${\displaystyle {{\bf Sycles}}\atop {\rm Cycles integration into Softimage}}$ 

# How to...

September 17, 2020

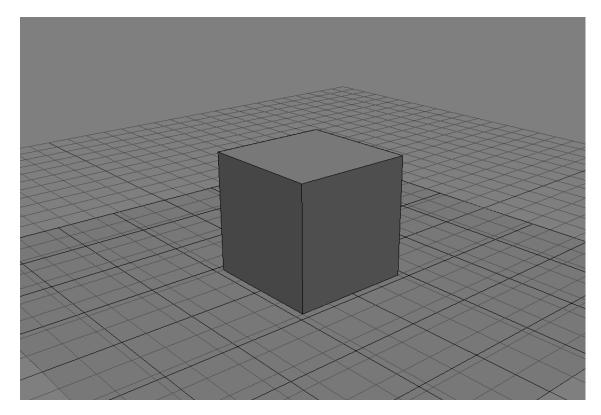
# Contents

1	How to start rendering	4
<b>2</b>	How to assign shaders	7
3	How to use sky and sun for environment	9
4	How to use HDR for the environment	13
5	How to use Bump	14
6	How to make true displacement	17
7	How to use Normal Map	19
8	How to render hairs	20
9	How to render DOF	<b>22</b>
10	How to render Object ID pass	<b>24</b>
11	How to render ICE-attributes	<b>28</b>
12	How to render subdivide surfaces	30
13	How to render ICE-instances	34
14	How to render Vertex Color	37
15	How to render to Multi-Layer EXR	39
16	How to use Stamp Output	42
17	How to use Cache	44
18	How to use color profiles	47
19	How to use shaderballs	50
20	How to use OSL-shaders	<b>54</b>
<b>21</b>	How to render standard XSI hairs	57
22	How to use ray visibility	61
23	How to use multiple uv coordinates	64
24	What is Camera Cull and Distance Cull	67
25	How to render motion blur	69
26	How to use VDB Primitive	73
<b>27</b>	How to render OpenVDB	76

28	How to render emFluid	78
29	How to render Explosia FX	80
30	How to render and use Cryptomatte passes	82
31	How to render AOVs	86

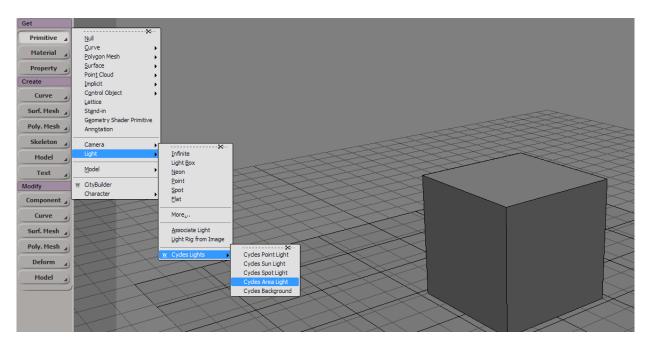
# 1 How to start rendering

Suppose we have a scene: a cube and a plane.



Add a rectangular light source, by choosing

Primitive - Light - Cycles Lights - Cycles Area Light



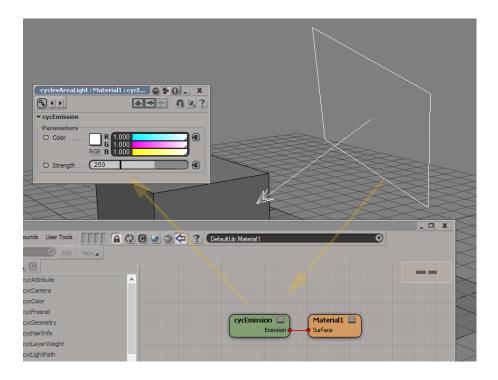
Set its size 5x5.

Scene_Root : cyclesAreaLight (Gen 🞱 🗣 🕖 💶 🗶	
cyclesArea	
Name _ cyclesAreaLight	
▼ cyclesArea	
Null Visible Settings           Arrow         1           Length         0           Arrow Size         0.05	
Area Light Settings           O Size U	
O Portal	
Generic Settings Max. 1024 Bounces Cast Shadow.	
CRay Visibility	
O Use Diffuse	
Use Glossy	
Use Transmission	
Use Scatter	

Then select the render area and in the settings we specify Cycles Renderer.

	ViewB : Render Region Options (Re 💿 🕤 📘 🗶	
	<b>▲</b> ◀► <b>∩ ∪</b> ?	
	Render Region Options mental ray	
	▼ Render Region Options ?	
	Render Region Options	
	UCycles Renderer Hardware Renderer	
	mental ray Selection	
	Use Scene Render Options Render Selection Only	
	Region Motion Blur	
	Enable Motion Blur	
	Set Pass Crop From Region	
	▼ mental ray ?	
	Rendering Preview Optimization Framebuffer	
	Shadows Ambient Occlusion Final Gathering	
	GI and Caustics Irradiance Particles Photon Global	
	Diagnostics mi Archives String Options	
	Primary Rays	
T	Type Scanline	
-+-	Sampling/Aliasing	
	Unified Sampling	
	Aliasing (Raytracing & Scanline Only)	$\mathbb{R}$
	Min Level	
+	Max Level	
-+-	Motion Samples	
	Sampling Contrast	
	<b>R</b> 0.200	
-	RGB B 0.200	
	A 0.200	
	Motion Steps	
	Transform (1	
t	Deformation (1	
-	CEnabled features	

Nothing to see. This is because the power of the light source is too small. Go to the Render Tree of the light source and set the value Strength of the node Emission equal to 250.

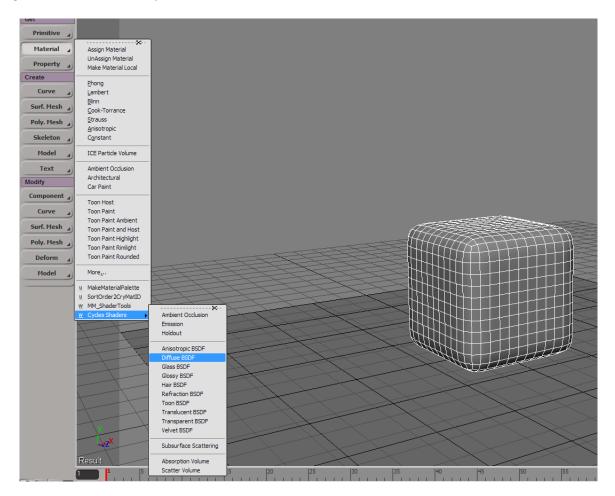


We get the result.

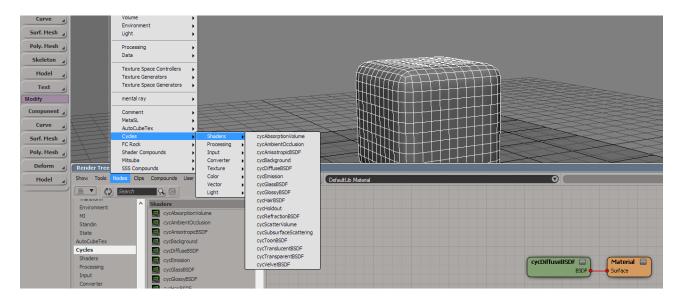
ViewB : Render Region Options (Rendering)		
	N	
Render Region Options Cycles Renderer Options		
✓ Render Region Options		
Render Region Options Use Current Pass Options		
Region Renderer Selection		
Cycles Renderer   Render Selection Only		
Region Motion Blur		
Enable Motion Blur		
Set Pass Crop From Region		
▼ Cycles Renderer Options		2
Sampling Light Paths Film Background Performance		8
O Sampling Path Tracing		
Samples O AA Samples (128		$\approx$
		$\geq$
Settings O Seed		$\sim\sim$
C Clamp Direct. 0		$\sim$
		$\sim$
		$\bigcirc$
Volume Sampling		$\sim$
O Step Size 0.1		$\sim$
☐ Max Steps (1024		$\sim$
		$\frown$
		$\sim$
		$\sim$
	$\square \times \times^{\bullet} \times ^{\bullet}$	<
	$ \land \land$	
		-

### 2 How to assign shaders

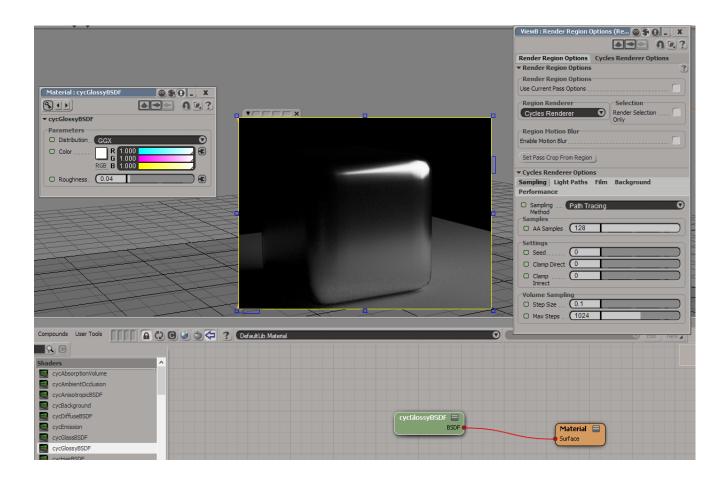
Two ways. You can use the command Material - Cycles Shaders and select which shader to assign to the selected object.



Or you can directly connect nodes in Render Tree. Everything that Cycles understands is in a separate section.

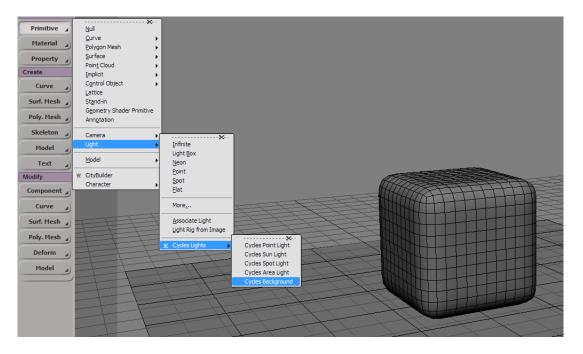


As an example, let's connect the node GlossyBSDF to the port Surface of the material's root node. We well obtain the result.

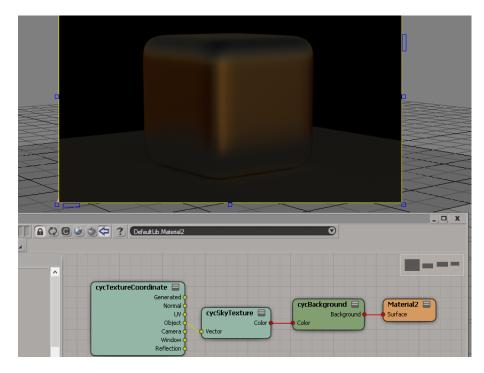


### 3 How to use sky and sun for environment

Suppose we have a scene: a cube and a plane. Adding a light source of the type Background.



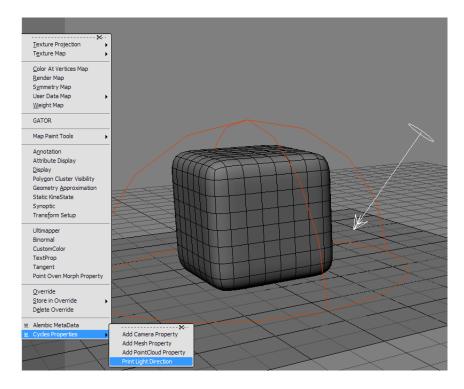
To the shader of this light source add the node SkyTexture. Connect it to the port Color of the node Background. To the port Vector of the node SkyTexture connect the output port Object of the node TextureCoordinate. This is necessary in order to explicitly tell the renderer which texture coordinates for the environment to use. Render.



Something wrong. This is because the vector for Sun Direction has the value (0, 0, 1). This is means that the sun directed horizontally. We should correct it. Add the light source Cycles Sun Light.

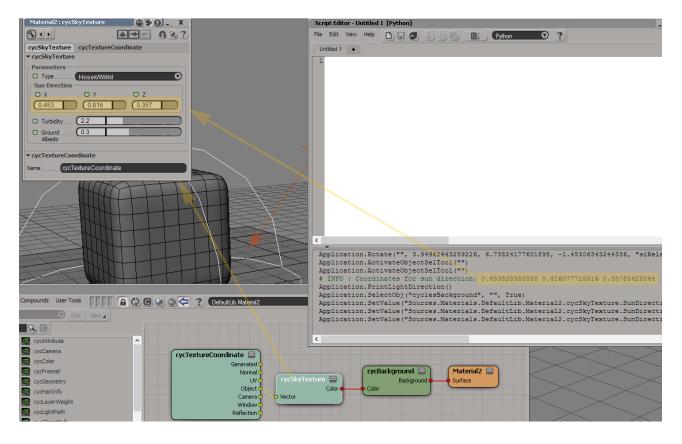
<u>F</u> ile	<u>E</u> dit	<u>V</u> iew	<u>D</u> isplay	Fabric	MaSs menu	GEAR_mc	<u>W</u> indow	<u>H</u> elp	Model	Animate	Render	ICE	Simulate
Mod	el	•	B	Car	mera 🔻		Ē Ū	۰ ش	ΥZ				
Get					Scene_Ro	oot : cycles	Backgrou	ınd : Li	05	Э_ Х			
_		-			×	· ]			•	9.93	2		
P	rimitiv	ve 🔺	<u>N</u> ull Cur			ture		_			_		
1	lateri	al 🦼	-	ve Igon Mesł		s s							
P	roper	tv .	Sur			• (Ho	sek/Wilkil		_	Ø			
	<u> </u>			n <u>t</u> Cloud		• ion							
Crea	ate	_	<u>I</u> mp			• c	) Y C		Οz				
	Curve	• 4	Con Latt	itrol Objei	ct		0		1				
S	urf. Me	sh .		nd-in		(2.		_					
_	_	_			ader Primitive			-		=			
Po	oly. Me	esh ∡	Ann	otation			3	_					
s	keleta	on 🖌	Can	nera				· · X· · ]					<u> </u>
$\sim$	Mode		Ligh	it		<u>I</u> nfinit	te	~	Ž=		- F		
_		· •	Mod	lel.		Light	<u>B</u> ox						<u> </u>
	Text		<u></u>			Neon			==+				
Mod	lify		₩ City			Point Spot				<u> </u>	<u> </u>	<u> </u>	
6	mpon	ent ⊿	Cha	racter		Elat			7		-		
_	mpon										77		$\rightarrow$
	Curve	• _				More,			$\rightarrow$		=7=	7-	
Su	urf. Me	sh 🖌				Assoc	iate Light				+	T_	
						Light	Rig from In	nage 📕	- <u>}-</u> ==	+		7-	
PC	oly. Me	esn ⊿	-+			<u>w</u> Cyde	s Liahts	•	Cycles	s Point Light	<·· ]	ل	
	Defor	n _)	-+				l Ligitto			s Sun Light			+ +
	Mode					-+				s Spot Light			-+
_	Floue	•	)			-+				s Area Light			+-+
						1			Cycles	s Backgroun	d		1 _
			Render						or or or or o	-			
			Show	Tools N	lodes Clips	Compound	s User To	ools			C		?
			<b>*</b>	Q	Search	90							

Next we need to know its direction. To do this, select the sun and choose

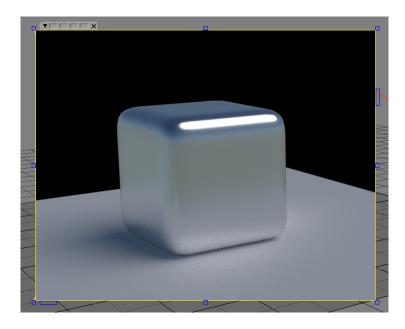


Property - CyclesProperties - Print Light Direction.

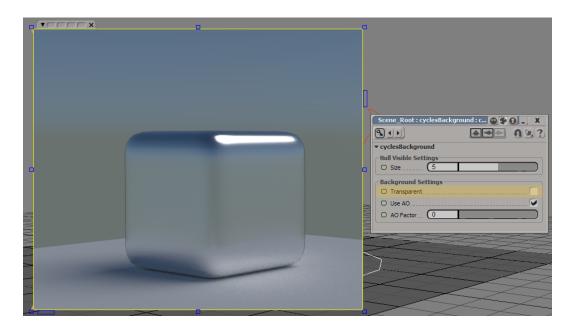
Three numbers will appear in the log, which we should insert into the fields of the direction of the sun in the node SkyTexture.



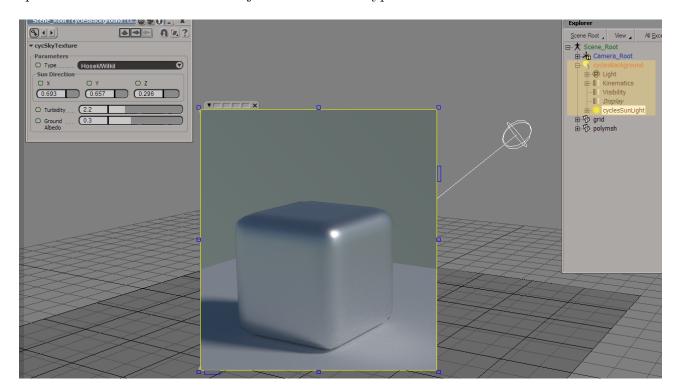
Render. Now the color of the sky and the direction of the sun are coordinated.



Yes, to see the actual color of the sky, it is necessary to check off the parameter **Transparent** in the properties of a light source of the type **Background**.

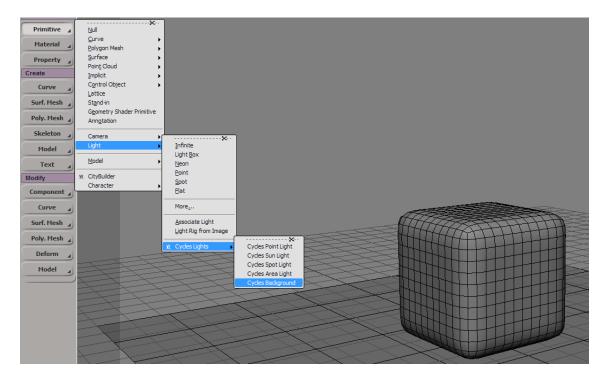


Maybe someone will say: "What the f..! Every time after the sun's shift, manually rewrite the values of the direction?!" Our answer is "No". If the sun is a child of the background object, then the renderer will ignore the values in the fields **Sun Direction**, and using the spatial values of the first child object that has the type of the sun.

