

DISCRIM: A Matlab Program for Testing Image Discrimination Models

User's Manual

Michael S. Landy[†]

Department of Psychology & Center for Neural Science
New York University

Discrim is a tool for applying image discrimination models to pairs of images to predict an observer's ability to discriminate the two images. It includes the ability to model both the human observer as well as the sensor and display system that is used to detect and display the image materials (e.g., night vision equipment). A general introduction to the design of *discrim* is described by Landy (2003). It is our hope that display modelers and evaluators will be able to use *discrim* to test the quality of displays and their usefulness for particular visual tasks. We are making the code and documentation freely available to the general public at

<http://www.cns.nyu.edu/~msl/discrim>.

We hope that others will make use of the software, and will let us know what other capabilities would be useful.

Discrim implements a simple sensor and display model consisting of four stages. These are, in order of application, (1) a linear spatial filter, (2) Poisson input noise, (3) a point nonlinearity, and (4) Gaussian output noise.

At present, *discrim* includes a single image discrimination model called the “single filter, uniform masking” (SFUM) model (Ahumada, 1996; Ahumada & Beard, 1996, 1997a,b; Rohaly, Ahumada & Watson, 1997; see <http://vision.arc.nasa.gov/personnel/al/code/filtmod1.htm>). The model includes a contrast sensitivity function as well as a simple model of pattern masking. SFUM was designed to estimate the value of d' for discriminating two given, fixed images. For example, in evaluating a lossy image compression scheme, SFUM will provide an estimate of the ability of an observer to discriminate an original image from its compressed, distorted counterpart. The resulting d' (pronounced “d prime”) value indicates the degree of discriminability. A d' value of zero indicates the two images are completely indiscriminable, so that an observer would be 50% correct (i.e., guessing) on a two-alternative forced-choice task. d' values of 1 and 2 correspond to performance of 76% and 92% correct, respectively.

[†] Address correspondence to: Michael S. Landy, Department of Psychology, New York University, 6 Washington Place, Rm. 961, New York, NY 10003, landy@nyu.edu, (212) 998-7857, fax: (212) 995-4349.

Discrim is written in Matlab, and is invoked (assuming your Matlab path is set appropriately), by typing “discrim” at the Matlab prompt. *Discrim* has the ability to read and save images, and to do some rudimentary image processing to create the images to be discriminated. *Discrim* assumes that images are presented at a reasonably large viewing distance on a CRT or other rectangularly sampled image display device. Image display geometry may thus be specified in units of display pixels, distance across the display device (in cm) or angular distance as viewed by the observer (in deg).

Images and Image Numbers

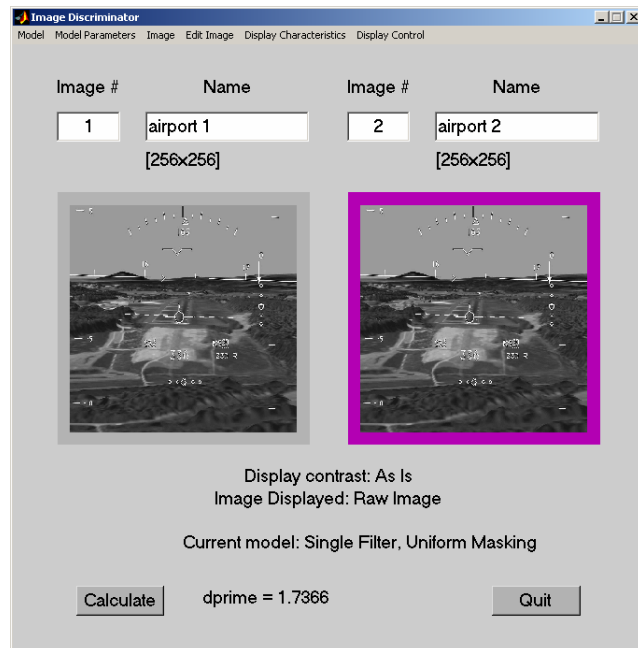
Discrim maintains a library of images. The images in this library are raw input images. That is, they are meant to be descriptions of a scene before that scene is distorted by the various elements of the image sensor and display model (i.e., by blur, noise or nonlinearities). Images may be created by reading in pre-computed images from disk, by combining images arithmetically, and by adding simple targets to images. The program keeps track of images the user has created, referring to these images by their *image number*. These image numbers are positive integers used to identify the images. Users may optionally give images a name as well, which is displayed next to the image number on each of the windows. An image directory may be displayed as well, listing all the images that have been defined. In many cases, when a tool requests an image number and the field is left blank by the user, a new image is allocated instead and given the first available image number.

