

# SHINE manual

## LIST OF CONTENTS

1. Initial Setup	p. 2
2. Test run	p. 2
3. Trouble shooting	p. 3
4. SHINE overview	p. 3
4.1 Applying SHINE to a set of images saved in the input folder	p. 3
4.2 Applying SHINE to a set of pre-loaded images	p. 4
4.3 Applying SHINE to a set of pre-loaded images using foreground/background templates	p. 5
5. Shine options	p. 6
6. Overview of the individual matching functions	p. 7
6.1 histMatch	p. 7
6.2 lumMatch	p. 8
6.3 specMatch	p. 8
6.4 sfMatch	p. 9
7. Overview of the plotting functions	p. 9
7.1 sfPlot	p. 9
7.2 spectrumPlot	p. 10
8. References	p. 10

## 1. Initial setup

a) After downloading and unzipping the SHINE toolbox folder, put it in the Matlab *work* folder (e.g., C:/Program files/MATLAB/work), as this is the default path specified in *SHINE.m*. If the toolbox is saved at another location, the pathname will have to be updated in *SHINE.m* as described in section 3.

b) Open Matlab

c) Verify that Matlab's Image Processing Toolbox is installed

d) In order to include the SHINE toolbox in the Matlab path, go to *File -> Set Path -> Add with Subfolders -> select the SHINE toolbox folder -> Open -> Save -> Close*

## 2. Test run

a) To do a test run with the provided sample images (see the *SHINE\_INPUT* folder), type *SHINE* in the Matlab command window and press enter.

The following should appear:

```
SHINE options      [1=default, 2=custom]
```

b) Press 1 (followed by enter)

The following summary should appear:

```
Number of images: 3
```

```
Option:  specMatch & histMatch with 1 iteration(s)
```

```
Option:  whole image
```

```
Progress: specMatch successful
```

```
Progress: histMatch successful
```

```
RMSE:    ...
```

```
SSIM:    ...
```

The output images matched in amplitude spectra and histograms should be saved automatically in the *SHINE\_OUTPUT* folder.

### 3. Trouble shooting

If Matlab could not read the images, double-check the location of the SHINE toolbox folder. SHINE assumes that it is in the Matlab *work* folder, which in turn is assumed to be at the root directory of Matlab installation. To see the root directory, type *matlabroot* in the command window.

The pathname(s) can be changed in *SHINE.m* at lines 70-72:

```
input_folder =  
fullfile(matlabroot, 'work', 'SHINEtoolbox', 'SHINE_INPUT');  
output_folder =  
fullfile(matlabroot, 'work', 'SHINEtoolbox', 'SHINE_OUTPUT');  
template_folder =  
fullfile(matlabroot, 'work', 'SHINEtoolbox', 'SHINE_TEMPLATE');
```

### 4. SHINE overview

There are different ways the main program (see *SHINE.m*) can be called, which will be explained in the following paragraphs:

```
SHINE  
newimages = SHINE;  
newimages = SHINE(images);  
newimages = SHINE(images, templates);
```

See also *help SHINE* for an explanation of the input and output arguments.

#### **4.1 Applying SHINE to a set of images saved in the input folder**

SHINE can be called without any input arguments (*SHINE* or *newimages = SHINE*). If that is the case, the program automatically loads the to-be-equated images from the input folder specified in *SHINE.m*. Specifically, using the *readImages* function, it loads ALL the images of the specified format (e.g., tif; see *SHINE.m*, line 69) in that folder, and it equates them in the low-level properties specified by the user (see below).

How to use SHINE with saved images:

- Put the to-be-SHINED grayscale images in the input folder (by default, it is the folder *SHINE\_INPUT*).

- If you intend to equate the foregrounds and the backgrounds separately, put the template(s) in the template folder (by default, this is the folder *SHINE\_TEMPLATE*). The template(s) must be of the same image format and size as the to-be-equated images and have a uniform background of a luminance that does NOT occur in the foreground (e.g., a white background of luminance 255 and a foreground of values between 0 and 254).
- Make sure that there are no other images of the same file format (e.g., .tif) in the input (and template) folder(s).
- .tif images are recommended because SHINE has most thoroughly been tested with those, but it should work with other image formats as well.
- Open *SHINE.m* (i.e., by typing *edit SHINE* in the Matlab command window)
- Specify the image format in line 69 (e.g., *imformat = 'tif';*)
- Save changes
  
- In the Matlab command window, type *SHINE* and press Enter.
- Choose among the SHINE options. If you want to quit the program, press ENTER without choosing an option.
  