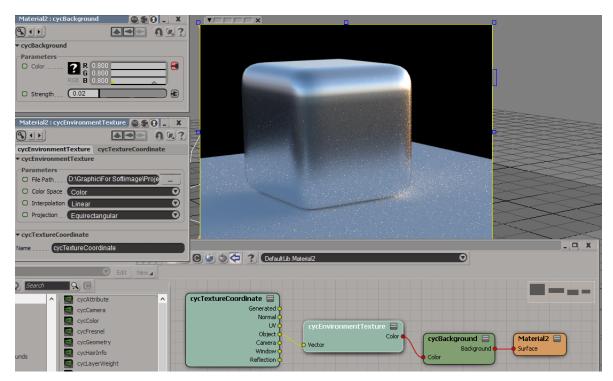


# 4 How to use HDR for the environment

Suppose we have a scene: a cube and a plane. Let's add a light source of the type Cycles Background.

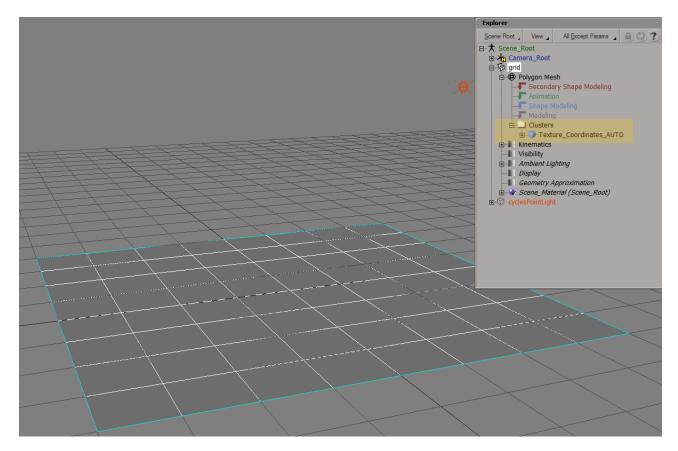


Add into the shader the node EnvironmentTexture and connect it to the port Color of the node Background. Connect the output of the node TextureCoordinates to the port Vector. Next, select in the node EnvironmentTexture any picture, set the strength of the light source and render it.

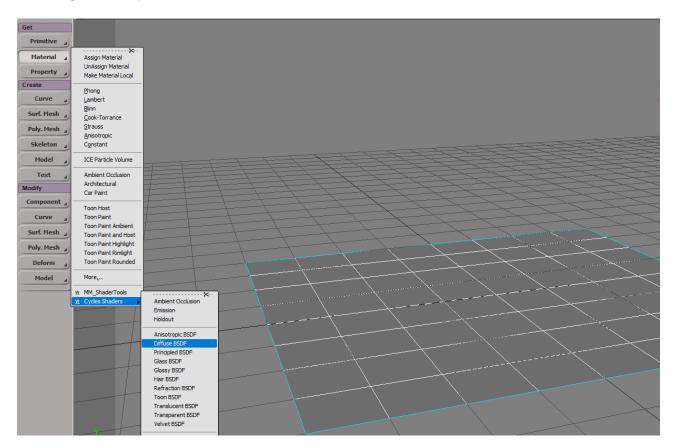


# 5 How to use Bump

Suppose we have a scene: the plane and the light source. The plane contains uv coordinates.



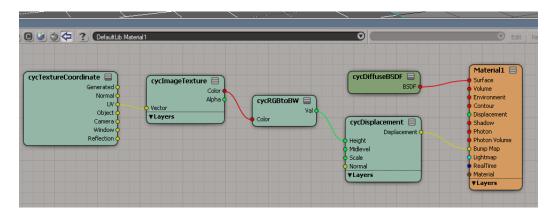
Assign to the plane the shaser DiffuseBSDF.



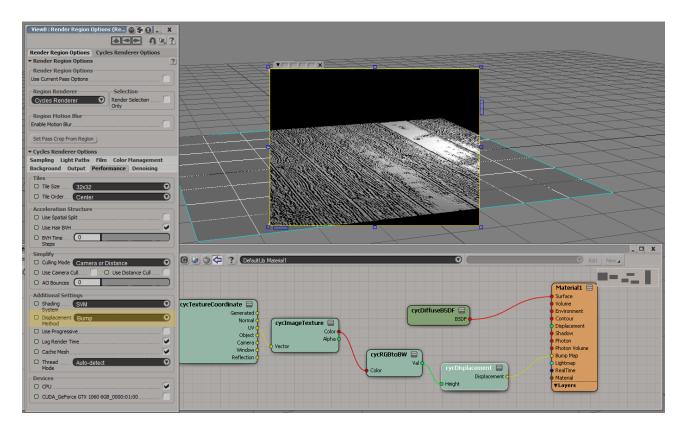
Next add to this shader two nodes: TextureCoordinate and ImageTexture. Connect the output UV of the first node to the input Vector of the second one. Select the image for the bump map.

S I S cycIma • cycIma • Param • File • Cok • Inte • Proj • Exte	geTexture cycTextureC ageTexture eters						lygon Mesh Secondary Shape Mo Animation Shape Modeling Modeling Clusters Texture_Coordina nematics sibility aterial1 mbient Lighting splay sametry Approximation
unds User Tools	a 🗘 o 😉 🌣 🗢	? Default Lib.Material 1			0		T E
G							
cycBrickTexture							
cycCheckerTexture							
cycEnvironmentTexture							
cycImageTexture	CY	TextureCoordinate and Generated		nageTexture 目			
cycMagicTexture		Norma		Color e	CucDi	ffuseBSDF	Material1
cycMusgraveTexture		U\		Alpha 🔿	Cycol	BSDF	- Surface
cycNoiseTexture		Object Camera			<u> </u>		
cycSkyTexture		Window					
cycVoronoiTexture		Reflection	19				
cycWaveTexture							

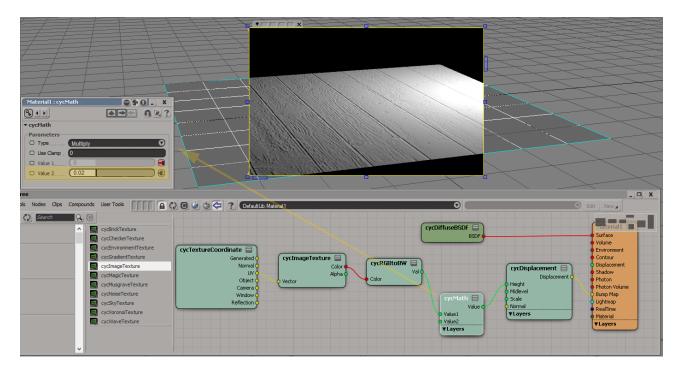
Convert the output Color into black-and-white image with the help of the node RGBtoBW, and connect the result to the input Height of the node Displacement. The unique output of this node has vector type. That's why we connect it to the input port Bump Map of the material's root node. This is the unique vector type port of it.



In the render properties set the mode Displacement Method to the Bump. Render.

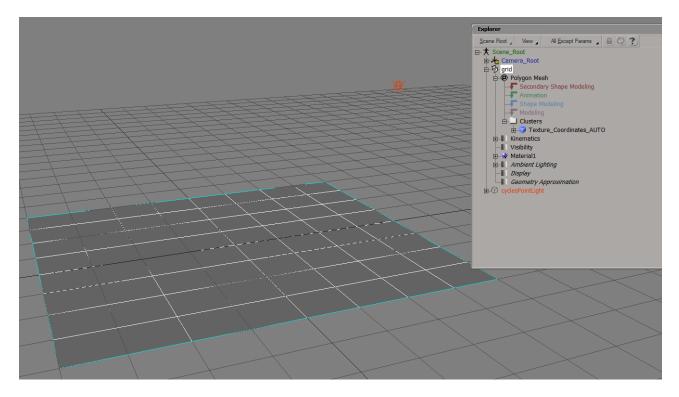


To adjust the bump strength we can use the node Math. Insert it before the port Height of the node Displacement, switch the node to Multiply and the value Value 2 be responsible to the power of the bump.

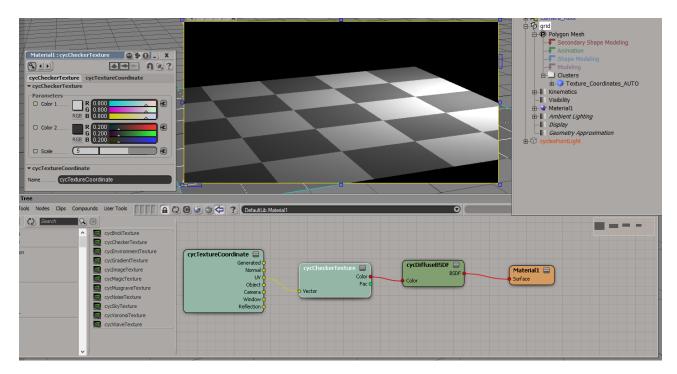


## 6 How to make true displacement

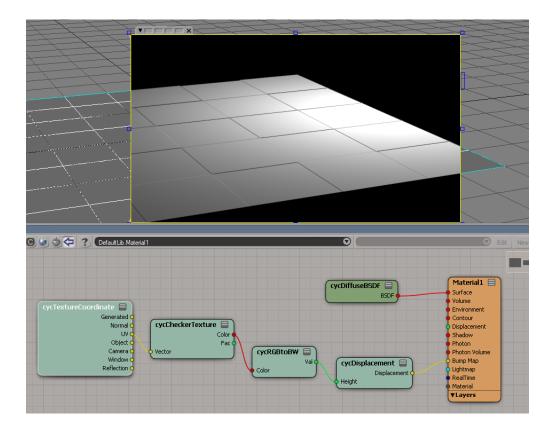
Suppose we have a scene: the plane and the light source. The plane contains uv coordinated.



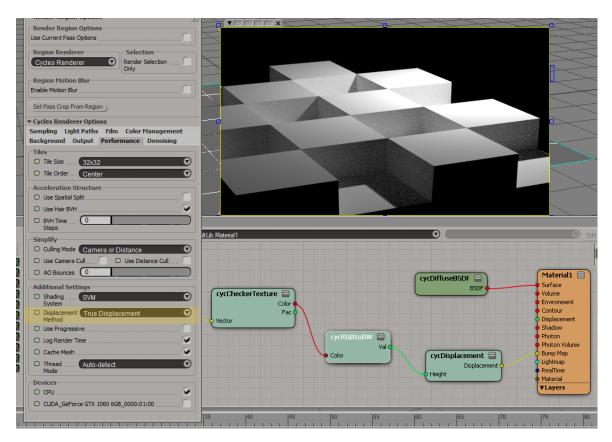
Assign the shader DiffuseBSDF o the plane and add to it the node CheckerTexture. Connect it output to the port Color of the node DiffuseBSDF. To make the texture overlap along texture coordinates, we should use the node TextureCoordinate.



Connect the output Color to the port Height of the node Displacement by passing throw the converter to white-and-black format. The output of the node Displacement connect to the port Bump Map of the material's root node. Render and see the simple bump.



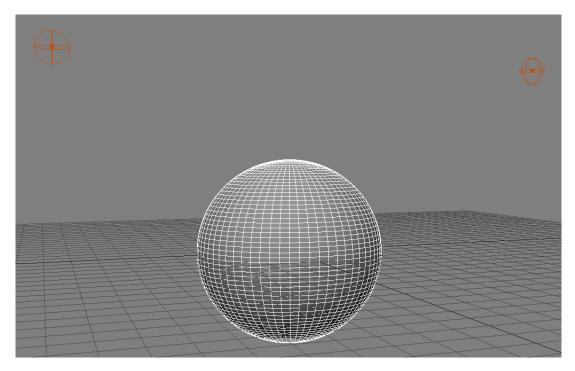
But how we can create displacement? To do this we should fo to the render properties and in the section Performance choose True displacement for the parameter Displacement Method. For correct result increase the number of polygons on the plane.



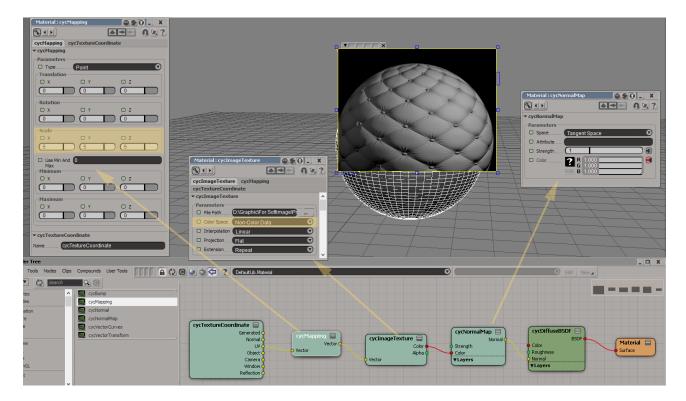
In the most cases we should use the method Both Displacement. In this mode the geometry deformed by texture, and if the number of polygons is not enough, then the deformation simulated by bump.

# 7 How to use Normal Map

Suppose we have a scene: a sphere and a pair of light sources. On the sphere, of course, there are texture coordinates.



Add to the shader of the sphere the node NormalMap. To the port Color connect the node ImageTexture. In the node for the texture chose the actual file with the normal map and select the value Non-Color Data for the value of the parameter Color Space (so that gamma correction does not occur). Finally, specify which texture coordinates should be used for the normal map. The connection is made through the node Mapping so that you can specify a five times tiling of the map.

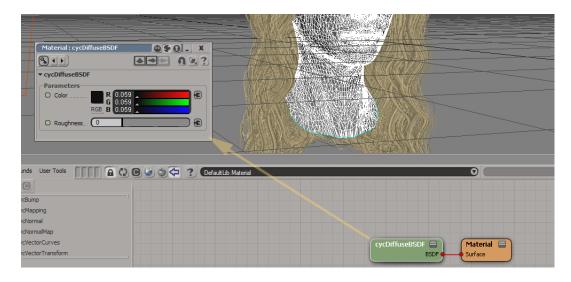


# 8 How to render hairs

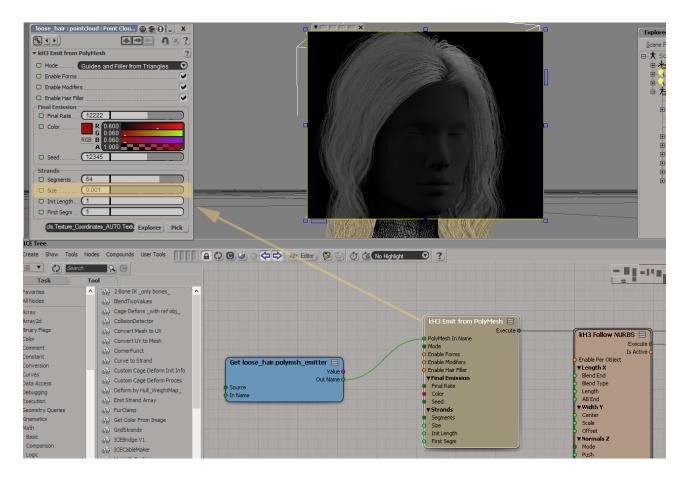
Suppose we have a head with hair made with strands.



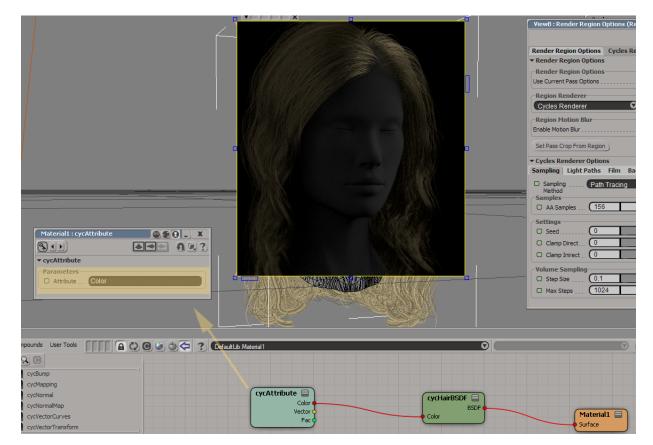
So that the head does not get in the way, we assign it a dark material.



We go into the ICE-tree of the hair and set their thickness. In our case, set the value of the parameter Size = 0.001. We do not touch anything else. Render.

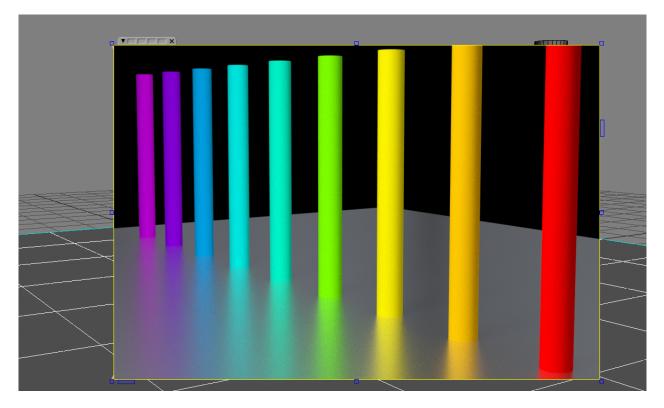


Now the hair has standard DiffuseBSDF shader, and he can not understand what color of the hair should be. Assign a shader HairBSDF, add the node Attribute. Set the name of the attribute Color and connect the output Color port to the port Color of the node HairBSDF. Now that's another matter.

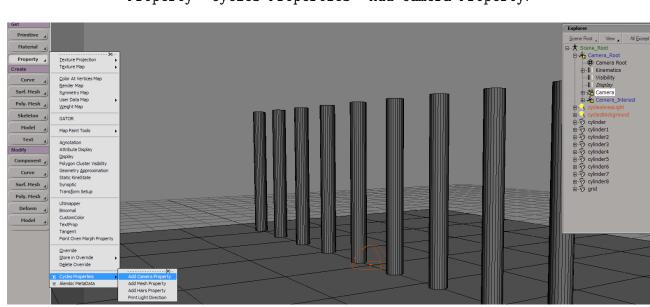


# 9 How to render DOF

Suppose we have a scene: the plane and the colored bars. We want the green bar to be in focus, and the rest are blurred.

