Controlling low-level image properties: The SHINE toolbox

Verena Willenbockel
Université de Montréal, Montréal, Québec, Canada

Javid Sadr
University of Massachusetts, Boston, Massachusetts

Daniel Fiset
Université de Montréal, Montréal, Québec, Canada

Greg O. Horne
University of Victoria, Victoria, British Columbia, Canada

Frédéric Gosselin
Université de Montréal, Montréal, Québec, Canada

and

James W. Tanaka
University of Victoria, Victoria, British Columbia, Canada

Visual perception can be influenced by top-down processes related to the observer's goals and expectations, as well as by bottom-up processes related to low-level stimulus attributes, such as luminance, contrast, and spatial frequency. When using different physical stimuli across psychological conditions, one faces the problem of disentangling the contributions of low- and high-level factors. Here, we make available the SHINE (spectrum, histogram, and intensity normalization and equalization) toolbox for MATLAB, which we have found useful for controlling a number of image properties separately or simultaneously. The toolbox features functions for specifying the (rotational average of the) Fourier amplitude spectra, for normalizing and scaling mean luminance and contrast, and for exact histogram specification optimized for perceptual visual quality. SHINE can thus be employed for parametrically modifying a number of image properties or for equating them across stimuli to minimize potential low-level confounds in studies on higher level processes.

Using identical stimuli across experimental conditions has the tremendous advantage that one does not have to worry about potential low-level confounds, such as differences in luminance, contrast, or spatial frequency, when studying higher level visual processes. For example, Tanaka and Curran (2001) presented the same set of dog and bird images to a group of dog experts and to a group of bird experts. The differences in brain activity observed when the experts categorized objects in their domain of expertise relative to when they categorized objects outside their domain could thus not be due to stimulus characteristics. In some studies on visual perception, however, it is impossible to use the same physical stimuli across psychological conditions. For instance, studies in which the domain-specificity versus domain-generality accounts of face processing are examined typically involve comparisons between faces and nonface objects, such as comparisons of

the inversion effect for faces and houses (e.g., Yin, 1969). When comparing effects across categories, one encounters the problem that there might be overall low-level differences between stimulus types or differences in the amount of within-category variation, which could potentially result in biases unrelated to the higher level processes meant to be studied. In general, when using different stimuli across conditions, one has to be careful to disentangle low-level and high-level factors (Fründ, Busch, Körner, Schadow, & Herrmann, 2007; Itier & Taylor, 2004; Luck, 2005; Rousselet, Macé, Thorpe, & Fabre-Thorpe, 2007; Sadr & Sinha, 2001, 2004; VanRullen & Thorpe, 2001).

Variations in low-level properties are thought to have contributed to controversies in the literature (e.g., Bentin et al., 2007; Dakin, Hess, Ledgeway, & Achtman, 2002; Hershler & Hochstein, 2006; Rousselet, Pernet, Bennett, & Sekuler, 2008; Thierry, Martin, Downing, & Pegna,

V. Willenbockel, verena.willenbockel@umontreal.ca

2007a, 2007b; VanRullen, 2006), and the importance of avoiding low-level confounds is an important and recurrent issue for a variety of research fields and measuring methods, such as eye tracking, functional magnetic resonance imaging (fMRI), magnetoencephalography (MEG), and electroencephalography (EEG). Inconsistent findings were obtained, for instance, about the earliest real event-related potential (ERP) differences between face and object perception. Some studies have been focused on the N170 component—a negative deflection occurring about 170 msec after stimulus onset—as the first marker of face processing (e.g., Carmel & Bentin, 2002; Rossion et al., 2000; Rousselet, Husk, Bennett, & Sekuler, 2008), whereas other studies revealed earlier differences between face and object processing, possibly as early as 50–80 msec after stimulus onset (e.g., George, Jemel, Fiori, & Renault, 1997; Seeck et al., 1997). Because the studies differ in the degree to which image properties across face and nonface categories were controlled, the different findings might, at least in part, be explained by differences in low-level influences (Rousselet, Husk, et al., 2008). Luck (2005) gave the general advice that one should "never assume that a small physical stimulus difference cannot explain an ERP effect" (p. 74).

In studies in which it is impossible to use identical stimuli across conditions, it can thus be important to match certain image properties across the stimulus set. Luminance is one property of interest, because early visual processes have been found to be sensitive to luminance variations, indicated by the modulation of early ERP components, such as the P1 (e.g., Johannes, Münte, Heinze, & Mangun, 1995). In some studies, this has been accounted for by equating stimuli in terms of mean luminance and contrast (for recent examples, see Finkbeiner & Palermo, 2009; Hardee, Thompson, & Puce, 2008; Liang, Zebrowitz, & Aharon, 2009; Zion-Golumbic, Golan, Anaki, & Bentin, 2008). It is possible to go a step further and precisely match the luminance histograms—which give the number of pixels at each luminance level—across images, thereby equating not only the means and standard deviations of the luminance distributions but also their shape. Exact histogram matching addresses the finding that certain neural mechanisms are sensitive to luminance histogram skewness (Olman, Boyaci, Fang, & Doerschner, 2008)—in particular, mechanisms involved in estimating surface properties (Motoyoshi, Nishida, Sharan, & Adelson, 2007) or texture discrimination (Chubb, Landy, & Econopouly, 2004).

Besides luminance, one might want to equate the images' spatial frequency content. Broadly speaking, low spatial frequencies represent the coarse information in an image, such as luminance blobs and blurred shapes, whereas high spatial frequencies carry the fine-grained information, such as the precise shape of an object. There is evidence that during early visual processing, the input is analyzed at multiple spatial frequencies by a number of channels, each tuned to a specific range (see De Valois & De Valois, 1990, for a review). Findings indicate that there are differences in sensitivity to specific spatial frequencies both between different visual areas and between the right

and left hemispheres (e.g., Ivry & Robertson, 1998). Variations in the spatial frequency domain could, for example, lead to differences in stimulus detectability (Campbell & Robson, 1968; Gold, Bennett, & Sekuler, 1999; Honey, Kirchner, & VanRullen, 2008). Carefully controlling luminance information and the energy at different spatial frequencies (i.e., Fourier amplitude spectra) should thus be of interest to a variety of studies designed to investigate high-level visual processes. However, numerous examples can be found in the literature where low-level properties remained uncontrolled, and to our knowledge, there is no standard program for systematically matching image properties across experimental stimuli.

Many researchers use MATLAB (The MathWorks, Natick, MA) for image preprocessing, running experiments, and data analysis. In recent years, the number of applications for MATLAB has grown enormously, and labs have shared their tools for experimentation to save time and reach a higher degree of standardization in the field. Besides several commercially available MATLAB toolboxes, such as the Image Processing Toolbox and the Signal Processing Toolbox, there are a number of freely available ones, such as the Psychophysics Toolbox (Brainard, 1997; Pelli, 1997), the Eyelink Toolbox (Cornelissen, Peters, & Palmer, 2002), EEGLAB (Delorme & Makeig, 2004), and the Biopsychology Toolbox (Rose, Otto, & Dittrich, 2008).

Here, we make accessible another MATLAB toolbox that we have found useful for controlling a number of low-level image properties in studies on visual perception. Specifically, our SHINE (spectrum, histogram, and intensity normalization and equalization) toolbox, which was written using functions from the Image Processing Toolbox of MATLAB, includes different equalization approaches that can be applied together or separately, depending on the requirements of the experiment. SHINE features functions for specifying the Fourier amplitude spectra of images or scaling the rotational average of the spectra only (i.e., the energy at each spatial frequency averaged across orientations). It also includes functions for exactly specifying the luminance histograms or for normalizing and scaling the means and standard deviations of the luminance distributions without affecting their shape. The program offers ways to equate the luminance properties across stimuli separately for the foregrounds and the backgrounds and to preserve perceptual image quality as much as possible. Although histogram specification and Fourier amplitude specification affect each other, we discovered that by using an iterative approach, a high degree of joint matching of the low-level properties of interest could be reached in many cases. SHINE also features tools for plotting the Fourier amplitude spectra or the average energy at each spatial frequency to verify the output or to monitor the ecological low-level variations.

In the following sections, we provide an overview of the individual SHINE functions and give examples regarding how to work with the toolbox, which can be downloaded at www.mapageweb.umontreal.ca/gosselif/shine. The

toolbox has successfully been used by us, as well as by others (e.g., Fiset, Blais, Gosselin, Bub, & Tanaka, 2008; Williams, Willenbockel, & Gauthier, 2009), in studies on visual perception to equate (or parametrically vary) low-level image properties across experimental stimuli. Moreover, prior to establishing the standard version of SHINE presented here, in a number of other behavioral and neuroscientific studies earlier or partial forms of these preprocessing steps were successfully used to achieve strict stimulus control (e.g., Adolphs et al., 2005; Liu, Harris, & Kanwisher, 2002; Loschky et al., 2007; Mack, Gauthier, Sadr, & Palmeri, 2008; Sadr & Sinha, 2001, 2004; Xu, Liu, & Kanwisher, 2005).

