A Python toolbox for Face Alignment and Decomposition (FAD)

Carl M. Gaspara,b

a Zayed University, P.O. Box 144534, Abu Dhabi, United Arab Emirates

b Center for Cognition and Brain Disorders, The Affilated Hospital of Hangzhou Normal University, Hangzhou, China; Institutes of Psychological Sciences, Hangzhou Normal University, Hangzhou, China; Zhejiang Key Laboratory for Research in Assessment of Cognitive Impairments, 311121 Hangzhou, China

+971 50 363 4709

[carl.gaspar@zu.ac.ae](mailto:carl.gaspar@zu.ac.ae)

[carl.michael.gaspar@icloud.com](mailto:carl.michael.gaspar@icloud.com)

Oliver G. B. Garrod

Fab Data

London, UK

[oliver.garrod@fabdata.io](mailto:oliver.garrod@fabdata.io__glasgow.ac.uk)

Images of faces are among the most popular stimuli in psychology research. Two of the most common steps in the preparation of facial stimuli are (1) the spatial alignment of faces and (2) the decomposition of faces into facial features that can be rearranged. We describe FAD, an open-source Python package (<https://github.com/SourCherries/fad>) that automates these two common steps and allows researchers to rapidly create large numbers of stimuli for a wide range of perception and cognition studies (whole-part effects, chimeric faces, Thatcher faces, double-face illusions, crowding stimuli, and many more).

Face alignment in FAD is the first open-source solution aimed at behavioral research that is both well documented and extensively tested on databases of face photos that span ethnicity and facial expression. Alignment in FAD allows for windowing, morphing and feature decomposition/rearrangement, all of which FAD implements in a single easy-to-use framework.

FAD implements a novel wavelet-based decomposition of faces into features (brows, eyes, nose and mouth) that mirrors neural and psychophysical models of early visual processing. Additionally, FAD features taper into the background in a natural and gradual way, allowing for seamless rearrangements and novel configurations that would not be feasible otherwise.

FAD documentation and modularity provides opportunity for the modification and extensibility by the scientific community. As a Python package, FAD provides a bridge between behavioral research and the large body of image processing and deep learning tools in machine learning and visual psychophysics.

open source, Python, faces, stimuli, facial features, image statistics, generalized Procrustes analysis, morphing, aperture

Declarations

**Funding:** National Natural Science Foundation of China (NSFC) to Carl M. Gaspar (grant No.: 31371132).

**Conflicts of interest/Competing interests:** None.

**Ethics approval:** Not applicable

**Consent to participate:** Not applicable

**Consent for publication:** Not applicable

**Availability of data and materials:** The 4 face databases (the data) we used to evaluate our software are only available by request from the researchers who created those databases (cited in this paper). Many of the most popular face databases used in perception and psychology research are only available this way, and we wanted our evaluation to reflect how AFA would perform with those sorts of images. However, anyone can easily drop new batches of faces into the results folder and run the various evaluation scripts to see for themselves how AFA performs.

**Code availability:** Publicly available on GitHub <https://github.com/SourCherries/fad>

Introduction

Everyone working with face stimuli has to align them in some way. The simplest reason is that faces must be seen in the same place on a computer display. Spatial alignment of faces can reduce spatial uncertainty; minimize eye movements; simplify the interpretation of neuroimaging results; ensure that the same areas of the face are being viewed trial after trial; and many other specific reasons. Typically, the exact reasons for alignment are not provided but some combination of the above reasons applies to almost all studies involving human responses to face stimuli. Therefore, face alignment can be considered a requirement by convention.

The alignment of faces by orientation and size can be just as important as alignment by position, especially given how the human visual system is finely tuned to specific bands of both spatial frequency and orientation (Gaspar et al., 2008; Pachai et al., 2013; Smith & Schyns, 2009; Yang et al., 2014). Just as with position, face size and orientation are also standardized in almost all studies. And while the exact reasons for standardization of size and position are often not stated, the reasons for such practice can include many of the same reasons stated for position alignment; in addition to the visual system’s precise preference for specific spatial frequency and orientation. Therefore, the standardization of face size and orientation can also be considered requirements by convention.

The alignment of face position and the standardization of face size and orientation are the most common steps for the preparation of face images for use as stimuli in experiments. Additionally, all of these steps are affine transformations of the image that require some estimation of facial landmarks. A facial landmark refers to the x and y pixel position of a distinctive part of the face like the left corner of the mouth or the center of the right pupil (see Figure 4 for an example of 51 different landmarks). We refer to the entire process of position, size and orientation normalization for a set of face images as simply *face alignment*.



Figure 1. FAD workflow. All operations rely on alignment (Align) as the initial step. Wavelet-based feature decomposition (Features) gives rise to a wide range of stimuli, many of which can be easily generated by FAD (only 2 are listed here to save space). This figure illustrates the basic workflow for most purposes. However, operations can be easily combined using a few lines of lines of code; e.g., feature-only faces (Features) that are then set in an aperture (Window) as in Figure 3.

Face alignment can be critical in a wide range of studies that use face stimuli. Face alignment is important in fMRI studies with a focus on retinotopy (Hsieh et al., 2010), brain areas that are retinotopic (Petro et al., 2013), face orientation (Goffaux et al., 2016) and face size (Iidaka et al., 2004). Face alignment is also important in behavioral studies and neuroimaging studies that require precise specification of image-information across different dimensions of the face. Examples include: measurement of an ideal observer (J. M. Gold et al., 2012; Lai et al., 2014; Peterson & Eckstein, 2012); measurement of observer spatial-frequency (Gaspar et al., 2008) or orientation channels (Pachai et al., 2013); reverse correlation (Sekuler et al., 2004; Vinette et al., 2004); and models of task-specific face information (Dakin & Watt, 2009a; Keil, 2009; Xu et al., 2014).

Despite the importance of face alignment in many studies, face alignment procedures used by many studies are often time-consuming and lack transparency. Typically, face alignment for a set of face photographs is primarily done using either one of 2 approaches: (1) manual landmark selection and affine manipulation using an image-editing application like Adobe Photoshop; or (2) manual landmark selection and affine manipulation using software like InterFace or custom software. The first approach is quite common, while the second approach is typically taken in studies that use morphed faces as morphing is one of the primary functions of InterFace. Regardless of approach, the manual selection of landmarks is required in both of these cases. That can be highly time consuming for large stimulus sets or it can present a disincentive from using larger sets that would allow for greater generalizability or novel types of hypotheses to be tested. Manual landmark estimation may also be subject to individual differences in landmark placement that simply would not exist with an algorithmic process.

Additionally, software used for alignment should be made available as open-source in a public repository with full documentation of its inner-workings. A fully open-source solution would provide greater transparency, reproducibility and extensibility. Ideally anyone in the scientific community should be able to fully understand and replicate the alignment process that is used for a given study. Finally, an open-source package would allow for the scientific community to make use of existing code to extend its functionality to test novel hypotheses.

Here we introduce Automatic Face Alignment (AFA), a Python package that fully automates all steps in the face alignment process and is available as a public repository on GitHub with documentation, demos and unit tests (<https://github.com/SourCherries/auto-face-align> ). We feel that these simple features make it an attractive option for stimulus preparation in a large number of scenarios. In fact, AFA has already been used to make stimuli for a combined psychophysical and electrophysiological study (Hashemi et al., 2020). For a demonstration of AFA, please see Figure 1, which shows the input and output of AFA face alignment as well as the automatic generation of an image aperture that conceals outer facial features. Next we describe additional features of AFA that make it an attractive option for stimulus preparation, such as the use of Generalized Procrustes Analysis (GPA) for alignment. Finally, we compare AFA to similar non-commercial software solutions. [WebMorphR](https://github.com/debruine/webmorphR) in particular is a very nice open-source suite of stimulus-preparation tools that includes face alignment. WebMorphR is written in R however, while AFA is written in Python. Therefore, AFA may be preferable to some researchers, especially given the wide range of image-processing and deep-learning frameworks available in that language. As this is a software paper, we focus on software usage. However, we also include additional sections on algorithmic detail and validity testing in the spirit of open-science and transparency.

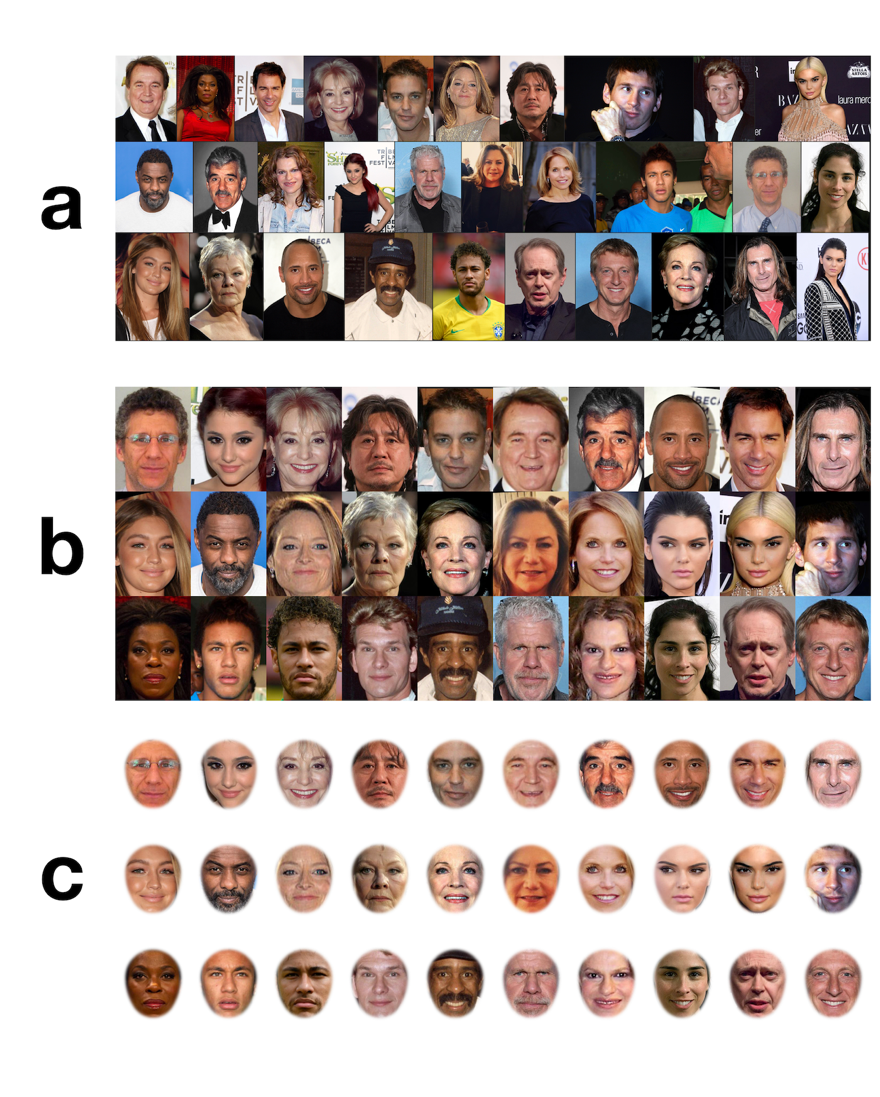


Figure 2. FAD takes a set of images with faces (a) and produces aligned versions of these images so that facial features are overlapping (b). AFA can also place a window over the aligned face images so that only inner facial features are shown (c). All photos licensed under CCL. Specific license info for each face can be found in <https://github.com/SourCherries/fad/demos/align/1_basic/>.

