Controlling low-level image properties: The SHINE toolbox

Verena Willenbockel[1], Javid Sadr[2], Daniel Fiset[1], Greg O. Horne[3],

Frédéric Gosselin[1], & James W. Tanaka[3]

[1] Centre de Recherche en Neuropsychologie et Cognition, Département de Psychologie, Université de Montréal, CP 6128, succ. Centre-ville, Montréal, QC, H3C 3J7, CANADA

[2] Department of Psychology, University of Massachusetts Boston, 100 Morrissey Boulevard, Boston, MA 02125, USA

[3] Department of Psychology, University of Victoria, PO Box 3050 STN CSC, Victoria, BC, V8W 3P5, CANADA

Address correspondence to

Verena Willenbockel
Département de psychologie
Université de Montréal
C.P. 6128 Succ. Centre-ville
Montréal, QC
H3C 3J7, Canada
Phone: 514 738 8662
Email: verena.vw@gmail.com

Abstract

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 contribution of low- and high-level factors. Here we make available the SHINE (Spectrum, Histogram, and Intensity Normalization and Equalization) toolbox written with Matlab, which we (and others) 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, as well as for exact histogram specification. SHINE can thus be employed for parametrically modifying a number of image properties or for equating them across the input set to minimize potential low-level confounds in studies on higher-level processes.

Keywords: visual perception, low-level image properties, luminance, histogram matching, spatial frequency, Fourier spectra

1. Introduction

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 a group of bird experts. The differences in brain activity observed when 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 examining the domain-specificity versus domain-generality accounts of face processing typically involve comparisons between faces and non-face 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 (Fruend, Busch, Koerner, 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, & Achtmann, 2002; Rousselet, Pernet, Bennett, & Sekuler, 2008; Thierry, Martin, Downing, Pegna, 2007a, 2007b), and the importance of avoiding low-level confounds has particularly been emphasized for the event-related potential (ERP) technique (Luck, 2005). Inconsistent findings were obtained, for instance, about the earliest "real" ERP differences between face and object processing. Some studies focus on the N170 component—a negative deflection occurring about 170 ms after stimulus onset—as the first marker of face processing (e.g., Carmel & Bentin, 2002; Rousselet, Husk, Bennett, & Sekuler, 2008; Rossion et al., 2000), whereas other studies revealed earlier differences between face and object processing, possibly as early as 50-80 ms after stimulus onset (e.g., George, Jemel, Fiori, & Renault, 1997; Seeck et al., 1997). As the studies differ in the degree to which they controlled image properties across face and non-face categories, the different findings might at least in part be explained by differences in low-level influences (Rousselet, Husk, Bennett, & Sekuler, 2008). In his book on the ERP technique, 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 where 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, as indicated for instance by the modulation of early ERP components such as the P1 (e.g., Johannes, Münte, Heinze, & Mangun, 1995). Some studies have accounted for this by equating stimuli in terms of mean luminance and contrast (e.g., Hardee, Thompson, & Puce, 2008; Liang, Zebrowitz, & Aharon, 2009; Sharpee, Miller, & Stryker, 2008). It is possible to go a step further and precisely match the luminance histograms—which give the number of pixels at each grayscale level—across images, thereby not only equating 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, 1999b). 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, USA) for image pre-processing, running experiments, as well as for 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 or the Signal Processing Toolbox, there are a number of freely available ones, such as the Psychophysics Toolbox (Brainard, 1997; Pelli, 1997), EEGLAB (Delorme & Makeig, 2004), the Biopsychology Toolbox (Rose, Otto, & Dittrich, 2008), and the alignment toolbox from our own lab (www.mapageweb.umontreal.ca/gosselif/align%20tools%20OSX).

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[1]) 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 precisely specifying the Fourier amplitude spectra of images, or scaling the rotational average of the amplitude spectra only (i.e., the energy at each spatial frequency averaged across orientations). It also includes functions for specifying the luminance histograms, or for normalizing and scaling 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. Although histogram specification and Fourier amplitude specification affect each other, we discovered that by using an iterative approach, in many cases a high degree of simultaneous matching of the low-level properties of interest could be reached. SHINE also features tools that allow for plotting the (rotational average of the) Fourier amplitude spectra in order to verify the output or monitor the ecological low-level variations.

The following sections provide an overview on the individual SHINE functions and give examples on how to work with the toolbox, which can be freely downloaded here: www.mapageweb.umontreal.ca/gosselif/shine. The toolbox has successfully been used by us as well as by others (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, a number of other behavioral and neuroscientific studies successfully used earlier or partial forms of these preprocessing steps to achieve strict stimulus control (e.g., Liu, Harris, & Kanwisher, 2002; Loschky et al., 2007; Mack, Gauthier, Sadr, & Palmeri, 2008; Sadr & Sinha, 2001, 2004; Xu, Liu, & Kanwisher, 2005).

