# Sharpening in Photoshop CS4

*Tony Freixas*

*http://freixas.deviantart.com/*

## Introduction

**A few months ago, I knew very little about how sharpening worked.** I started researching the topic and found something very strange: some of the information repeated in tutorial after tutorial is incorrect. I discovered this by the simple process of testing each claim to see if it was true.

Even if you consider yourself an expert on Photoshop and sharpening, I believe you will find something in this paper that will surprise you.

I have divided the paper into two parts. Part 1 explains how sharpening works and then describes what I learned about the unsharp mask filter in Photoshop CS4. Part 2 provides much more detail for each claim I make. Please report any errors you find.

My testing was done using Photoshop CS4, but I believe you will find useful information, even if you use other software.

***Some of the information repeated in tutorial after tutorial is incorrect.***

Copyright © 2010, Antonio Freixas. All Rights Reserved

# Part 1: An Overview

## Sharpening basics

### Deconvolution

True sharpening is called deconvolution. Deconvolution restores sharpness to an out-of-focus image. Wikipedia claims[1]:

> *For deconvolution to be effective, all variables in the image scene and capturing device need to be modeled, including aperture, focal length, distance to subject, lens, and media refractive indices and geometries. Applying deconvolution successfully to general-purpose camera images is usually not feasible, because the geometries of the scene are not set.*

Blind deconvolution attempts to make guesses about the scene characteristics in order to recover sharpness. There are commercial deconvolution tools available for general photography[2]—I suspect that they use blind convolution techniques.

Photoshop may or may not contain deconvolution filters. The most likely candidate is Smart Sharpen's Remove Motion Blur option.

### Edge enhancement

Perceived image sharpness is actually an optical illusion created by edge contrast[3]. Most of the sharpening tools you find in Photoshop perform what might best be called "edge enhancements"—the software increases contrast at the edges. This technique makes it easier for our eyes to perceive the sharpness present in an image; it will not help a blurry image look sharp.

Edge enhancement must be used carefully. Increase the edge contrast too much and you will create visible "haloes" around the edges. Sharpening a smooth sky may make any noise in the image more apparent.

1 **http://en.wikipedia.org/wiki/Deconvolution**

2 **http://en.wikipedia.org/wiki/Deconvolution** (refer to the Software section near the end of the article)

3 **http://www.digitalphotoartists.org/pdffolder/sharpening digital images.pdf**

Copyright © 2010, Antonio Freixas. All Rights Reserved

# Understanding Photoshop's sharpening tools

### Low pass filters

To understand sharpening, you need to first understand low pass filters. You may be surprised since low-pass filters suppress the higher frequencies (the edges) of an image and are found under Filters→Blur. But here's the trick: subtract a low-pass filtered image from its original and what's left is equivalent to a high-pass filter.

Photoshop provides several blurring filters. The important lesson here is that it is the Gaussian blur[page 8] that is the exact inverse of Filters→Other→High Pass[page 9].

A key point to understand is that Gaussian blurs are performed on a per-channel basis. Each channel is blurred independently, so it's often easier to understand the blur by using images with just shades of gray.[page 8]

### High pass filters

A high pass filter allows high-frequency information to pass through while suppressing low-frequencies. How does this bring us closer to understanding sharpening?

Think about subtracting an 8-bit image from its blurred copy. The resulting values could range from -255 to +255. This range of values cannot be represented in an 8-bit image—the results need to be adjusted to fit the range 0 to 255. Photoshop constructs its high pass filter by adding 128 to the results of the subtraction, so that pixels with no difference appear at about 50% gray.[page 9] Higher-frequency pixels will be either lighter or darker.

### Overlay blending

Fill a layer with 50% gray, put it above another layer and set its mode to Overlay and you will see the original layer reappear unchanged. Paint this layer with black or white and these will appear as black or white over the lower layer.[page 12] The closer a channel gets to 128, the less effect the channel has on the corresponding channel on the underlying layer.

### Sharpening with a high pass filter

The high pass filter's results are a perfect match for Overlay blend mode. Run a high pass filter on a copy of your image and place it over the original. Set the high pass layer to Overlay mode.