Images are kept as Matlab arrays in which each pixel value represents image contrast. That is, a pixel value of zero represents a mid-gray pixel, a value of -1 represents a black pixel, and a value of +1 represents a pixel with double the luminance of the mid-gray. When images are imported from or exported to standard 8-bit formats (e.g. JPEG or TIFF), contrast is linearly scaled to pixel value so that a contrast of zero maps to pixel value 128, -1 maps to zero, and +1 maps to 256.

The Main Window

The main window of *discrim* is generally kept visible the entire time one uses this program. This window contains two images. The images that are displayed may be altered by typing in a new image number. If a new number is entered, an image is created (with size determined by the current viewing conditions; see the description of the *Viewing Geometry* window under *Menu: Display Characteristics* below). The image name may be entered here as well.

One of the two images on the main window is designated the



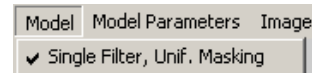
current image. This image is the default for many of the image manipulation tools. The current image is displayed with a magenta border. The user can switch to the other image by clicking on that image. When an image manipulation tool is used, the image that is created or modified will be displayed and made into the current image unless both main window image slots are already occupied by other images. The images that are displayed and the image designated as the current image may also be specified in the image directory window (see *Menu: Image* below). Below the images is an indication of the way the image contrast is displayed and which version of each image is displayed (see *Menu: Display Control* below) as well as the current observer model (see *Menu: Model* below).

The *Calculate* button is used to calculate the perceptual difference, in d' units, between the two displayed images.

The main window has six menus that we describe next. Many of the menu options result in a pop-up window. These windows may be used to perform various image manipulations. In the various image manipulation windows (invoked from the *Image* and *Edit Image* menus), no actions take place until the user hits the *Okay* button and if the action is successfully completed, the window disappears. The *Cancel* button may be used to dismiss the window without any action taking place. However, the items in the windows invoked by the *Model Parameters* and *Display Characteristics* menus are changed the moment they are changed on-screen.

Menu: Model

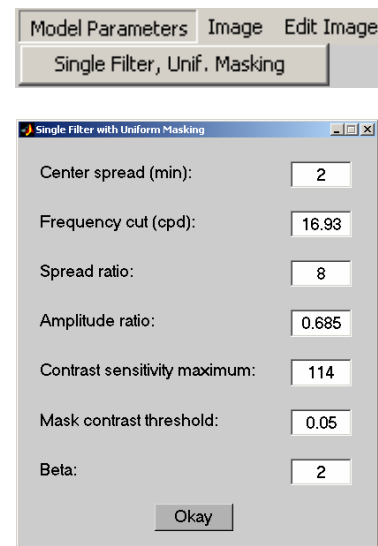
The *Model* menu is used to choose the observer model to be applied to the image pair. For now, the only model that has been implemented is a simple, single-channel SFUM model developed by Al Ahumada. It would be a relatively simple matter to add other models to *discrim* when the time comes.



Menu: Model Parameters

The Model Parameters menu is used to alter the parameters of the discrimination models. There is one item for each available discrimination model.

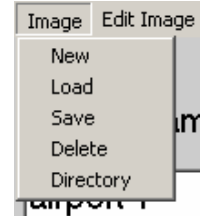
The Single Filter Model with Uniform Masking has seven parameters. This model uses a single channel with a contrast sensitivity function (CSF) modeled as a difference of Gaussians as suggested by Barten. The CSF parameters include the *center spread* (in min arc) and its corresponding high frequency cutoff, the *spread ratio* (the ratio between the two Gaussian sigma parameters) and the *amplitude ratio* of the two Gaussians. The contrast sensitivity specifies the sensitivity at the peak of the CSF. After filtering by the CSF, the pixel differences between the two images are summed with a power of *beta*. The resulting value is then scaled down by a masking factor



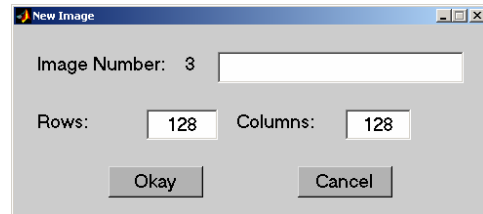
determined by the degree to which the standard deviation of the contrast in the left-hand image exceeds a *mask contrast threshold*.

