CentiLeo for Cinema 4D
User Manual Version 0.44 alpha

Last update: 06 November 2016

www.centileo.com

Contents

1. CentiLeo Renderer for Cinema 4D: cntlc4d 4
2. System requirements 5
3. Features of CentiLeo for Cinema 4D plugin 5
    Geometry 5
    Light transport (GI) solver 5
    Textures 5
    Materials and Shaders 5
    Light sources 6
    AOVs for compositing 6
    Camera 6
    Interactive Preview Render 6
    Limitations 6
    Special note on GPU memory consumption 7
    Features in TODO list 7
5. Getting started 8
    Download link and Installation 8
    Starting CentiLeo Renderer in Cinema 4D 8
6. CentiLeo Renderer settings 9
    Tab Render Parameters (affect rendering performance) 9
        Image resolution 9
        Production render iterations 9
        Max sample intensity (direct light) 9
        Max sample intensity (specular light) 9
        Max sample intensity (indirect) 9
        Firefly killer 10
        Render iteration compute power 10
        Max ray bounces 10
        Pixel filter size [pixels] 10
        Default environment color 10
    Tab IPR 11
    Tab AOVs 11
    Tab Camera 12
    Tab Post-render Parameters 13
    Tab Displacement Mapping 14
    Tab Hardware Settings 15
    Tab Scene Converter 15

7. Textures Cache System ... 16
8. Materials and shaders ... 17
  CentiLeo Standard Material (cntlStdMat) ... 17
    Layer weights ... 17
    Diffuse layer ... 17
    Reflection1 layer ... 18
    Reflection2 layer ... 19
    Sub-surface scatter (SSS) layer ... 20
    Transmission layer ... 21
  CentiLeo Multi Material (cntlMultiMat) ... 22
    Multiple coating layers ... 22
    Geometry Masking ... 23
  Displacement map ... 23
  CentiLeo Textures (shaders) ... 24
    cntlTexture (image file) ... 25
    cntlConst (constant value) ... 26
    cntlNoise (procedural) ... 27
    cntlBlend (mix of textures) ... 28
    cntlFalloff (angular mix of textures) ... 28
    cntlColorCorrection ... 29
    cntlTriplanar (UV projection) ... 30
    cntlBinary (math ops on textures) ... 30
9. Light sources ... 31
  Tag cntlTagLight for light sources (and Portal accelerator) ... 31
  Tag cntlTagEnvMap for Environment Mapping ... 34
10. Interactive Preview Rendering (IPR) ... 35
11. User Manual changelog ... 37
  cntlc4d 0.44 alpha ... 37

1. CentiLeo Renderer for Cinema 4D: cntlc4d

CentiLeo renderer for Cinema 4D (in short **cntlc4d**) is the integrated rendering plugin with GPU acceleration which is currently in Alpha stage.

This User Manual document is an ongoing work and will be extended with more materials, pictures and other examples describing the workflow of CentiLeo renderer.

The news on CentiLeo project is available on www.centileo.com and downloads regarding Cinema 4D plugin are available on this forum http://centileo.com/forum/forum18/.

The purpose of CentiLeo is to combine many features and make the use of them as easy to use as possible and provide the artists with a high degree of freedom, allow making simple or very complex and detailed 3D projects with easier workflow and the highest processing speed. It is still a work in progress expecting a lot of improvements.

Shibaya scene (on the background) was provided by www.nonecg.com

In CentiLeo Renderer our team has designed:

1) Scalable and most efficient support for various scene sizes (from small to very large scenes containing hundreds million unique polygons or many dozen or hundred GB of textures) on GPUs wherein the scene size can potentially exceed the GPU memory capacity.
2) Powerful light transport system which can render exteriors and interiors with easy to setup settings wherein many complex things are hidden under the hood of the renderer. CentiLeo light transport system is already quite fast and adapts more compute power based on the noise distribution across the image.
3) Powerful Material, Shading and Lighting infrastructure. It already supports complex Standard CentiLeo material with 2 reflection anisotropic lobes, diffuse, 3 SSS layers

and transmission. Several materials can be united in one layered multi-material and combined with geometry mask and displacement mapping.

4) Fluent communication between interactive rendering mode and Cinema 4D modeling workflow via CentiLeo interactive preview render (IPR) which reacts very quickly on events of mesh change, instances, shading and lighting change.

It should be noted that by initial Alpha version the basic infrastructure of the renderer was created which is highly scalable and flexible and allows integrating many more features, interesting shaders and more compositing tools. There is huge potential to accelerate CentiLeo light transport for highly challenging lighting situations even further.

2. System requirements

- OS Windows 7 or higher
- NVIDIA GPUs with at least 2 GB of VRAM and compute capability of 2 or higher.
- GPUs with at least 3 GBs are recommended.
- Latest GPU drivers are recommended.
- Main RAM with at least 4 GB (8 or 16GB are recommended).
- Cinema 4D R16-R18

3. Features of CentiLeo for Cinema 4D plugin

Geometry

1) Out-of-core geometry meshes (even hundreds millions polygons) with GPU acceleration and without GPU memory limits.
2) Adaptive on-the-fly displacement mapping with pixel precision.
3) Fast support of dynamic meshes at scene load time and during modeling with opened CentiLeo IPR window.
4) Native Cinema 4D instancing accounting on-the-fly.

Light transport (GI) solver

1) Noise-driven path tracer which accelerates rendering of noisier image parts when other parts are already clean.

Textures

1) Out-of-core and out-of-CPU RAM textures with GPU acceleration. This means that sometimes the textures may not fit even CPU memory and this case is handled.
2) Up to 16K x 16K texture image files.
3) Up to 100K textures in the scene.

Materials and Shaders

1) CentiLeo Standard material (**cntlStdMat**) with 2 reflection lobes (also anisotropic), 3 SSS layers, diffuse layer and transmission layer, 2 bump map slots (global and reflection only). 30 properties overall, shaders can be applied to everything.
2) CentiLeo Multi Material (**cntlMultiMat**) which can combine up to 10 cntlStdMats.
3) Multi Material has displacement mapping and geometry mask slots.
4) Currently only own CentiLeo shaders are supported:

a. cntlTexture (Bitmap, file with image)
b. cntlNoise (procedural noise)
c. cntlHDRI (used for environmental maps)
d. cntlConst (for arbitrary values)
e. cntlTriplanar (implements triplanar UV projection)
f. cntlFalloff (mixes two shaders based on IOR falloff)
g. cntlColorCorrection (edits the color output of another shader)
h. cntlBlend (mixes two shaders with a shadered blend)
i. cntlBinary (performs binary basic math operation on 2 shaders like add, mul, sub, div, pow, min, max)
j. Substantially more shaders will be added in this list with special priority to Dirt, Carpaint, Gradient, Multi-Texture and more Noise types and 3rd party shaders.

Light sources

1) Basic Area, Directional, Spot and Point (spherical) light types are supported as objects.
2) Options of lights visibility for certain material layers.
3) Option to select the light pass id for certain light source.
4) Environment mapping can be enabled in CentiLeo Renderer settings for different material layers.

AOVs for compositing

1) Support for rendering up to 8 light passes which are rendered in parallel with Beauty image. These can be saved as separate image layers for further manipulation in compositing tools.

Camera

1) Only perspective camera view is currently supported
2) Depth of Field controls with sensor width, focus distance, number of blades and blade angle.
3) Motion blur is yet to be done (after Hair and Fur).

Interactive Preview Render

1) Same scene rendering as production mode.
2) Interactive reaction to camera motion
3) Interactive reaction to geometry mesh edits
4) Interactive reaction to object motion/rotation and etc.
5) Correct recognition of Cinema 4D hierarchical geometry instances and objects on the fly while modeling
6) Interactive reaction to lights/environment/materials/shaders edits.
7) Switch on/off reaction to mesh edits
8) Switch on/off per frame retesselation if displacement mapping is present
9) Selector of auto-focus point on the screen

Limitations

1) Up to 30 total ray bounces and up to 10 diffuse ray bounces.
2) Up to 16K textures
3) Up to 100K textures

4) Up to 2M instances of meshes in one scene
5) NVIDIA only GPUs with compute capability 2.0 and higher and at least 2 GBs of device memory (however we recommend 3 and more GB of memory).
6) Windows only version of Cinema 4D is currently supported.

Special note on GPU memory consumption

In order to support out-of-core geometry CentiLeo renderer generates an index structure that consumes around 3% of total scene geometry size. And this should be stored in GPU memory during render time. The rest 97% of geometry data may be stored in CPU RAM if it doesn't fit to GPU memory. However during render time these portions of geometry are delivered to GPU memory on demand for efficient GPU utilization.

Also the rays that participate in light transport computation may consume GPU memory: from 512MB (for small GPUs) to 1024MB (for larger GPUs). So this memory block is excluded from capacity which can be used for geometry/texture storage.

So, basically for CentiLeo to run correctly it is necessary to have at least 896MB of free defragmented GPU memory. But in a real system with only single GPU the display/driver/viewport may consume some GPU memory and make it fragmented. That's why for robustness we enable the minimum requirement for CentiLeo running to be a GPU with 2GB+.

