

# *Windo Watch*



---

**December 1996**  
**Vol 2      No. 10 B**

**NEXT**

***ww***

---

# WHAT'S INSIDE

Vol.2 No. 10B

December 1996

---

**This is WWIN210B.PDF**

The Editor's Soapbox

**Behind the Scenes: Creating Computer Art**

Teaching Old Dogs New Tricks:*HTML III-A Tutorial*

Taming of the Wild DNS: *A MetaInfo Product Review*

Terraforming Templates

Reflections of A ModemJunkie

Corel Presentations:*WordPerfect Suite Part III*

Adobe's PhotoDeluxe for Windows: *A Review*

Adobe's Acrobat v.3 Flies High: *A Product Review*

Charles Babbage- A Computer Pioneer

The Making of a Genius

Internet Humor: *An Anonymous Collection*

Cookies in the Oven

Searching for the Middle Ground

The WindoWatch Art Gallery

The LastWord

Lois Laulicht

**Herb Chong**

Gregg Hommel

Jim Plumb

Jack Passarella

Leonard Grossman

Frank McGowan

Linda Rosenbaum

Lois Laulicht

Russ Jensen

David Kindle

Daniel Christle

Lois Laulicht

Original Work from Paul Kinnaly and Herb Chong

Ben M. Schorr

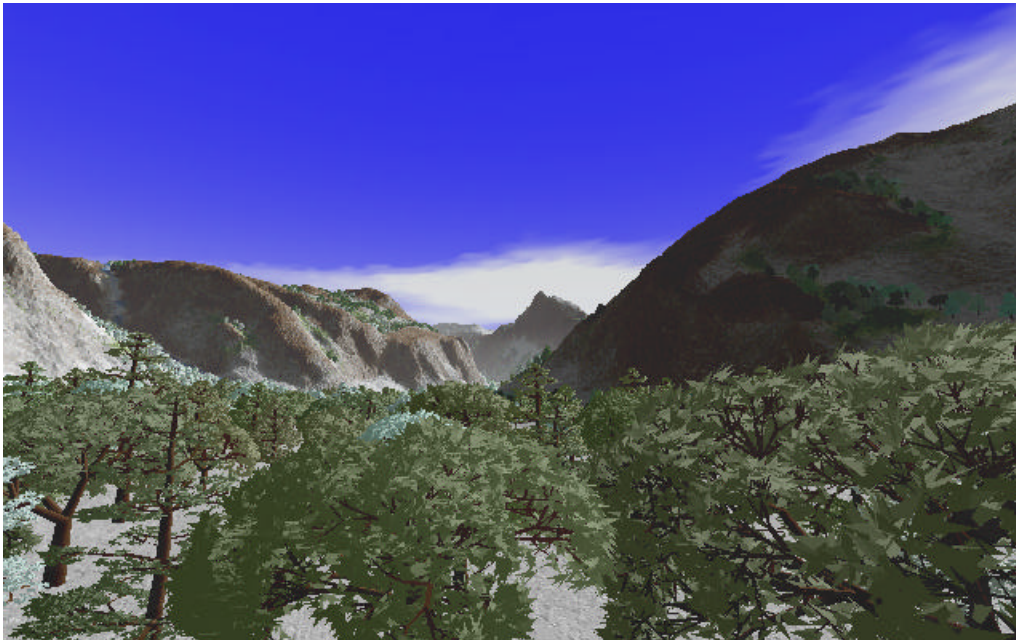
## Behind the Scenes: Creating the Art Gallery

Copyright 1996 by Herb Chong

It's been almost a year since I started the Art Gallery part of the *WindoWatch* home page. I haven't really had a chance to describe much in the way of how I create the types of pictures that have appeared in Art Gallery or the others that are on my web page. This article will describe some of the things I've done to create my art along with the tools I use and how I use them to build images. The techniques have changed over the years as I acquired new tools and upgraded the older ones.

What makes this easier for me is my training in photography. Although I never took any formal courses, I studied many techniques on my own and experimented with composition and layout so that I learned how to balance and place elements of a picture to build a pleasing image. Having good material and good ideas are not enough without the training of the eye to create and recognize good composition. After the original idea, having the tools to perfect the idea is most important. Following are descriptions of several of the images I created, the tools I used and how I used them to produce the images.

### Winter in Yosemite



**Figure 1** (yos1.bmp)

import landscapes from a bitmapped data format. VistaPro can color code the terrain by elevation and populate the landscape with trees of various types. Here is what the basic user interface looks like. (Note: VP 4.0 is now in beta and has a radically different interface.)

I used VistaPro 3.0 for Windows to create this image. There are several others like it in the series. VistaPro is published by Virtual Reality Labs, Inc. They recently merged with RomTech, Inc and are now a single company. You can reach their web site at <http://www.romt.com> VistaPro is a 3D rendering and animation program designed to take US Geological Survey (USGS) Digital Elevation Modem (DEM) data, randomly generated terrain, or imported data and render it as a 3D landscape. You can alter the landscape with its internal tools or export and

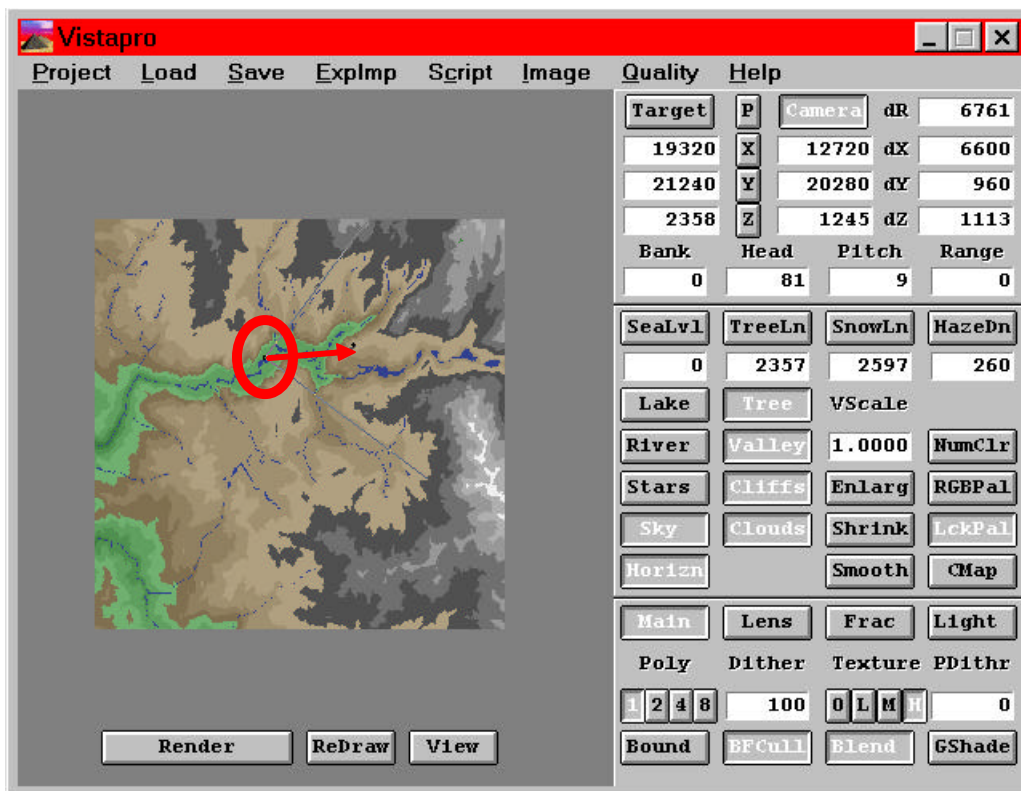


Figure 2 (vp1.bmp)

Although there are a lot of settings on the screen, once you start working with landscapes in VistaPro for a while, you will find that you change most of them only once and never do anything more with the setting. I've added a red circle and arrow showing the camera position and the direction it is pointing. When using the program, it's a little easier to see the dot and gray marker lines it uses than in this screen shot.

I work with VistaPro by first choosing an interesting place to build a scene from. Usually, this is

some place with mountains and lakes. Then, I go looking for the data. VistaPro 3.0 for Windows comes on a CD-ROM with low resolution data for many interesting areas and also for the entire continental US. More detailed data can be obtained from the USGS going to <http://www.usgs.gov> and following the links for online mapping data. For this sample, I used data on the VistaPro CD-ROM of Yosemite valley and area.

The data included doesn't have water features such as lakes and rivers. I manually added them in by dropping some water on a spot and letting VistaPro figure out where the water is supposed to run. Naturally, I choose spots where I know water really runs in the actual landscape. VistaPro also lets you choose the elevations where the treeline and snowline begin and how fuzzy that line is. For this example, I let VistaPro default the values. The one other thing I had to choose was the color scheme. VistaPro allows you to assign the color to each type of object it generates. I chose to use the supplied Winter color scheme. This makes everything look snow-covered with bare spots where the wind has blown. I increased the tree density for the picture so that there would be less bare rock.

Finally, with the right terrain and setting in mind, I had to find an interesting scene and adjust the lighting. I do this by simple experimentation. I imagine where there might be a good picture and do a test render. With the detail turned down, a test render might take about 10 seconds. I'll usually try three or four places and camera angles before I settle on one I like. Then, I tweak the light. VistaPro allows you to set the sun angle and position. I usually place it to get a good mix of sun and shadows, again, doing test renders to be sure that I like what I see. Once I am satisfied, I turn on all the fancy rendering options and render to the output size I want. The image in Figure 1 was done at 640x480. It took about 4 minutes to render on a Pentium 90. Others in the Winter in Yosemite pictures can be found on my web page. The only difference between them is the position and aiming of the camera. All other parts and settings of the landscapes are the same.