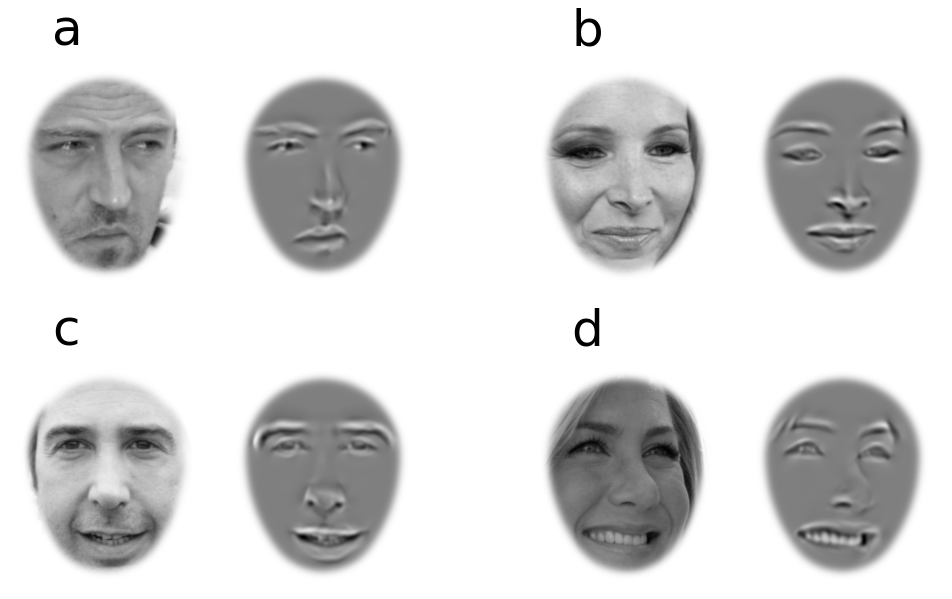


Figure 3. Original photos and wavelet-reconstructed versions of actors from the Friends TV show. (a) Matthew Perry who plays Chandler; (b) Lisa Kudrow who plays Phoebe; (c) David Schwimmer who plays Ross; and (d) Jennifer Aniston who plays Rachel. Wavelet-reconstructed facial features (WRFF) are comprised of 6 internal features (both brows, both eyes, nose and mouth) that can be independently manipulated and share a common background (see Figures 6 and 7. In this figure, all faces are set in an aperture to allow comparison of inner facial features between original and corresponding WRFF without influence of external features. All photos licensed under CCL. Specific license info for each face can be found in <https://github.com/SourCherries/fad/results/friends/>.

Additional features of AFA

Generalized Procrustes Analysis (GPA)

Due to the time-consuming nature of manual landmark estimation, most studies align faces on the basis of only 2 or 3 landmarks: the centers of the pupils and sometimes the center of the mouth. Alignment procedures are often not fully reported in publications; even when they are reported, they are quite variable. Examples of clearly detailed alignment procedures from existing studies are summarized in Table 1. Dupuis-Roy et al. (2014) is a notable exception to the norm, in that they devise a sophisticated method to ensure face alignment that does not bias toward perfect alignment of any single landmark, while allowing greater variability in other landmarks. This type of unbiased alignment can be quite advantageous for a number of reasons. For example, any kind of pixel-based analysis of image information like an ideal observer model (J. M. Gold et al., 2012; Shen & Palmeri, 2015) or Gabor-jet model (Xu et al., 2014) will naturally depend on how well different types of facial features overlap. Greater variance in the position of one facial feature relative to another will then lead to a higher concentration of information in that feature (Shen & Palmeri, 2015), which can be seen as an artifact if it wasn’t explicitly controlled for or modelled. And even in studies where this type of modelling is not an explicit part of the underlying hypothesis, biased alignment might still be cause for concern considering how rapidly and efficiently human participants can isolate such concentrations of image-based information (Peterson et al., 2009).

Table 1. Previous alignment types.

|  |  |  |  |
| --- | --- | --- | --- |
| **Authors** | **Scaling** | **Rotation** | **Translation** |
| (Dakin & Watt, 2009b) | Normalize distance between center of mouth and midpoint between eyes. | Vertical line connecting center of mouth and midpoint between eyes. | Center of image is midpoint between eyes. |
| (Gronenschild et al., 2009) | Average semi-major and -minor axes of fitted ellipse. | Not specified. | Ellipses position on each face determined by eye centers, with common aspect ratio. Ellipses translated to overlap after scaling. |
| (Dal Martello & Maloney, 2010) | Distance between pupils. | Pupils aligned. | Pupils same position. |
| (Dupuis-Roy et al., 2014) | Minimize mean-square difference between 20 landmarks of each face with average of those landmarks across all faces. | Same. | Same. |
| (Gilad-Gutnick et al., 2018) | Distance between eyes. | Not specified. | Nose tip. |

Following the lead of Dupuis-Roy et al. (2014), AFA also adopts an alignment algorithm that aims at an unbiased global alignment of all landmarks. AFA uses Generalized Procrustes Analysis (GPA) which is a common procedure for normalizing *shapes* in a diverse range of research disciplines like ecology and evolutionary biology (Adams & Otárola-Castillo, 2013; Gower, 1975; Rohlf & Slice, 1990). Significantly, GPA is also an important part of preprocessing images that are used to train machine-learning models of face detection (Sagonas et al., 2013, 2016). A *shape* is simply a collection of landmarks that describe the form of an object like an animal’s skeleton. One goal in ecology and evolutionary biology is to analyze a pure form of these shapes that is invariant to size, position and orientation. In order to do this, each shape must be affine transformed to an ideal reference shape such that corresponding landmarks across shapes are as close together as possible. Whereas conventional face alignment might only ensure alignment of pupils or only alignment of the tip of the nose, AFA uses the whole shape of eyebrows, eyes, nose and mouth (51 different landmarks altogether). In other words, using GPA-based face alignment naturally results in the type of unbiased alignment that we described earlier. As one can see, the method used by Dupuis-Roy et al. (2014) has a similar goal. The critical difference between the method used by Dupuis-Roy and GPA is that GPA estimates the reference shape using an iterative process that leads to the greatest minimization of distance between corresponding landmarks (described in Figure 5). Dupuis-Roy et al. (2014) use the average of landmarks across faces as the reference, which has the following problem: if there is great variability in the global position, orientation or size across faces, then the average of landmarks may be highly distorted. GPA’s iterative process for estimating the best reference shape for alignment does not have this problem. As described in the Evaluation section of this paper, AFA does quite well. Please see Figure 7 which shows distributions of landmarks across various databases of faces that we already aligned using AFA. As one can see, positional distributions are equally small across the different landmarks. For emotionally expressive faces with large variability in mouth and eyebrow shape, one can set AFA to align using only the eye landmarks. The results of eyes-only alignment of emotionally expressive faces is shown in Figure 8. We recommend that GPA-based alignment should be the default method for face studies that have no particular theoretical stake in how face alignment is done

Warping functionality

The main function of AFA is to align faces using affine transforms, which preserve shape (the relative positions and lengths of landmarks within an object). However, it is also possible to perform non-affine transformations, which distort shape. We refer to these additional transformations as image warping. AFA provides basic functions for general purpose warping, and also higher-level functions that perform 2 of the most popular face warping procedures in the literature: morphing (gradually transforming one face into another), and enhanced averaging (warping faces to a common set of landmarks prior to averaging). An example of face morphing produced by AFA is shown in Figure 2, and an example of an enhanced average produced by AFA is shown in Figure 3.



Figure 4. Morphs images produced by FAD that gradually transform [Choi Min-sik](https://en.wikipedia.org/wiki/Choi_Min-sik) (on the far left) into [Sarah Silverman](https://en.wikipedia.org/wiki/Sarah_Silverman) (on the far right) in 9 equal steps. Image of Choi Min-sik is by Petr Novák [Licensed under CC BY-SA 3.0] and image of Sarah Silverman is by Joan Garvin [Licensed under CC BY 2.0].

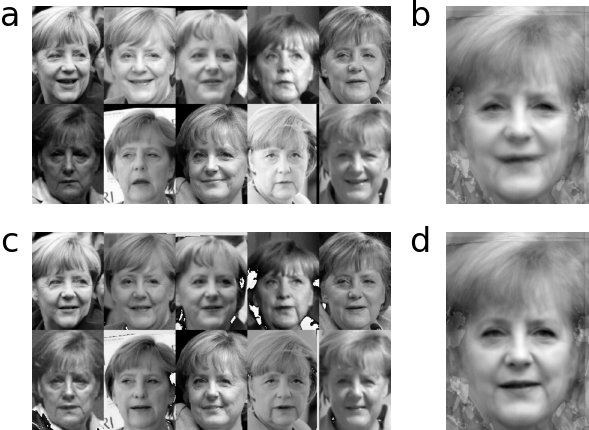


Figure 5. Morphing can also be used to generate an enhanced average of a given facial identity. Comparison of the average of Angela Merkel faces (b) and the enhanced average of Angela Merkel faces (d). The normal average (b) is simply the mean of all the AFA-aligned faces in (a). The enhanced average is the mean of all the shape-corrected faces in (c). Landmarks in the shape-corrected faces share the exact same positions because each has been warped to the average of landmarks across faces in (a). Facial features are more well-defined in the enhanced average, with especially high local contrast around the lips and nostrils. All photos of Angela Merkel are CC license; license info for each photo in LICENSE file of demo\_3\_averaging folder of repo.

Comparison to similar software

Here we compare AFA against 3 similar types of free (non-commercial) software solutions for face alignment: InterFace (Kramer et al., 2017); Facemorpher (<https://github.com/stheakanath/facemorpher>); and WebMorphR (<https://github.com/debruine/webmorphR>). Each tool, AFA included, has its own advantages and disadvantages (shown in Table 2). Compared to InterFace and Facemorpher, AFA and WebMorphR are better suited for the construction of face stimuli that are reproducible. In fact, AFA and WebMorph are almost identical except for the fact that WebMorph is written in R and AFA is written in Python. We highly recommend WebMorph as a fantastic tool for those whose primary programming language is R. However, AFA is a great option for those whose primary programming language is Python. AFA may be a more suitable option for those wishing to expand on stimulus construction using the ever-growing ecosystem of popular Python packages in the areas of image processing ([scikit-image](https://scikit-image.org/)), perceptual-systems modelling ([plenoptic](https://github.com/LabForComputationalVision/plenoptic) and [pyrtools](https://github.com/LabForComputationalVision/pyrtools)) and perceptual modelling, and deep neural networks ([TensorFlow](https://www.tensorflow.org/) and [PyTorch](https://pytorch.org/)).