METHOD

The main m-file is *SHINE*, which calls the functions for the individual adjustment steps, such as histogram matching or Fourier amplitude matching. In *SHINE*, one can specify the parameters—for instance, the type of matching desired, the number of iterations, and whether to perform the luminance adjustment on the whole image or separately for selective regions, such as foreground and background. The individual functions called by *SHINE* are described in the following sections, and Figure 1 provides an overview of the structure of the toolbox.

Normalizing and Scaling Mean Luminance and Contrast

In some cases, it might be desirable to equate the images in terms of mean luminance and contrast (i.e., the standard deviation of the luminance distribution) only, instead of specifying the exact histogram shape and thereby equating the means and standard deviations automatically. The former typically results in higher image quality because no noise is introduced, as it is in most cases of exact histogram matching. The SHINE function *lumMatch* can be used for a simple normalization:

$$Z = \frac{X - m}{s},$$

where X is the input image matrix containing the original grayscale values—that is, $X = \lfloor 255 \ (L - L_{min})/(L_{max} - L_{min}) \rfloor$, where L is the luminance matrix as measured on the computer monitor, and L_{min} is the minimum and L_{max} the maximum luminance that can be displayed on the monitor; see the SHINE and Monitor Calibration section, m is the mean of the original values, and s is the standard deviation. For simplicity, we will refer to the grayscale values contained in X as *luminance values* throughout the article. The desired mean (M) and desired standard deviation (S) can then be applied to obtain the output image matrix E containing the scaled luminance values:

$$E = ZS + M.$$

The default values for M and S are obtained by averaging the means of all input stimuli and the standard deviations of all input stimuli, respectively.

As with exact histogram matching (see the Luminance Histogram Matching section), there are cases in which normalization on the whole image will give the best results and cases in which it might be advantageous to consider the foreground and the background separately. Using the function *separate*, SHINE transforms the input image into a binary template from which it extracts which pixels belong to the foreground and which belong to the background. A simple example illustrates why separate equalization might be important: Assume one wanted to equate a stimulus consisting of a relatively bright object on a midgray uniform background with a stimulus depicting a relatively dark object on the same midgray background. If the luminance adjustment was applied to the whole image, the first stimulus would end up with a darker background
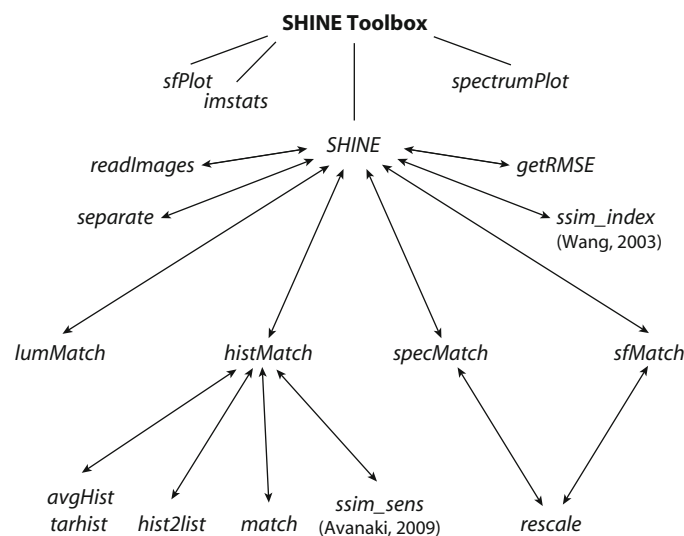

Figure 1. Overview of SHINE toolbox functions and their calling structure. For example, the *SHINE* function calls the *specMatch* function, which calls the *rescale* function.
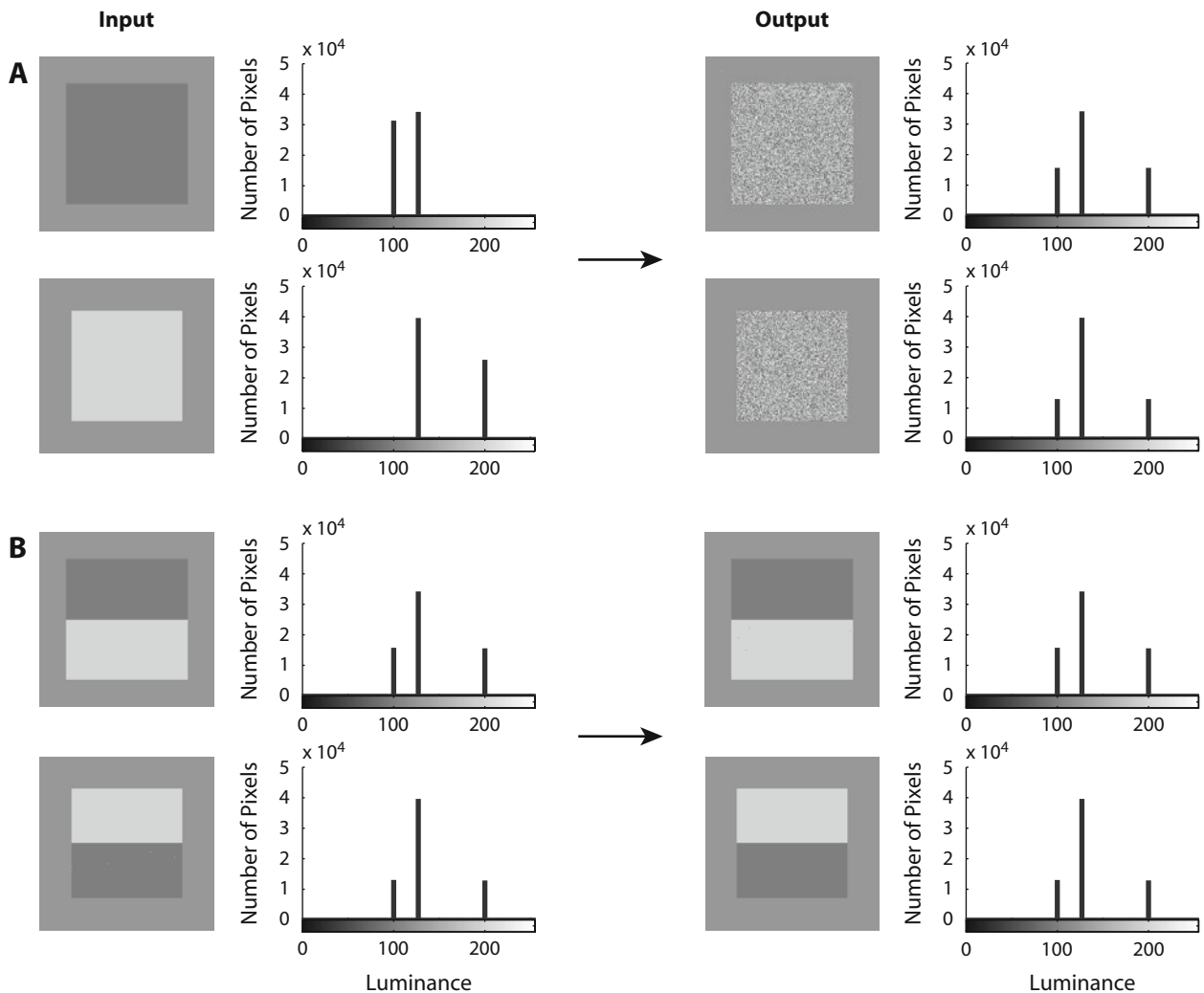
Figure 2. Illustration of the basic luminance histogram-matching approach using simple patterns. The histograms show luminance (in arbitrary linear units [ALU]) on the *x*-axis and the number of pixels on the *y*-axis. The target histogram was computed by averaging the histograms of the two input images. (A) Histogram matching was performed on two uniform surfaces (100 and 200 ALU) of slightly different sizes centered on a uniform gray background (127 ALU). The output surfaces contain 50% 100-ALU pixels and 50% 200-ALU pixels in randomized order. The background remained unaltered. (B) Two surfaces of the same size as in panel A with 50% 100-ALU and 50% 200-ALU pixels served as input. The output is identical to the input.
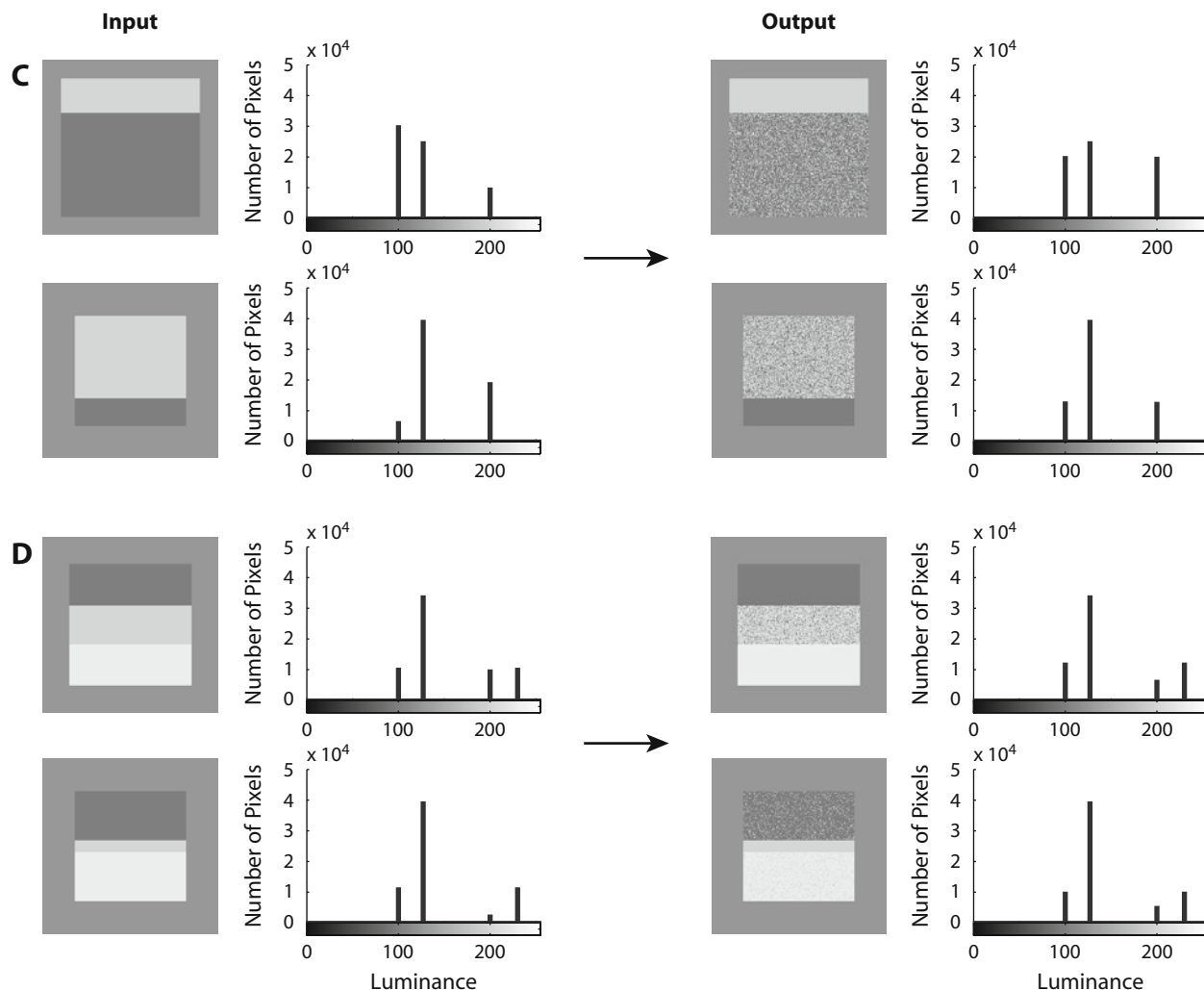
than the second image, so that the background itself would contain diagnostic information. In some cases, applying the matching step separately to selective regions might therefore be preferable.