Menu: Image

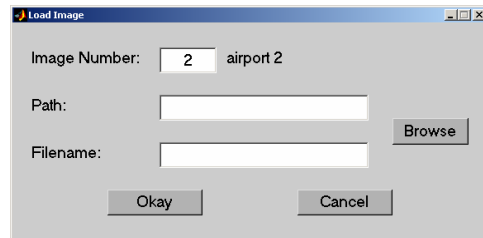
The image menu allows the user to manage the current set of defined images. There are five options in this menu.



New. This option allows the user to define a new image, giving it a name and optionally specifying a nonstandard image size.

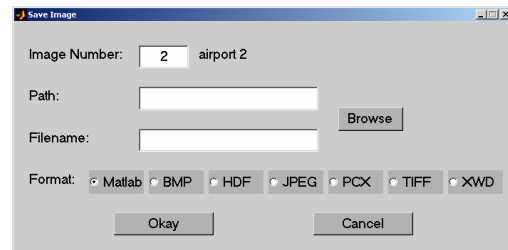


Load. This option allows the user to load in an image from disk. This image will either replace a currently-defined image or will be used to create a new image. A *Browse* button allows the user to search for the appropriate image file. If the filename ends in “.mat”, it is assumed to be a Matlab “save” file that defines an array variable called *discrim_img*. It will complain if this variable is not defined in the file. For all other filenames, the Matlab routine *imread* is used. Thus, images in all of the formats handled by *imread* may be read, including BMP, HDF, JPEG, PCX, TIFF and XWD files. If the image is new or the image name was blank, it will be given the filename as the image name.

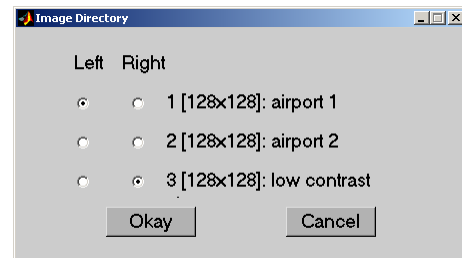


Save. This option allows the user to save an image to disk. The same image types may be saved as can be loaded (see above).

Delete. This option will delete the current image.



Directory. This option opens a window listing all currently-defined images. If this window is kept open while the user works, any changes made by other commands will be registered in this directory window. The directory window includes buttons that allow the user to choose the images displayed in the two display areas on the main window. These choices take place when the user hits the *Okay* button.



Menu: Edit Image

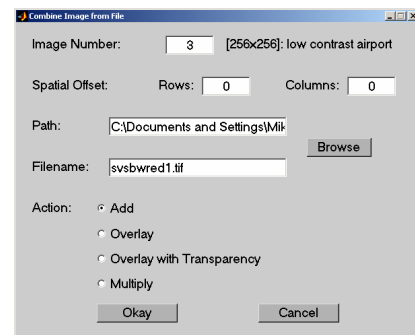
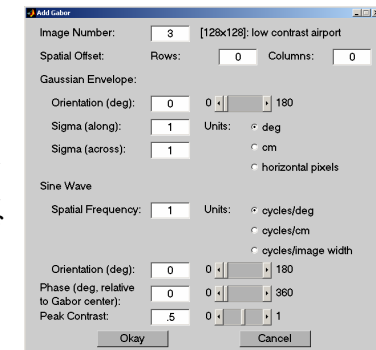
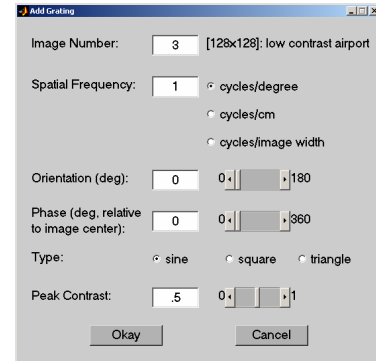
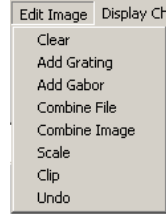
The edit image menu allows the user to modify the contents of the currently-defined images or to create new ones. There are eight options in this menu. With any of the *Add* or *Combine* options, if an image number is specified that is new, it will be created. If no image is specified, one will be created with a new number.