Areas with smooth tones will be near 128 in the high pass image and will not change the original. Areas with edges will have values greater or less than 128 and will lighten or darken the pixels underneath. You have sharpened your image![page 10]

*To avoid color artifacts,[page 12] convert the image copy so that it contains just the luminosity information before running the high pass filter. One way to do this is to create a blank layer below the copy, fill that layer with black, set the copied layer to Luminosity blend, and then merge these two layers.[page 16]*

*Some people believe that image subtraction can be accomplished by placing the image to be subtracted over the image to be subtracted, inverting the top image, setting the blend mode to Linear Light and the opacity to 50%[4]. That almost works, but not quite.[page 11]*

### Unsharp mask (USM)

There are a wide variety of low pass and high pass filters.

The Unsharp Mask (USM) filter (Filter→Sharpen→Unsharp Mask) is based on one variation. It begins with a Gaussian blurred image. Like the high pass filter we just studied, the next step is to subtract the blurred image from its original. Since the goal is to sharpen the image (and not to produce a high pass filter), the unadjusted results are scaled (multiplied by a scaling factor) and added back to the original pixel.[page 12] In this case, the negative values present  no problem because we never need to save them to an 8-bit image.

4 http://retouchpro.com/tutorials/?m=show&id=147

Copyright © 2010, Antonio Freixas. All Rights Reserved

After the subtraction, areas of smooth tones produce a value of 0 and so have no effect when added back to the original. Areas with larger or smaller values will lighten or darken the original pixels. Again, we have sharpened the image.

The mathematics of the subtraction is exactly equivalent to that used to created the high pass filter. The difference is in the Overlay blend—the math is in part 2.[page 10]

The USM has three adjustments:

***Amount***
How much to exaggerate the edges differences. This value is the scaling factor I described above. You can scale by any amount, which is why the control can go up to 500% and answers the question: how can something be more than 100% sharp?

***Radius***
The radius is actually the standard deviation of the Gaussian blur on which the USM is based. The blur extends three times the standard deviation from the center of the pixel being blurred. Because it's measured from the center of the pixel, a radius of 1.0 pixel will affect up to 7 pixels.[page 8] Photoshop seems to allow you to enter radius values of arbitrary precision. In reality, the radius is rounded to the nearest 0.1 unit.[page 8]

***Threshold***
Each processed pixel is brought this many intensity units closer to its original pixel intensity (but never past it). For example, if the threshold is 5, then a pixel with a brightness of 128 and a sharpened brightness of 135, would be set to 130. If the threshold were 10, the pixel would be set to 128. Another example: the threshold is 5, the original brightness is 128 and the sharpened darkness is 121 (this pixels is on the dark side of the edge). The pixel is set to 126. With a threshold of 10, it is set to 128.[page 12]

All descriptions of the threshold setting that I've found, including the one in the Photoshop manual, have been incorrect.

### USM Shortcuts

Photoshop offers some sharpening menu items that have no adjustments. These appear to be variations of USM with hardcoded settings, but I have been unable to verify this. I have found some USM settings that produce results that are very close to those obtained with these menu items, but have not found an exact match.

In order to verify an exact match, the results have to be identical for a wide variety of images. Without an exact match, it is difficult to determine which settings come closest because "closest" can be measured in various ways and is dependent on the image being used. And because of the inverse relationship between Amount and Radius, someone may find a close match to Filters→Sharpen→Sharpen using a USM with amount 130% and radius 0.4, while someone else may find the best match at amount 122% and radius 0.5.

### Noise removal

Some of the articles I found mention that the Photoshop USM contains some automatic noise removal (besides the threshold control). I ran some tests and could not find any sign of noise suppression.[page 14] In fact, because the USM is more aggressive than high pass filter sharpening, noise was more pronounced in the USM.

### Sharpening in steps

Are there any advantages to sharpening in small increments? According to Wikipedia[5]:

> *Applying multiple, successive Gaussian blurs to an image has the same effect as applying a single, larger Gaussian blur, whose radius is the square root of the sum of the squares of the blur radii that were actually applied.*

5 http://en.wikipedia.org/wiki/Gaussian_blur

Copyright © 2010, Antonio Freixas. All Rights Reserved

Since the USM is based on a Gaussian blur, it would seem that this statement also applies to it.