Luminance Histogram Matching

The SHINE toolbox includes the function *histMatch*, which exactly matches the luminance histograms of a number of source images with a specified target histogram. Specifically, it calls the function *match*, which transforms one luminance distribution into another by remapping the pixel values to control how frequently they occur relative to others. The average luminance distribution of the input set, which is computed using *avgHist* and then transformed into a sorted list of luminance values using *hist2list*, serves as the default target (Figures 2A–2D). Alternatively, it is possible to provide the *match* function

with another target—for example, by using the function *tarhist*, which sorts the luminance values of the input images in ascending order and then averages the values across images (e.g., the darkest values across all images, the second darkest, etc.) to obtain the target (Figures 3A and 3B).

In general, two strategies can be applied for histogram matching: One can exactly match the histograms, or one can approximately match them by using the best suboptimal pointwise transformation of the luminance values. The former is what SHINE does (the built-in MATLAB function *histeq* does the latter). Even though exact histogram matching comes at the expense of increased noise in the image, we observed that in many cases image quality is still very high after applying SHINE (see Figures 3A–3C for examples and the Joint Matching of Certain Low-Level Properties section for image quality measures). Several

Figure 2, continued. Illustration of the basic luminance histogram-matching approach using simple patterns. The histograms show luminance (in arbitrary linear units [ALU]) on the *x*-axis and the number of pixels on the *y*-axis. The target histogram was computed by averaging the histograms of the two input images. **(C)** Illustration with surfaces similar to panel B but with different amounts of dark and light pixels. **(D)** Illustration with input surfaces of three different luminance levels (100, 200, and 230 ALU) with different numbers of pixels of each.

ways of exact histogram matching have been proposed and evaluated in terms of visual quality of the result and computational complexity (e.g., Avanaki, 2009; Bevilacqua & Azzari, 2007; Coltuc, Bolon, & Chassery, 2006; Morovic, Shaw, & Sun, 2002; Rolland, Vo, Bloss, & Abbey, 2000).

SHINE follows the exact global histogram-matching approach described in Table 1 (see also Avanaki, 2009; Rolland et al., 2000). Basically, the pixels of the source image and the pixels of the target distribution are sorted separately by their luminance value from darkest to lightest. The source pixel with the smallest value is then assigned the smallest value of the target, the source pixel with the second smallest value is assigned the second smallest value of the target, and so forth. Ambiguity arises when a number of source pixels with the same luminance value have to be broken down into two or more

Table 1
Algorithm for Exact Global Histogram Specification

Step 0	Let the target histogram be $H = \{h_0, h_1, \ldots, h_{D-1}\}$, where D is the number of possible luminance values (e.g., 256 in an 8-bit image). It is assumed that $\sum_{i=0}^{D-1} h_i = N$, where N equals the number of pixels in the image. If this assumption is not met, scale H (and perhaps round some h_i) to satisfy this assumption.
Step 1	Sort the pixels of the source image by their luminance value in ascending order (and in the cases in which the values are the same, randomize the pixels).
Step 2	Starting from the first pixel on the sorted list, assign the first h_0 pixels a new value of 0. Continue by assigning the next h_1 pixels a new value of 1, and so on until all pixels are allotted their new values.

groups to match the target: It has to be decided which pixels get new values. For example, this would be the case if the two darkest pixels of the source image had the same luminance value (e.g., 0), but the target histogram contained only one pixel with a value of 0. So which of the two source pixels should be assigned a new value? The *match* function of SHINE uses random assignment (as was applied in Williams et al., 2009), but different approaches have been suggested (Coltuc et al., 2006; Wan & Shi, 2007). In all cases, the histogram-matching step produces a set of images that are made up of the same pixels, only with a different arrangement (Figures 3A and 3B). In this way, mean luminance, contrast, and all other characteristics of the histogram (e.g., skew) are equated across stimuli.

In order to generate output images that not only are exactly equated in histograms, but also retain the structure of the original image as much as possible, the *histMatch* function includes the method developed by Avanaki (2009): It allows for iteratively optimizing the structural similarity (SSIM) index (Wang, Bovik, Sheikh, & Simoncelli, 2004; see the Joint Matching of Certain Low-Level Properties section for more details on the SSIM index) between the original and histogram-matched images. In each iteration, the *match* function is applied to obtain an image with the specified target histogram. The visual quality of the histogram-matched image is measured using the SSIM index, and an SSIM gradient with respect to that image is computed using the *ssim_sens* function.[1] The gradient is then employed to adaptively increase the SSIM of the output (see Avanaki, 2009, for details). The algorithm is summarized in Table 2. Using SSIM gradient ascent, the perceptual image quality can be improved considerably relative to basic histogram matching alone, whereas the histogram remains exactly as specified. This benefit, however, comes at the cost of increased runtime, which might make the SSIM optimization option unsuitable for the matching of very large image sets.

SHINE also includes the option for applying the histogram matching selectively to subsets of pixels—for example, separately to the foreground and the background. This can improve output quality in some cases, because it prevents object pixels from ending up in the background and vice versa (e.g., one might want to choose this option for stimuli displaying a single object on a uniform or noisy background; also see the Specifying Fourier Amplitude Spectra section).

Specifying Fourier Amplitude Spectra

Using Fourier analysis, any complex two-dimensional image can be broken down into the sum of a set of sinusoidal gratings defined by four parameters: spatial frequency, orientation, amplitude, and phase. The spatial frequency refers to the width of the gratings' bars and can be specified as the number of light:dark cycles per image. *Orientation* refers to the angle of the light and dark bars, specified in degrees counterclockwise from vertical; *amplitude* is given by the difference in luminance between the lightest and the darkest parts of the grating; and *phase* refers to the position of the sinusoid relative to some reference point.

After performing a Fourier transform on an image (e.g., with the built-in MATLAB function *fft2*), one can obtain two components: the amplitude spectrum and the phase spectrum.[2] The amplitude spectrum specifies the amplitude of each constituent grating at a particular spatial frequency and orientation. The phase spectrum specifies the phase of each grating at a particular spatial frequency and orientation. If all of the gratings at the corresponding phases and amplitudes were summed, they would exactly result in the original image.

SHINE includes the function *specMatch*, which matches the amplitude spectrum of the source image with a specified target spectrum. Each source image is submitted to a fast Fourier transform (FFT), and the spectrum is shifted so that the low spatial frequencies occupy the central region. If not specified otherwise, the average spectrum is obtained across all input images and serves as the default target. The phase of the original image is then combined with the target amplitude spectrum and back-transformed from the frequency into the spatial domain using an inverse fast Fourier transform (IFFT).

