



[Tileable Textures](#) - [Easy Bump Mapping](#) - [Intermediate Bump Mapping](#) - [Advanced Bump Mapping](#)

DISCLAIMER:

This is not a tutorial on how to use any of the Halo Editing Kit components or 3DS/G Max. You will need a basic competence with these tools in order to follow this tutorial. If you are having trouble with the HEK, read the documentation and tutorial that came with it.

IMPORTANT!

In order to follow this tutorial you will need a copy of The GIMP. Stop laughing! It's a powerful image editing program that is free to download. There are two reasons why I'm writing the tutorial using this program:

1) So that everyone can use the same instructions without having to spend time figuring out what the equivalent commands are on their favourite Image Processing package.

2) I make extensive use of the 'Offset' command to create seamless textures and bump maps. It's essential to this tutorial and I don't know of any other program that has it. So if you're using Photoshop and you want to know how to get your texture to 'Offset' like I've shown you, don't email me, because I don't know.

Anyway, this tutorial is quite in-depth, so don't worry if you haven't used The GIMP before, there'll be plenty of instructions and lots of helpful pictures. You can get the latest version of The GIMP [here](#). You'll also need to download and execute the GTK+2 program at the top of that page *before* installing The GIMP, or it won't work.

CONTACT

If you have any questions, spot any mistakes or have suggestions on how to make the tutorial better, [email me](#).

Introduction

Now that I've got that out of the way, welcome to The Halo CE Bump Mapping Tutorial. I hope to ease you through the various stages of getting bump maps working in-game and have provided three varying degrees of difficulty to work through. If you're unfamiliar with how to make tileable textures (ie. seamless when you repeat them) then you should do that tutorial first, since it's not only invaluable when it comes to making your own textures, but the techniques used in it will appear in all of the other stages of bump mapping.

You don't have to read all of this introductory section, but it will help you to understand exactly what it is that you are doing.

What is bump mapping?

Simply speaking, bump mapping is a way to make flat surfaces react to light as if they were three dimensional. This is a fairly old technique in computer graphics, but it has only recently been used in games due to advances in video card hardware. The reasons for using bump mapping are many, but suffice to say that it helps make games look a lot more realistic.



Without Bump Mapping
(click to enlarge)



With Bump mapping
(click to enlarge)

How does it work?

The way this is achieved is to overlay a grayscale image onto a surface. In this grayscale, white is equal to high (sticks out the most), black is equal to low (the deepest part) and gray is somewhere in between. The game engine will consult this grayscale image when a dynamic light is activated. It then shades the texture according to the angle that the light is hitting the surface.



Bump Maps React To Light From Different Angles
(click to enlarge)

What is a dynamic light?

Basically, this is a light source that is not part of the original lightmap created for a level (in Halo this lightmap is called 'Radiosity'). The original lightmap is static and does not change. Dynamic lights, however, can:

- 1) *Change in both colour and intensity*
- 2) *Move around*
- 3) *Affect bump mapping*

Without a dynamic light source affecting it, a bump map is totally invisible. This is because the texture it would shade will appear completely flat. Dynamic light sources specific to Halo are:

- 1) *The flashlight*
- 2) *The headlights from both the Warthog and the Scorpion*
- 3) *Muzzle flashes from various guns as they are fired*
- 4) *Grenades, Fuel Rod Gun and Plasma Pistol Overcharge projectiles*
- 5) *Explosions from the Needler, Rocket Launcher and Grenades*
- 6) *Special kinds of scenery objects*



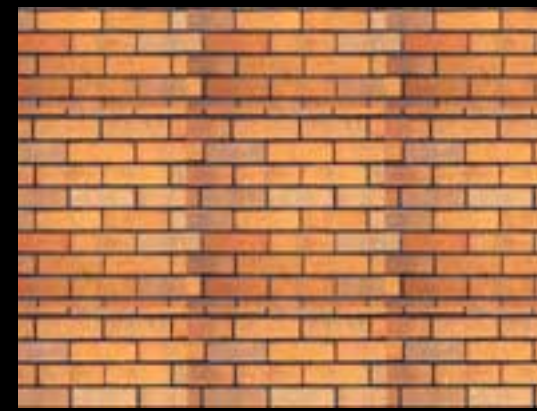
[Introduction](#) - [Easy Bump Mapping](#) - [Intermediate Bump Mapping](#) - [Advanced Bump Mapping](#)

Tileable Textures

This section deals with making seamless textures from images that have patterns but can't be used as tiled texture maps because their edges don't match. It will be worth working through this part of the tutorial even if you don't plan on making your own textures, because it introduces the power of The Gimp's 'Offset' command, as well as explaining how to use the Clone Stamp tool. We will also be using the finished texture that you make in this section to create a bump map in 'Easy Bump Mapping'.



Original Non-Seamless Texture
(click to enlarge)



Bad Tiling Because Edges Don't Match
(click to enlarge)

1) The first thing you need to do is download the Original Non-Seamless Texture above (right click on it, then select 'Save Target As') and save it into the bitmaps folder of your map directory. For example, if you were making a map called 'Box', you would save the picture in:
HaloCE\data\levels\test\box\bitmaps.

2) Load the texture into The GIMP using the *File > Open* menu at the top of the main window.

3) As you can see from the second picture above, the edges do not match and so the texture looks terrible when it is tiled. What we need to do is cut out an area where we know the texture repeats itself. Looking at the image, we can see that the pattern is repeated for every one brick width horizontally, and every two layers of brick vertically.



Area Of Pattern Repetition
(click to enlarge)

While we could easily make a tileable texture from just one of those red squares, this is something that you will usually want to avoid doing. The reason for this is that when you restrict your texture to such a small sample, it lacks variety when it is tiled. In short: it looks fake and unnatural.

So, we will want to select a large area of pattern repetition using the Rectangular Region tool in the top left of the toolbox, as shown below.

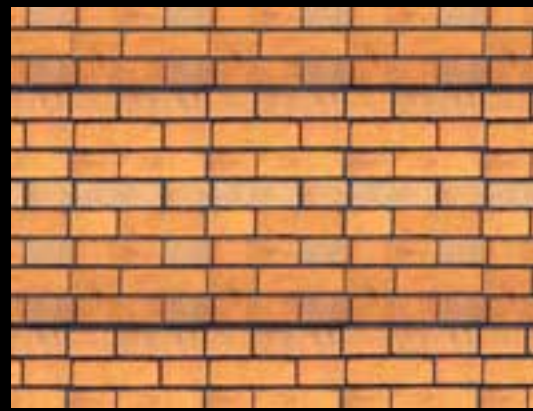


4) Select an area of the image like that indicated in the example picture below. It doesn't matter if you don't get it 100% correct, we will be editing the edges later on anyway. You may want to zoom in for better accuracy (the Zoom tool is directly below the Rectangular Region tool).



Select An Area Similar To This
(click to enlarge)

Note that I have avoided the black line of mortar in the top of my selection area, but included it at the bottom. Likewise, I have avoided the mortar lines at the left hand side, while including them on the right. If you don't do this then your pattern will have both sets of mortar lines when it is tiled. This results in two black bands and a number of smaller bricks. Not good.



Doublebanding: Rubbish
(click to enlarge)

- 5) Copy the selected area of the image by selecting *Edit>Copy* from the menu above the image.
- 6) Create a new window by selecting *File>New* from the menu, then hit OK.
- 7) Paste the selected area of the image into the new window by selecting *Edit>Paste* from the menu.
- 8) From the menu above the image, select *Layer>Transform>Offset*. Press the button marked 'Offset by (x/2),(y/2)', then hit OK. Your image should now look something like this:



The Offset Image
(click to enlarge)

What this has done is taken edges of the image and pushed them together so you can see how they line up. This is *exactly* how the edges of your texture would look when tiled in-game. You can see that although the seams match up pretty well, there are some very noticeable differences in colour.

Now we come to the real power of the 'Offset' command: we can edit the image in this state to make it appear as though it were a naturally tiled texture.

9) Select the Clone Stamp tool, then the brush selector option, then Circle Fuzzy (07), exactly as is shown in the diagram to the right. The Clone Stamp tool allows you to copy an area of an image and paint it wherever you like.

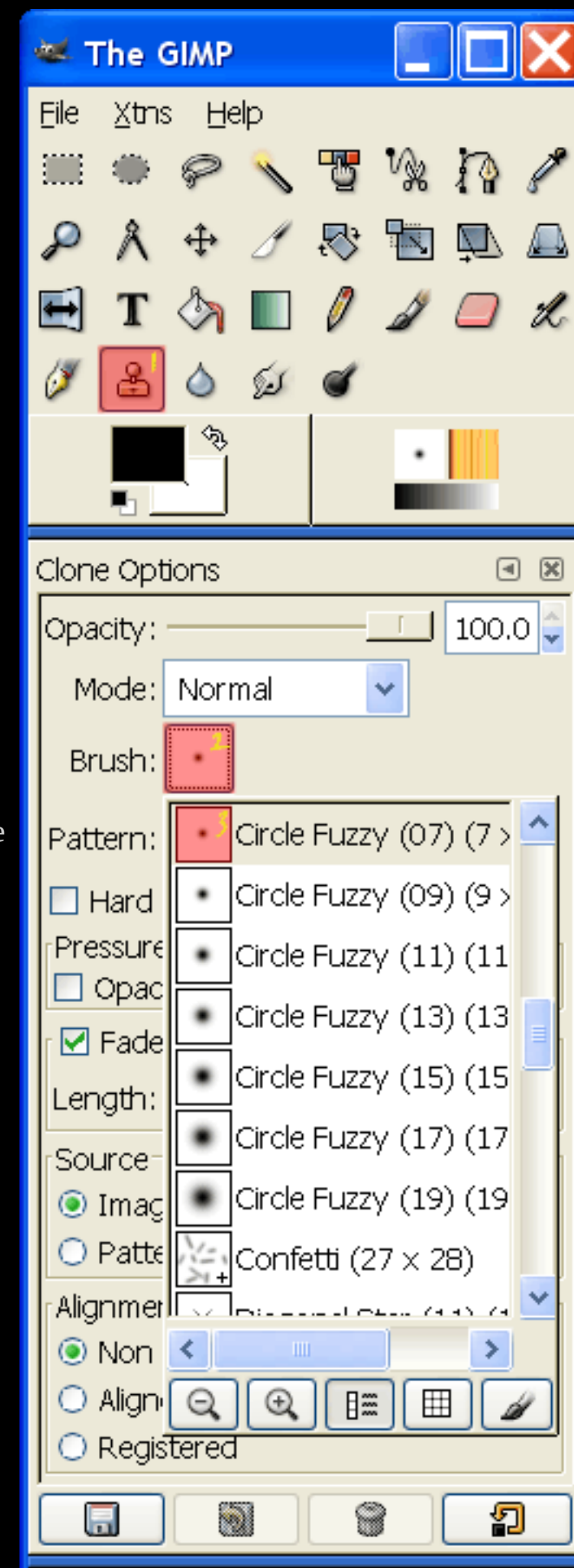
This is very useful for us, since it will allow us cover up the edges that don't match using other parts of the image. This makes it much more powerful than simple smearing or blurring and as a result will look much more convincing and natural.

You may actually use any size of brush you wish. When dealing with larger or smaller images than the one we have here, or if you just want to be more/less accurate, it's a very good idea. But for now, a size 7 brush will work well. Choosing a fuzzy brush is almost always the best option though, because when you paint with the Clone tool it will mix better with what you're painting on and won't leave any hard edges.

10) The Clone Stamp tool is very easy to use: just hold down the Ctrl key and aim the mouse at the point on the image that you want to copy *from*, then click the left mouse button. Now whenever you paint onto the image, it will be cloned from the point you just designated.