Table 2. Comparison of similar software.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Software | AFA | webmorphR | Facemorpher | InterFace |
| Auto landmarks | Yes | Yes | Noa | No |
| Procrustes alignment | Yes | Yes | No | Unknown |
| GPA | Yes | Yes | No | Unknown |
| Alignment evaluated | Yes | Partiallyc | No | No |
| Landmarks provided | Yes | Yes | No | No |
| Oval aperture | Yes | Yes | No | No |
| Moss’s egg aperture | Yes | No | No | No |
| Open source | Yes | Yes | Yes | No |
| Scriptable | Yes | Yes | Yes | No |
| Programming language | Python | R | Python | NAb |
| Unit tested | Yes | Yes | No | Unknown |

Note. aAlignment by bounding box around detected face. bProvided as a Matlab binary application; might be scriptable if code is obtained from the authors. cDLIB underlying WebMorphR was evaluated (Jones et al., 2021), but not the entire process of alignment.

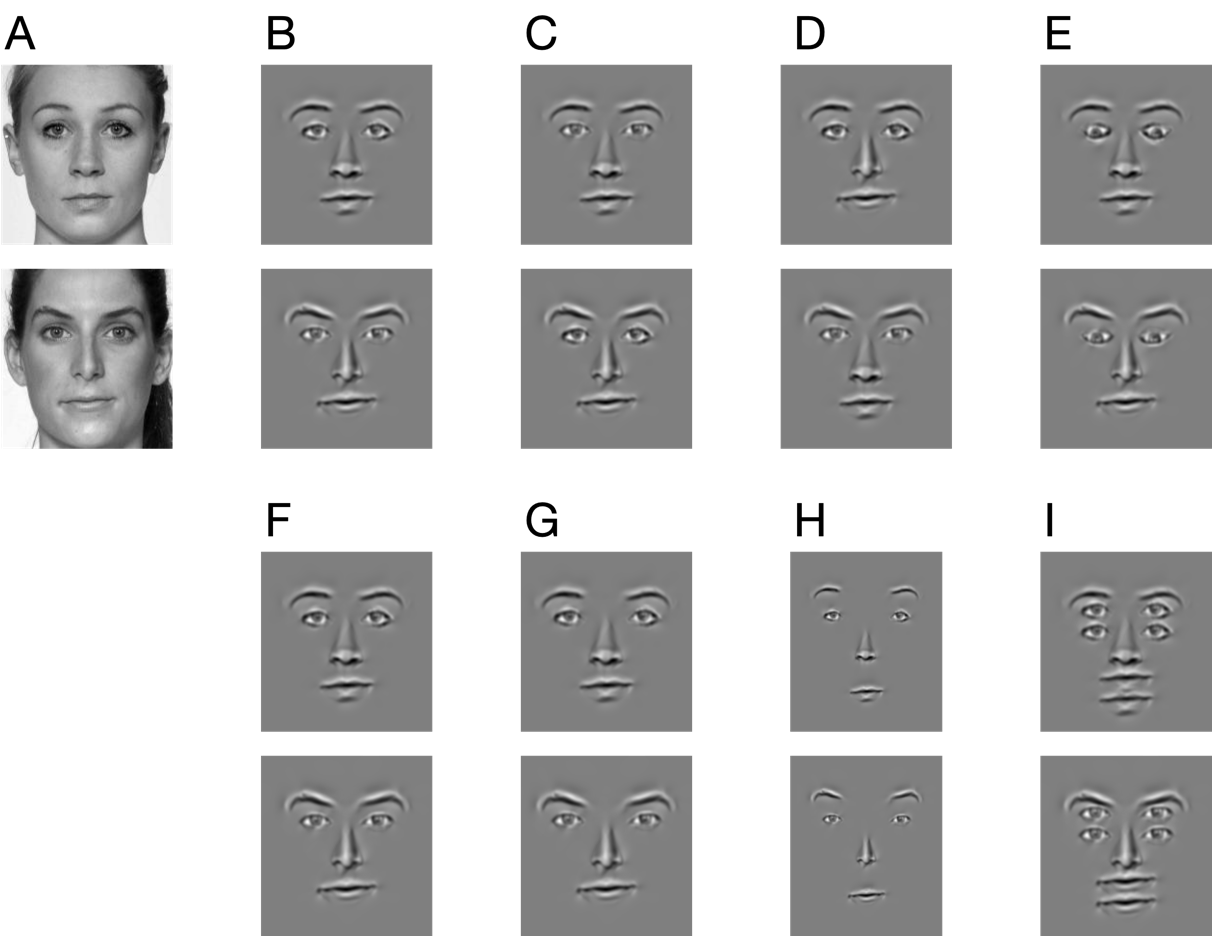


Figure 6. Automated re-arrangements of facial features (using WRFF) that represent popular experimental stimuli (columns c, d, e, g, h and i). Each lettered section represents 2 different facial identities (top and bottom). WRFF in section (b) are generated from original sources in column (a). WRFF in section (f) are the same as (b); repeated in this row of the figure for comparison with experimental stimuli (g, h and i). Stimuli in section (c) are typically used to study the whole-part effect (eyes swapped between identities in this example). Stimuli in section (d) are chimeric faces (bottom halves are swapped between identities) that can be made using the same FAD command as for whole-part stimuli. Stimuli in section (e) are Thatcher faces, which have eyes inverted (rotated 180-degree) relative to the other features. Stimuli in section (g) represent a manipulation of feature spacing (second order relations). In this example, eyes and brows are shifted outward to achieve a 50-percent increase in inter-eye distances. Stimuli in section (h) show features spaced out from a common center to alleviate crowding in the periphery. Stimuli in section (i) show the double-face illusion. Stimuli in all sections can be created with a single FAD command.

Main concepts

Facial landmarks

As stated in the Introduction, a landmark is the x and y pixel position of a distinctive facial part like the left corner of the mouth or the right pupil center. AFA uses DLIB to automatically detect 68 different landmarks, as shown in Figure 2. Specifically, DLIB detects 5 landmarks for the left eyebrow, 5 for the right eyebrow, 6 for the left eye, 6 for the right eye, 9 for the nose, 13 for the outline around the outer edge of the mouth, and 7 for the inner mouth. DLIB also detects landmarks for the jawline but by default AFA does not use jawline for alignment, as those are not reliable. By default, AFA uses 51 landmarks for face alignment. When AFA estimates landmarks, a JSON-formatted file called landmarks.txt is saved, which contains the landmarks for all face images analyzed. JSON is a widely adopted format for hierarchical data, and the availability of landmarks.txt in this format should promote the extensibility of AFA by the scientific community. Details of landmarks.txt are provided on the GitHub repository. Each entry in the landmarks.txt file is the filename of the image analyzed, followed by a list of labelled vectors corresponding to left eyebrow, right eyebrow, left eye, right eye, nose, mouth outline and inner mouth. Each vector is a list of coordinates alternating between x and y values; i.e., values are arranged in pairs of x and y, so each pair is a single landmark.

Affine transformations

Affine transformations are image transformations like translation, resizing and rotation. In AFA, face alignment only uses affine transformations that preserve shape. For example, face alignment using AFA does not involve image stretching or shearing. However, AFA provides additional functions for the generation of morphed images and enhanced facial averages. Those additional functions make use of non-affine image transforms.

Procrustes analysis and Generalized Procrustes analysis

In AFA, a *shape* refers to the set of landmarks associated with a given face. Procrustes analysis determines the translation, scaling and rotation required to align one (target) shape as closely as possible to another (reference) shape. Typically, a least-squares criterion applied to the distances of landmarks between shapes is used; and that is what AFA uses. Notice that Procrustes analysis is asymmetric. For example, if the target is shaped like the letter T and the reference is a tilted T then we get the transform required to tilt an upright T appropriately. In order to align more than one face, we need to use Procrustes analysis to align each shape to a single reference shape. For AFA, the reference shape is determined using Generalized Procrustes analysis (GPA), which is described in Figure 5.

Facial feature extraction

Automatic alignment is performed to ensure that facial features overlap as much as possible. As shown in results, deviation in feature spacing is minimal (🚧Section Reference🚧). Therefore, once features are generated, feature swapping (e.g., chimeric faces) can be performed with a simple centering operation. Interpupilary distance is set to be constant. While not necessary, constant interpupilary distance makes it easier to generate uniform spatial-frequency characteristics across faces; i.e., a single set of wavelet filters can be defined for an entire set of faces.

Each face is decomposed into a set of 6 images of inner facial features that can be recombined in various ways -- left eyebrow, right eyebrow, left eye, right eye, nose and mouth. Each facial-feature image is a wavelet-based reconstruction, ensuring none of the edge artifacts associated with conventional image-based cutting and pasting. We call each feature a Wavelet-Reconstucted Facial Feature (WRFF). To generate WRFFs, we first establish a ROI for each feature, perform a wavelet transform on the image, and then selected wavelet coefficients within each ROI.

Each facial feature is associated with a set of automatically detected landmarks. Region-of-interests (ROI) are based on these landmarks. ROI were initialized differently for eyebrows and other features, but each ROI consists of a binary window the same size as the entire face image. Initial eyebrow ROI were constructed by connecting landmarks into a multi-segment line and then dilating until ROI approximately encompasses eyebrow hair. Initial ROI for other features were the convex hull of the associated landmarks. Final ROI were obtained by dilating all of binary windows incrementally until they are as large as they can be without any overlap among ROI. ROI used to generate the WRFF of a specific face can be easily recomputed for feature rearrangements and background separation (as in Figures 6 and 7).

A wavelet-transform was performed on each aligned face photo. Wavelet parameters were carefully chosen to ensure close correspondence with characteristics of simple cells estimated from human physiology, and the psychophysics of face identification. For example, log Gabor were chosen as the wavelet basis, partly because their transfer functions in the Fourier domain are symmetric in log space - just like human simple cells (Anderson & Burr, 1985; Wilson et al., 1983). Each wavelet has a fullwidth frequency bandwidth (at half-height) of approximately 2 octaves, which falls into the range of human simple cells - from 0.5 to 3 octaves (Daugman, 1985; Webster & De Valois, 1985). We designed wavelets across 6 orientations (increments of 30 degrees) and 3 spatial frequencies. We chose spatial frequencies that center around those that are optimal for face identification by human observers: 6, 10 and 16.6 cycles per face width (Gaspar et al., 2008; J. Gold et al., 1999; Näsänen, 1999).