- If SHINE was called without an output argument, the equated images should be saved in the output folder (by default, this is the folder *SHINE\_OUTPUT*). The images are saved under the name *SHINEd\_originalname*. If desired, the output name can be changed in *SHINE.m* (see the bottom of the script in the lines beginning with *imwrite*).
  
- If SHINE was called with an output argument (*newimages = SHINE*), a cell containing the output images should appear in the Matlab workspace. The images are not written to a file. To see one of the output images, use e.g., *figure; imshow(newimages{1})*.

## 4.2 Applying SHINE to a set of pre-loaded images

If the images are already loaded into the workspace, one can call *SHINE(images)* or *newimages = SHINE(images)* to equate them.

The variable *images* has to be a  $1 \times N$  cell ( $N$  = number of to-be-SHINEd images) containing the individual image matrices. The *readImages* function can be used to create such a cell.

Alternatively, it is possible, to read the images individually:

Example:

```
img1 = imread('example1.tif');
img2 = imread('example2.tif');
```

```
images = cell(2,1);  
images{1,1} = img1;  
images{2,1} = img2;  
images = SHINE(images);
```

If *SHINE* was called without output argument (*SHINE(images)*), the equated images should be saved in the output folder (by default, this is the folder *SHINE\_OUTPUT*). The images are saved under the name *SHINED\_imagenumber*.

If *SHINE* was called with an output argument (*newimages = SHINE(images)*), a cell containing the output images should appear in the Matlab workspace. The images are not written to a file. To see one of the output images, use e.g., *figure; imshow(newimages{1})*.

### **4.3 Applying SHINE to a set of pre-loaded images using foreground/background templates**

If one wants to equate the foregrounds and the backgrounds of a set of pre-loaded images separately, *SHINE* can be called with the *images* and *templ* input arguments (*SHINE(images, templ)* or *newimages = SHINE(images, templ)*).

The variable *images* has to be a  $1 \times N$  cell ( $N$  = number of to-be-SHINED images) containing the individual image matrices. The *readImages* function can be used to create such a cell. Alternatively, it is possible, to read the images individually (see example in section 4.2).

The *templ* variable must contain the template(s) for figure-ground segregation. It can be a single matrix (of the same size as the to-be-SHINED pictures) or a cell of  $N$  matrices. Importantly, the background of the templates must be uniform and of a luminance that does NOT occur in the foreground (e.g., it is recommended to use a background of 255 (white) and a foreground of values between 0 and 254).

If *SHINE* was called without output argument (*SHINE(images, templ)*), the equated images should be saved in the output folder (by default, this is the folder *SHINE\_OUTPUT*). The images are saved under the name *SHINED\_imagenumber*.

If *SHINE* was called with an output argument (*newimages = SHINE(images, templ)*), a cell containing the output images should appear in the Matlab workspace. The images are not written to a file. To see one of the output images, use e.g., *figure; imshow(newimages{1})*.

## 5. SHINE options

After calling *SHINE* in the Matlab command window, a number of options will appear. The first choice is between [1=default, 2=custom]. If the default is chosen, SHINE will equate the Fourier spectra followed by the histograms. The matching will be done only once and applied to the whole image. For any other type of matching (e.g., mean luminance adjustment, histogram matching only, Fourier amplitude matching only, or matching of the foregrounds and the backgrounds separately), choose the custom option. If custom is chosen, a number of other options will appear, which will be briefly explained in the following paragraphs.

If you want to **quit** the program during the dialog, **press *Enter* without entering an option**.  
Tipp: If you do not want to go through the custom SHINE options each time, you can easily change the default settings in *SHINE.m*.

After choosing custom, SHINE will ask you whether you would like to equate luminance properties only, spatial frequency content only, or both ([1=luminance, 2=spatial frequency, 3=both]).

### **Option 1: Luminance matching**

If you choose luminance matching, you will be asked whether you would like to scale the mean luminance and standard deviation only, or equate the luminance histograms ([1=lumMatch, 2=histMatch]).