One difference is that the USM includes an Amount adjustment, so applying multiple sharpening filters might be useful if you vary the amount at each step.

Another difference is that, because the radius can be specified with resolution smaller than a pixel, each successive USM will introduce small errors.

Finally, because the USM radius can only be specified in steps of 0.1, it might be possible to use incremental sharpening to achieve radii that would not otherwise be available. For instance, two applications of a 1.5 radius would be equivalent to using a radius of ~2.1213. I suspect, that you would get more precise results using a radius of 2.1 in one step because the error introduced by using two steps would be worse than the error from the missing 0.0213 in the radius size.

### Ultra-fine USM

Photographer Marc Adamus popularized (but perhaps did not invent) a method of sharpening that I have dubbed *utlra-fine USM*. The idea is to achieve more control over the USM radius by first resizing an image to slightly larger than the final size. Sharpening is applied on this image and then it is resized to its intended size. The technique is intended for sharpening images for viewing on a monitor.

I have found some tutorials and Photoshop actions that resize the image to 1⅔ the final size. I have no idea where this scaling factor came from and have no evidence that there is anything special about it.

Because sharpening, in the end, is an aesthetic decision, I can't say that this method produces better or worse images than any other technique. I can say that I have some reservations about it.[page 14]

First, the difference between a 10X image with a radius of 10 shrunk down to 1X size and the same image first resized to 1X and processed with a radius of 1 is very small. I question whether anyone could tell the difference when the image is viewed at 100% magnification from a normal screen-viewing distance.

Second, the radius can already be defined in steps of 0.1.

Finally, in my tests, resizing by small steps introduces errors into an image that essentially "blurs" an image more than resizing in a single step. The ultra-fine USM takes two steps to reach the final size and slightly degrades the image.[page 14]

The process described by Marc also incrementally sharpens the image, saving each increment in a separate layer and blending them to achieve the final result. As described above, each sharpening step introduces errors, so it might be better to create each layer by sharpening the original with a larger radius, rather than sharpening the prior iteration.

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Sharpening tricks

### Sharpen just the Luminosity

Sharpening all the channels may over-saturate some colors.[page 12] You may prefer to sharpen just the image's luminosity.

- Run a sharpening filter and the use Edit→Fade and fade 100% (i.e. full filter effect) using a Luminosity blend.
- Alternatively, copy the layer, sharpen the copy and set the sharpened layer to Luminosity blend. The result is exactly equivalent to the fade effect.[page 12]

Luminosity is based on our visual perception of various colors[page 12] and is not the same as desaturating the image, which gives each channel equal weight.

### Sharpening the Lightness channel

Some people prefer to sharpen the lightness channel of LAB mode. The lightness channel is superior to the luminosity in capturing our perception of brightness.[page 13]

In my tests, Lightness sharpening seems to yield a little more visible detail when compared to Luminosity sharpening, but the differences are very small and may not be visible at 100% magnification. To avoid conversion errors, you should place your image in 16-bit mode before converting to LAB and keep it in 16-bit mode until after you convert back to RGB.[page 16]

### Sharpen the darks more than the lights

Each edge has a dark side and a light side. You may be able to darken the dark side more than you can lighten the light side before haloes become visible.

- Duplicate the image twice.
- Sharpen one duplicate more than the other.
- Set the lightly sharpened layer's blend mode to Lighten.
- Set the more heavily sharpened layer's blend mode to Darken.

## Final thoughts for Part 1

Sharpening is a creative decision. There is no single right answer.

- Web image sharpness is best judged on a monitor at 100% zoom.
- Print image sharpness is best judged by viewing the print at its final size and from its intended viewing distance.

To learn more, please continue to Part 2.

Copyright © 2010, Antonio Freixas. All Rights Reserved

# Part 2: Photoshop's Sharpening Algorithms Revealed

## The Skeptical Inquirer

Adobe's policy over the years seems to be to provide more features and less documentation. We have to guess at how things work. These guesses get passed around in discussion groups and tutorials until they start being accepted as facts, with few people taking the effort to verify them.

I've tried to personally verify most statements in this white paper. This part of the paper will present you with the steps I used to do so.

I encourage you to be skeptical of my article and any others. Check my reasoning and results. If you find any mistakes in this article, please let me know.