Features in TODO list

1) Multi-GPU support
2) Further light transport render math improvement
3) Forester support
4) Hair and Fur support
5) Full motion-blur
6) Further displacement mapping improvement
7) More AOVs (more beauty, info and mask with more customization)
8) Post production Glare, Bloom and Flare, LUTs
9) Many explicit mesh UV sets
10) More procedural UV sets
11) Dirt, special carpaint, gradient, multi-texture, more noise types and other shaders
12) More photography camera options
13) Mesh lights with emission texture
14) Include/Exclude lists for lights and objects
15) IES lights
16) Shadow catcher
17) Procedural Sky system
18) Volumetrics
19) Proxy meshes
20) Support CentiLeo textures display in viewport
21) Support most common Cinema 4D and 3rd party materials, lights and maps as direct translation and conversion to CentiLeo analogues
22) Utility light lister
23) Utility texture cache upfront creation
24) Utility of production rendering for prepared scene outside of Cinema 4D

25) And more features, connections with 3rd party plugins and popular data formats improving the renderer

5. Getting started

Download link and Installation

The latest version of CentiLeo for Cinema 4D plugin can be downloaded using this link http://centileo.com/forum/forum18/.

CentiLeo GPU renderer supports the plugins for Cinema 4D versions R16, R17 and R18 (still Windows only).

CentiLeo plugin for Cinema 4D R16 is stored in "C4D R16" folder of CentiLeo distribution package. CentiLeo plugin is stored "C4D R16/plugins " in "CentiLeo" folder and this folder must be copied to the folder with Cinema plugins, e.g. to the "C:/Program Files/MAXON/CINEMA 4D R16/plugins/".

Similar procedures install the plugin for R17 and R18.

Normal 1-click installer will be implemented slightly later, sorry for this inconvenience.

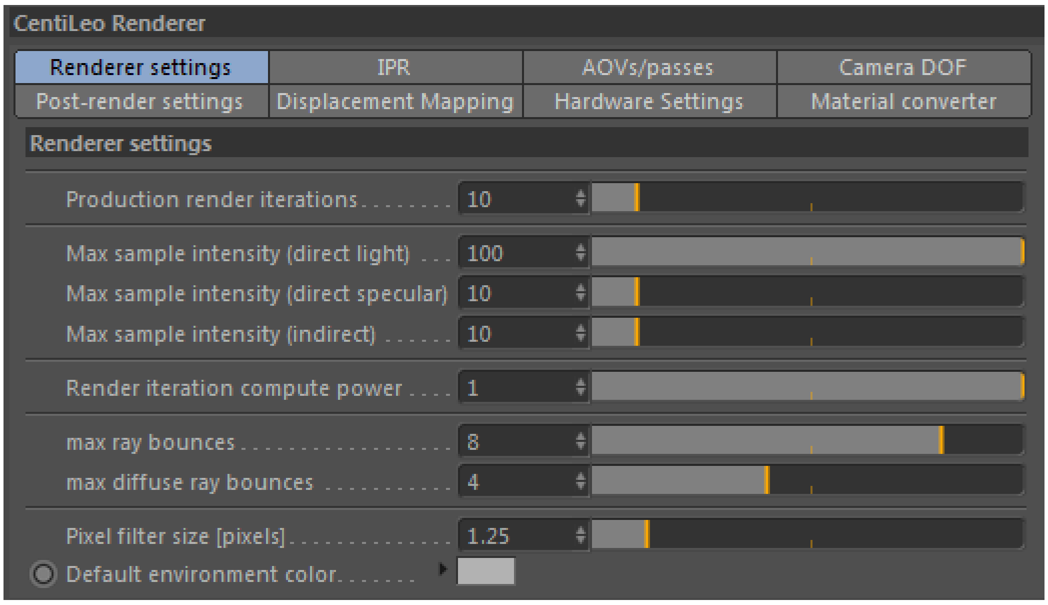
Starting CentiLeo Renderer in Cinema 4D

Once installed CentiLeo interactive renderer (IPR) can be started by finding CentiLeo menu item in main Cinema 4D menu. It can also be enabled as a renderer for Picture Viewer and Cinema Viewport selecting CentiLeo in a usual Cinema Render Settings menu.

6. CentiLeo Renderer settings

When CentiLeo plugin is selected in Cinema Render Settings there appears several menu tabs: Renderer Parameters, AOVs, Camera DOF, Post-render Parameters, Displacement Mapping, Hardware Settings, Material Converter.

Tab Render Parameters (affect rendering performance)


CentiLeo Renderer

Renderer settings | IPR | AOVs/passes | Camera DOF
Post-render settings | Displacement Mapping | Hardware Settings | Material converter

Renderer settings

```
Production render iterations . . . . . . . .        10
Max sample intensity (direct light) . . .           100
Max sample intensity (direct specular)              10
Max sample intensity (indirect) . . . . . .         10
Render iteration compute power . . . .              1
max ray bounces . . . . . . . . . . . . . . . . . . 8
max diffuse ray bounces  . . . . . . . . . . .      4
Pixel filter size [pixels] . . . . . . . . . . . . . 1.25
Default environment color. . . . . . .
```

Image resolution (range: arbitrary) it can be changed for both production render mode and CentiLeo IPR (interactive preview render) in the standard Output Tab of Cinema 4D Render Settings.

Production render iterations (range: 0..infinity) the total number of full image passes (iterations) per frame for standard Cinema 4D Production rendering mode which works for rendering to Picture Viewer, View render (Cinema's viewport) and Render region. Each one complete render iteration generates several dozen samples for each pixel.

Max sample intensity (direct light) (range: 0..infinity) maximum sample intensity for direct lighting. This parameter clamps direct lighting component for each sample at render time which may help reduce the noise of direct lighting if present.

Max sample intensity (specular light) (range: 0..infinity) maximum sample intensity for direct lighting visible through perfect specular reflections or refractions (e.g. mirror or glass). This parameter clamps specular direct lighting component for each sample at render time and helps reduce the hotspots form e.g. high frequency specular surface bumps and hence accelerates the overall noise reduction process.

Max sample intensity (indirect) (range: 0..infinity) maximum sample intensity for indirect lighting (global illumination). This parameter clamps GI component for each sample at render time and helps reduce the hotspots in the rendered image and helps accelerate the noise reduction.

Notice

It is recommended to tune max sample intensity (MSI) parameters before computing clean rendering. Too high MSI values may result in long process of cleaning the noise of the render. Too low values may reduce the overall brightness of the image.

Firefly killer (range: 0..1). After algorithmic reconsiderations in 0.43 version this parameter has no effect currently. But it may return in the future.

Render iteration compute power (range: 0..1). When this parameter is closer to 0 then small number of samples is computed for each image iteration. When this parameter is close to 1 then more samples are allocated for each image iteration (the actual number of samples is determined under the hood of the renderer engine based on the hardware).

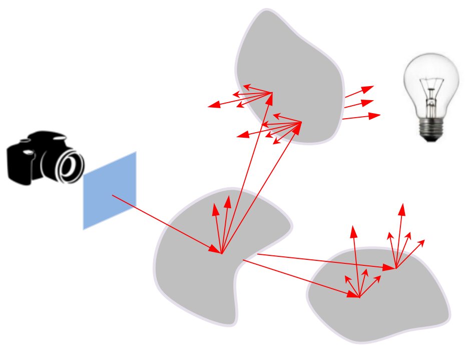
E.g, when render iter power is zero then each image iteration (pass) executes much faster but due to using fewer samples it brings less valuable information which is necessary for noise reduction in a long run. This parameter can be valuable for quicker interactive rendering mode reaction on low performance GPUs.

Notice

It is recommended to set **render iteration compute power = 1** for high performance GPUs and for production rendering mode (render to Picture Viewer). It also better saturates the GPU.

Max ray bounces (range 0..30) determines the total number of ray bounces that are allocated to find the light source contribution to the image. The rays may bounce through diffuse surfaces or specular surfaces (like perfect mirror or glass).

Max diffuse ray bounces (range 0..10) determines the maximum number of accounted ray bounces through diffuse surfaces in the process of searching the light source contribution.



Notice

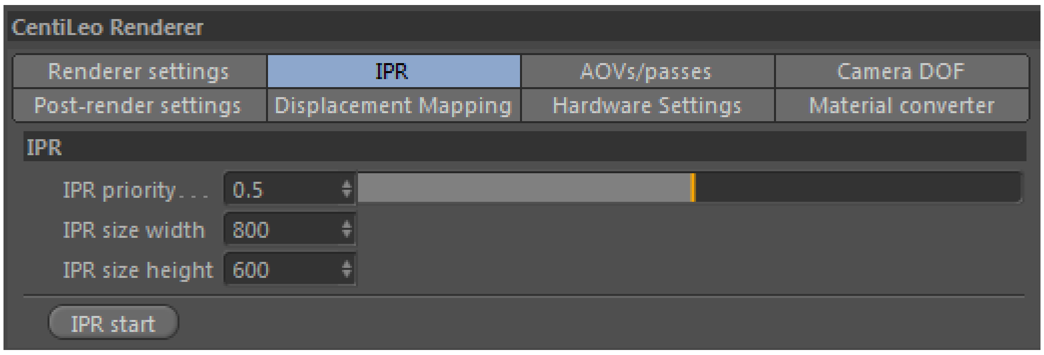
When there are a lot of mirrors or glass in the scene it is recommended to increase the Max ray bounces parameter value to catch all reflections or refractions.

Increasing the max diffuse ray bounces adds more realism but with more bounces it changes the rendered image very slightly and reduces render speed.

Pixel filter size [pixels] (range 0..2) determines the width of image gaussian blur filter in pixels. Higher value of this parameter helps reduce image aliasing at the border of very bright light highlight and non-bright image part.