## UFO Overseas

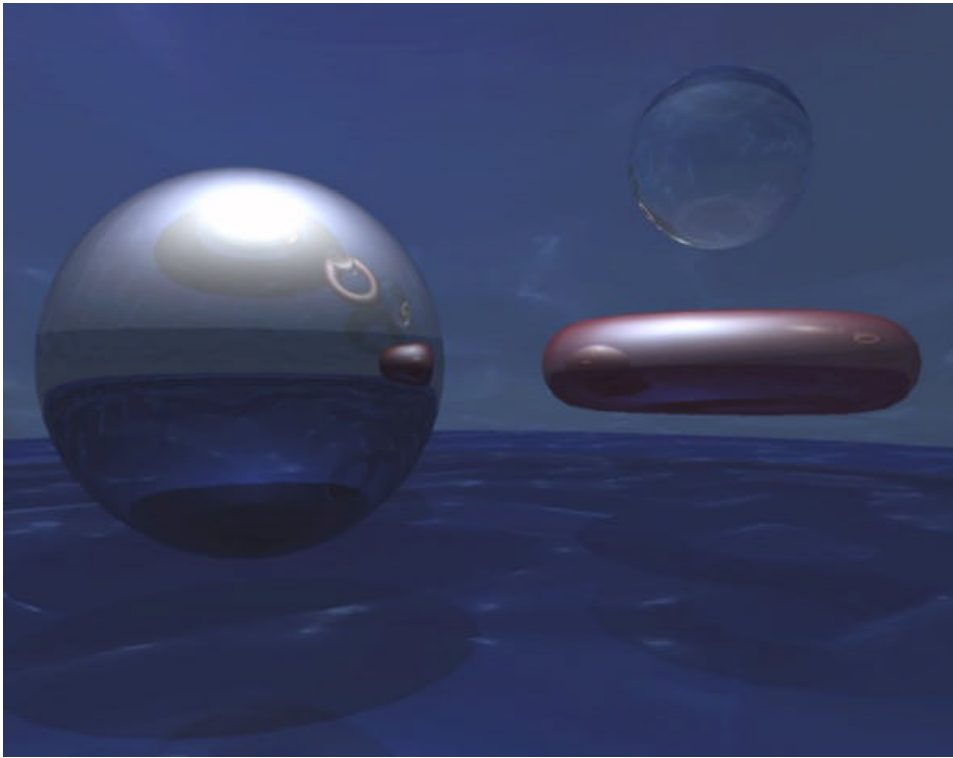


Figure 3 (overseas.bmp)

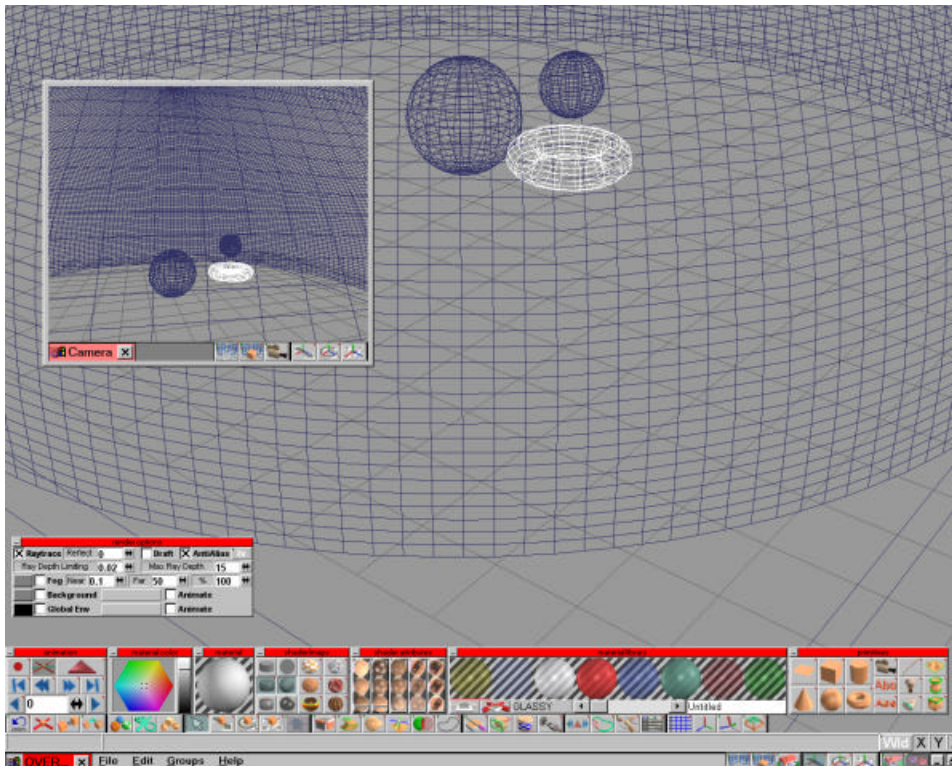


Figure 4 (ts1.bmp)

and did a few test renders until I was happy with the basic layout. I did have to adjust the opacity of the slab that formed the

I used trueSpace 2.0 for Windows to create this image. trueSpace is a 3D modeling and animation rendering program that I use to create most of my non-landscape-based images. The program is published by Caligari Inc. You can check out their web page at <http://www.caligari.com>.

Caligari publishes several programs related to 3D modeling and rendering. Their latest suite of programs is designed to build VRML worlds for the web. Despite the seeming complexity of this image, the hardest part about creating it wasn't putting the thing together but waiting the 25 hours needed for the image to render.

Figure 4 is a screen shot of the working version of the scene with many of the trueSpace tools visible. I won't go into the user interface other than to say that trueSpace is one of the easiest to use of the 3D modeling and rendering programs. So many others are more complicated and less easy to use.

UFO Overseas is basically a very simple scene. There are two spheres and a torus, all inside a hemisphere. Underneath the hemisphere is a rectangular slab of semitransparent blue-green glasslike material. One sphere is shiny almost perfectly reflective silver blue metal. The other is shiny transparent glass. The torus is reddish shiny metal. The inside of the dome is also shiny silver blue metal. There is a single spotlight aimed straight down from the top of the hemisphere just below the top surface. I moved the camera around until I got something interesting in the wire-frame view of the camera window



water and I also ended up adding a bump map so that the water would not be flat but would be rippled and reflect patterns, but I did this after a test render and not seeing enough interesting details in the resulting image. In all, I probably spent no more than half an hour fooling with the scene before I did my final render.

## The Web Surfer

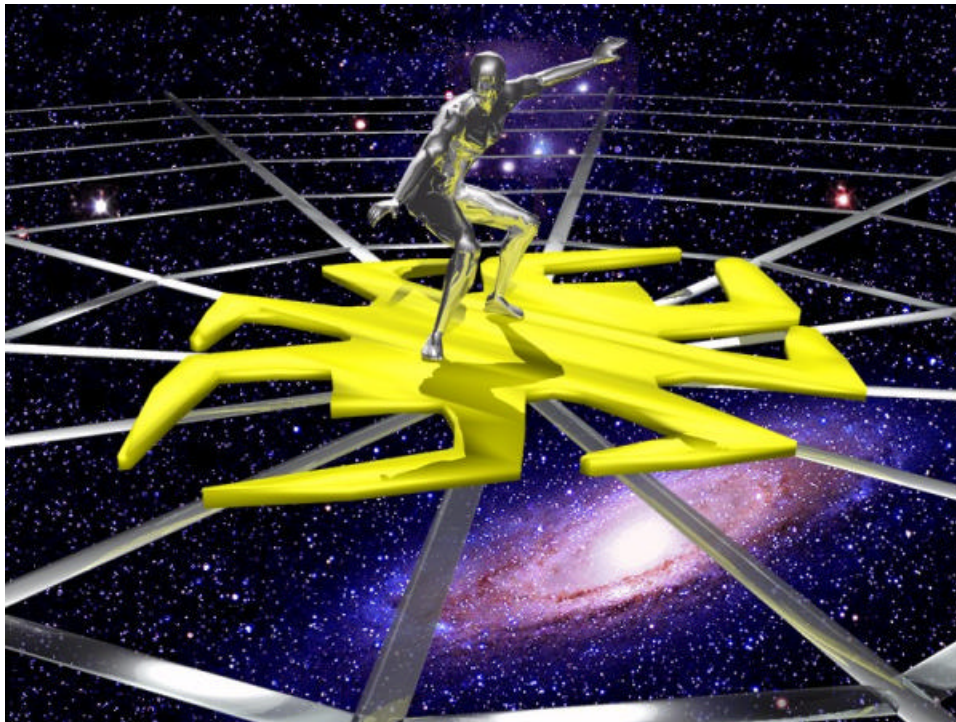


Figure 5 (websurf.bmp)

program designed to create 3D human figures in a variety of poses and simulate the movement ability, the degree of freedom, of the human body. As it happens, they have a surfer pose in one of their sports pose libraries. Figure 6 shows a portion of the user interface. Since all I cared about was the shape of the object and nothing else, I just clicked on the various poses in the library and then exported the 3D model when I found the one I wanted.

This is one of my better known pictures. I originally started it for a project at work, but have tinkered with it some since then. This is the original and best looking of them. The original idea came about because I needed a graphic to symbolize surfing the World Wide Web without just coming out and saying so. I figured I had the tools and could come up with something flashy without too much effort. Some guy on a spiderlike thing on a web was what I was after. Despite appearances, doing the surfer was dead easy. The hardest part was doing the fake spider web.

The first thing I started with was the surfer guy. It was easy to do because just before I needed this image, I had bought Fractal Design Poser for Windows (go to <http://www.fractal.com> ). It is a

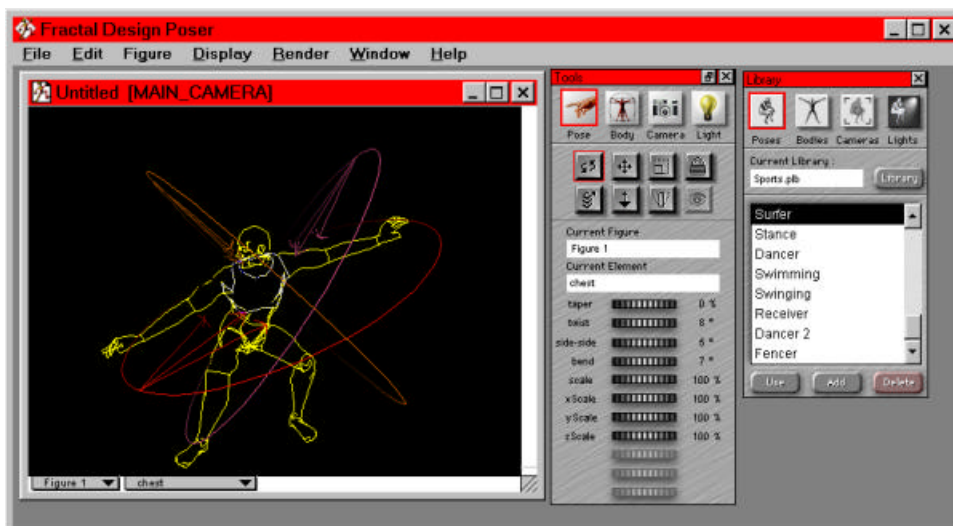


Figure 6 (poser.bmp)

The next most difficult part was coming up with the spiderlike thing that I wanted the surfer to stand on. I remembered seeing a image similar to what I had in mind on the Webmaster, Inc. home page <http://www.webmaster.com>.



