placed on all possible positions in the image without changing the shape of the graph, i.e. the relative positions of the nodes. At every node, the different jets are compared via a similarity function with the jet at the position in the image. The most similar jet is chosen and its similarity value is chosen as similarity for this node. The similarity of the graph at this position is than calculated as

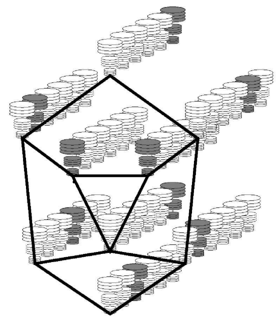

FIGURE 2. Bunch graph concept: At every graph node a bunch of jets stored. In the matching process the best fitting jet is selected.

the average of the similarities of the nodes. The graph state which delivers the maximum graph similarity is used as initialization of the subsequent moves.

In the following moves, the shape of the graph is changed in different ways. In the *scale move*, the size of the graph is changed either only in one dimension or in both dimension simultaneously. In this way the method can deal with small variations in scale or the shape of the face. In a last step, for every node in the graph the most similar position in a small area around the node is searched and the node is placed on it. Since the nodes are moving almost independently from each other, this move is called *local move*.

Two similarity functions are in use for computing the similarity between two jets. The first one only takes into account the magnitudes of the complex-valued Gabor filter responses. These magnitudes are arranged into a vector and a normalized inner product is computed. The second similarity function is used in the final steps of landmark finding for very precise positioning and uses the phase information. The inner product of the first function is modified by the difference of the phases here.

UNSUPERVISED COLLECTION OF MODEL GRAPHS

The organization process that selects the data for the bunch graph is rather simple. Starting with one hand labeled model graph, the graphs of 516 faces of different individuals are assembled in an unsupervised manner into a bunch graph. This method can only work on a set of images, which shows faces of similar size, pose and orientation. We chose a subset of the well known FERET database [3], which matched this criteria. The first step for learning the model graphs is to match the hand labeled model graph with all images. The image with the highest similarity is selected, and the matching graph with jets from that image is added to the bunch graph, which now consists of two model graphs.

Then the just added image is removed from the image set and the bunch graph is again matched with the remaining image set. Again, the best matching graph is added to the

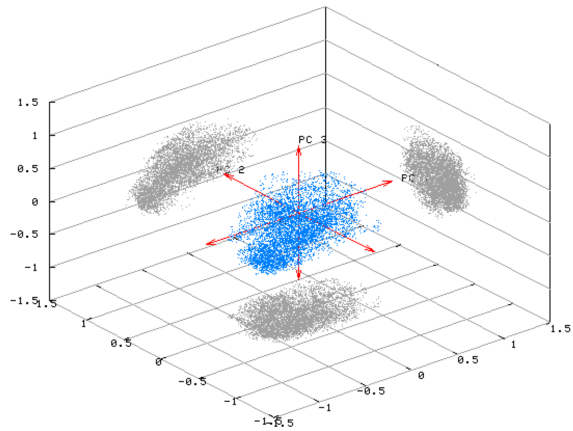

FIGURE 3. Feature space spanned by the first three Principal Components for an arbitrary node

bunch graph and the relating image is removed from the data set. In this way, the bunch graph grows and finally contains the information of all images in the data set. The value of the maximal similarity, i.e. the similarity to the graph, which is put to the bunch graph in the next step, is controlled, to prevent a very erroneous match to become part of the bunch graph.

When the value of the maximal similarity begins to decrease for several learning steps, the learning can be aborted, because this shows that the remaining images are difficult to match, and the possibility for errors grows. Depending on how the graphs will be used (bunch graph-building or as a basis for a statistical model), the requirements can be adjusted.

CONCEPT OF LOCAL FEATURE MODELS

The idea of statistical feature models is motivated by the concept of feature bunches as realized in the bunch graph concept. Instead of saving all features at a given node and using them for comparison with features drawn from the image immediately, we use them to create a statistical model, which describes the distribution of these features with respect to a mean feature. This leads to one model for every node of the graph. Unfortunately it is not possible to estimate the feature distribution directly. This is due to the fact that the features of interest are distributed in an 80-dimensional space. The well-known curse of dimensionality prohibits implementation of a statistical model which could satisfy higher moments in the data distribution. Therefore, a dimension reduction has to be performed in advance. We decided to use Principal Component Analysis (PCA), which has proved to be an accurate tool for these kinds of problems. This makes it possible to decompose a given $n-$dimensional data cloud into n one-dimensional signals. This assumes that the distributions along the axes are statistically independent or can at least be treated as such in a first order approximation. Since PCA decorrelates the samples, it at least guarantees that the second statistical moment as well as the first one (samples have zero-mean) vanish. The n one-dimensional distributions are described using a Gaussian. This, as we could conclude from examples, seems to be a reasonable assumption. Nevertheless, there can exist axes according to which the distribution can become different a Gaussian. In order to separate the former from the latter only a subset of principal axes were used for estimating the one-dimensional distributions. A reasonable selection was part of the experiments.

