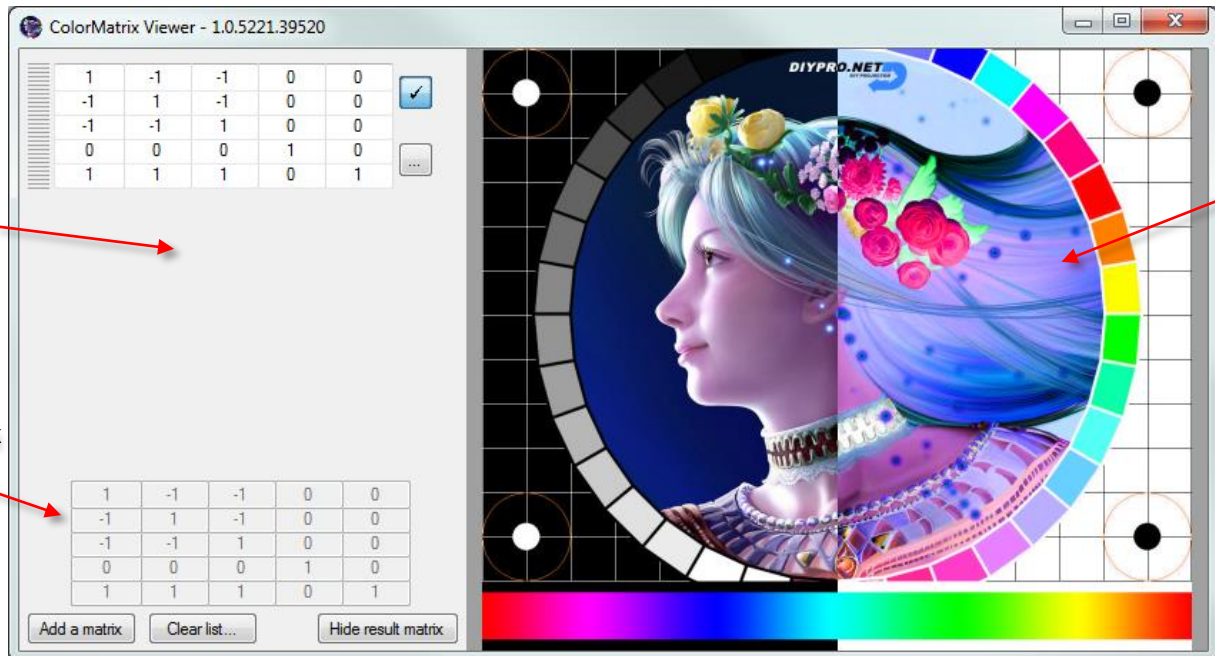


COLORMATRIX VIEWER – USER DOCUMENTATION

Melvyn Laïly – arcanesanctum.net

Table of contents

Image Viewer	2
Matrix List	2
Result Matrix	3
About Color Matrices	4
Basics	4
Pragmatic information and examples	5



Matrix List

Result Matrix

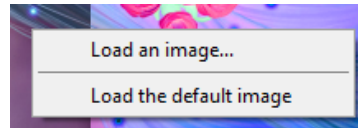
Image Viewer

Image Viewer

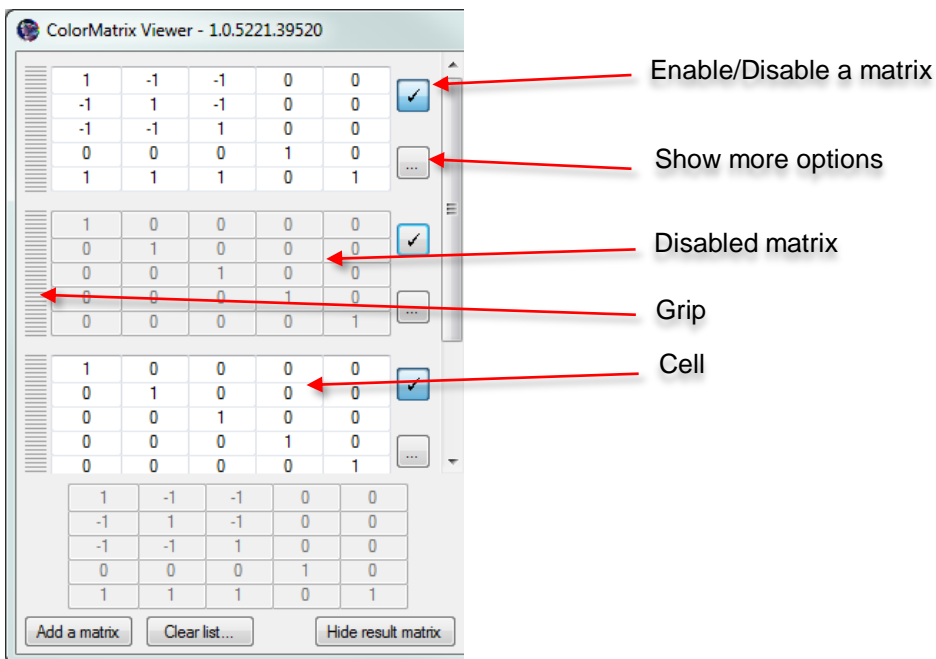
The image viewer allows you to view in real-time the effects of the matrices you are editing.