This approach matches the source spectrum exactly with the target spectrum at each spatial frequency and orientation (Figure 4C). The output images typically look a bit cloudy, and their visual quality depends on the similarity of the input images; for example, equating the spectra of a number of faces will likely yield better results than equating the spectra across different categories, such as

---

Table 2
SSIM Optimization Algorithm of Avanaki (2009)

Step 0	Let X show the original input image. Set $X_{new} = X$.
Step 1	Apply the *match* function to X_{new} to obtain image Y with given histogram H.
Step 2	Compute $\nabla_y \text{SSIM}(X,Y)$ and $\text{SSIM}(X,Y)$ using the *ssim_sens* function.
Step 3	If convergence is reached, then break.
Step 4	Set $X_{new} = Y + \mu N \nabla_y \text{SSIM}(X,Y)$ and go to Step 1.
Step 5	Output Y.

Note—μ is a positive constant denoting the step size, ∇_y represents gradient with respect to image Y, and N is the number of pixels in X. $\text{SSIM}(X,Y)$ is a measure of perceptual similarity of X and Y. The convergence criterion in Step 3 may be given by desired output quality [i.e., $\text{SSIM}(X,Y) >$ threshold] or by a certain number of iterations (or a combination thereof). The updated X_{new} in Step 4 is more similar to X, because each pixel of Y was altered so that the SSIM was increased. The histogram of the output image remains exactly as specified.
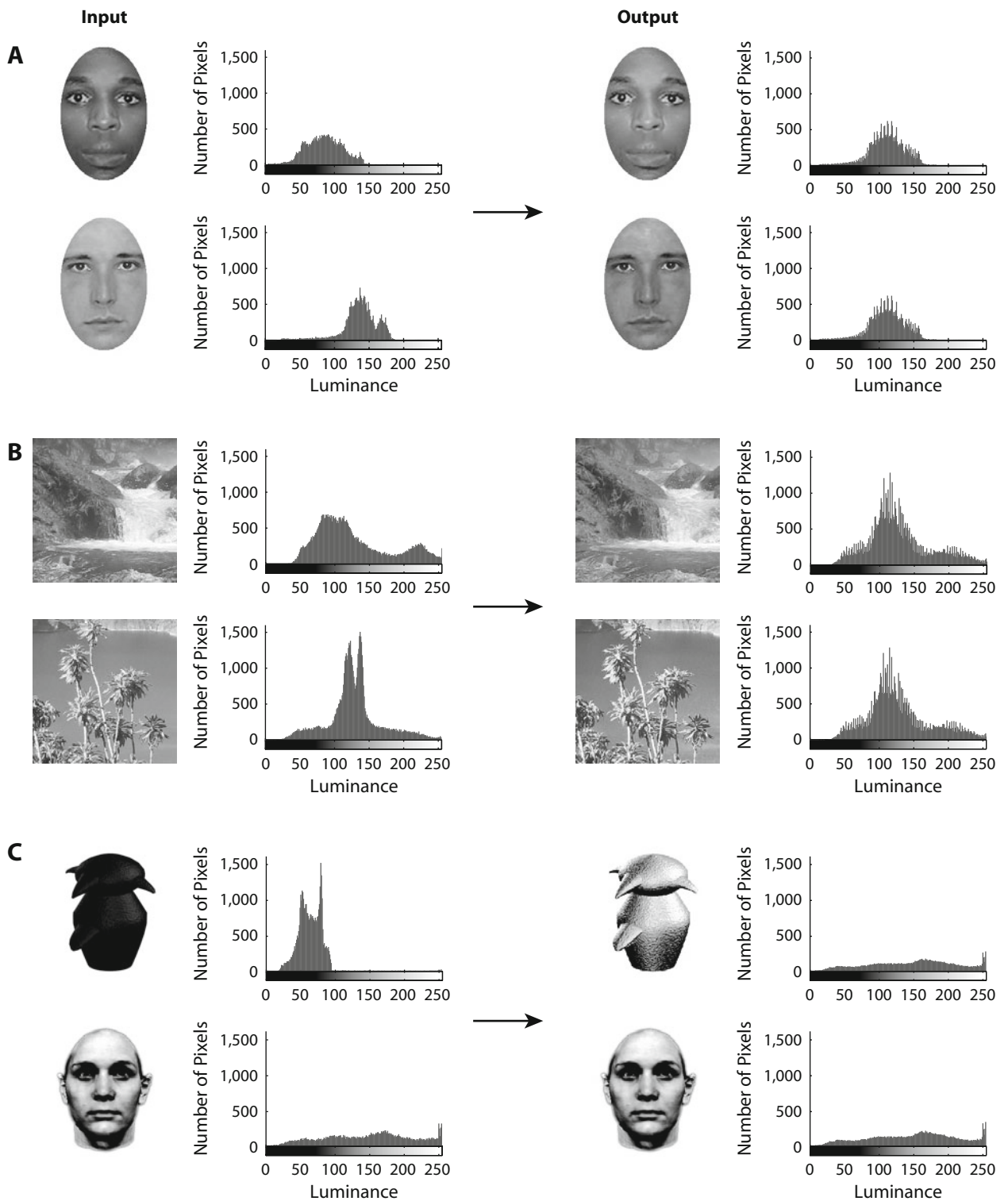
Figure 3. Illustration of the *match* function of SHINE. (A) Two base face images with their luminance histograms (left) and the corresponding SHINE output images with their matched histograms (right). The target histogram was obtained using the function *tarhist*. (B) Histogram matching was performed on two natural scenes. As in panel A, the function *tarhist* was used to obtain the target histogram. (C) Histogram specification is illustrated for a face and a greeble. Here, the target was computed independently of the two input images by averaging the histograms of several faces (not shown) using *avgHist*. As a result, the greeble was altered more than the face when applying the target.

A   Input

B   Output (*sfMatch*)
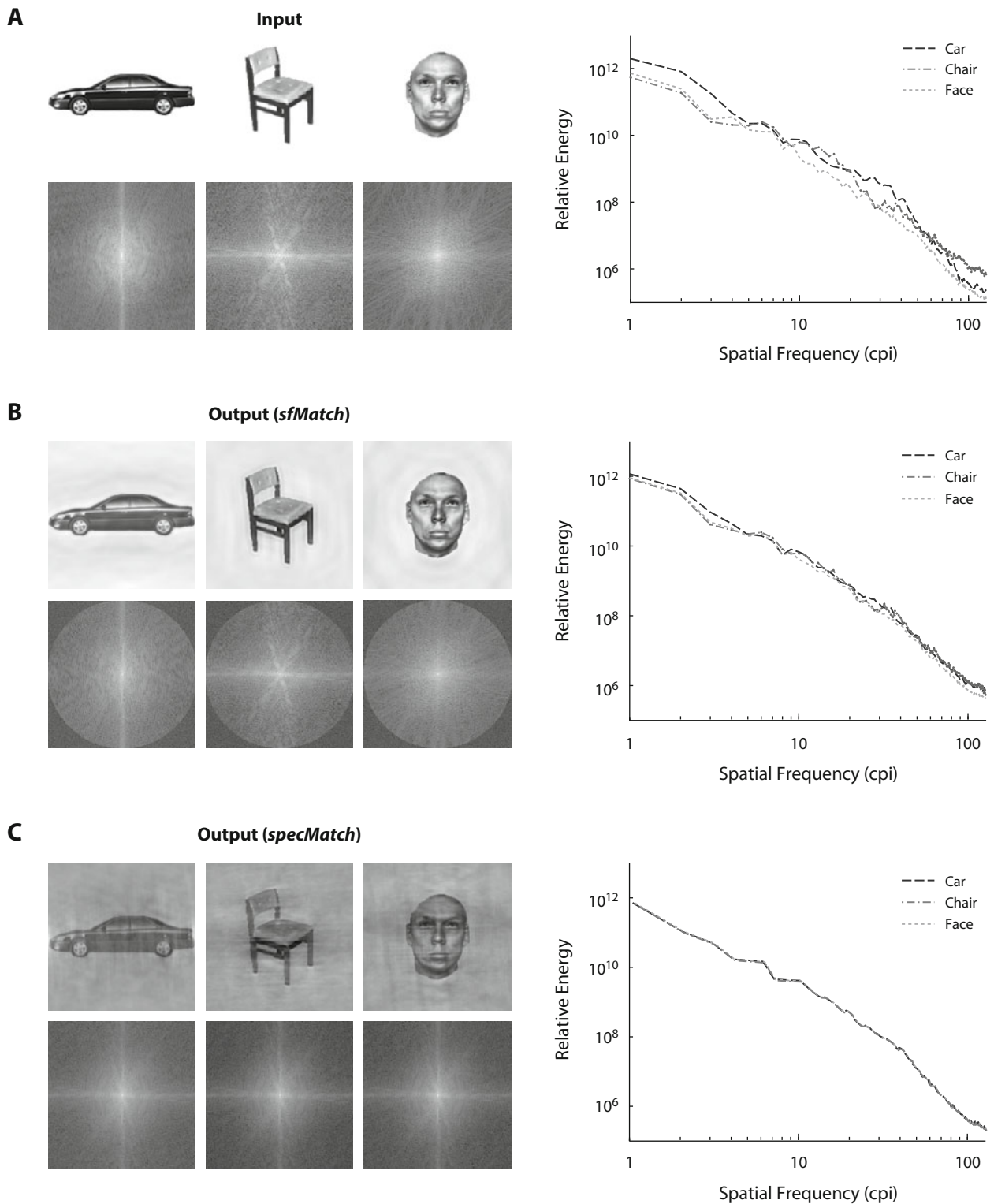

C   Output (*specMatch*)