2. Methods

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 input and output folders, as well as the parameters, e.g., for 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.

Insert Figure 1 about here

2.1 Normalizing and scaling mean luminance and contrast

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

$$Z = (X - m) / s$$

X is the input matrix containing the original grayscale values, m is the mean luminance of the original values, and s is the standard deviation. The desired mean (M) and desired standard deviation (S) can then be applied to obtain the output image matrix E containing the adjusted grayscale values:

$$E = Z*S+M$$

The default values for M and S are the average of the means of all input stimuli and the mean standard deviation across all input stimuli, respectively.

As with exact histogram matching (see section 2.2, below), 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 wants to equate a stimulus consisting of a relatively bright object on a mid-gray uniform background with a stimulus consisting of a relatively dark object on the same mid-gray background. If the default matching step is applied to the whole image, the first stimulus will end up with a darker background than the second image, and the background alone would thus contain diagnostic information.

Therefore, in some cases, one might want to consider doing the matching step separately for the foreground and the background.

2.2 Luminance histogram matching

The SHINE toolbox includes the function *histMatch* that 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 one 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,* serves as default target (Figure 2). Alternatively, it is possible to provide the *histMatch* function with another target histogram, 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 (i.e., the darkest values across all images, the second-darkest, etc.) to obtain the target (Figure 3a and b).

Insert Figure 2 about here

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 sub-optimal point-wise transformation of the grayscale levels. 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 Figure 3 for examples and section 2.6 for an image quality measure).

Different ways for 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).

SHINE follows the exact histogram matching approach as described in Box 1 (see also Avanaki, 2009; Coltuc, Bolon, & Chassery, 2006; Wan & Shi, 2007). First, the pixels of the source image and the pixels of the target distribution are sorted separately by their grayscale level from darkest to lightest. The darkest pixel of the source image is then assigned the darkest value of the target histogram, the second darkest pixel of the source image is assigned the second darkest value of the target and so forth. Ambiguity arises when a number of pixels with the same original intensity have to be broken down into two or more groups to match the target: it has to be decided which pixel(s) get(s) a new intensity. For example, this would be the case if the two darkest pixels of the source image had the same intensity (e.g., 0) but the target histogram contained only one pixel with an intensity of 0. So which of the two source pixels should be assigned a new value? The current version of SHINE uses random assignment (as applied in Williams, Willenbockel, & Gauthier, 2009), but different approaches have been suggested (Avanaki, 2009; Coltuc, Bolon, & Chassery, 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 (Figure 3). In this way, mean luminance, contrast, and all other characteristics of the histogram (e.g., skew) are fully equated across all stimuli.

SHINE includes an option for applying the histogram matching step to the whole image (e.g., this might be used for photographs of scenes) or to subsets of pixels, for example, to the foreground and background separately (this might be used for stimuli displaying a single object on a uniform or noisy background; see the previous section for a brief discussion about the advantages of separate matching).

Insert Figure 3 about here

2.3 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 as specified in degrees counter-clockwise 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* that matches the amplitude spectrum of the source image with a specified target spectrum (Figure 4). First, each *m* x *m* source image is submitted to a fast Fourier transform (FFT). If the input image is not in square format, the *padding* function is used to place it on a larger square field—which by default is of the image's background luminance—before applying the Fourier transform. Then 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). If the input image was not m x m, the output is cropped to the original size.

This approach matches the source spectrum exactly with the target spectrum at each spatial frequency and orientation. The output images typically look a bit "cloudy" and their visual quality depends on how similar the input images were—for example, equating the spectra of a number of faces will likely yield better results than equating the spectra across different categories such as faces and cars. If image quality is a concern, we propose to use the *sfMatch* function described below.

Insert Figure 4 about here

2.4 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 source and target images (Figure 4). The initial steps are the same as described above for *specMatch*: the (padded) 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. Then, if not specified otherwise, the target spectrum is obtained by averaging across all input spectra. For each spatial frequency, a coefficient is computed: the amplitudes across all 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 then divided by the corresponding source sum to get the respective coefficient. After computing the coefficients, the amplitude at each spatial frequency and orientation is multiplied with the corresponding coefficient. The phase of the original image is then 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 arise), while the distribution of amplitudes among orientations is preserved. While controlling overall amplitude, this often results in higher similarity of the output image with the source image than strictly equating the amplitudes at each spatial frequency and each orientation.