Default environment color determines the render background color in case the Sky objects are not placed to the scene.

Tab IPR


CentiLeo Renderer

Renderer settings | IPR | AOVs/passes | Camera DOF
Post-render settings | Displacement Mapping | Hardware Settings | Material converter

IPR

IPR priority . . . 0.5
IPR size width 800
IPR size height 600

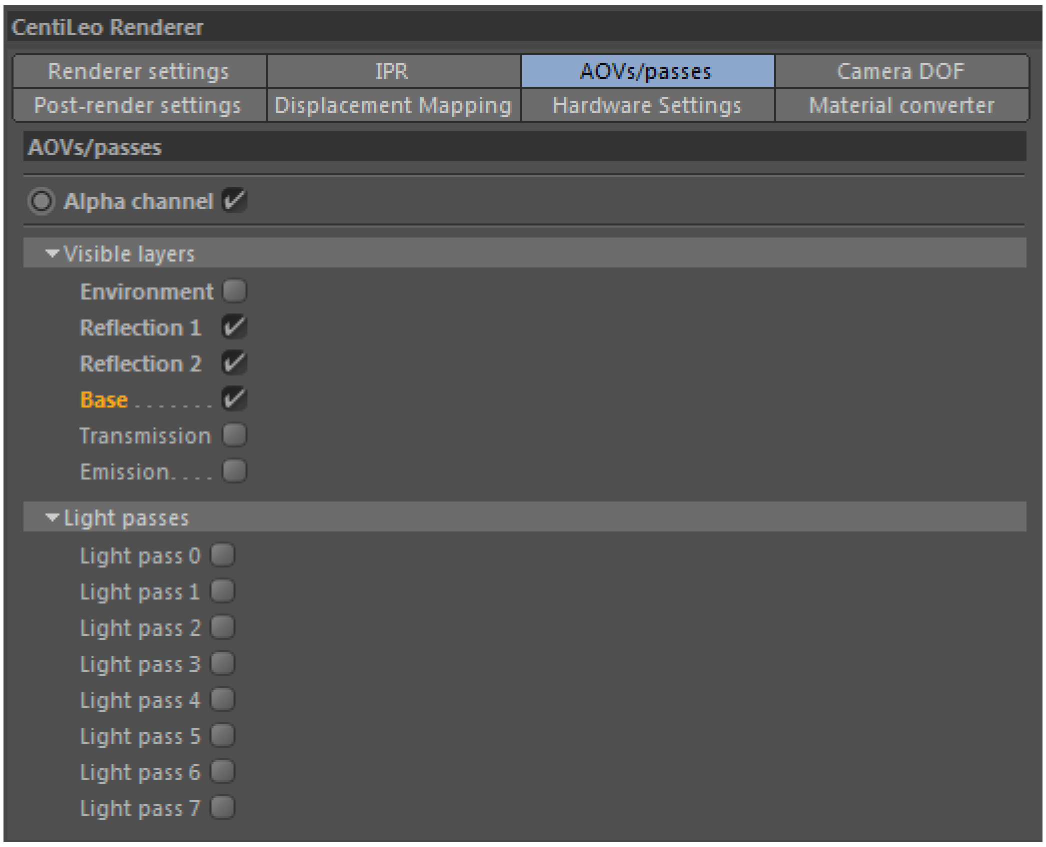
IPR start

CentiLeo supports interactive rendering mode (called IPR) which opens a window wherein the scene is rendered and automatically updated when user adds, deletes or edits lights/materials/shaders/geometry etc. This IPR window works in a separate thread to the main thread of Cinema 4D GUI.

IPR priority - larger priority increases render speed (samples/sec indicator in IPR window) but can produce Cinema 4D GUI lags. Lower priority decreases render speed by 10-20% but makes work with GUI much smoother.

IPR size width / height – determines the render image resolution in IPR mode.

Tab AOVs


CentiLeo Renderer

Renderer settings | IPR | AOVs/passes | Camera DOF
Post-render settings | Displacement Mapping | Hardware Settings | Material converter

AOVs/passes

Alpha channel ✓

Visible layers
- Environment ☐
- Reflection 1 ✓
- Reflection 2 ✓
- Base . . . . ✓
- Transmission ☐
- Emission. . . . ☐

Light passes
- Light pass 0 ☐
- Light pass 1 ☐
- Light pass 2 ☐
- Light pass 3 ☐
- Light pass 4 ☐
- Light pass 5 ☐
- Light pass 6 ☐
- Light pass 7 ☐

In this Tab the AOVs are checked that are computed in parallel with the Beauty image with almost no compute overhead but with higher memory consumption.

If it is necessary to observe any specific AOV then the corresponding checkbox must be enabled. For each computed AOV a separate image buffer of the same size as Beauty image is allocated in CPU RAM.

Currently only two separate groups of AOVs are supported: “Visible layers” and “Light passes”. For example all the elements of “Visible layers” group may be summed into main Beauty image. Alternatively all the elements of “Light passes” group may be summed into Beauty image too (but the elements of the sum are different).

For example all the lighting that contributes to the rendered image using the Base surface material layer (Diffuse and SSS material components) corresponds to “Base AOV”. All the lighting that contributes to the rendered image using Reflection 1 material layer corresponds to “Reflection 1 AOV”, etc.

The contribution of all the light sources with selected light_pass property (range of values from 0 to 7) are assigned to corresponding “Light pass AOV”.

The contribution of Environment maps are assigned to light pass 0 and the other light sources may select other light passes based on user needs within the tag of the light source.

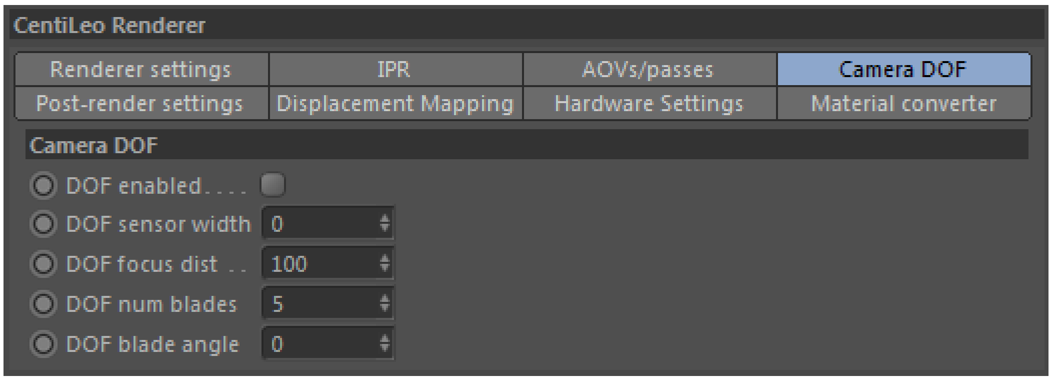
Notice

Each computed AOV can be observed in CentiLeo IPR render in real-time.

AOVs can be observed and saved to disk if rendered in Cinema Picture Viewer based on the standard settings in Render Settings in Output and Save pages.

If selected image resolution is small enough then AOVs may be stored in GPU memory which can be accelerate IPR mode.

Tab Camera



DOF enabled	☐
DOF sensor width	0
DOF focus dist . .	100
DOF num blades	5
DOF blade angle	0

In this Tab the Depth of Field camera parameters can be tuned. Currently there are no Motion Blur params which will be implemented in the near future.

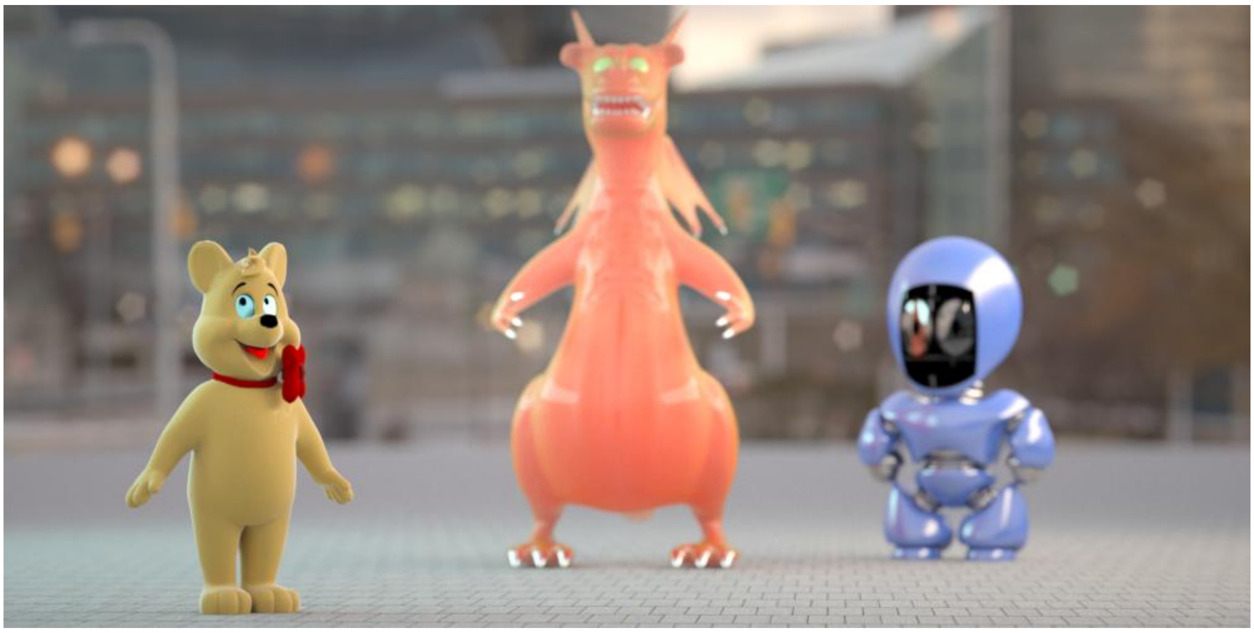
DOF enabled – enables/disables Depth of Field effect.