The Gaussian distribution is given by the mean value and the variance, both of which are already determined by the previously performed principal component analysis. Figure 3 shows the distribution at a given node reduced to the first three principal components.

After the statistical model has been determined for each node, which includes estimating the mean value and variances for each principal axis, it can immediately be used for matching using the following similarity function.

$$S(\mathrm{node}_k) = \sum_{i=n(k)}^{N(k)} \exp\left(-\frac{\left\langle \vec{j}-\vec{m} \,\middle|\, \vec{P}_i \right\rangle^2}{2\sigma_i}\right).$$

Here $\vec{m}$, σ_i and $\vec{P}_i$ represent the mean, the i-th variance and i-th principal component, respectively, while $\vec{j}$ denotes a jet drawn from the image at the current node position. $\langle\cdot|\cdot\rangle$ is the inner product, and the sum runs over all axes for which the assumption of a Gaussian distribution is sufficiently well fulfilled. The node-dependent lower bound can also be used to ignore the principal components with highest variance (see section for details). The total similarity is finally given by

$$S_{\mathrm{total}} = \sum_{k=1}^{M} S(\mathrm{node}_k),$$

where M is the number of nodes within the graph. The matching process itself remains otherwise unchanged in comparison to bunch graph matching. After all moves have been performed, the location which leads to the highest total similarity determines the best match. The original jets at these positions are then compared with the original jets of all persons. The computational cost of this final step is still proportional to the total number of persons, but it is much faster than the landmark finding, which in PCA matching requires time proportional to the number of principal components used.

DATA FOR EXPERIMENTS

Artificial FaceGen images

To evaluate the accuracy of node positioning on landmarks, one needs to know the exact landmark positions in a face. This is very hard for natural images, so in order to get precise localization errors, we have first made experiments with artificial images, where we can evaluate both accuracy of node positioning and recognition rate.

FaceGen, a program by Singular Inversions Inc. provides the possibility to generate images of a face, controlled by several parameters. Because these are rendered from a three-dimensional model, precise landmark positions are known. Together with the image, a corresponding graph file with 52 exactly defined landmark positions is created. The data set we have used for the positioning experiments, as well for the recognition experiments, consists of 2000 randomly created FaceGen images, 1000 persons in two expressions (neutral and smiling) with little pose variation (up to 5 degrees from frontal view horizontally and vertically). For training 500 different persons have been created in the same way. The resolution of the images is 192x192. All of the faces in the images have the same size and are located at the center of the image. These 500 images are the basis of the bunch graph, from which the statistical model in the PCA graph is built.

FERET database

In order to provide exact landmark positions in real images one would have to label all images manually. in the case of real images we have measured the recognition rates from feature comparison on found landmarks. We expect, that a more precise landmark positioning system leads to a better comparison of these landmarks and so to higher recognition rates. This hypothesis can be tested by the experiments on artificial images.

For the recognition experiments on natural images two large galleries of the same persons are needed. All faces should be located at the center of the image, they should have about the same size. The pose variation should be minimal. For this purpose, we have taken pictures of the FERET database [5]. The fa- and fb-pictures provide images of frontal view with little pose variation. To get images in which all faces have approximately the same size, we have done the following preprocessing.

First the resolution has been reduced to 256×256 pixels. A Facefinder [7] has been applied, that uses bunch graph matching with three bunch graphs of different sizes. The area of the best fit has been scaled to an image of size 128×128.

As this preprocessing does not work perfectly, some so created images only show parts of a face or no face at all. These image pairs have been removed manually. If the same person was represented by more than one image pair these image pairs have also been removed. Please note that these manual interventions are not part of the algorithm, they have only been applied to get clean data for quantitative experiments.

This preselection has provided two galleries of 957 persons. The bunch graph and the PCA graph have been build from 516 different images as it was described in Section .