## Notation

I will provide the mathematical formula for some of the blend modes. Here is the notation I use:

- *A* represents a channel in the layer **a**bove (the one where the blend mode is set) and *B* represents the channel in the layer **b**elow it.
- Unless otherwise stated, the blend mode formula is applied independently to each pixel channel.
- The channel values have been normalized to be from 0.0 (black) to 1.0 (white).
- The opacity has also been normalized from 0.0 (fully transparent) to 1.0 (fully opaque).
- The notation A' means 1.0 - A. This is the same as inverting.
- Anything within « and » is limited to the range 0.0 to 1.0. Smaller values become 0.0 and larger values become 1.0.

All the blend mode formulas were verified by writing a program that used the formula to combine two carefully generated test images and then comparing the results I produced with what Photoshop CS4 produced.

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Definition of Gaussian blur

To apply a Gaussian blur to an image, we create a new image where each new pixel is assigned the weighted average of the original pixel and its neighbors. The weights are determined using this formula[6]:

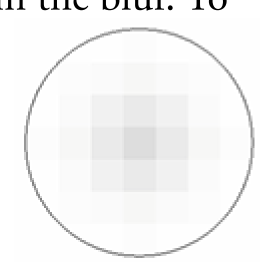
$$G(x,y) = \left(\frac{1}{2\pi\sigma^2}\right) e^{-\left(\frac{x^2+y^2}{2\sigma^2}\right)}$$

where $x$ and $y$ are coordinates in a 2-dimensional grid. Position (0, 0) generates the weight for the pixel, (1, 0) is the weight for the pixel immediately to the right, (0, 1) is the weight for the pixel immediately above, and so on.

Various weighted averages can be generated by changing σ, the *standard deviation.* What Photoshop calls the radius is actually the standard deviation.

In practice, when computing a discrete approximation of the Gaussian function, pixels at a distance of more than 3σ contribute so little to the weighted average that they can be ignored. If you specify a radius of 1 pixel, the affected pixels are within 3 pixels of the point being blurred. However, because the point being blurred is considered to be at the center of the pixel, the blur will affect 7 pixels horizontally and 7 vertically (5 pixels along the diagonal).

This seems to be a great source of confusion and you will find all sorts of claims about the true radius of the circle of points included in the blur. To verify that the radius is 3σ as measured from the center of the pixel, put a single black pixel on a white background and blur it with a radius of 1. Look for all pixels whose values have changed. You will probably need to use the information palette when viewing the outermost pixels, as they are nearly white (254, 254, 254).



### Gaussian blur is calculated per channel

The weighted average is calculated independently for each channel. You can verify this by duplicating an image and applying the blur to each channel in one copy and to the whole image in the other. The results are identical.

### Sub-pixel radii

What does a radius of 0.1 mean? Pretend that each pixel can be divided into a 10x10 grid of mini-pixels. We apply the blur to the mini-pixels and then average the 100 pixels to get the value to assign to the real pixel.

Pixel blurring still occurs from the center of the pixel. Since the center of a 10x10 grid falls at the corner of four mini-pixels, the blurring circle will cover at most 6 mini-pixels in any single direction. Since all mini-pixels within a pixel have the same intensity, the blurred mini-pixels will retain their original values, as will the real pixel.

If this is true, a blur radius of 0.1 should have no effect, which is indeed the case. A blur radius of 0.2, on the other hand, will affect 12 mini-pixels and so extends partly into the neighboring pixels and starts producing a visible effect. The best way to see this is to blur a single black pixel on a white background.

### Verifying the resolution of Photoshop's radius

The resolution of the radius is no more than steps of 0.1. To verify this:

- Load any image
- Duplicate the background layer
- Apply a sharpen or blur with a radius of 0.25 to the background layer
- Apply a sharpen or blur with a radius of 0.3 to the top layer
- Compare the two layers

6 http://en.wikipedia.org/wiki/Gaussian_blur

Copyright © 2010, Antonio Freixas. All Rights Reserved

I have not done extensive testing of this, but in the case above (using USM), the resulting images are exactly the same. In addition, Photoshop remembers the last radius it used. If you enter 0.25, apply the USM and then bring up the USM dialog again, Photoshop will display 0.3.

It is possible that larger radii might have less resolution. I have not checked.