You should find it much easier and more accurate when working zoomed in on the image. You'll only be painting over a very small area anyway.



Holding down the mouse button while moving the mouse will also draw the area around the point you designated, up to the maximum number of pixels indicated in the 'Fade Out' section of the Clone Options (the default is 100 pixels). Letting go of the mouse button then drawing again will cause the cloning to reset back to the point you designated.

11) Use the Clone tool to get rid of the harsh edges by painting across them from different sides. You'll want to use a lot of different cloning points to get a natural non-repetitive seam. Check the picture below to see how the edges and colours are much smoother on the top half, which has been worked on, compared to the bottom, which has had nothing done to it.



Smoothing The Edges
(click to enlarge)

Don't be afraid to use clone points from anywhere on the image in order to get a good result. The only rule here is: *If it makes the seam look like a naturally tiled texture, it's fair game.*

The example here doesn't have a horizontal harsh edge because the way original image was cut, it lined up nicely on the top and bottom. If you have a horizontal edge, you would use the Clone Stamp tool on it just as you have done to the vertical edge.

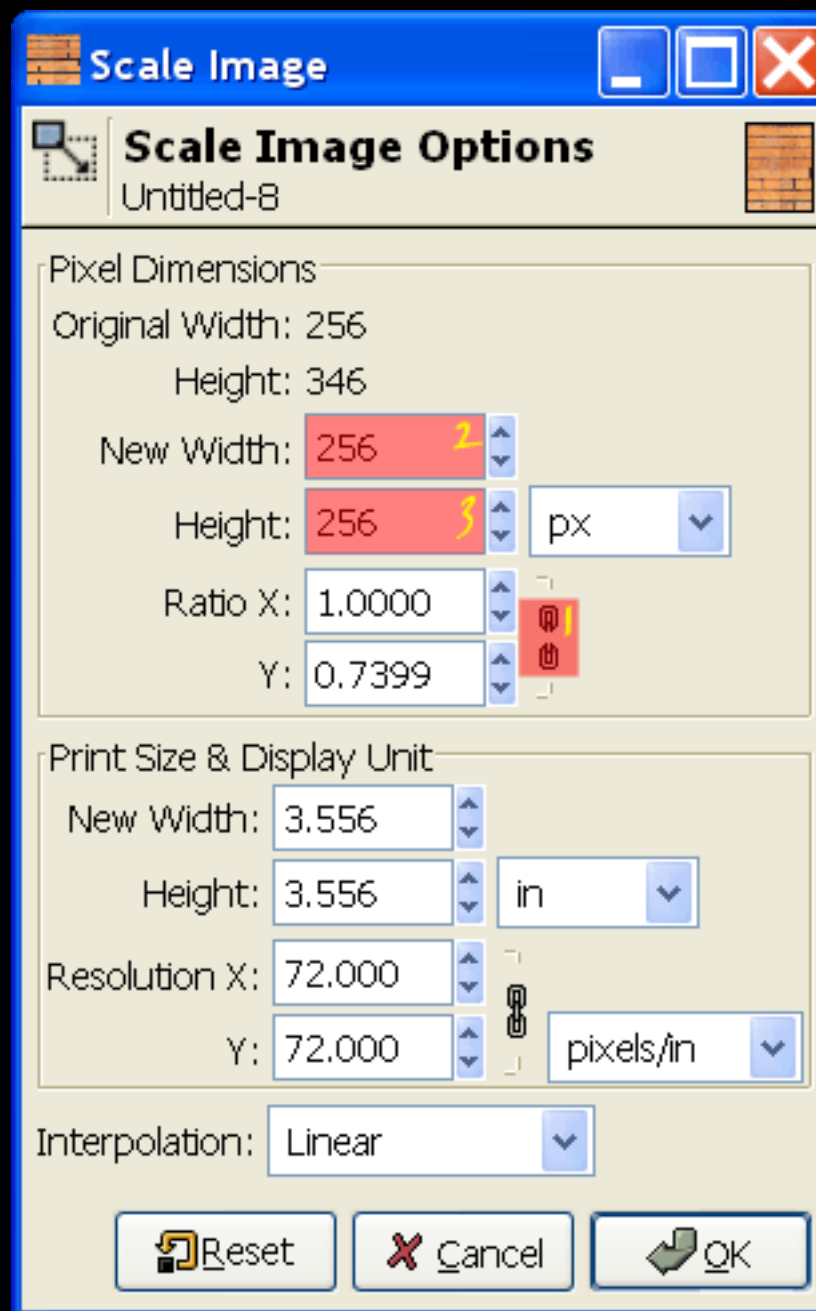


Most Textures Have Horizontal & Vertical Harsh Edges
(click to enlarge)

12) Once you're happy with the way the edges look, you'll need to return the image back to normal by using the 'Offset' command again. From the menu above the image, select *Layer>Transform>Offset*. Press the button marked 'Offset by (x/2),(y/2)', then hit OK.

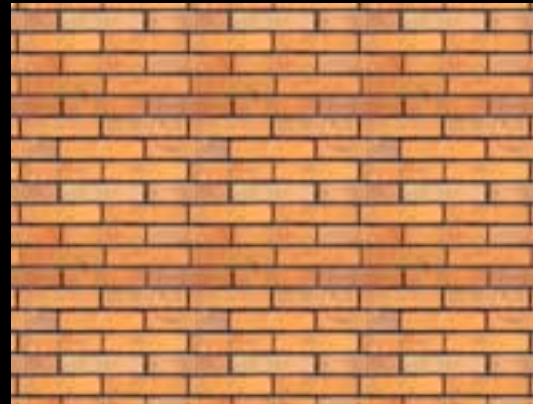
13) Before the texture can be used with Halo, the pixel size must be a power of two (ie. 2, 4, 8, 16, 32, 64, 128, 256, 512, 1024 or 2048 pixels in width or height). The image I have is 256x346. Yours may be different by a small amount, but that doesn't matter, we're going to be converting it to a 256x256 pixel texture.

From the menu above the image, select *Image>Scale Image*. In the window that pops up, make sure that the top chain link is broken (so that the width and height are not linked), then enter 256 into the New Width box and 256 into the Height box. Click on OK.



14) The texture is now ready to be saved as a .tif file (the Halo Editing Kit requires textures to be save as .tif files). If you've used more than one layer in your image (it tells you how many layers you're using in the blue window at the top of the image), then you'll need to merge them all using *Layer>Merge Down* before saving.

Save the file in the same directory that you loaded it from. You're now ready to create a basic bump map for this texture, so move on to [the next section](#).



A Successfully Tiled Texture
(click to enlarge)



[Introduction](#) - [Tileable Textures](#) - [Intermediate Bump Mapping](#) - [Advanced Bump Mapping](#)

Easy Bump Mapping

This section deals with bump mapping at its most basic. It will introduce you to a way of creating bump maps quickly and easily. It is a simple method that should work for a large number of textures. We'll also be detailing how to get your texture and bump map in-game, using Tool and Guerilla.

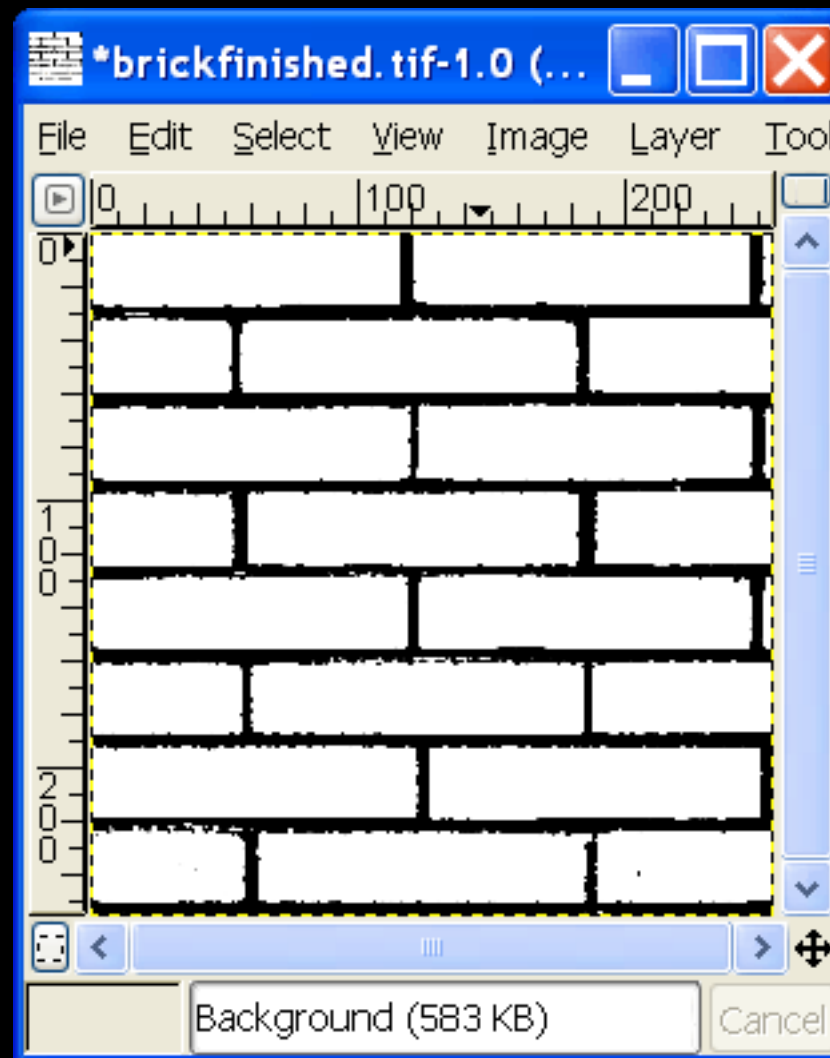
1) If you followed the previous Tileable Textures tutorial, load your finished brick texture into The GIMP. If you didn't follow that tutorial and the terms 'Offset' or 'Clone Stamp' mean nothing to you, please go back and work through it, because the techniques shown to you in that section will be essential to making bump maps.

If you already know about making tileable/seamless textures but didn't follow the previous tutorial, the finished brick texture can be downloaded [here](#). You'll need it to work through this section.

2) As you may know, bump maps are grayscale images where white equals high, black equals low and the various shades of gray representing the values in between. So, the first thing we need to do is convert the image to Grayscale by selecting *Image>Mode>Grayscale* from the menu above the texture.

3) You can see that the mortar lines around the bricks are a darker shade of gray than the bricks. This is handy for us because they are also the lowest points on a physical brick wall. So what we need is a way to separate the mortar lines as pure black and the bricks as pure white. This can be easily done using the Contrast and Brightness tools.

From the menu, select *Layer>Colors>Brightness-Contrast*. Move the two sliders around to the right until your image looks similar to the one below (the values that I used are Brightness: 23, Contrast: 127, but your image may look better at different settings).



4) The edges here are pretty harsh and won't look good when used as a bump map, so we will

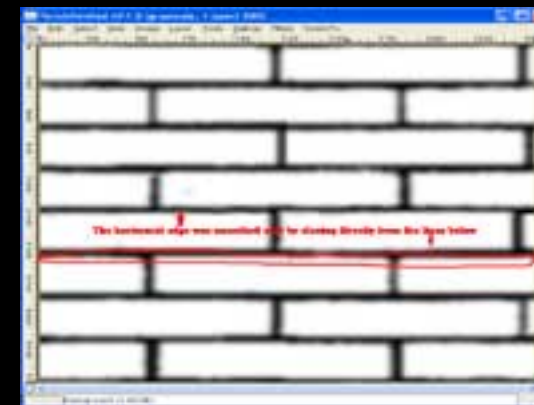
need to blur the image. From the menu, select *Filters>Blur>Gaussian Blur(IIR)*. A small window will pop up asking how much you want to blur the image. Enter 5.0 into both boxes (when using other images, you will probably need to blur by a different amount, but 5 will do fine for this example).