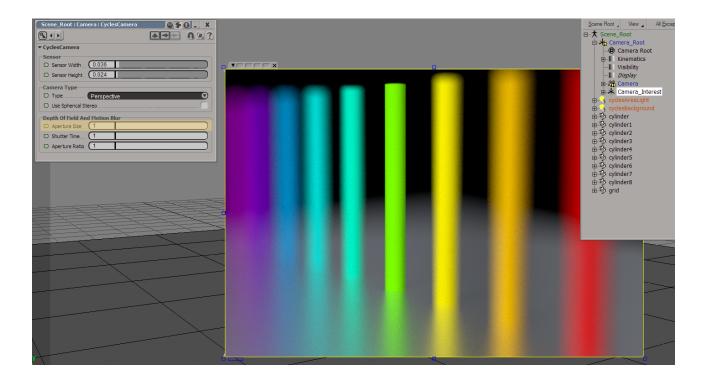


Select the render camera and add the property by the command



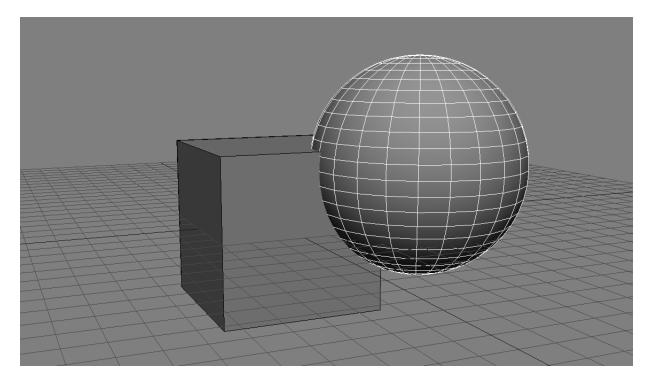
 $\label{eq:property-cycles} {\tt Properties-Add \ Camera \ Property}.$ 

Set Apertue Size = 1. The focus of the camera will be in the same place as Camera\_Interest.



# 10 How to render Object ID pass

Suppose we have a scene: a cube and a sphere. And they are intersected.

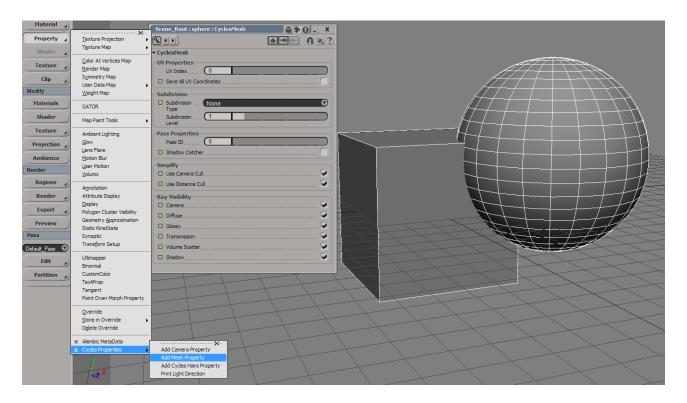


We would like to render Object ID pass, and then use it for compositing. Now this channel is empty. Everything is black.

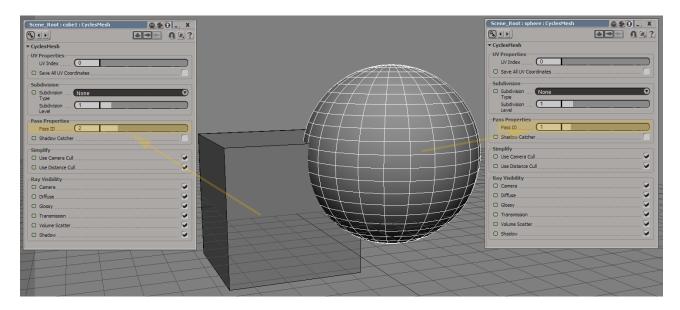
ViewB : Render Region Options (Rendering)		
A		
Render Region Options Cycles Renderer Options		
▼ Render Region Options ?		
Render Region Options		
Use Current Pass Options		
Region Renderer Selection		
Use Scene Render Options 💿 Render Selection Only		
Region Motion Blur		
Enable Motion Blur		
Set Pass Crop From Region		
▼ Cycles Renderer Options		
Sampling Light Paths Film Color Management Background	2017	
Output Performance Denoising		
Passes O Fill Pass Alpha	The second se	
O Assign Unique Pass Id		
O Show Pass PASS_OBJECT_ID		
Multi-Layer EXR		
Combine Render Passes To Single EXR	-	
O Save Separate Passes		
Land		
- Land - to the		
	L.	

Select both objects and assign them the  $\tt CyclesMesh$  property. To do this, choose the command

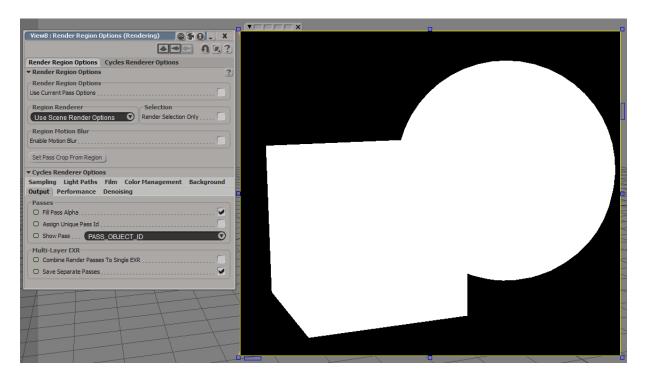
Property - Cycles Properties - Add Mesh Property.



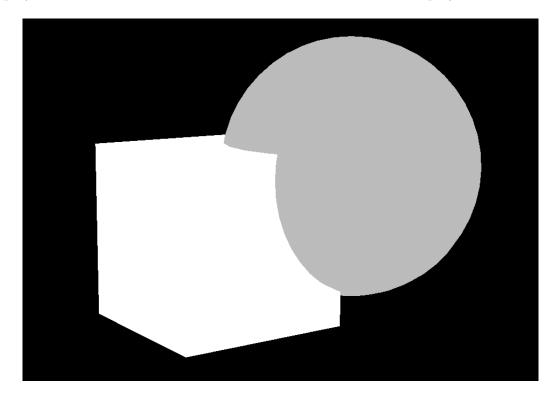
For the sphere set Pass ID = 1, and for cube Pass ID = 2.



Render, and see that now the channel contains the data.



It seems that both objects are filled with the same color, but it is not. These are different shades of gray. If you render a 32-bit channel and shift levels, you can easily verify this. In general, the meaning of the channel Object ID is that objects with the same value Pass ID are displayed in one color. Those who have Pass ID = 0 are not displayed at all.



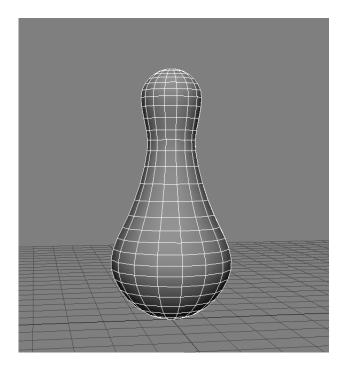
In the render properties you can turn on the option Output - Passes - Assign Unique Pass Id. After that, all the objects in the scene will automatically be assigned different values of the Pass ID.

ViewB : Render Region Options (Rendering)						
Render Region Options Cycles Renderer Options						
▼ Render Region Options ?						
CRender Region Options						
Use Current Pass Options						
Region Renderer Selection						
Use Scene Render Options   Render Selection Only						
Region Motion Blur						
Enable Motion Blur						
Set Pass Crop From Region						
▼ Cycles Renderer Options						
Sampling Light Paths Film Color Management Background						
Output Performance Denoising						
Passes						
O Fill Pass Alpha						
O Assign Unique Pass Id						
O Show Pass PASS_OBJECT_ID						
Multi-Layer EXR						
O Combine Render Passes To Single EXR						
O Save Separate Passes						

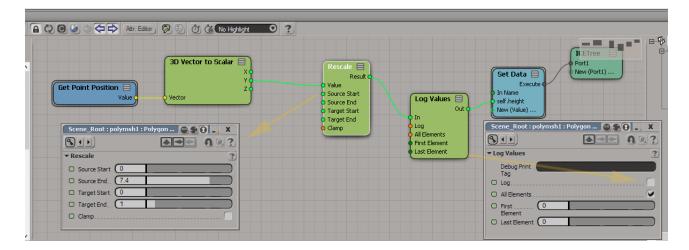
The output of the channel Material ID works in a similar way. It always have different values for different materials.

#### 11 How to render ICE-attributes

Suppose we have a scene with a simple object.

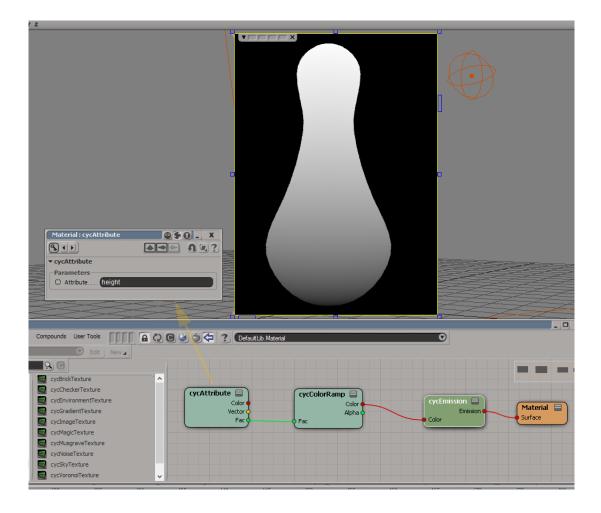


This object contains the following ICE-tree:



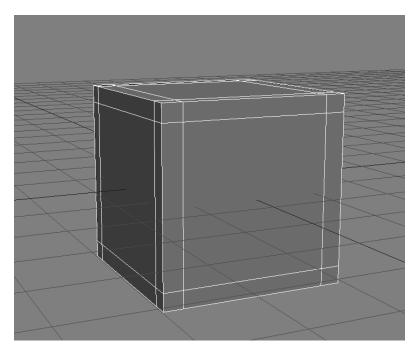
This ICE-tree writes to each vertex a value from 0 to 1, depending on how high the vertex is above the ground. In the node Rescale the value 7.4 is the average height of the object. All values are stored to the height attribute. The node Log Values is used to force the ICE to calculate values. The ICE is very clever, and if it think that the attribute is not used anywhere, it often does not calculate it. By the node Log Values we force to do all calculations, even with the Log option disabled.

Assign to our object Emission-shader. Add the node Attribute and set the name of the ICE attribute (height in our case). Finally path the float-output of this node through Color Ramp and connect everything to the port Color. On the render we see, how the object's color changes from black to white from the bottom to top.

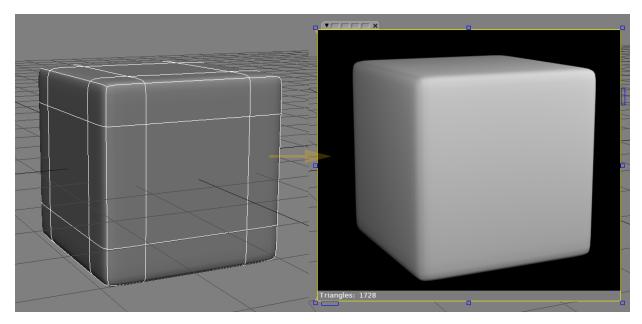


# 12 How to render subdivide surfaces

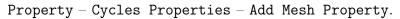
Suppose we have a cube.

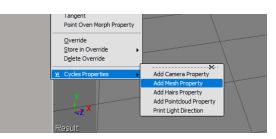


If you press plus on the keyboard 3 times, the cube will become smoothed. And on the render it is exactly the same.



If you would like to do a subdivision of the object only during the render, then assign the property CyclesMesh to the object by choosing the command





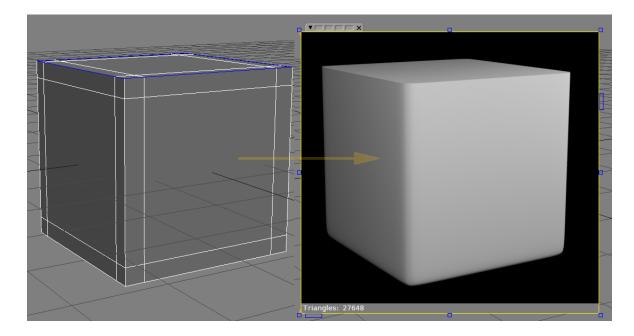
On the tab Subdivision choose the type Catmull-Clark and set the number of subdivision steps equal to 4. Render it.

	Scene_Root : cube : CyclesMesh 🚳 🧙 🚯 💶 🗶	
	▼CydesMesh	
	UV Subdivision Pass Motion Simplify Visibility	
	Properties	1
7		
$\rightarrow$	Туре	
	Subdivision 4	
_	_Rules	
	O Vertex Sharpen Edges and Corners O	
	Boundary Interpolation	
	O Triangle Catmark Weights	
	Subdivision Weights	
7		~~
+		
	Triangles: 27648	

If you select the type Linear, then the polygons will be subdivided, but the form of the object remains the same. Useful when using displacement.

Scene_Root : cube : CyclesMesh 🕲 🛠 🖸 💶 🗶	
▼CyclesMesh	
UV Subdivision Pass Motion Simplify Visibility	
Type Subdivision (4	
Level	
Rules Vertex Sharpen Edges and Corners	
Boundary Interpolation	
Triangle Catmark Weights     Subdivsion	
Weights	$\rightarrow$
T	~~~
Triangles: 27648	

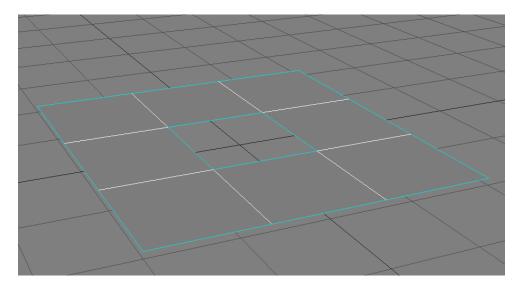
An object can have hard edges, but even in this case the result on the render is correct.



Properties have two additional options:

Vertex Boundary Interpolation and Triangle Subdivision Weights.

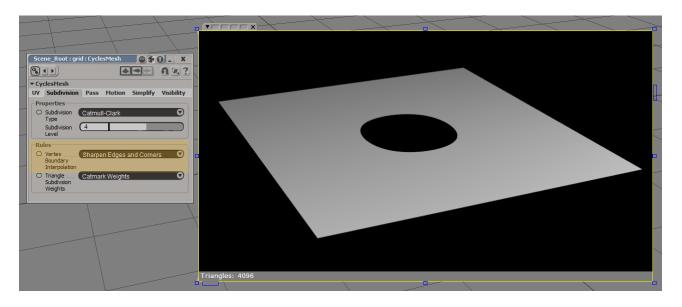
The first option Vertex Boundary Interpolation define how to smooth the edges and vertices on the boundary. Consider an example. Take a plane with a hole in the middle.



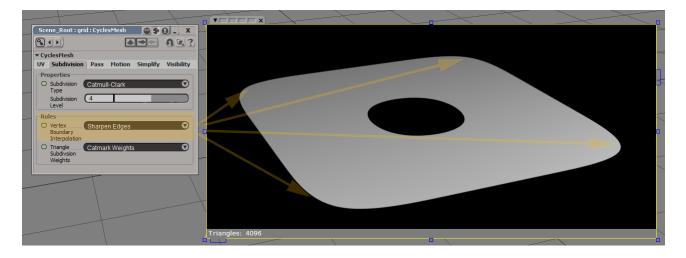
If the value of the parameter Vertex Boundary Interpolation is equal to

#### Sharpen Edges and Corners,

then both edges and vertices will be snapping to the boundary of the original mesh.



If the value of this parameter is equal to Sharpen Edges, then corners will be smoothed.



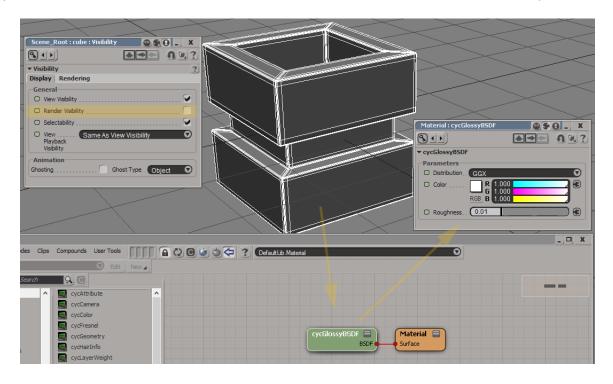
Parameter Triangle Subdivision Weights influences the way of subdividing the triangular polygons of the model. The difference between modes

Catmark Weights and Smooth Triangle Weights

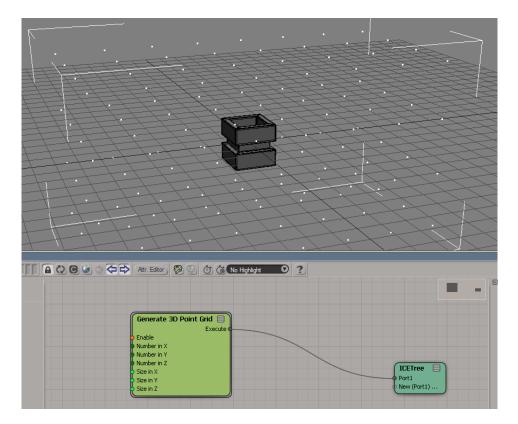
in most cases quite small.

#### 13 How to render ICE-instances

There are no features here. Everything works just like for Mental Ray. Suppose we have an object that we want to instantiate. It has some shader and exclude from visibility on render.



Create an empty Point Cloud, an ICE-tree and generate a cloud of points.



Next add the node Emit from Position. Set parameters as in the picture and connect the node Get Point Position to the port Emit Position. This will generate particles just in the places where the grid points were.