Figure 4. Illustration of the *sfMatch* and *specMatch* functions of SHINE. (A) Three source images (a car, a chair, and a face) are shown with their Fourier spectra displayed as polar plots, where energy is plotted as a function of spatial frequency (distance from the origin) and orientation (angle). The log–log plot on the right depicts the rotational average of the spectra (i.e., the energy at each spatial frequency, in cycles per image, cpi, averaged across orientations). (B) Using *sfMatch*, the rotational average of the Fourier spectra was equated while the energy distribution across orientations was preserved (see the text for details). (C) Using *specMatch*, the Fourier spectra were equated on spatial frequencies and orientations. The output in panels B and C is shown after the rescaling of the luminance values so that absolutely all pixels of the three images are in the range of 0–255.

faces and cars. If image quality is a concern, we propose to use the *sfMatch* function described below.

Specifying the Rotational Average Amplitude at Each Spatial Frequency

The function *sfMatch* applies a more lenient approach for equating the Fourier amplitudes across stimuli. Unlike *specMatch*, it preserves the amplitude distribution across orientations while ensuring that the rotational average amplitudes for a given spatial frequency are equated between images. The initial steps are the same as those described above for *specMatch*: The source images are subjected to an FFT, and the output is shifted so that the low spatial frequencies are in the center of the spectrum. If not specified otherwise, the target spectrum is obtained by averaging across input spectra. A coefficient is then computed for each spatial frequency: The amplitudes across orientations at the given spatial frequency are summed, separately for the source and the target spectrum. For each spatial frequency, the resulting target sum is divided by the corresponding source sum to obtain a coefficient. Afterward, the amplitude at each spatial frequency and orientation is multiplied with the respective coefficient. The phase of the original image is combined with the modified amplitude spectrum and back-transformed into the spatial domain using an IFFT. As a result, the rotational average of the amplitude spectrum is equated between images (besides slight rounding errors that may arise), whereas the distribution of amplitudes among orientations is preserved. Although this controls overall amplitude, it often results in higher similarity of the output image with the source image than would strictly equating the amplitudes at each spatial frequency and each orientation (Figures 4A–4C).

Rescaling of Luminance Values After the IFFT

It is possible that after the Fourier amplitudes are equated and the IFFT is applied, the luminance values of the resulting images are shifted out of the desired range or out of the range that can be displayed—for example, below 0 or above 255 for 8-bit images. In this case, the negative values and the ones larger than 255 will be clipped to 0 and 255, respectively. This results in a change of the actual final luminance and contrast but usually only a small change in the amplitude spectrum. One might want to rescale the luminance values after Fourier amplitude matching in such a way that all or at least the majority of the values fall back into the range of 0–255 (Sadr & Sinha, 2001, 2004). This is not done separately for each image, because individual rescaling would result in images that no longer match the properties that one set out to normalize in the first place. The *rescale* function of SHINE therefore first obtains the full range of luminance values coming out of the IFFT, for all of the images, and then computes one set of rescaling values to apply to each image. Specifically, after going through all of the images and calculating the lowest and highest luminance values, there are two options: It is possible to rescale all images with the same linear scaling parameters so that absolutely all pixel values for all images are in the range of 0–255 or, on average, the smallest and largest pixel values are rescaled to 0 and

255, but some clipping is still allowed to occur in the final product. The second alternative gives a slightly imperfect result, which is why the default SHINE is set to the first alternative. However, the second approach squashes the images' luminance less than the first one and so might give nicer-looking images with a little more contrast.

Joint Matching of Certain Low-Level Properties

Theoretically, one could perfectly match all low-level properties across an image set; however, then the images would be identical. One challenge is to equate certain properties while preserving others. SHINE attempts to preserve the phase information in the image while matching the luminance histograms and Fourier amplitudes. This is associated with at least two problems: First, phase can also be affected by histogram matching (through the introduction of noise), and the visual quality of the image is thus decreased; second, histogram matching and Fourier amplitude matching affect each other. That is, Fourier amplitude matching performed after histogram matching will distort the histograms again to some extent and vice versa.

To measure the decrease in visual quality, we added two functions for image quality assessment to SHINE. The *getRMSE* function computes the root-mean square error (RMSE), which is a simple and widely used metric for quantifying the visibility of differences between a distorted and a reference image. Because it is not always well matched to perceived visual quality, we also included the SSIM index (Wang et al., 2004) computed by Wang, Bovik, Sheikh, and Simoncelli's (2003) *ssim_index* function. The SSIM index has been developed for quality assessment based on the degradation of structural information. If the input images are identical, the SSIM index is 1, and if they are uncorrelated, it is very small. Although it will be impossible to completely avoid alterations to the visual quality, these two quality assessment functions provide means for comparing the different matching options objectively and could be helpful for finding the appropriate low-level equalization procedure (or parameters).

In order to address the problem of histogram and Fourier amplitude specification affecting each other, we developed and implemented an iterative approach: The histogram and Fourier amplitude-matching steps can be performed a number of times, whereby the respective target is recalculated at each iteration. Using this iterative strategy, we were able to reach a high degree of joint matching in previous studies using SHINE (e.g., Williams et al., 2009). Figures 5–6C illustrate the increasing degree of joint equalization over iterations for a few typical image sets. Specifically, Figure 5 shows one car, one chair, and one face stimulus from the set used by Williams et al. with the corresponding histograms after 0, 1, and 3 iterations. One iteration consisted of *histMatch* followed by *sfMatch* (with default rescaling of the luminance values). It can be seen that the histograms gradually became more similar to each other.

In Figures 6A–6C, we used the same face and car images as in Figure 5 (Set 1) and the scenes depicted in Figure 3B (Set 2). We ran 20 iterations separately on the two sets, whereby one iteration consisted of *histMatch* followed by *specMatch* (with default rescaling) or vice

Input

Output (Iteration 1)
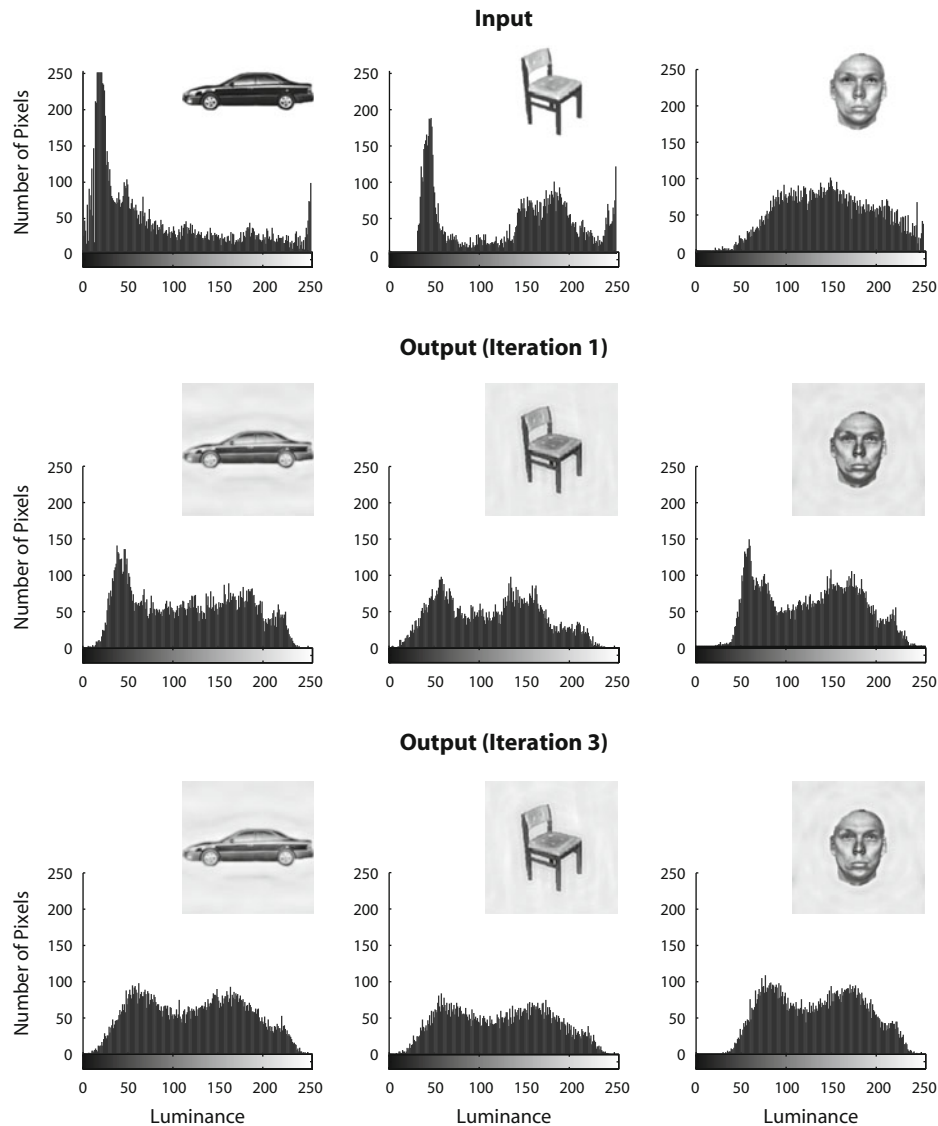

Output (Iteration 3)