Clear. This option clears the current image, setting all pixel values to zero.

Add Grating. This option adds a grating to an image. The grating may either be a sine wave, a square wave or a triangle wave. The user may specify all of the relevant parameters of the grating.

Add Gabor. This option adds a Gabor patch (a sine wave windowed by a Gaussian envelope) to an image. The envelope may be an arbitrary Gaussian. The user specifies its orientation and the spread parameters along and across that orientation. The sine wave modulator is specified as in the *Add Grating* command. This patch is normally centered in the image, but may be shifted an integral number of pixels horizontally and vertically. (Note that typical image row numbering is used, so a positive row offset shifts the patch downward.)

Combine File. This option combines an image read from disk with a currently-defined image (or creates a new image). If a currently-defined image is specified, the user may specify a spatial offset for the image from the disk file; by default that image is centered. The pixels from the image file may be added to the image pixels, overlaid onto the image pixels (replacing or occluding them), overlaid with transparency (so that image file pixels with a value of zero do *not* replace image pixels) or multiplied. Note that if the image file is larger than the image, or if the spatial offset shifts part of the image file off of the image, those pixels lying outside the image are discarded.

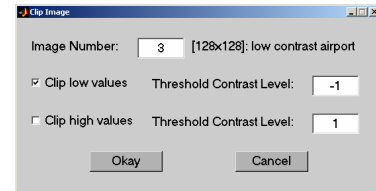
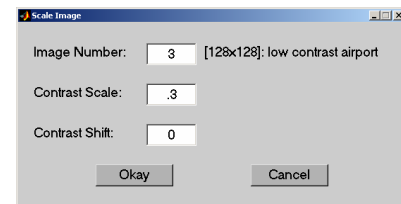
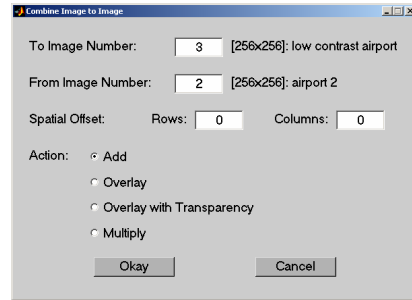


Combine Image. This option combines one currently-defined image (the “From” image) with another currently-defined image (or creates a new image), called the “To” image. The user may specify a spatial offset to shift the From image; by default that image is centered. The pixels from the From image may be added to the To image pixels, overlaid onto the To image pixels (replacing or occluding them), overlaid with transparency (so that From image pixels with a value of zero do *not* replace To image pixels) or multiplied. Note that if the From image is larger than the To image, or if the spatial offset shifts part of the From image off of the To image, those pixels lying outside the To image are discarded.

Scale. This option allows the user to linearly scale a currently-defined image, first multiplying by a contrast scale factor, and then adding a contrast shift amount.

Clip. This option allows the user to threshold an image, clipping values that are too low and/or too high.

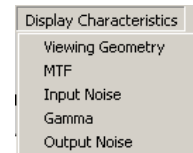
Undo. This option allows the user to undo the most recent image edit, reverting to the previous contents of that edited image.



Menu: Display Characteristics

The display characteristics menu allows the user to specify parameters that relate to the sensor and display model. There are five options, each of which invokes a pop-up window. The first four options correspond to the four stages of the display model, and the fifth describes the viewing geometry.

Viewing Geometry. The viewing geometry window specifies the viewing distance to the simulated CRT, the pixel sampling and the image size. The screen-related variables may be specified in units of visual angle (deg), distance (cm) or pixels. Since these quantities are all interrelated, changing any one variable is likely to change others. And, which others are changed depends on which variables one chooses to hold constant. In most cases, the viewing distance and the number of image pixels are held constant. But, it is a good idea to check all the variables once you've made changes to ensure that you have things set the way you want them. When the number of image rows or columns is changed, this has consequences for other things such as how images are displayed, the interpretation of the MTF, and the model calculations. The Viewing Conditions window



assumes that both images are square (so that changes to horizontal size also change the vertical size and vice versa) with square pixel sampling (so that changes to horizontal pixel sampling density change the vertical pixel sampling density and vice versa). Check boxes are supplied to allow the user to turn off this behavior.