If you choose, *histMatch*, you will see the option to optimize the structural similarity (SSIM; Wang, Bovik, Sheikh, & Simoncelli, 2004) index between the original image and the image with the specified histogram (Avanaki, 2009). Choose 1 to use basic histogram matching (faster) or 2 for maximizing perceptual image quality using SSIM optimization (relatively slow; you can control this by specifying the number of iterations in *histMatch.m*).

For both *lumMatch* and *histMatch*, you will be asked whether the matching step should be applied to the whole image or to the foreground and background separately ([1=whole image, 2=foreground/background]). If you choose 1, the “SHINEing” will begin. If you choose 2, the program will ask you whether the to-be-equated images also serve as templates for foreground segmentation or whether the templates are stored separately in the *SHINE\_TEMPLATE* folder ([1=source images, 2=template(s)]).

In both cases, you will be asked whether you want to specify the background luminance of the templates or have SHINE automatically detect it ([1=specify lum, 2=find automatically (most

frequent lum in the template])). If you choose to specify it, you will be asked to enter an integer between 0 and 255. Note that 0 is black and 255 white. The correct specification of the background is very important, as it will determine which regions in the images are equated. If you choose to have SHINE detect the background luminance automatically, it will use the value that occurs most frequently in the image. That is, SHINE assumes that the background makes up the largest uniform region in the image. After the background luminance of the template is specified, the SHINEing will start.

### Option 2: Spatial frequency matching

If you choose “2=spatial frequency”, the next two options that appear will be [1=sfMatch, 2=specMatch]. *sfMatch* equates the rotational average of the Fourier amplitude spectrum (i.e., the average energy at each spatial frequency). *specMatch* equates the whole amplitude spectra (for each orientation at each spatial frequency). Both *sfMatch* and *specMatch* operate on the whole image.

### Option 3: both

If the third option “3=both” is chosen, SHINE will offer the following options, differing in whether *sfMatch* or *specMatch* are called in combination with *histMatch*, and in which order the equalization steps are performed: [1=hist&sf, 2=hist&spec, 3=sf&hist, 4=spec&hist]. Afterwards, it will ask you to specify the number of iterations. If e.g., the number of iterations is 5, SHINE will perform each of the equalization steps (e.g., *histMatch* and *sfMatch*) 5 times in alternating order. The following dialog will be the same as for *histMatch* alone (see Option 1).

## 6. Overview of the individual matching functions

For a complete overview of the structure of the toolbox, see Willenbockel et al. (in press).

### 6.1 histMatch

The function `newimages = histMatch(images,optim,hist,mask)` equates a set of images in terms of luminance histograms by calling the function `match.m`.

INPUT:

(1) `images`: a cell (1xN or Nx1) that contains N source image matrices

Example 1:

```
images = cell(3,1);  
images{1} = pic1;  
images{2} = pic2;  
images{3} = pic3;
```

Example 2:

```
[images,N] = readImages(pathname,imformat);
```

- (2) *optim* : optional ; if *optim* is set to 0, only basic histogram matching is performed; if it is set to 1, the method by Avanaki (2009) is used to optimize the structural similarity index (SSIM) between the original input image and the image with the specified histogram – thereby improving the perceptual image quality. See *histMatch.m* for specifying the number of iterations and the step size.
- (3) *hist*: optional; target histogram (e.g., *hist=imhist(image)*); if not specified or *hist=[]*, the target will be computed using *avgHist.m*
- (4) *mask*: optional; can be a single matrix (of the same size as the pictures) or a cell of N matrices; *mask* should contain ones where the histograms will be equated (e.g., foreground) and zeros elsewhere

OUTPUT:

- (1) *newimages*: a cell containing the histogram-equated images

## 6.2 lumMatch

The function *images = lumMatch(images,mask,lum)* matches the mean luminance and contrast (i.e., the standard deviation of the luminance distribution) of a set of images.

INPUT:

- (1) *images*: a cell (1xN or Nx1) that contains N source image matrices (see 6.1 for examples)
- (2) *mask*: optional; can be a single matrix (of the same size as the pictures) or a cell of N matrices; *mask* should contain ones where the histograms are obtained (e.g., foreground) and zeros elsewhere. If *images=lumMatch(images)* or *images=lumMatch(images,[],val)*, the whole image is used.
- (3) *lum*: optional; desired mean and standard deviation (e.g., [128 32])  
If not provided, the average mean and std across the input set is used.

OUTPUT:

- (1) *images*: a cell that contains the output images equated in mean luminance and contrast

## 6.3 specMatch

The function *images = specMatch(images,rescaling,tarmag)* matches the Fourier amplitude spectra of a set of images.