RESULTS

Positioning results on FaceGen images

At first, we compare the positioning of PCA matching against bunch graph matching. For this, we compute the average pixel error for each of the 2000 images for both

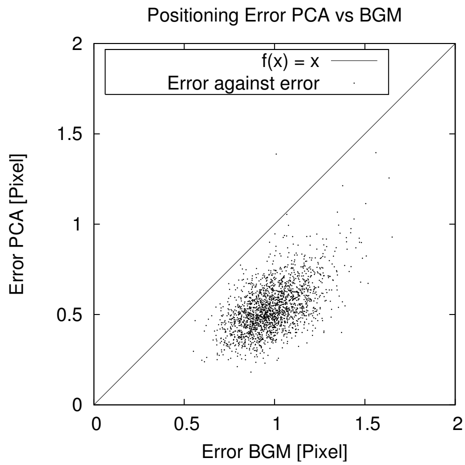

FIGURE 4. Average positioning error PCA matching vs. bunch graph matching for 2000 test images.

TABLE 1. Positioning errors for 2000 FaceGen Images

graph matching Method	Positioning Error [Pixel]
No matching	2.30
BGM (500 graphs)	0.99
PCA (500 graphs)	0.54
Ground Truth	0.00

matching methods. The result is shown in figure 4. The average errors for all the images are shown in table 1. There also the average error for the case, that the average graph is just placed in the center of the images, is presented. Using PCA matching instead of bunch graph matching, the positioning error could be reduced from 0.99 pixel to 0.54 pixel.

Another great advantage of PCA matching is that matching speed is proportional to the number of used PCs, which are 80, if all PCs are used. The matching time for bunch graph matching is proportional to the number of model graphs inside the bunch graph, which is 500 in our case.

TABLE 2. Recognition error for 1000 FaceGen persons

graph matching Method	Recognition Error
No matching	14.0%
BGM (500 graphs)	10.3%
PCA (500 graphs)	9.7%
Ground Truth	9.5%

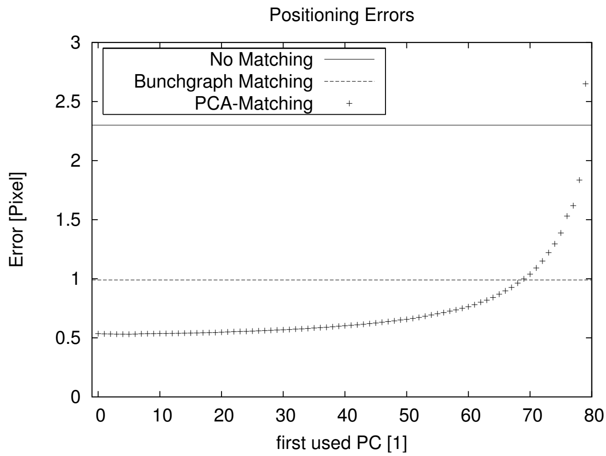

FIGURE 5. Positioning errors of PCA matching over first principal component used for 2000 test images.

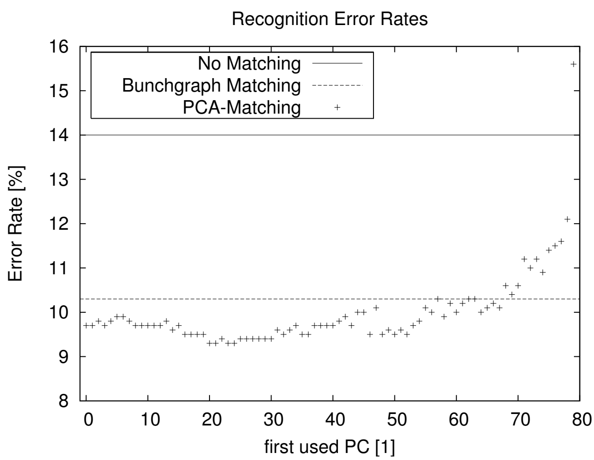

FIGURE 6. Recognition error for 1000 FaceGen Persons.

It has turned out that for PCA matching not even all PCs need to be used. Normally, the PCs with the least variance are not used, but for landmark finding it is useful to ignore the first PCs (the ones with the highest variance). This makes sense, because for searching a landmark, the feature dimensions which are similar for all persons are more important than those that vary a lot from one person to the other, but for the features that are the same with many persons. Figure 5 shows that positioning works well if the first PCs are disregarded.

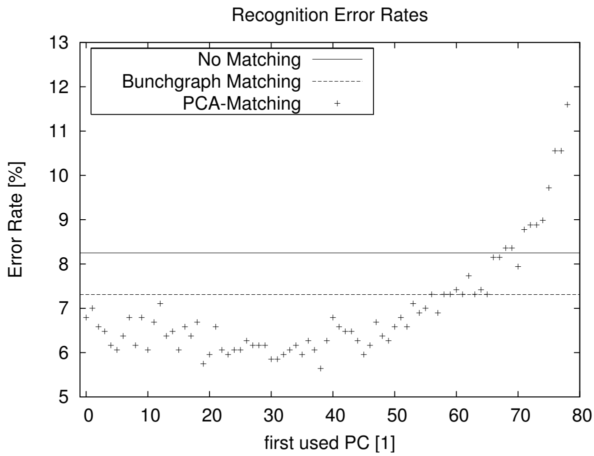

FIGURE 7. Recognition error for 957 persons of the FERET database.