2.5 Rescaling of intensity values after the IFFT

It is possible that after equating the Fourier amplitudes and applying the IFFT, the intensity 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, as well as in a small change in the amplitude spectrum. Thus, one might want to rescale the luminance after Fourier amplitude matching in such a way that all or at least the majority of the luminance values fall back into the range of 0 to 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 one set out to normalize in the first place. The *rescale* function of SHINE therefore first obtains the full range of intensity values coming out of the IFFT, for all the images, and then computes one set of rescaling values to apply to each image. Specifically, after going through all 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 either (a) absolutely all pixels for all images are in the range of 0 to 255, or (b) on average, the darkest and lightest pixels are rescaled to 0 and 255, but still allowing some clipping 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.

2.6 Matching certain low-level properties simultaneously

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 histograms and Fourier amplitudes. However, 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, and 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 address the first problem of decreased visual quality, we added an image quality measure function (i.e., the root-mean square error) to SHINE. By calling *getRMSE*, the *SHINE* function automatically provides the root-mean square error, which shows how much the input and output images were altered. While it will be impossible to completely avoid alterations to the visual quality, this approach allows one to compare the different matching approaches objectively.

In order to account for the second problem of histogram and Fourier amplitude specification affecting each other, we have 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 simultaneous matching in previous studies using SHINE (Figure 5). However, depending on the input set and the desired degree of matching (and visual quality), one might chose different numbers of iterations and might also want to apply the steps in a different order: that is, if matched histograms are most important, one might want to do histogram matching last; otherwise, amplitude matching might go last.

Insert Figure 5 about here

2.7 Plotting functions and image statistics

The Image Processing Toolbox of Matlab already 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 spectra itself. SHINE includes these last two plotting functions (*sfPlot* and *specPlot*, respectively) which can be useful for checking the SHINEd 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).

2.8 Applying SHINE to color images

SHINE was originally designed for the pre-processing of grayscale images, however, 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 the HSL color space does. One would then equate the histograms—or just the mean and standard deviation of the luminance distribution—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 is thus analogous to how it is described for one-layer grayscale images but would have to be applied three times for each RGB color image.

3. Summary: The default SHINE

The default version of SHINE exactly equates the luminance histograms across all input images, so that each output image has the same luminance distribution as the average of the input set. Afterwards, it equates the Fourier amplitude spectra, whereby the average spectrum obtained across all input images serves as target. Rescaling ensures that luminance values stay in the desired range and are not clipped. Histogram and Fourier amplitude matching are performed iteratively a number of times, as specified by the researcher depending on the requirements of the experiment, to obtain a high degree of simultaneous matching of luminance properties and Fourier amplitudes.

4. 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 luminance histogram specification, as well as 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 simultaneous 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 stressed the importance of disentangling low- and high-level factors when using different physical stimuli across psychological conditions (e.g., Luck, 2005; Rousselet, Husk, Bennett, & Sekuler, 2008; Sadr & Sinha, 2001, 2004). For example, Luck (2005) emphasized in his book on the ERP technique that even small physical stimulus differences can lead to significant differences in the ERPs. 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 on visual perception 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, however 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 the *match* function made available here. Furthermore, SHINE extends the toolbox by including tools for the direct manipulation and visualization of the (rotational average of the) Fourier amplitude spectra.

SHINE has recently been applied in a study comparing the spatial frequency sensitivity of face and object (i.e., car and chair) processing (Williams, Willenbockel, & Gauthier, 2009). The rotational average of the Fourier amplitude spectra as well as luminance histograms were matched across a total of 45 images from three categories (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 levels were then applied to both Caucasian and African-American faces (Figure 6). Likewise, a number of other studies have also used earlier or partial forms of these processing steps to create well-controlled stimulus image sets (e.g., Liu, Harris, & Kanwisher, 2002; Loschky et al., 2007; Mack et al., 2008; Sadr & Sinha, 2001, 2004; Willenbockel et al., in press; Xu, Liu, & Kanwisher, 2005).

Insert Figure 6 about here

The SHINE approach has thus been successfully used for both equating and parametrically varying low-level properties, while 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.

Footnotes

[1] We focus here on equating the low-level properties between stimuli and do not refer to methods for contrast enhancement / flattening the histogram which are referred to as *histogram equalization* in the computer science literature.

[2] *fft2* outputs the sin and cosin coefficients, which can be transformed into amplitudes using abs(fft2(an_image)) and in phases using angle(fft2(an_image)).

References

Avanaki, A. N. (2009). Exact histogram specification optimized for structural similarity. Pre-print arXiv:0901.0065.

Bentin, S., Taylor, M. J., Rousselet, G. A., Itier, R. J., Caldara, R., Schyns, P. G., Jacques, C., & Rossion, B. (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. *Proc. IEEE of Int. Conf. On Image Analysis and Processing* (ICIAP'07), 623–628.

