IC

tions
ags
sers
ompan…
nanswe…
IS

low for
s – Start
orating
naring
izational
edge.

eate a
Team

Why
ams?

Creating Normal maps from a Texture?

Ask Question

Asked 6 years ago
Modified 3 years, 1 month ago
Viewed 59k times

13

15

Is it possible to create a Normal Map from a Texture, not a mesh? If so, how?

As I'm working on an Eye, and have already made the textures, I was wondering if it was possible to create Normal Maps out of those Textures.

texturing | normals | texture-baking

Share
Improve this question
Follow

asked May 18, 2016 at 14:20

user15147

Normal maps are pretty hard to make,But bump maps are not and it would achieve a similar result,Will that work for you?
– Omar Emara
May 18, 2016 at 14:40

2

Related - blender.stackexchange.com/questions/390/…
– Mr Zak
May 18, 2016 at 14:41

Add a comment

3 Answers
Sorted by:
Highest score (default)

7

While normal maps from textures can also be created within blender, there is another FLOSS software called AwesomeBump. It's not quite as user friendly as CrazyBump but it gets the job done.

If you are on an AMD GPU system you will have to use the stable version 3.14 for now.
Share
Improve this answer
Follow

edited Oct 23, 2018 at 16:00

answered May 19, 2016 at 0:00

metaphor_set
6,495 2 20 34

Add a comment

36

Theory

First thing normal map generators do is to converts the input image into a grayscale image, even if it is

The Overflow Blog

- The Great Decentralization? Geographic shifts and where tech talent is moving...
- Want to be great at UX research? Take a cue from cultural anthropology (Ep. 451)

Featured on Meta

- Announcing the arrival of Valued Associate #1214: Dalmarus
- Improvements to site status and incident communication

Linked

Trace visualisation in 3D - how to wrap curves onto a 3D mesh

How do I make a normal/displacement map when all I have is a photo?

Related

Creating a texture and normal map from a 3D model / scene

3
Importing normal maps from obj/mtl

1
How do I bake multiple texture maps into 1 UV, In Blender Cycles?

1
High poly from low poly with normal maps

Hot Network Questions

- lifetime and validiy of exported variables in bash script
- One of my first Python projects - an adventure game
- Are there any bacteria that could be used as food?
- How to evaluate when recommender systems are influencing behavior?
- Solutions for transporting a unicycle by bike

more hot questions

Question feed

colored, that is because color information is practically useless in the generation process, the algorithm used to convert the RGB colors into single values varies from a generator to another, so it is recommended that you convert your input image into grayscale before using the normal map generator, the reason will become clear in the next section.

Imagine the image as a grid of vertices, each vertex represents a pixel, the z location (height) of each vertex is equal to its value (Which is just a scalar not an RGB vector; because it is a grayscale image, remember). So we realize that the image is just a 2D function that associates a value to each point in space, except it is not a continuous function, since it is only defined on some discrete points in space. This surface we just described has **normals** at each vertex (just like your mesh has normals), and those normals is what a normal map stores, so the generator's job is to simply compute the normals of the surface and encode it in an image. It can be proven that the normals' components relates to *partial derivatives* of the surface using the equation:

$$\overrightarrow{N_x} = \frac{-\frac{\partial F}{\partial x}}{\sqrt{\frac{\partial F}{\partial x}^2 + \frac{\partial F}{\partial y}^2 + 1}}$$
$$\overrightarrow{N_y} = \frac{-\frac{\partial F}{\partial y}}{\sqrt{\frac{\partial F}{\partial x}^2 + \frac{\partial F}{\partial y}^2 + 1}}$$
$$\overrightarrow{N_z} = \frac{1}{\sqrt{\frac{\partial F}{\partial x}^2 + \frac{\partial F}{\partial y}^2 + 1}}$$

Where F is the function that represents the surface and N is the unit normal vector. Partial derivatives describes the steepness and direction of the surface and they can be computed using what is know as **Symmetric Derivatives** which I explained in more details in my answer here.