5) Offset the image by selecting *Layer>Transform>Offset* from the menu. Press the button marked 'Offset by (x/2),(y/2)', then hit OK.

6) The blurring has caused the image to be mismatched around both the horizontal and vertical edges, so you will want to use the Clone Stamp tool to smooth it all out, just like we did in the previous section. However, this time you will want to be much more specific about the points you clone from. Aim to smooth the edges by painting over them with the lines that already exist, as seen in the picture below.



Harsh Edges From Blurring
(click to enlarge)



Direct Cloning From Existing Lines
(click to enlarge)

7) Once you're happy with the way the horizontal and vertical edges look, return the image back to normal by using the Offset command again.

8) The bump map is now ready to be saved as a .tif file (the Halo Editing Kit requires bump maps to be save as .tif files). If you've used more than one layer in your image (it tells you how many layers you're using in the blue window at the top of the image), then you'll need to merge them all using *Layer>Merge Down* before saving.

Save the file in the same directory that you loaded it from and don't forget to give it a different name from the original texture, or you'll overwrite it.

9) Run Tool using the following command:

```
tool bitmap <path to your brick texture>
```

For example, if your map was named 'Box' and you had saved the brick texture as 'bricks', you

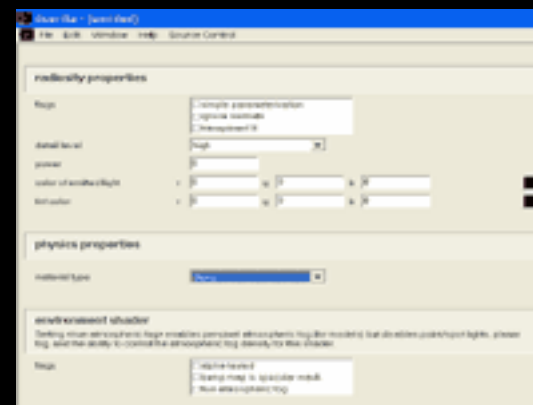
would type in:

tool bitmap levels\test\box\bitmaps\bricks

10) Repeat step 9, but this time for the bump map you just created. Don't worry when it gives you a warning message, Tool doesn't have all of the information it needs at the moment for the bump map, but we will be fixing this soon.

11) Open Guerilla and select *File>New* from the top menu. Select 'shader_environment' from the pulldown menu in the window that pops up.

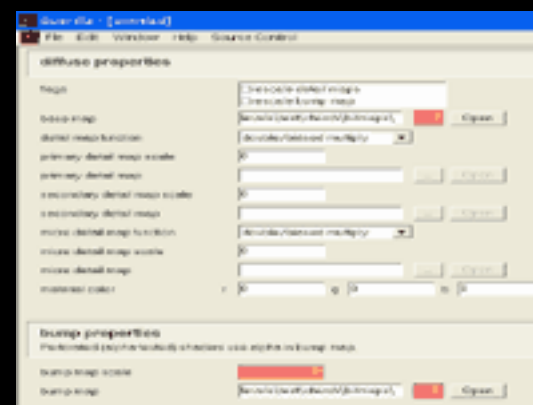
12) Our brick wall is made of stone so select 'stone' from the pulldown menu under the 'Physics Properties' section.



Select Stone
(click to enlarge)

13) Scroll down to the 'Diffuse Properties' section and hit the button with three dots next to the 'Base Map' tab. Within the window that pops up, locate where you saved the brick texture, click on it then hit 'Open'.

Under the 'Bump Properties' section, insert 1 into the 'Bump Map Scale' tab. Click on the three dots next to the 'Bump Map' tab, then point it towards your bump map.



14) From the top menu, select *File > Save As*. Rename the file so it matches the name of the original texture. For example, if the original texture was 'bricks', rename the file to 'bricks.shader_environment'. The location where you need to save the file is a little different from before so be careful; if your map is called 'Box', save the file to: *tags\levels\test\box\shaders*.

15) Now, click on the 'Open' button next to the 'Bump Map' tab, under the 'Bump Properties' section. This will open a new window which contains the data relating to your bump map.

16) Use the pull down menu under the 'Format' section to select '32-bit color'. Use the pull down menu under the 'Usage' section to select 'height map'. Still under the 'Usage' section, tick the box labeled 'disable height map compression'.

17) Scroll down to the section called 'Post Processing' and enter 0.2 into the 'Bump Height' tab. This controls the depth of the bump map and the number entered must be between 0 and 1. Zero equals no bump height and one equals total depth (so only use values in between). It's worth noting that higher values tend to not to look very good, so it's best to stick to numbers under 0.3.

18) Scroll to the very bottom section, named 'Bitmaps' and untick the box beneath it labeled 'compressed'.

19) Using the menu, select *File > Save*. Exit Guerilla.

20) Repeat the command in Tool that you used in step 9, but this time only on the bump map. You should see that there is no warning message now.

21) Use the brick texture on a surface in Max, just as you would normally, then compile the map using Tool. The texture should appear where you have placed it and will now react to dynamic lights with the bump map. (Note that bump maps are more visible in darker areas, so if the Radiosity power of the sky you are using is very high, it may be hard to make out the bump mapping of textures that are in direct sunlight.)



Finished Texture And Bump Map In-Game
(click to enlarge)

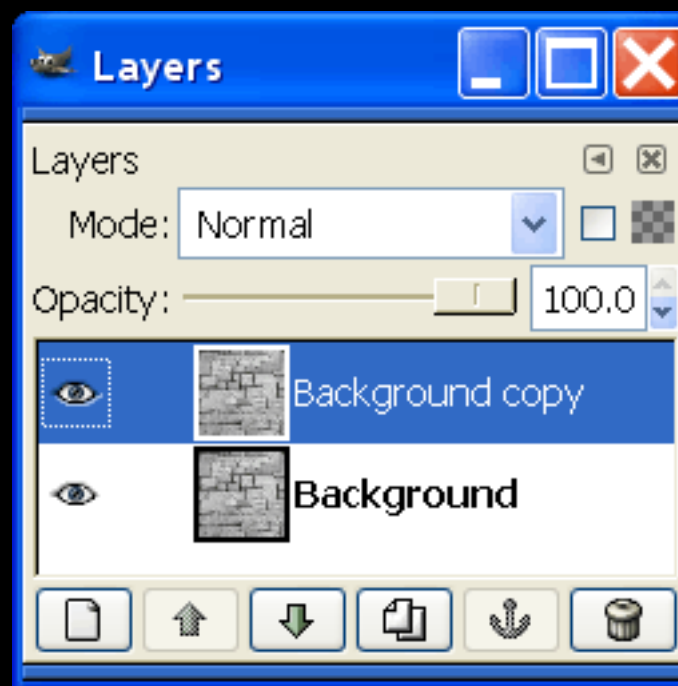


[Introduction](#) - [Tileable Textures](#) - [Easy Bump Mapping](#) - [Advanced Bump Mapping](#)

Intermediate Bump Mapping

In the previous section we separated the grayscale image into pure black and white to create a very simple bump map. In this section we will be using a variety of techniques to create something with much more detail. Before we start, download [this texture](#) and save it into the bitmaps folder of your map directory. For example, if you were making a map called 'Box', you would save the picture in: *HaloCE\data\levels\test\box\bitmaps*. Now load the image into The GIMP. It's already been made seamless, so there's nothing you need to do to it.

- 1) Convert the image to Grayscale by selecting *Image>Mode>Grayscale* from the menu above the texture.
- 2) Copy the image into a separate layer by selecting *Layer>Duplicate Layer*. Now activate the Layers window by selecting *Dialogs>Layers*, then select the top layer.



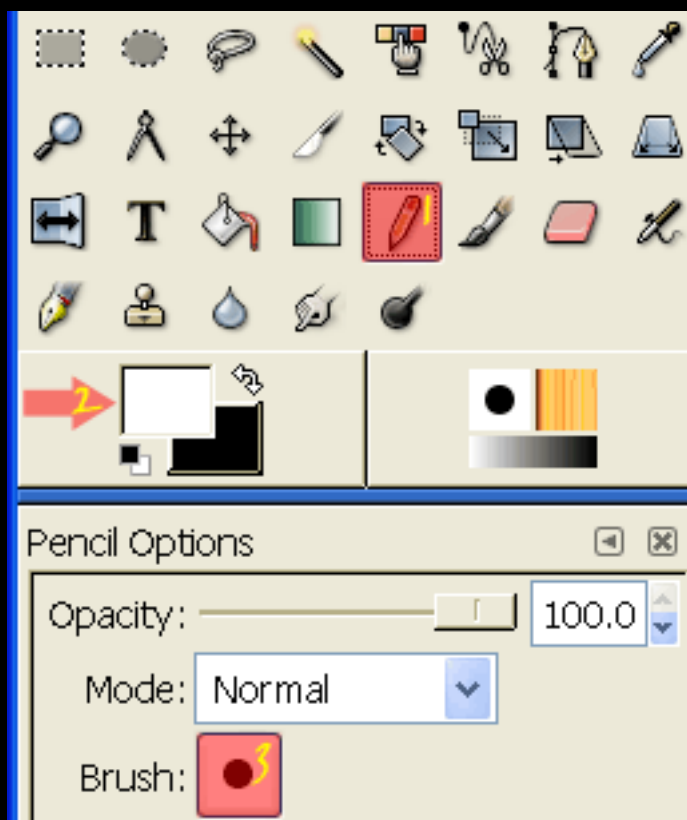
Select The Top Layer

3) From the menu, select *Layer > Colors > Brightness-Contrast*. Move the two sliders to the values of Brightness: 60, Contrast: 127. Your image will now look like the one below.



(click to enlarge)

4) You'll notice that this time there are a lot more patches of black within the bricks themselves. This is because the source image is more complex and has a greater variety of heights and patterns on the bricks. We will eventually represent this with the bump map, but for now we need to paint over these patches using the pencil tool. Make sure your paint colour is set to pure white and your brush type is a solid circle.



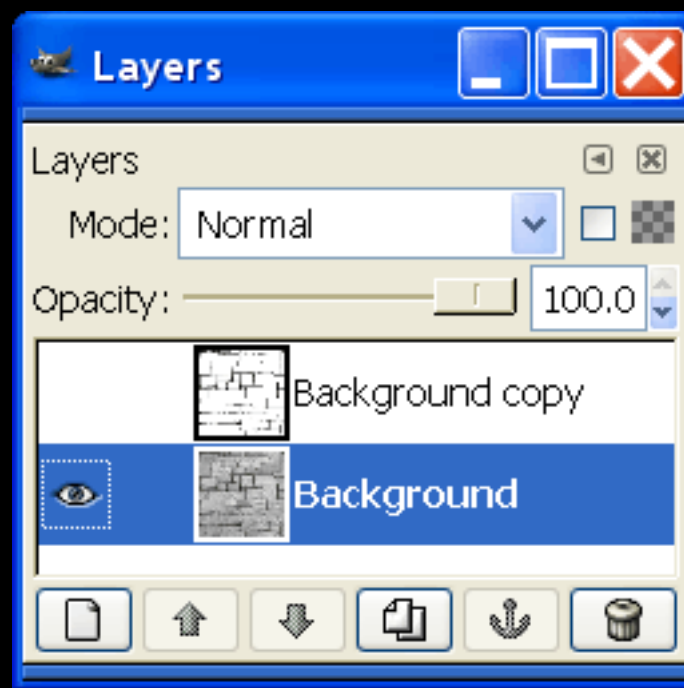
Paint within all of the bricks that have black patches until your image resembles the one below.