**\*\****\*Perhaps move this paragraph to Discussion\****\*\***

Note that the wavelet transform we chose is an overcomplete basis. Unlike wavelet transforms associated with popular image compression methods, there are no off-the-shelf algorithms. However, our code was partially adapted from Matlab code (https://www.peterkovesi.com/matlabfns/) used to estimate perceptually meaningful features based on local phase-congruency (Kovesi, 2000). Unlike prior attempts to deconstruct the human face with wavelets in a perceptually meaningful way (Xu et al., 2014; Yue et al., 2012), the transform procedure of Kovesi (2000) does attempt to mirror human physiology in as close detail as possible. Incidentallly, this same procedure has also been widely used in practical computer vision applications (Despotović et al., 2015; Moccia et al., 2018).

**\*\****\*Keep this here\****\*\***

For each facial feature, 50-percent of the wavelet coefficients within the corresponding ROI with the largest real values (even phase) were selected. Each facial feature was then reconstructed from the sum of those even-phase wavelets (in the spatial domain). Virtually identical results are obtained from selecting and reconstructing from only odd-phase wavelets notwithstanding a simple translation of the entire image by a few pixels. For each individual face, relative contrast across WRFFs was preserved whilst maximizing the grayscale range in a way that keeps the background value constant across all faces (127 is the default).

Windowing

In many studies, faces are shown behind an image aperture so that only the inner face can be seen. Faces are aligned first and then a single aperture is used for all faces, with aperture dimensions chosen to show as much of the inner face as possible while concealing parts of the outer face like the ears, hair and jawline. Various shapes of aperture are used but the most common is an ellipse. AFA can automatically determine aperture shape using landmark data, and then generate a set of faces that are both aligned and set within an aperture. Output faces are in PNG format so that the aperture can exist as an alpha channel in each image; therefore, preserving the original full image in each file, and providing an easy way for researchers to recover the aperture for further analysis and documentation. AFA provides two options for aperture shape: ellipse and Moss’s Egg.

Ellipse

When AFA performs face alignment, image dimensions (width and height) are the same across all aligned images, and each face is centered (see Figure 4d). To construct the elliptical aperture, AFA sets the vertical axis of the ellipse to be image-centered and have a length that is a fixed proportion of image height. The length of the horizontal axis is then calculated by AFA to be just widen enough to include landmark positions across all faces.

Moss’s Egg

A more attractive alternative to an ellipse-shaped aperture is one shaped as a Moss’s Egg. An ellipse is vertically symmetric and therefore introduces an undesirable trade-off between including too much of the outer-face in the top-half (like hair) or too much of the outer-face in the bottom-half (like jawline). A Moss’s Egg is not vertically symmetric (it’s an ovoid) and therefore better suited to capture the greater spatial extent of facial features in the top half of the face. Therefore, compared to an ellipse, a Moss’s Egg does a better job of maximizing inner-face shown while minimizing outer-face shown. The method AFA uses to fit a Moss Egg to a set of faces is described in Figure 4 and examples of faces set in a Moss Egg aperture can be seen in Figure 1c and Figure 2.

Contrast

Contrast is the difference between light and dark and is one of the most important independent variables in visual psychophysics. Luminance contrast is outside the scope of AFA, as its specification depends on display technology and measurement ones display with a photometer. Nonetheless, AFA documentation and function calls offer transparency in how the RGB values [0-255] of an image are treated, and also provides convenience functions for manipulating both overall image contrast and contrast only within an aperture that excludes the outer face. Specifically, AFA’s default behavior is to scale each image to range from 0 to 255 (maximum contrast range for an image) as the final step of alignment. After alignment, one may want to set the faces within a common aperture. By default, AFA will scale pixel values within the aperture region to range from 0 to 255. However, AFA users have other options for setting within-aperture contrast. For example, AFA users can scale each inner-face region to have as high a contrast as possible whilst retaining a mean pixel value of 127. This is appropriate when making an average face image. It is also standard practice for those who want a common average luminance, although that will only be roughly true as there is a nonlinear relation between pixel value and luminance on most physical displays. Other options can be discovered by examining the contrast\_tools module of AFA.

Warping

Warping is a non-affine image transformation; it changes the shape of image content. In the context of AFA, the goal is to warp the face so that facial landmarks are in new positions. Those new positions can be defined however one wants. When morphing from one facial identity to another, the new positions will be a weighted sum of landmarks between the 2 identities. When constructing an enhanced average, the new positions will be the average of landmarks across instances of the same individual. AFA follows the standard warping algorithm used in popular Active Appearance Models (Matthews & Baker, 2004). AFA warping will use Delaunay triangulation to decompose an image into triangles whose vertices are the facial landmarks detected by DLIB. Corresponding triangles, defined by the same vertices, are calculated for the new positions (the desired landmark locations). Image content in the source triangles will then be mapped on to corresponding triangles defined by new positions, using bilinear interpolation. This same warping procedure forms the backbone of many influential generative models of facial appearance (Yu et al., 2012).

Availability and installation

AFA is an open-source GitHub repository available at <https://github.com/SourCherries/auto-face-align> .

AFA has been successfully tested on Windows, macOS and Linux (Ubuntu) platforms.

AFA has 3 dependencies:

* Python version 3.9 or higher
* DLIB (King, 2009)
* scikit-image (van der Walt et al., 2014)

DLIB and scikit-image each have their own dependencies, such as Numpy and Matplotlib. The installation method described below takes care of all of that.

It is highly recommended that conda is installed and used for the installation of AFA. If you do have conda, then do the following to install:

conda create --name myenv conda-forge::dlib "python>=3.9" scikit-image

conda activate myenv

pip install "alignfaces @ git+https://git@github.com/SourCherries/auto-face-align.git"

This will create a new virtual environment called myenv. You can use another name for that. You will need to activate that environment using conda activate myenv whenever you want to use AFA. To deactivate this environment, simply type conda deactivate myenv. Windows users may encounter a problem with plotting. That is a general issue with Matplotlib on some versions of Windows. To fix, simply type the following while is activated:

conda install freetype=2.10.4

Use of the software

All of the uses described below are written as Python scripts available for anyone to use as templates for their own research code (<https://github.com/SourCherries/auto-face-align/tree/main/demos> and <https://github.com/SourCherries/auto-face-align/tree/main/results> ).

The main purpose of this Python package is to align and window a set of face images. And for most users, the following Python script will generate the appropriate set of aligned and windowed face images:

import alignfaces as afa

faces\_path = "/Users/Me/faces\_for\_my\_study/"

afa.get\_landmarks(faces\_path)

aligned\_path = afa.align\_procrustes(faces\_path)

afa.get\_landmarks(aligned\_path)

the\_aperture, aperture\_path = afa.place\_aperture(aligned\_path)

The variable faces\_path specifies where the original face images reside on your computer. This folder can have a complicated structure with images residing in various subdirectories of this folder. As an example, the original face images might look like those in Figure 1a.

The call to afa.get\_landmarks() measures landmarks for every face and writes those values to a JSON-formatted file called landmarks.txt, placed under faces\_path. This is necessary before alignment can be done.

The call to afa.align\_procrustes() uses GPA to align all of the face images and writes those new images to a new folder whose full path is in the variable aligned\_path. These aligned face images will also be cropped to the same image dimensions so that, on average, the full head is just contained within the image. As an example, the aligned face images might look like those in Figure 1b.

If one then wants to place a window over each face to show only the inner facial features, then landmarks must be estimated again on the aligned faces. Then calling afa.place\_aperture() will create a new set of face images that are both aligned and windowed appropriately. Those new images will be put into a new folder whose full path name is in the variable aperture\_path. As an example, the windowed face images might look like those in Figure 1c.

Handling interruptions

Landmark detection in AFA is fast because DLIB is fast. However, processing a very large database can take time. If landmark detection is interrupted in the middle of processing a large batch of faces, that work will not be lost because AFA creates temporary files during processing. In order to recover previous calculations and begin landmark detection at the file when interruption occurred, simply rerun af.get\_landmarks() with start\_fresh set to True:

af.get\_landmarks(my\_faces\_path, file\_prefix, file\_postfix, start\_fresh=False)

All supplied arguments are the same as previously described, except start\_fresh whose default value is True.

Verification

The Python script showing an example of how to do verification is [https://github.com/SourCherries/auto-face-align/tree/main/results/results\_1\_preprocessing.py](https://github.com/SourCherries/auto-face-align/tree/main/results/results_1_preprocessing.py%20./results_1_preprocessing.py).

DLIB is highly reliable. Based on our own evaluation of DLIB described in Results, failure rates are very rare – unless photographic subjects are wearing glasses or have hair covering their face.

Nonetheless, we recommend that each set of landmarks is visually inspected to confirm accuracy. After landmark estimation, one can call this function:

afa.plot\_faces\_with\_landmarks\_one\_by\_one(faces\_path)

One-by-one, each face image will be shown with landmarks overlaid as small red discs and the corresponding file name. The user should record file names for all faces with inaccurate landmarks to a CSV format file called bad-landmarks.csv within faces\_path. After creating this file, run this function:

afa.exclude\_files\_with\_bad\_landmarks(faces\_path)

This function ensures that the images listed in bad-landmarks.csv are excluded from all subsequent analyses. Basically, a new landmarks.txt file is created with the bad images removed and the original is saved as landmarks-original.txt.

Alignment

The Python script showing an example of how to do alignment is <https://github.com/SourCherries/auto-face-align/tree/main/demos/demo_1_alignment> .

AFA’s alignment function has few parameters that should be explained:

aligned\_path = afa.align\_procrustes(faces\_path, file\_prefix='',

file\_postfix='jpg', exclude\_features=['jawline', 'left\_iris', 'right\_iris', 'mouth\_inner'],

include\_features=None, adjust\_size='default', size\_value=None, color\_of\_result='grayscale')

The values for faces\_path, file\_prefix and file\_postfix should be the same as previously used during the call to afa.get\_landmarks().

The default for color\_of\_result is “grayscale” but can be set to “rgb” to generate aligned faces that are in color.

