
RGB Image Color Space Transformations

Release 0.01

Robert Tamburo¹

December 10, 2010

¹robert.tamburo@gmail.com

Abstract

This paper describes a set of pixel accessors that transform RGB pixel values to a different color space. Accessors for the HSI, XYZ, Yuv, YUV, HSV, Lab, Luv, HSL, CMY, and CMYK color spaces are provided here. This paper is accompanied with source code for the pixel accessors and test, test images and parameters, and expected output images.

Latest version available at the [Insight Journal](http://hdl.handle.net/10380/3237) [<http://hdl.handle.net/10380/3237>]
Distributed under [Creative Commons Attribution License](#)

Contents

1	Introduction	1
2	Implementation of Color Space Transforms	2
3	Example	2
4	Testing	2
5	Software Used	5

1 Introduction

When working with RGB images it is sometimes convenient to map the images to a different color space. ITK currently provides a means to calculate luminance from RGB images, but no classes exist for mapping to other color spaces. Described in this paper are a collection of pixel accessors for mapping RGB images to other color spaces.

2 Implementation of Color Space Transforms

Color space transforms for RGB images are performed via pixel accessors. A separate pixel accessor exists for each color space covered in this paper. The pixel accessors are templated over input value type and output value type. Internally, the input pixel type is an RGB pixel and the output pixel type is a vector. Any of the color transform pixel accessors can be used in conjunction with an `itk::ImageAdaptor` and `itk::VectorIndexSelectionCastImageFilter` to obtain the computed components.

3 Example

Example use of a pixel accessor is shown below. The pixel accessor is declared:

```
typedef itk::RGBPixel<unsigned char> RGBPixelType;
typedef itk::Image<RGBPixelType, 2> RGBImageType;
typedef itk::Accessor::RGBToHSIColorSpacePixelAccessor
    <RGBPixelType, float> RGBToHSIColorSpaceAccessorType;
```

The accessor is hooked into an image adaptor:

```
typedef itk::ImageAdaptor<RGBImageType, RGBToHSIColorSpaceAccessorType>
    RGBToHSIAdaptorType;
RGBToHSIAdaptorType::Pointer rgbToHSIAdaptor = RGBToHSIAdaptorType::New();
rgbToHSIAdaptor->SetImage(rgbImage); // the RGB image
```

The color space images are extracted:

```
typedef itk::Image<float, 2> FloatImageType;
typedef itk::VectorIndexSelectionCastImageFilter<RGBToHSIAdaptorType, FloatImageType>
    VectorCastFilterType;
VectorCastFilterType::Pointer vectorCastFilter = VectorCastFilterType::New();
vectorCastFilter->SetInput(rgbToHSIAdaptor);
vectorCastFilter->SetIndex(0); // Get the Hue image
...
vectorCastFilter->SetIndex(1); // Get the Saturation image
...
vectorCastFilter->SetIndex(2); // Get the Intensity image
...
```

The above code was used with `peppers.png` (Fig. 1) from the Data directory as the input image. The output images are shown in Figures 2, 3, and 4 (the output of `VectorIndexSelectionCastImageFilter` was rescaled to unsigned char pixels with a minimum of 0 and maximum of 255).

4 Testing

The test code included with this article will transform `peppers.png` to other color spaces with the following arguments, where `COLORSPACE` is HSI, XYZ, Yuv, YUV, HSV, Lab, Luv, HSL, CMY, or CMYK.



Figure 1: The input image `peppers.png` for this example.

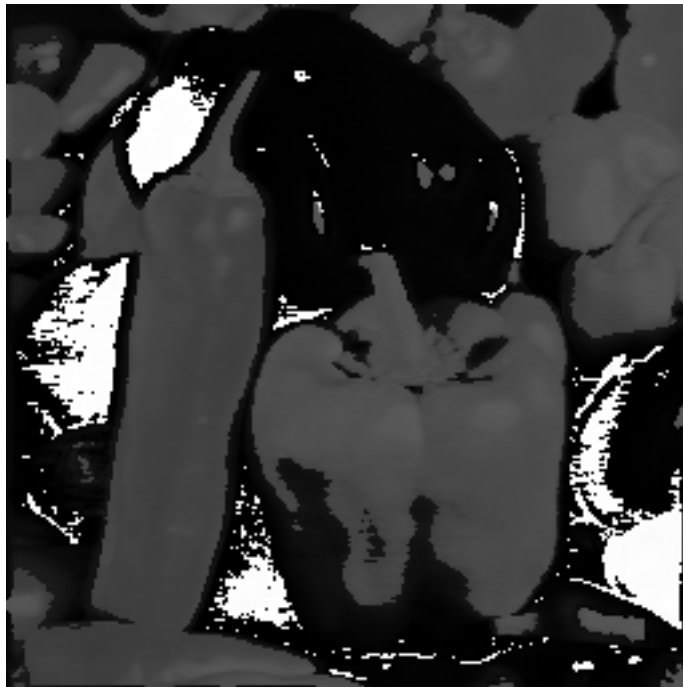


Figure 2: The hue of the input image.

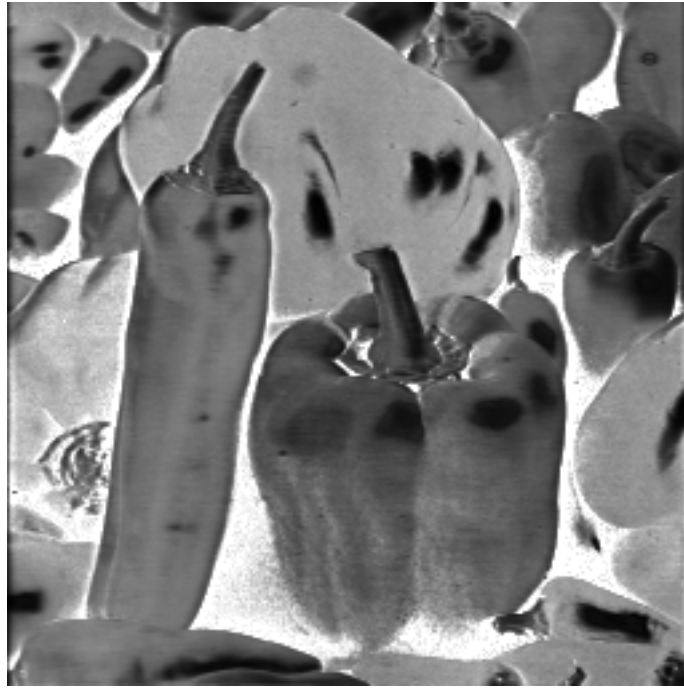


Figure 3: The saturation of the input image.



Figure 4: The intensity of the input image.

```
RGBImageColorTransformations.exe peppers .png COLORSPACE .png
```

5 Software Used

This filter was developed on a Windows 7 64-bit computer. It has been successfully tested with ITK version 3.18.0, MinGW version 5.1.6, and CMake version 2.8.2 (Windows binary), and gcc version 4.3.4 20090804 release 1 under cygwin.