MTF (modulation transfer function). The input spatial filter may be used to mimic the effects of the optics of the image sensor. The *MTF* window gives four choices of filter type: Flat (no filtering), Gaussian, Difference-of-Gaussians (DoG) and Empirical. The Gaussian and Empirical filters are Cartesian-separable. Very loosely speaking, Cartesian-separable means that the filter may be specified as a product of a filter applied to the horizontal frequencies multiplied by one applied to the vertical frequencies. For the DoG filter, each of the constituent Gaussian filters is Cartesian-separable. For the Gaussian and DoG filters, the user may specify whether the horizontal (x) and vertical (y) filters are identical (in which case the parameters of only the horizontal filter are specified) or not. The Gaussians are specified by the standard deviation (*sigma*) in either the spatial or spatial frequency domain in a variety of units. The DoG filter consists of a Gaussian with a high cut-off frequency minus a second Gaussian with a lower cut-off frequency. The user specifies the relative amplitude of the two filters (the *peak ratio*), a value which must lie between zero and one. The DoG filter is scaled to have a peak value of one.

For the empirical filter, the user specifies the location of a text file that describes the filter. The format of the file is simple. Each line contains either two or three values. The first is the spatial frequency in cycles/deg. The second is the horizontal MTF value, which must lie between zero and one. If all entries in the file have only two values, then the program assumes the vertical and horizontal filters are identical. Otherwise, all lines should contain three values, and the third value is the vertical MTF value. The supplied MTF is interpolated and/or extrapolated as needed using the Matlab *interp1* spline interpolation routine. The file should be sorted in ascending order of spatial frequency.

The MTFs are graphed in the bottom half of the MTF window.

The MTF is applied in the Fourier domain. That is, the input images (clipped to the size of the display window and/or extended with zeroes to that size) are Fourier

Viewing Geometry

Viewing distance (cm): 100

Horizontal pixels/deg: 30 ☒ Square pixels

Vertical pixels/deg: 30

Horizontal pixels/cm: 17.19

Vertical pixels/cm: 17.19

Number of image columns: 128 ☒ Square image

Number of image rows: 128

Horizontal image size (cm): 7.447

Vertical image size (cm): 7.447

Horizontal image size (deg): 4.267

Vertical image size (deg): 4.267

Okay

MTF

☐ Flat ☒ x MTF = y MTF

☐ Gaussian Sigma: 15 15 ☐ cycles/degree ☐ degrees

Sigma 2: 5 5 ☐ cycles/cm ☐ cm

☒ DoG Peak Ratio: 0.9 ☐ cycles/image width ☐ image widths

Path:

☐ Empirical Filename:

Modulation Transfer

Spatial Frequency

Okay

transformed. The transforms are multiplied by the MTF and then inverse transformed. This is performed using a discrete Fourier transform (DFT), and hence is subject to the usual edge effects of the DFT. These edge effects can be ameliorated by using a windowing function (e.g., use the *Add Gabor* with zero spatial frequency to create a Gaussian image, and use *Combine Image* to multiply images with this Gaussian window).

Input Noise. The user may specify, or have the program compute, the mean quantum catch of the individual sensor pixels. When that quantum catch is low, as it must be in the low-light conditions for which night vision equipment is designed, the effects of the Poisson statistics of light become important. To compute the mean quantum catch, the user specifies the mean luminance, the integration time, the dominant wavelength of the light, the quantum efficiency of the individual pixels (i.e., the percent of the photons incident on the square area associated with the pixel that are caught), and the sensor aperture diameter. The calculation, based on the formula in Wyszecki and Stiles (1982), only makes sense if the image consists primarily of visible light. Thus, for infrared-sensitive equipment (dominant wavelengths above 700 nm), the user must supply the mean quantum catch.