Figure 5. Iterative histogram and Fourier amplitude matching. The histograms of three sample images (a car, a chair, and a face) are shown before applying SHINE (top), after one iteration (middle), and after three iterations (bottom). Each iteration consisted of equating the histograms using *histMatch* (with the average histogram serving as the target) and then equating the rotational average of the Fourier spectra using *sfMatch*. The latter step altered the histograms; however, after a number of iterations, the histograms typically converge toward the target.

versa. The histogram matching was applied separately to the foreground and the background for the images in Set 1 and to the whole image for those in Set 2. To trace the increasing degree of joint equalization of amplitude spectra and luminance histograms, we computed the RMSE between the amplitude spectra of the two images in each set for each of the iterations, and similarly, we computed the RMSE between the normalized histograms of each image pair. As a comparison, for the original images, the RMSE between the amplitude spectra was 30.7394 for Set 1 and 21.3955 for Set 2; the RMSE between the normalized histograms was 0.0069 for Set 1 and 0.0046 for Set 2. As an image quality measure, we computed the mean SSIM index between the original image and the SHINE output

for each set at the respective iteration. The results show that the individual matching steps worked precisely, because the property that was matched last (e.g., amplitude spectra for *histMatch* followed by *specMatch*) was nearly perfectly equated for all iterations. The property that was matched first was altered less as the number of iterations increased, thus converging toward 0. Clearly, the greatest improvements in joint equalization were reached within the first four iterations, independently of equalization order. The SSIM index, which initially equaled 1, remained fairly high and constant through Iterations 1–20. For Set 1, it equaled .76 at Iteration 1 and .74 at Iteration 20, when *histMatch* was performed first. When *specMatch* was applied first, it equaled .80 and .78 for Itera-
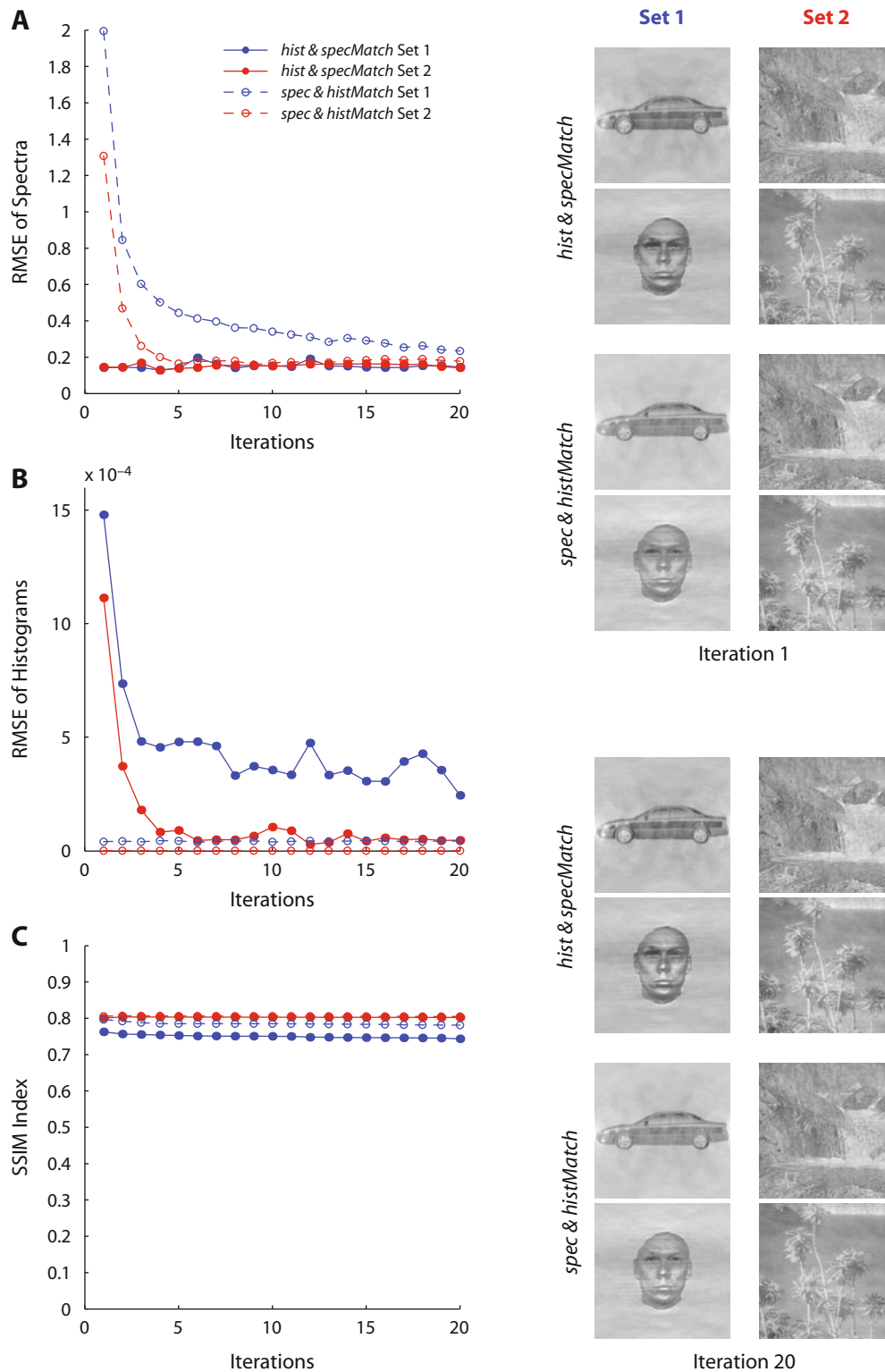
Figure 6. The iterative equalization approach illustrated for two typical image pairs (Sets 1 and 2). Left: (A) For each set, the root-mean square error (RMSE) between the two images' Fourier spectra is plotted over Iterations 1–20. The dotted line shows the results for Fourier amplitude spectrum matching (*spec-Match*) followed by histogram matching (*histMatch*) and the full line for *histMatch* followed by *specMatch*. (B) Analogous to (A), the RMSE between histograms is shown for each image pair over iterations. (C) The average structural similarity (SSIM) index between the original images and the corresponding SHINE output is plotted for each set over Iterations 1–20. Right: The output images of Set 1 (left column) and Set 2 (right column) are shown after 1 and after 20 iterations of *histMatch* followed by *specMatch* or vice versa.

tions 1 and 20, respectively. Similarly, for Set 2, the SSIM index was .8 (Iterations 1–20) when the histograms were equated first and remained at .81 for all iterations when the spectra were matched first. Depending on the input set and the desired degree of matching, one might want to choose different numbers of iterations and might want to apply the steps in a different order.

Plotting Functions and Image Statistics

The Image Processing Toolbox of MATLAB features a function for plotting the luminance histogram of a given image (the *imhist* function). However, it does not provide functions for directly plotting the rotational average of the Fourier amplitude spectrum or the amplitude spectrum itself. SHINE includes these two plotting functions (*sfPlot* and *spectrumPlot*, respectively), which can be useful for checking the SHINE output, or for assessing low-level variations in the source images. Moreover, SHINE features a function for computing a number of image statistics across an image set (the *imstats* function).

Applying SHINE to Color Images

SHINE was originally designed for the preprocessing of grayscale luminance images, but there are ways to apply it to color images as well. In particular, the luminance-matching functions can be directly applied to color images if the images are in a color space that allows for separating luminance as one dimension, as does the HSL color space. One would then equate the histograms—or just the mean and standard deviation of the luminance distribution—of the respective layer without altering the hues and saturation of the image.

For equating the Fourier amplitudes, one could use the RGB color space and do a separate matching of amplitude spectra for each of the color layers. The basic matching step would thus be analogous to how it is described for one-layer grayscale images but would have to be applied three times for each RGB color image.