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

Campbell, F. W., & Robson, J. (1968). Application of Fourier analysis to the visibility of gratings. *Journal of Physiology, London*, 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(27)*, 3223–3232.

Coltuc, D., Bolon, P., & Chassery, J.-M. (2006). Exact histogram specification. *IEEE Trans. Image Processing*, *15(5)*, 1143–1152.

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(14)*, 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*.

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):258, 258a, http://journalofvision.org/8/6/258/, doi:10.1167/8.6.258.

Fruend, I., Busch, N. A., Koerner, 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. (1999b). Identification of band-pass filtered letters and faces by human and ideal observers. *Vision Research, 39(21),* 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 and Affective Neuroscience, 3(1)*, 47–45.

Itier, R. J., & Taylor, M. J. (2004). N170 or N1? Spatiotemporal differences between object and face processing using ERPs. *Cerebral Cortex, 14(2)*, 132–142.

Ivry, R., & Robertson, L. C. (1998). *The Two Sides of Perception*. Cambridge: 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(3),* 189–205.

Liang, X., Zebrowitz, L. A., & Aharon, I. (2009). Effective connectivity between amygdala and orbitofrontal cortex differentiates the perception of facial expressions. *Social Neuroscience, 4*(2), 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., Ochs, D., & Corbeille, J. (2007). The Importance of Information Localization in Scene Gist Recognition. *Journal of Experimental Psychology: Human Perception and Performance, 33(6)*, 1431–1450.

Luck, S. J. (2005). *An introduction to the event-related potential technique*. Cambridge, MA: The 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.

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):345, 345a, http://journalofvision.org/8/6/345/, doi:10.1167/8.6.345.

Pelli, D. G. (1997). The VideoToolbox software for visual psychophysics: transforming number into movies. *Spatial Vision, 10,* 437–42.

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 (2000). The N170 occipitotemporal 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., Macé, M. J. M., Thorpe, S. J., Fabre-Thorpe, M. (2007). Limits of ERP differences in tracking object processing speed. *Journal of Cognitive Neuroscience, 19(8)*, 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., Landis, T., & Schomer, D. L. (1997). Evidence for rapid face recognition from human scalp and intracranial electrodes. *Neuroreport, 8*, 2749–2754.

Sharpee, T. O., Miller, K. D., & Stryker, M. P. (2008). On the importance of static nonlinearity in estimating spatiotemporal neural filters with natural stimuli. *Journal of Neurophysiology, 99,* 2496–2509.

Tanaka, J. W., & Curran, T. (2001). A neural basis for expert object recognition. *Psychological Science, 12(1)*, 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., & 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., & Thorpe, S. J. (2001). The time course of visual processing: from early perception to decision-making. *Journal of Cognitive Neuroscience, 13(4)*, 454–461.

Wan, Y., & Shi, D. (2007). Joint exact histogram specification and image enhancement through the wavelet transform. IEEE Trans. *Image Processing, 16(9),* 2245–2250.

Willenbockel, V., Fiset, D., Chauvin, A., Blais, C. Arguin, M., Tanaka, J. W., Bub, D. N., & Gosselin, F. (in press). Does face inversion change spatial frequency tuning? *Journal of Experimental Psychology: Human Perception and Performance.*

Willenbockel, V., Fiset, D., & Tanaka, J. W. (2008). The role of luminance and facial features in race categorization. North West Cognition and Memory 10th Annual Meeting, Seattle, USA.

Williams, N. R., Willenbockel, V., & Gauthier, I. (2009). Sensitivity to spatial frequency and orientation content is not specific to face perception. *Vision Research*, 49(19), 2353–2362.

Xu, Y., Liu, J., & Kanwisher, N. (2005). The M170 is selective for faces, not for expertise. *Neuropsychologia, 43*, 588–587.

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

Box 1 (adapted from Avanaki, 2009): Algorithm

Step 0: The target histogram is given by H = {h_0, h_1, …, H_{L-1}}, where L is the number of possible grayscale levels (e.g., 256 in an 8-bit image). It is assumed that sum(H) = M, where M 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 original image by their intensity in ascending order (and in the cases of same intensity randomize the pixels).

Step 2: Starting from the first pixel on the sorted list, assign the first h_0 pixels a new intensity of 0. Continue by assigning the next h_1 pixels a new intensity of 1, and so on until all pixels are assigned their new intensities.

Figure captions

Figure 1. Overview of SHINE toolbox functions and their calling structure. For example, the *SHINE* function calls the *histMatch* function, which calls *avgHist* and *match*.