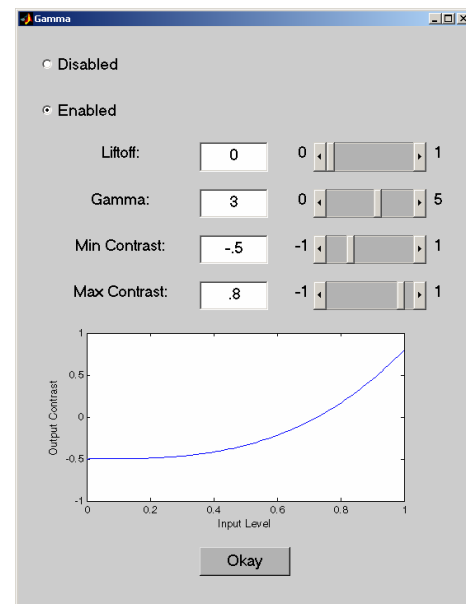
The input images are specified in terms of image contrast. Input image pixels are linearly mapped to expected quantum catches so that a contrast of -1 corresponds to an expected catch of zero, and a contrast of zero corresponds to the mean expected quantum catch.

Gamma. The user may specify the nonlinearity applied to individual pixels. Typically, both image sensors (film, vidicons, etc.) and image displays (CRTs, in particular) are characterized by a so-called “gamma curve”. *Discrim* includes a single nonlinearity in its sensor and display model. One can think of it as a lumped model that combines both the sensor and display nonlinearities. We implement a generalization of the gamma curve by allowing for an input level below which no output occurs (which we call “liftoff”), and a minimum and maximum output contrast (e.g., due to a veiling illumination on the display). Thus, the output of the nonlinearity is

$$y = C_{min} + (C_{max} - C_{min}) \left[x - x_0 \right]^\gamma.$$

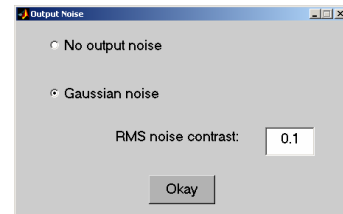
x is the input level (a number between 0 and 1). x_0

is the liftoff level. γ is the degree of nonlinearity (2.3-3 is a typical range of values for a CRT system, and such values are often built into devices, such as DLP projectors, which



don't have an implicit pixel nonlinearity). C_{min} and C_{max} are the minimum and maximum output contrast values. y is the resulting output contrast (a number between -1 and 1). The resulting gamma curve is plotted in the lower half of the window.

Output Noise. The next and final stage of the display model is output noise. The model allows for Gaussian noise to be added to the displayed pixels after the point nonlinearity has been applied. This may be used to model imperfections in the display device, but can be used for other modeling applications as well (e.g., models of medical imaging devices). The user specifies the root-mean-square (RMS) noise contrast, i.e., the standard deviation of the noise in units of image contrast.



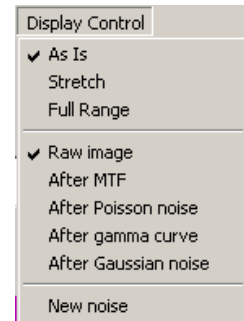
Menu: Display Control

The *Display Control* menu has three sets of items. The first three items relate to how the image contrast is scaled prior to display on the main window. The next five items are used to select what form of the images are displayed. And, the last item allows the user to display a new set of noise samples.

By default, images are displayed “As Is”, meaning that an image contrast of -1 is displayed as black, and +1 as white. The *Display Control* menu allows the user to either stretch the contrast (as much as possible, leaving 0 as mid-gray) or to linearly scale displayed contrast so as to use the *full range* of displayable contrasts (mapping the lowest pixel value to black and the highest to white). Note that the *full range* setting effectively undoes the liftoff value of the gamma function, at least as far as the appearance of the displayed images is concerned.

By default, the images displayed on the main window are the input images, unmodified by the various distortions in the image display model. However, using the *Display Control* menu, the user may select for display the image after each stage of the image display model: after the MTF has been applied, after the Poisson noise has been added, after the gamma curve nonlinearity, or after the Gaussian output noise has been added. The two images each have their own separate samples of Poisson noise (since Poisson noise variance depends on image content). On the other hand, the same Gaussian output noise sample is used for both displayed images.

Finally, if the user specifies *New Noise*, new samples of Poisson and/or Gaussian noise are computed, changing the displayed images if the appropriate image display selection has been specified.



Working with Images of Different Sizes

Discrim has the ability to work with images varying in size. The set of images in the current directory can include images with different numbers of rows and/or columns. Throughout, it is assumed that the pixel sampling is identical. So, for example, when images are added, the pixels are added one at a time, in order. Images are also assumed to

be centered on the display and with respect to each other (although spatial offsets are included with the Add Gabor, Add File and Add Image commands).