DOF sensor width – camera sensor width.

DOF focus distance – camera focus distance which together with sensor width determines the look of Depth of Field. This is the distance from camera position to the objects in focus.

DOF num blades – determines the shape of confusion region in rendered.

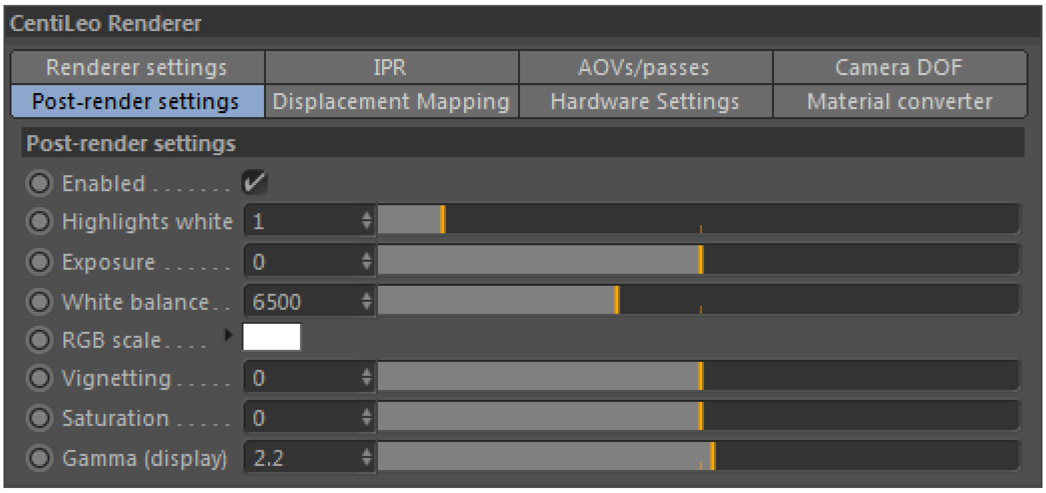
DOF blade angle – rotation of bladed confusion shape.



An example of image rendered with Depth of Field effect and pentagon confusion region (num blades = 5). These toy models are freely available on the website www.3ddd.ru (link1, link2, link3).

In IPR mode it is possible to choose the image region in focus using "focus pick" button.

Tab Post-render Parameters



Enabled ✓
Highlights white 1
Exposure 0
White balance 6500
RGB scale
Vignetting 0
Saturation 0
Gamma (display) 2.2

These parameters perform basic simple post production control over the rendered image without forcing to restart rendering (in interactive IPR mode).

enabled flag switches on/off the post production for image.

gamma value is always applied to the rendered image regardless of enabled flag.

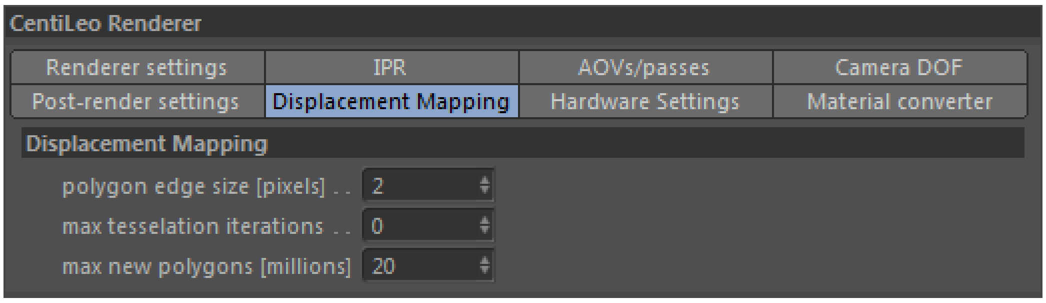
highlights white point (range 0..infinity) determines the brightness value of raw rendered image that will be white in post production output. It can be used to compress the highlights.

exposure (range -10..10) determines the overall image brightness. Positive values increase the brightness and negative values decrease the brightness.

White balance[K] (range 1500K..15000K) makes an image of cool tone with white balance less than 6500K and makes it of warmer tone with white balance greater than 6500K.

Saturation (range -1..+1) with default value 0 (unchanged image). Saturation value = -1 makes it black and white while positive values change the colors.

Tab Displacement Mapping



Displacement Mapping

polygon edge size [pixels] . . 2
max tesselation iterations . . 0
max new polygons [millions] 20

CentiLeo Renderer supports adaptive displacement mapping with interactive reaction in IPR mode to manipulations of displacement shader.

All the surfaces with displacement mapping applied are subdivided into **micropolygons** adaptively and then displaced using these parameters:

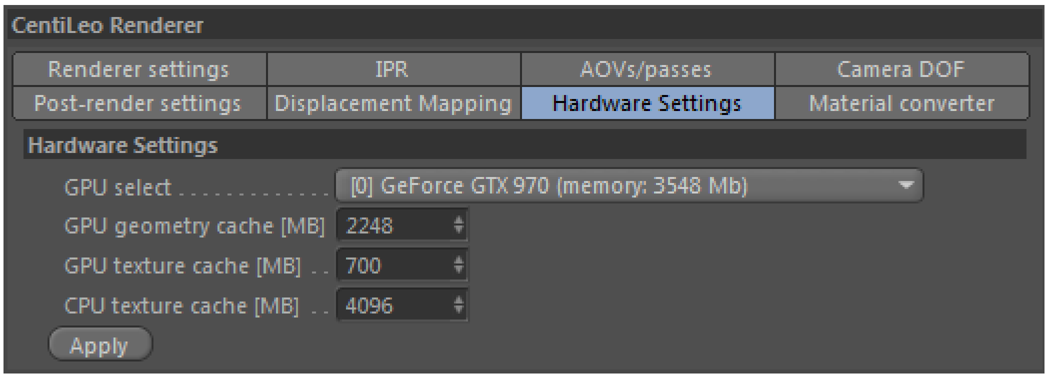
Polygon edge size[pixels] determines the size of micropolygon projection to pixels when further tessellation is not needed (i.e. tessellation precision is enough).

Max tessellation iterations parameter determines the number of tesselation iterations which are used to compute each surface tesselation and displacement.

Max new polygons [millions] determines the maximum recommended number of the micropolygons (in millions) produced by tesselator. E.g. if pixel precision of tesselation is too high then the tesselator may produce too many micropolygons but this parameter may stop tesselator if the number of already produced polygons is close to the value of **Max new polygons**.

Displacement mapping can already be used in render production. However its architecture has huge potential for improvements: vector (3D) displacement mapping, significantly faster tesselation process, memory savings and render speed increase.

Tab Hardware Settings


CentiLeo Renderer

Hardware Settings

GPU select . . . . [0] GeForce GTX 970 (memory: 3548 Mb)
GPU geometry cache [MB] 2248
GPU texture cache [MB] . . 700
CPU texture cache [MB] . . 4096

Currently CentiLeo renderer supports just one GPU from the whole computer setup. Implementing support for many GPUs is the highest priority for development team now.

The **dropdown box** lists all available computer GPU supported by CentiLeo renderer. And one can be selected. If there are several similar GPUs then it is recommended to select the one not attached to display.

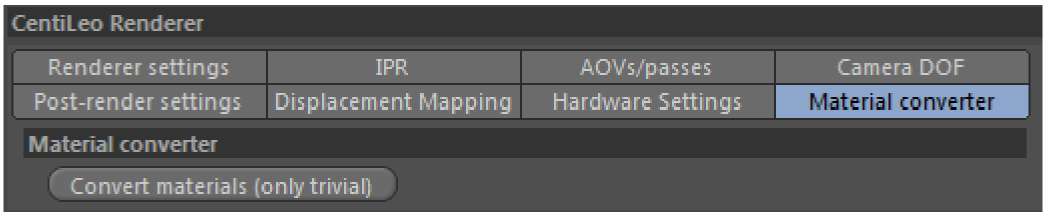
GPU geometry cache [megabytes] – determines the cache size used for scene meshes and instances storage. The parts of geometry meshes are temporarily stored in this cache when needed. The whole mesh geometry is stored in CPU RAM. Larger GPU cache size may accelerate rendering time for scenes with heavy geometry.

GPU texture cache [megabytes] – determines the cache size allocated on GPU used for textures. The parts of textures are temporarily stored in this cache when needed. Larger texture cache size may accelerate rendering time for scenes with heavy textures, however several hundred megabytes is typically enough.

CPU texture cache [megabytes] – determines the cache size allocated on CPU used for textures. The parts of textures are temporarily stored in this cache when needed. Larger texture cache size may accelerate rendering time for scenes with heavy textures avoiding many reads of textures from disk. However, several GB is typically enough.

The usage of textures (and need for larger cache) is usually driven by rendered image resolution.

Tab Scene Converter


CentiLeo Renderer

Material converter

Currently CentiLeo renderer has very limited materials converter which we just started to implement. Currently it understands standard Cinema 4D material diffuse color and diffuse color bitmap and converts it to cntlStdMat.