(click to enlarge)

5) From the menu, select *Filters>Blur>Gaussian Blur(IIR)*. A small window will pop up asking how much you want to blur the image. Enter 5.0 into both boxes.

6) Click on the Layers window and deselect the eye symbol to the left of the top layer. This makes the top layer invisible. Now click on the bottom layer so we can work on it.



7) You can see from this layer that the differing colours of the original image make some bricks noticeably darker than others. From what we know of bump maps, this will make the whiter ones stick out more. We obviously don't want this to happen since the bricks on a real wall will all be relatively the same distance out.

What we have to do here is adjust the Contrast and Brightness so that enough of the texture of the bricks is visible, while lessening the difference in colour of the individual bricks. Select *Layer>Colors>Brightness-Contrast*. I found a Brightness of 50 and a Contrast of 50 to work well, but you may try different settings if you wish.

8) Now select *Layers>Colors>Auto>Stretch Contrast*. What this does is ensure that the darkest part of the image is as dark as it can be and the lightest part as light as it can be. This is a very useful tool to stabilize a bump map when you've had to raise the brightness very high in order to get a good result.

9) We need to blur this layer very slightly so that the bump mapping isn't horribly sharp, so select *Filters>Blur>Blur*. Hit OK in the window that appears; the default settings will be fine.

10) Go to the Layers window and activate the top eye again, then click on the top layer. Now move the Opacity slider until it reads roughly 85 (it's hard to get it exact). Don't worry that you can't see the layer underneath very clearly, the slight shades of gray will be picked up in the bump map.

11) Merge the two layers into one whole image using *Image>Merge Visible Layers*.

The reason we used two layers for this example is because overuse of the Brightness-Contrast tool

results in a lack of definition. We had to get the deep parts around the brick to be clear, yet also have the subtleties of the brick texture. This is not something that's possible with the use of just one layer.

11) Changing the opacity like we did in step 10 will have reduced the overall lightness of the image, so once again run *Layers>Colors>Auto>Stretch Contrast*, so we get as much depth as possible.

12) Offset the image by selecting *Layer>Transform>Offset* from the menu. Press the button marked 'Offset by (x/2),(y/2)', then hit OK. Check to see if there are any harsh edges and then fix them using the techniques you learned from the previous two parts of this tutorial. When you're finished, offset the image again to return it to normal.

13) Save the bump map as a .tif file in the same directory that you loaded it from and don't forget to give it a different name from the original texture, or you'll overwrite it.

14) Repeat stages 9-21 from [Easy Bump Mapping](#) for the new texture and bump map.



Finished Texture And Bump Map In-Game
(click to enlarge)



[Introduction](#) - [Tileable Textures](#) - [Easy Bump Mapping](#) - [Intermediate Bump Mapping](#)

Advanced Bump Mapping

In this section I won't be providing a tutorial for you to follow, but instead I hope to encourage you to find your own ways of creating bump maps. The tutorials in the previous sections were merely a guide for you to become comfortable with making bump maps. They were the most appropriate way to bump map those textures, but you may have to use wildly different techniques when trying to achieve the best result for other textures.

The point is that there is no one correct way of doing things. I hope to show you through various examples how it is important to be aware of the capabilities of the The GIMP and to be able to use them in the manner *most appropriate* for the texture in hand.

Please Note:

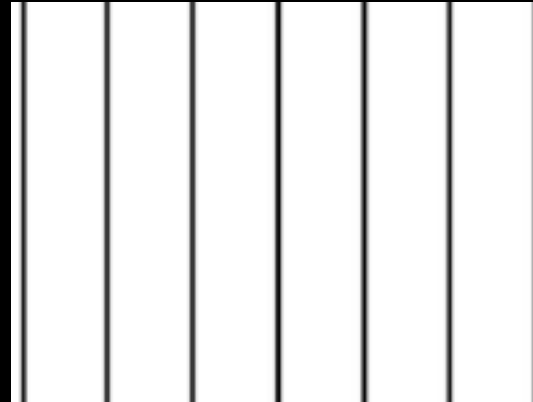
The textures shown in the examples below are not the originals; they were rather large and had to be resized in order to preserve bandwidth. The in-game shots are taken using textures and bump maps that I created from the originals.

Simplicity

Sometimes all that is needed for an effective bump map is something incredibly basic. Look at the wood texture below and you can see that a flat shiny surface with a dip at the side of each plank is all that is necessary. The bump map was incredibly easy to create; a straight black line was drawn down the gaps between the planks and slightly blurred.



Wooden Planks Texture
(click to enlarge)



Bump Map
(click to enlarge)



Bump Mapped Texture In-Game
(click to enlarge)

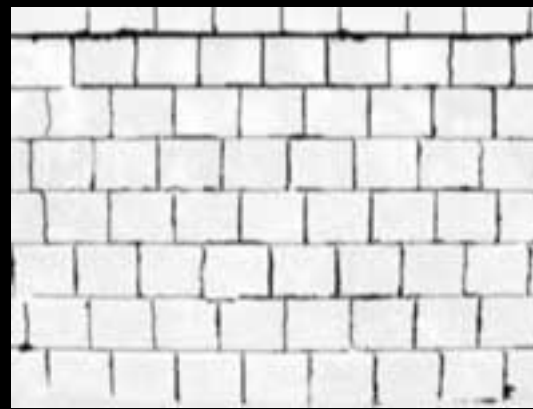
Attention To Detail

There are times when no amount of Brightness-Contrast adjustment will give you a worthwhile result and you simply have to go in and do it all manually. Check the wall texture below and see how some of the brick edges are not very distinct (especially the horizontal ones). This, coupled with the fact that some of the bricks had colouring *that was darker than the gaps*, meant that the only way to get satisfactory looking edges was to go in and paint around every brick as accurately as possible.

This took about an hour and a half to achieve, but sometimes it's necessary to go to these lengths in order get the best result.



Wall Texture
(click to enlarge)



Bump Map
(click to enlarge)



Bump Mapped Texture In-Game
(click to enlarge)

Ignore The Texture

All of the other techniques involve making use of the original texture to create the bump map, but you don't *have* to do this. You could use another texture as the base for a bump map. You could simply draw your own pattern. You could also use multiple techniques across many layers. It's really only limited by your imagination and creativity.

Note, however, that this won't really work on things like brick textures, where the eye *expects* to see light and shadow around the obvious areas. Therefore, it's best to use this technique on textures not containing specific objects, like the one below. Notice how the bump map has no relevance to the texture at all, yet still provides a nice, unique depth that doesn't look out of place.



Texture
(click to enlarge)



Bump Map
(click to enlarge)



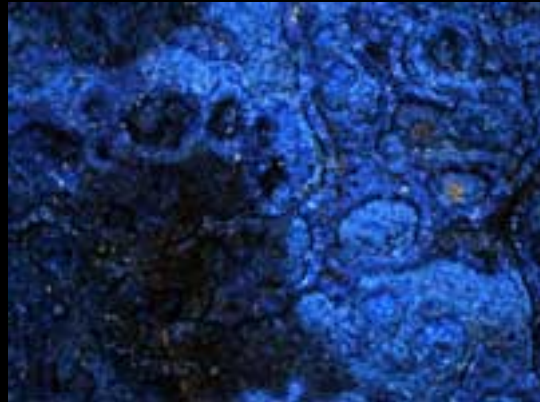
Bump Mapped Texture In-Game
(click to enlarge)

Respect The Edges

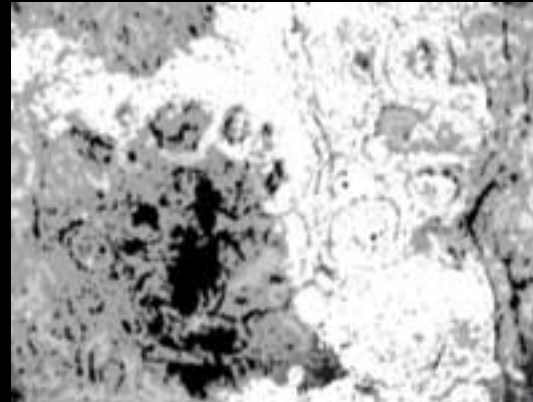
In the long pursuit of the the most amazingly detailed bump mapping possible, it can be easy to

forget what really matters. Solid black has exactly the same effect as solid white. When the difference between them occurs, that is what matters. Sometimes the finer details are unimportant and focusing on purely on the gray edges can yield better results more efficiently.

The texture below is incredibly detailed, yet only the stark edges were grabbed for the bump map. This has the effect of making the light and shadows that *do* occur, appear more noticeable. It also allows the flat areas to have more apparent specular effects, which can be a very good thing in some cases.



Texture
(click to enlarge)



Bump Map
(click to enlarge)



Bump Mapped Texture In-Game
(click to enlarge)

Specular

When creating bump maps for use in Halo it can be useful to know how they also affect specular lighting. Specular lighting in Halo comes in two forms, the first being a simple reflection of light that is dictated in colour and intensity by the surface being lit (as opposed to bump mapping, which uses colour and intensity settings from the light source).

The second form is more relevant to us however, because bump mapping can affect a specular cube map in a three dimensional manner, just as it does to dynamic lighting (a cube map is a way of simulating a reflection of your surroundings by arranging six separate bitmaps into a cube; this sounds complicated but it isn't, think of it like a fake mirror).

A normal specular cube map will be just a straight, flat reflection of the bitmap, but when combined with bump mapping, the results look much more realistic. This small video shows a bumped specular cube map in action: [Specular Movie \(837 Kb\)](#). Notice how the light flows around the contours of the glass as opposed to just being flat.

You must be careful with how you apply bump mapping to *any* specular effect. The results can look awful for two reasons, the first being the dreaded 'plastic' effect. Because specular effects are relatively new, they don't yet have the level of customization to be able to look good when applied

to any surface you want. As you saw from the video above, the glass was an appropriate surface for this effect. But if you were to apply it to, say, the grass in *de-aztec* (*Counter Strike: Source*), the result can look horribly plastic and fake.



Ick
(click to enlarge)

The second problem to watch out for is that if you have a deep, complex bump map, the reflections can be barely noticeable because of all the sharp angles. The window below uses a bump height of 0.01, which is tiny compared to the wall in the Attention To Detail section above, which uses 0.25.



Window Texture
(click to enlarge)



Bump Map
(click to enlarge)



Bumped Specular Cube Map In-Game
(click to enlarge)

The bump map for this example was easy to create, yet required some forethought into what the end result was trying to achieve. The grayscale of the texture was inverted so as to allow better distinction around the edges (they would otherwise just follow the shading of the texture and it wouldn't stand out at such a low bump height). This also made the black lead around the stained glass the most reflective part of the texture, just as it should be.