This is what it looked like.

**Figure7 (original\_spider.bmp)**

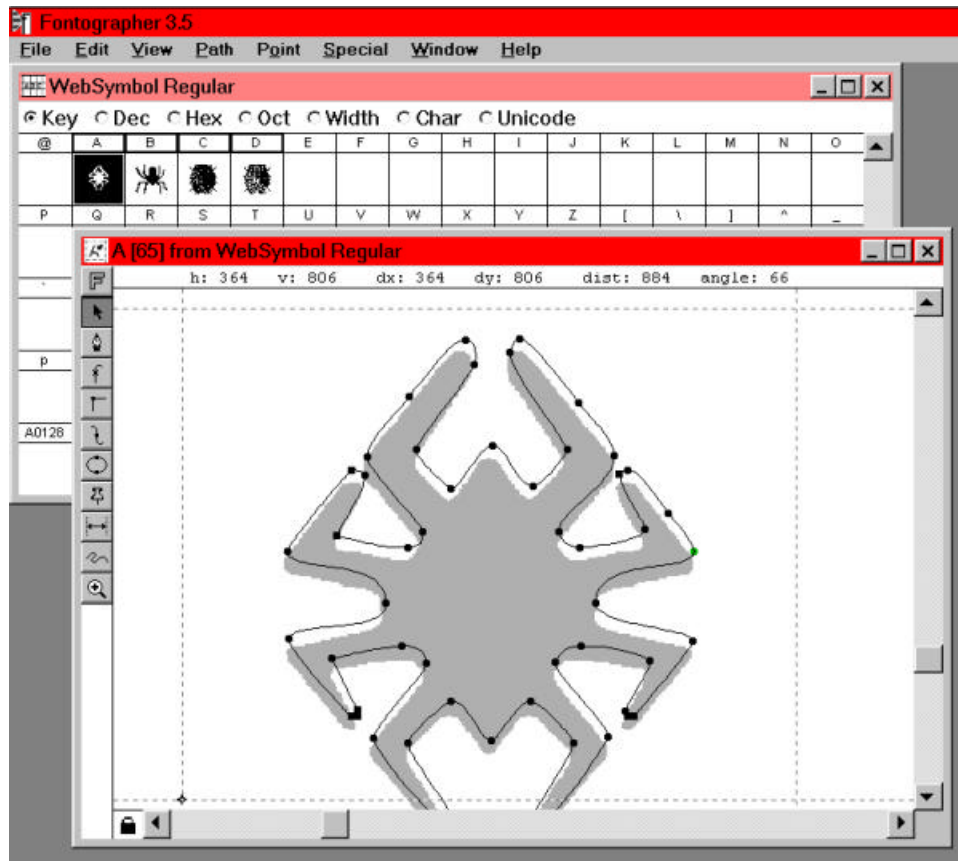
I couldn't use it the way it was because it was very small and also a plain flat image. I needed a 3D object for my scene. There are many ways I could have done it, but I chose to use Fontographer <http://www.altsys.com> to create a special font out of the shape and then enter that as a symbol into trueSpace. First, I had to convert the picture in Figure 7 to something easier for Fontographer to handle. Figure 8 shows the cleaned up image ready for import into Fontographer.



**Figure8 (spider\_outline.bmp)**

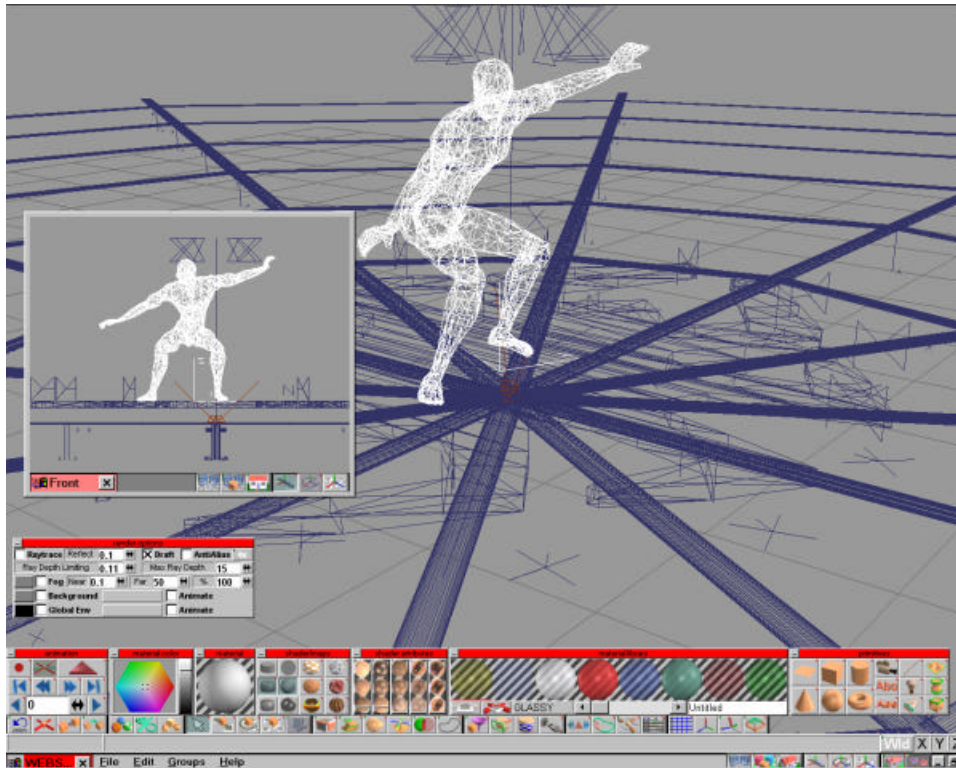
Once in Fontographer, I had to convert the bitmap image into an outline. Luckily, this is something that font developers do often, so I used the trace outline command to find the outer shape of the spider and touch up the shape so that it would be more symmetric. You can see the symbol as it appears in Fontographer in Figure 9. I could have just as easily traced the outline using Corel Trace and exported as an Adobe Illustrator shape, but I was going to need the symbol as a font character for other things, so using Fontographer seemed the best. I never did use the symbol as a font.

With the surfer and the surf board built, I had to construct the web. I first tried to find a real spider der web and trace its outline just as did for the spider shape, but that turned out to be an eight hour waste of time. I eventually settled on constructing my own fake web.



**Figure 9 (fontog.bmp)**

**Figure 10** shows the in-progress view of the desktop. The surfer is highlighted in white. The spider shape is below him. The web is nothing more than a bunch of rotated thin cylinders joined together with other thin cylinders. They are easily visible in the perspective wireframe view. For a variety of other reasons, the surfing spider had to be a bright yellow. I experimented with different surfaces for the surfer until I settled on a very shiny silver metal. I played a little with the web, but finally decided on a mostly transparent shiny plastic. Then, I found a background image of some stars and a galaxy from one of the NASA web pages and used it as the back-ground. The two lights in the picture are two of the infinite lights that are created by default in trueSpace. I deleted the other two that trueSpace creates because they would have lit the surfer from below and cast shadows that I didn't want.