The parameter exclude\_features is a list of labels for facial features that are to be excluded from consideration when aligning the faces. By default, jawline is excluded and we recommend to users that it always be excluded. In our experience, all of the landmarks returned by DLIB are highly accurate except for those associated with jawline. In fact, the default parameters result in the best alignments. If one would rather base alignment on a few facial features then it is easier to provide a list of labels to include\_features, which overrides any arguments provided to exclude\_features. Providing an argument to include\_features can be useful if for example one wanted to perform a more conventional alignment that is based only on the eyes like this:

include\_features = ['left\_eye', 'right\_eye']

The parameter adjust\_size can be set to one of 4 different strings: set\_eye\_distance, set\_image\_width, set\_image\_height or default. This allows the user to set the size of the aligned images based on either eye-distance, image width, image height or to use default values for all of these dimensions. If non-default values are used then size\_value must be set to the value required in pixel units. For example, we can supply these arguments to afa.align\_procrustes()

adjust\_size="set\_image\_width", size\_value = 256

This ensures that all aligned images have a width of 256 pixels, and the rest of the dimensions are set according to fixed proportions among these dimensions. Those fixed proportions are important to ensure that the entire head is contained within the image, and any subsequent generation of an image window (next section) will work correctly.

Windowing

The Python script showing an example of how to do windowing is <https://github.com/SourCherries/auto-face-align/tree/main/demos/demo_1_alignment> .

Remember to first align the faces and then measure the landmarks for those aligned faces.

AFA’s windowing function has few parameters that should be explained:

the\_aperture, aperture\_path = afa.place\_aperture(aligned\_path,

file\_prefix, file\_postfix, aperture\_type="MossEgg", contrast\_norm="max", color\_of\_result="rgb")

You should supply the same arguments to file\_prefix and file\_postfix as before. aligned\_path is the folder where your aligned faces reside, which was output when you made the aligned faces.

The parameter color\_of\_result can be set to either rgb for color or grayscale. There are two types of aperture shape: MossEgg, which is described in this paper, and Ellipse. MossEgg is the preferred aperture shape for showing as much of the inner facial features as possible while showing as little of the rest of the face.

When contrast\_norm is set to max, we ensure that image values within the aperture area have the largest range possible - between 0 and 255. See the contrast\_tools module for other options.

Morphing

The Python script showing an example of how to do morphing is <https://github.com/SourCherries/auto-face-align/tree/main/demos/demo_2_morphing> .

The easy and recommended way to make morphs is to create a new folder that contains only the 2 faces we wish to morph into each other. We specify the full pathname of that folder like this, for example:

source\_dir = “users/bob/face\_experiments/an-unusual-pair/”

Next we align the faces and then measure the landmarks for those aligned faces:

af.get\_landmarks(source\_dir, file\_prefix, file\_postfix)

aligned\_path = af.align\_procrustes(source\_dir, file\_prefix,

file\_postfix,

color\_of\_result="rgb")

af.get\_landmarks(aligned\_path, file\_prefix, file\_postfix)

Now we are ready to morph the 2 aligned faces within the folder aligned\_path :

face\_array, p, morph\_path = afa.morph\_between\_two\_faces(aligned\_path, do\_these=[0, 1],

num\_morphs=num\_morphs,

file\_prefix=file\_prefix,

file\_postfix=file\_postfix,

new\_dir = "morphed",

weight\_texture=True)

Whenever there are just 2 faces within aligned\_path we always set do\_these=[0, 1] . num\_morphs is the number of morphing steps. new\_dir is set to “morphed” so all the morphed images will be written to a new directory whose name is the concatenation of the aligned\_path and “morphed”. When weight\_texture is true then intermediate morphs are weighted combinations of the images, which is preferable.

Our call to morph\_between\_two\_faces produces all of the morph images and writes them to the folder specified by morph\_path. The filenames of the morphs range from **N0.png** for face A to **N10.png** for face B. If we wanted 20 morphs then the filenames would range from **N0.png** for face A to **N20.png** for face B.

Finally, if we want to window our morph faces so that only inner facial features are shown, then we can do this:

the\_aperture, aperture\_path = af.place\_aperture(morph\_path, "N",

"png",

aperture\_type="MossEgg",

contrast\_norm="max",

color\_of\_result="rgb")

You may not wish to create a new folder with only 2 faces to morph. Instead, you may prefer to specify a pair of faces from a folder containing many faces. In that case, you need to determine the indices of the 2 faces you would like to morph into each other. First, examine a list of all the images that you’ve aligned:

image\_files = afa.get\_source\_files(aligned\_path, file\_prefix, file\_postfix)

The variable image\_files is a list of all the aligned images. Examine this variable and find the indices of the 2 images you would like to combine. For demo 2, Choi-min Sik and Sarah Silverman are images 9 and 29. These are the two faces we wish to combine so:

do\_these = [9, 29]

Now we are ready to morph these 2 faces using a call to afa.morph\_between\_two\_faces .

Enhanced average

The Python script showing an example of how to make an enhanced average is <https://github.com/SourCherries/auto-face-align/tree/main/demos/demo_3_averaging> .

Remember to first align the faces and then measure the landmarks for those aligned faces.

Methods

AFA is based on facial landmark detection that is powered by the reliable and open-source DLIB library (King, 2009). DLIB’s landmark detection has been used in numerous machine-learning and deep neural-net applications, a small fraction of which we cite here (Afchar et al., 2018; Amos et al., 2016; Korshunova et al., 2017; Liu et al., 2017; Nirkin et al., 2018; Scherhag et al., 2018). DLIB is used to preprocess training faces for an open-source deep neural network (Amos et al., 2016). Finally, empirical research demonstrates excellent agreement between DLIB-estimated landmarks and those given by human observers (Jones et al., 2021).

The alignment procedure

Using GPA (shown in Figure 5 and described in the next section), a suitable reference shape (green dots in Figure 4a) is derived from a set of many input shapes (the landmarks for all faces that need to be aligned together). In Figure 4a, the black cross is the average of landmarks (green dots) for x and y axes; which is centered at 0-origin. This reference shape is scaled so that distance between the eyes (solid black line) is specified by the user (94 pixels here). Notice the solid black line is tilted away from horizontal (dashed black line), by 5 degrees. This is dealt with afterward; between the steps described by Figure 4 c and d. Figure 4 b shows an example input shape (red dot for each landmark) overlaid on its corresponding face. The black cross is the average of landmarks for x and y axes in all panels. Note that the facial image has been translated so that the average of landmarks is centered in the image (0, 0). In Figure 4 c, we use Classical Procrustes Analysis (PA) to obtain the optimal rotation and scaling to align the input landmarks (red dots) with the reference (green dots). In this case, required scaling-factor and rotation is 1.2 and -12 degrees, respectively. In Figure 4 d the image has been scaled and rotated appropriately (by 1.2 and -12 degrees). An additional rotation of +5 degrees is done because of the tilt in the reference shown in Figure 4 a. An additional translation shifts the image down by 5/12 times eye-distance, which ensures the entire head is within the image. Therefore, the mean of landmarks (black cross) is now shifted below image center (red cross). Finally, the image is cropped to the area shown in green. This is done by fixed proportions between eye distance, image width and image height. The user can decide which of these 3 dimensions to specify and the rest are calculated using the appropriate proportions (as shown by the area of Figure 4 d shaded in green).

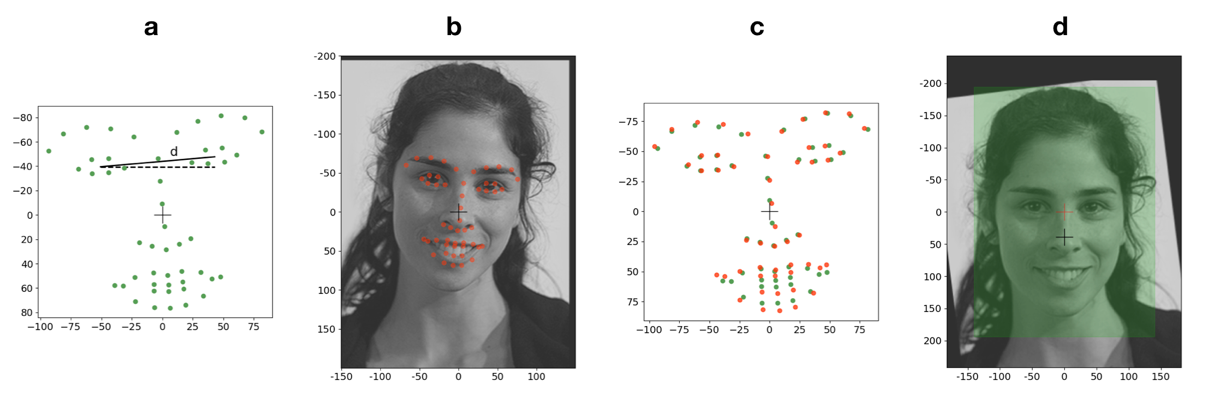


Figure 7. Overview of the alignment procedure. (a) Using GPA (described in Figure 5), a suitable reference shape (green dots) is derived from a set of many input shapes. An example input shape (red dot for each landmark) is shown in the next panel (b in red dots), overlaid on its corresponding face. In panel (c), we use Classical Procrustes Analysis (PA) to obtain the optimal rotation and scaling to align the input landmarks (red dots) with the reference (green dots). In panel (d) the image has been scaled and rotated appropriately (by 1.2 and -12 degrees). An additional rotation of +5 degrees is done because of the tilt in the reference shown in panel (a). Additional translation as well as cropping is performed such that either eye distance, image width or image height is fixed across all images. This image of Sarah Silverman is by [Joan Garvin](https://www.flickr.com/photos/garvinj/%20-%20https:/www.flickr.com/photos/garvinj/136275373/) [Licensed under CC BY 2.0]



Figure 8. Flow chart of General Procrustes Analysis (GPA). The goal is to obtain a suitable reference shape that minimizes the least-squares distances between it and a set of input shapes, when each is superimposed onto the reference. By superimpose, we mean Classical Procrustes Analysis (PA) between pairs of shapes.

Moss’s egg aperture

Figure 6 shows the construction of a landmark-based facial aperture shaped after a Moss’s egg. The Moss’s egg is a combination of 4 circles whose centers are colored discs with corresponding colors in this image. The top part of this aperture lies above a horizontal line connecting the blue and green discs and is made up of the top half of the orange circle. The middle part lies between the aforementioned horizontal line and another that connects the intersections (not shown) between the red circle and each of the green and blue circles. This middle section of the aperture is the intersection of areas within the green and blue circles. Finally, the bottom part of this aperture is the red circle. The orange disc lies ¼ of the way down a vector that goes toward the mean of landmarks (shown in Figure 4) from the average of all eye landmarks. The radius of the orange circle is 47/100 of the width of an ellipse that is centered on the mean of all landmarks and just fits all the landmarks across all faces. The radii of the green and blue circles are each 2 times the orange radius. The green and blue discs each lie a distance from the orange disc equal to the orange radius and are aligned with image horizontal. The vertical position of the red disc is the average y position of the nose-tip landmark across all faces, and its horizontal position is the same as the orange disc. The radius of the red circle is calculated so that it intersects the green and blue circles exactly once.