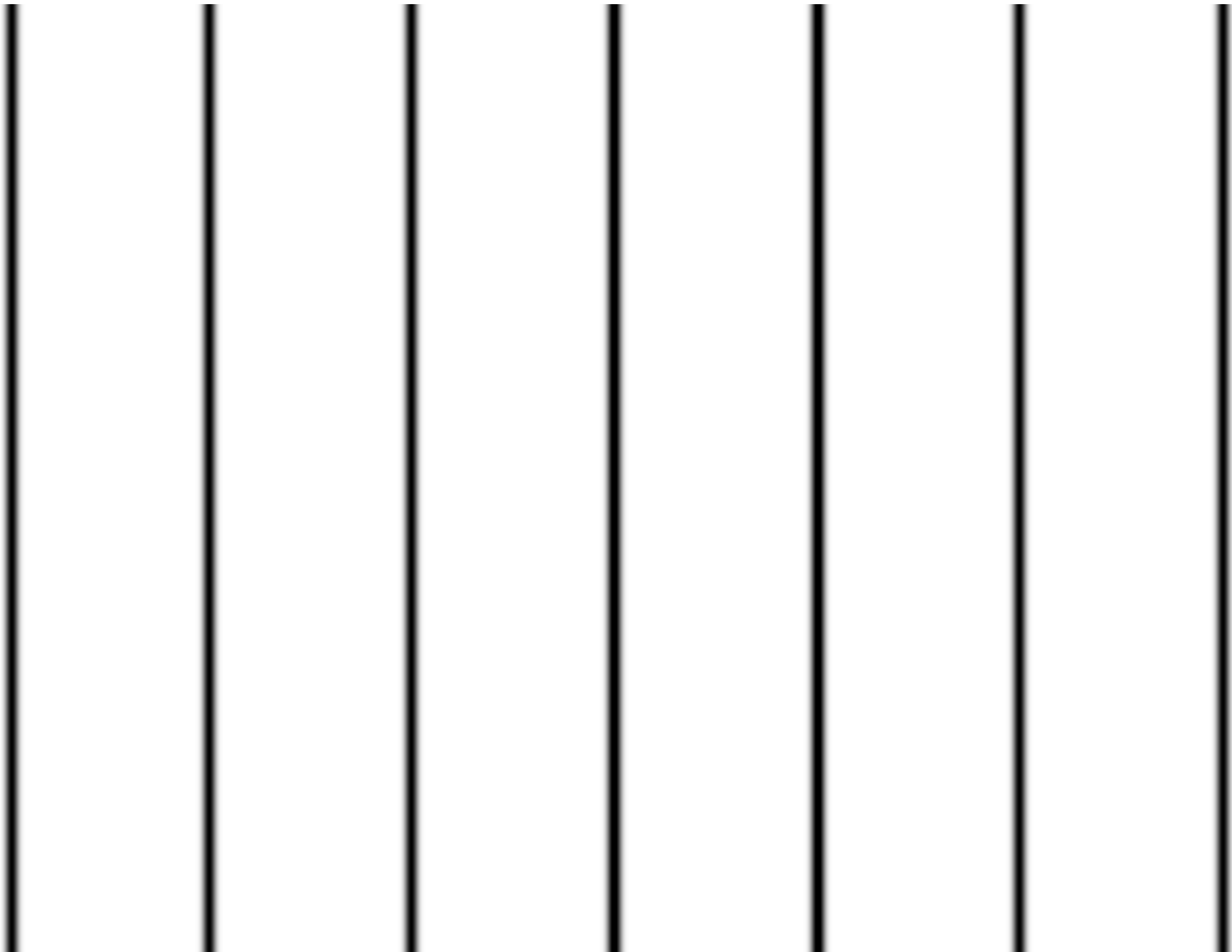
The bump map was then subjected to *Layer>Colors>Auto>Equalize*, which is very useful for bringing out the differences in the shades of gray and thereby increasing the detail level. It was

then blurred slightly and *Stretch Contrast*-ed.

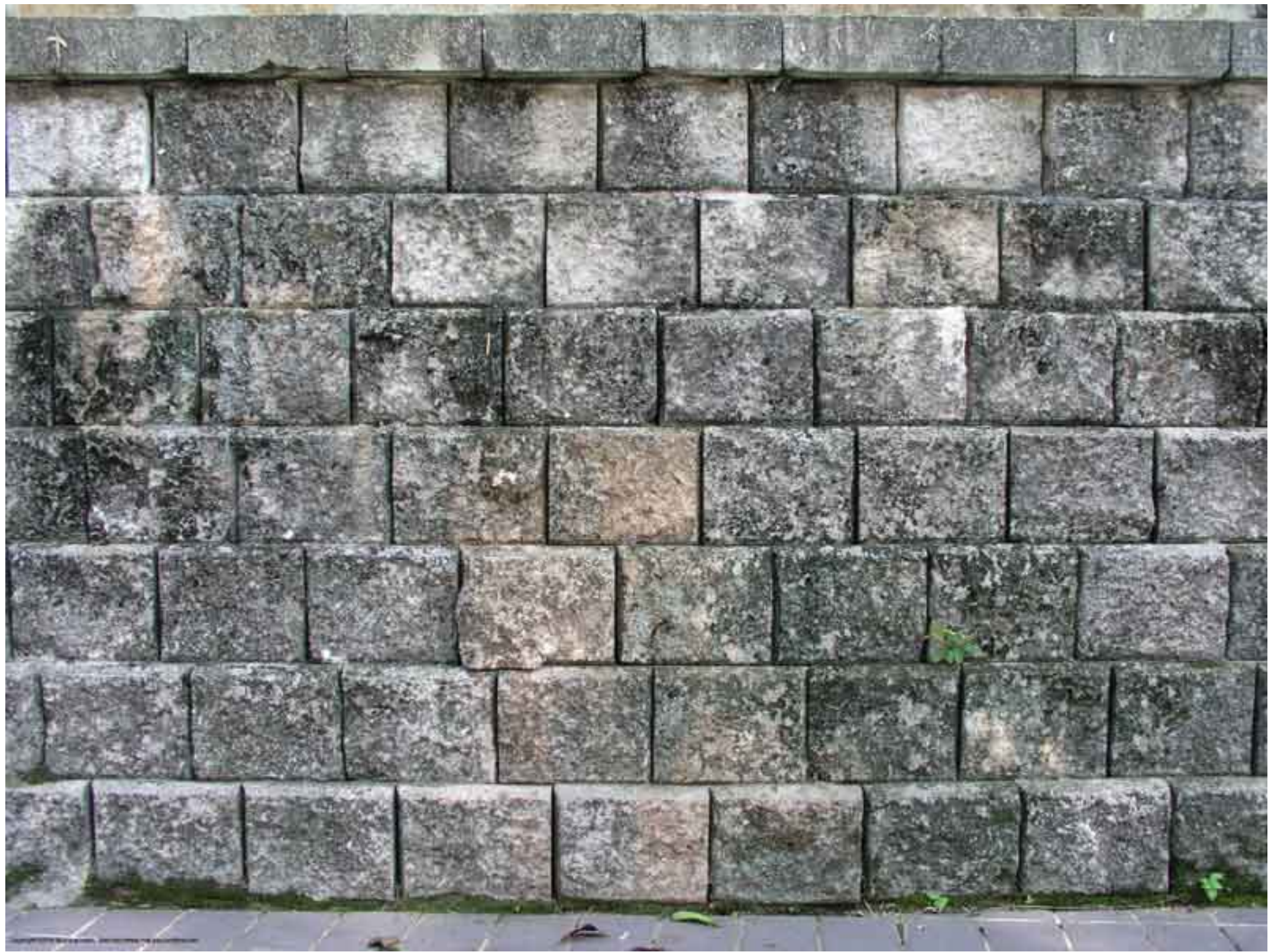
Conclusion

Hopefully I've shown you some different approaches to creating bump maps and inspired you to use whatever means are necessary in order to get the best result. Take a look at the filters and other tools in The GIMP and learn how they work. You'll then be to apply them effectively, when appropriate, to create the best possible bump map for your texture.