**Figure 10** (ts2.bmp)

## Penguin Dreams

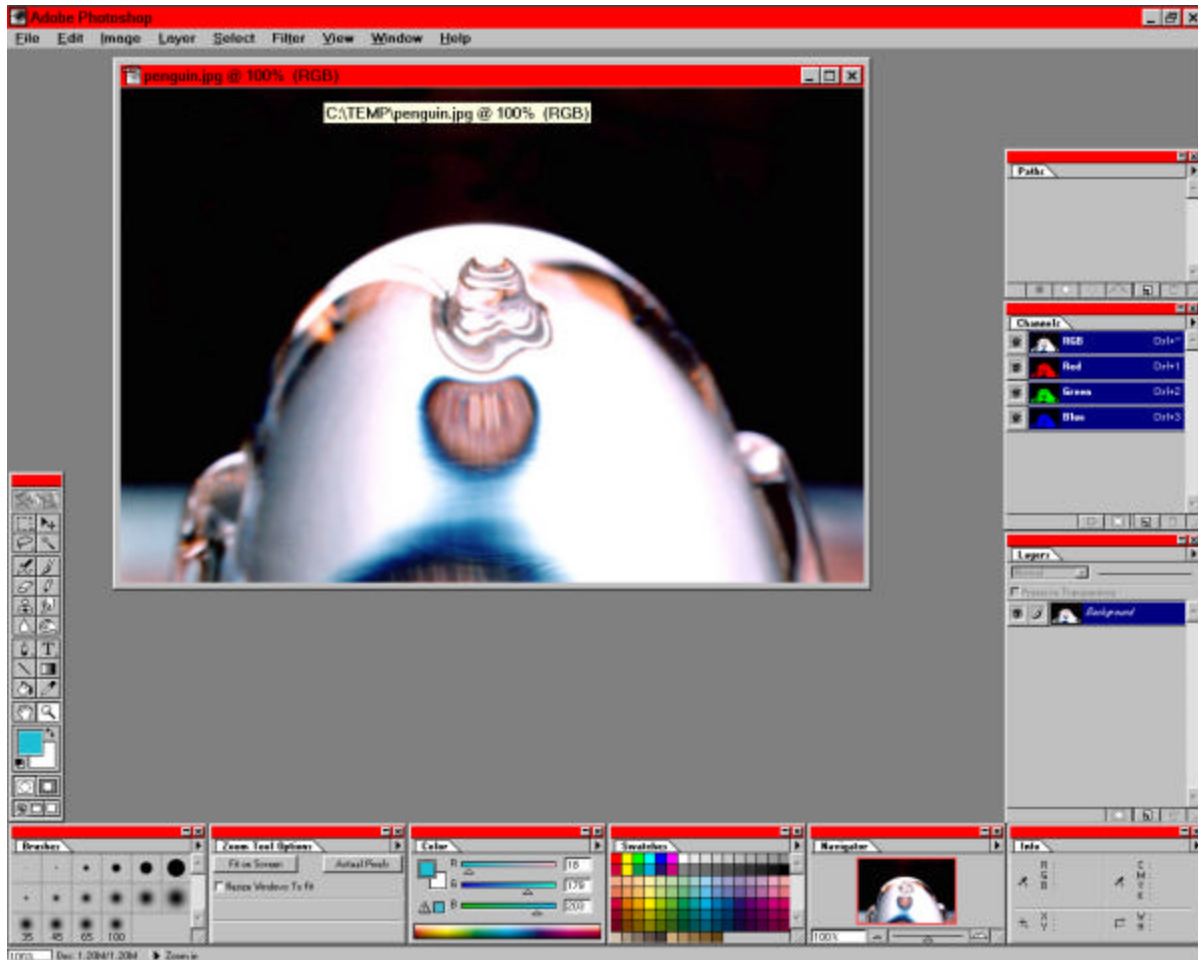


**Figure 11** (penguin\_gate.bmp)

This is the image I use on the top of my home page. It's really simple how it was done, but it has the advantage of starting from an interesting picture. I did all of the work on this image using Adobe Photoshop 3.0.5 <http://www.adobe.com>. The following screen shots are taken from a recreation of the process using Photoshop 4.0. Essentially, all I did was apply a succession of Photoshop filters to the original image.

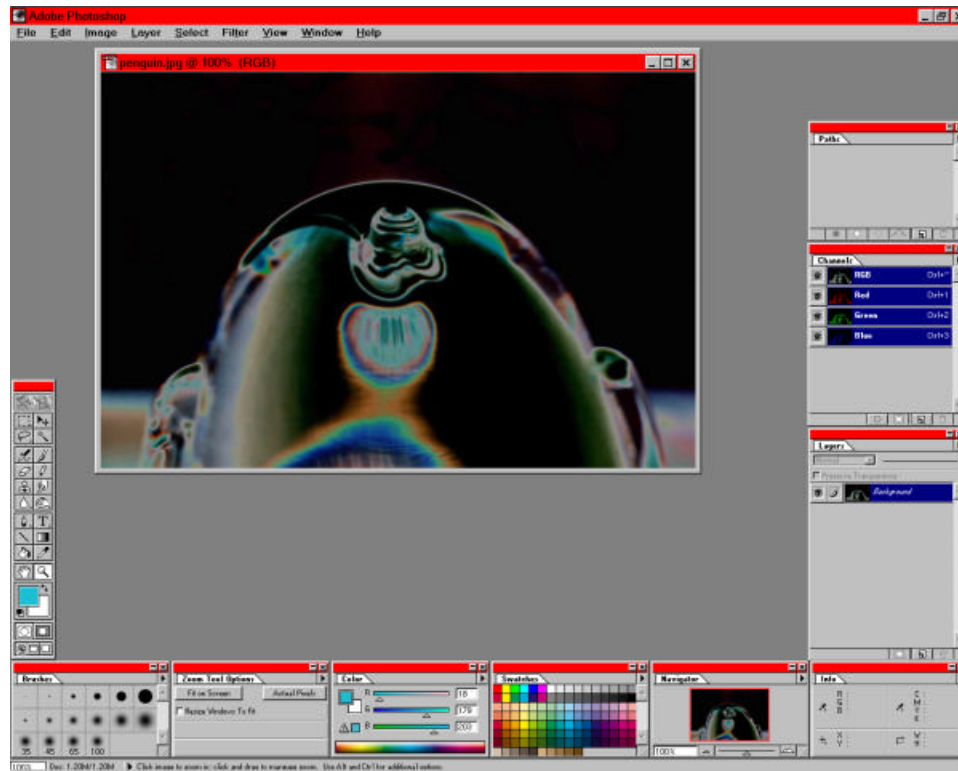


**Figure 12** shows the starting image. Pretty weird and abstract, right? This is a close-up of the top half of a lead crystal penguin. The beak and the wings are really in the glass, but the rest of the image is just reflected and refracted light coming through the window above and behind the glass penguin. I spent a lot of time moving the camera around until it looked like there was a mouth and eyes.



**Figure12 (psb1.bmp)**

I wanted a weird, spacey effect so I thought about it and chose the solarization filter that comes with Photoshop. It has no settings and I applied it to the entire image. The end result is **Figure 13**. I then tried jazzing it up with more special effects. I have a demo filter from Andromeda that takes an image and replicates copies around the central image. You can buy photographic prism lenses to get this effect for a regular 35mm camera.



**Figure 13** (psb2.bmp)

The filter's interface looks like Figure 14. As you can see, there are a wide variety of settings to choose from. Except for the positioning of the center and the width of the segments, I left the other settings alone. Once the filter is applied, you get Figure 15.

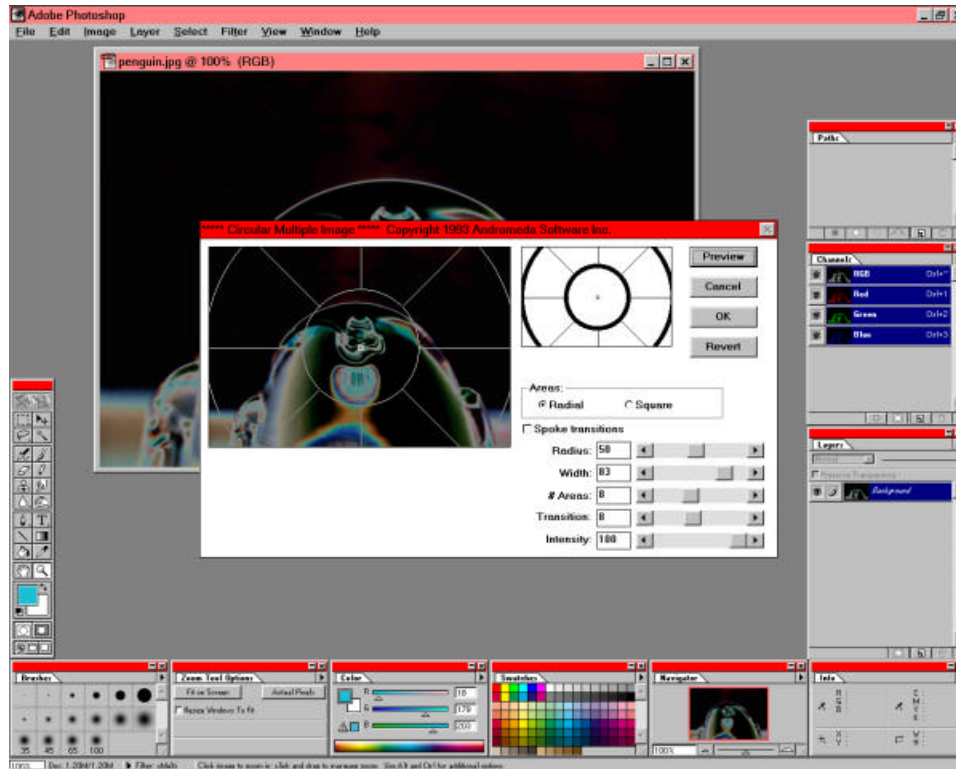
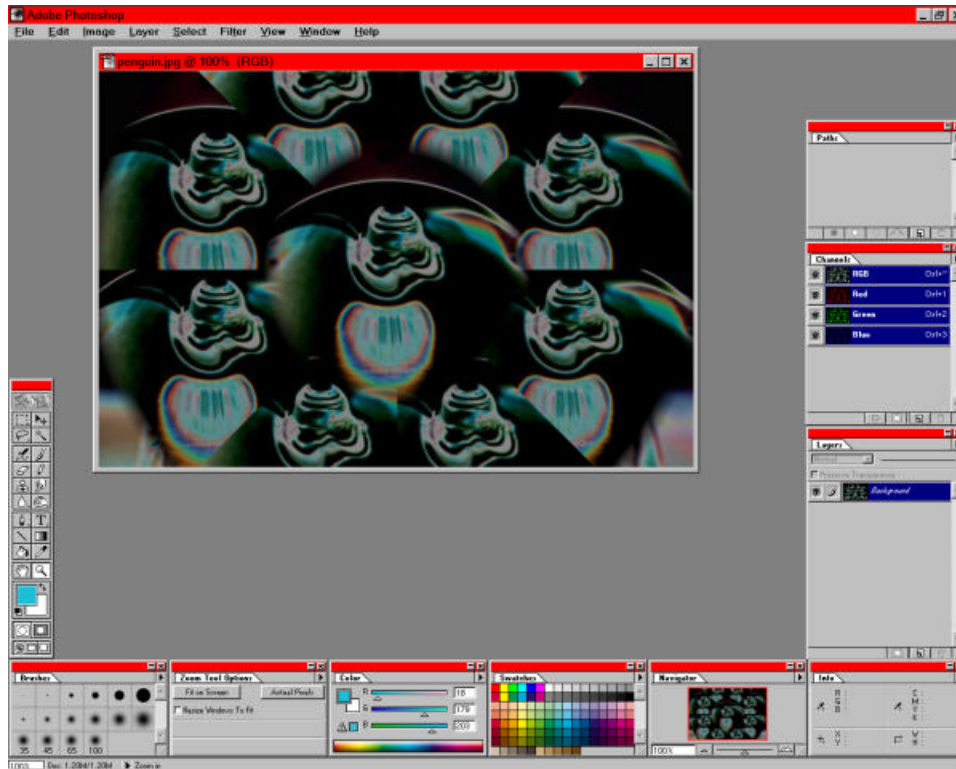


Figure 14 (psb3.bmp)

**Figure 15** (psb4.bmp)



**Figure 15** The picture is nearly done. This image has more segments than what I actually used on my web page. What was missing was something that created a focal point in the picture. Since it was to be the gateway to my home page, it had to look like there was something to go into. What better way than to simulate a light streaming out through an opening? Photoshop as a lens flare filter that adds what looks like an out-of-focus bright light to an image at any spot you want. Figure 16 has its interface.