For display and image discriminability calculations, however, *discrim* has a fixed image size as specified in the *Viewing Geometry* window. The larger of the two image display dimensions (the larger of the number of horizontal and vertical pixels specified in the *Viewing Geometry*) controls the scaling of images in the display areas of the main window. Thus, if the image size is changed in the *Viewing Geometry* window, displayed images may become smaller or larger. When images are displayed, they are shown centered in the corresponding window, with a purple or gray border outside of the defined image area. If an image is larger than the size specified by the current Viewing Conditions, pixels beyond that size will fall outside the image display window and will not be visible.

When the user hits the Calculate button, the two displayed images are compared using the current image discrimination model. However, if an image is smaller than the current *Viewing Geometry* image size, it will be extended with zeros. If an image is too large, the pixels that lie outside the display window will not be used in the discriminability calculation.

The SFUM Model Implementation

The *SFUM* model was designed to estimate the discriminability of a pair of fixed images. The display model involves two possible sources of noise: input Poisson noise and output Gaussian noise. When either or both of those noise sources are enabled, the intent of the *discrim* program is to allow the user to estimate the discriminability of the two input images under conditions of stochastic variability due to the noise source(s) (and other image distortions). That is, on any given trial an observer will see a different retinal image due to the variability of the noise from trial to trial. If we simply added different, independent Poisson and/or Gaussian noise samples to each of the two images and then applied the SFUM model, SFUM would attempt to estimate discriminability not only of the underlying images, but of the two noise samples as well. Clearly, this is not appropriate. What is of interest is the observer's ability to discriminate the underlying scenes despite the noise, not their ability to discriminate the noise samples.

Thus, the SFUM model is not well-suited to the problem at hand. However, we have implemented the SFUM model in a way that should allow it to provide reasonable estimates. We do this by using the same sample of noise, Poisson and/or Gaussian, for both images. Thus, differences between the two noise samples are not there to inflate the d' estimates. Gaussian noise is an additive process that is independent of the image content. It is a simple matter to generate a Gaussian noise image, and add it to both input images. On the other hand, Poisson noise depends on the image content. The variance of the noise added to any given pixel is equal to the value of that pixel. This means that the use of the same noise image for both input images is not an accurate reflection of Poisson statistics. We have settled on an approximation that we feel is adequate for the sorts of threshold detection tasks for which *discrim* is most appropriate. When Poisson noise is used with the SFUM model, the two input images are first blurred using the current MTF. Then, the image in the left-hand window is subjected to Poisson noise. The difference between the noisy image and the blurred left-hand image (the *error image*) is treated as

an additive noise source. That error image is then added to the individual blurred images to simulate a Poisson noise source that perturbs both images identically. The reason the left-hand image is used to generate the Poisson noise is due to an asymmetry inherent in the SFUM model. The SFUM model treats the left-hand image as a “mask”, and the difference between the two images as a “signal”. As long as the “mask” (or background) is kept constant, then d' is proportional to the strength of the signal. Thus, to determine the strength of signal required to produce a d' of 1.0, for example, one need only divide the current signal strength by the current value of d' .

Note that each time *discrim* calculates a d' value, new samples of Poisson and/or Gaussian noise are used. Thus, the user can average over several such calculations to guard against an outlier value due to an atypical noise sample. Also note that the noise samples used to calculate the SFUM model are not the same samples as are used for image display in the main window. Finally, note that the images are not clipped prior to applying the SFUM model, so that some pixels may have contrast values above 1 or below -1 (which is, of course, not a displayable contrast).

Additional Observer Models

The *discrim* program is set up so as to allow for the addition of additional vision models. In particular, there is a large literature (mostly from the medical imaging community) of visual detection and discrimination models for visual targets in patterned and noisy backgrounds (Barrett, Yao, Rolland & Myers, 1993; Bochud, Abbey & Eckstein, 2000; Burgess, 1999; Burgess, Li & Abbey, 1997; Eckstein et al., 2003; King, de Vries & Soares, 1997; Myers et al., 1985; Rolland & Barrett, 1992; Wagner & Weaver, 1972; for reviews see Eckstein, Abbey & Bochud, 2000; Wagner & Brown, 1985). These models provide an estimate of d' given the input images and descriptions of the noise (variance, spatial correlation, etc.). Thus, for these models, *discrim* is already set up to provide the required information, and the issue of using identical noise samples for the two input images shouldn't arise. It would be a relatively simple task to add such models to the *discrim* model palette.