Scene_Root : poin	tcloud : Point Clo 😳 🗣 💽 💶 🗶			
<b>SI)</b>	A < A . ?			
✓ Emit from Positio		· ·		
* Elline from Posicio			and the second	
O Enable			47477777777777777777777777777777777777	
Emission Parame		£===£=£=£=£=£=£		
🛛 🔘 Select Rate . 🌔	Fotal Number of Particles 🛛 🕥			t B B B B B B B B B B B B B B B B B B B
Туре			+++++++++++++++++++++++++++++++++++++++	- C- C- C- C-
O Rate		E + + + + + + + + + + + + + + + + + + +	+++++++++++++++++++++++++++++++++++++++	
🗌 🗆 Seed 🕻	89012			1-01-2
Direction and Spe	hed			
Direction				
I O X	OY OZ			
		-t-t-t-t-t-	the second second	X
	0	-t-t-t-t-t-t-t	The the the	
<ul> <li>Direction</li> <li>Start Angle</li> </ul>		- Internet	Laterta to the total	+
	0		The forther the the	
Direction	· · · · · · · · · · · · · · · · · · ·	Internet the transfer to	-t-t-t-t-t-t-	
End Angle		Internet and the		
- O Speed (	•		The first of the T	
⊂ Initial Values —		and the prost	The first and the	
🗆 🗆 Mass 🕻	0.1	The first	- I - to the front	
O Size	0.1		the first limit	
			The I when a	
E Tree				
eate Show Tools N	odes Compounds User Tools	🔒 🗘 🖸 🎍 🗇 🖨 🖨 Attr. Editor 🗔 🖗	) () () No Highlight 🔍 ?	
▼ () Search		Enable		
Search	<u> </u>	Number in X		
Task	Tool	Number in Y		
Particles 💌	Add Bouding Box Point	Number in Z	Emit from Position	
in real	<ul> <li>Emit Blast</li> </ul>	Size in X		
leting Particles	🚔 Emit from Curve	Size in Y	• Enable	
fects	Emit from Geometry	Size in Z	Emission Parameters	
ission Control	Emit from Null		Emit Position	
nitters	Emit from Position		<ul> <li>Select Rate Type</li> </ul>	
rces ometry Instancing		Get Point Position 🗐 📃 🖉	• Rate	ICETree =
tters	Emit Splash from Surface Collision	Value 🦕 🚽 👘	Seed	Port1
als	Emit Trail at Surface Intersection		Direction and Speed     Direction	Port2
difiers			Randomize Direction Start Ang	🛛 New (Port2)
tion Control			Randomize Direction End Angl	
ientation			O Speed	
esets			▼ Initial ¥alues	
tters			Mass	
awning			O Size	
ates			Color     Shape	
	✓		O Drientation	

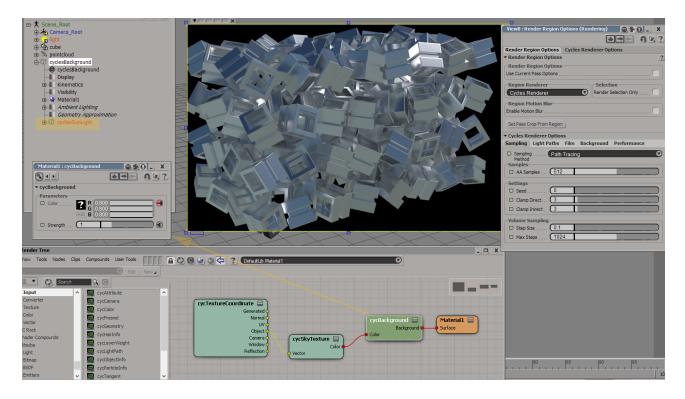
Instead of particles, you can use primitives like Disc, Rectangle, Sphere, Box, Cylinder, Cone and Capsule (looks like a Sphere). You can also use any polygon object. Using the node Instance Shape set our master object.

Explorer  Scene Root , Vew Al Except Params  Scene Root  Scene Roo		
O Index		
	🔒 🗘 🖸 🍯 🗢 🗘 Atr. Editor 🗔 💿 👌 🍓 No Hi	ighlight 🖸 ?
E V Search Q C		Emit Position
Task Tool		Select Rate Type
Particles   Add Bouding Box Point	Get Point Position	Seed
Deleting Particles		Direction and Speed     Port1     Port2
Effects Effects		Direction Randomize Direction Start Ang New (Port2)
Emission Control Emit from Geometry		Randomize Direction End Angl
Emitters 👜 Emit from Null		Speed
Forces Emit from Position		Mass
Geometry Instancing Emit Splash from Surface Collision		Size
Getters Emit Trail at Surface Intersection	Instance Shape	Color
Modifiers	Shape O	Shape Orientation
Motion Control	Index	State ID
Orientation		Execute on Emit1
Presets		New (Execute on Emit1)
Setters		

Add a bit of randomness to the orientation of the instances.

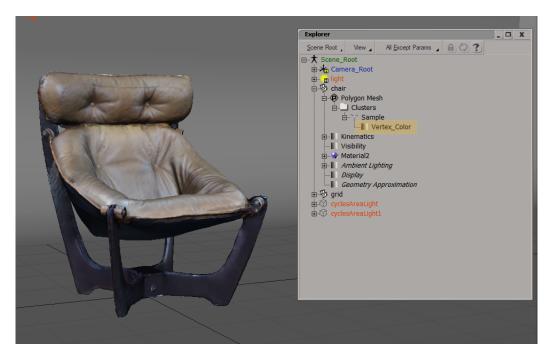
	id : Point Clo 😳 🗣 💽 💶 🗙 🕂 👘			
▼ Randomize Rotation b				
🗌 🗆 Twist Min 🛄 💶 -180				
Value				
- O Twist Max (180 Value	A			
O Start Angle. 0				
C End Angle 720				A A A A A A A A A A A A A A A A A A A
O Animated				
O Seed (51238				
ICE Tree				
Create Show Tools Node	s Compounds User Tools	🗘 🖸 🎱 🗢 🖨 Atr. Editor 🔗 😓 🙋	) 🐻 No Highlight 🔍 ?	
				P Rate
E 🔹 🗘 Search	<b></b>			
			Value o	Seed
Task Too			Value	Direction and Speed
Particles V			Value	VDirection and Speed
Particles V	l		Value	Direction and Speed
Particles   Deleting Particles	Add Bouding Box Point		Value	Direction and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed
Particles   Centroise  Deleting Particles  Effects	Add Bouding Box Point		Value g	VDirection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     VInitial Values
Particles   Deleting Particles	Add Bouding Box Point		Value d	Direction and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass
Particles   Particles  Deleting Particles  Effects  Emission Control	Add Bouding Box Point     Emit Blast     Emit finan Curve     Emit from Geometry		Value	VDirection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     VInitial Values
Particles  Particles Deleting Particles Effects Emission Control Emitters	al Add Bouding Box Point		Instance Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape
Particles  Particles Particles Proces Proces Geometry Instancing Getters	a) Add Bouding Box Point Comment Blast Comment Blast Comment From Curve Comment From Geometry Comment From Null Comment From Position	(Randomize Rotation by cone	Instance Shape	▼ Direction and Speed     ■     ■       Direction     Randomize Direction Start Ang       Randomize Direction End Angl       Speed       ♥ Initial Values       Mass       Size       Color       Shape       Orientation
Particles   Particles Particles Parti	al	Randomize Rotation by cone	Instance Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID
Particles  Particles Parti	al	Cone Axis	Instance Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID     Execute on Emit 1
Particles  Particles Parti	al	Cone Axis Twist Min Value	Instance Shape Index	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID
Particles  Particles Particles Prects Emission Control Emitters Forces Geometry Instancing Getters Goals Modifiers Modifiers Motion Control Orientation	al	Cone Axis Twist Min Value Twist Max Value	Instance Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID     Execute on Emit 1
Particles   Particles Particles Pa	al	Cone Axis O Twist Min Value O Twist Max Value Start Angle	Instance Shape Shape Index	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID     Execute on Emit 1
Particles  Particles Particles Prects Emission Control Emitters Forces Geometry Instancing Getters Goals Modifiers Modifiers Motion Control Orientation	al	Cone Axis Twist Min Value Twist Max Value	Instance Shape Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID     Execute on Emit 1
Particles   Particles Particles Parti	al	Cone Axis Cone Axis Twist Min Value Twist Max Value Start Angle End Angle	Instance Shape Shape	Virection and Speed     Direction     Randomize Direction Start Ang     Randomize Direction End Angl     Speed     Vinitial Values     Mass     Size     Color     Shape     Orientation     State ID     Execute on Emit 1

Finally, create a light source of the type **Background**, add a light source of the type **Sun** as a child object. To the background shader add the node **SkyTexture**. Render.

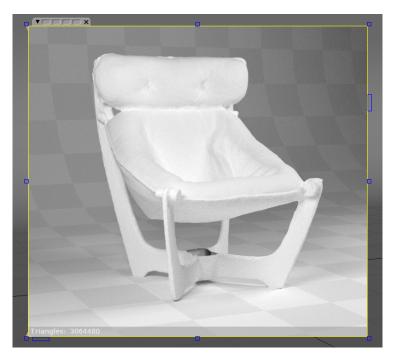


## 14 How to render Vertex Color

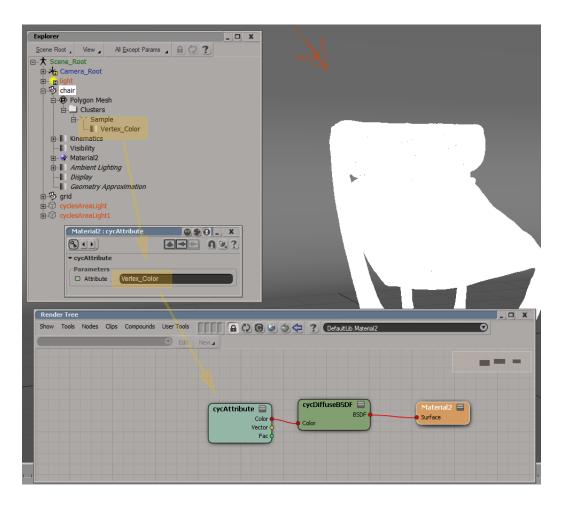
Suppose we have a scene with an object that contains Vertex color.



On the render this object has a neutral color, since Cycles does not know how it should use vertices' colors.



Add to the material the node Attribute and set the name of the cluster, which containing the vertex colors data (Vertex\_Color in our case).

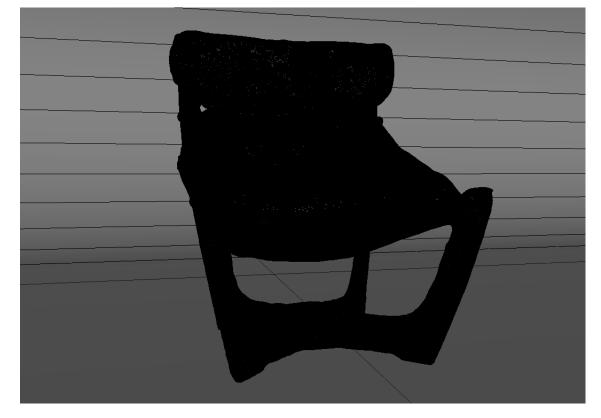


#### Render.

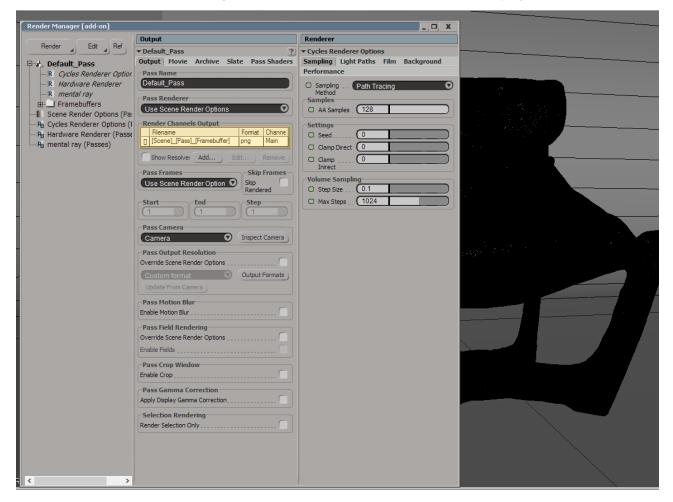


# 15 How to render to Multi-Layer EXR

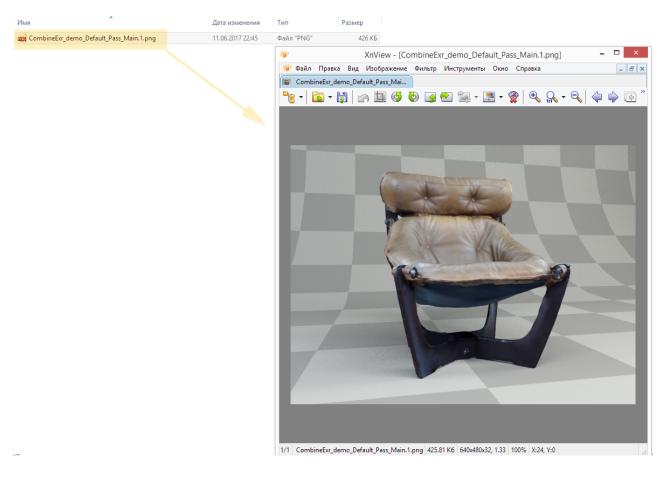
Suppose we have a scene: an armchair in a simple studio.



Select in the Render Manager one output channel and save it in \*.png format.



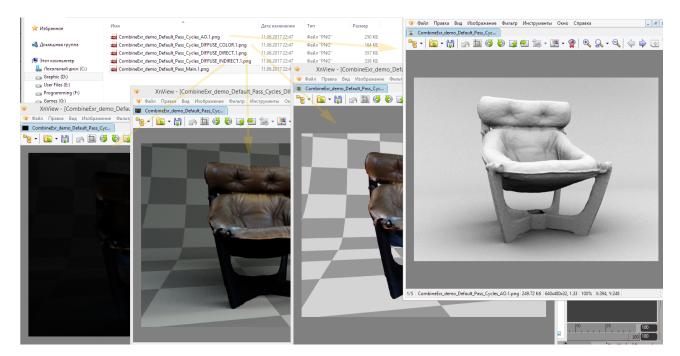
As a result we obtain one png-file.



Next add additional channels to the render. We will also save them in \*.png format.

Render Manager [add-on]				
	Output			Renderer
Render _ Edit _ Ref	▼ Default_Pass ?)			▼ Cycles Renderer Options
Default_Pass	Output Movie Archive Slate Pass Shaders			Sampling Light Paths Film Background
R Cycles Renderer Option	Pass Name			Performance
R Hardware Renderer	Default_Pass			Sampling Path Tracing
R mental ray	Pass Benderer			Method
Framebuffers	Use Scene Render Options			O AA Samples (128
Scene Render Options (Pa:	· · · · · · · · · · · · · · · · · · ·			
- RB Cycles Renderer Options (F		1 -		Settings
- Rg Hardware Renderer (Passe	Filename	Format	Channel	O Seed 0
- R <sub>B</sub> mental ray (Passes)	[] [Scene]_[Pass]_[Framebuffer]	png	Main	O Clamp Direct
	[ [Scene]_[Pass]_[Framebuffer]	png	Cycles AO	
	[ [Scene]_[Pass]_[Framebuffer]	png	Cycles DIFFUSE COLOR	O Clamp
	[ [Scene]_[Pass]_[Framebuffer]	png	Cycles DIFFUSE DIRECT	Inirect
-	[ [Scene]_[Pass]_[Framebuffer]	png	Cycles DIFFUSE INDIRECT	Volume Sampling
				O Step Size (0.1

As a result we get a set of files, each with its own data.

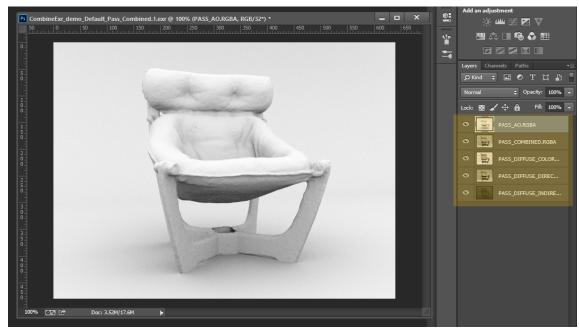


All selected for the render channels can be saved to a single layered 32-bit exr-file. For this, in the section Output - Multi-Layer EXR turn on the parameter

Combine Render Passes To Single EXR.

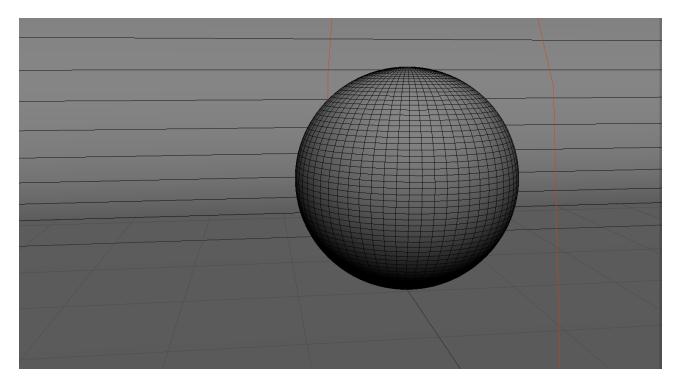
Render Manager [add-on]			_ O X	
	Output		Global Renderer	
Render _ Edit _ Ref	▼ Default_Pass	<u>e</u>	?」 ▼ Cycles Renderer Options	
🕀 🚽 Default Pass	Output Movie Archive Slate	Pass Shaders	Sampling Light Paths Film Color Management	
R Cycles Renderer Option	Pass Name		Background Output Performance Denoising	
R Hardware Renderer	Default_Pass		Passes	
R mental ray	David Barris		O Fill Pass Alpha	
R TestOslPlugin Options	Pass Renderer		O Assign Unique Pass Id	
Framebuffers	Use Scene Render Options	0		
Scene Render Options (Pa:	CRender Channels Output		Show Pass PASS_COMBINED	
- R Cycles Renderer Options (F	Filename Fo	omat Channel	-Multi-Layer EXR	
- Re Hardware Renderer (Passe	[ [Pass]_[Framebuffer] pn	ng Main	Combine Render Passes To Single EXR	
- R mental ray (Passes)	[ [Pass]_[Framebuffer] pn	ng Cycles AO	O Save Separate Passes	
RB TestOslPlugin Options (Pas	[ [Pass]_[Framebuffer] pn	ng Cycles DIFFUSE COLOR		
B restouringin options (rat	[ [Pass]_[Framebuffer] pn	ng Cycles DIFFUSE DIRECT		
	[ [Pass]_[Framebuffer] pn	ng Cycles DIFFUSE INDIRECT		
	Show Resolved Paths	Add Edit Remove		

As a result an exr-file with all selected channels will be created additionally. If you turn off the Save Separate Passes option, then no other files other than this layered file will be saved.