7. Textures Cache System

The texture system of CentiLeo renderer is a complex system that was designed for high performance in the presence of huge amount of texture files potentially of high resolution. The total amount of textures may even exceed the available RAM of computer or exceed the user defined cache size in RAM (see **Hardware Settings** rollout and parameter **CPU texture cache size**).

All the scene textures (image files) are stored on the disk in unpacked mode. For each texture in the scene CentiLeo builds the corresponding cache prepared ***.mml** file if it is not available at the first render iteration (pass).

Notice

These caches *.mml files are stored in the same folder as original referenced textures.

Building *.mml files may be a long process for the first time for 1000s of textures and consumes disc space. But later upon further scene openings and re-renderings such files are accessed much faster by the renderer than regular image formats.

The generator of the texture cache for each file is as fast or faster than the necessary native image texture reader of Cinema 4D.

The *.mml files are not cleaned automatically because they can be needed in further rendering tasks. They can be removed by a user manually.

The user defined texture caches are allocated on the CPU RAM (larger capacity) and GPU memory (smaller capacity).

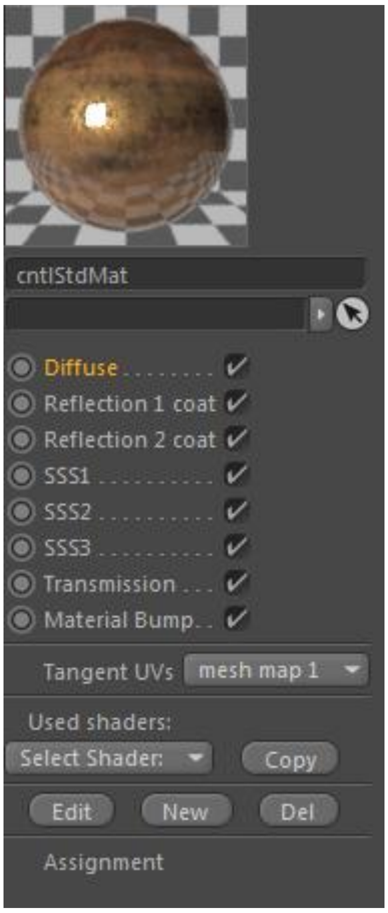
This 3-level texture caching organization provides high performance with huge amounts of textures and doesn't waste CPU RAM for storing all the textures.

8. Materials and shaders

CentiLeo Renderer currently works only with own materials:

1) CentiLeo Standard Material (in short cntlStdMat). cntlStdMat has 30 different properties including several reflections, subsurfae scatter and transmission.
2) CentiLeo Multi Material (in short cntlMultiMat). It can combine up to 10 cntlStdMat materials and has displacement and geometry mask slots.

CentiLeo Standard Material (cntlStdMat)



The stucture of *cntlStdMat* contains several internal layers wherein *Diffuse, Reflection1*, *Reflection2*, *three Subsurface Scattering layers (SSS)* and *Transmission.* The last one determines transmission/refraction properties of the surface.

cntlStdMat also has a slot *Material Bump* for bump mapping (black and white textures) or normal mapping (blue images working in tangent space). Using this slot it is possible to imitate minor details on top of the object surface. Read notice on creating bump using cntlTexture.

cntlStdMat has the dropdown list which determines the UV set that is used to compute the **tangent space** for anisotropic reflections if used in cntlStdMat.

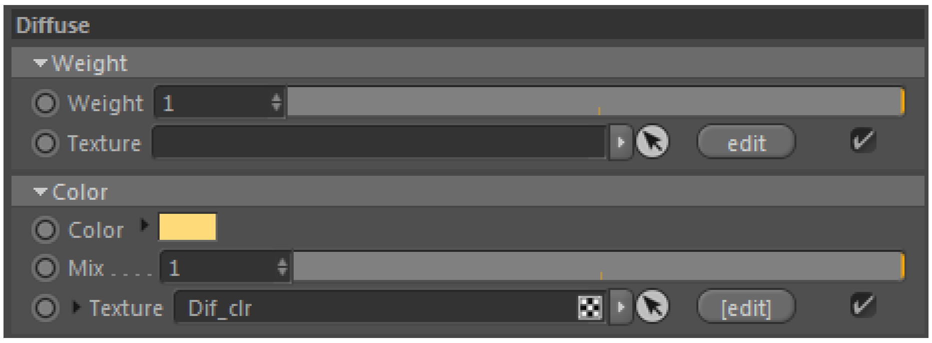
The Shader list contains all the shaders which are used in edited material.

Layer weights

Each material layer *Diffuse, Reflection1*, *Reflection2*, *SSS* and *Transmission* has weights. Each weight determines the lighting contribution coming through this particular layer relative to the contributions of the other layers.

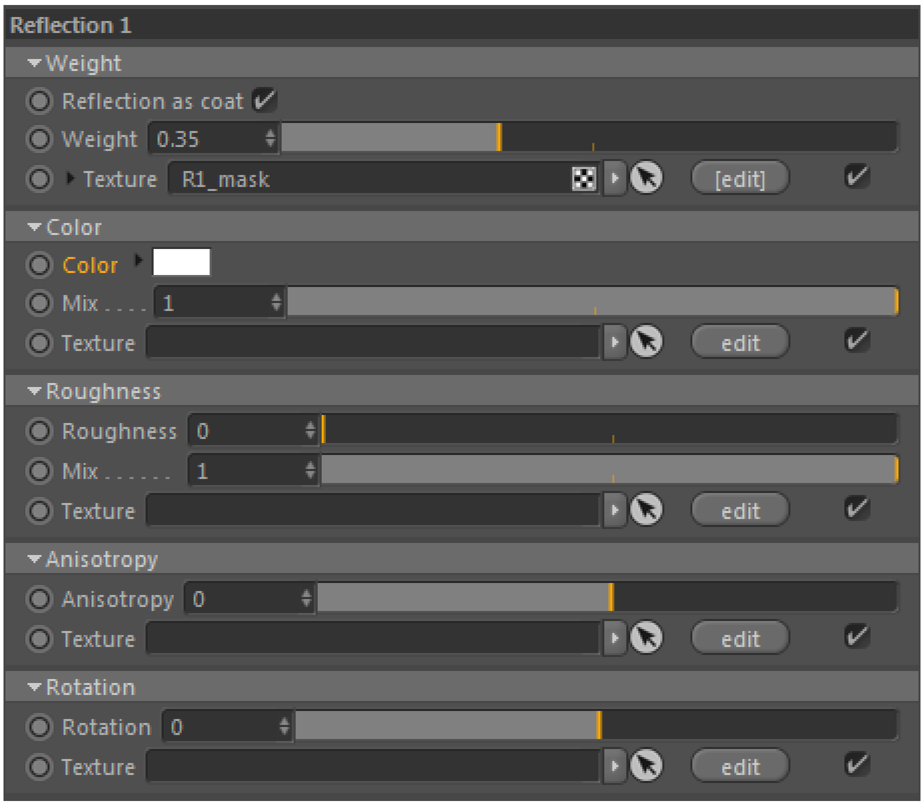
The cheboxes to the right of material property enable/disable it without actually unlinking assigned texture.

Diffuse layer



Weight	Lighting contribution coming through this layer, 1 component, range 0..1.
Color	Color of diffuse reflection, 3 RGB components, range 0..1 for each component. Mix value determines the blend between constant Color and the one obtained using the Texture.

Reflection1 layer



Reflection 1

Weight
- Reflection as coat ✓
- Weight 0.35
- Texture R1_mask [edit] ✓

Color
- Color
- Mix 1
- Texture edit ✓

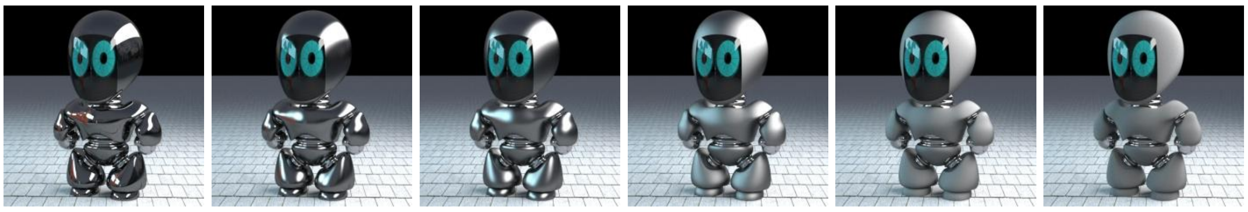
Roughness
- Roughness 0
- Mix 1
- Texture edit ✓

Anisotropy
- Anisotropy 0
- Texture edit ✓

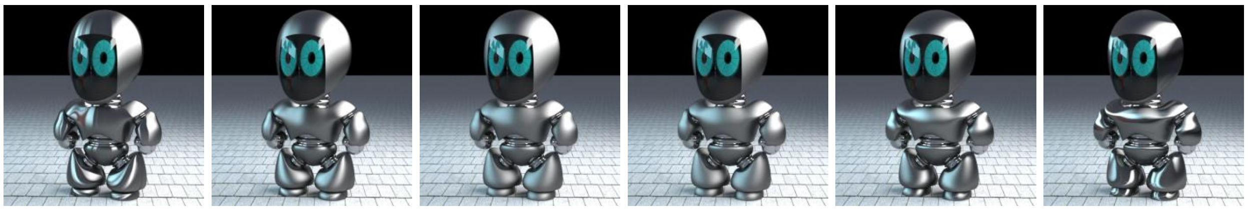
Rotation
- Rotation 0
- Texture edit ✓

Reflect as coat	Determines whether this layer contribution has more priority on top of other layers (very valuable when cntlFalloff is applied to the Weight property of reflection).
Weight	Lighting contribution coming through this layer, 1 component, range 0..1.
Color	Reflection color, 3 RGB components, range 0..1 for each component. Mix value determines the blend between constant Color and the one obtained using the Texture.
Roughness	Reflection roughness, 1 component, range 0..1 If the roughness is closer to 0 then reflections are sharper, if the roughness is closer to 1 then reflections are blurry. Mix value determines the blend between constant Color and the one obtained using the Texture.
Anisotropy	Reflection anisotropy, 1 component, range -1..1 If anisotropy > 0 (or anisotropy < 0) then reflection is stretched, i.e. it is sharper along v-component (or u-component) of texture coordinates mapped on the object surface.
Anisorot	Reflection anisotropy rotation, 1 component, range -1..1 It makes sense if anisotropy != 0. Then this parameter rotates reflection strechiness wherein the value 0.25 of anisotropy rotation corresponds to 90^o rotation and the value 0.5 of anisotropy rotation corresponds to 180^o.