Figure 9. Construction of a landmark-based facial aperture shaped after a Moss’s egg. The Moss’s egg is a combination of 4 circles whose centers are colored discs with corresponding colors in this image. This image of Sarah Silverman is by [Joan Garvin](https://www.flickr.com/photos/garvinj/%20-%20https:/www.flickr.com/photos/garvinj/136275373/) [Licensed under CC BY 2.0]

Results/performance/evaluation

Python code for the analyses described below are available within the GitHub repository where AFA is made available (<https://github.com/SourCherries/auto-face-align/tree/main/results> ). One must simply request the face databases (described in the next section) from their corresponding authors and put those faces in the appropriate folders; e.g., male faces from the GUFD database should go into results/GUFD-male/. The availability of these analyses scripts brings greater transparency to AFA, and examination of the code can also help users to learn more about AFA functionality.

AFA also comes with a number of unit-tests which help to confirm the validity of functions within AFA. All of these reside in the folder <https://github.com/SourCherries/auto-face-align/tree/main/alignfaces/src/alignfaces/tests> . Looking at these tests may bring greater clarity to the underlying algorithms of GPA and warping for users who are interested in extending AFA functionality.

Landmark detection

To test DLIB landmark detection and our GPA-based alignment procedure in the

typical use-case for psychology researchers, we process and analyze 4 different databases of face photographs. Verification was done by visual inspection and showed landmark placements, and alignments were of good quality in every case (summarized in Table 3). The Python script used to perform this verification of AFA is included in the GitHub repository. All one needs to do is request the face databases from each laboratory, include those image files in the corresponding folders of the GitHub repository, then run the analysis.

The databases we used: the CAS-PEAL database of Chinese faces (Gao et al., 2008); the Glasgow Unfamiliar Face Database GUFD (Burton et al., 2010); faces from the Kent Face Matching Test KFMT (Fysh & Bindemann, 2018); and emotionally expressive faces from the NimStim database (Tottenham et al., 2009). The first 3 databased comprise mostly neutrally expressive faces, which is the most typical kind used in face perception research. The NimStim database is has faces with 8 different categories of emotional expression, along with open- and closed-mouth versions of each. We report results separately for the 3 neutral and the NimStim databases. These 3 emotion-neutral databases were chosen to reflect the diversity of frontal-pose face photos that are normally used psychology and neuroscience research. In particular, GUFD and KFMT are tailored to the study of face perception in realistic scenarios, so the faces from these databases were photographed with variable quality cameras and resolutions and allowed for natural poses instead of a strictly frontal-parallel orientation of the face. Photographic subjects in both the GUFD and KFMT databases were allowed to have their hair worn in any fashion. And unlike the other two databases, photographic subjects in the KFMT database were allowed to wear glasses. Image resolutions in width-by-height (pixels) are 360 x 480 and 350 x 518 for the CAS-PEAL and GUFD databases, respectively. KFMT was divided between KFMT-DC and KFMT-ID which were photos taken in-lab (283 x 332 pixels) and pre-existing photo-identification (142 x 192) obtained from the same people. Given variability in the area occupied by faces across these databases (due to camera distance and focus), perhaps a better metric for image resolution that allows comparison across databases is the mean eye distance measured using AFA. Those eye distances (pixels) are listed in Table 3. As one can see from Table 3, landmark detection for the 3 neutral databases is quite robust.

Emotionally expressive faces

Can emotionally expressive faces be successfully aligned using AFA?

DLIB's landmark detector was trained mostly on faces either a near neutral expression, or a slight smile. It is possible that DLIB may either fail to detect a face or provide poor estimates of landmarks. However, our results suggest that DLIB is quite robust as long as mouth landmarks are excluded from consideration. Results for the NimStim database of emotionally expressive faces are shown separately in Table 4. There were no failed face detections so Table 4 shows only the numbers of failed landmark placements in each category of emotional expression. Visual inspection of landmarks overlaid on the photos show that mouths were the most poorly estimated by DLIB. We can reject a face based on all landmark estimates, including those of the mouth. That provides a conservative estimate of landmark accuracy, with greater numbers of rejected faces. The rows indicating “Considered mouth” in Table 4 show the total number of faces with bad landmarks in each of the emotion categories according to this conservative criterion. Total number in each category are in parentheses. Anger and Disgust are problematic categories when the mouth is allowed to open. However, this is less of a problem when the mouth is closed. Given the poor estimation of mouth landmarks for some categories of emotionally expressive faces (disgust and anger), it might be best to exclude mouth landmarks when aligning faces with AFA. In the rows of Table 4 indicating the mouth is not considered, we show the total number of faces with bad landmarks in each of the emotion categories according to this liberal criterion (not considering the mouth). These results are quite good considering that photographic subjects are repeated across category and a single individual is responsible for most of the failed landmarks in this table.

Table 3. Evaluation of DLIB and AFA across face databases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Database | Eye Distance, pixels a | Race | *n* | Failed Face Detections | Failed Landmark Placement |
| CAS-female | 84 | Chinese | 437 | 0 | 0 |
| CAS-male | 84 | Chinese | 589 | 0 | 0 |
| GUFD-female | 112 | Predominantly Caucasian | 106 | 0 | 0 |
| GUFD-male | 112 | Predominantly Caucasian | 323 | 0 | 0 |
| KUFT-DC | 55 | Mostly Caucasian | 252 | 0 | 2b |
| KUFT-ID | 43 | Mostly Caucasian | 252 | 1 | 10c |
| Total |  |  | 1959 | 1 | 12 |

Note. a Measured on the original photos using AFA b Subjects wearing glasses. c 5 of these subjects wore glasses, 1 with an extreme viewpoint, and 1 with hair covering an eyebrow. Other 3 were non-Caucasian females.

Table 4. Evaluation of AFA for emotional faces (NIMSTIM database)

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Consider mouth | Image category | | | | | | | | |
| Mouth | Angry | Calm | Disgust | Fear | Happy | Neutral | Sad | Surprised |
| yes | Open | 9 (42) | 2 (41) | 31 (42) | 3 (42) | 1 (42) | 1 (41) | 3 (41) | 7 (42) |
| no | Open | 1 (42) | 1 (41) | 1 (42) | 1 (42) | 1 (42) | 1 (41) | 1 (41) | 3 (42) |
| yes | Closed | 1 (42) | 1 (41) | 7 (40) | 4 (37) | 1 (41) | 1 (42) | 4 (41) | 1 (2) |
| no | Closed | 1 (42) | 1 (41) | 1 (40) | 1 (37) | 1 (41) | 1 (42) | 1 (41) | 1 (2) |

Note: There were 0 failed detections in all categories. Additionally, individuals are repeated across category. Therefore, a single individual is responsible for the failed landmarks in most of these cells.

Limitations.

We examined how well DLIB can tolerate rotations in the image-plane. Our approach is the same: Visually confirm the accuracy of landmark-placement by showing landmarks overlaid on its corresponding face image. For a set of 32 unique identities, we inspected landmark placements across image rotations ranging from 10 to 50 degrees, in increments of 10 degrees. The set of faces at 0-degree tilt was already aligned by AFA using default settings; therefore, orientation was measured by the angle of the vector connecting left and right eyes (in the average of landmarks across all 32 faces), with respect to the image x-axis. Visual inspection shows that DLIB produces accurate landmark placements up to 20 degrees of tilt but begins to produce errors at 30 degrees. We have not done extensive testing of this kind because even 20 degrees offset from horizontal is quite unusual for the types of naturally taken face photos that are most commonly used in psychology and neuroscience experiments. However, it is important that users of AFA are aware of this limitation.

Generalized Procrustes Analysis (GPA)

Unit-testing.

Many of the functions required for GPA were extensively unit tested, including the main function that computes an ideal reference shape generalized\_procrustes\_analysis(). Because of its critical role in AFA, that function alone was subject to 7 different unit tests. This included two different comparisons of AFA’s GPA output against that of a popular R package which does the same analysis (Adams & Otárola-Castillo, 2013); and also a unique test of validity which we call the “donut test”

The donut-test. Input shapes were pseudo-randomly generated triangles. Half were random affine perturbations of a “base-shape”. For each of these shapes, we also include a corresponding shape that is equidistant and oppositely directed from the base shape in multidimensional space. The base shape itself is not included as an input shape. And critically, the base shape cannot be recovered from simply averaging all the shapes. However, GPA should be able to recover the base shape if our AFA implementation is correct. AFA recovers the base shape with high precision.

Evaluating the final product of all alignment steps

We can evaluate the final outcome of AFA alignment for each face database by (1) visually confirming that the mean of all aligned faces retains high local contrast around the 51 landmark areas that were used for alignment; and (2) visually comparing the distributions of aligned points for each landmark against the mean aligned face, in order to confirm the small size of these distributions relative to facial features. Additionally, for the neutral databases we can visually confirm the uniformity of spatial distribution size across landmarks, as GPA is meant to align each landmark position across faces as best as possible. Results for the neutral face databases are shown in Figure 7, and they provide positive confirmation of AFA alignment validity. For the NimStim faces, we opted to align only by eye landmarks. Therefore, uniformity of spatial distribution size across landmarks is not expected; instead we expect extremely small spatial distributions for each of the eye landmarks and a high degree of local contrast around the eyes. Results for the NimStim are shown in Figure 8, and they provide positive confirmation of AFA alignment validity.