I chose a light that was fairly bright and narrow, but had effects around it that looked like glare. We're now used to seeing things like that in movies for bright objects seen from a dark place, so I used that effect. The final result for this image is in Figure 17.

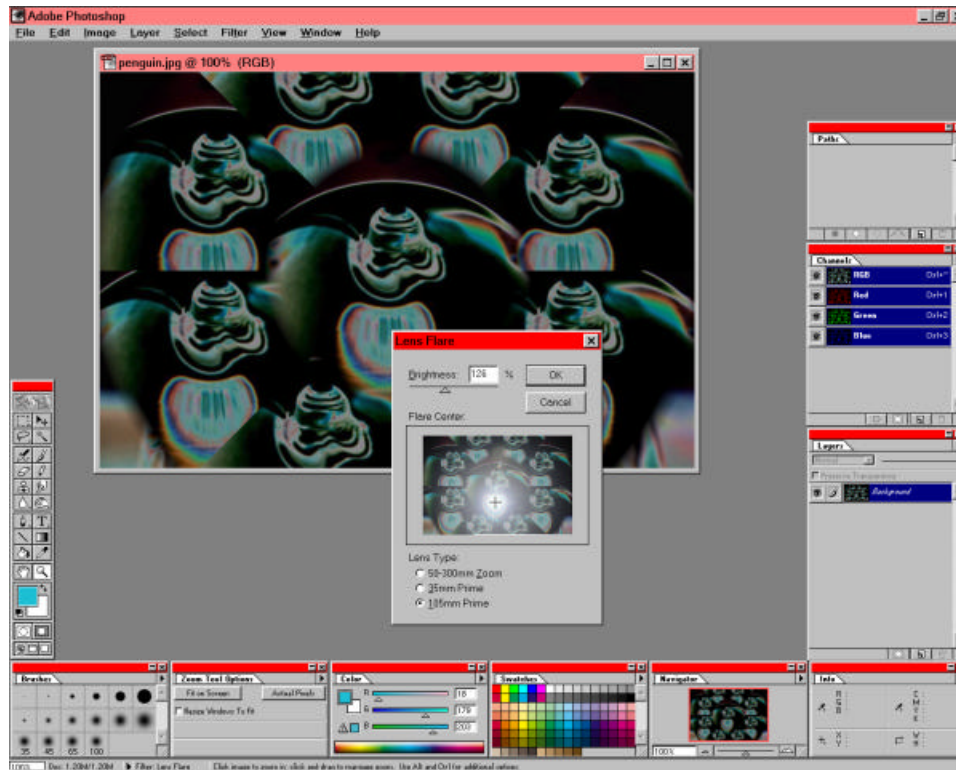


Figure 16 ( psb5.bmp)

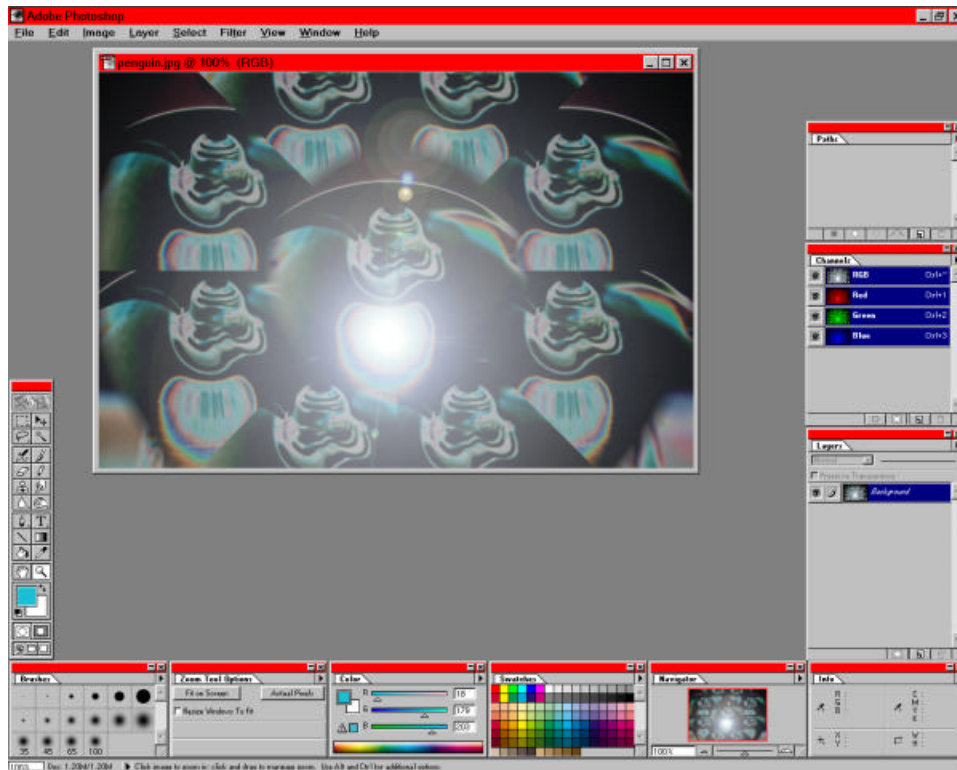


Figure 17 (psb6.bmp)

## Grand Canyon and Area

### Shaded Relief Map



Figure18 (gc.bmp)

highest with shades of gray in between. I imported and exported all of the data I had for the Grand Canyon area in carefully selected segments so that I could join them together seamlessly in a paint program. This requires a feature that only available in VistaPro 4, still in beta. Once exported, I loaded each of the pieces into Photoshop and joined them together into one large image there. Figure 19 is the resulting 256 color gray scale image, after I reduced it for this article.

This picture was produced with only two tools, Adobe Photoshop and VistaPro, but there were lots of manual steps involved. Eventually, I had to write a program to take care of one group of steps, but even so, it took about an hour to build this image even though I had done several others just like it recently. To be able to build this image, I took advantage of a feature of VistaPro that most people don't use. After importing a landscape into VistaPro, you can export it as an image. The image would have black for the lowest elevation and white for the

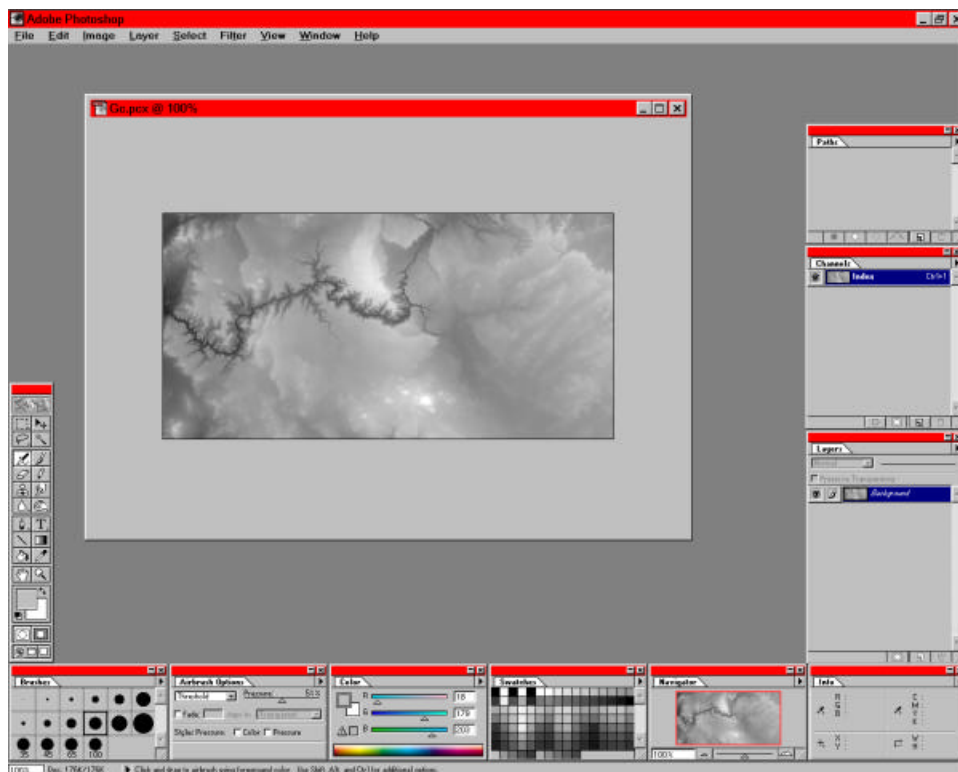


Figure 19 (psa1.bmp)

You can see the heights and depths fairly clearly, but it really looks like an X-ray of the landscape. The picture is interesting, but not interesting enough. The next step is to apply a color map to the image so that different elevations are color coded. If you look at some good topographic maps, you will see that different colors are used for different elevations and that they blend together to produce a pleasing image. Figure 20 shows Photoshop with the custom color palette I used loaded.