Varying rougness example:

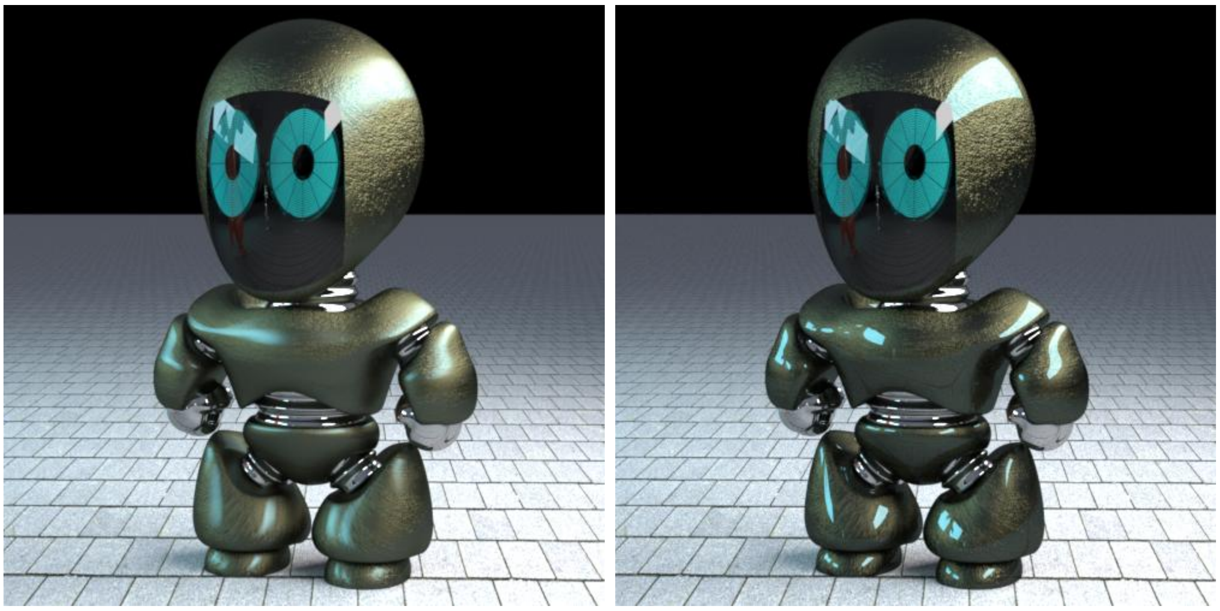

roughness = 0.0 roughness = 0.05 roughness = 0.1 roughness = 0.25 roughness = 0.5 roughness = 1.0

Varying anisotropy example:

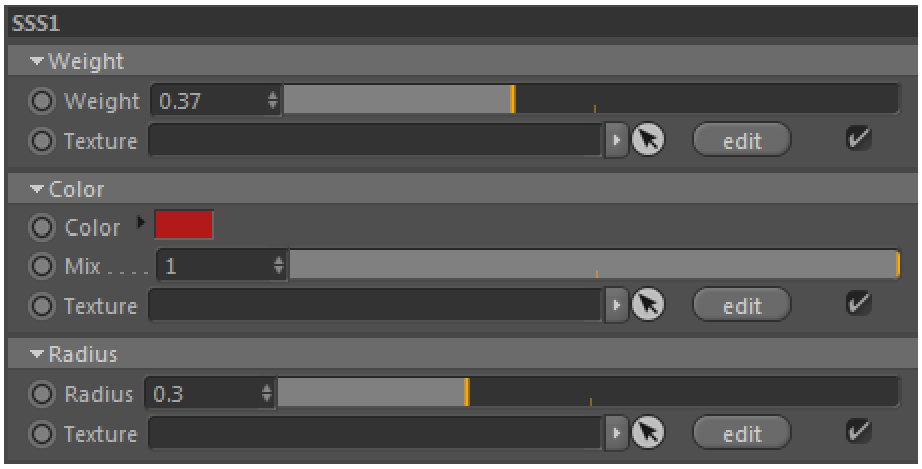

roughness = 0.0 roughness = 0.05 roughness = 0.1 roughness = 0.25 roughness = 0.5 roughness = 1.0

Reflection2 layer

Everything works the same way as for *Reflection1*. However this layer has an **additional slot for bump or normal mapping** inside only one layer - *Reflection2*. This slot was allocated only for the purposes of easier creation of reflection variation such as on example which has no bump for Reflection1 (the one with blue color) and has bump for Reflection2 (with yellow color):