SHINE and Monitor Calibration

The matching functions of SHINE (except *histMatch*) assume that the relationship between the stored luminance values that make up the image matrix X and the luminance intensity produced at the face of the screen is linear. However, on a typical computer–monitor system, this is not the case. When using LCDs, one can correct for the nonlinearity by setting the gamma parameter to 1. When using a CRT monitor, however, this will not suffice. To meet the linearity assumption of SHINE, it is important in such a case to perform gamma correction after equating the low-level properties of a set of images expressed as Xs (see the Normalizing and Scaling Mean Luminance and Contrast section). For example, the inverse power function converting the luminance intensity (L) into RGB values (V)—

$$V \approx (L - k)^{\frac{1}{\text{gamma}}},$$

where gamma is a positive constant usually close to 2.5 and k is a constant that corresponds to the minimum luminance of the monitor—can be applied to correct for the nonlinearity. Gamma correction is a common practice and can be done in various ways (e.g., Metha, Vingrys, & Badcock, 1993; Pelli & Zhang, 1991; Poynton, 1993; Stanislaw & Olzak, 1990).

SUMMARY
The Default SHINE

The default version of SHINE first exactly equates the Fourier amplitude spectra, whereby the average spectrum obtained across all input images serves as the target. Rescaling ensures that the luminance values stay in the desired range and are not clipped. Afterward, SHINE equates the luminance histograms across all input images, so that each output image has the same luminance distribution as the average of the input sets. Fourier amplitude and histogram matching are performed iteratively a number of times specified by the researcher and depending on the requirements of the experiment to obtain a high degree of joint matching of Fourier amplitudes and luminance properties. Demos of the main program and individual functions are available online at www.mapageweb.umontreal.ca/gosselif/shine.

DISCUSSION

The SHINE toolbox contains a number of MATLAB functions for controlling low-level image properties, such as luminance, contrast, and spatial frequency. Specifically, it can be used for specifying the (rotational average of the) Fourier amplitude spectra, for exact luminance histogram specification, and for normalizing and scaling mean luminance and contrast. SHINE offers ways to apply the luminance adjustments selectively to a subset of pixels (e.g., separately to the foreground and the background), and it includes the option to perform the histogram and Fourier amplitude specification iteratively a number of times to reach a high degree of joint matching of luminance and spatial frequency properties between source and target.

The main motivation behind SHINE was to provide tools that can easily be applied (even by novice MATLAB users) for equating a number of low-level properties across a stimulus set in order to minimize low-level confounds in studies on higher level processing. It has been shown that early vision is sensitive to variations in luminance and spatial frequency content (see De Valois & De Valois, 1990, for a review), and several researchers have recently emphasized the importance of disentangling low- and high-level factors when using different physical stimuli across psychological conditions (e.g., Luck, 2005; Rousselet, Husk et al., 2008; Sadr & Sinha, 2001, 2004). Controlled image normalization is an important issue in various research fields, including psychophysics and visual search, and for different brain-imaging techniques, such as fMRI, MEG, and EEG. SHINE, which allows for controlling both luminance and spatial frequency characteristics of images with great precision, should thus be useful in a variety of experiments in order to minimize low-level confounds.

To our knowledge, there is no other commonly available program or toolbox for directly equating low-level image properties across a set of images. Photo-editing software, such as Photoshop, allows for individual luminance adjustments, but it will hardly be possible to use it for exact histogram matching of a large number of complex images. Furthermore, photo-editing programs typically do not provide tools for matching images in the Fourier domain. The Image Processing Toolbox of MATLAB, which SHINE is based on, contains the function *histeq* for histogram equalization; however, it serves the purpose of contrast enhancement and does not perform exact histogram matching, as does the *match* function made available here. Furthermore, SHINE extends the toolbox by including the implementation of a state-of-the-art method to optimize the structural similarity between original and histogram-matched images (Avanaki, 2009), as well as tools for the direct manipulation and visualization of the Fourier amplitude spectra and the average energy at each spatial frequency.

Knebel, Toepel, Hudry, Le Coutre, and Murray (2008) proposed a different approach for generating controlled stimulus sets, which one might want to apply in combination with SHINE. Specifically, they introduced a method for spectral distance optimization, which does not alter physical image properties but, rather, identifies the subgroups of images that are spectrally most similar. For example, given 100 images of Category A and 100 images of Category B, their method could be applied to select 40 images per category so that the spectral differences between the subsets are minimized. Functions from the SHINE toolbox could then be used to match the selected images exactly; one could apply SHINE to all 80 images at once or to a smaller group (e.g., 10 images per category) at a time. As a result, the low-level properties would be equated across categories, whereas the alterations introduced by the exact-matching steps of SHINE would be minimized by choosing the most similar images in the first place.

Functions from the SHINE toolbox have recently been applied in a study comparing the spatial frequency sensitivity of face and object processing (Williams et al., 2009). The rotational average of the Fourier amplitude spectra and the luminance histograms were matched across a total of 45 images from three categories (faces, cars, and chairs; Experiment 1). SHINE was also applied in a study on race categorization, where 100 Caucasian, 100 Asian, and 100 African American faces were matched in terms of histograms and energy at each spatial frequency (Fiset et al., 2008). Furthermore, SHINE has been employed for parametrically varying the luminance properties in a study on the relative influence of luminance and featural information in race categorization (Willenbockel, Fiset, & Tanaka, 2008). Specifically, using the histogram specification function of SHINE, five luminance levels were created, ranging from the original distribution of an African American face over the average distribution to the original distribution of a Caucasian face. All of the levels were then applied to both the Caucasian and the African American faces. Likewise, a number of other researchers have used earlier or partial forms of these processing steps to create well-controlled image sets (e.g., Adolphs et al., 2005; Liu et al., 2002; Loschky et al., 2007; Mack et al., 2008; Sadr & Sinha, 2001, 2004; Willenbockel et al., 2010; Xu et al., 2005).

The SHINE approach has thus been successfully used for both equating and parametrically varying low-level properties, whereas the structure contained in the images was largely preserved. We hope that other laboratories will find SHINE as useful as we have for minimizing potential low-level confounds in studies on higher level processes.

REFERENCES

ADOLPHS, R., GOSSELIN, F., BUCHANAN, T. W., TRANEL, D., SCHYNS, P. G., & DAMASIO, A. R. (2005). A mechanism for impaired fear recognition after amygdala damage. *Nature*, **433**, 68-72.

AVANAKI, A. N. (2009). Exact global histogram specification optimized for structural similarity. *Optical Review*, **16**, 613-621.

BENTIN, S., TAYLOR, M. J., ROUSSELET, G. A., ITIER, R. J., CALDARA, R., SCHYNS, P. G., ET AL. (2007). Controlling interstimulus perceptual variance does not abolish N170 face sensitivity. *Nature Neuroscience*, **10**, 801-802.

BEVILACQUA, A., & AZZARI, P. (2007). A high performance exact histogram specification algorithm. In R. Cucchiara (Ed.), *14th International Conference on Image Analysis and Processing (ICIAP '07)* (pp. 623-628). Los Alamitos, CA: IEEE Computer Society Press.

BRAINARD, D. H. (1997). The Psychophysics Toolbox. *Spatial Vision*, **10**, 433-436.

CAMPBELL, F. W., & ROBSON, J. G. (1968). Application of Fourier analysis to the visibility of gratings. *Journal of Physiology*, **197**, 551-566.

CARMEL, D., & BENTIN, S. (2002). Domain specificity versus expertise: Factors influencing distinct processing of faces. *Cognition*, **83**, 1-29.

CHUBB, C., LANDY, M. S., & ECONOPOULY, J. (2004). A visual mechanism tuned to black. *Vision Research*, **44**, 3223-3232.

COLTUC, D., BOLON, P., & CHASSERY, J.-M. (2006). Exact histogram specification. *IEEE Transactions on Image Processing*, **15**, 1143-1152.

CORNELISSEN, F. W., PETERS, E. M., & PALMER, J. (2002). The Eyelink Toolbox: Eye tracking with MATLAB and the Psychophysics Toolbox. *Behavior Research Methods, Instruments, & Computers*, **34**, 613-617.

DAKIN, S. C., HESS, R. F., LEDGEWAY, T., & ACHTMAN, R. L. (2002). What causes non-monotonic tuning of fMRI response to noisy images? *Current Biology*, **12**, R476-R477.

DELORME, A., & MAKEIG, S. (2004). EEGLAB: An open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, **134**, 9-21.

DE VALOIS, R. L., & DE VALOIS, K. K. (1990). *Spatial vision*. New York: Oxford University Press.

FINKBEINER, M., & PALERMO, R. (2009). The role of spatial attention in nonconscious processing: A comparison of face and nonface stimuli. *Psychological Science*, **20**, 42-51.

FISET, D., BLAIS, C., GOSSELIN, F., BUB, D., & TANAKA, J. (2008). Potent features for the categorization of Caucasian, African American, and Asian faces in Caucasian observers [Abstract]. *Journal of Vision*, **8**(6), 258a.

FRÜND, I., BUSCH, N. A., KÖRNER, U., SCHADOW, J., & HERRMANN, C. S. (2007). EEG oscillations in the gamma and alpha range respond differently to spatial frequency. *Vision Research*, **47**, 2086-2098.

GEORGE, N., JEMEL, B., FIORI, N., & RENAULT, B. (1997). Face and shape repetition effects in humans: A spatio-temporal ERP study. *NeuroReport*, **8**, 1417-1423.

GOLD, J. G., BENNETT, P. J., & SEKULER, A. B. (1999). Identification

of band-pass filtered letters and faces by human and ideal observers. *Vision Research*, **39**, 3537-3560.