## Conversion of a Gaussian blur into a high pass filter

As described in part 1, you can use a Gaussian blur to create a high pass filter by subtracting the blurred image from the original and adding 0.5, so the formula is:

« $(B - A) + 0.5$ »

where B is the original image and A is the blurred copy.

If A and B are the same, the result is middle gray or 0.5.

Since A and B are in the range 0.0 to 1.0, the formula $(B - A) + 0.5$ will result in a range of -.05 to 1.5, which we then limit to 0.0 to 1.0. While we cannot completely capture the full range of values, the difference between a pixel and its blurred twin will usually fall into this range (if we discount noise).

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Understanding overlay mode

The formula for Overlay blending is:

*Overlay(A, B) = If A < 0.5 then 2AB else (2A'B')'*

When A = 0.5, it doesn't matter which formula you pick—the result is always B. Therefore, if you place a 50% gray layer over an image and set the blend mode to Overlay, you will see the image.

If A < 0.5, then 2A is less than 1.0. Therefore, 2AB will be less than B and the smaller that A is, the smaller the result. If A = 0, then the result is also 0.

If A > 0.5, then A' < 0.5 and 2A' is less than 1.0. Therefore, 2A'B' will be less than B'. If 2A'B' < B', then (2A'B')' > (B')' and (2A'B')' > B. When A = 1, the result is always 1 (since A' = 0 and 0' = 1).

## Sharpening with the high pass filter

The formula for sharpening with a high pass filter is:

*Overlay(« (B - A) + 0.5 », B)*

Expansion of this formula is left as an exercise for the reader.

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Approximating subtraction by using Linear Light at 50%

The formula for Linear Light is:

*If A < 0.5 then LinearBurn(2A, B) else LinearDodge(2(A - 0.5), B)*

The formula for Linear Burn is:

*max(0, A + B - 1)*

The formula for Linear Dodge is:

*min(1, A + B)*

The formula for an opacity blend on a layer with a blend mode is:

*blend(A, B)opacity + B (1 - opacity)*

If we invert A and then apply a Linear Light blend at 50% opacity and work through the math, we get:

*If B < 2A - 1 then 0.5B*
*If B > 2A then 0.5(1 + B)*
*If 2A - 1 <= B <= 2A then (B - A) + 0.5*

Therefore, we can represent accurate results only when *2A - 1 <= B <= 2A*. Outside of this range:

*If B > 2A, then the error is | 0.5(1+B) - (0.5 + B - A) | = | A - 0.5B |*
*If B < 2A -1, then the error is | 0.5B - (0.5 + B - A) | = | A - 0.5B -0.5 |*

*This trick came from an article[7] by Doug Nelson, but he is incorrect in believing that this correctly simulates subtraction and that any differences from the expected results are due to round-off errors.*

## Verifying the conversion of a Gaussian blur into a high pass filter using Photoshop

Using the Linear Light trick, you can convert a Gaussian blur into a high pass filter just using Photoshop. As noted above, the result is not exact, but because the differences tend to be small (specially at low blur radii), the trick tends to get reasonably close results.

1. Load any image. Flatten, if necessary.
2. Duplicate the Background layer and name it "Original".
3. Convert this layer to contain only Luminance information:
   a. Create a blank layer between the Original layer and the Background.
   b. Fill the blank layer with black.
   c. Set the Original layer to "Luminosity" blend.
   d. Merge the two layers.
5. Duplicate the Original layer. Label it "Gaussian blur".
6. Duplicate the Gaussian blur layer. Label it "High pass filter".
7. Blur the Gaussian blur layer with Filters→Blur→Gaussian Blur...
8. Blur the High pass layer with Filters→Other→High Pass. Use the same radius as for the Gaussian blur.
9. Subtract the Gaussian blur layer from the Original layer:
   a. Invert the Gaussian blur layer.
   b. Set its blend mode to "Linear Light" and reduce the opacity to 50%.
   c. Merge the Original and Gaussian blue layers.
4. Set the High pass layer to "Difference" blend.
5. The resulting image should be completely black. If you look at the histogram you will see most pixels are black. A small number of pixels might not be black due to the fact that this method only approximates subtraction.