# 16 How to use Stamp Output

Suppose we have a scene: a glass sphere in a simple studio.



As a result, the rendered image is the following:



In the section Film - Stamp Output of the render properties turn on the options Render Time, Samples and Scene.

	_ <b>_</b> X
Output	Renderer
▼ Default_Pass ?	▼ Cycles Renderer Options
Output Movie Archive Slate Pass Shaders	Sampling Light Paths Film Background
/ Pass Name	Performance
Default_Pass	O Exposure 1
Pass Renderer	Filter Type Gaussian
Use Scene Render Options 📀	O Width 1.5
Render Channels Output	Stamp Output
Filename Format Channe	O Render Time
[ [Scene] png Main	🖸 Samples
Show Resolver Add Edit Remove	O Date
Pass Frames Skip Frames	O Frame
Use Scene Render Option 🛇 Skip	O Scene
Rendered	O Camera
Start End Sten	

As a result, a strip with information about the selected parameters will be added to the rendered image.



The meaning of the parameters is as follows:

Render Time – the render time of the image;

Samples – the number of AA samples in the render settings;

Date – date of rendering (day, month, year and time);

Frame – the number of the rendered frame;

Scene – the name of the scene;

Camera – the name of the camera from which the render takes place;

Triangles – the total number of triangles in all polygonal object of the scene;

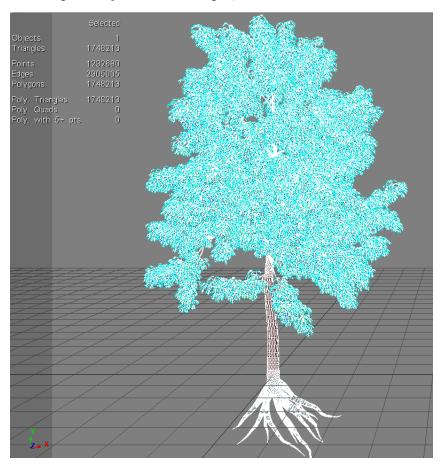
Curves – the total number of curves in all hair objects of the scene;

Objects – the number of objects on the scene;

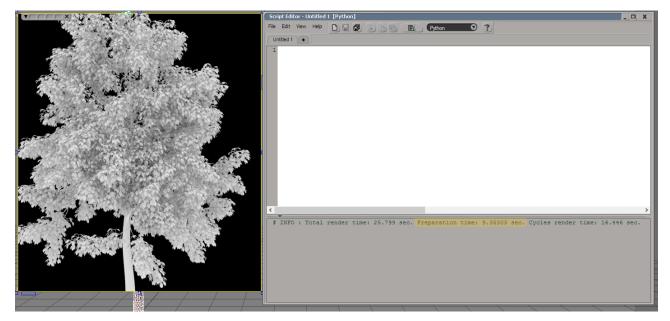
Lights – the number of light sources on the scene.

#### 17 How to use Cache

Suppose we have a complex object. For example, a tree.

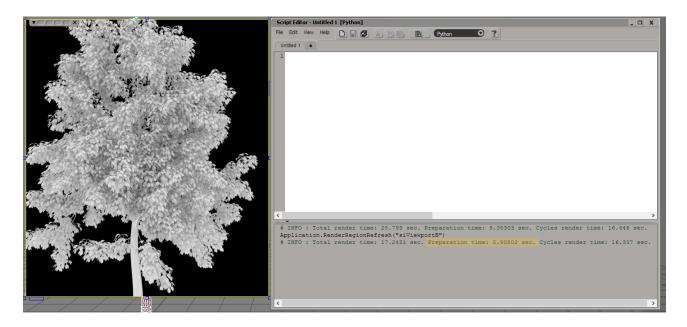


Render it and look at the log.



It says that the total render time is 25 seconds, of which 9 seconds is the preparation of the render. Preparation includes the collection data about the scene, what objects are on it, what shaders are assigned to them, and the longest time is export of all geometry of the scene to a format that the rendering engine understands. As a matter of fact all 9 seconds was spend to export geometry of a tree.

Do not touch anything and just click to update the render.



Now the preparation time for the scene took less than a second. For this, the cache is intended. After each render, all the data about the objects exported to the engine is saved, and if the object is to be rendered at the next time, the saved data is used instead of creating new ones. The cache is not used in the following cases:

- 1. The object changes the number of vertices or polygons;
- 2. The number of subdivision subdivision type is changed (from linear to Catmull-Clark or vice versa);
- 3. The index of used uv-coordinates map is changed;
- 4. The object shader requires an attribute that was not stored in the cache;
- 5. A full pass rendering.

In connection with these, it is necessary to remember all the time about the following:

- 1. If the object has a polygon cluster with new material, then the renderer will notice this only when the cache is updated;
- 2. If the object is deformed (by bones, for example), then the renderer will not update the location of the vertices;
- 3. If the shader requires an attribute that can not be generated (for example, texture coordinates are needed, but there is no ones on the object), the geometry export will happen every time again.

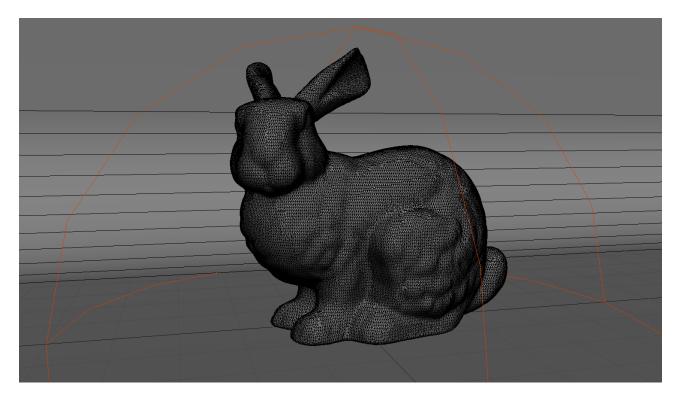
To force to export all geometry each time, it's enough to turn off the Cache Mesh parameter in the tab Performance of the render settings.

ViewB : Render Region Options (Rendering)
Render Region Options Cycles Renderer Options
▼ Render Region Options
Render Region Options
Use Current Pass Options
Region Renderer Selection
Cycles Renderer   Render Selection Only
Region Motion Blur
Enable Motion Blur
Set Pass Crop From Region
▼ Cycles Renderer Options
Sampling Light Paths Film Background Performance
O Tile Size 32x32
Tile Order Center
Additional Settings
🖸 Override Alpha in Passes
O Alpha Value (1
Convert to sRGB Buity Pass.
O Displacement Bump 🕥
O Progressive Rendering
O Log Render Time
O Cache Mesh
O Assign Unique Pass Id
O Show Pass PASS_COMBINED
Multi-Layer EXR
O Combine Render Passes To Single EXR
O Save Separate Passes

Since the re-creation of the cache occurs every time at the end of the rendering, the easiest way to update the cache of one particular object is to exclude this object from one render session, and then turn it back on.

## 18 How to use color profiles

Suppose we have a scene: a glass bunny in a simple studio.



Render it.



By default, if the content of the Combined pass is displayed in Region preview or stored in \* .png format, then it converts the color space to sRGB. This option can be turned off by turning off the option

Apply Color Profile to LDR Combined Pass

in the tab Color Management of the render settings.



After the shutdown, the render result is stored in the linear space.



There is the possibility to use the color profiles  $\tt OpenColorIO.$  To do this, to the folder

...\Sycles\Data\ColorManagement\

you should placed the file config.ocio and all connected with it files.

			ColorManag	ement
Вид				
				Sycles → Data → ColorManagement →
Имя	*	Дата изменения	Тип	Размер
퉬 looks		11.07.2017 18:09	Папка с файлами	
퉬 luts		11.07.2017 18:09	Папка с файлами	
config.ocio		11.07.2017 16:17	Файл "ОСІО"	7 КБ

These profiles can be downloaded from the site http://opencolorio.org/. The add-on already includes profiles from the "Filmic Blender" add-on for Blender. To use them, select the value of the parameter Mode - OpenColorIO Config.

▼ Cycles Renderer Options					
Sampling Light Paths Film Color Management					
Background Performance					
Mode					
O Apply Color Profile to LDR Combined Pass					
O Mode OpenColorIO Config					
Parameters					
O Device SRGB / BT.709					
O View Filmic Log Encoding Base					
O Look Base Contrast					
O Exposure 0					
O Gamma					

The rendering result is now obtained as follows:



Another example. The same scenes with the same lighting parameters. In the first case, the color space sRBG is used:

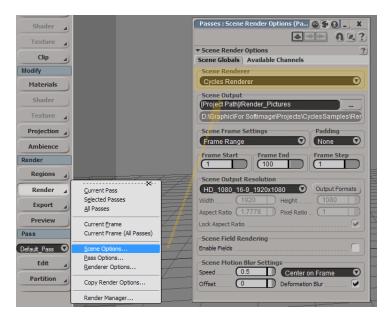


In the second case, from the OpenColorIO config:



## 19 How to use shaderballs

To make shaderballs work, you need to make sure that Softimage understands which renderer to use when rendering it. You can do this by setting the value Cycles Renderer of the parameter Scene Renderer in the window Render - Scene Options....



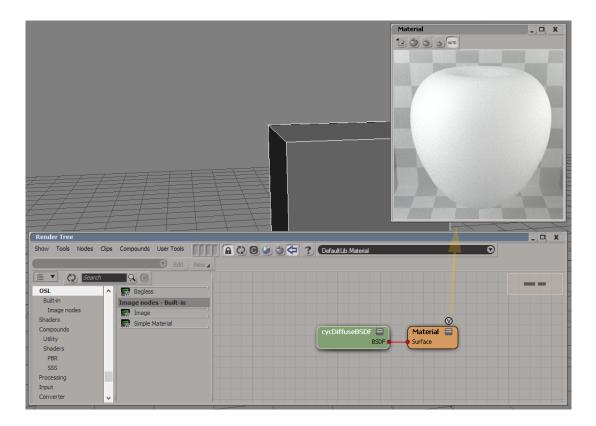
Or in the settings

#### File - Preferences - Shaderball

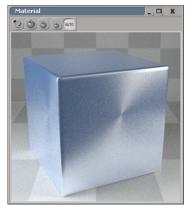
sets the value Cycles Renderer for the parameter Renderer.

Preferences		_ <b>—</b> X
Preferences (Application)     E-     Editors     Custom     Commands     Commands     Compositing     Data Management     Display     General     Display     General     Interaction     Modeling     Output Format     Rendering     Shaderball     Scene Colors     Scripting     Simulation     Units     ViewCube	Shaderball     Default Models     Spatial (Position-dependent shaders)     Model     Apple     Background     Grey_Lines     Parametric (UV-dependent shaders)     Model     Grid     Background     Checkerboard     Renderer     Quality     Parametric (UV-dependent shaders)     Model     Grid     Background     Checkerboard	

By clicking on preview icon the window with shaderball appears



Instead of the standard apple, you can use several other objects.



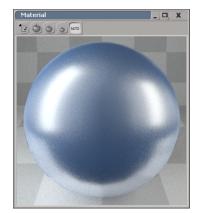
Sycles\_Cube



Sycles\_Fabric02

Also there is a new object for the background:

Sycles\_Cylinder



Sycles\_Sphere



Sycles\_Fabric01



Sycles\_Substance-like



Sycles\_Back

To add new models of shaderballs, you should copy the content of the folder

...\Sycles\Application\Shaderballs

to the folder

#### Path-to-Softimage\Application\Shaderballs

Shaderball render settings, camera position, light sources and the background object shader are hardcoded. There are three light sources: left, right and behind the camera. A checker texture is used on the background object. Some parameters can be changed by editing the file

...\Sycles\Application\Plugins\config.ini

Parameters that can be edited:

 $checker_light_color$  – the color in the interval (0,1) of the light cell of the checker texture;

**checker\_dark\_color** – the color in the interval (0,1) of the dark cell of the checker texture;

checker\_scale - number of cells in the checker texture;

render\_mode - value 1 switches the render mode to progressive;

render\_samples - the number of samples;

film\_exposure – an analog of the parameter Film – Exposure of the render settings;

max\_bounces - an analog of the parameter Light Paths - Bounces - Max of the render
settings;

diffuse\_bounces - an analog of the parameter Light Paths - Bounces - Diffuse of the render settings;

glossy\_bounces – an analog of the parameter Light Paths – Bounces – Glossy of the render settings;

transmission\_bounces - an analog of the parameter Light Paths - Bounces - Transmission of the render settings;

clamp\_direct - an analog of the parameter Sampling - Settings - Clamp Direct of the render settings; clamp\_indirect - an analog of the parameter Sampling - Settings - Clamp Indirect
of the render settings;

shading\_system - value 1 activates osl-shaders support;

left\_light\_color\_r, g, b - color components in the interval (0,1) of the left light source;

left\_light\_strength - intensity of the left light source;

left\_light\_ignore\_glossy - the value 1 makes the left light source invisible in the reflections;

right\_light\_color\_r, g, b - color components in the interval (0, 1) of the right light source;

right\_light\_strength - intensity of the right light source;

right\_light\_ignore\_glossy - the value 1 makes the right light source invisible in reflections;

center\_light\_color\_r, g, b - color components in the interval (0, 1) of a central light source;

center\_light\_strength - intensity of the central light source;

center\_light\_ignore\_glossy - the value 1 makes the central light source invisible in reflections;

#### 20 How to use OSL-shaders

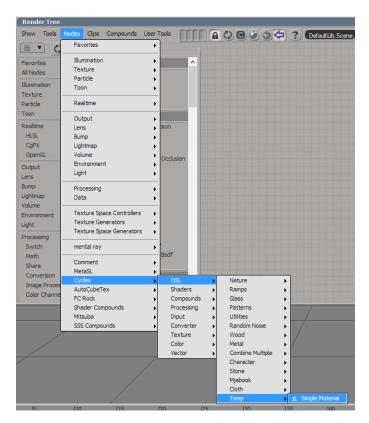
Let's create the simplest OSL-shader. For this, in any text editor write the following code:

```
//XSICategory: Temp
shader simple_material(
    color Diffuse_Color = color(0.6, 0.8, 0.6),
    float Noise_Factor = 0.5,
    output closure color BSDF = diffuse(N))
{
    color material_color = Diffuse_Color*mix(1.0, noise(P*10.0), Noise_Factor);
    BSDF = material_color * diffuse(N);
}
```

Save it as a file with any name and extension **\*.osl**. Put this file to the folder

```
\Application\osl\
```

of any workgroup. Next run Softimage and see that our shader Simple Material appeared in category



 ${\tt Cycles-OSL-Temp}.$ 

To what category the shader is placed depends on what is written in the first commented line of the shader after the words XSICategory. If nothing is written or there is no such line at all, then the shader will be placed to the category

Cycles - OSL.

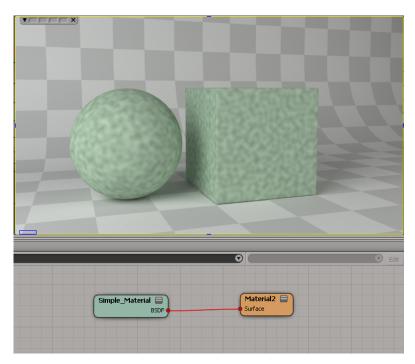
If it is written something like

#### Category/Subcategory/Subsubcategory

then the shader will be placed to the category

Cycles - OSL - Category - Subcategory - Subsubcategory.

Connect the output port of our shader Simple Material to the port Surface of the material's root node and see the result.



If you don't see result, then you should turn on OSL-shaders support in render settings. Go to the tab Performance of the render settings and set value of the parameter Shading System equal to OSL.

▼ Cycles Renderer Options
Sampling Light Paths Film Color Management
Background Performance
Tiles
Tile Size 32x32
Tile Order Center
Additional Settings
O Override Alpha in Passes
O Alpha Value 1
System
O Displacement Bump
O Progressive Rendering
O Log Render Time
O Cache Mesh
O Assign Unique Pass Id
Show Pass PASS_COMBINED
Multi-Layer EXR
O Combine Render Passes To Single EXR
O Save Separate Passes

Use the rendering mode with OSL-shaders support very carefully and only if it absolutely necessary, since it greatly increases the rendering time. Here is an example: a simple scene with a glass ball. The glass shader is the usual GlassBSDF. Without support of OSL-shaders:



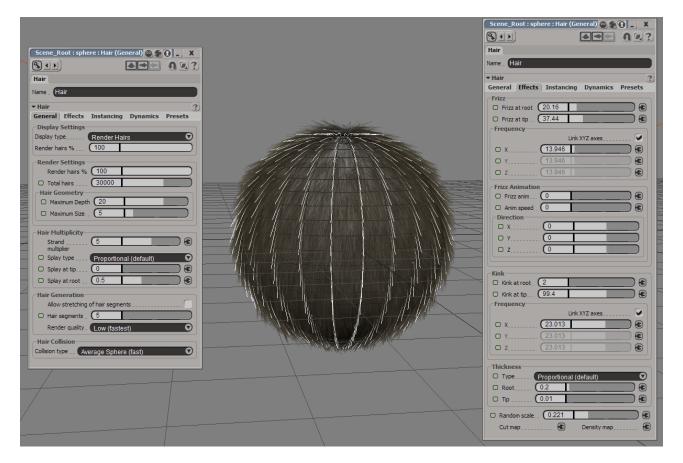
With OSL-shaders support:



The results are identical, but the rendering time differs by almost two times.

# 21 How to render standard XSI hairs

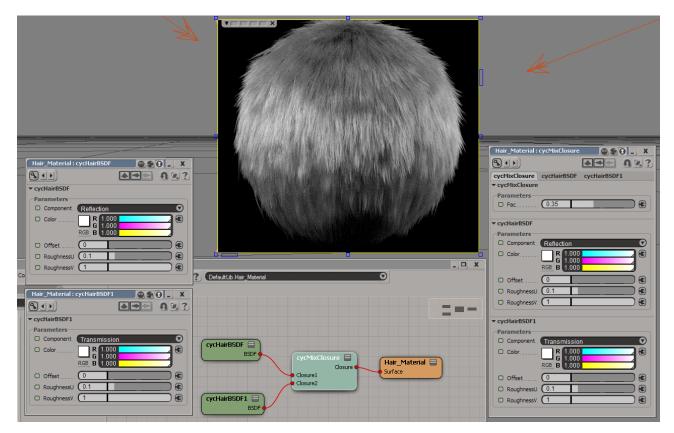
Suppose we have a scene: a ball with standard Softimage hairs.