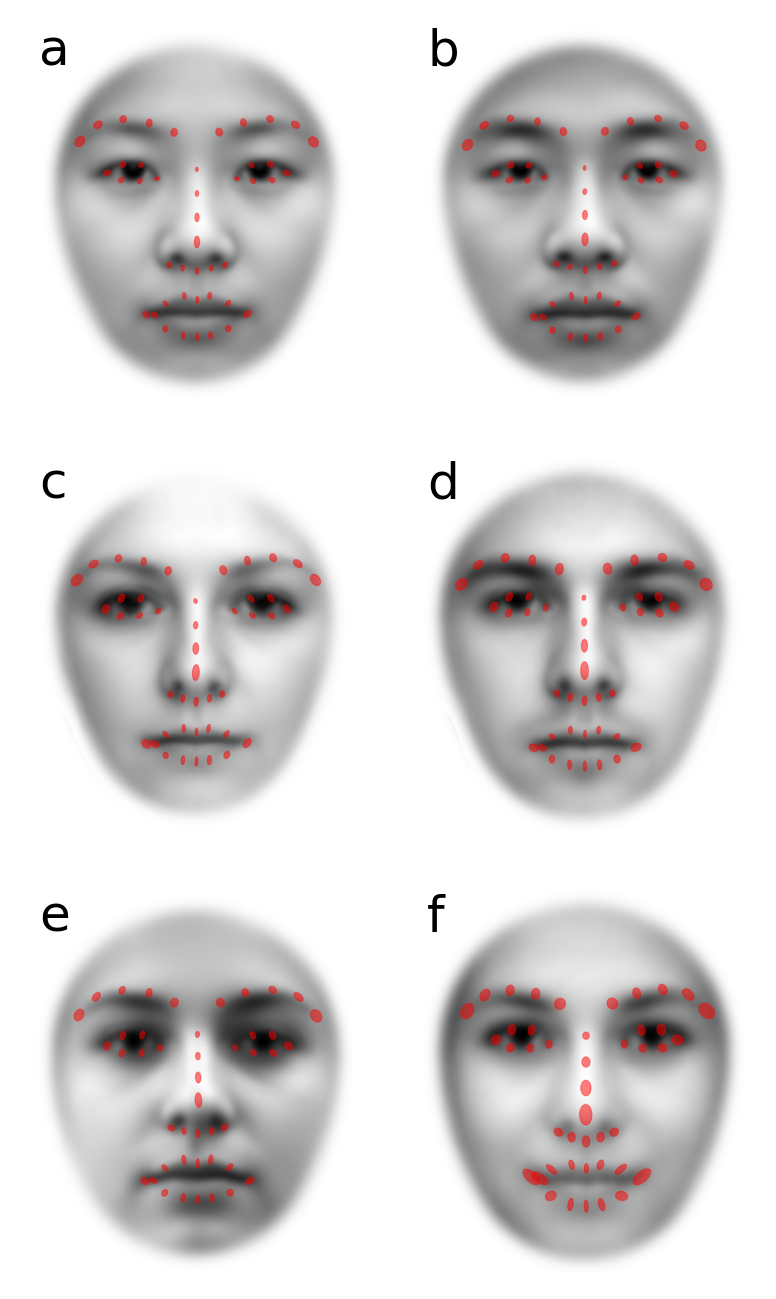


Figure 10. Distributions of landmarks (red ellipses) overlaid on mean of GPA-aligned images. Ellipse orientation based on angle between first and second eigenvectors of PCA applied to landmarks across all faces in a single category. Ellipse length and width are 2 SD of variability along those eigenvectors. Analysis is done separately for 4 categories of faces: Chinese females from the CAS database (a); Chinese males from the CAS database (b); predominantly Caucasian female faces from the GUFD database (c); predominantly Caucasian male faces from the GUFD database (d); in-lab photos of mostly Caucasian faces (mixed gender) from the KUFD database; and low-resolution photo-ID of mostly Caucasian faces (mixed gender and expression) from the KUFD database. Despite SD being a non-robust measure of variability that is strongly inflated by outliers, each ellipse is quite small relative to the size of facial features. Ellipse sizes are also roughly uniform across the different landmarks. The means of aligned faces have high contrast and normal-looking facial features. That provides additional, qualitative validation of the entire GPA-alignment process performed by AFA (including landmark detection).

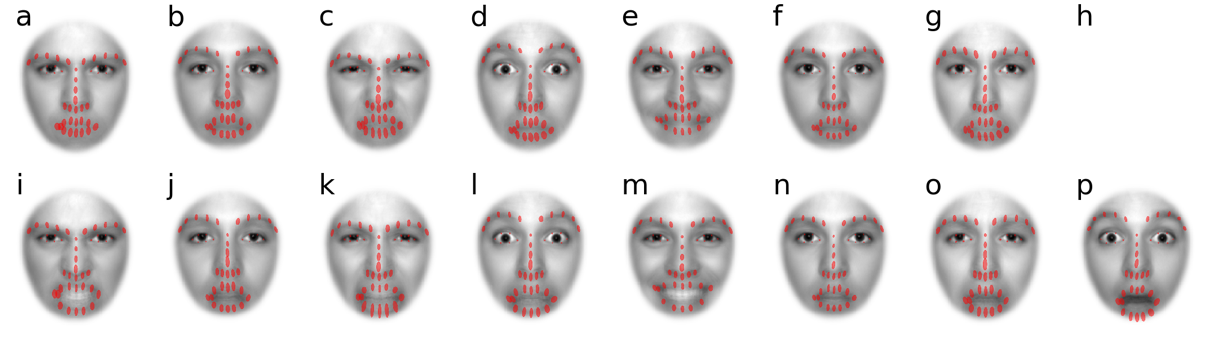


Figure 11. Distributions of landmarks (red ellipses) overlaid on mean of GPA-aligned images of emotionally expressive faces. Unlike in the previous figure, alignments for these faces were based only on eye landmarks. [a to h] corresponds to anger, calm, disgust, fear, happy, neutral, sad, and surprised for closed mouth expressions. There was only 1 face available for closed-mouth surprise so that category has been excluded. [i to p] corresponds to anger, calm, disgust, fear, happy, neutral, sad, and surprised for open mouth expressions. As can be seen by the size of red ellipses around the eyes and the strong local contrast of eyes in the mean faces, AFA does an excellent job of aligning by eye landmarks alone. As expected from common experience, landmark variability is largest around the mouth.

Discussion

Automatic Face Alignment (AFA) is a well-tested, fully documented, open-source tool that can make face perception research more efficient, experimentally rigorous, and transparent for a wide range of researchers. It focusses mainly on one thing - the automatic alignment of face images by facial landmarks – and it achieves this goal using an application of generalized Procrustes analysis which can provide a superior alignment result, as it takes into account 51 different facial landmarks. Various helper functions aid in the confirmation of alignment and in the execution of batch processing. Additional warping functions extend the functionality of AFA but also demonstrate the various possibilities opened up by having an open-source tool for face alignment available to the perceptual research community.

WebMorphR is an alternative face-alignment solution that is written in R. For those who prefer R to Python, WebMorph shares most of the features of AFA described above, and with additional morphing functions. But for those who prefer Python, AFA may be a good choice. The Python ecosystem includes an ever-expanding array of deep-neural network, image processing and perceptual model tools. Therefore, AFA might be a more attractive building block for those wishing to build a sophisticated pipeline for stimulus construction or image analysis.

AFA currently relies on DLIB, which we demonstrate to be quite robust across a wide range of types of face images common in face perception research. However, the modularity of AFA also affords the scientific community the opportunity to push the limit even further by a simple plugging in of new functions. For example, alternative landmark detection can be “plugged in” as an alternative module to face\_landmarks.py. One promising alternative to DLIB is the [face-alignment package](https://github.com/1adrianb/face-alignment) by [Adrian Bulat](https://www.adrianbulat.com/), which is based on a deep neural network (Bulat & Tzimiropoulos, 2017).

The typical use-case will be stimulus preparation for face images taken by individual research groups. However, the efficiency and precision offered by AFA makes other possibilities more feasible. For example, the synthesis of many face databases and large-scale analyses of image statistics for massive face databases.

Open Practices Statement

The evaluation of our open-source software (the empirical portion or “experiment” of this paper) was not pre-registered. The software itself, along with code for performance evaluation, is publicly available on <https://github.com/SourCherries/auto-face-align>. The 4 face databases (the data) we used to evaluate our software are only available by request from the researchers who created those databases (cited in this paper). Many of the most popular face databases used in perception and psychology research are only available this way, and we wanted our evaluation to reflect how AFA would perform with those sorts of images. However, anyone can easily drop new batches of faces into the results folder and run the various evaluation scripts to see for themselves how AFA performs.

References

Adams, D. C., & Otárola-Castillo, E. (2013). geomorph: An R package for the collection and analysis of geometric morphometric shape data. *Methods in Ecology and Evolution*, *4*(4), 393–399.

Afchar, D., Nozick, V., Yamagishi, J., & Echizen, I. (2018). MesoNet: A Compact Facial Video Forgery Detection Network. *2018 IEEE International Workshop on Information Forensics and Security (WIFS)*, 1–7. https://doi.org/10.1109/WIFS.2018.8630761

Amos, B., Ludwiczuk, B., & Satyanarayanan, M. (2016). *OpenFace: A general-purpose face recognition library with mobile applications* (CMU-CS-16-118; p. 20). CMU School of Computer Science.

Anderson, S. J., & Burr, D. C. (1985). Spatial and temporal selectivity of the human motion detection system. *Vision Research*, *25*(8), 1147–1154. https://doi.org/10.1016/0042-6989(85)90104-X

Bulat, A., & Tzimiropoulos, G. (2017). *How Far Are We From Solving the 2D & 3D Face Alignment Problem? (And a Dataset of 230,000 3D Facial Landmarks)*. 1021–1030. https://openaccess.thecvf.com/content\_iccv\_2017/html/Bulat\_How\_Far\_Are\_ICCV\_2017\_paper.html

Burton, A. M., White, D., & McNeill, A. (2010). The Glasgow Face Matching Test. *Behavior Research Methods*, *42*(1), 286–291. https://doi.org/10.3758/BRM.42.1.286

Dakin, S. C., & Watt, R. J. (2009a). Biological “bar codes” in human faces. *Journal of Vision*, *9*(4), 2.1-10. https://doi.org/10.1167/9.4.2

Dakin, S. C., & Watt, R. J. (2009b). Biological “bar codes” in human faces. *J Vis*, *9*(4), 2.1-10. https://doi.org/10.1167/9.4.2

Dal Martello, M. F., & Maloney, L. T. (2010). Lateralization of kin recognition signals in the human face. *J Vis*, *10*(8), 9. https://doi.org/10.1167/10.8.9

Daugman, J. G. (1985). Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters. *JOSA A*, *2*(7), 1160–1169. https://doi.org/10.1364/JOSAA.2.001160

Despotović, I., Goossens, B., & Philips, W. (2015). MRI Segmentation of the Human Brain: Challenges, Methods, and Applications. *Computational and Mathematical Methods in Medicine*, *2015*(1), 450341. https://doi.org/10.1155/2015/450341

Dupuis-Roy, N., Fiset, D., Dufresne, K., Caplette, L., & Gosselin, F. (2014). Real-world interattribute distances lead to inefficient face gender categorization. *J Exp Psychol Hum Percept Perform*, *40*(4), 1289–1294. https://doi.org/10.1037/a0037066

Fysh, M. C., & Bindemann, M. (2018). The Kent Face Matching Test. *British Journal of Psychology*, *109*(2), 219–231. https://doi.org/10.1111/bjop.12260

Gao, W., Cao, B., Shan, S., Chen, X., Zhou, D., Zhang, X., & Zhao, D. (2008). The CAS-PEAL large-scale Chinese face database and baseline evaluations. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, *38*(1), 149–161.

