**LAB 1.2 - EIGENFACES**

Núria Codina - 240359, Maria Guasch - 242289 (January 2024)

**Introduction**

The aim of this project is to extract representative facial bases from two types of data: a) facial images (i.e. following the eigenfaces algorithm); b) facial landmarks.

**State of the art**

In the field of facial analysis, Eigenfaces have been widely applied as a powerful method for facial representation and recognition. Turk and Pentland (1991) developed the Eigenfaces algorithm, introducing the concept of representing facial images as linear combinations of eigenfaces. This work set the basis for subsequent research in face recognition by highlighting the efficiency of principal component analysis (PCA) in capturing essential facial features. The Eigenfaces method has, since then, been employed in various applications due to its robustness against variations in lighting, pose, and facial expressions.

Furthermore, in a more recent study by Zhang et al. (2017), the authors applied Eigenfaces in the context of large-scale face recognition datasets. Their work emphasizes the scalability and efficiency of Eigenfaces in handling extensive databases, demonstrating its practical utility in real-world scenarios. This reference underscores the ongoing relevance and applicability of the Eigenfaces method, particularly in the context of contemporary challenges posed by increasing database sizes.

Together, these references highlight the historical significance of Turk and Pentland's Eigenfaces algorithm, its evolution through techniques like Fisherfaces, and its continued efficacy in addressing diverse challenges in facial recognition.

**Dataset**

To accomplish the objectives of the project a dataset has been chosen meticulously.

The Chicago Faces Dataset (CFD) consists of images of 597 unique individuals. It includes Asian, Black, Latino, and White female and male models, recruited in the United States. All models are represented with neutral facial expressions, as well as a subset featuring various expressions. However, our focus was exclusively on the neutral expressions due to the availability of corresponding landmarks for these images.

The decision to opt for this dataset was motivated by several factors, including the accessibility of landmarks, the high quality of images, and the absence of pre-processing requirements. Notably, all images were consistently centered, captured from the same perspective, under identical illumination, and without variations in rotation.

**Eigenfaces algorithm**

We have followed more or less the same steps to obtain the eigenfaces from both images and landmarks.

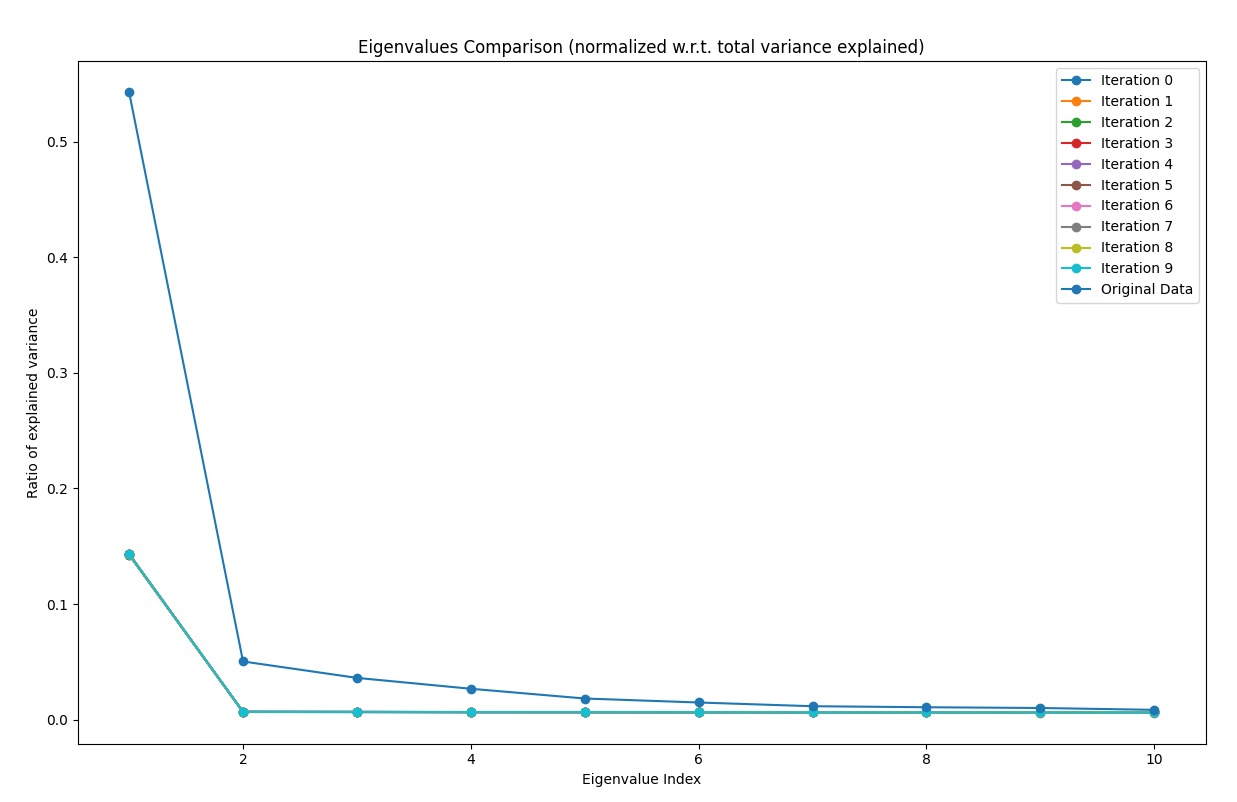
First, we have performed some preprocessing on the images. We have cropped the images to keep just the face itself and reshaped them to mitigate computational load concerns. As for the landmarks, we have applied *procrustes alignment*, which ensures that all landmarks are correctly aligned, to avoid unwanted variation.