7 http://retouchpro.com/tutorials/?m=show&id=147

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Understanding Luminosity

Most blend modes operate independently on each channel. Some, however, operate on all channels—luminosity is one of these. Luminosity is a measure of the brightness of a pixel, independent of the pixel's hue and saturation.

A Luminosity blend replaces the luminosity of layer B with the luminosity of layer A. Since grayscale images have no hue or saturation, you can place any image over any grayscale image and if the top image is set to Luminosity blend, you will be viewing the luminosity of the top layer. Changing the bottom grayscale image with any other grayscale image will not change the result.

For RGB, the luminosity formula is:

$$Luminosity = 0.30R + 0.59G + 0.11B$$

This formula tries to match our visual response to each channel. You can verify the formula by following these steps:

1. Load any image.
2. Duplicate the background.
3. If necessary, promote the background to a layer.
4. Add a layer below the background and fill it with white.
5. Set the background layer to Luminosity blend.
6. Select the top layer and use the Channel Mixer adjustment. Enable the Monochrome option and set the Red, Green and Blue channels to 30, 59 and 11, respectively.
7. Compare the top layer to the bottom two layers.

In my tests, there is a very small difference between the two, suggesting that more digits of precision are required in the formula, but the channel mixer can only handle integer values.

Luminosity is not the same as the Lightness channel in LAB mode. The formulas for calculating Lightness are much more complicated (refer to the next section for more on LAB mode).[8]

Luminosity is also not the same as the L in HSL (hue, saturation, lightness)[9], although you may encounter a lot of confusion about this if you search the web. For HSL, L is defined as:

$$L = \frac{max(R,G,B) - min(R,G,B)}{2}$$

This is not the same as our luminosity formula.

### Sharpening and Luminosity

Since edge sharpening is performed identically on each channel, the final effect may not be exactly what you want. For example, imagine that you have an area in your image which has only blue channel values. Sharpening will lower the blue value of the dark side of an edge and raise the blue value of the light side. Increasing the blue value makes the blue more intense, which may not be the effect you desire.

One alternative is to sharpen just the image's luminosity. You can do this in two ways:

1. Sharpen the image and then fade the sharpening operation with an opacity of 100% and a mode of Luminosity.
2. Duplicate the layer, sharpen the duplicate and then set its blend mode to Luminosity.

The two methods yield exactly the same results.

8 http://en.wikipedia.org/wiki/Lab_color_space
9 http://en.wikipedia.org/wiki/HSL_and_HSV

Copyright © 2010, Antonio Freixas. All Rights Reserved

## Understanding LAB Mode

The name LAB is not short for "laboratory" but for the three channels that comprise it: Lightness, A and B.

The LAB color space is perceptually uniform, which means that a change of a given amount in a color value will produce a change of about the same perceived visual importance.[10] In other words, numbers in LAB space match the way we see things.

Luminosity and the Lightness channel of LAB mode try to do the same thing in tracking the way we perceive brightness. The formula for luminosity is fairly simple; the one for Lightness is rather more involved (check the Wikipedia page in the footnote if you're interested). The point is that, if the Lightness layer does a better job of matching the way we see brightness, it might represent the better starting point for a sharpening operation.

### RGB to LAB Mode and Back

Can you convert from RGB mode to LAB mode and back without introducing conversion errors?

The surprising answer is yes as long as you do the conversion in 16-bit mode and as long as the final image will be in 8-bit mode. Errors introduced from the conversion are smaller than 256.

## Shrinking an image in multiple steps

Some people appear to believe that you lose less image information by resizing in small steps rather than all at once. This cannot be true as I'll explain, but you can try it yourself with this test:

- Create a large image with an area of black alongside one of white.
- Duplicate the image.
- Reduce the duplicated image size by 90% 10 times using bicubic interpolation.
- Resize the original image to the same size as the duplicate using bicubic interpolation.
- Inspect the edge.

With a perfect resizing algorithm, there should be at most one pixel of gray between the black and white regions. In my tests, the bicubic resize yielded perfect results when using a single resize, but not when doing incremental resizes.

The reason seems pretty obvious. When resizing to a smaller size, each resulting pixel is a blend of multiple original pixels and information is lost. With a single step resize, the lost information is not relevant—you have created the best representation of the original image in its new size. When resizing in small steps, each resize after the first will be increase the error and will spread the error out to more and more adjacent pixels.