TABLE 3. Recognition errors for 957 preprocessed real persons of the FERET-Database

graph matching Method	Recognition Error
No matching	8.25%
BGM (516 graphs)	7.31%
PCA (516 graphs, PCs 38-79)	5.64%

Recognition results on FaceGen images

One important question is of course, if the improvement in landmark positioning achieved with PCA matching also results in a higher recognition rate. Table 2 shows the result.

The recognition rates show that fewer positioning errors lead to fewer recognition errors. Especially, if one compares the recognition errors with the recognition error at ground truth positions, the improvement from bunch graph matching to PCA matching is really significant. This can be improved by leaving out the first PCs, as figure 6 shows.

With the found coherence between positioning error and recognition error, it is possible to evaluate the positioning error indirectly by measuring the recognition error. Figure 7 shows the recognition errors over the first used PC as in the figures for FaceGen images.

The characteristics are about the same as for artificial images. The results are also very similar, which is shown in table 3. Performing no matching at all (relying on accurate centering of the faces) leads to a high recognition error (8.25%), bunch graph matching works better (7.31%) and PCA matching yields the best result (5.64%).

CONCLUSION

By application of Organic Computing principles we have conceived a novel method of encoding local facial features using a statistical PCA model fitted to sample data. Using this model on artificial face images, we could show an improvement both in positioning accuracy and recognition rate. This confirms the expectation that higher accuracy in landmark finding improves recognition. A higher recognition rate could also be achieved on natural images of faces. Furthermore the new approach is more efficient for large data sets, because the computation time for landmark finding is independent of the number of samples, only the much faster recognition step still requires time proportional to the number of persons.

Acknowledgments

We gratefully acknowledge partial funding from Deutsche Forschungsgemeinschaft (grants WU 314/2-2 and MA 697/5-1). Portions of the research in this paper use the FERET database of facial images collected under the FERET program, sponsored by the DOD Counterdrug Technology Development Program Office.

REFERENCES

1. Hertz, J., Krogh, A., and Palmer, R. G. (1991). *Introduction to the theory of neural computation*. Addison-Wesley, Redwood City.
2. Lades, M., Vorbrüggen, J. C., Buhmann, J., Lange, J., von der Malsburg, C., Würtz, R. P., and Konen, W. (1993). Distortion invariant object recognition in the dynamic link architecture. *IEEE Transactions on Computers*, **42**(3), 300–311.
3. Loui, A. C., Judice, C. N., and Liu, S. (1998). An image database for benchmarking of automatic face detection and recognition algorithms. In *Proceedings of the IEEE International Conference on Image Processing*, Chicago, Illinois, USA.
4. Messer, K., Kittler, J., Sadeghi, M., Hamouz, M., Kostin, A., Cardinaux, F., Marcel, S., Bengio, S., Sanderson, C., Poh, N., Rodriguez, Y., Czyz, J., Vandendorpe, L., McCool, C., Lowther, S., Sridharan, S., Chandran, V., Palacios, R. P., Vidal, E., Bai, L., Shen, L., Wang, Y., Yueh-Hsuan, C., Hsien-Chang, L., Yi-Ping, H., Heinrichs, A., Müller, M., Tewes, A., von der Malsburg, C., Würtz, R., Wang, Z., Xue, F., Ma, Y., Yang, Q., Fang, C., Ding, X., Lucey, S., Goss, R., and Schneiderman, H. (2004). Face authentication test on the BANCA database. In *Proceedings of ICPR 2004, Cambridge*, volume 4, pages 523–532.
5. Philips, P. J., Moon, H., Rizvi, S. A., and Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **22**(10), 1090–1104.
6. Weng, J., Zhang, Y., and Hwang, W.-S. (2003). Candid covariance-free incremental principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **25**(8), 1034–1040.
7. Wieghardt, J. and Loos, H. S. (2001). Finding faces in cluttered still images with few examples. In G. Dorffner, H. Bischof, and K. Hornik, editors, *Artificial Neural Networks – ICANN 2001*, volume 2130 of *Lecture Notes in Computer Science*, pages 1026–1033, Vienna, Austria. Springer Verlag.
8. Wiskott, L., Fellous, J.-M., Krüger, N., and von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **19**(7), 775–779.
9. Würtz, R. P. (2005). Organic Computing methods for face recognition. *it – Information Technology*, **47**(4), 207–211.