Render it. There are two parameters for the thickness of hairs: the thickness of the roots is taken from Thickness – Root and for tips from Thickness – Tip.

ViewB : Render Region Options (Rendering)	
Render Region Options Cycles Renderer Options	
▼ Render Region Options ?	
Render Region Options	
Use Current Pass Options	
Region Renderer Selection	
Use Scene Render Options 💿 Render Selection Only	
Region Motion Blur	
Enable Motion Blur	
Set Pass Crop From Region	
▼ Cycles Renderer Options	
Sampling Light Paths Film Color Management Background	
Performance Denoising	
O Sampling Path Tracing	
Samples	
O AA Samples 256	
Settings	
O Seed	
- O Clamp Direct 0	
O Clamp Indirect 0	
C Light Sampling 0.01	
O Pattern Sobol	
Volume Sampling	
O Step Size	
O Max Steps	

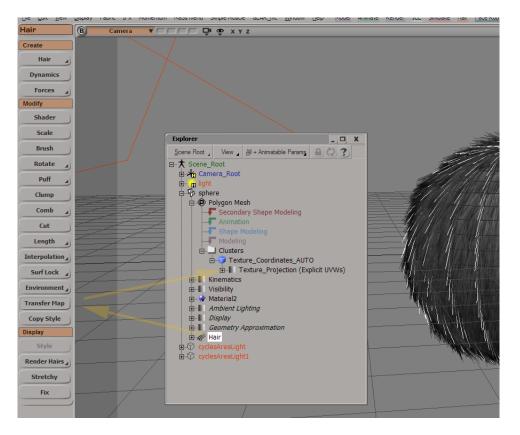
Select the hair and assign a new material to it. Mix two instances of the node HairBSDF with proportion 0.35. The first node has the mode Reflection, the second – Transmission. Render.



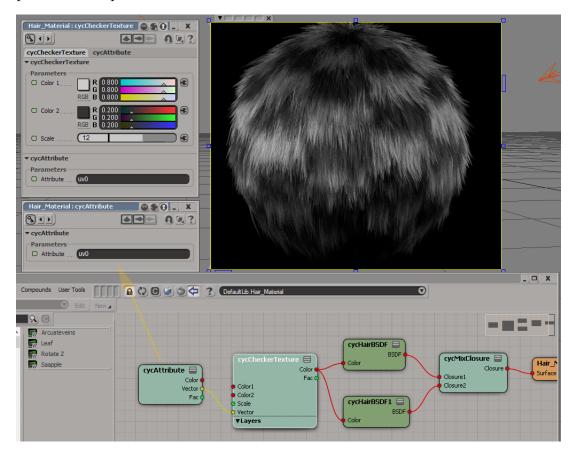
The only way to specify the color of each individual hair is to use texture coordinates. Add spherical texture coordinates to the sphere.

Primitive				
Frindive 1				
Material 🔺	, j			
Property 4	Texture Projection	Create New Projection		
	Texture Map	<u>u</u> v		
Shader 🔺		Planar <u>X</u> Y		
Texture	<u>C</u> olor At Vertices Map Render Map	Planar <u>Y</u> Z		
	<u>R</u> ender Map Symmetry Map	Planar X <u>Z</u>	$\sim$	
Clip 🔺	User Data Map	Cylindrical		
Modify	Weight Map	Sp <u>h</u> erical Spatial		
Materials		Cubic		
Traceriais	GATOR	Camera Projection		
Shader	Map Paint Tools	Unigue UVs (polymesh)		
Texture		Contour Stretch UVs		
Texture 1	Ambient Lighting	Constant in Destantion		
Projection 🖌	Glow	Connect to Projection Connect to Support		
Ambience	Lens Flare Motion Blur	Constrain Support to Bounding Box	X	
	User Motion			
Render		Unfold	I + Animatable Params 🔒 🖓 🍸	
Regions 🖌		Scene_Root		
	Annotation	🖽 🛨 Camera_Root		
Render 🔺	Attribute Display Display	ight		
Export 🖌	Polygon Cluster Visibility	sphere		
	Geometry Approximation			
Preview	Static KineState			
Pass	Synoptic			
Default_Pass	Trans <u>f</u> orm Setup			
Edit	Ultimapper			
Edit 🔺	Binormal			
Partition _	CustomColor			
	TextProp			, but the second s

Transfer it to the hairs.



During hairs rendering all texture coordinates on the object are written to vector attributes with the names uv0, uv1 etc. In the hair material add the node Attribute with value \verbuv0---- and connect the output Vector to the input Vector of the node CheckerTexture. Also connect the output Color to ports Color of nodes HairBSDF.



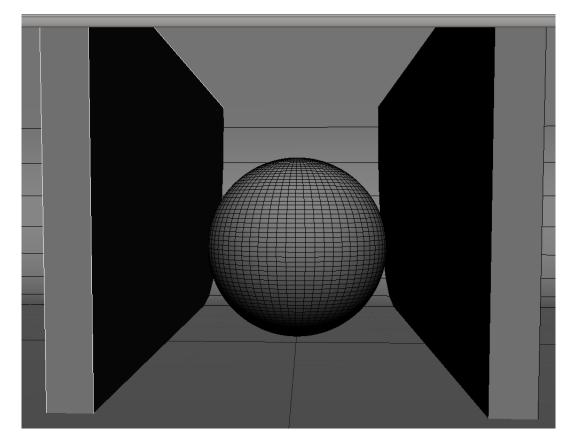
Now we can see that the hairs have a checker coloring. We can set the quality of hairs geometry in the render properties Performance – Hair Type.

ViewB : Render Region Options (Rendering)
Render Region Options Cycles Renderer Options
Render Region Options
▼ Cycles Renderer Options
Sampling Light Paths Film Color Management Background
Output Performance Denoising
Tiles
Tile Size 32x32
O Tile Order Center
Acceleration Structure
O Use Spatial Split
O Use Hair BVH
O BVH Time 0
Simplify
Culling Mode Camera or Distance
O Use Camera Cull
AO Bounces
Additional Settings
Shading SVM System
Displacement Both Displacement     Method
Hairs Type Ribbon
O Use Progressive Segments
O Log Statistics
🖸 Cache Mesh
Thread Mode Auto-detect
Devices
О СРИ
O CUDA_GEForce GTX 1060 6GB_0000:01:00

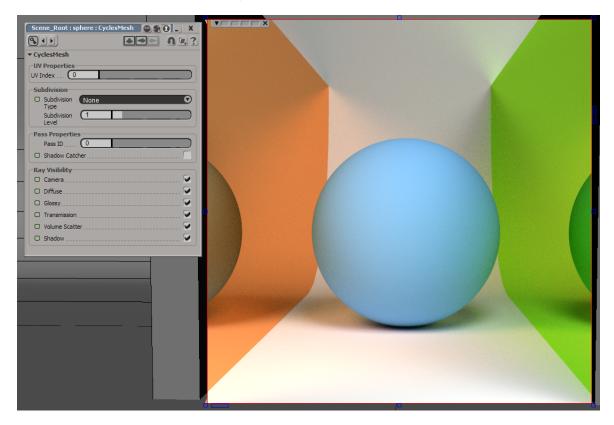
If the mode Ribbon is selected, then the hairs will be rendered as flat planes oriented towards to the camera. In the mode Segments hairs rendered as cylindrical shapes. In the mode Line Segments the hairs rendered as isolated cylindrical segments with gaps between individual segments.

## 22 How to use ray visibility

Suppose we have a scene with sphere ad boxes.



Add to the sphere the property CyclesMesh. Render.



In the case of parameter Ray Visibility – Camera is tuned off the object is not rendered directly, but it visible in reflections and it participates in the GI calculation.

Scene_Root:sphere:CyclesMesh         Scene_Root:sphere:CyclesMesh         Scene_Root:sphere:CyclesMesh         Scene_Root:sphere:CyclesMesh         VProperties         UV Properties         Subdivision         Subdivision         Subdivision         Subdivision         Pass Properties         Pass Properties         Pass Properties         Pass Properties         O         Shadow Catcher         Ray Visibility         Camera         Offuse         Glossy         Volume Scatter         Shadow	

In the case of parameter Ray Visibility – Diffuse is tuned off the object does not participate in the GI calculation.

Scene_Root:sphere:CyclesMesh         State         CyclesHesh         UV Properties         UV Index         Subdivision         Subdivision         Subdivision         Subdivision         Pass Properties         Pass D         O         Shadow Catcher         Diffuse         Glossy         Volume Scatter         Shadow	

In the case of parameter  ${\tt Ray}~{\tt Visibility-Glossy}$  is tuned off the object becomes invisible in reflections.

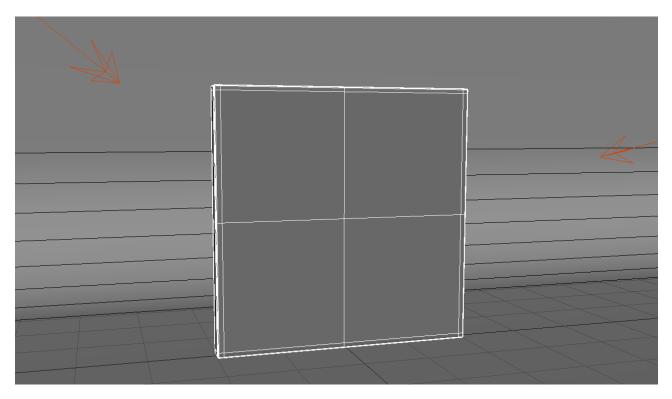
Scene_Root:sphere:CyclesMesh       Image: Scene and Scen	

In the case of parameter  ${\tt Ray}\ {\tt Visibility}-{\tt Shadow}$  is tuned off shadows for the object are not calculated.

Scene_Root : sphere : CyclesMesh         Statistical intervention         V Properties         UV Properties         UV Index         0         Subdivision         Subdivision         Subdivision         Subdivision         Subdivision         Subdivision         Subdivision         Subdivision         Pass Properties         Pass Properties         Pass Properties         Pass Properties         O         Shadow Catcher         Ray Visibility         Camera         Olffuse         Glossy         Value Scatter         Shadow	

# 23 How to use multiple uv coordinates

Suppose we have a simple scene with quad and some light sources.



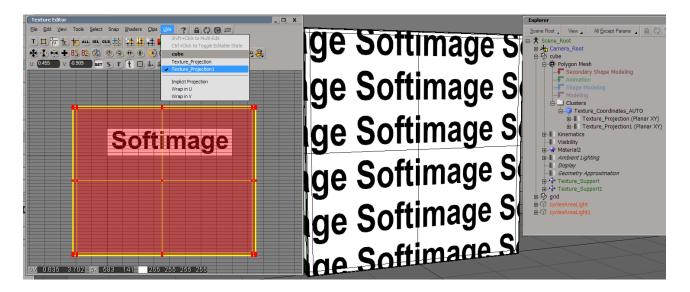
We would like to make it looks like a brick wall with some label on it. For example this label:

# Softimage

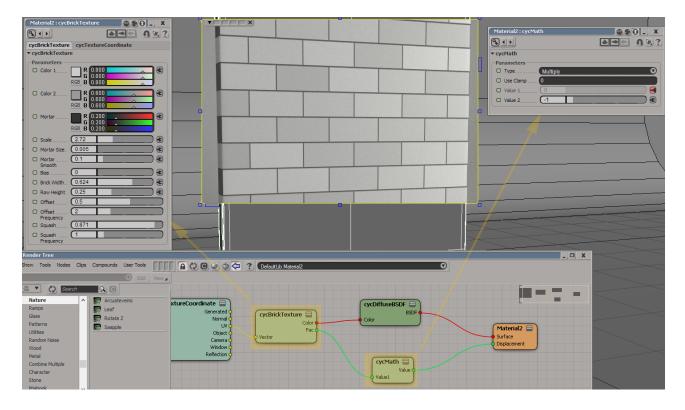
Add to out quad two texture projections to the XY-plane. The first projection leave unchanged.

exture Editor		_ <b>_</b> X		Explorer
				Scene Root View All Except Params 🔒 📿
6 <b>* * +</b> E	┝╸▲╙, ISL, CLS ‡‡ ‡‡ ‡ 85, 85, 65, 67, 67, 67, 77, 77, 77, 77, 77, 77, 77	Sinit-Statu Dradietani Ctri-Citic to Togle Editable State cube 2 V Texture_Projection Texture_Projection 1	<sup>9</sup> 1 <sup>9</sup> 2 <sup>9</sup> 3 <sup>9</sup> 4 <sup>9</sup> 5 <sup>9</sup> 6 <sup>9</sup> 7 <sup>9</sup> 8 <sup>9</sup> 9	3-大 Scene_Root 변 가 Camera_Root 마 양 cube
1		Implicit Projection Wrap in U	<sup>8</sup> 1 <sup>8</sup> 2 <sup>8</sup> 3 <sup>8</sup> 4 <sup>8</sup> 5 <sup>8</sup> 6 <sup>8</sup> 7 <sup>8</sup> 8 <sup>8</sup> 9	F Secondary Shape Modeling F Animation F Shape Modeling
	<sup>1</sup> 1 <sup>2</sup> 2 <sup>3</sup> 3 <sup>4</sup> 4	Wrap in V	7 1 7 2 7 3 7 4 5 7 6 7 7 8 7 9	Modeling     Gusters     Build Distance Coordinates_AUTO
		5 <sup>6</sup> 6 7 8 9	<sup>6</sup> 1 <sup>6</sup> 2 <sup>6</sup> 3 <sup>6</sup> 4 <sup>6</sup> 5 <sup>6</sup> 6 <sup>6</sup> 7 <sup>6</sup> 8 <sup>6</sup> 9	Texture_Projection (Planar XY)     Texture_Projection1 (Planar XY)     Texture_Projection1 (Planar XY)     Texture_Projection1 (Planar XY)
		5 *6 *7 *8 *9	<u>-51 52 53 54 5 56 57 58 59</u>	Visibility  Aterial2  Ambient Lighting
— <mark>-</mark>	1 2 3 4	5 <del>5 7 5 9</del>	4 4 4 3 4 5 4 6 4 7 4 8 4 9	Display     Geometry Approximation     End     Texture_Support
	41 42 43 44 4	5 46 47 48 9	3 3 3 3 3 4 5 3 6 3 7 3 8 3 9	
	<sup>3</sup> 1 <sup>4</sup> 2 <sup>3</sup> 3 <sup>3</sup> 4	5 *6 *7 *8 *9	$2_{1}$ $2_{2}$ $2_{3}$ $2_{4}$ $2_{5}$ $2_{6}$ $2_{7}$ $2_{8}$ $2_{9}$	
	<sup>2</sup> 1 <sup>2</sup> 2 <sup>2</sup> 3 <sup>2</sup> 4 <sup>2</sup> 4	5 6 7 8 9		
		5 6 7 8 9		
1.186 0.	.114 🕬 381 233 22	16 28 255		

Scale the second projection so that the label placed in the desired location.

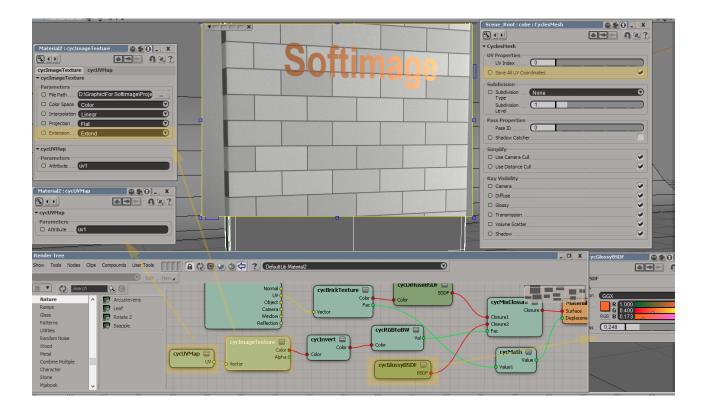


Apply to the quad the material like a brick wall. Use the node **TextureCoordinate** for setting texture coordinates. These coordinates contains data about the first (unchanged) texture projection.



Now we should say to render engine how to use the second texture projection. Add the property Cycles Mesh to the quad and turn on the parameter Save All UV Coordinates. Now with the help of the node UVMap we can select data from any texture coordinates on the object. All these coordinates have names uv0, uv1, ....

Add to the material the image with the label. In the node ImageTexture set the value of the parameter Extension is equal to Extend. This will set the texture untiled. Set as texture coordinates the result of the node UVMap with attribute value is equal to uv1. Finally, use this texture as a mask.



#### 24 What is Camera Cull and Distance Cull

In the section Performance — Symplify of the render properties we can turn on parameters Use Camera Cull and Use Distance Cull.

🛛 ViewB : Render Region Options (Rendering) 🛛 🕲 🌒 💽 💶 🗶
A
Render Region Options Cycles Renderer Options
▼ Render Region Options ? ^
Render Region Options
Use Current Pass Options
Region Renderer Selection
Use Scene Render Option 💿 Render Selection Only
Region Motion Blur
Enable Motion Blur
Set Pass Crop From Region
▼ Cycles Renderer Options
Sampling Light Paths Film Color Management
Background Output Performance Denoising
Tiles
O Tile Size 32x32
O Tile Order Center
Acceleration Structure
O Use Spatial Split
O Use Hair BVH
O BVH Time 0
Simplify
Culling Mode. Camera or Distance
🖸 Use Camera Cull 🦳 🔽 🔿 Use Distance Cull 🦳 🔽
O Camera Cull . 0.1
Margin O Distance (50
O AO Bounces
Additional Settings
O Shading SVM System
Displacement     Both Displacement     Method
O Use Progressive
O Log Render Time
O Cache Mesh
Thread Mode Auto-detect
Devices
O CUDA_GeForce GTX 1060 6GB_0000:01:00
· · · · · · · · · · · · · · · · · · ·

When the parameter Use Camera Cull is turned on, the each object on the scene is checked before being added to the rendering engine, whether it is visible from the camera or not. If the object is outside the camera's visibility cone, the value of Camera Cull for it becomes True and the object does not appear on the render. The parameter Camera Cull Margin define how far the object should be from the boundary of visibility cone in order to still used in the render. The larger this parameter, the farther the object can be from the visibility cone and at the same time take the value of the parameter Camera Cull = False.