I don't believe the particular algorithm makes any difference. Each resize necessarily loses information and each subsequent resize increases the information loss. Artistically, of course, this might yield some interesting effects, but you are not preserving the maximum amount of captured information.

10 http://en.wikipedia.org/wiki/Lab_color_space

Copyright © 2010, Antonio Freixas. All Rights Reserved

## The USM formula

Thanks to Bob Fisher, Simon Perkins, Ashely Walker and Erik Wolfart for their article[11] on the USM. The formula for a basic USM is

« *B + k(B - A)* »

where A is the Gaussian blurred image, B is the original image and *k* is a scaling constant. The value of *k* corresponds exactly with the Photoshop Amount adjustment. In other words, a *k* of 1 is the same as an adjustment of 100%

Through testing, I was able to determine how to add Photoshop's threshold value to the formula. In this formula, *t* is the normalized threshold (i.e. threshold / 255.0):

« *B + sign(k(B - A))* « | *k(B - A)* | *- t)* » »

Essentially, the change to B is a linear equation (an equation in the form *ax* +*b*) where we control both the scale (*a*) and offset (*b*). Note that if the difference is greater than 0, the threshold can never reduce it to less than 0. Likewise, if the difference is less than 0, the threshold can never increase it to more than 0.

This formula is *not* the same as using the high pass filter sharpening technique, even if we omit the scaling and threshold adjustments (see the formula for high pass in a preceding section).

11 http://maths.sci.shu.ac.uk/units/20-6553/jw/FilterImage/unsharp.pdf

### A common falsehood

Here's what the Photoshop CS4 manual has to say about threshold:

> *Drag the Threshold slider or enter a value to determine how different the sharpened pixels must be from the surrounding area before they are considered edge pixels and sharpened by the filter. For instance, a threshold of 4 affects all pixels that have tonal values that differ by a value or 4 or more, on a scale of 0 to 255. So, if adjacent pixels have tonal values of 128 and 129, they are not affected.*

You can easily disprove this description:

- Fill an image with 50% gray (RGB 128, 128, 128).
- Using the pencil tool with a 1-pixel brush size, paint a few pixels at random locations and with an RGB value of 129, 129, 129.
- Use the USM amount to 500%, and the radius to 250.

Now set the threshold to 4 as described in the manual. The image should be unaffected, but you will see an obvious change.

### USM and Noise Reduction

Some people believe that the USM includes some noise reduction. I don't believe this is the case.

Try starting with the same test image as above, but leave the threshold at 0.

With noise suppression, you could reasonably expect that the image would stay gray with the RGB 129, 129, 129 points unaffected. This is not what I observed.

### Ultra-fine USM

If you would like to a radius with more resolution than 0.1, you can apply the radius to an image larger than the final target size and then shrink the image to the intended size. For instance, to achieve 0.01 resolution, you could work with an image 10 times larger than the target.

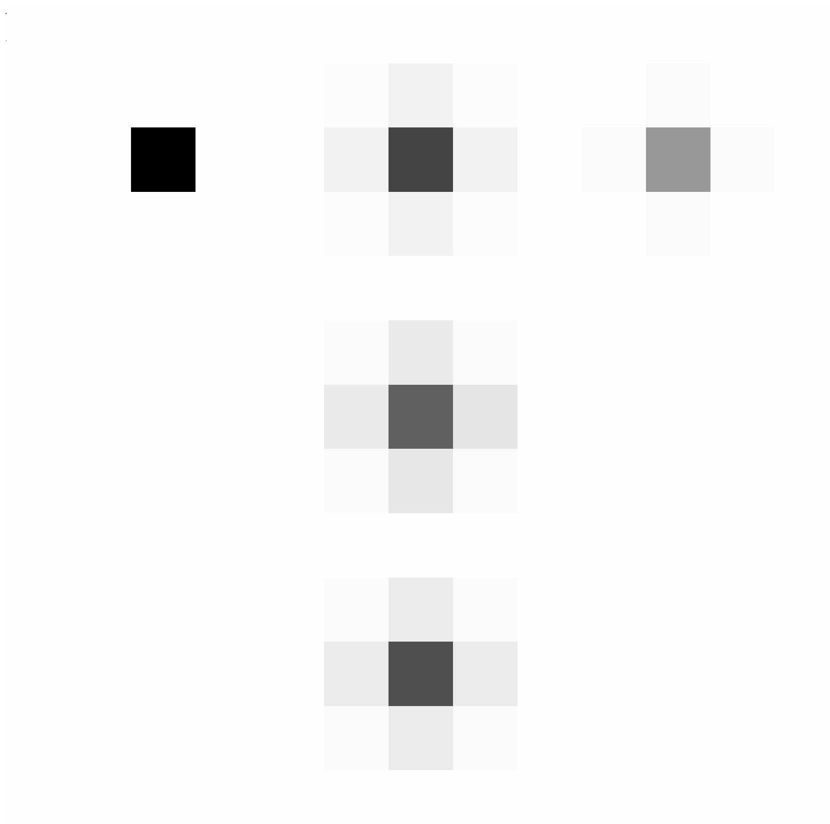
Copyright © 2010, Antonio Freixas. All Rights Reserved