Sub-surface scatter (SSS) layer


SSS1

Weight

Weight 0.37

Texture

Color

Color

Mix 1

Texture

Radius

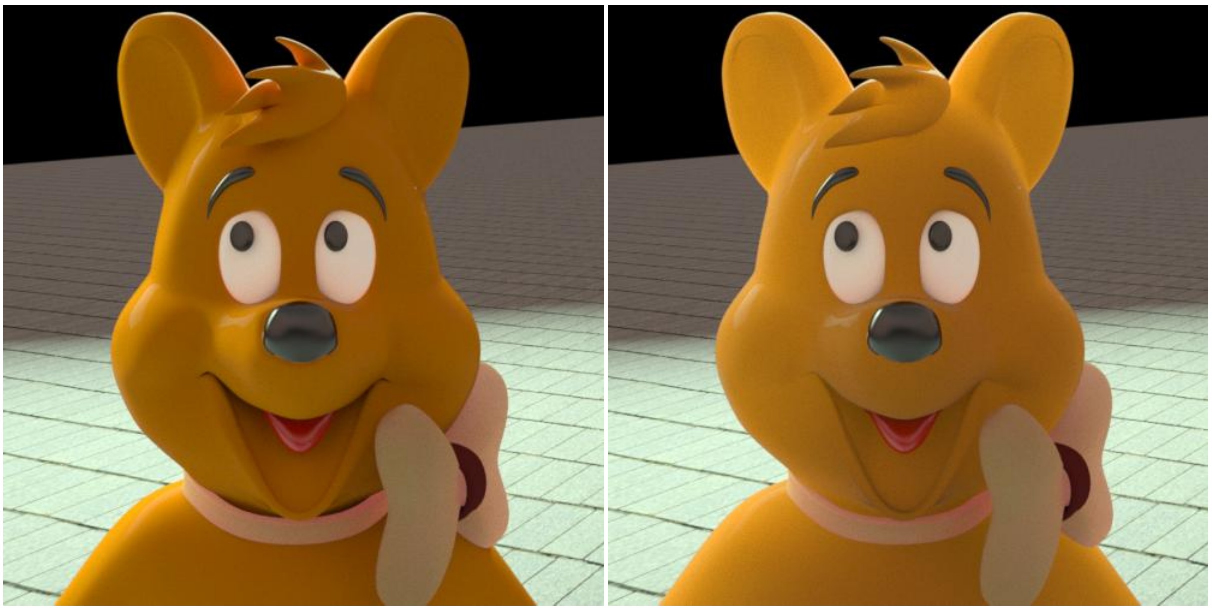
Radius 0.3

Texture

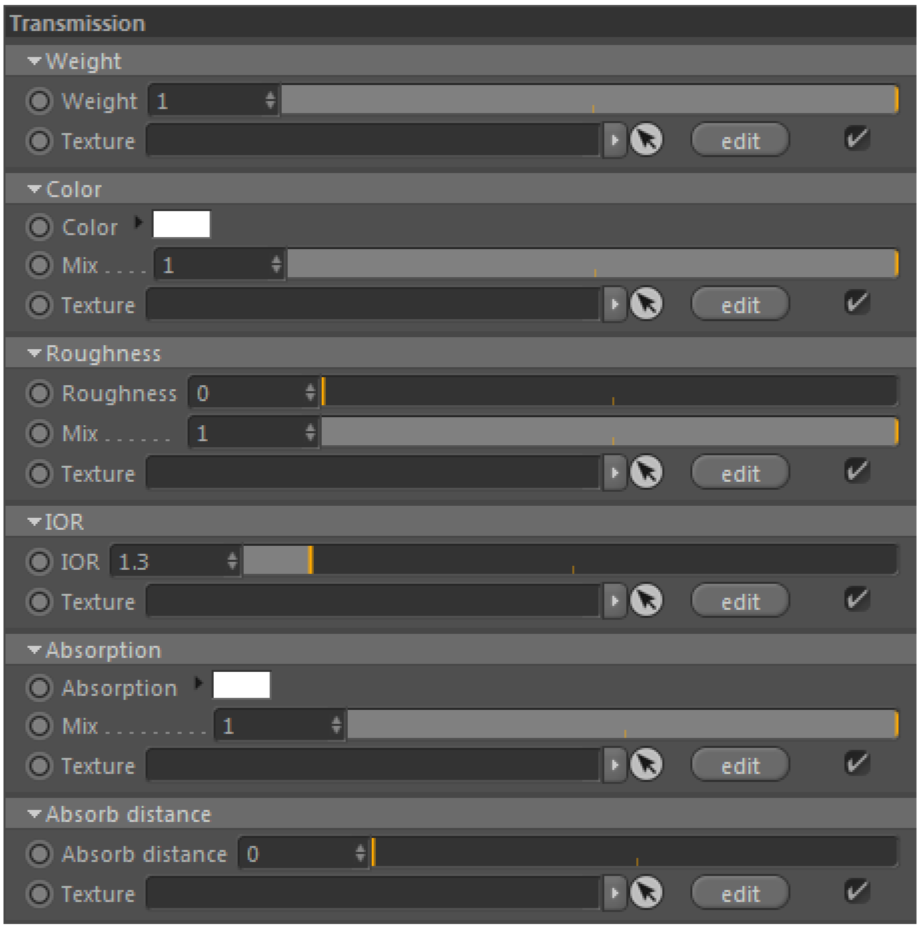
There are 3 layers with subsurface scattering that can have different radius of light scattering under the surface of the object.

Weight	Lighting contribution coming through this layer, 1 component, range 0..1.
Color	Color of subsurface scatter component, 3 RGB components, range 0..1 for each component. Mix value determines the blend between constant Color and the one obtained using the Texture.
Radius	Radius of subsufrace scatter penetration under the surface of object. Uses selected scene units.

An example of Diffuse only layer vs. Subsurface Scattering (SSS) both with a slight tint reflection with falloff mask on top**:**

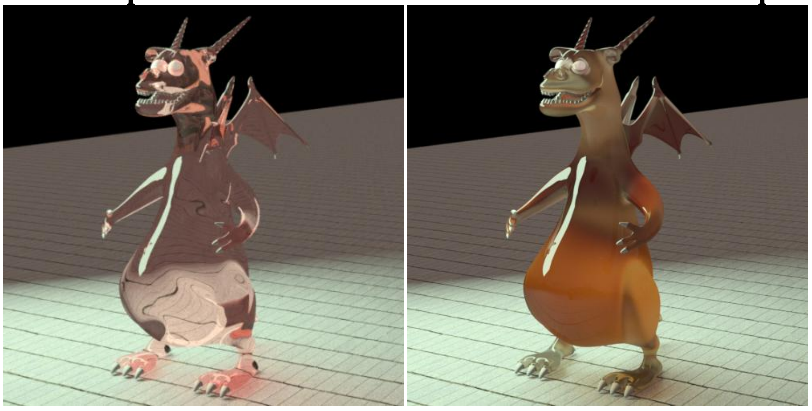


Transmission layer

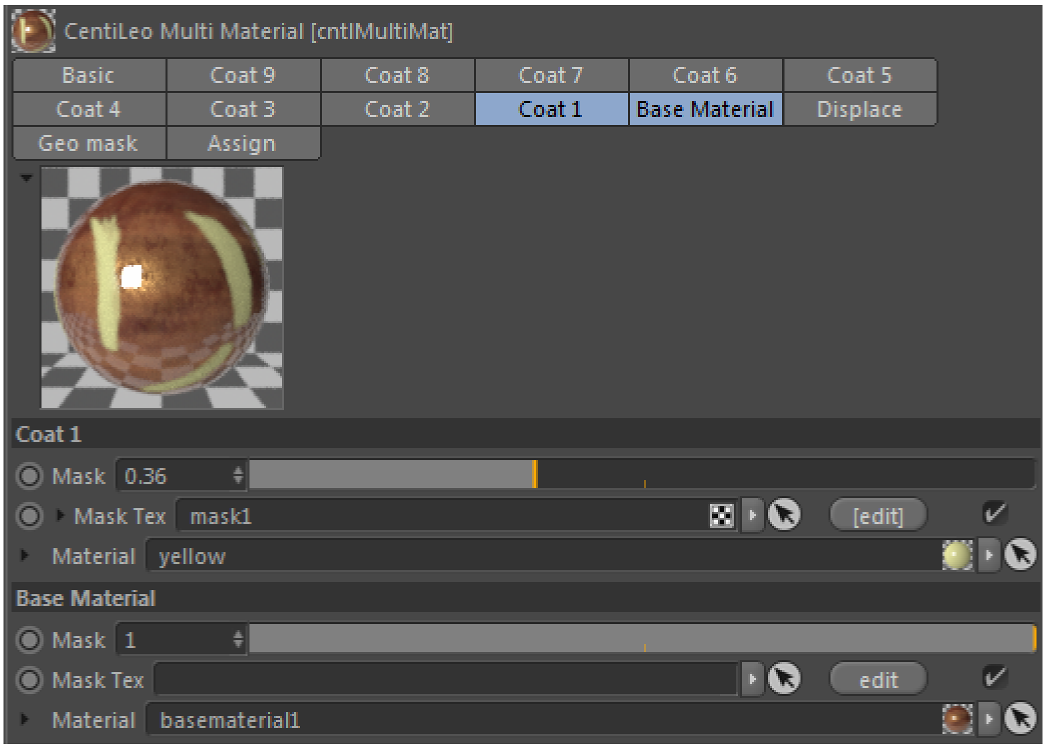

Transmission
Weight
Weight 1
Texture edit
Color
Color
Mix 1
Texture edit
Roughness
Roughness 0
Mix 1
Texture edit
IOR
IOR 1.3
Texture edit
Absorption
Absorption
Mix 1
Texture edit
Absorb distance
Absorb distance 0
Texture edit

Weight	Lighting contribution coming through this layer, 1 component, range 0..1.
Color	Ccolor, RGB. Mix value determines the blend between constant Color and the one obtained using the Texture.
IOR	Fresnel coefficient (index of refraction), 1 component, range 0 .. +infinity
Roughness	Transmission roughness, range 0..1 Transmitted light may be blurry and it can be defined with a higher value of roughness.
Absorption	Absorption coefficient, RGB, range 0..1 for each component. Transmitted light will become equal to absorption after travelling the distance absorbdist (if absorbdist > 0) inside the object. Absorption can be enabled if absorption is not white while absorbdist is greater than 0.
Absorbdist	Absorption distance, range 0..inf.

An example of transmission without and with absorption:



CentiLeo Multi Material (cntlMultiMat)



Coat 1

Mask 0.36

Mask Tex mask1 [edit]

Material yellow

Base Material

Mask 1

Mask Tex edit

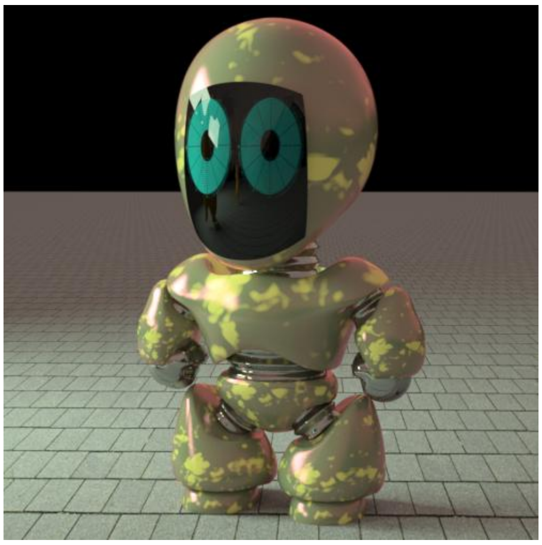
Material basematerial1

cntlMultiMat can mix up to 10 standard CentiLeo materials (cntlStdMats) wherein there is one base layer and up to 9 coats.

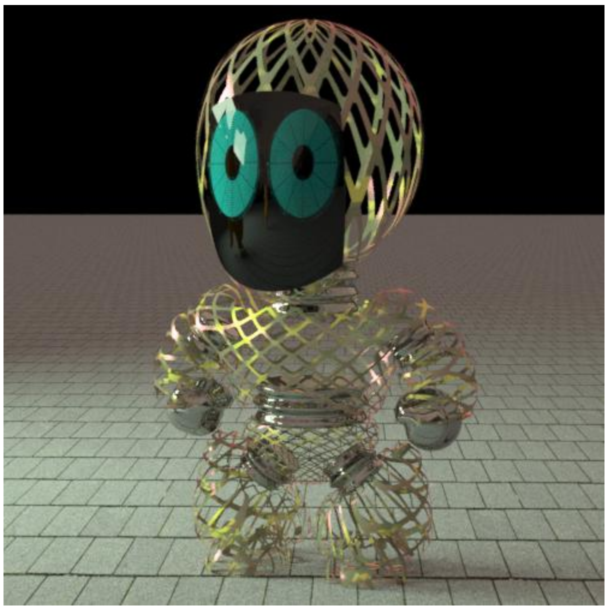
Multiple coating layers

The visibility of base and coat materials within the cntlMultiMat is determined using stack-based priority and the values of masks. In some hit point the cntlStdMat is better visible if the value of the mask is closer to one for this coat at this point. And if cntlStdMat is partially visible or not (which is indicated with a lower value of corresponding mask) then materials with lower coating number (or base material) are checked for visibility. Coating materials with higher number have higher priority over the materials with lower coat number.

