MixImages Script

Alex Woronow, 2021 Version 0.1.1 Released 02.19.21

Contents

[Purpose: 1](#_Toc64878212)

[Implementation: 1](#_Toc64878213)

[Discussion: 2](#_Toc64878214)

[Why use Mixing Equations to ReStar and Image? 2](#_Toc64878215)

[Acquiring the Script: 2](#_Toc64878216)

[Installation: 3](#_Toc64878217)

[Usage: 3](#_Toc64878218)

[Iterating: 3](#_Toc64878219)

[The Blue Triangle: 3](#_Toc64878220)

[Quirks and Foibles: 3](#_Toc64878221)

[Reporting Problems & Submitting Suggestions: 3](#_Toc64878222)

[Future Possibilities: 3](#_Toc64878223)

[Known Bugs: 3](#_Toc64878224)

# Purpose:

This script offers three different equations that function to mix the contents of one image, the “Target Image” with a second image, the “Source Image.” Three of the mixing equations utilize a mask during the mixing, and two equations do not. Also, the mixing ratios by two of the equations have a dependency upon the relative brightness of the Target and Source images.

One of the principal uses for these mixing equations is the reintroduction of stars into starless images. This write-up focuses on that task.

# Implementation:

Internally, PixelMath applies the selected equation to the Target and Source images. The images are as follows:

1. MixedImage = K\*Image\_1 + ~K\*Image\_2
2. MixedImage = max( KTarget\* Target, Mask\*KSource\* Source )
3. MixedImage = ~Mask\*KTarget\*Target + Mask\*KSource\*Source
4. MixedImage = max( KTarget\*Target, KSource\*Source )
5. MixedImage = Mask\*(K1\*Image\_1 +~K1\*Image\_2) + ~Mask\*(K2+\*Image\_2+~K2\*Image\_1)

The values “K…” indicate constants that multiply each pixel value in their respective images. The symbole ~ indicates the inverse of a value or of pixel values. PixelMath defines the inverse of a value, v, as (1-v).

Equation 0: A basic mixing equation where K [0,1] is the mixing coefficient that multiplies the first image and 1-K weights the second image.

Equation 1: The Target is unmasked, and it is and mixed with a weighted proportion of a second, masked image (Source). The greater of the two adjusted image contributions form the output image. Say the Target image had stars that you wanted to replace with those from the Source image, or was starless and you wanted to insert stars from the Source image into it. The stars-transfer might be accomplished using a star mask and a properly adjusted mixing coefficient. Most likely, not all of the fainter stars will be transferred, depending the value of KSource. An important feature is that the transferred stars replace the background they cover in the target image and are not added to it.

Equation 2: This equation functions similarly to Equation 1, except the transfer is not predicated on which of the two values is greatest. If this equation is used for star transfer, if the mask is not a binary mask, the stars will be added to the background of the Target image.

Equation 3: This equation also functions similarly to Equation 1 except, this time, no mask comes into play.

Equation 4: “Now for something completely different.” Mix Mask-Concealed regions differently from Revealed regions according to the two mixing coefficients, K1 and K2, respectively. This equation could be used with a Range Mask to mix bright and dark areas of two images, emphasizing the brighter areas from one image and the darker areas from the other, but blending both the bright and dark areas to specified (but different) degrees.

The output image (MixedImage) automatically rescales to the interval [0,1] unless the “Rescale Result” checkbox is unchecked. Also, if the equation selected includes the use of a mask, but the mask is left blank, a value of 0.50 substitues for the mask, effectively nullifying the influence of the mask.

Equations 2 and 4 are good starting points for reintroducting stars into an image. Be aware that Equation 4 should have the image containing the starts in the first selection box and the starless image in the second selection box. For Equation 2 that order is reversed.

# Discussion:

This script can be used for many image-mixing requirements. But this script was targeted at re-introducing stars into a starless image. I have found that, under different circumstances, one or another of the three equations does a better job at this; the reason for this variability currently eludes me, however.

## Why use Mixing Equations to ReStar and Image?

Probably the most common method to separate stars from an image is to apply StarNet++. With it, you can even produce an image containing “only” the stars. An identical star-only image can be obtained by subtracting the starless image from the source image. In either case, the stars do not have the correct color. This problem arises because the starless image has had a colored background substituted where none existed—where the stars were removed. So the subtraction does not render the separated stars in either the correct color or intensity. Also, StarNet can leave artifacts in the starless image and consequently in the subtracted stars-only image. These artifacts include crosshatches in the stars, dark/light spikes where bright stars were, and removal, at least in part, of areas external to the stars. And these artifacts get transmitted back into images that are reconstructed by the starless+stars arithmetic. Additionally, if the starless image is modified (as it surely will be), adding the stars back, arithmetically, compounds the errors by adding the stars to an altered color and intensity background.

There’s another approach that we can take in reuniting stars and starless images: Enter MixImages. The idea is to transfer the stars from an image wherein the stars have their true colors, such as an RGB Photometrically Color Calibrated and stretched image. A star mask could assist in this, or an intensity-dependent transfer might work, or maybe some combination of both. Different approaches seem to work in different circumstances.

# Acquiring the Script:

The current version of the script and documentation are available by downloading the folder at this link:

https://www.dropbox.com/sh/dpd1zs5fl576ohh/AACtBkC16yB2EiqRuhBDmfxga?dl=0

# Installation:

Installation procedures follow standard PixInsight protocols. Place the download folder in an accessible and stable location; open the Scripts tab in PixInsight; select Feature Scripts; select Add; Point the function to the downloaded folder and press the Done button when PixInsight completes its task.

# Usage:

The user interface should be self-explanatory, along with the tooltips provided for each user input. If one of the mixing equations does not accomplish the result you wanted, consider that other adjustments outside of the script may improve the rendering. Or, try a different mixing equation within the script.

## Iterating:

The script can be iterated (repeated without restarting the script) by checking the box at the lower right.

## The Blue Triangle:

An icon instance of the script can be dragged onto the desktop by dragging the blue triangle to the desktop.

# Quirks and Foibles:

If you work with linear images, the script does not automatically apply a screen stretch (STF) to the new image.

# Reporting Problems & Submitting Suggestions:

Please let me know of any bug you uncover; contact me at [Alex@FaintLightPhotography.com](mailto:Alex@FaintLightPhotography.com). Include a description of the problem and any error messages generated. Also, describe the data and procedures that led to the error.

# Future Possibilities:

None identified.

# Known Bugs:

None identified.