HARDEE, J. E., THOMPSON, J. C., & PUCE, A. (2008). The left amygdala knows fear: Laterality in the amygdala response to fearful eyes. *Social Cognitive & Affective Neuroscience*, **3**, 47-54.

HERSHLER, O., & HOCHSTEIN, S. (2006). With a careful look: Still no low-level confound to face pop-out. *Vision Research*, **46**, 3028-3035.

HONEY, C., KIRCHNER, H., & VANRULLEN, R. (2008). Faces in the cloud: Fourier power spectrum biases ultrarapid face detection. *Journal of Vision*, **8**(12, Art. 9), 1-13.

ITIER, R. J., & TAYLOR, M. J. (2004). N170 or N1? Spatiotemporal differences between object and face processing using ERPs. *Cerebral Cortex*, **14**, 132-142.

IVRY, R., & ROBERTSON, L. C. (1998). *The two sides of perception*. Cambridge, MA: MIT Press.

JOHANNES, S., MÜNTE, T. F., HEINZE, H. J., & MANGUN, G. R. (1995). Luminance and spatial attention effects on early visual processing. *Cognitive Brain Research*, **2**, 189-205.

KNEBEL, J.-F., TOEPEL, U., HUDRY, J., LE COUTRE, J., & MURRAY, M. M. (2008). Generating controlled image sets in cognitive neuroscience research. *Brain Topography*, **20**, 284-289.

LIANG, X., ZEBROWITZ, L. A., & AHARON, I. (2009). Effective connectivity between amygdala and orbitofrontal cortex differentiates the perception of facial expressions. *Social Neuroscience*, **4**, 185-196.

LIU, J., HARRIS, A., & KANWISHER, N. (2002). Stages of processing in face perception: An MEG study. *Nature Neuroscience*, **5**, 910-916.

LOSCHKY, L. C., SETHI, A., SIMONS, D. J., PYDIMARI, T. W., OCHS, D., & CORBEILLE, J. L. (2007). The importance of information localization in scene gist recognition. *Journal of Experimental Psychology: Human Perception & Performance*, **33**, 1431-1450.

LUCK, S. J. (2005). An introduction to the event-related potential technique. Cambridge, MA: MIT Press.

MACK, M. L., GAUTHIER, I., SADR, J., & PALMERI, T. J. (2008). Object detection and basic-level categorization: Sometimes you know it is there before you know what it is. *Psychological Bulletin & Review*, **15**, 28-35.

METHA, A. B., VINGRYS, A. J., & BADCOCK, D. R. (1993). Calibration of a color monitor for visual psychophysics. *Behavior Research Methods, Instruments, & Computers*, **25**, 371-383.

MOROVIC, J., SHAW, J., & SUN, P.-L. (2002). A fast, non-iterative and exact histogram matching algorithm. *Pattern Recognition Letters*, **23**, 127-135.

MOTOYOSHI, I., NISHIDA, S., SHARAN, L., & ADELSON, E. H. (2007). Image statistics and the perception of surface qualities. *Nature*, **447**, 206-209.

OLMAN, C., BOYACI, H., FANG, F., & DOERSCHNER, K. (2008). V1 responses to different types of luminance histogram contrast [Abstract]. *Journal of Vision*, **8**(6), 345a.

PELLI, D. G. (1997). The VideoToolbox software for visual psychophysics: Transforming numbers into movies. *Spatial Vision*, **10**, 437-442.

PELLI, D. G., & ZHANG, L. (1991). Accurate control of contrast on microcomputer displays. *Vision Research*, **31**, 1337-1350.

POYNTON, C. A. (1993). "Gamma" and its disguises: The nonlinear mappings of intensity in perception, CRTs, film and video. *SMPTE Journal*, **102**, 1099-1108.

ROLLAND, J. P., VO, V., BLOSS, B., & ABBEY, C. K. (2000). Fast algorithms for histogram matching: Application to texture synthesis. *Journal of Electronic Imaging*, **9**, 39-45.

ROSE, J., OTTO, T., & DITTRICH, L. (2008). The Biopsychology-Toolbox: A free, open-source Matlab-toolbox for the control of behavioral experiments. *Journal of Neuroscience Methods*, **175**, 104-107.

ROSSION, B., GAUTHIER, I., TARR, M. J., DESPLAND, P., BRUYER, R., LINOTTE, S., & CROMMELINCK, M. (2000). The N170 occipito-temporal component is delayed and enhanced to inverted faces but not to inverted objects: An electrophysiological account of face-specific processes in the human brain. *NeuroReport*, **11**, 69-74.

ROUSSELET, G. A., HUSK, J. S., BENNETT, P. J., & SEKULER, A. B. (2008). Time course and robustness of ERP object and face differences. *Journal of Vision*, **8**(12, Art. 3), 1-18.

ROUSSELET, G. A., MACÉ, M. J. M., THORPE, S. J., & FABRE-THORPE, M. (2007). Limits of event-related potential differences in tracking object processing speed. *Journal of Cognitive Neuroscience*, **19**, 1241-1258.

ROUSSELET, G. A., PERNET, C. R., BENNETT, P. J., & SEKULER, A. B. (2008). Parametric study of EEG sensitivity to phase noise during face processing. *BMC Neuroscience*, **9**, 98.

SADR, J., & SINHA, P. (2001). *Exploring object perception with random image structure evolution*. MIT Artificial Intelligence Laboratory Memo No. 2001–6.

SADR, J., & SINHA, P. (2004). Object recognition and random image structure evolution. *Cognitive Science*, **28**, 259-287.

SEECK, M., MICHEL, C. M., MAINWARING, N., COSGROVE, R., BLUME, H., IVES, J., ET AL. (1997). Evidence for rapid face recognition from human scalp and intracranial electrodes. *NeuroReport*, **8**, 2749-2754.

STANISLAW, H., & OLZAK, L. A. (1990). Parametric methods for gamma and inverse gamma correction with extensions to halftoning. *Behavior Research Methods, Instruments, & Computers*, **22**, 402-408.

TANAKA, J. W., & CURRAN, T. (2001). A neural basis for expert object recognition. *Psychological Science*, **12**, 43-47.

THIERRY, G., MARTIN, C. D., DOWNING, P., & PEGNA, A. J. (2007a). Controlling for interstimulus perceptual variance abolishes N170 face selectivity. *Nature Neuroscience*, **10**, 505-511.

THIERRY, G., MARTIN, C. D., DOWNING, P. E., & PEGNA, A. J. (2007b). Is the N170 sensitive to the human face or to several intertwined perceptual and conceptual factors? *Nature Neuroscience*, **10**, 802-803.

VANRULLEN, R. (2006). On second glance: Still no high-level pop-out effect for faces. *Vision Research*, **46**, 3017-3027.

VANRULLEN, R., & THORPE, S. J. (2001). The time course of visual processing: From early perception to decision-making. *Journal of Cognitive Neuroscience*, **13**, 454-461.

WAN, Y., & SHI, D. (2007). Joint exact histogram specification and image enhancement through the wavelet transform. *IEEE Transactions on Image Processing*, **16**, 2245-2250.

WANG, Z., BOVIK, A. C., SHEIKH, H. R., & SIMONCELLI, E. P. (2003). *The SSIM index for image quality assessment*. Retrieved January 27, 2010 from www.ece.uwaterloo.ca/~z70wang/research/ssim/

WANG, Z., BOVIK, A. C., SHEIKH, H. R., & SIMONCELLI, E. P. (2004). Image quality assessment: From error visibility to structural similarity. *IEEE Transactions on Image Processing*, **13**, 600-612.

WILLENBOCKEL, V., FISET, D., CHAUVIN, A., BLAIS, C. ARGUIN, M., TANAKA, J. W., ET. AL (2010). Does face inversion change spatial frequency tuning? *Journal of Experimental Psychology: Human Perception & Performance*, **36**, 122-135.

WILLENBOCKEL, V., FISET, D., & TANAKA, J. W. (2008, June). *The role of luminance and facial features in race categorization*. Poster presented at the North West Cognition and Memory 10th Annual Meeting, Seattle.

WILLIAMS, N. R., WILLENBOCKEL, V., & GAUTHIER, I. (2009). Sensitivity to spatial frequency and orientation content is not specific to face perception. *Vision Research*, **49**, 2353-2362.

XU, Y., LIU, J., & KANWISHER, N. (2005). The M170 is selective for faces, not for expertise. *Neuropsychologia*, **43**, 588-597.

YIN, R. K. (1969). Looking at upside-down faces. *Journal of Experimental Psychology*, **81**, 141-145.

ZION-GOLUMBIC, E., GOLAN, T., ANAKI, D., & BENTIN, S. (2008). Human face preference in gamma-frequency EEG activity. *NeuroImage*, **39**, 1980-1987.

NOTES

1. The *ssim_sens* function was obtained from A. N. Avanaki. For further details, please contact him at avanaki@yahoo.com.

2. *fft2* outputs the sine and cosine coefficients, which can be transformed into amplitudes using abs(fft2(an_image)) and in phases using angle(fft2(an_image)).