

14.6 Example: The Mona Lisa’s Missing Blue

The Mona Lisa, displayed in black and white in Figure 14.9 has a distinct yellowish cast. Even the sky and the river coursing through the background are yellow. Since the painting is 500 years old, it’s reasonable to expect that some colors of the paint have faded and the that the varnish has yellowed. Also, the colors may have been altered in the process of photographing and reproducing the image. Can we correct the colors to show what the original painting may have looked like?



Figure 14.9. The Mona Lisa, also called *La Gioconda*, painted by Leonardo da Vinci around 1503–1505. (File: monaLisaLouvre.jpg, from the Louvre, <http://www.louvre.fr/img/photos/collec/peint/grande/inv0779.jpg>)

14.6 Example: The Mona Lisa’s Missing Blue | 369

One possible clue to the colors of the original image is the eye. We expect the white of the eye to be white, and not the jaundiced yellow of the figure. The color white is an even mixture of red, green and blue, at high intensity. Let’s examine the pixels of the eye to see what the actual color mixture is. First, we convert the image to a floating point format:

```
>> mona = image2double(imread('monaLisaLouvre.jpg'));
```

and then grab the pixels near the eye:

```
>> eye = imcrop(mona);
```



Using the mouse, grab the region around Mona’s left eye, as shown in the margin: At this scale, it’s easier to see the white of the eye, which is covered by only about 4 pixels. Use `imcrop` to grab just these pixels:

```
>> eyewhite = imcrop(eye)
ans: eyewhite(:,:,1) =
    0.7333 0.8667 0.7843
    0.9647 0.9490 0.8784
    0.9686 0.8902 0.8588
    eyewhite(:,:,2) =
    0.5412 0.6824 0.6078
    0.7686 0.7490 0.6784
    0.7725 0.6784 0.6392
    eyewhite(:,:,3) =
    0.2431 0.3608 0.2902
    0.4196 0.3922 0.3294
    0.3843 0.2863 0.2588
```

A white pixel should have values close to 1 in all three planes, but these pixels have too little green (plane 2) and much too little blue (plane 3). This appears to be a general trend in the image, as a histogram of the three color planes shows in Figure 14.10.

What seems to be required is to increase the amount of green and blue. There are all sorts of ways of doing this: Essentially, what is needed is to transform small values into large ones while keeping the relative relationship between pixels intact. One simple way of doing this is called *histogram equalization*, which transforms the values in such a way that the histogram becomes uniform and flat. The `equalize` function does this for us, one plane at a time.

```
>> r = mona(:,:,1);
>> g = mona(:,:,2);
>> b = mona(:,:,3);
>> newimage = cat(3,equalize(r),equalize(g),equalize(b));
```

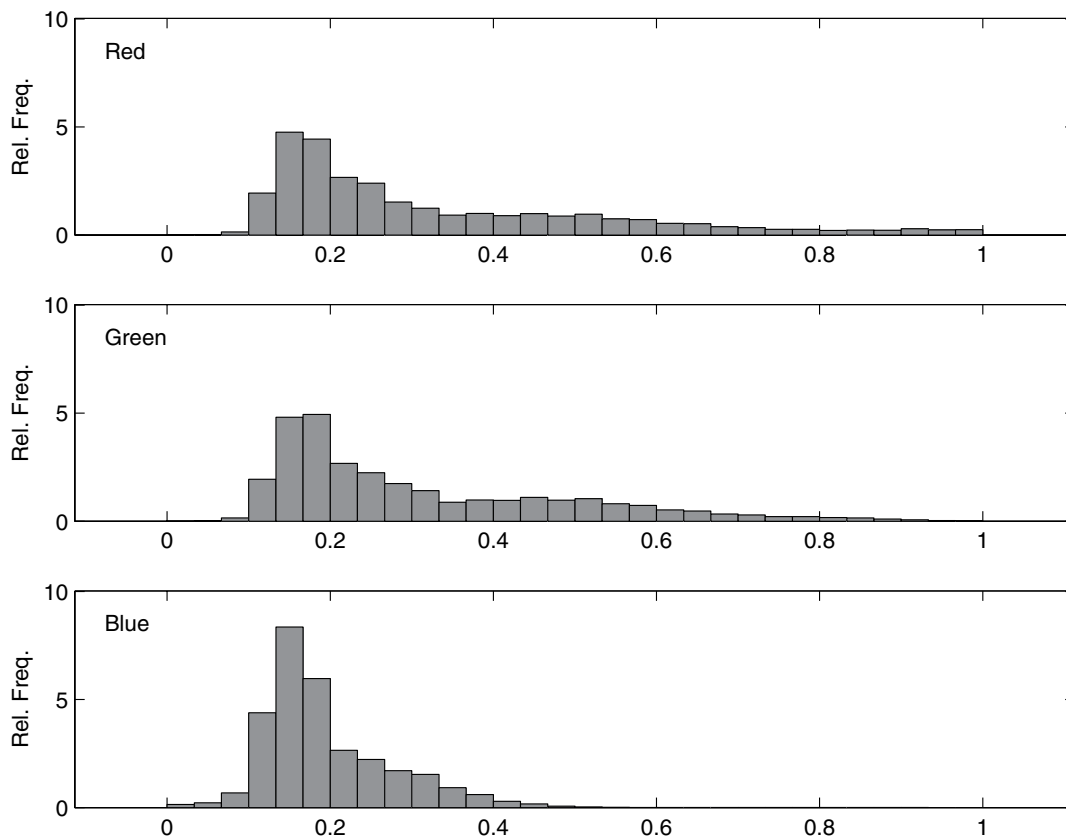


Figure 14.10. Histograms of the three color planes in the Mona Lisa. The histogram can be made by extracting each plane and using the `hist` command:

```
>> r = mona(:, :, 1);
>> hist(r(:), 50)
```

Note the use of the `(:)` to convert the matrix into a vector: `hist` takes a vector as an argument.

This transformation creates an image that is too blue; it’s overcorrected. A compromise is needed; one is shown in Figure 14.11.

The color correction shown in Figure 14.11 isn’t perfect. Part of the reason for this might be that the different pigments contained in different parts of the image might fade a different rates, so that some of the blues are being overcorrected. Doing a definitive restoration of the colors would require a study of the pigment types, as well as an art historian’s understanding of the styles of the day.