References

Ahumada, A. J., Jr. (1996). Simplified vision models for image quality assessment. In J. Morreale (Ed.), *SID International Symposium Digest of Technical Papers*, 27, 397-400. Santa Ana, CA: Society for Information Display.

Ahumada, A. J., Jr. & Beard, B. L. (1996). Object detection in a noisy scene. In B. E. Rogowitz & J. Allebach (Eds.), *Human Vision, Visual Processing, and Digital Display VII*, 2657, 190-199. Bellingham, WA: SPIE.

Ahumada, A. J., Jr. & Beard, B. L. (1997a). Image discrimination models predict detection in fixed but not random noise. *Journal of the Optical Society of America A*, 14, 2471-2476.

Ahumada, A. J., Jr. & Beard, B. L. (1997b). Image discrimination models: Detection in fixed and random noise. In B. E. Rogowitz & T. N. Pappas (Eds.), *Human Vision, Visual Processing, and Digital Display VIII*, 3016, 34-43. Bellingham, WA: SPIE.

Barrett, H. H., Yao, J., Rolland, J.P. & Myers, K. J. (1993). Model observers for assessment of image quality. *Proceedings of the National Academy of Sciences USA*, 90, 9758- 9765.

Bochud, F. O., Abbey, C. A. & Eckstein, M. P. (2000). Visual signal detection in structured backgrounds III, Calculation of figures of merit for model observers in non-stationary backgrounds. *Journal of the Optical Society of America A*, 17, 193-205.

Burgess, A. E. (1999). Visual signal detection with two-component noise: low-pass spectrum effect. *Journal of the Optical Society of America A*, 16, 694-704.

Burgess, A. E., Li, X. & Abbey, C. K. (1997). Visual signal detectability with two noise components: anomalous masking effects. *Journal of the Optical Society of America A*, 14, 2420-2442.

Eckstein, M. P., Abbey, C. K. & Bochud, F. O. (2000). A practical guide to model observers for visual detection in synthetic and natural noisy images. In J. Beutel, H. L. Kundel & R. L. van Metter (Eds.), *Handbook of Medical Imaging, Vol. 1, Physics and Psychophysics* (pp. 593-628). Bellingham, WA: SPIE Press.

Eckstein, M. P., Bartroff, J. L., Abbey, C. K., Whiting, J. S., Bochud, F. O. (2003). Automated computer evaluation and optimization of image compression of x-ray coronary angiograms for signal known exactly tasks. *Optics Express*, 11, 460-475. <http://www.opticsexpress.org/abstract.cfm?URI=OPEX-11-5-460>

King, M. A., de Vries, D. J. & Soares, E. J. (1997). Comparison of the channelized Hotelling and human observers for lesion detection in hepatic SPECT imaging. *Proc. SPIE Image Perc.*, 3036, 14-20.

Landy, M. S. (2003). A tool for determining image discriminability. <http://www.cns.nyu.edu/~msl/discrim/discrimpaper.pdf>.

Myers, K. J., Barrett, H. H., Borgstrom, M. C., Patton, D. D. & Seeley, G. W. (1985). Effect of noise correlation on detectability of disk signals in medical imaging. *Journal of the Optical Society of America A*, 2, 1752-1759.

Rohaly, A. M., Ahumada, A. J., Jr. & Watson, A. B. (1997). Object detection in natural backgrounds predicted by discrimination performance and models. *Vision Research*, 37, 3225-3235.

Rolland, J. P. & Barrett, H. H. (1992). Effect of random inhomogeneity on observer detection performance. *Journal of the Optical Society of America A*, 9, 649-658.

Wagner, R. F. & Brown, D. G. (1985). Unified SNR analysis of medical imaging systems. *Phys. Med. Biol.*, 30, 489-518.

Wagner, R. F. & Weaver, K. E. (1972). An assortment of image quality indices for radiographic film-screen combinations – can they be resolved? *Appl. of Opt. Instr. in Medicine, Proc. SPIE*, 35, 83-94.

Wyszecki, G. & Stiles, W. S. (1982). *Color Science: Concepts and Methods, Quantitative Data and Formulae* (2nd Ed.). New York: Wiley.