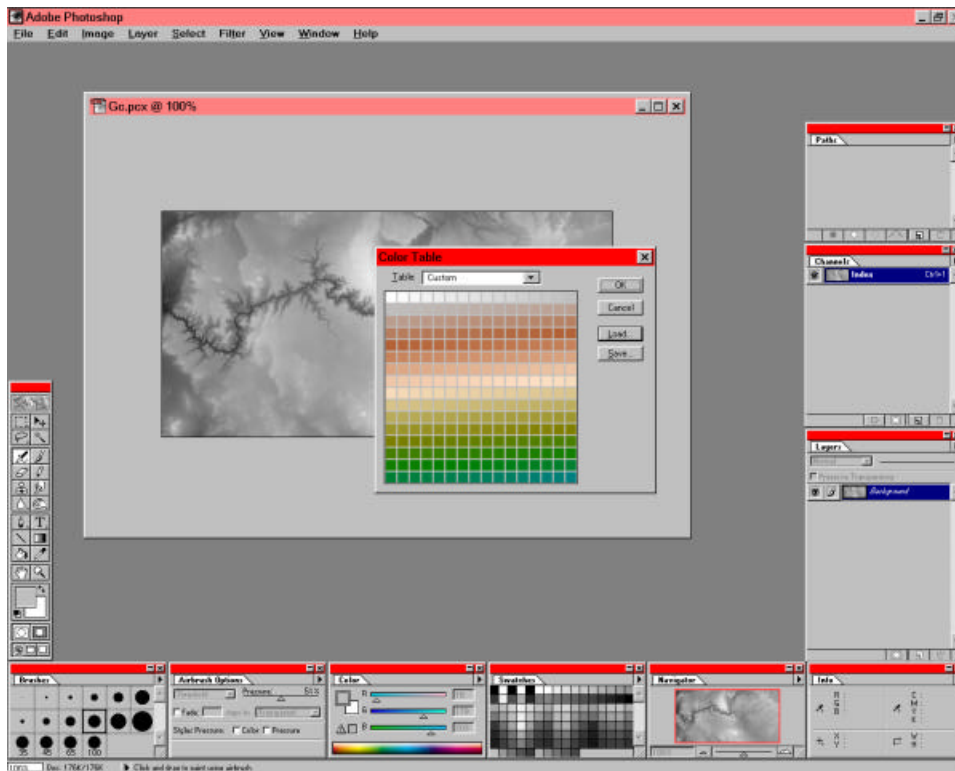


Figure 20 (psa2.bmp)

At one time, I had to manually come up with each of the colors needed for shading a landscape. It was hard enough and tricky enough so that I finally gave up and wrote a program to generate all 256 of the colors I needed. Before the program, it used to take me about 4 hours just to come up with a good looking color table. With the program, it would take about 30 seconds. This color table had already been tuned for this image because I have used it before. The result of applying the color table is Figure 21.

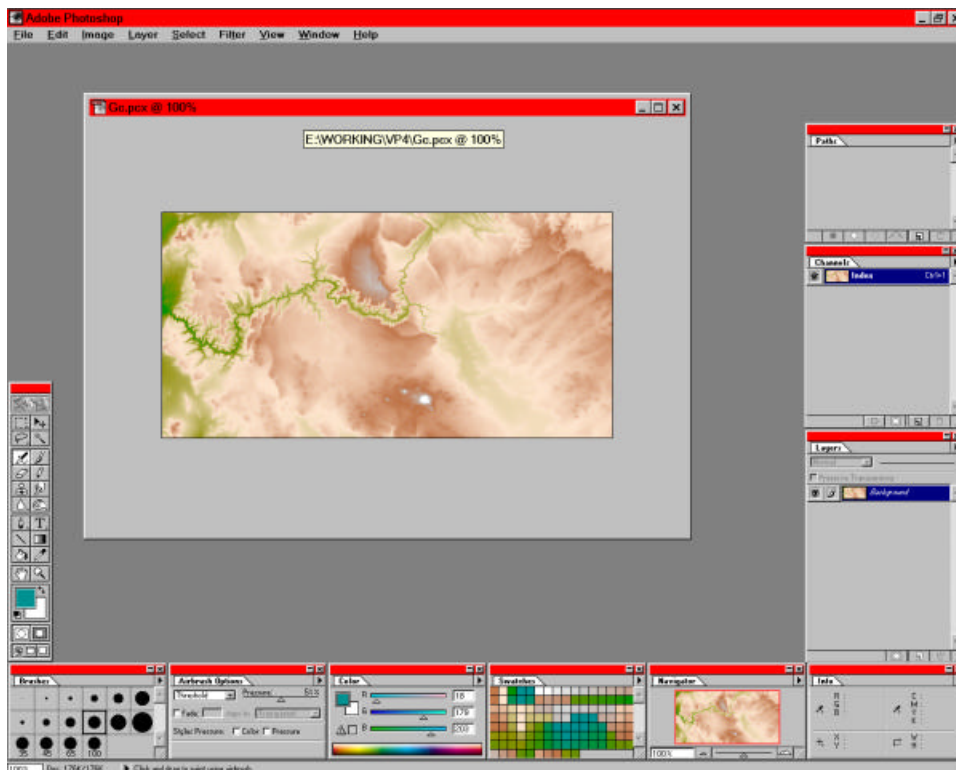
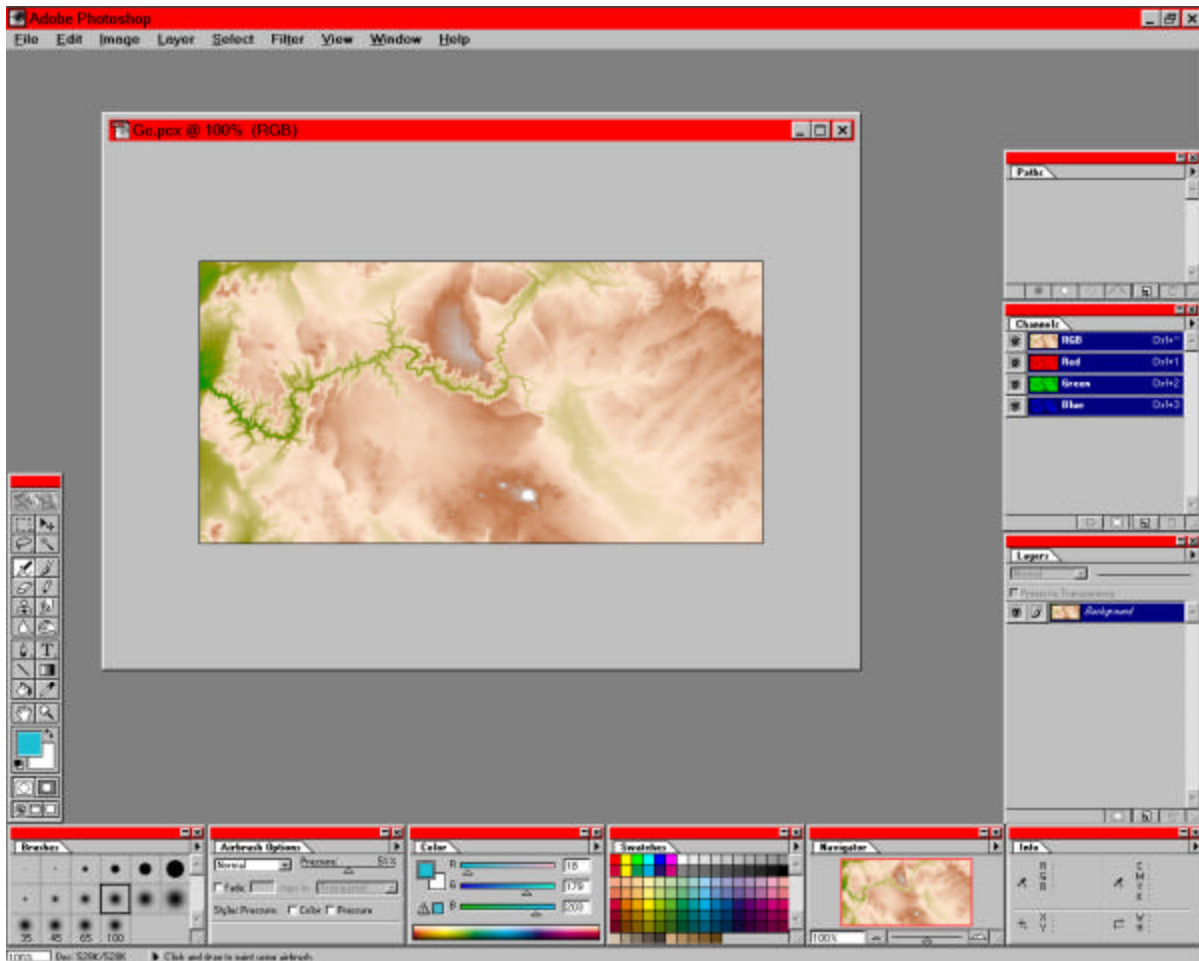


Figure21 (psa3.bmp)

