Phased Hyper Texture

A Hyper Texture is a texture that can drive colours and output beyond the normal limits. Adjusting the output of conventional Hyper Textures is a bit tricky. The output of the new Phased Hyper Texture can be adjusted over a wide range in a deterministic manner.

Overview

Good practise is adhering to a linear workflow for the renders. Bryce lends itself to the linear workflow because lights and colours exhibit a linear behaviour. The advantages of a linear workflow are discussed in the document *Linear Workflow*. That the material colours, lights and IBL follow a linear behaviour is belaboured in the document *Brightness and Falloff*.

Conventional Hyper Texture (HT)

How to set-up this type of Hyper Texture is covered in the one-page Memo *Hyper Texture* and is also shown in several videos. The conventional type starts with a negative value and the polarity may have to be swapped if a positive value is needed. Since the output value is exceedingly high, it usually has to be divided down to get it to a manageable range.

The advantage of this Hyper Texture type is that it can output amazingly high values. The disadvantage is that it cannot be precisely levelled. Adjustments have to be done in the Materials Lab and sometimes controls have to be adjusted in 0.01 increments.

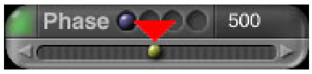
Phased Hyper Texture (PHT)

This type — which has been developed by *David Brinnen* like the conventional one — is yet simpler to set up in the DTE; polarity can be changed with one click. The output can be calculated and fine adjusted right in the DTE over a wide range.

The Phased Hyper Texture

The table shows all the settings in the Deep Texture Editor to create a Phased Hyper Texture. Only one component is shown; if two or three components are used to increase the output, all share the same settings and are combined using *Multiply* for the *Blend Mode*.

DTE Settings for Phased Hyper Texture – Component 1					
Colour	Swatches	All Black	**Filter**	Type	None
	Color Mode	RGB, None		a, b, c	n/a
	Output Type	Alpha Channel			
Noise	Type	Distance Squared	**Phase**	Type	Sine
	Mode	Standard		Mode	Standard
	Octaves	0, (attenuator)		Octaves	0
	Frequency	X = Y = Z = 0		Frequency	X = Y = Z = 0
	Dimensions	1D **+**, 2D & 3D **–**		Dimensions	3D
	Direction	XY = YZ = 0		Direction	XY = YZ = 0
	Freq. Adj.	Do Not Touch		Level	Output Control
Only Component 1 is used. **Combination**: Swatches and Color Mode are set as for Component 1.					



Setting Polarity

Positive: set Noise Dimension to 1D.

Negative: set Noise Dimension to 2D or 3D. The output level of 3D is approximately one third lower than the output level of 2D.

Setting Output Level

The output level is set with the Phase control value. The higher the Phase value, the higher the output.

Output level by Noise Octaves (Attenuator)

The Noise Octaves can also be used to control the output. Adding Octaves to the Noise works like an attenuator. Octave 0 gives the strongest output. A positive (1D) value is most positive and a negative (2D, 3D) value most negative. By increasing Octave, the output value gets weaker.

Adjusting the Output value

There are three means to adjust the output level of a Phased Hyper Texture: Phase, Octave and the number of components used.

Phase

The Phase value changes the output with a square function. This means doubling the Phase value will quadruple the Output value. Equally, halving the Phase value will result in a fourth of the Output value.

To double the current Output the current Phase value must be multiplied by the square root of two ($\sqrt{2} = 1.414$). Halving the Output can be done by dividing the Phase value by the square root of two or multiplying by 0.707.

This works fine and accurately for Phase values from 1000 (max) down to 250. Below the value 178 a small error is introduced that grows towards 64 and gets dramatic the more 0 is approached. To adjust the Phase value precisely, click with the right mouse button on the arrow at right.