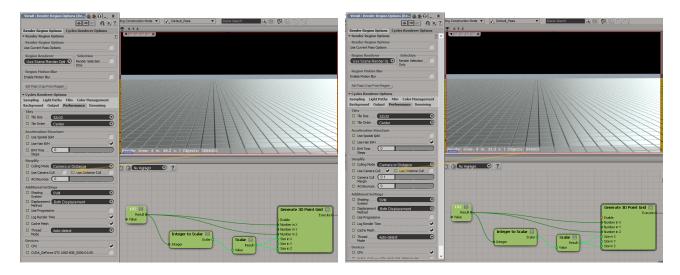
If the parameter Use Distance Cull is turned on, then the render check the distance between the object and the camera. If the distance between the object and the camera is greater than the value of the parameter Distance, then the value of the parameter Distance Cull for this object becomes True and this object excluded from the render.

The value of the parameter Culling Mode can take two values: Camera or Distance and Camera and Distance. The value of this parameter does not matter if only one method of excluding objects from the render is used. But if both methods are enabled, then for the mode Camera or Distance the object is excluded if it should be excluded by at least one of the methods, and for the mode Camera and Distance it is excluded in the case when it should be excluded by both methods.

An each object can be forcibly excluded from the culling. For this, in the properties CyclesMesh and CyclesHairs you should turn off the options Use Camera Cull and Use Distance Cull. For objects that do not have these properties, the parameters are considered to be enabled.

Scene_Root : grid : CyclesMesh 🛛 🕥 🕤 📮 🗶
▼ CyclesMesh
UV Properties
UV Index 0
O Save All UV Coordinates
Subdivision
O Subdivision None
Subdivision (1
Pass Properties
Pass ID
O Shadow Catcher
O Holdout
Simplify
🖸 Use Camera Cull
O Use Distance Cull
Ray Visibility
🖸 Camera
O Diffuse
🖸 Glossy
O Transmission
O Volume Scatter
O Shadow

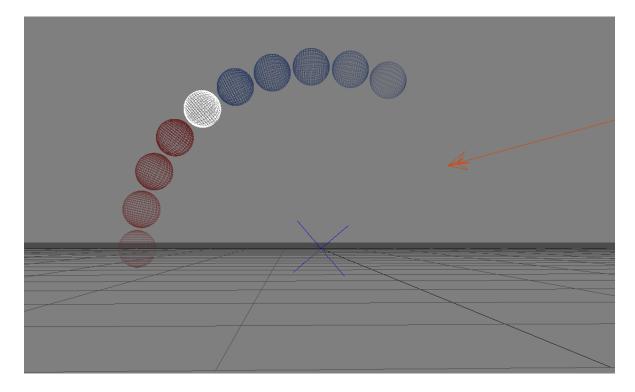
Let's consider an experiment. Generate a lot of cubes and render it with two modes: first we will render all objects, and then render with culling invisible object. Results:



In the first case the number of objects 3 686 401, render time 4 min. 48 sec. In the second case the number of objects 949 001, and render time 4 min. 33 sec. Slightly faster, but not too much. It is because with the culling we should check conditions and this increase computations before the render.

# 25 How to render motion blur

Suppose we have a scene in which the ball is spinning in a circle. Actually, this ball is a child of the null, which located in the center of the scene and this null is spinning around the z axis.



To enable effect of the motion blur we should activate the parameter Film - Motion Blur in the render properties.

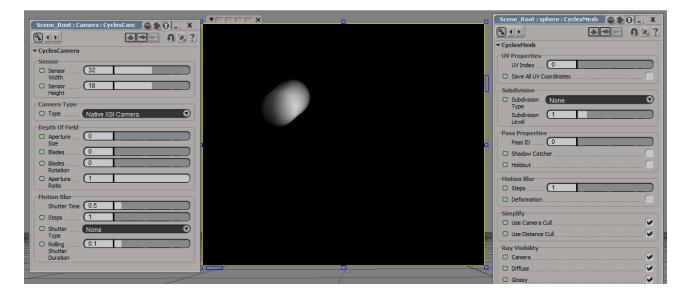
ViewB : Render Region Options (Rendering) 🛛 🕲 🕤 💶 🗶 🔤	TTTTX
A	
Render Region Options Cycles Renderer Options	
Render Region Options     ()	
▼ Cycles Renderer Options	
Sampling Light Paths Film Color Management Background	
Output Performance Denoising	
Film Parameters	
O Filter Type Gaussian O Width	
O Motion Blur	
Mist	
O Start	
O Depth	
© Falloff	
Stamp Output	
O Render Time	
O Samples	
O Date	
O Frame	
O Scene	
O Camera	
O Objects	
O Lights	

The result.

ViewB : Render Region Options (Rendering)	
Render Region Options Cycles Renderer Options	
► Render Region Options	
▼ Cycles Renderer Options	
Sampling Light Paths Film Color Management Background Output Performance Denoising	
Film Parameters	
O Exposure 1	
O Filter Type Gaussian	
O Width	
O Motion Blur	
Mist	
O Start	
O Depth	
O Falloff	
Stamp Output	
O Render Time.	
O Samples	
O Date	
O Frame	
O Scene	
C Camera	
O Objects	

For a more accurate adjustment of the effect, it is necessary to add to the camera and to the ball properties by commands

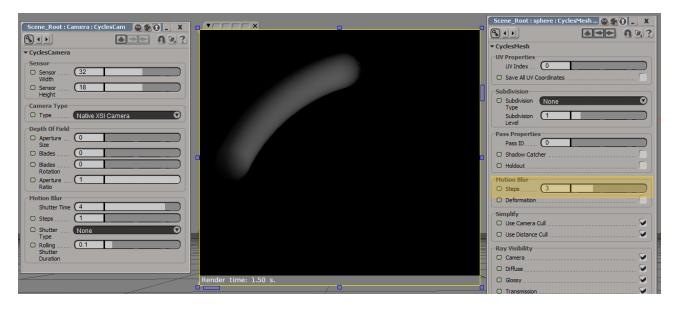
Property - Cycles Properties - Add Camera Property and Property - Cycles Properties - Add Mesh Property.



Change value Shutter Time on the camera to something greater (for example 4). But we see that the trajectory of the blur is linear.

Scene_Root : Camera : CyclesCam	Scene_Root : sphere : CyclesMesh 🕥 🗑 🕢 💶 🗙
▼ CyclesCamera	▼ CyclesMesh
Sensor	UV Properties
Sensor	UV Index 0 O Save All UV Coordinates
Width	
Height	Subdivision O Subdivision None
Camera Type	O Subdivision . (None 🗸
C Type Native XSI Camera	Subdivision 1
Depth Of Field	Pass Properties
O Aperture (0)	Pass ID. 0
O Blades	O Shadow Catcher
O Blades	O Holdout
O Aperture (1	Motion Blur
Ratio	O Steps 1
Shutter Time (4	O Deformation
	Simplify
	🖸 Use Camera Cull
O Shutter None 🔽	🔘 Use Distance Cull
O Rolling 0.1	Ray Visibility
Duration	O Camera
	O Diffuse

To fix it we should change the parameter **Steps** on the ball to some greater number (for example 3). Now all looks correct.



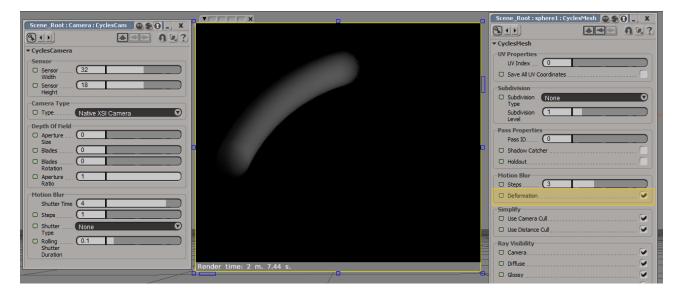
Next consider the same scene, but only difference that the ball is skinned to rotated null.

Scene_Root : Camera : CyclesCam @ @ @ _ X @ 1 ) @ @ @ @ @ ? • CyclesCamera Sensor • Sensor	
Serior UZ Widt Sensor 18 Height	
TypeNative XSI Camera      Depth Of Field      Aperture	
Blades         0           Blades         0           Rotation         0           Aperture         1	
Ratio Motion Blur Shutter Time (4 O Steps	
Shutter None Type Roling 0.1 Duration	

Render and see that there are no motion blur effect.

Scene_Root : Camera : CyclesCam 🚳 🌒 🕒 🗶			
▼ CyclesCamera			
Sensor			
O Sensor 32			-
O Sensor (18 Height			
Camera Type			
Type Native XSI Camera			
Depth Of Field			
O Aperture 0			
Size			
O Blades			
O Blades 0			
O Aperture			
Ratio			
-Motion Blur			
Shutter Time 4			
□ Steps			
Shutter None			
Туре	-		
O Rolling 0.1			
Duration			
		<u>, 0</u>	3.

It's because the object is not moved in the space, but it's vertices are moved by skinning deformation. To fix it add to the ball the property Cycles Mesh and turn on the parameter Motion Blur - Deformation.



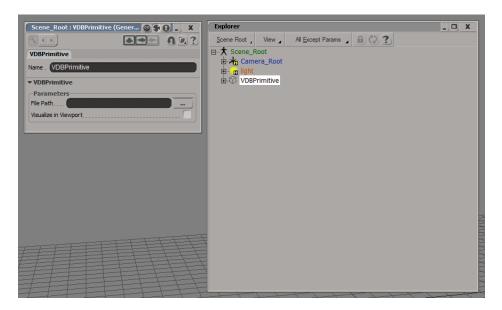
By default this parameter is turned off, because is some cases it increase the render time. Like in our case: rendering without deformation blur is spend one and a half seconds, but with deformation blur is nearly two minutes.

## 26 How to use VDB Primitive

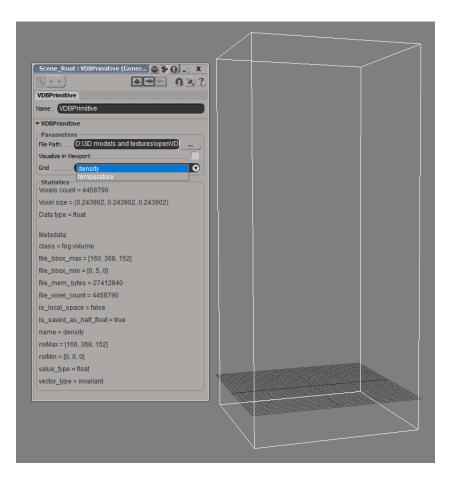
The render Sycles allows to use a new type of primitive — VDBPrimitive. This primitive allows to import static \*.vdb files to the scene. To add such a primitive to the scene — choose Get — Primitive — VDBPrimitive.

Get	
Primitive	⊿ <u>N</u> ull
Material	Curve Polygon Mesh
Property	Surface
Shader	Implicit
Texture	Control Object
Clip	Stand-in Geometry Shader Primitive
Modify	Ann <u>o</u> tation
Materials	Camera +
Shader	Light •
Texture	Model →
Projection	v kCS_IceRuler v CityBuilder
Ambience	👿 VDB Primitive
Render	<u>MaiveRT Primitives</u> Character
Regions	

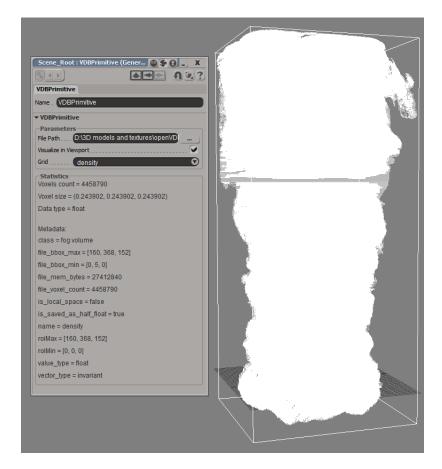
An empty object will be added. We can specified only one parameter for the object — the path to \*.vdb file.



Set in the parameter File Path full path to any file with \*.vdb extension. After that, the object will be updated. The bounding box container in the viewport and some statistical information about the selected file in the property page will appear. The most important thing for us in the property page are names and types of grids in the selected file. In out case the file contains two grids — density grid with float values and temperature grid with also float values.



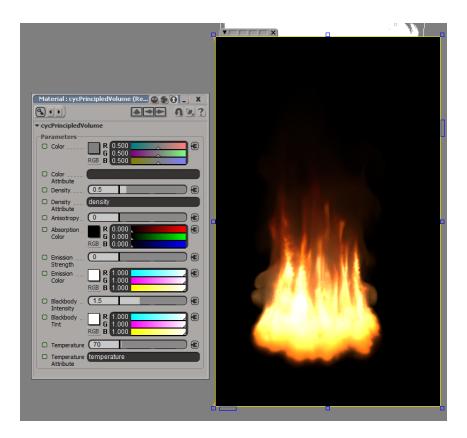
We can display active points of the selected grid in the viewport. To do this, enable the **Visualize in Viewport** parameter. It is disabled by default, since for large files this display slows down the rendering of the viewport.



Next, assign the material with the shader Principled Volume. Set

```
Density = 0.5, Blackbody Intensity = 1.5 and Temperature = 70.
```

Render.



Notice that the shader for rendering needs to specify two attributes — for density and temperature. The names of these attributes must match the names of the grids contained in the file. In our case they are called in standard way — density and temperature respectively.

# 27 How to render OpenVDB

Suppose we have a scene with \*.vdb file loaded by

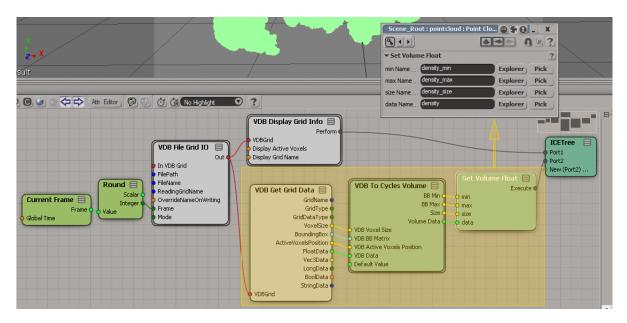
ttr. Editor Current Frame Global Time Global Time Current Frame Current Frame Curr	7 VDB File Grid ID Out In VDB Grid FilePath FileVame ReadingGridName OverrideNameOnWriting Frame Mode	VDB Display Grid Info Perform Perform VDBGrid Display Active Voxels Display Grid Name	ICETree Port1 Port2 New (Port2)	

OpenVDB for Softimage.

In our case this file contains only one grid with the name density. To transfer the data about this grid to the render, we should create four attributes:

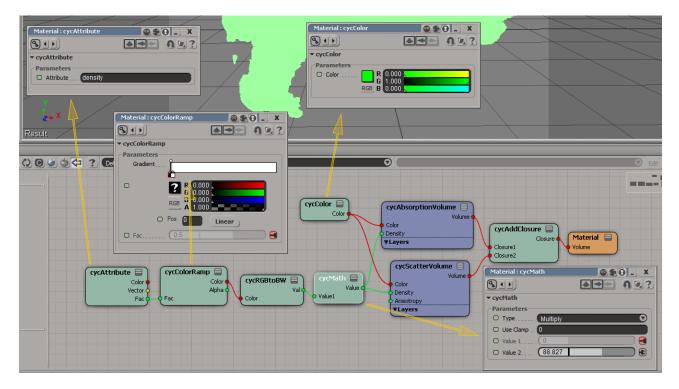
- 1. density with array of float-values, which define density of voxels;
- 2. density\_size with one vector, which contains three integers. These integers define the number of voxels in each dimension. ;
- 3. density\_min with coordinates of the minimum vertex of the voxel cube. All three coordinates of this vertex are minimal among all cube's vertices;
- 4. density\_max with coordinates of the maximal vertex of the voxel cube.

The name of the attribute with data not necessary coincide with the name of the grid, but other three attributes should start with the same name and ended by \_size, \_min and \_max.

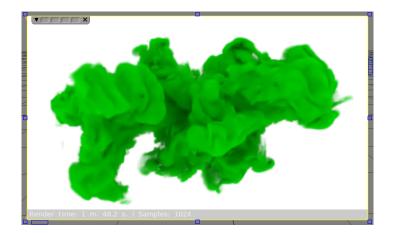


Get data from the grid by th node VDB Get Grid Data. Output port FloatData o this node contains all needed data, but it arranged in wrong order and there are some gaps in it for voxels with zero density. To restore the complete data we should use the node VDB To Cycles Volume. Next we create all desired attributes by the node Set Volume Float.

Assign to the object material like in the picture. The node Attribute is the most important because it allows the render to use stored volume data.

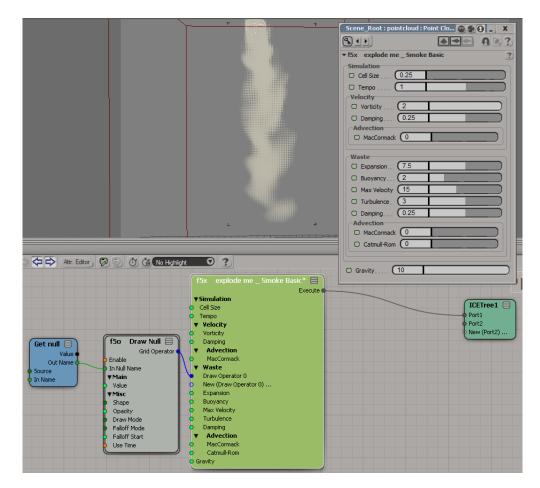


Render.



# 28 How to render emFluid

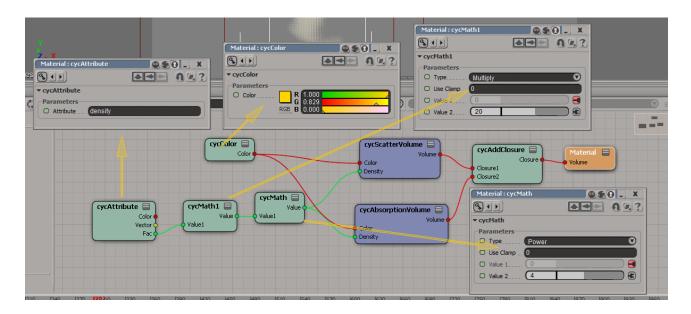
Suppose we have a scene with a smoke generated by emFluid.