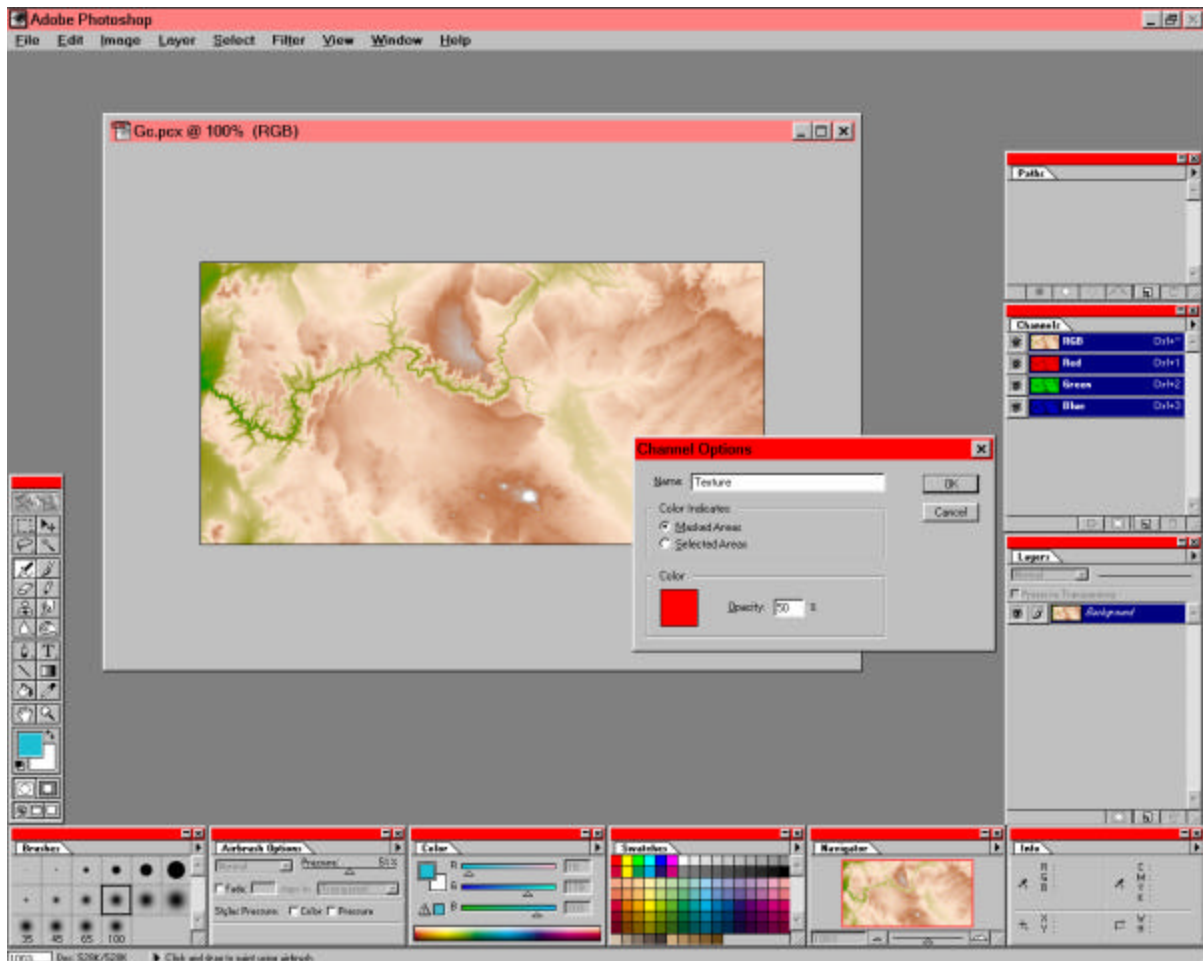
The image looks pretty nice, but if you were to look at the full sized image on a monitor, it would still look flat. It is a color-coded X-ray and nothing more. There was something more missing that was needed to emphasize the sense of depth of looking over a real landscape. That missing ingredient is shaded relief. Photoshop has a filter that can be coerced into doing shaded relief of an image. It works only on 24-bit RGB-encoded images and works best with a texture channel that encodes exactly how high each point of the image is. The next steps in Photoshop are to get to the state where the filter will work.





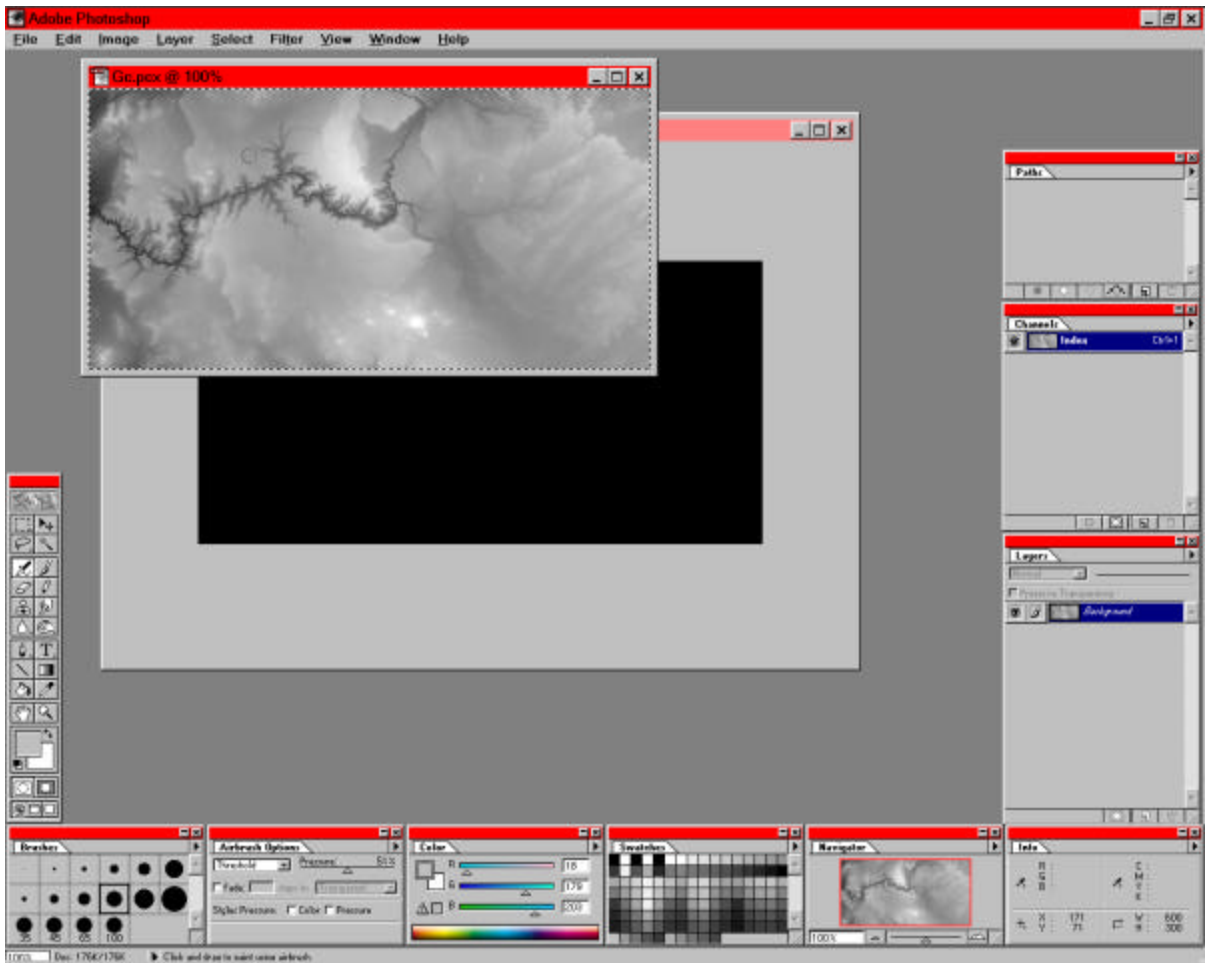
**Figure 22 (psa4.bmp)**

This is what the image looks like in Photoshop when you convert the original 256-color colorized image to RGB color. The channels palette now lists 4 channels instead of just one. Compare the right part of the screen shots in Figures 21 and 22.

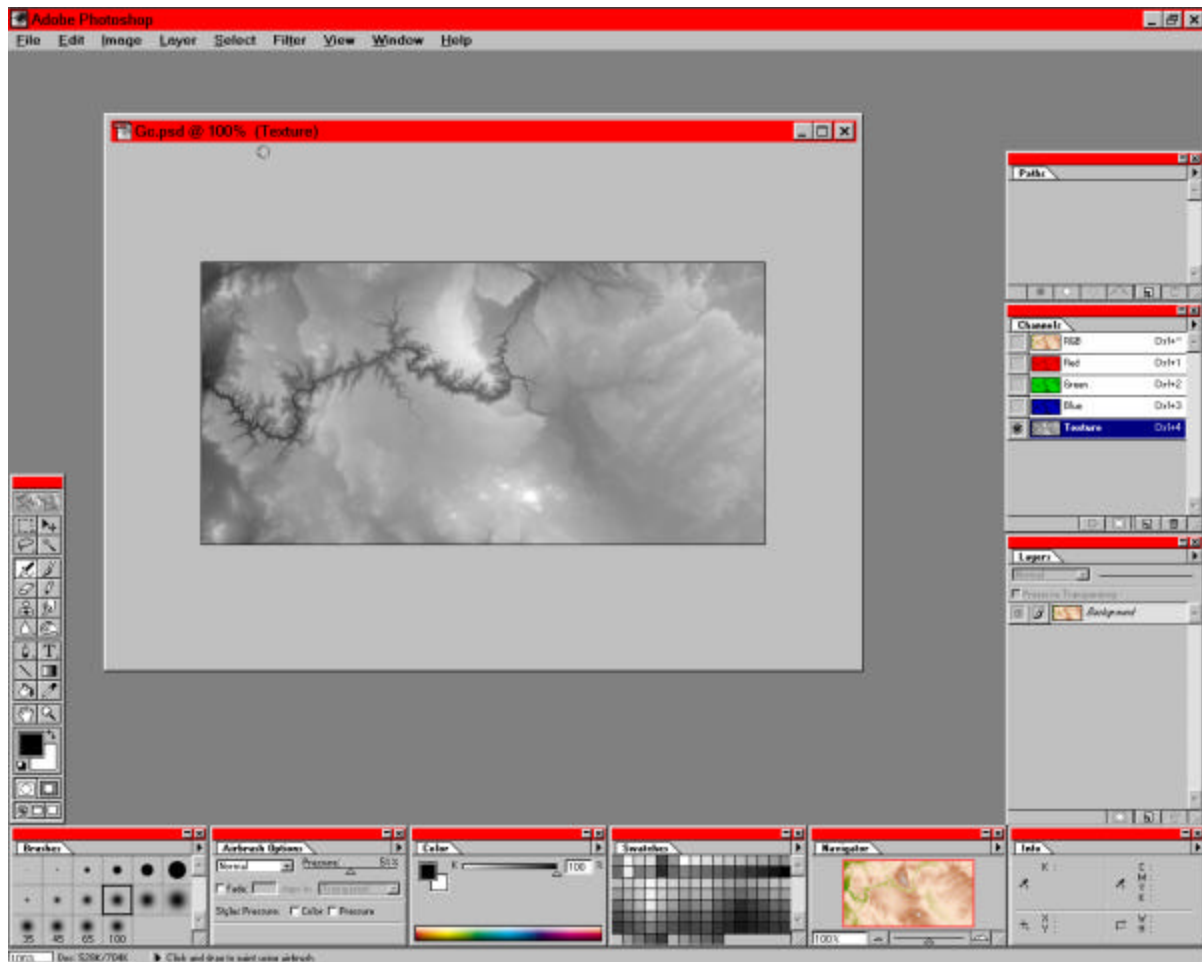


**Figure 23 (psa5.bmp)**

The next thing to do is to add the texture channel. This texture will tell the lighting filter how high each part of the image is. The obvious approach is to use the original gray scale image since that is exactly what the filter needs. The lighter the gray, the higher things are supposed to be. The texture is an image channel, in Photoshop parlance, and so, using the channel palette, I created a new texture channel and selected it, as shown in Figure 23.