Figure 2. Illustration of the luminance histogram matching using simple patterns. The histograms show luminance (arbitrary linear unit—or ALU, for short) on the x-axis and the number of pixels on the y-axis. The target histogram was the average of the histograms of the two input images. a: Histogram matching was performed on two uniform surfaces (100 ALU and 200 ALU) of slightly different sizes centered on a uniform gray background (127 ALU). The output surfaces contain 50% of 100 ALU pixels and 50% of 200 ALU pixels in randomized order. The background was unaltered. b: Two surfaces of the same size as in a) with 50% 100 ALU and 50% 200 ALU pixels served as input. The output is identical to the input. c: Illustration with surfaces similar to b but with different amounts of dark and light pixels. d: Illustration with input surfaces of three different grayscale levels (100, 200, and 230 ALU) with different numbers of pixels of each.

Figure 3. Illustration of the luminance histogram matching function of SHINE. a: Two base face images with their luminance histograms (left) and the corresponding SHINEd 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 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.

Figure 4. A car, a chair, and a face image as well as their Fourier spectra before (top) and after having applied SHINE (middle and bottom). Middle: Using the function *sfMatch*, the rotational average of the Fourier spectra, i.e., the energy averaged across orientations at each spatial frequency (cpi = cycles per image), was matched while the energy distribution across orientations was preserved (see text for details). Bottom: Using *specMatch*, the Fourier spectra were equated on spatial frequencies and orientations. Shown is the output after the rescaling of the luminance values so that absolutely all pixels of the three images are in the range of 0 to 255.

Figure 5. Iterative histogram and Fourier energy matching. The histograms of three sample images (a car, a chair, and a face) are shown before applying SHINE (top), after 1 iteration (middle) and after 3 iterations (bottom). Each iteration consisted of equating the histograms and then equating the rotational average of the Fourier spectra. The latter step altered the histograms, but, as shown, after a number of iterations, the histograms typically converge toward the target histogram.

Figure 6. Subset of the stimuli used by Willenbockel, Fiset, and Tanaka (2008) in a study on the relative influence of skin tone and featural information on race categorization. Using the *histMatch* function of SHINE, the histograms of African-American and Caucasian faces were parametrically altered. The top right shows the original Caucasian face and the bottom left shows the original African-American face. The faces that are vertically aligned have the same luminance distributions.

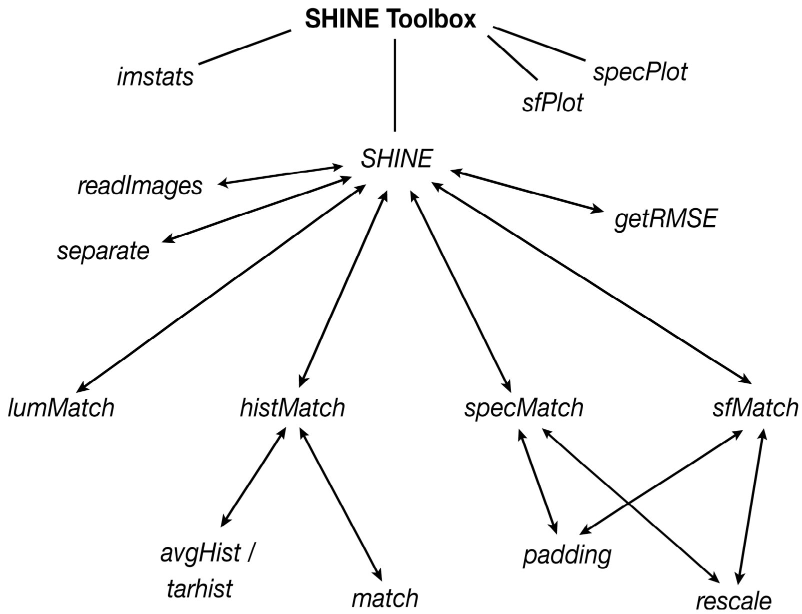
SHINE Toolbox
imstats
sfPlot
specPlot
SHINE
readImages
separate
getRMSE
lumMatch
histMatch
specMatch
sfMatch
avgHist /
tarhist
match
padding
rescale