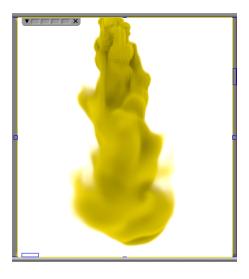
We can get the data by using the node Post Get Grid Data. As identifier of the grid we should use Waste, because in our case all simulated data stored in this grid. Record four attributes like in the case of rendering of the OpenVDB.

Get null Gid Operator of Source Enable In Null Name		ICETree1
▼ In Name	Integer Size To V	- Porcz
Value Value	X Size	Vector Size
Shape	Y Size	Set Volume Float
Opacity	Z Size	Execute
Draw Mode Falloff Mode	f5t Post Get Grid Data 😑	p min
Falloff Start	Valid 🖕	size
Use Time	Cell Size	data
	Num Cells X	
Scene_Root : pointcloud : Point Clo 🕲 🕤 💽 🔔 🗶	Num Cells Z	Scene_Root : pointcloud : Point Clo 🎯 🌎 🚷 🔔 🗶
	Lower Corner Cupper Corner	
▼ f5t Post Get Grid Data 2	Centers of Cells	▼ Set Volume Float
O Enable	P Enable	min Name density_min Explorer Pick
O FBox Index.	P In Name	max Name density_max Explorer Pick
O Grid ID 005_wa (Waste)	FBox Index Grid ID	size Name density_size Explorer Pick
		data Namedensity Explorer Pick

Assign the material to the smoke.

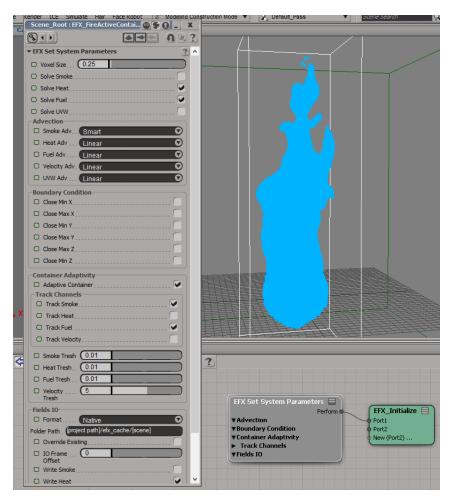


Render.

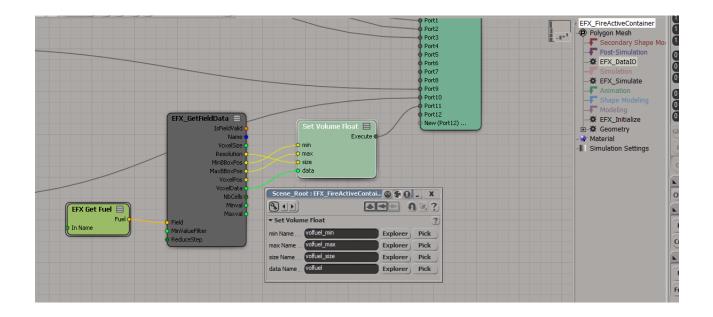


### 29 How to render Explosia FX

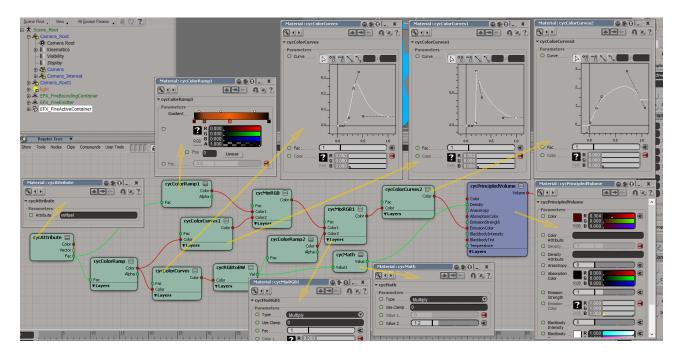
We will use the standard scene with fire.



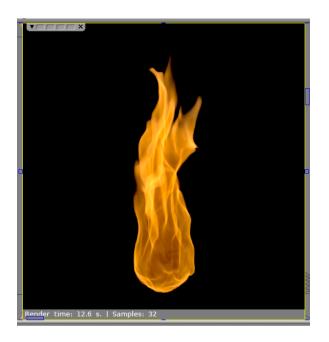
For the fire all desired data contains in the fuel attribute. We should get data from this attribute in Post-Simulation section of the ICE-stack. Use the node EFX\_GetFieldData and connect it to the node Set Volume Float with the attribute volfuel (because the attribute fuel is busy by Explosia).



Use material like in the picture bellow. All nodes except illustrated has default values of parameters.

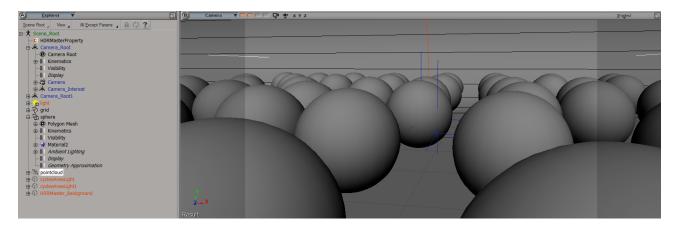


Render.

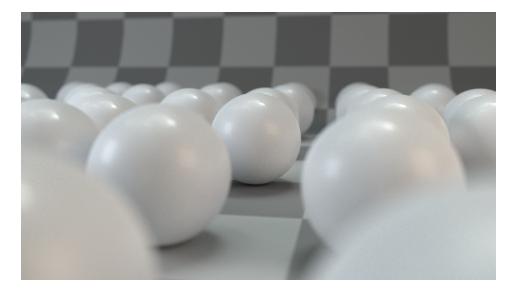


# **30** How to render and use Cryptomatte passes

Suppose we have a simple scene with spheres.



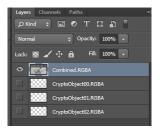
#### Render result:



To saving Cryptomatte passes, go to the tab Output of the render settings and turn on the parameter Cryptomatte – Object. This means that during rendering, information about where each object is located will be saved.

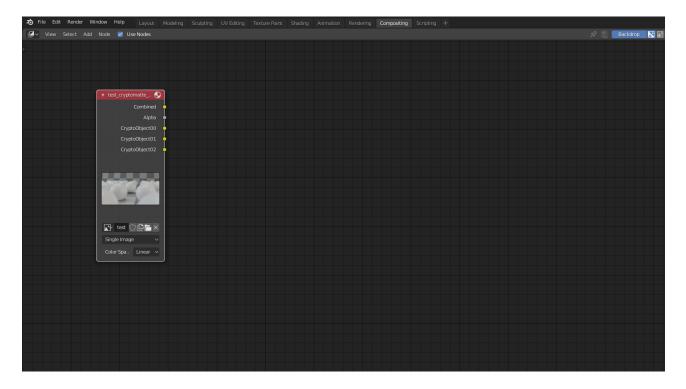
Render Manager [add-on]		_ <b>_</b> X
	Output	Renderer
Render _ Edit _ Ref	▼ Default_Pass ?	▼ Cycles Renderer Options
🕀 🕁 Default_Pass	Output Movie Archive Slate Pass Shaders	Sampling Light Paths Film Color Management
R Cycles Renderer Option	Pass Name	Background Output Performance Denoising
R emFluid5 Renderer Opt	Default_Pass	Passes
R Hardware Renderer	Pass Renderer	O Fill Pass Alpha
R mental ray	Use Scene Render Options	O Assign Unique Object Pass Id
		Multi-Layer EXR
Scene Render Options (Pa:	Render Channels Output Filename Format Channe	O Combine Render Passes To Single EXR
RB Cycles Renderer Options (F	[ [Scene]_[Pass]_[Framebuffer] png Main	O Save Separate Passes
<ul> <li>B emFluid5 Renderer Option:</li> <li>B Hardware Renderer (Passe)</li> </ul>		
	Show Resolver Add Edit Remove	Cryptomatte
ing mental ray (rasses)	Pass Frames Skip Frames	O Object
	Use Scene Render Option 👽 Skip	O Material
	Rendered	O Asset
	Start End Step	O Levels 6

After rendering, an additional file with the extension \* .exr will be created. The name of this file ends with the word Cryptomatte and it contains 4 layers: an image with the final render and three layers with information about the scene objects.

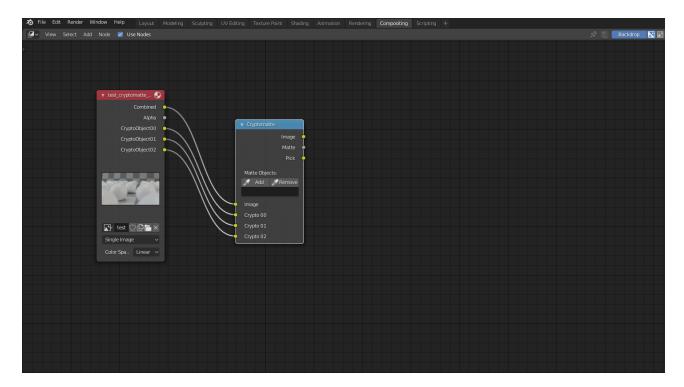


Now, as an example, let's try to use the rendered Crytomatte passes to change the color of one sphere. We will use Blender, although you can use any other composer.

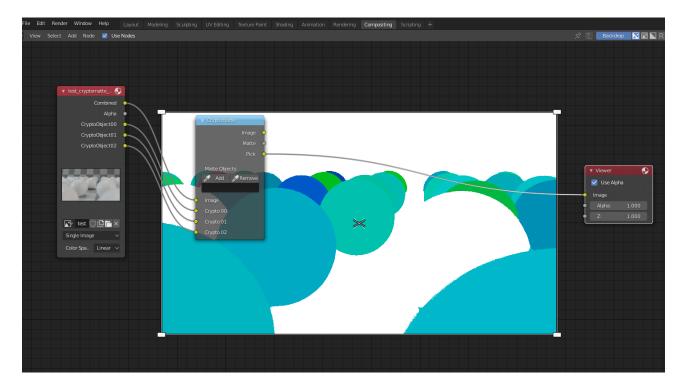
Add the Input – Image node to the Blender working canvas and choose our exr-file:



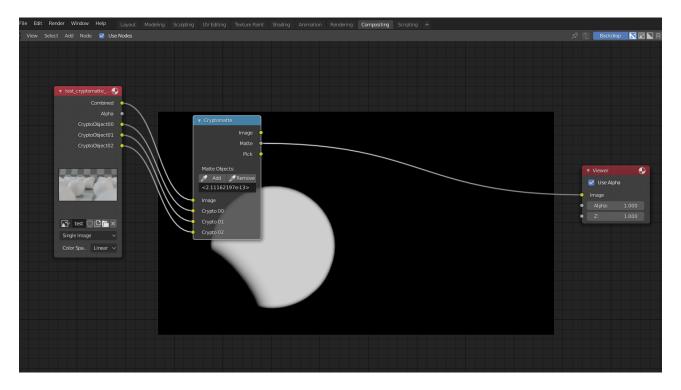
Add the node Matte - Cryptomatte and connect ports:



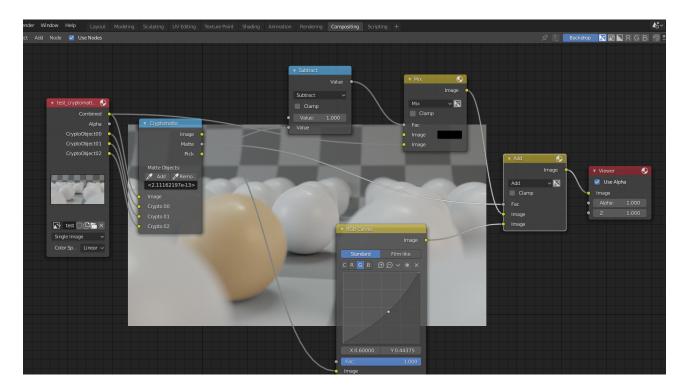
To see the information inside the file, add the node Output - Viewer and connect the Pick port to it:



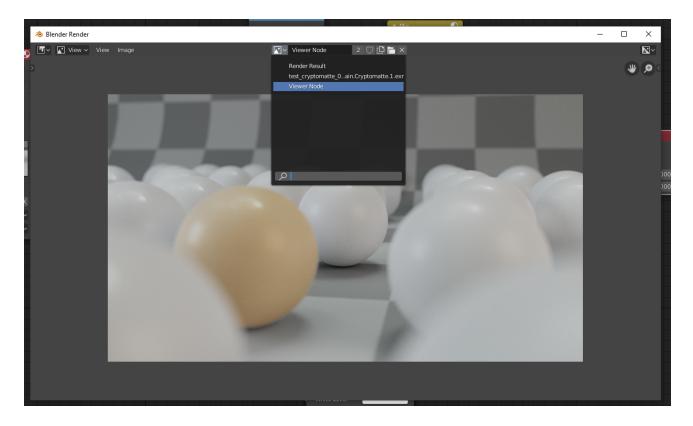
Now pick any sphere and look at the generated mask:



Add some compositing magic:



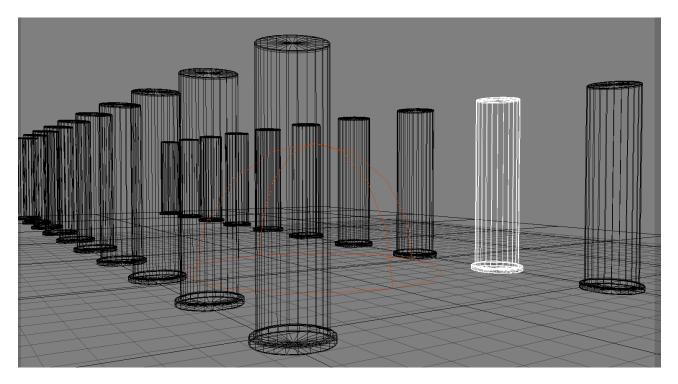
To save the composition result, open the window for the rendered results by pressing F11 and selecting in this window ViewerNode:



Press  $\tt Image-Save As...$  and save the result.

## 31 How to render AOVs

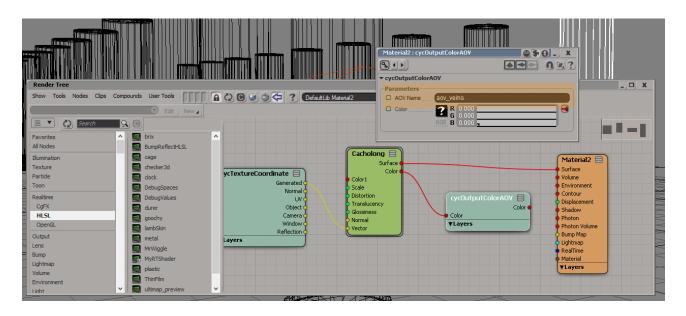
AOV (Arbitrary Output Variables) is a user pass, which we can use for store any data from shaders during rendering. Consider an example, where we have a scene – columns on a plane.



Material of the columns looks like a stone

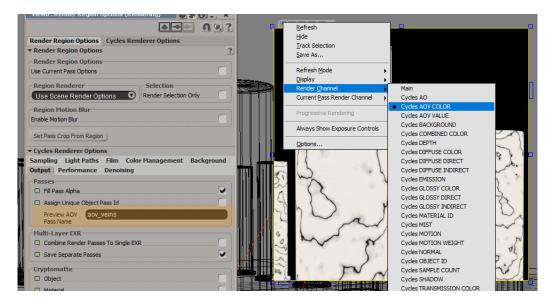


We would like to save the black veins from the texture into a separate pass. To do this, add the node cycOutputColorAOV to the material of columns and connect to its input port Color port of the stones shader node, which contains a mask of these veins. Of course, what connect to the input of the node cycOutputColorAOV we should prepare before. Call the pass by aov\_veins.



Connect the output of the node cycOutputColorAOV somewhere longer is not necessary. But you can also connect to where the output data port should be connected. In this case, the node will be used as a simple passthrough node.

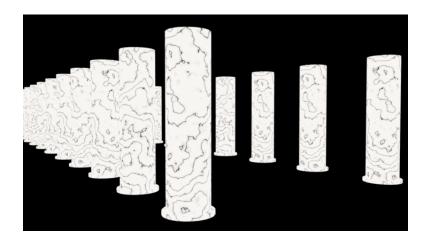
We can view the content of AOV passes in the quick render window. To do this, select Cycles AOV COLOR in the Render Channel of the preview window. In general, we can store information in many different passes, so the render does not know in which one to show. It is necessary to set the name of the AOV pass to the parameter Preview AOV Pass Name in the Output tab of the render preferences.



The only way to save all AOV passes as a picture is to use the function for save all output passes as single multilayer exr file. To do this, turn on the parameter Combine Render Passes To Single in the Output tab of render settings.

Render Manager [add-on]		X
	Output	Renderer
Render _ Edit _ Ref	▼ Default_Pass ?	▼ Cycles Renderer Options
Default Pass	Output Movie Archive Slate Pass Shaders	Sampling Light Paths Film
R Cycles Renderer Option	Pass Name	Color Management Background Output
R emFluid5 Renderer Opt	Default_Pass	Performance Denoising
R Hardware Renderer	Pass Benderer	Passes
R mental ray	Use Scene Render Options	🔘 🖸 Fill Pass Alpha
🗄 🛄 Framebuffers		O Assign Unique Object Pass Id
	Render Channels Output	Preview
R <sub>B</sub> Cycles Renderer Options (F	Filename Format Channe [[Scene]_[Pass]_[Framebuffer] png Main	AOV Pass
R <sub>B</sub> emFluid5 Renderer Option:		Name
R <sub>B</sub> Hardware Renderer (Passe	Show Resolver Add   Edit   Remove	Multi-Layer EXR
R <sub>B</sub> mental ray (Passes)	Pass Frames Skip Frames	Combine Render Passes To Single EXR
	Use Scene Render Option Skip	O Save Separate Passes
	Rendered	Cryptomatte
	StartEndStep	O Object
		O Material
		O Asset
	Pass Camera	
	Camera 🕥 Inspect Camera	O Levels (6
	Pass Output Resolution	O Accurate Mode
	Override Scene Render Options	
	Custom format Output Formats	
	Update From Camera	
	Pass Motion Blur	

#### The result:



All the same works for the case when we want to save in the pass not a color, but a numerical value. The only difference is that we need to use the cycOutputValueAOV node.