Phase value x 2 = Output value x 4 **Phase value / 2 = Output value / 4**

Noise Octaves

Octaves can be set in the Noise dialogue. Octaves = 0 outputs the maximal value. Each additional octave halves the Output value. Therefore one octave results in half the Output value, two octaves in a fourth, three octaves on an eighth, and eight octaves in 1/256 of the level at octaves 0.

$$\textbf{Output value} = 1 / 2^{\text{Octaves}}$$

Using more than one component

The Phased Hyper Texture works with component 1. However, if a higher Output level is required than component 1 with Phase at 1000 and 0 Octaves can produce; a second or even a third component can be added.

To add a ***second component***, just copy component 1 into component 2 and use *Multiply* as blend mode. The outputs of the two components are multiplied and the final output is multiplied by ten.

Output value = Output component 1 x Output component 2 x 10

Assuming that the Phase value is set so that the output of each component is 4 (an arbitrary value for the sake of argument), the final Output value will be 4 x 4 x 10 = 160. If one component outputs 4 and the other 8, the result will be 4 x 8 x 10 = 320. It does not matter whether component 1 or component 2 outputs the higher value (has the higher Phase value) since it is a multiplication.

The Octaves settings add up. This means that the octaves settings of both components are added and each octave halves the output. If, for example, component 1 has Octaves at 1 and component 2 at 3, there are 4 octaves and the output will be 1/16. In the example above where we had an Output level of 320, it will now be 20.

$$\textbf{Output value} = 1 / 2^{(\text{Octaves component 1 + Octaves component 2})}$$

Obviously, the Output level can be reduced to 1/65,536 if the Octaves for both components are set to 8.

To add the ***third component***, just copy component 2 into component 3 and use again *Multiply* as blend mode. Essentially, everything works the same as for a single component.

Component 1 and 2 are combined and form a single component — let us call it component A. Component A is now multiplied with component 3 and the final output is again multiplied by 10.

Output value = Output component A x Output component 3 x 10 or

Output value = (Output component 1 x Output component 2 x 10) x Output component 3 x 10

Assuming all three components output 4 each, the resulting Output value will be 4 x 4 x 10 x 4 x 10 = 160 x 4 x 10 = 6400.

Also when using all three components, the Octaves settings add up and it is not important how it is set in the individual components.

$$\textbf{Output value} = 1 / 2^{(\text{Octaves component 1 + Octaves component 2 + Octaves component 3})}$$

In this way, the Output level can be reduced to 1/16,777,216 if the Octaves for all three components are set to 8.

Negative values

Any Phased Hyper Texture can also output negative values if the Noise is set to either 2D or 3D. This is straightforward enough if only component 1 is used. If more than one component is used, it can get a bit more complicated.

Considering only one component, using 2D gives a value 1.5 times more negative than 3D. If the Output value from component 1 with 2D is -3, changing to 3D results in an output of -2.

Output value 2D = 1.5 x Output value 3D

If ***two*** components are used, only one must be set negative because negative times negative is positive. I have not measured the value for a doubly negated texture (too convoluted). Either component 1 or 2 can be set to negative, either 2D or 3D, the factor 1.5 applies here as well.

Using two components introduces a multiplication of 10 as we know from two positive components, so this is consistent.

If all ***three*** components are used, either one can be set to negative, 2D or 3D. But it is also possible to have all three components negative since three is an odd number and the result is different whether 2D is used trice or 3D.

Using trice 2D introduces another multiplier of 3.6, the Output value will be 3.6 times more negative than when only one component is set to 2D.

Using trice 3D doubles the multiplier from 3.6 to 7.2. The Output value will be 7.2 times more negative than when only one component is set to 3D.