INPUT:

- (1) *images*: a cell (1xN or Nx1) that contains N source image matrices (see 6.1 for examples)
- (2) *rescaling*: optional; determines whether the luminance values are rescaled (0=no rescaling, 1=rescale absolute max/min (default), 2=rescale average max/min) after Fourier amplitude specification
- (3) *tarmag*: optional; target spectrum; if not specified, the average spectrum across the input



images is used

OUTPUT:

(1) images: cell containing the images equated in amplitude spectra

#### 6.4 sfMatch

The function `images = sfMatch(images,rescaling,tarmag)` matches the rotational average of the Fourier amplitude spectra (i.e., the average energy at each spatial frequency) of a set of images.

INPUT:

(1) images: a cell (1xN or Nx1) that contains N source image matrices of the same size (see 6.1 for examples)

(2) rescaling: optional; determines whether the luminance values are rescaled after the image modification (0=no rescaling, 1=rescale absolute max/min, 2=rescale average max/min)

(3) tarmag: optional; target spectrum

OUTPUT:

(1) images: cell containing the sf-matched images

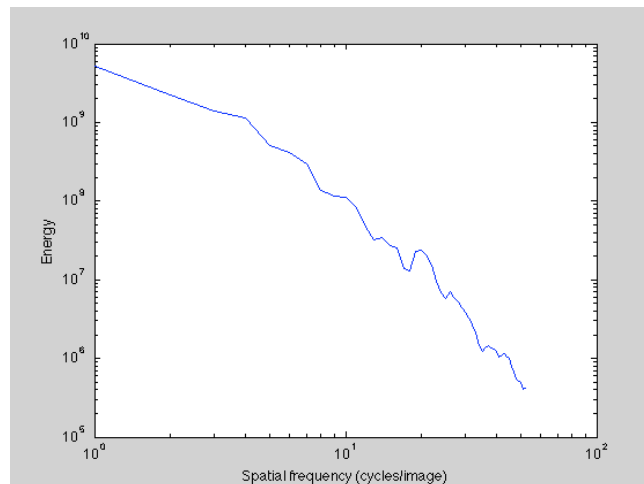
### 7. Overview of the plotting functions

#### 7.1 sfPlot

The function `sfPlot(im)` plots the rotational average of the Fourier energy spectrum. The x-axis of the log-log plot shows the spatial frequency in cycles per image, and the y-axis the energy (arbitrary unit).

INPUT:

im: image matrix [e.g., `im = imread('example.tif');`]

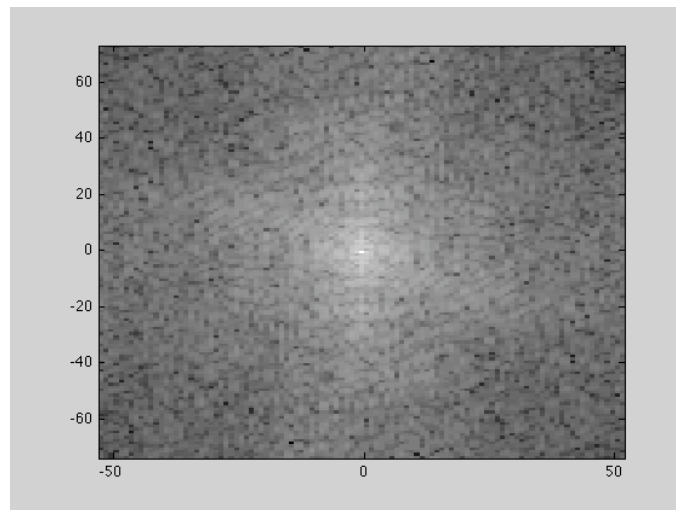


## 7.2 spectrumPlot

The function `spectrumPlot(im)` plots the Fourier amplitude spectrum of an image. The spectrum is displayed as polar plot, where log energy is plotted as a function of spatial frequency (distance from the origin; low-to-high) and orientation (angle).

INPUT:

im: an image matrix (e.g., `im = imread('example.tif');`)



## 8. References

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

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., Sadr, J., Fiset, D., Horne, G. O., Gosselin, F., & Tanaka, J. W. (in press). Controlling low-level image properties: The SHINE toolbox. *Behavior Research Methods*.