Click anywhere in the frame to move the separator between the initial image (left) and the resulting image (right).

Right click anywhere in the frame to display a context menu allowing you to load an image, or to reset to the default image.



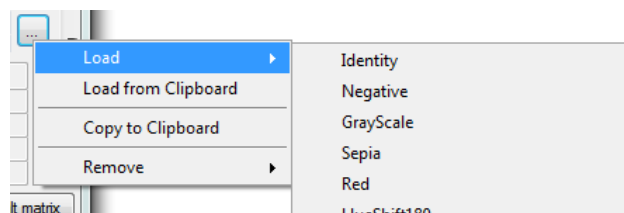
Matrix List



The Matrix List contains a scrollable list of matrices.

To edit a matrix, you can directly type into a cell, or maintain `Shift` and scroll to increment/decrement a value by 1, or maintain `Ctrl` and scroll to increment/decrement by 0.1

You can also load a preset by clicking on the option button [...] and choosing the "load" item.



You can temporarily disable or enable a matrix by clicking the [✓] button.

You can reorganize matrices by drag and dropping them in the list. To do so, use the grip.

Result Matrix

The Result Matrix shows the combination of all the matrices in the list.

As this is done by multiplying the matrices, their order is important and can impact the result.

This matrix is the one actually applied to the image.

1	-1	-1	0	0
-1	1	-1	0	0
-1	-1	1	0	0
0	0	0	1	0
1	1	1	0	1

You can easily export the result matrix by right clicking and selecting "Copy to Clipboard".

You may then paste it to a text file...

About Color Matrices

Basics

Quoting from <http://msdn.microsoft.com/en-us/library/windows/desktop/ms533875%28v=vs.85%29.aspx> :

Bitmap objects store the color of each pixel as a 32-bit number: 8 bits each for red, green, blue, and alpha. Each of the four components is a number from 0 through 255, with 0 representing no intensity and 255 representing full intensity. The alpha component specifies the transparency of the color: 0 is fully transparent, and 255 is fully opaque.

A color vector is a 4-tuple of the form (red, green, blue, alpha). For example, the color vector (0, 255, 0, 255) represents an opaque color that has no red or blue, but has green at full intensity.

Another convention for representing colors uses the number 1 for maximum intensity and the number 0 for minimum intensity. Using that convention, the color described in the preceding paragraph would be represented by the vector (0, 1, 0, 1).

You can apply linear transformations (rotation, scaling, and the like) to color vectors by multiplying by a 4 x 4 matrix. However, you cannot use a 4 x 4 matrix to perform a translation (nonlinear).

If you add a dummy fifth coordinate (for example, the number 1) to each of the color vectors, you can use a 5 x 5 matrix to apply any combination of linear transformations and translations.

A transformation consisting of a linear transformation followed by a translation is called an affine transformation. A 5 x 5 matrix that represents an affine transformation is called a homogeneous matrix for a 4-space transformation.

The element in the fifth row and fifth column of a 5 x 5 homogeneous matrix must be 1, and all of the other entries in the fifth column must be 0.

Example: suppose you want to start with the color (0.2, 0.0, 0.4, 1.0) and apply the following transformations:

1. Double the red component
2. Add 0.2 to the red, green, and blue components

The following matrix multiplication will perform the pair of transformations in the order listed.

$$\begin{bmatrix} 0.2 & 0.0 & 0.4 & 1.0 & 1.0 \end{bmatrix} \begin{bmatrix} 2 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0.2 & 0.2 & 0.2 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 0.6 & 0.2 & 0.6 & 1.0 & 1.0 \end{bmatrix}$$

Applying a Color Matrix to a whole image is just the generalization of applying the matrix to a single color (a single pixel).

The 5 ×5 identity matrix (shown in the following illustration) has 1s on the diagonal and 0s everywhere else. If you multiply a color vector by the identity matrix, the color vector does not change. A convenient way to form the matrix of a color transformation is to start with the identity matrix and make a small change that produces the desired transformation.

$$\begin{bmatrix} 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Identity Matrix

For more information about matrix representation of transformations: <http://msdn.microsoft.com/en-us/library/windows/desktop/ms536397%28v=vs.85%29.aspx>

For information about translating, scaling, rotating, or shearing color vectors, see <http://msdn.microsoft.com/en-us/library/windows/desktop/ms533809%28v=vs.85%29.aspx>

Pragmatic information and usage examples

Source: <http://docs.rainmeter.net/tips/colormatrix-guide>

```
ColorMatrix1=1; 0; 0; 0; 0  
ColorMatrix2=0; 1; 0; 0; 0  
ColorMatrix3=0; 0; 1; 0; 0  
ColorMatrix4=0; 0; 0; 1; 0  
ColorMatrix5=0; 0; 0; 0; 1
```

The values on the main diagonal are, from top-left to bottom-right: Red, Green, Blue, Alpha and a placeholder.

The values represent the percentage of the particular value present in the image, where 0.0 is none and 1.0 is normal.

The remaining entries in the matrix allow a color to have its value modified in terms of another color (ex: the value of Red might have half of the Blue value added), with the entries in the final row (ColorMatrix5) determining offset values that are added directly to the color. (ex: ColorMatrix5=0.5; 0; 0; 0; 1 will add 50% to the red value)