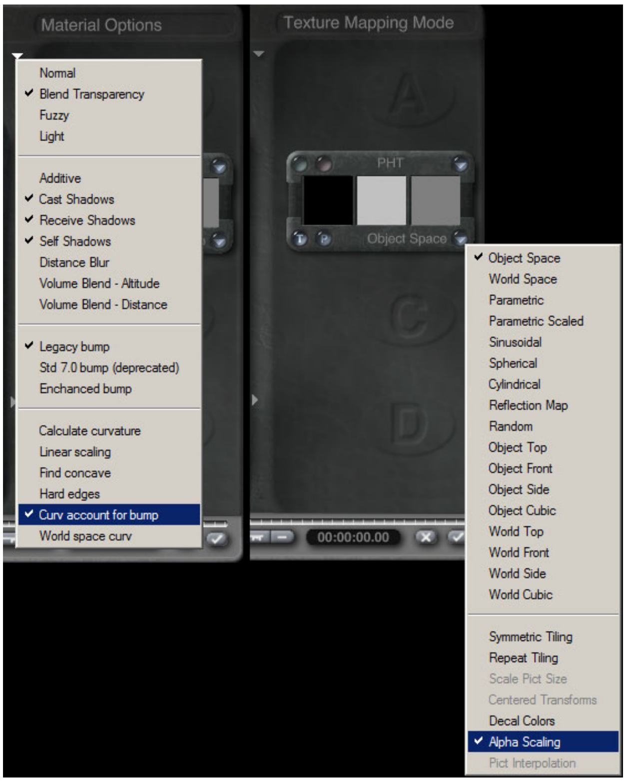
If we negate all three components, we can also mix 2D and 3D. We can have two components at 2D and one at 3D, or two at 3D and one at 2D. Let us assume the combination that outputs the weakest value is defined as 1, and then we find:

(a) 2D – 2D – 3D = 1.0;

(b) 2D – 3D – 3D = 1.5 x (a);

(c) 3D – 3D – 3D = 1.5 x (b) = 2.25 x (a).

This is all awfully complicated and mixing 2D and 3D is probably not the easiest way to go about this. But it does follow rules and can therefore be calculated.



The Phase value and Octaves behave exactly the same as for positive values. Double the Phase value makes the Output value four times more negative and adding an Octave makes it half as negative. Unfortunately, it was not possible to establish the relationship of the values between positive and negative outputs.

Limitation

If the *Curvature accounts for bump* option is enabled in the *Material Tools*, *Alpha Scaling* for the PHT does not work even if it is enabled. This may be considered a bug. However, the level of the PHT can be controlled by the Phase value and Octave setting to compensate for that.

Measurement methods

The behaviour of the positive Phased Hyper Texture was simpler to assess than for the negative one. A dark sphere lit by a white HDRI at quality 256 was brightened to a defined grey level by the PHT. The grey level was measured on screen and Diffusion adjusted until the predefined pixel values could be measured on screen with the *Just Color Picker*. The Diffusion setting was used to determine the level. The display is gamma corrected and useless for linear measurements, except when always the same brightness is measured.

For the negative PHT, the sphere was lit by Ambience using a separate PHT to give it enough brightness. Negative diffusion was used to dim the grey sphere to the predefined display brightness. The Diffuse values to dim the sphere were very low and thus some precision was lost. Turning up Ambience with the separate positive PHT was limited by strange artefacts that appeared so the level had to be kept quite low.

The measurements were taken in steps of half-light and double-light because this corresponds to how a full f-stop on a real camera works. This is also consistent with a linear workflow. The measured values were then normalised so they could be compared.

Phase works square (changing by two full f-stops) and Octaves linear (changing by 1 full f-stop). Adding a component multiplies the output by ten.

Conclusion

This Phased Hyper Texture is easy to use and works like Bryce light sources do. It is perfectly suited for a linear workflow. It has a wide output range that can be easily controlled.

References

Linear Workflow: https://www.horo.ch/docs/mine/pdf/LinearWorkflow.pdf

Brightness and Falloff: https://www.horo.ch/docs/mine/pdf/Brightness.pdf

Hyper Texture: https://www.horo.ch/docs/memo/pdf/HyperTexture.pdf

Deep Texture Editor DTE: https://www.horo.ch/docs/mine/pdf/DTE.pdf

Software Mentioned

Just Color Picker: http://annystudio.com/software/colorpicker/

August 2014/horo