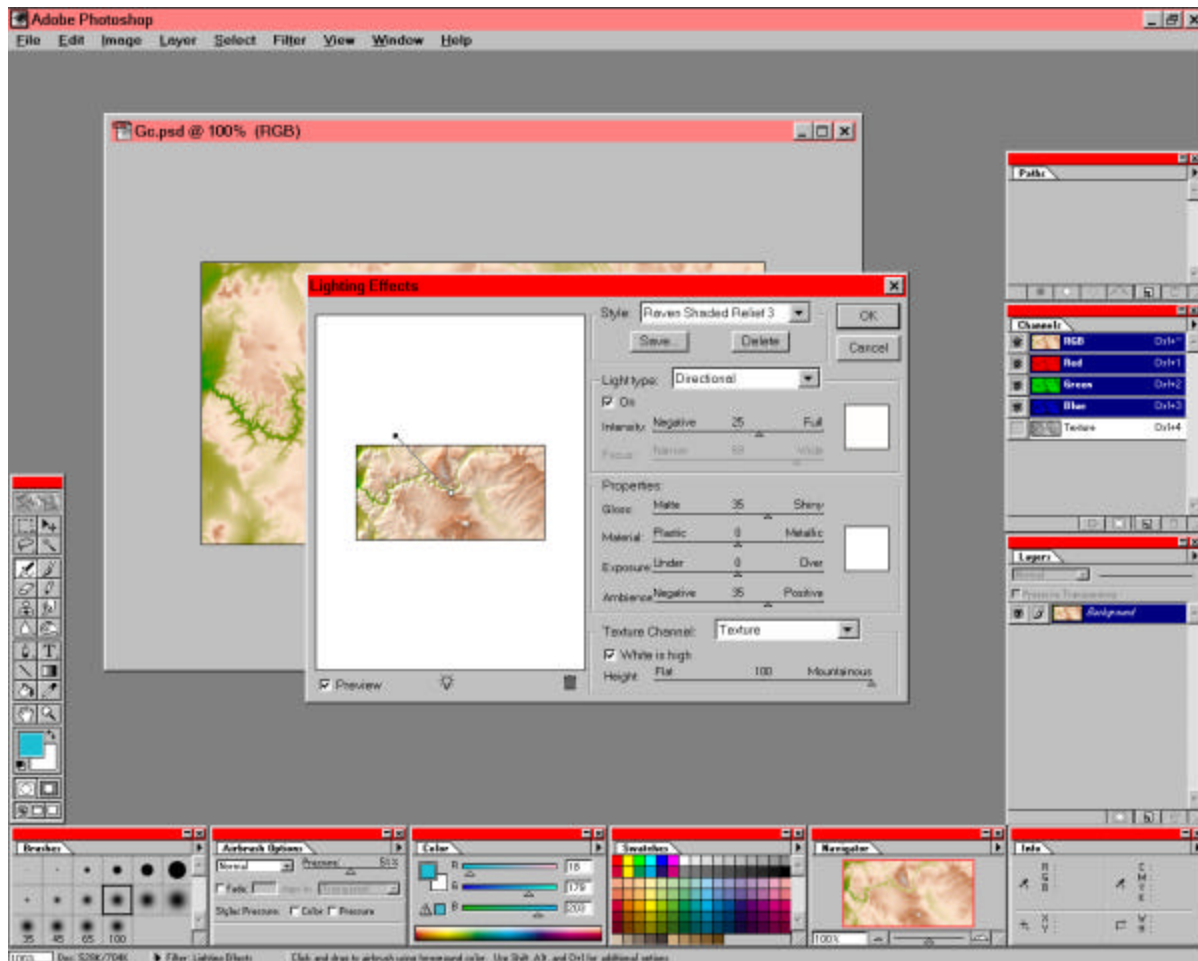
**Figure 24 (psa6.bmp)** shows the new channel, blank because I haven't put anything into it, and the original gray scale image. I just copy and paste the old image into the texture channel of the new image.



**Figure 25 (psa7.bmp)**

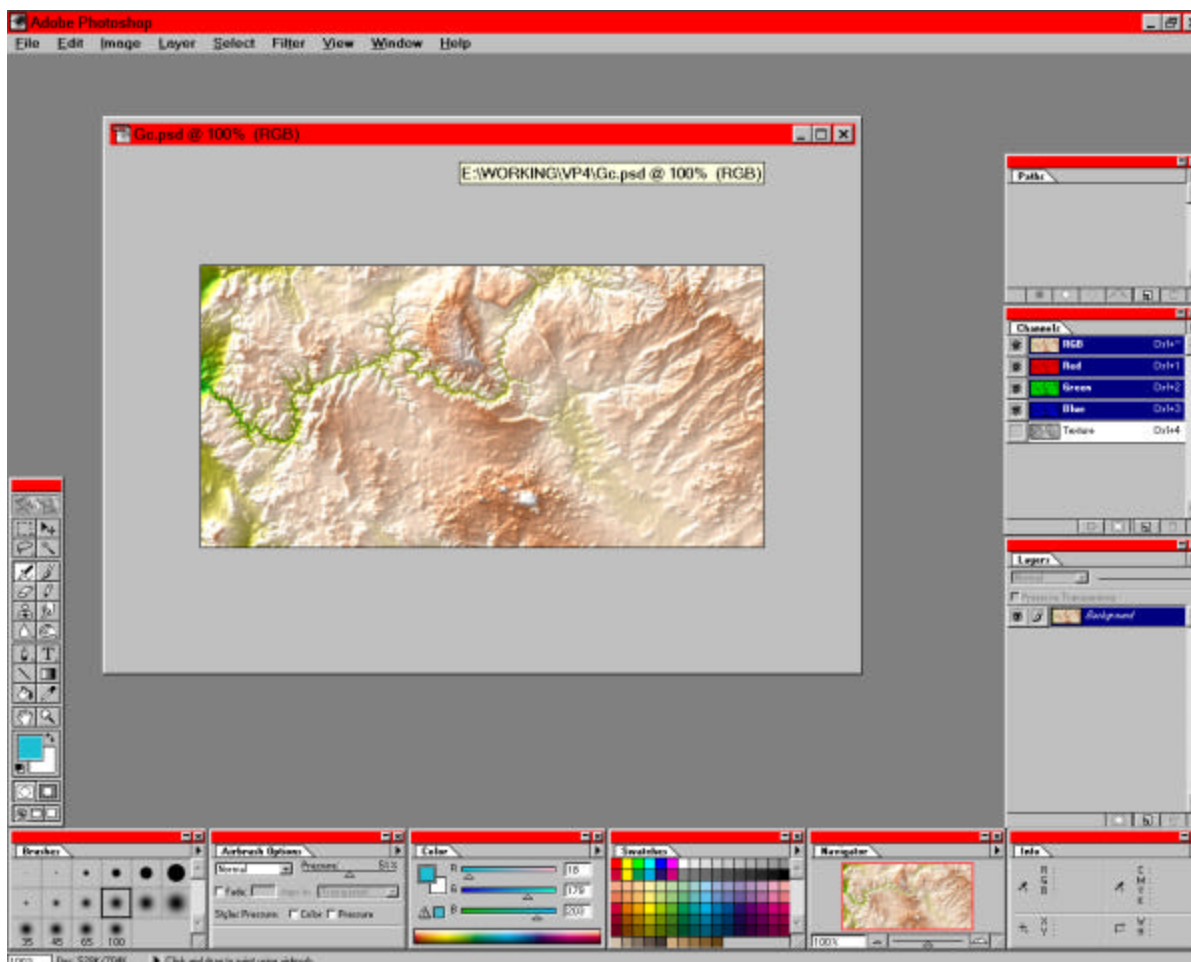
This leaves me with Figure 25, the complete image in the Red, Green, and Blue channels, and the height data coded in the texture channel. Now that I have all of the information needed to add the shaded relief to the image, I need to set up the filter correctly. Figure 26 shows the filter interface with the image in the background.





**Figure 26 (psa8.bmp)**

I've already selected a saved setting of lighting that I use for shaded relief maps. It specifies a broad light from the upper left and the texture channel to use to determine heights of points on the image. I also have surface characteristics and roughness selected. The settings depend strongly on the size of the image in pixels, so if you want to do this, experiment. Unfortunately, the lighting effects filter in Photoshop is one of the slowest. You'll just have to be patient. Figure 27 shows the results of applying the filter to the entire image.



**Figure 27 (psa9.bmp)**

A More Detailed Look You can visit my Web page at <http://acheron.cilia.org> to see more examples of my computer artwork and other types of imaging. The Domain Name Server of the .org suffix is slow and so you must try to get to the site when the Internet isn't very busy. You'll also have to be patient because the page was designed and ran for a long time on a Wide Area Network where connection speeds are significantly higher than a modem or many ISDN connections can support. The page states that you need at 256 colors and your display set to 800x600 to see it well, but the pages were actually designed for 1024x768 or higher screen resolution with 16-bit (64K) color. What you will see with fewer colors is a less true renditions of my original images. I normally run my system at 1280x1024 with 24-bit color. Any less and I can't tell if my images are rendering the way I want or not.

*Herb Chong has made a fantastic presentation on these pages! Quite aside from his great talent to teach, that is, to talk about a complicated procedure in quite simple and easy to understand language, he has produced images that are, indeed, art!*

*From the inception of the magazine Herb Chong has brought his considerable programming strengths to our enterprise. His understanding of computer architecture along with intellectual integrity has given the magazine a level of credibility that would have taken longer to achieve. No matter how much he is pressed for time, he always has time for us!*

*We at [WindoWatch](#) are very proud to be associated with him.*