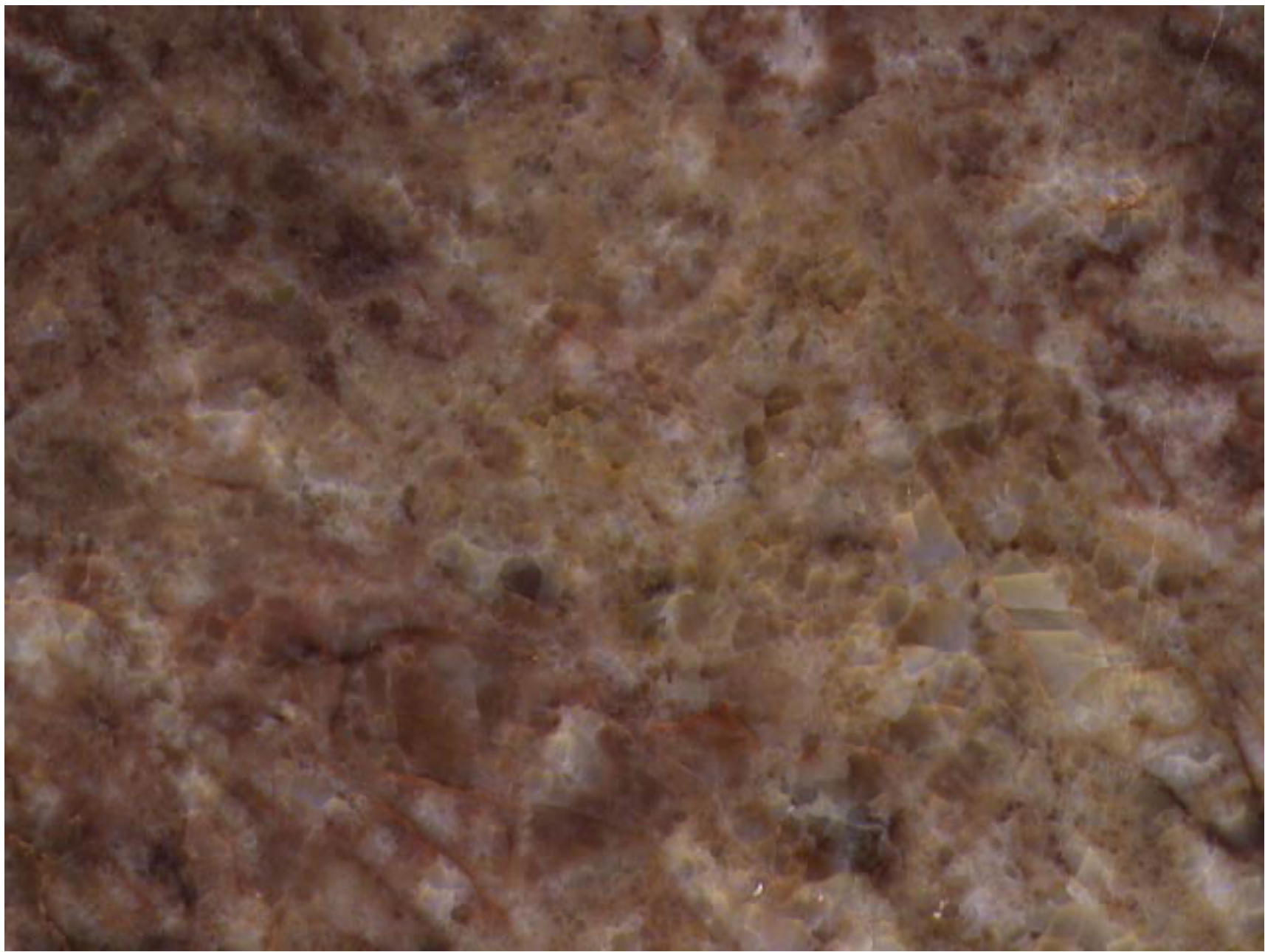


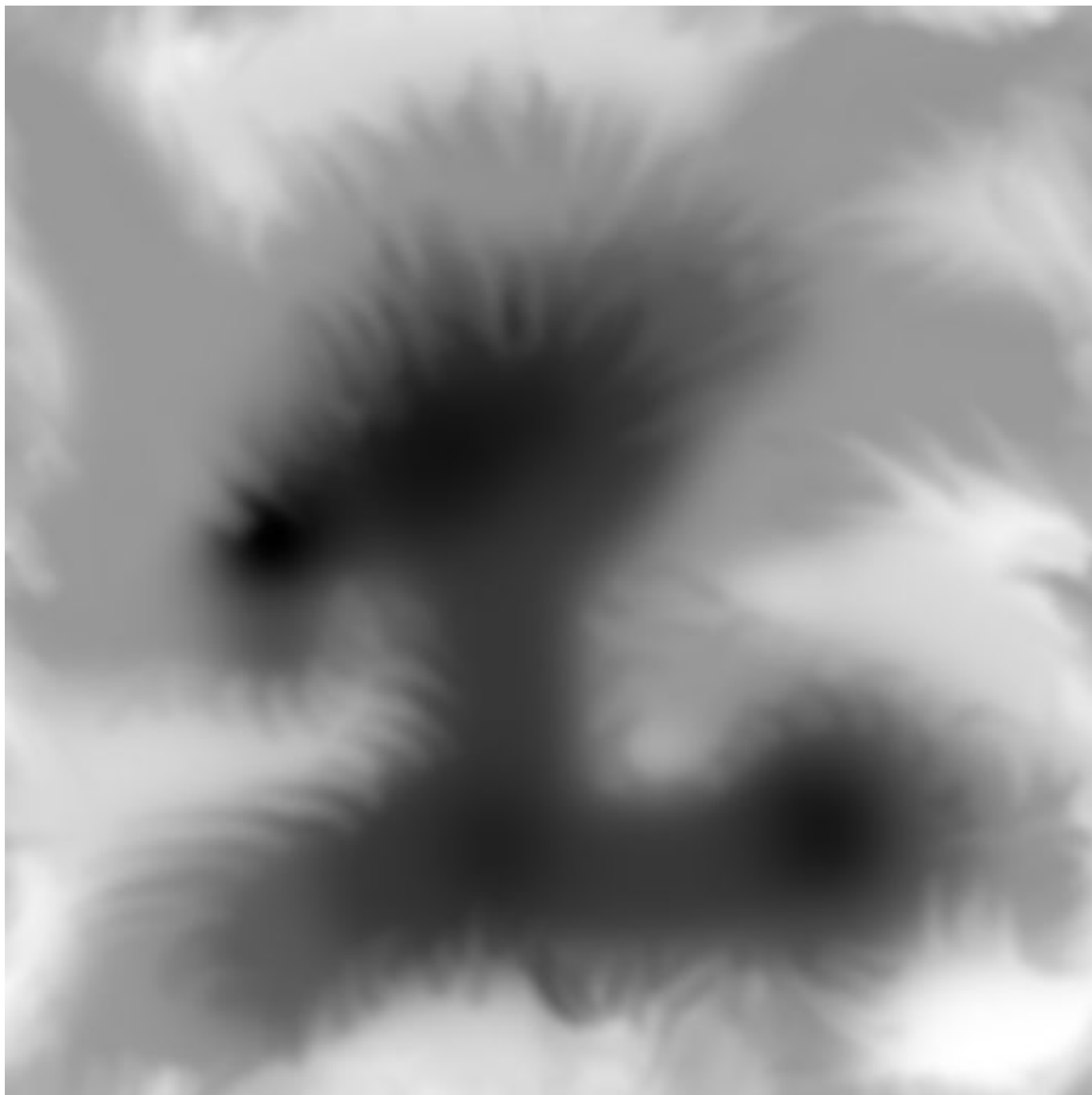


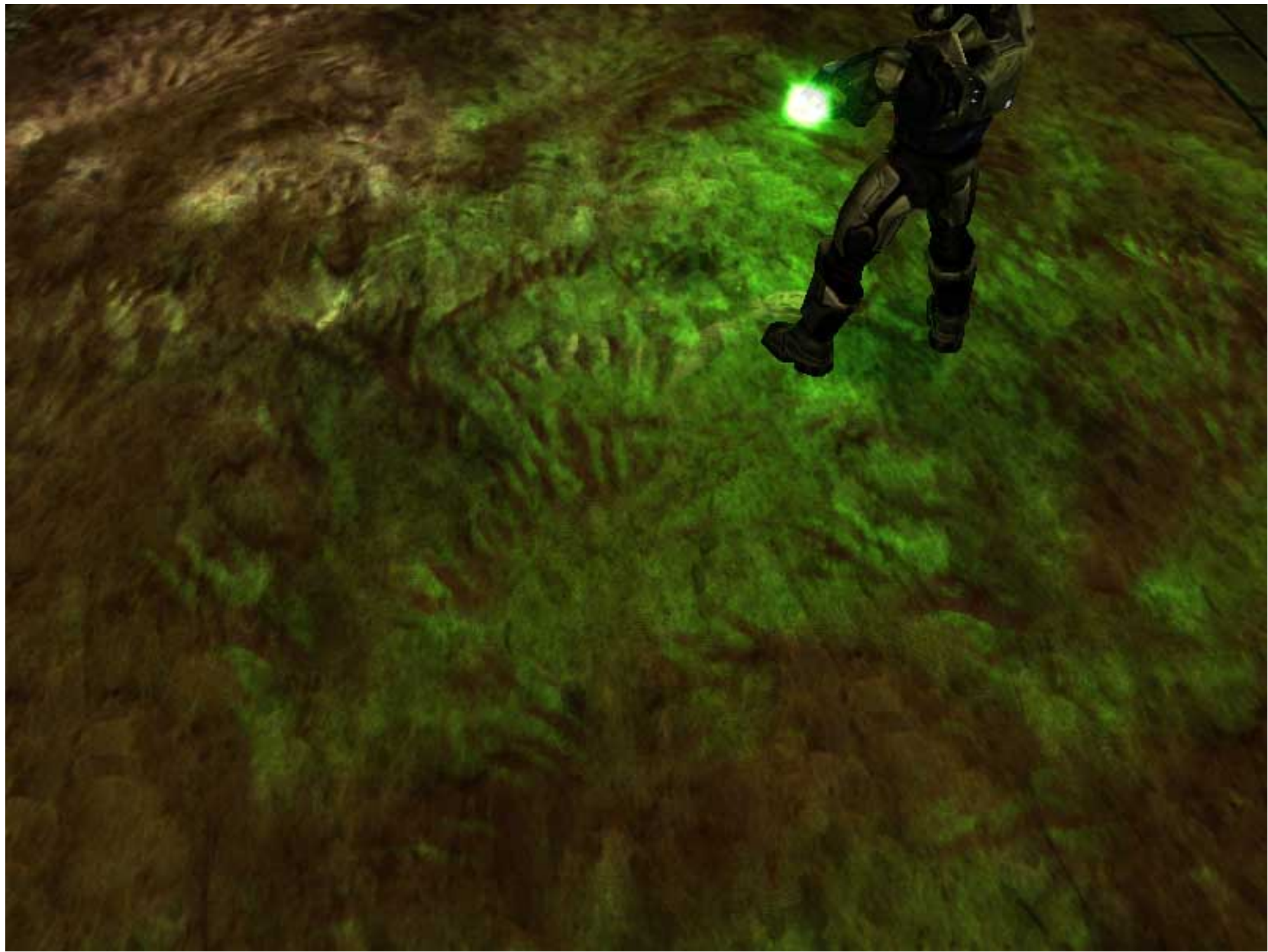


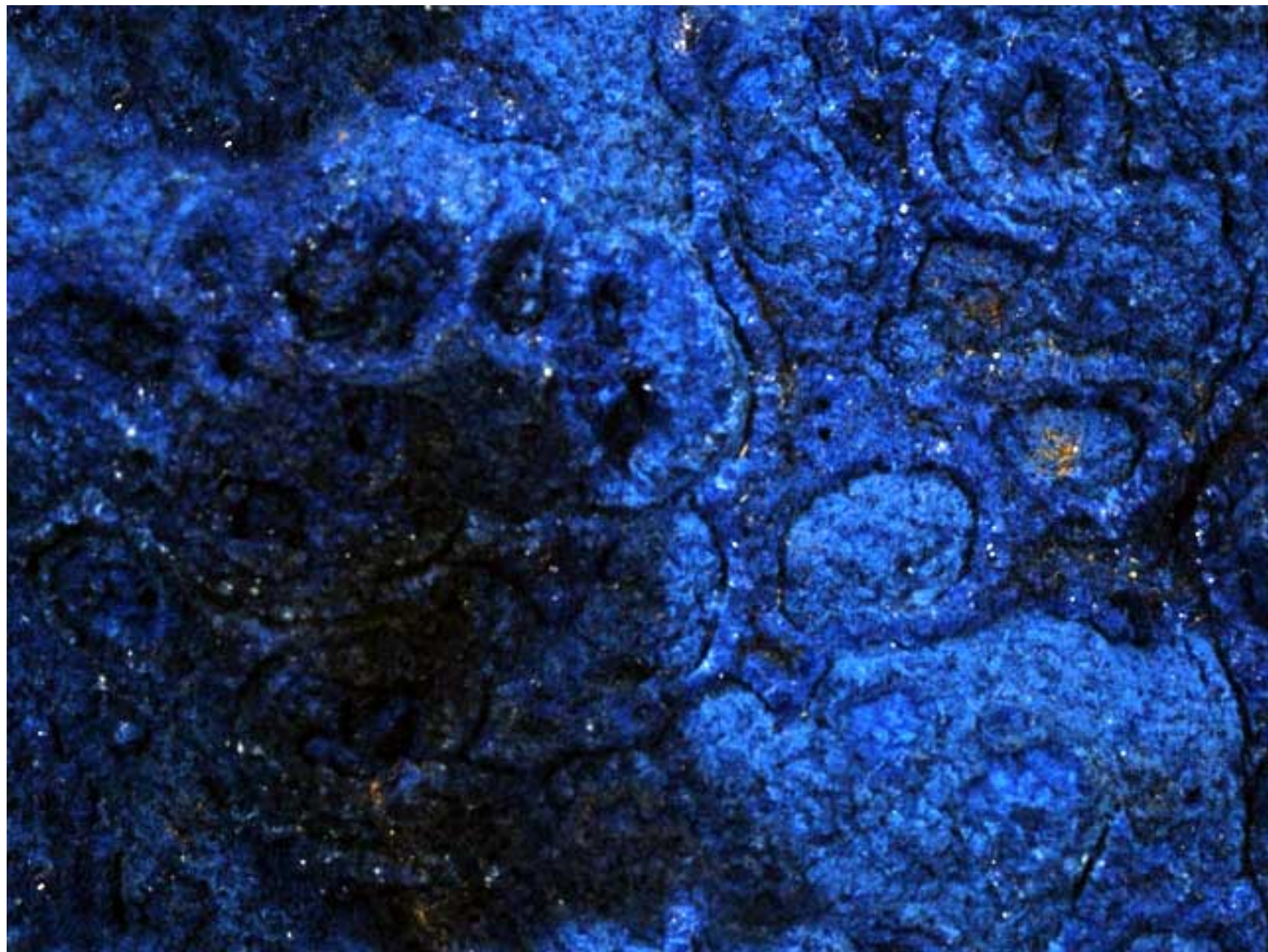


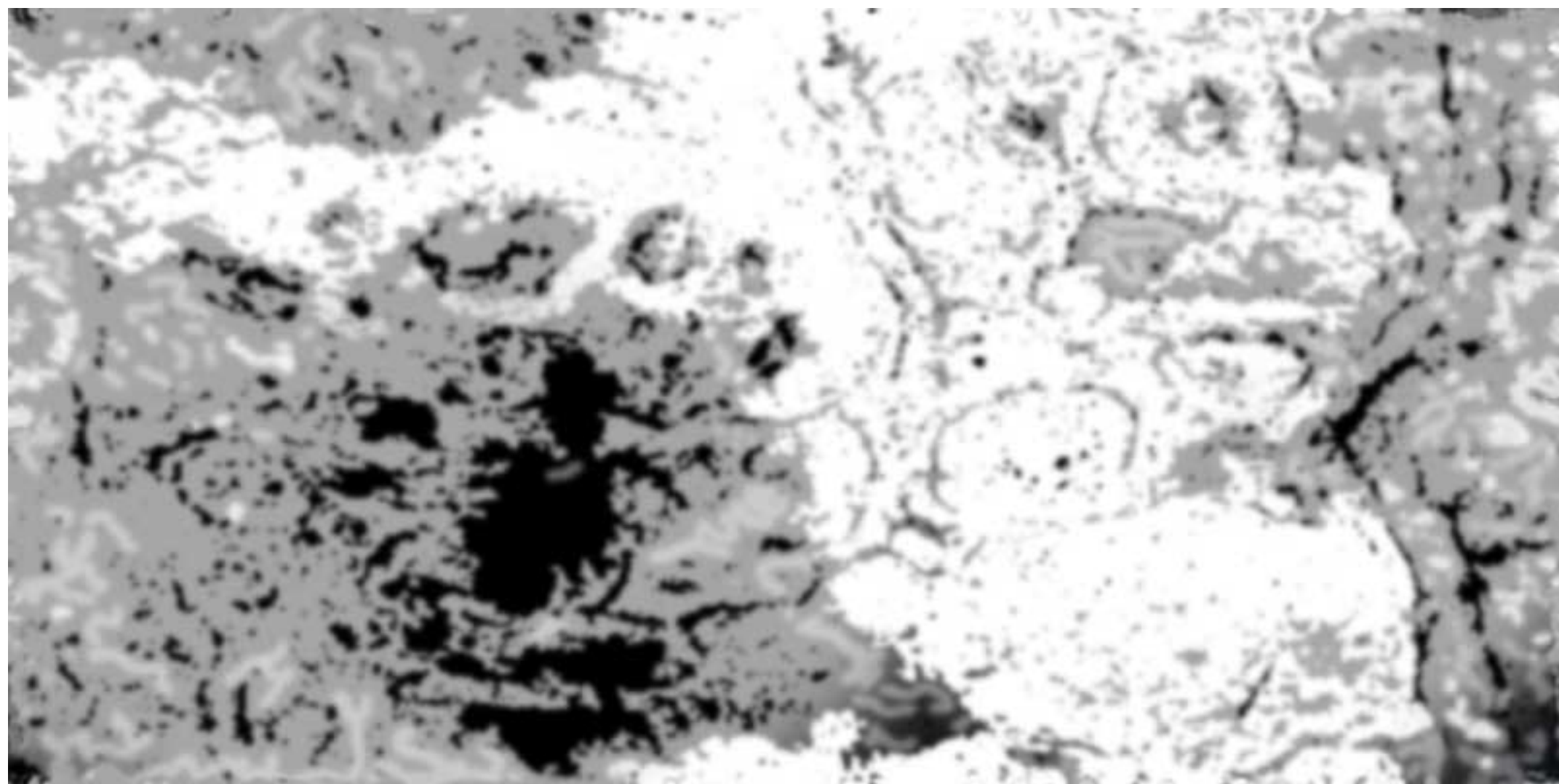


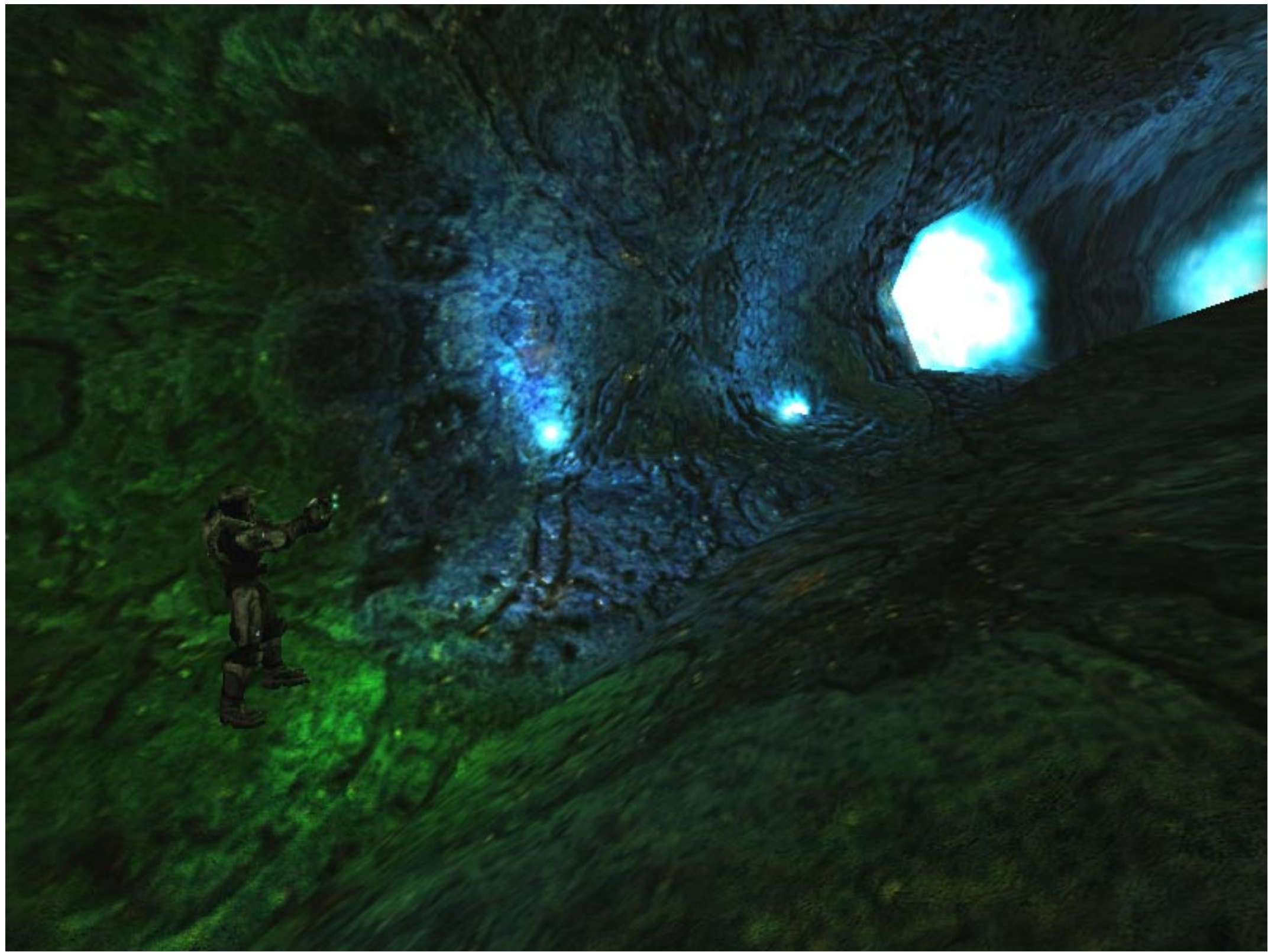














Courtyard



100



0:00

\$ 800
9 | 0

45

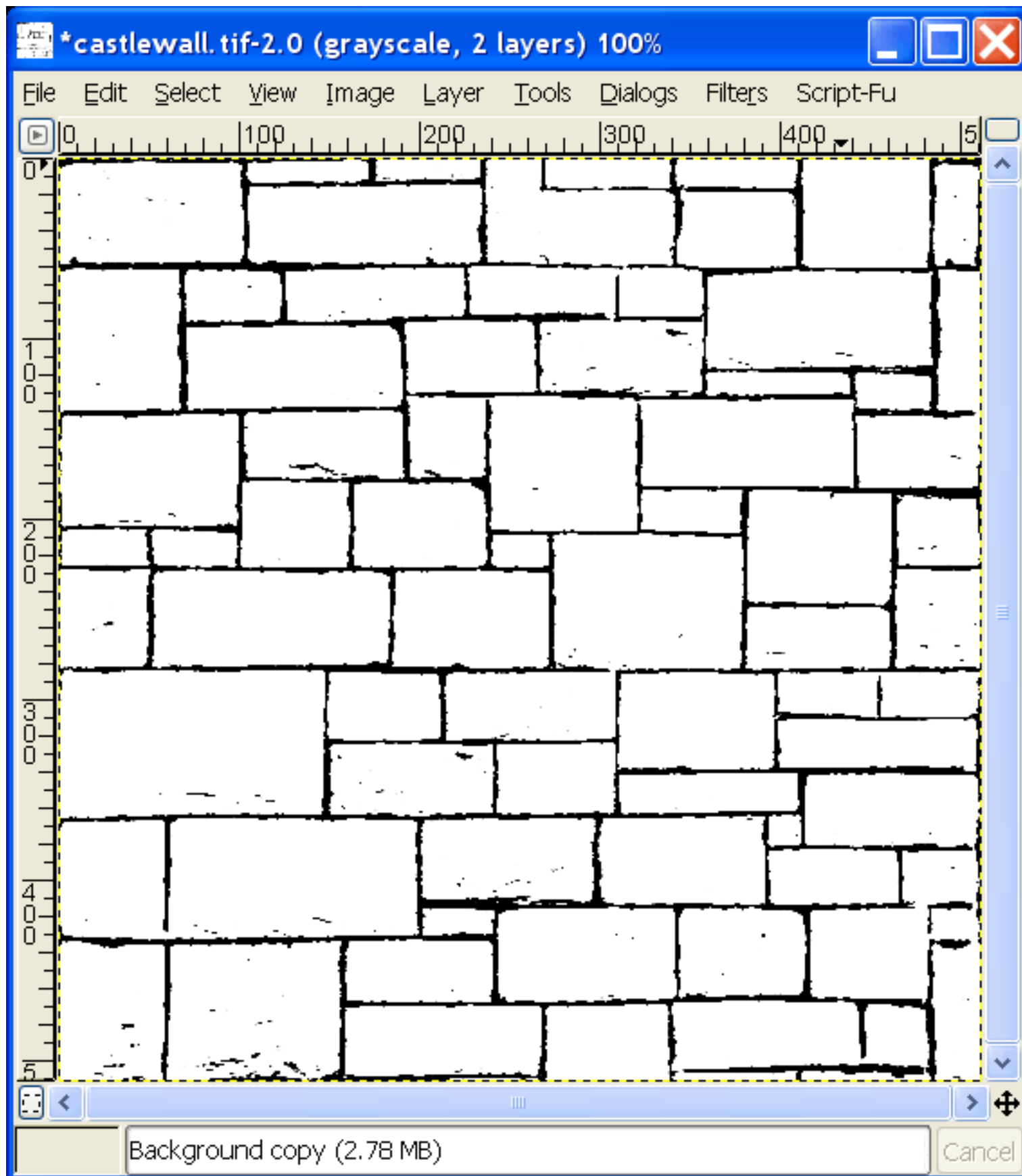


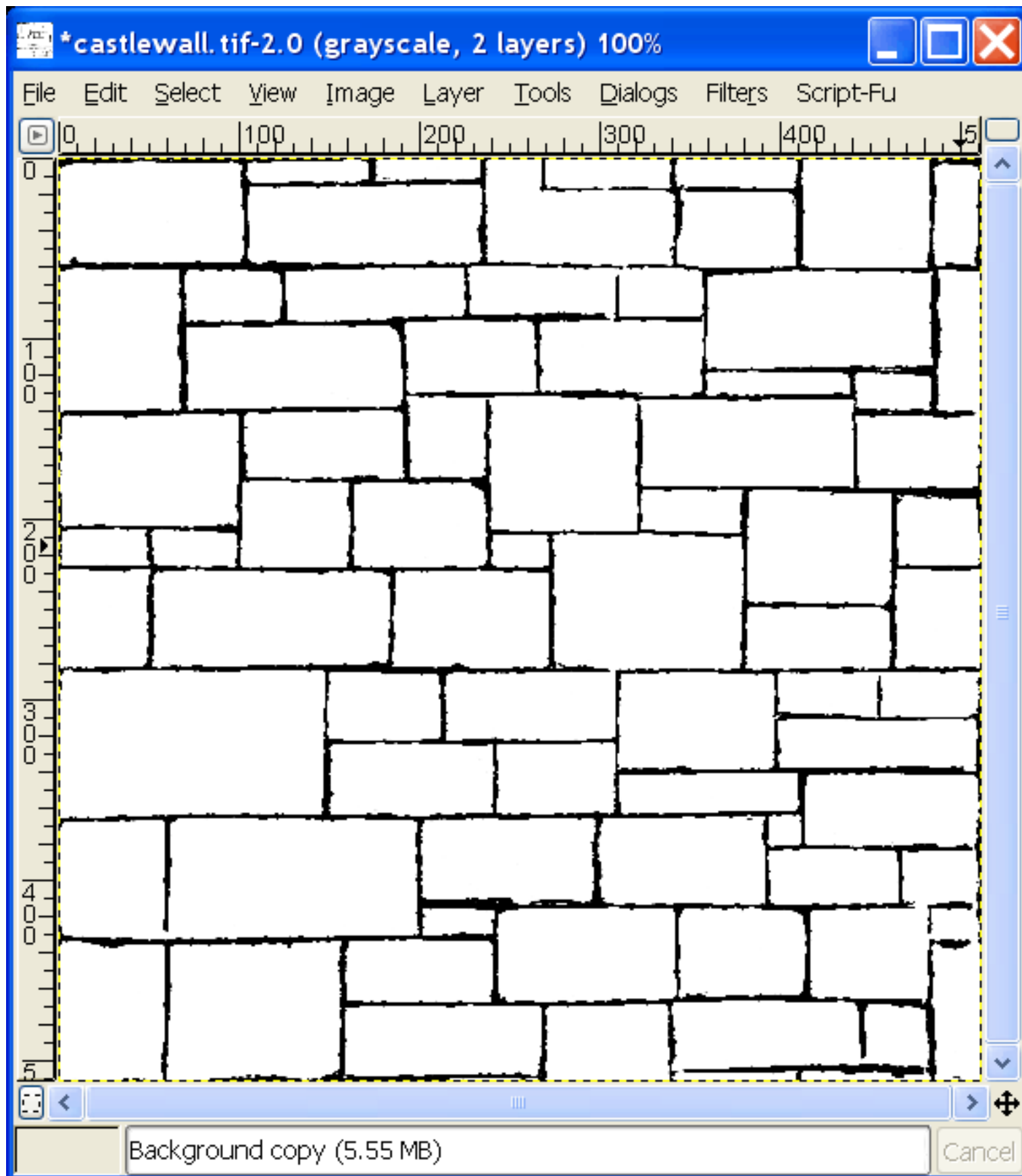
The St Margaret Altar Cloth

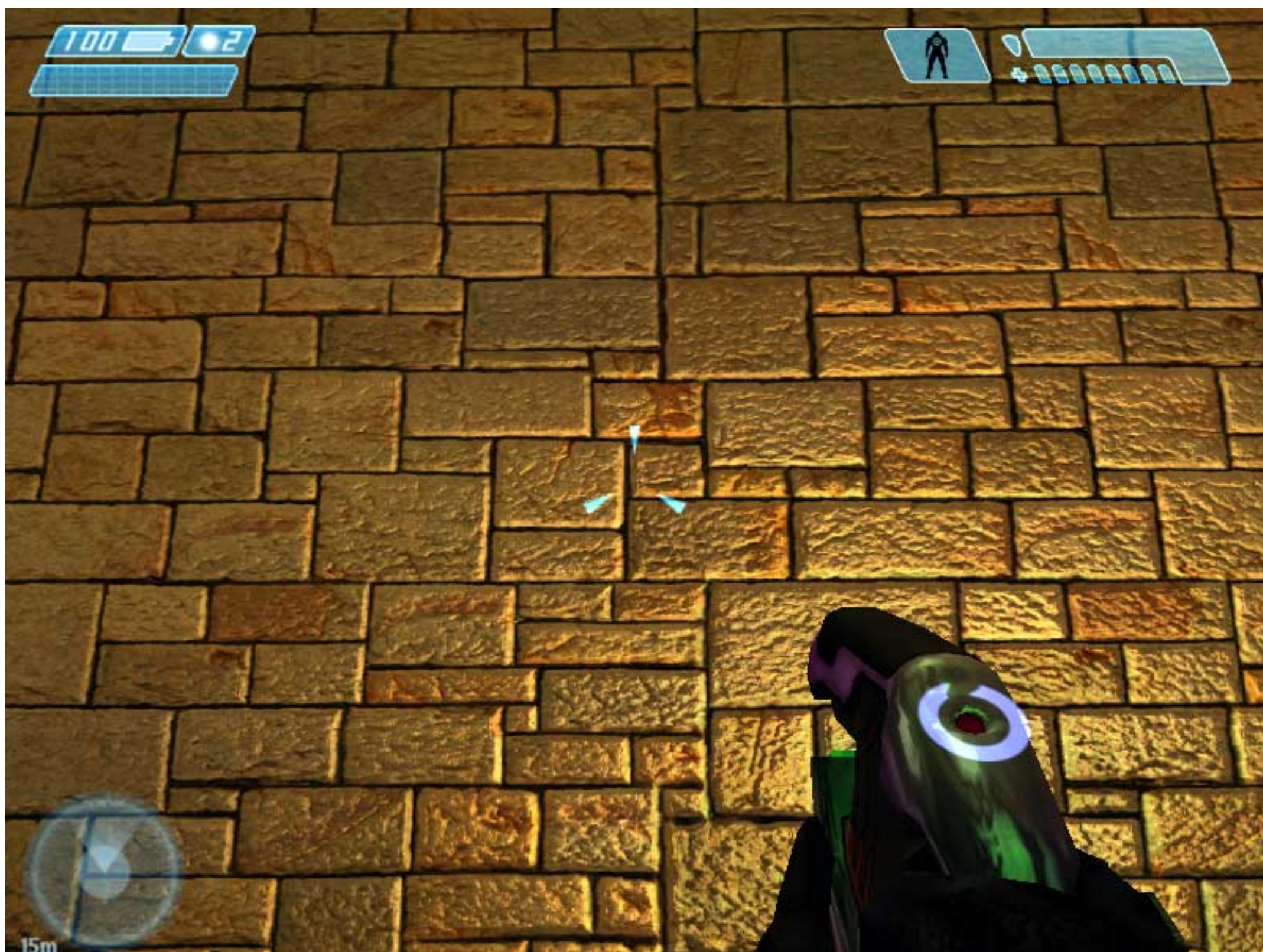


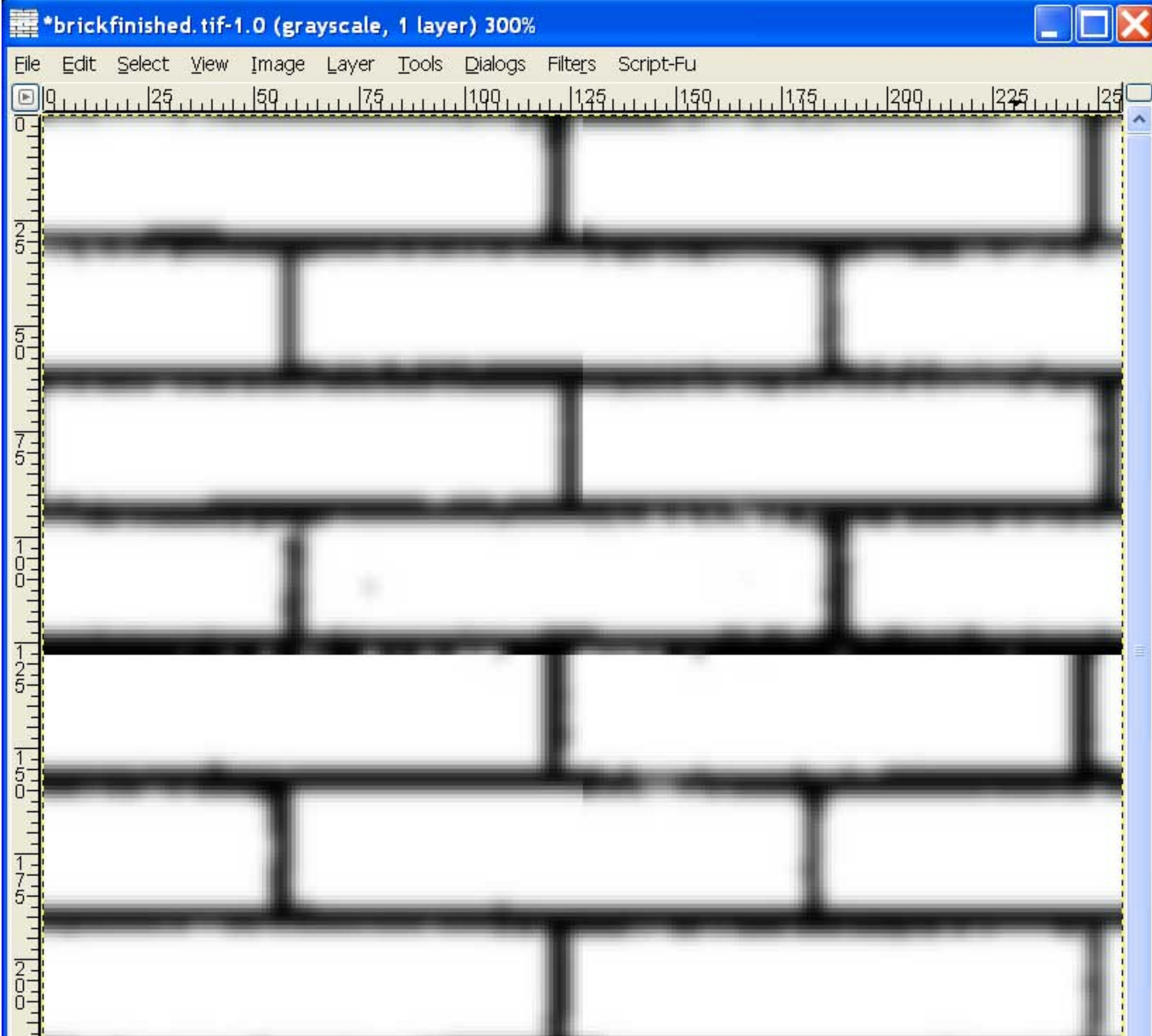




















radiosity properties

flags

- ☐ simple parameterization
- ☐ ignore normals
- ☐ transparent lit

detail level

high ▼

power

0

color of emitted light

r 0 g 0 b 0



tint color

r 0 g 0 b 0



physics properties

material type

stone ▼

environment shader

Setting <true atmospheric fog> enables per-pixel atmospheric fog (for models) but disables point/spot lights, planar fog, and the ability to control the atmospheric fog density for this shader.

flags

- ☐ alpha-tested
- ☐ bump map is specular mask
- ☐ true atmospheric fog



diffuse properties

flags	☐ rescale detail maps ☐ rescale bump map		
base map	levels\test\church\bitmaps\		Open
detail map function	double/biased multiply ▾		
primary detail map scale	0		
primary detail map		...	Open
secondary detail map scale	0		
secondary detail map		...	Open
micro detail map function	double/biased multiply ▾		
micro detail map scale	0		
micro detail map		...	Open
material color	r	g	b
	0	0	0

bump properties

Perforated (alpha-tested) shaders use alpha in bump map.

bump map scale	1		
bump map	levels\test\church\bitmaps\		Open