To measure the effectiveness of the ultra-fine USM, I constructed two test images. One was a 10x10 image with a white background and one black pixel in the middle. The other was a 100x100 image with a white background and a 10x10 black square in the middle.

Since the USM is based on a Gaussian blur, I blurred the smaller image with a radius of 0.1 and 0.2, saving each result independently. I then blurred the larger image with a radius of 1.5. This is the furthest we can get from 0.1 and 0.2.

The ultra-fine USM technique usually specifies bicubic resampling, so I used this method to reduce the larger image to 10x10. As a "control", I also reduced the original 100x100 image (i.e. without the blur) to 10x10.

I used one other method to reduce the image. First, I manually averaged each 10x10 block in the large image using Edit→ Blur→Average. Then I reduced the image to 10x10 using nearest neighbor resampling. In theory, this procedure best captures our intent of increased radius resolution while introducing the fewest artifacts from the size reduction.



From left to right, the top row shows the results of using radii of 0.1, 0.15 and 0.2. The 0.15 radii was generated using the nearest neighbor resampling method.

The next row shows a 0.15 radius achieved by using a bicubic reduction. This looks good until we compare it to the bottom row, where I simply reduced the 10x10 black square. The changes created by the bicubic reduction seem to swamp the added resolution of the blur.

When viewing this sample image in actual size, the differences in each of the above is very subtle (because of the way the images in this paper are viewed, I won't attempt to reproduce the image at full size here).

Normally, an ultra-fine USM is executed with a magnification around 1.6X, which provides little added radius resolution It also introduces a second resize and that resize will add some blur (as demonstrated in the bottom row of the image on the left).

My conclusion is that the ultra-fine USM method is not a good way to sharpen an image.

I would be interested in any counter-arguments as a lot of people appear to believe in this approach.

Copyright © 2010, Antonio Freixas. All Rights Reserved

# Future Topics

Having spent several months researching this paper, I understand why most people are willing to accept the information they are given. I hope you have acquired some skepticism and that a few of you may be willing to research the topic of edge sharpening.

There are many areas I haven't had time to examine and these could be material for a future paper:

- Smart Sharpen
- Adaptive Sharpening
- Sharpening with edge masks
- Deconvolution filters

Some questions:

- Is Smart Sharpen / Remove Gaussian Blur the same as USM?
  (I think it is.)
- Is Smart Sharpen / Remove Lens Blur a deconvolution filter?
  (I don't believe so.)
- Is Smart Sharpen / Remove Lens Blur a USM with an edge mask?
  (Possibly.)
- Is the Smart Sharpen / Remove Motion Blur a deconvolution filter?
  (Possibly.)
- What do the Advanced mode settings do?
- What does the More Accurate setting do?
  (The rumor is that it runs the algorithm twice, the second time with a smaller radius.)

A good analysis of deconvolution filters would be interesting. My current thinking is that they are good for scientific and forensic uses, but less so for artistic uses because some noise will always be introduced. I would be interested in knowing more.

If you've made it through this paper, I thank you for your time and I hope I have clarified more than I've confused.

—*Tony Freixas*
*June 9, 2010*

Copyright © 2010, Antonio Freixas. All Rights Reserved