Figure 1

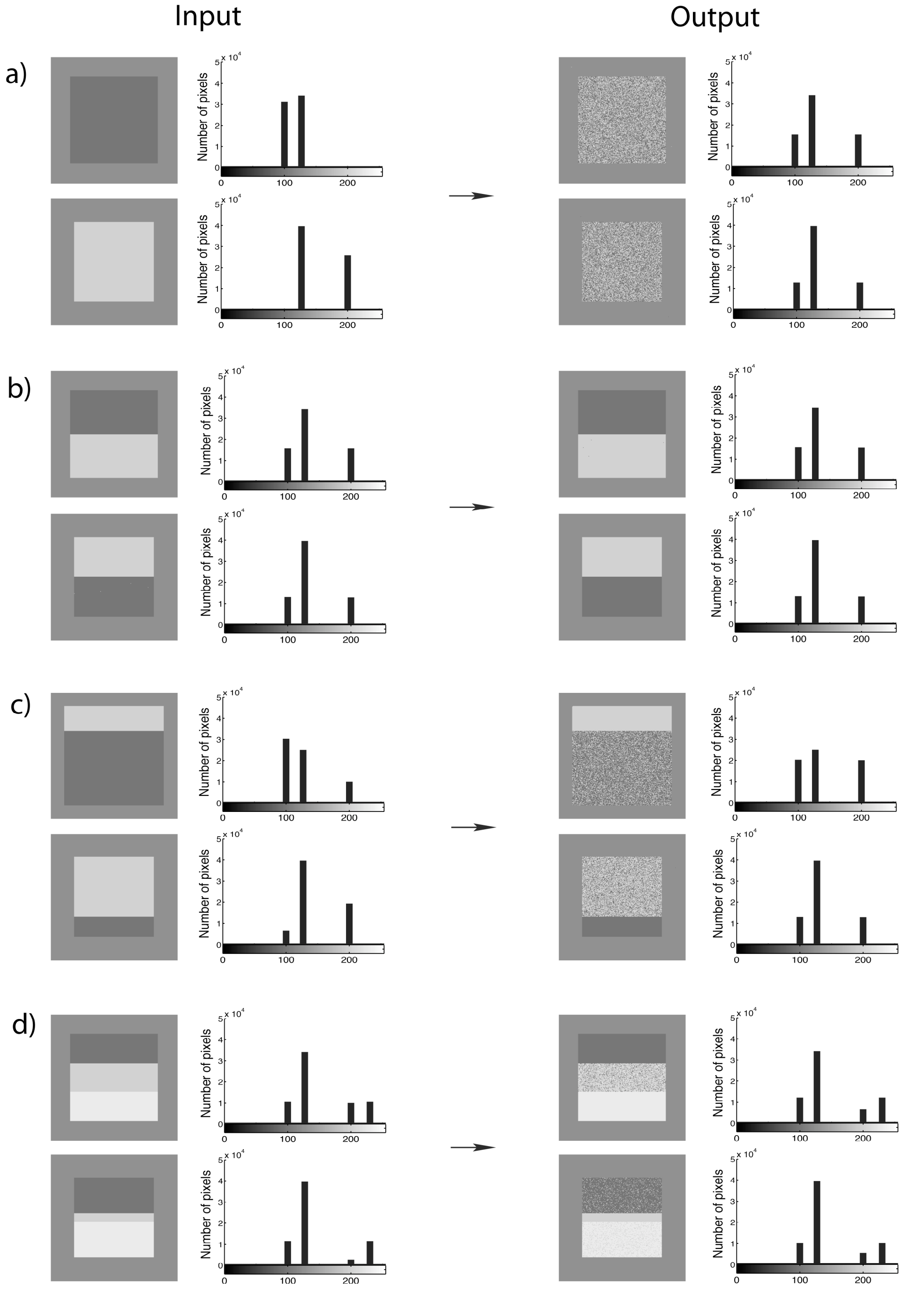

Figure 2

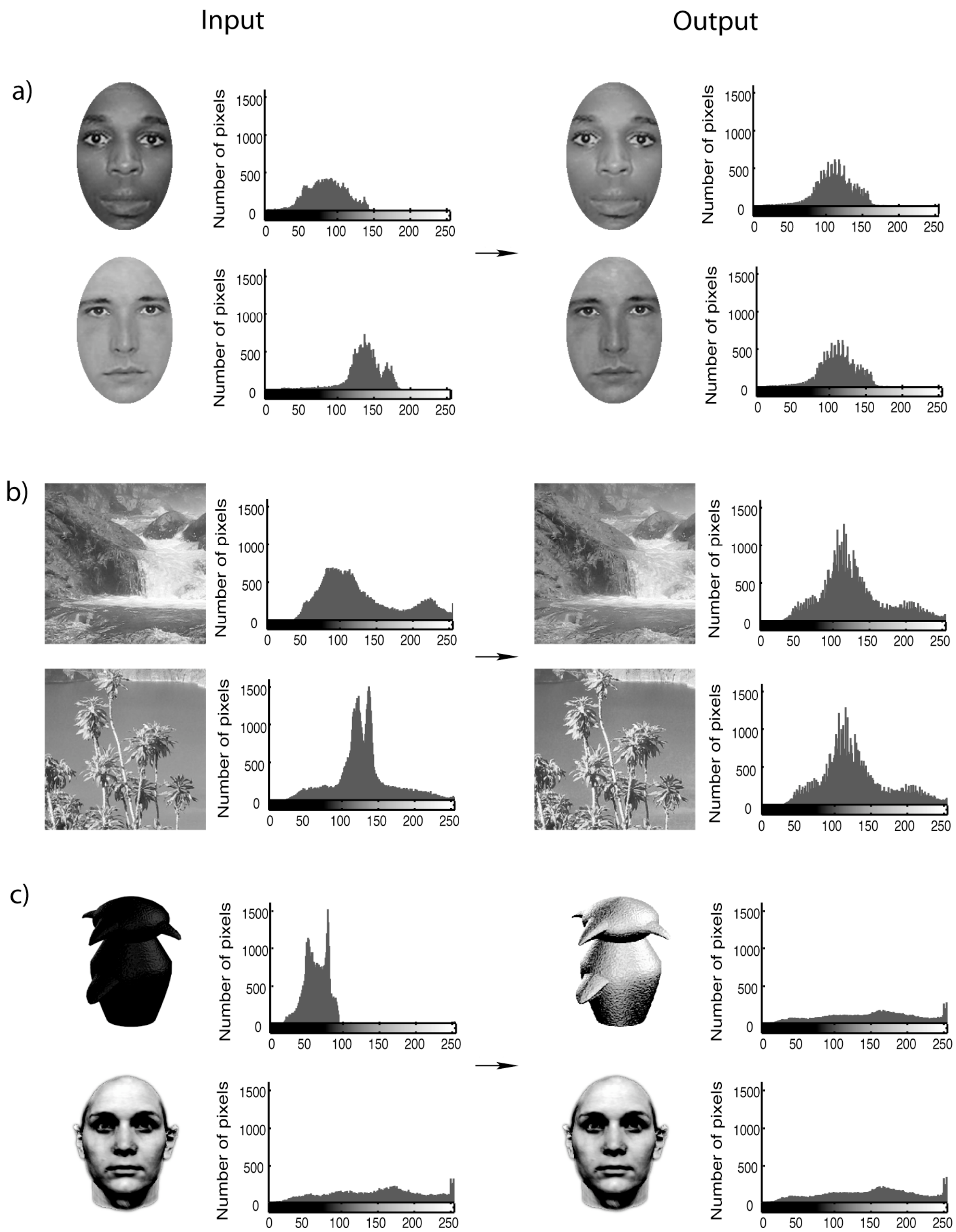
Input
Output
a)
b)
c)
Number of pixels
0
500
1000
1500
0 50 100 150 200 250

Figure 3

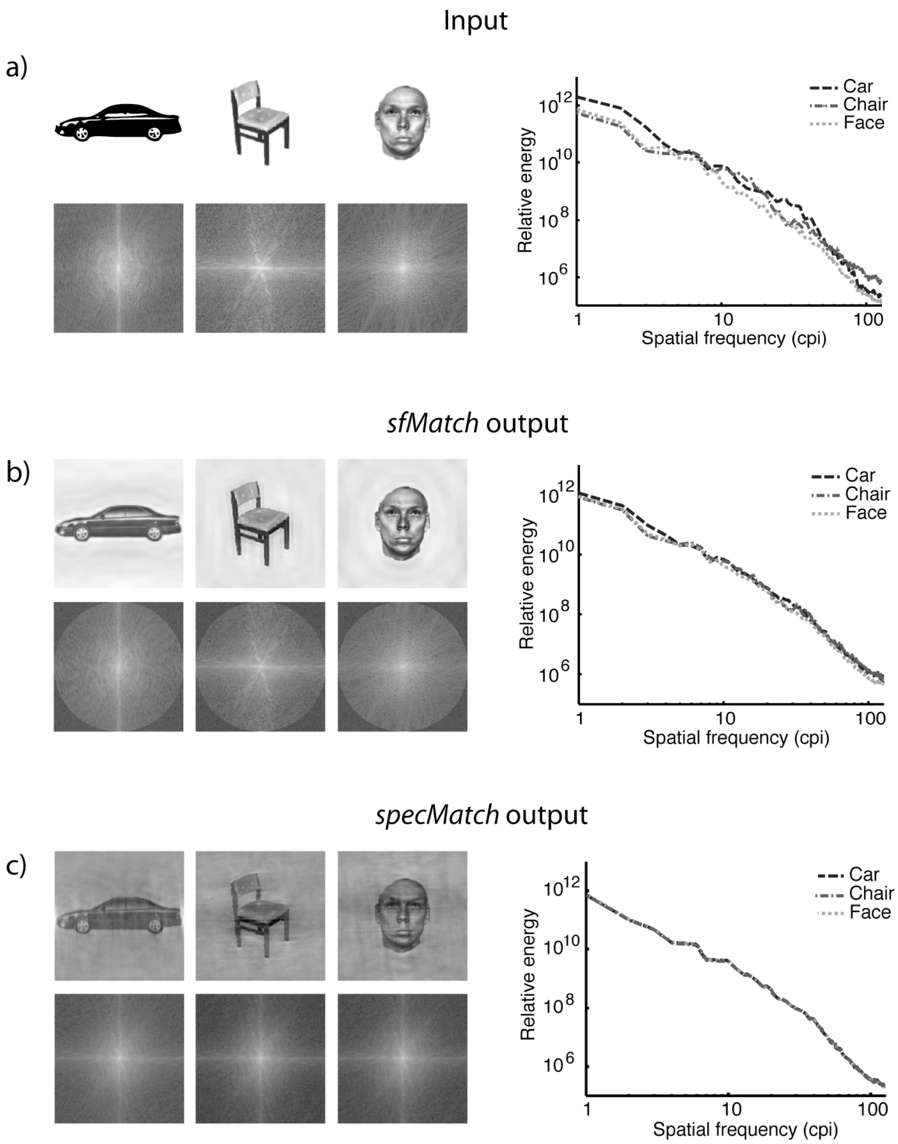
Input
a)
Car
Chair
Face
Relative energy
Spatial frequency (cpi)
sfMatch output
b)
Car
Chair
Face
Relative energy
Spatial frequency (cpi)
specMatch output
c)
Car
Chair
Face
Relative energy
Spatial frequency (cpi)