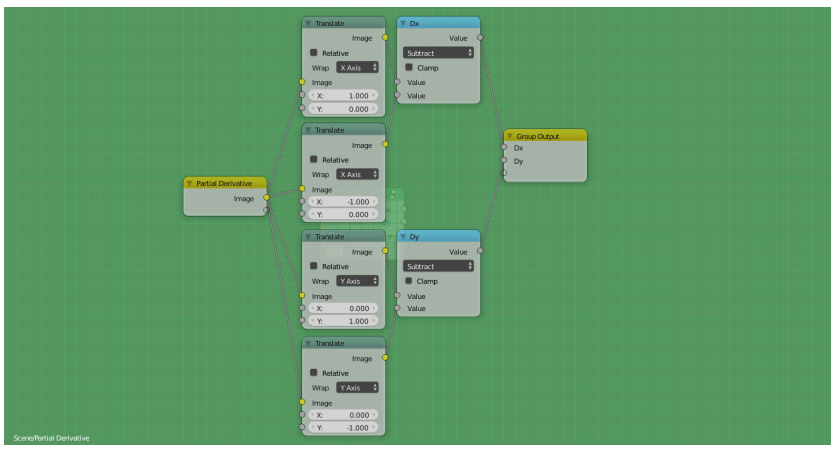
Assuming we have the partial derivatives, we can compute the normal vector components using the previous equation. A normal map stores the x, y and z components of the normal vector into the R, G and B channels of the image respectively. If we attempted to do that, we will encounter a problem, the image can only store positive values that are less than one (Assuming a standard image format is used), the normal vector components will always be smaller than one because it is a unit vector, however the vector components can be negative denoting directions opposite to the fundamental vectors of the space, so we have to find a way to encode these data which belongs to the interval $[-1, 1]$ to the interval $[0, 1]$ which is the possible values for a pixel. This is a simple remapping task that can be performed using the equation:

$$R = \frac{(\vec{N_x} + 1)}{2}$$
$$G = \frac{(\vec{N_y} + 1)}{2}$$

Where R and G are the red and green channels respectively, notice that we didn't perform the same remapping to the z component because it is always positive, think about the surface that represents the image again, its normals will always be facing the positive direction of the z axis. By using the previous equation, we get the RGB values for the normal map, so we are ready to do the implementation.

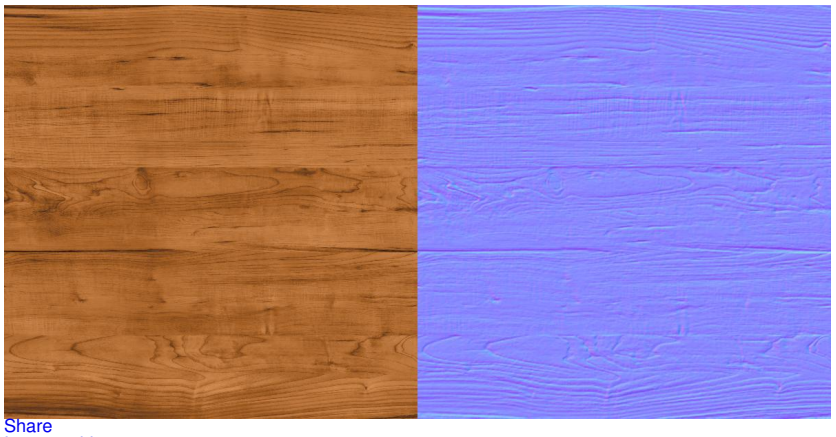
Implementation

We can make a node group that computes both the $\frac{\partial F}{\partial x}$ and $\frac{\partial F}{\partial y}$ partial derivatives given an input image. The implementation is as simple as a translation and a subtraction:



Refer to the article for more information, the rest of the implementation is as easy as applying the two equations above and creating an image out of the individual channels.

An example result



Share
Improve this answer
Follow

edited Apr 16, 2019 at 19:19

answered Sep 16, 2016 at 6:38

Omar Emara
22.1k 5 47 98

Could you add a bit to your answer explaining how to use it. As it is now all you really have here is an ad.
– David
Sep 18, 2017 at 12:01

@David Making the answer an ad. was truly not my intention. I wrote a detailed article on how to make the generator from scratch, will an answer containing the article link and a small summary be better? I know it will still be technically an ad. for my blog but the article is too long to post in an answer so that's my only option.
– Omar Emara
Sep 18, 2017 at 13:19

The link to the article doesn't work any more.
– Ray Mairlot
Apr 16, 2019 at 19:12

2

@RayMairlot Yes, I have since taken the site down. Thanks for notifying. I will try to add all the missing information to this answer for completeness.
– Omar Emara
Apr 16, 2019 at 19:18

Add a comment

2

Especially if you are already using Photoshop in your workflow, the Nvidia Texture Tools are simple to use and pack some good features.
Share
Improve this answer
Follow

answered Jun 18, 2016 at 1:54

justinbot
111 4

Add a comment

Your Answer

Post Your Answer

By clicking “Post Your Answer”, you agree to our terms of service, privacy policy and cookie policy

BLENDER
Tour
Help
Chat
Contact
Feedback
COMPANY
Stack Overflow
Teams
Advertising
Collectives
Talent
About
Press
Legal
Privacy Policy
Terms of Service
Cookie Settings
Cookie Policy
STACK EXCHANGE NETWORK
Technology
Culture & recreation
Life & arts
Science
Professional
Business
API
Data
Blog
Facebook
Twitter
LinkedIn
Instagram