The next step was to vectorize the data. In both cases, we have taken each data sample (an image or an array of x, y coordinated for each of the landmarks), and transformed it into an array, to then create a matrix of all the vectorized data. From this matrix we subtracted the “mean face” to ensure proper alignment.

Then, we could compute the covariance matrix, which would be then used to do eigen decomposition. On the one hand, for the dataset of images we have calculated the *pseudo covariance matrix*, which is computed in the following way: .

We chose this one instead of the regular one, to avoid problems with the dimensions. In particular, with the covariance matrix (computed ) we would obtain a covariance matrix of dimensions , which becomes too computationally expensive. By doing this, we obtain a matrix of dimensions (N being the number of images in the dataset), which is much more affordable, in terms of memory and computational efficiency. Later, we had to do an extra step of multiplying the eigenvectors by the original data to obtain the eigenvectors as if we computed them with the regular covariance matrix. On the other hand, we could directly use the covariance matrix because the number of dimensions was much smaller (each face became a vector of 378 positions: 189 pairs of x,y coordinates). Finally, in both cases, we normalized the resulting eigenvectors, and sorted them w.r.t. the magnitude of their corresponding eigenvalues (only keeping the positive ones).  
  
At this point, we could reshape the eigenvectors to visualize the “eigenfaces”. Later in this report we provide an explanation of them.

**EIGENFACES APPLIED TO FACE IMAGES**

**Meaningfulness of extracted bases**

In this scenario, the selection of the ten selected bases is grounded in their statistical significance.The accompanying plot illustrates this significance by comparing our eigenvalues to those obtained from ten random datasets. Notably, the eigenvalues in our dataset surpass the values observed in the random datasets, affirming their statistical significance and reinforcing their relevance in the analysis.

**Bases representation**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

By visualizing the eigenfaces, we can kind of interpret which variation they are capturing. We can see some eigenfaces emphasizing the eyes and the mouth (1, 10), others the eyebrows area (4, 5, 6) and some others the neck (2). But generally, all of them capture some face variance in a way that, when combining all of them, we can get the most significant features of a face.