Gaspar, C., Sekuler, A. B., & Bennett, P. J. (2008). Spatial frequency tuning of upright and inverted face identification. *Vision Research*, *48*(28), 2817–2826. https://doi.org/10.1016/j.visres.2008.09.015

Gilad-Gutnick, S., Harmatz, E. S., Tsourides, K., Yovel, G., & Sinha, P. (2018). Recognizing Facial Slivers. *J Cogn Neurosci*, *30*(7), 951–962. https://doi.org/10.1162/jocn\_a\_01265

Goffaux, V., Duecker, F., Hausfeld, L., Schiltz, C., & Goebel, R. (2016). Horizontal tuning for faces originates in high-level Fusiform Face Area. *Neuropsychologia*, *81*, 1–11. https://doi.org/10.1016/j.neuropsychologia.2015.12.004

Gold, J., Bennett, P. J., & Sekuler, A. B. (1999). Identification of band-pass filtered letters and faces by human and ideal observers. *Vision Research*, *39*(21), 3537–3560. https://doi.org/10.1016/S0042-6989(99)00080-2

Gold, J. M., Mundy, P. J., & Tjan, B. S. (2012). The perception of a face is no more than the sum of its parts. *Psychological Science*, *23*(4), 427–434.

Gower, J. C. (1975). Generalized procrustes analysis. *Psychometrika*, *40*(1), 33–51.

Gronenschild, E. H. B. M., Smeets, F., Vuurman, E. F. P. M., van Boxtel, M. P. J., & Jolles, J. (2009). The use of faces as stimuli in neuroimaging and psychological experiments: A procedure to standardize stimulus features. *Behav Res Methods*, *41*(4), 1053–1060. https://doi.org/10.3758/BRM.41.4.1053

Hashemi, A., Roudaia, E., Anderson, N. D., Alain, C., Aleong, R., Khatri, N., Freedman, M., & Sekuler, A. B. (2020). Behavioural and electrophysiological measures of visual processing for early detection of Alzheimer’s disease. *Journal of Vision*, *20*(11), 1624–1624. https://doi.org/10.1167/jov.20.11.1624

Hsieh, P.-J., Vul, E., & Kanwisher, N. (2010). Recognition Alters the Spatial Pattern of fMRI Activation in Early Retinotopic Cortex. *Journal of Neurophysiology*, *103*(3), 1501–1507. https://doi.org/10.1152/jn.00812.2009

Iidaka, T., Yamashita, K., Kashikura, K., & Yonekura, Y. (2004). Spatial frequency of visual image modulates neural responses in the temporo-occipital lobe. An investigation with event-related fMRI. *Brain Research. Cognitive Brain Research*, *18*(2), 196–204. https://doi.org/10.1016/j.cogbrainres.2003.10.005

Jones, A. L., Schild, C., & Jones, B. C. (2021). Facial metrics generated from manually and automatically placed image landmarks are highly correlated. *Evolution and Human Behavior*, *42*(3), 186–193. https://doi.org/10.1016/j.evolhumbehav.2020.09.002

Keil, M. S. (2009). “I look in your eyes, honey”: Internal face features induce spatial frequency preference for human face processing. *PLoS Comput Biol*, *5*(3), e1000329. https://doi.org/10.1371/journal.pcbi.1000329

King, D. E. (2009). Dlib-ml: A machine learning toolkit. *Journal of Machine Learning Research*, *10*(Jul), 1755–1758.

Korshunova, I., Shi, W., Dambre, J., & Theis, L. (2017). Fast Face-Swap Using Convolutional Neural Networks. *2017 IEEE International Conference on Computer Vision (ICCV)*, 3697–3705. https://doi.org/10.1109/ICCV.2017.397

Kovesi, P. (2000). Phase congruency: A low-level image invariant. *Psychological Research*, *64*(2), 136–148. https://doi.org/10.1007/s004260000024

Kramer, R. S. S., Jenkins, R., & Burton, A. M. (2017). InterFace: A software package for face image warping, averaging, and principal components analysis. *Behav Res Methods*, *49*(6), 2002–2011. https://doi.org/10.3758/s13428-016-0837-7

Lai, J., Pancaroglu, R., Oruc, I., Barton, J. J. S., & Davies-Thompson, J. (2014). Neuroanatomic correlates of the feature-salience hierarchy in face processing: An fMRI -adaptation study. *Neuropsychologia*, *53*, 274–283. https://doi.org/10.1016/j.neuropsychologia.2013.10.016

Liu, H., Lu, J., Feng, J., & Zhou, J. (2017). Group-aware deep feature learning for facial age estimation. *Pattern Recognition*, *66*, 82–94. https://doi.org/10.1016/j.patcog.2016.10.026

Matthews, I., & Baker, S. (2004). Active Appearance Models Revisited. *International Journal of Computer Vision*, *60*(2), 135–164. https://doi.org/10.1023/B:VISI.0000029666.37597.d3

Moccia, S., De Momi, E., El Hadji, S., & Mattos, L. S. (2018). Blood vessel segmentation algorithms—Review of methods, datasets and evaluation metrics. *Computer Methods and Programs in Biomedicine*, *158*, 71–91. https://doi.org/10.1016/j.cmpb.2018.02.001

Näsänen, R. (1999). Spatial frequency bandwidth used in the recognition of facial images. *Vision Research*, *39*(23), 3824–3833. https://doi.org/10.1016/S0042-6989(99)00096-6

Nirkin, Y., Masi, I., Tran Tuan, A., Hassner, T., & Medioni, G. (2018). On Face Segmentation, Face Swapping, and Face Perception. *2018 13th IEEE International Conference on Automatic Face Gesture Recognition (FG 2018)*, 98–105. https://doi.org/10.1109/FG.2018.00024

Pachai, M. V., Sekuler, A. B., & Bennett, P. J. (2013). Sensitivity to Information Conveyed by Horizontal Contours is Correlated with Face Identification Accuracy. *Frontiers in Psychology*, *4*, 74. https://doi.org/10.3389/fpsyg.2013.00074

Peterson, M. F., Abbey, C. K., & Eckstein, M. P. (2009). The surprisingly high human efficiency at learning to recognize faces. *Vision Research*, *49*(3), 301–314. https://doi.org/10.1016/j.visres.2008.10.014

Peterson, M. F., & Eckstein, M. P. (2012). Looking just below the eyes is optimal across face recognition tasks. *Proceedings of the National Academy of Sciences of the United States of America*, *109*(48), E3314-3323. https://doi.org/10.1073/pnas.1214269109

Petro, L. S., Smith, F. W., Schyns, P. G., & Muckli, L. (2013). Decoding face categories in diagnostic subregions of primary visual cortex. *The European Journal of Neuroscience*, *37*(7), 1130–1139. https://doi.org/10.1111/ejn.12129

Rohlf, F. J., & Slice, D. (1990). Extensions of the Procrustes method for the optimal superimposition of landmarks. *Systematic Biology*, *39*(1), 40–59.

Sagonas, C., Antonakos, E., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2016). 300 Faces In-The-Wild Challenge: Database and results. *Image and Vision Computing*, *47*, 3–18. https://doi.org/10.1016/j.imavis.2016.01.002

Sagonas, C., Tzimiropoulos, G., Zafeiriou, S., & Pantic, M. (2013). A Semi-automatic Methodology for Facial Landmark Annotation. *Proceedings of the 2013 IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 896–903. https://doi.org/10.1109/CVPRW.2013.132

Scherhag, U., Budhrani, D., Gomez-Barrero, M., & Busch, C. (2018). Detecting Morphed Face Images Using Facial Landmarks. In A. Mansouri, A. El Moataz, F. Nouboud, & D. Mammass (Eds.), *Image and Signal Processing* (pp. 444–452). Springer International Publishing. https://doi.org/10.1007/978-3-319-94211-7\_48

Sekuler, A. B., Gaspar, C. M., Gold, J. M., & Bennett, P. J. (2004). Inversion leads to quantitative, not qualitative, changes in face processing. *Current Biology: CB*, *14*(5), 391–396. https://doi.org/10.1016/j.cub.2004.02.028

Shen, J., & Palmeri, T. J. (2015). The perception of a face can be greater than the sum of its parts. *Psychonomic Bulletin & Review*, *22*(3), 710–716. https://doi.org/10.3758/s13423-014-0726-y

Smith, F. W., & Schyns, P. G. (2009). Smile through your fear and sadness: Transmitting and identifying facial expression signals over a range of viewing distances. *Psychological Science*, *20*(10), 1202–1208. https://doi.org/10.1111/j.1467-9280.2009.02427.x

Tottenham, N., Tanaka, J. W., Leon, A. C., McCarry, T., Nurse, M., Hare, T. A., Marcus, D. J., Westerlund, A., Casey, B., & Nelson, C. (2009). The NimStim set of facial expressions: Judgments from untrained research participants. *Psychiatry Research*, *168*(3), 242–249. https://doi.org/10.1016/j.psychres.2008.05.006

van der Walt, S., Schonberger, J. L., Nunez-Iglesias, J., Boulogne, F., Warner, J. D., Yager, N., Gouillart, E., & Yu, T. (2014). scikit-image: Image processing in Python. *PeerJ*, *2*, e453. https://doi.org/10.7717/peerj.453

Vinette, C., Gosselin, F., & Schyns, P. G. (2004). Spatio-temporal dynamics of face recognition in a flash: It’s in the eyes. *Cognitive Science*, *28*(2), 289–301. https://doi.org/10.1016/j.cogsci.2004.01.002

Webster, M. A., & De Valois, R. L. (1985). Relationship between spatial-frequency and orientation tuning of striate-cortex cells. *Journal of the Optical Society of America A*, *2*(7), 1124. https://doi.org/10.1364/JOSAA.2.001124

Wilson, H. R., Mcfarlane, D. K., & Phillips, G. C. (1983). Spatial frequency tuning of orientation selective units estimated by oblique masking. *Vision Research*, *23*(9), 873–882. https://doi.org/10.1016/0042-6989(83)90055-X

Xu, X., Biederman, I., & Shah, M. P. (2014). A neurocomputational account of the face configural effect. *Journal of Vision*, *14*(8), 9. https://doi.org/10.1167/14.8.9

Yang, N., Shafai, F., & Oruc, I. (2014). Size determines whether specialized expert processes are engaged for recognition of faces. *Journal of Vision*, *14*(8), 17. https://doi.org/10.1167/14.8.17

Yu, H., Garrod, O. G. B., & Schyns, P. G. (2012). Perception-driven facial expression synthesis. *Computers & Graphics*, *36*(3), 152–162. https://doi.org/10.1016/j.cag.2011.12.002

Yue, X., Biederman, I., Mangini, M. C., Malsburg, C. von der, & Amir, O. (2012). Predicting the psychophysical similarity of faces and non-face complex shapes by image-based measures. *Vision Research*, *55*, 41–46. https://doi.org/10.1016/j.visres.2011.12.012

Acknowledgements

The authors would like to thank Alex Burmester, Canhuang Luo, Martin Bo Nielsen, Wei Chen, and Ali Hashemi for their invaluable assistance with testing. We would also like to thank both Wei Chen and Eugenie Roudaia for bringing greater clarity to our manuscript. This work was funded by the National Natural Science Foundation of China (NSFC) to Carl M. Gaspar (grant No.: 31371132).