Figure 4

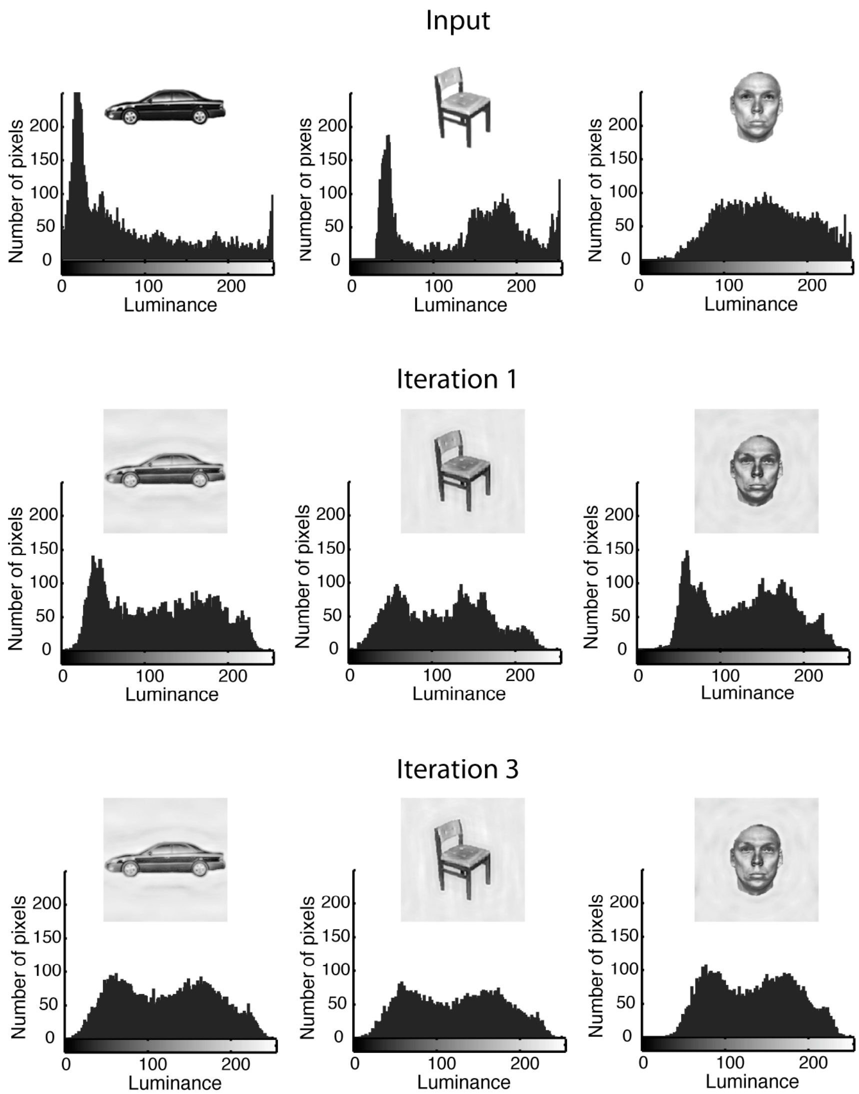

Figure 5

Figure 6