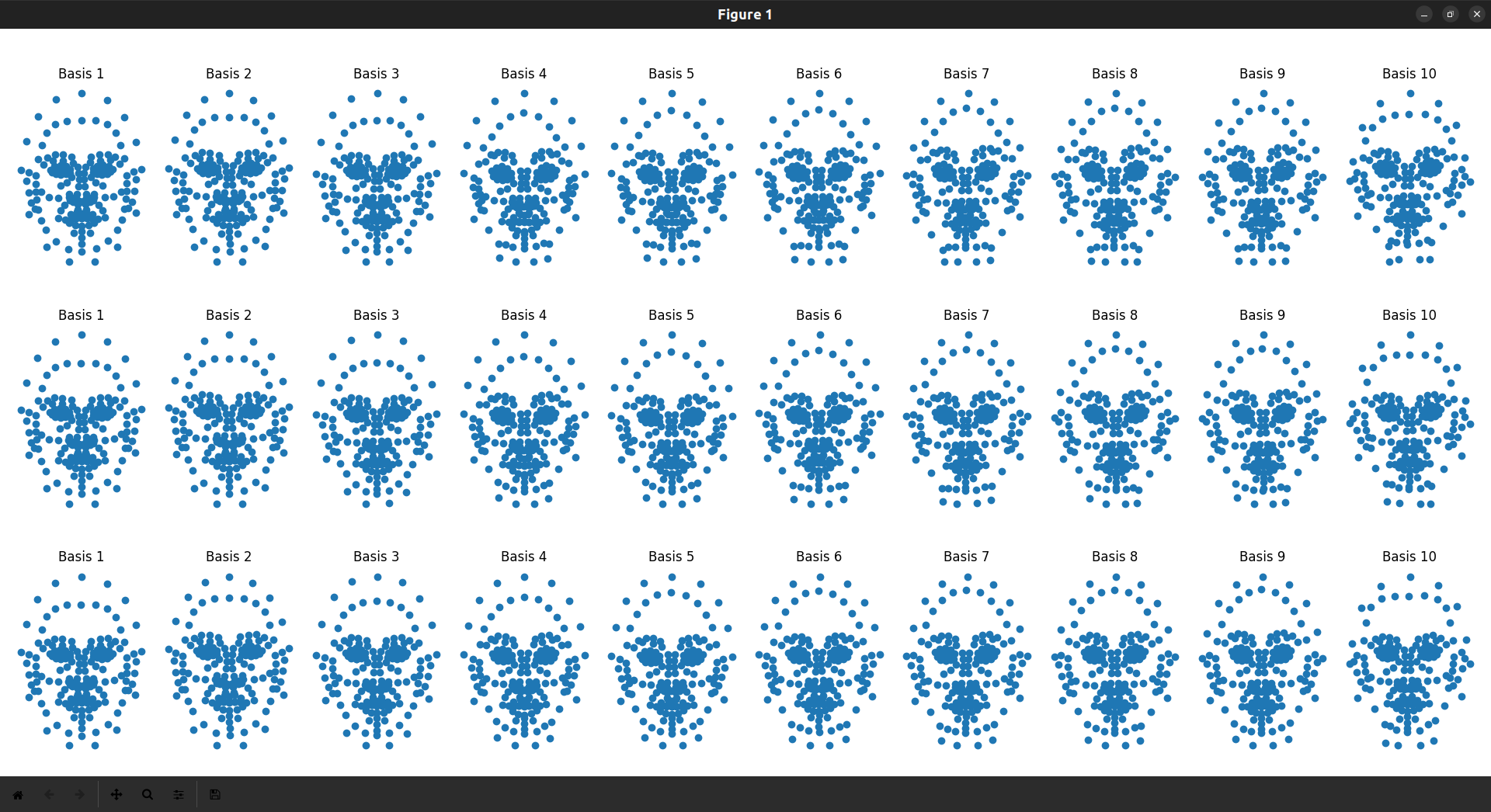
**Image reconstruction**

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |

So, while each original image had ​​2444 × 1718 pixels, by using only 10 eigenface encodings for each image, we were able to rebuild each image with quite good resemblance.

**EIGENFACES APPLIED TO LANDMARKS**

**Meaningfulness of extracted bases**



We have decided to apply *modes of variation* in this case, to see the significance of the different components. In the image above we can see for 3 different image samples, how the landmarks change when adding more components.  
For example, we can see that the 3 images to the left are the same, but when we add more components the faces start to be a bit different. However, we can also see that the last components do not add much new information, so they are not as significant as the first ones.

As the landmarks do not explain as much information as the images, we can see above that the variation through the eigenvectors is not much considerable.

**Bases representation**

We have created a scatter plot of each of the eigenfaces (the obtained eigenvectors reshaped to a list of x,y coordinates). Each of these 10 eigenfaces (in blue) is plotted along with a reference face (in red), to have some context of the points.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
|  |  |  |  |  |

Again, we can take a look at these eigenfaces to see in which area each component has more impact. It is a bit more difficult to interpret with points than with images, but still we can see that the first eigenface is located around the nose area or the second one around the eyes. The last ones do not give much new information, so they correspond to noise.

**Image reconstruction**

Again, we can see a few examples of reconstruction with only 10 bases.

|  |  |  |  |
| --- | --- | --- | --- |
| Original | Reconstruction | Original | Reconstruction |
|  |  |  |  |

These reconstructions are pretty accurate, which follows what we mentioned about the significance of components when working with landmarks. Still, there is some variation in the mouth and chin area.

**Conclusions**

In conclusion, this project successfully applied the Eigenfaces algorithm to both facial images and landmarks, achieving effective reconstruction using a minimal set of eigenface encodings. The demonstrated success in image and landmarks reconstruction underscores the utility and adaptability of the Eigenfaces algorithm in the field of facial analysis.

**References**

*CFD | Chicago Face Database*. (n.d.). <https://www.chicagofaces.org/>

Luo, L., Swamy, M. N. S., & Plotkin, E. I. (2003, May). A modified PCA algorithm for face recognition. In CCECE 2003-Canadian Conference on Electrical and Computer Engineering. Toward a Caring and Humane Technology (Cat. No. 03CH37436) (Vol. 1, pp. 57-60). IEEE.

Muller, N., Magaia, L., & Herbst, B. M. (2004). Singular value decomposition, eigenfaces, and 3D reconstructions. SIAM review, 46(3), 518-545.

Turk, M. A., & Pentland, A. P. (1991, January). Face recognition using eigenfaces. In Proceedings. 1991 IEEE computer society conference on computer vision and pattern recognition (pp. 586-587). IEEE Computer Society.

üge Çarıkçı, M., & Özen, F. (2012). A face recognition system based on eigenfaces method. Procedia Technology, 1, 118-123.

Walia, E. (2008, May). Face recognition using improved fast PCA algorithm. In 2008 Congress on Image and Signal Processing (Vol. 1, pp. 554-558). IEEE.

**Note:**

Since the dataset is too heavy, we provide a link to our GitHub repository with the code and the dataset

<https://github.com/mariaguasch/ACG_2024_Nuria_Maria.git>

Either way, we provide a pickle file with the preprocessed images, so to run the eigenfaces.py code it is not necessary to have the images.