If some coat or base material is visible then the lighting contribution at this point is computed based on the optical properties of the corresponding material. See example below combining 3 materials.



Geometry Masking



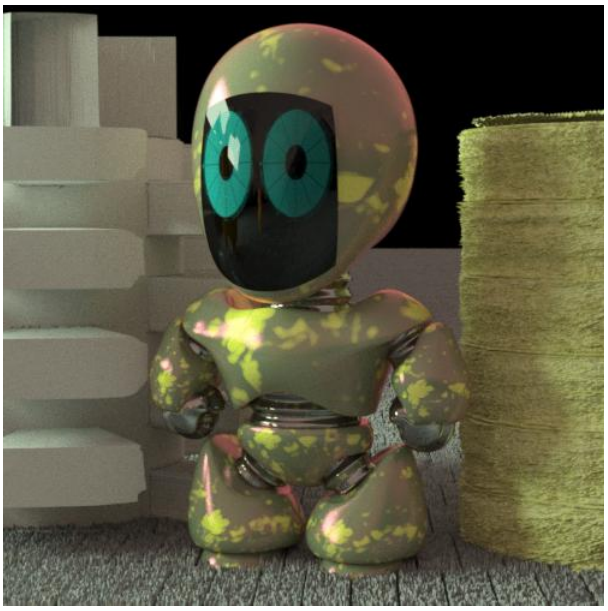
Geo Mask - is a slot of cntlMultiMat that allows creating visible or invisible surface regions (holes).

Notice

This property is driven by cntlTexture only (other shaders are not applicable in this slot for performance reasons and for unlimited depth).

This surface mask is discrete only, i.e. visible surface regions should have cntlTexture value greater than 0.5 and invisible regions should have cntlTexture value less or equal than 0.5 on the surface point. This property can be used to create the leaves with simple base geometry.

Displacement map



Displace - is a slot of cntlMultiMat that allows creating displacements.

Notice

This property is driven by cntlTexture only (other shaders are not applicable in this slot). This may change in the future.

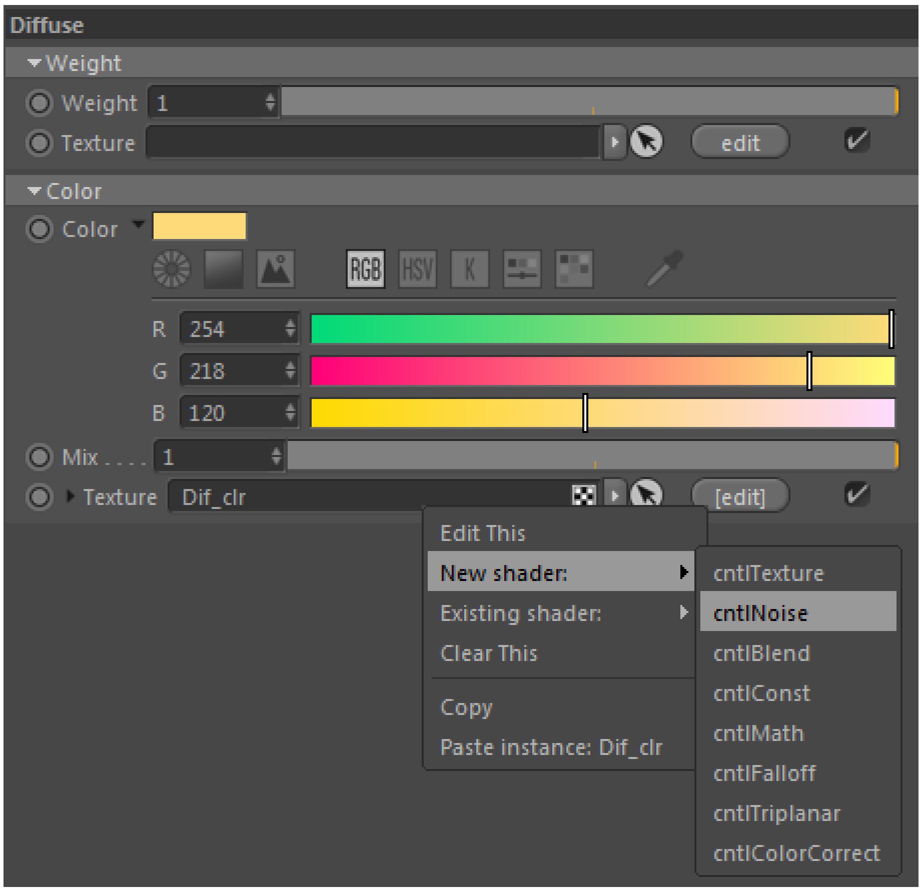
In order to enable displacement working the user should enable global displacement mapping parameters in CentiLeo Renderer settings -> Tab Displacement Mapping. There **max tesselation iterations** must be larger than zero.

In CentiLeo displacement mapping works using the tesselation process which works in the first couple of burning render iterations which can be slow. In the later render iterations the tesselator is not working and iterations run without this slowdown.

This tesselation process is a work in progress and has potential to be tremendously accelerated and vector displacement mapping can be also integrated there.

When CentiLeo IPR is working then by default it retesselates the whole scene each time the viewpoint is changed. When this is unnecessary it is possible to disable retesselation and keep existing displacement configuration. It is possible to displace the surface arbitrarilly up and below the surface using larger than 1 or even negative output values of **cntlTexture** output range.

CentiLeo Textures (shaders)



The texture can be assigned to each material property or each subtexture of the texture if applicable. To the right of each property there is a button marked with **[edit]** if the texture is assigned and just **edit** if no texture is assigned to this property.

When click edit button the following menu appears:

Edit This – goes to editor of attached texture

New Shader – shows the list of potentially new shaders

Existing Shader – shows the list of avaiable textures attached to the whole material (if we are in material editor) or texture (if we are in the texture editor).

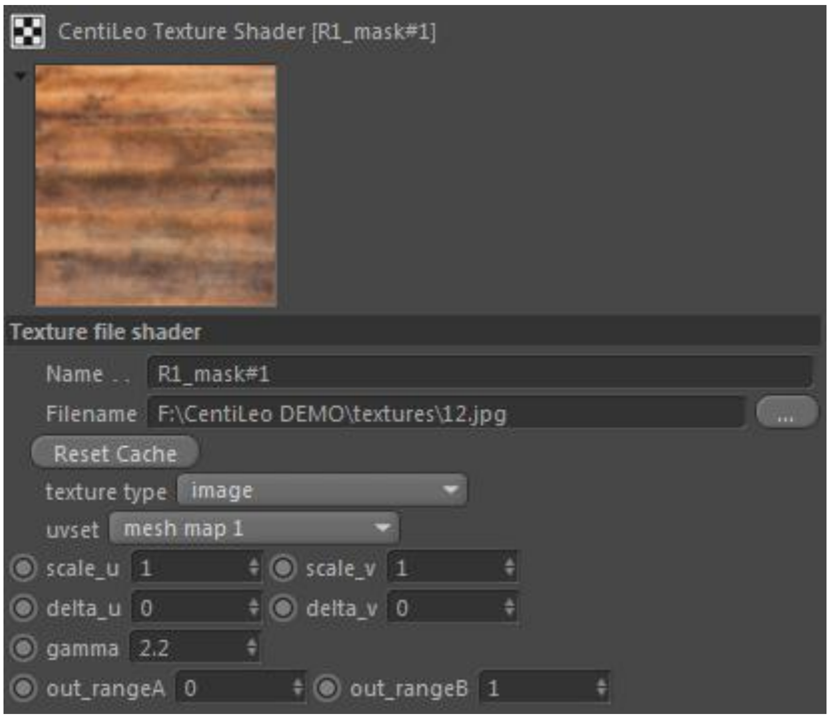
Clear This – unlinks the texture from corresponding property.

Copy – copies the shader (and it's subtree) attached to the corresponding property.

Paste instance – pastes the instance of the copied texture into corresponding property. This copy/paste supports easy sharing of textures among different materials. In later versions the paste as copy (without instancing) will be implemented.

cntlTexture (image file)

This shader defines the visual appearance of every surface point of geometry object based on fetching the texels (pixels) from the image stored in the file specified in `filename` wherein the image is mapped on top of the surface. The range of the source texel values is mapped to the range 0..1 (not 0..255). Fetched texel value is raised to the power of `gamma` and then it is linearly scaled to the output range `[out_rangeA ..out_rangeB]`.


CentiLeo Texture Shader [R1_mask#1]

Texture file shader

Name .. R1_mask#1

Filename F:\CentiLeo DEMO\textures\12.jpg

Reset Cache

texture type image

uvset mesh map 1

scale_u 1 scale_v 1

delta_u 0 delta_v 0

gamma 2.2

out_rangeA 0 out_rangeB 1

Parameters:

Filename - path to file with source texture data. Supported file formats: BMP, PNG, JPG, JPEG, TGA and others supported by FreeImage.

uvset - texture UV coords that will be applied for this texture map. Currently support only the single mesh_uvset and spherical mapping. More UV sets will be added later.

tex_type – either **tex_image** (for regular images) or **tex_normal** (for bluish normal map) or **tex_bump** (for black/white bump map)

scale_u - texture scale for u-coordinate of shader mapping (implements tiling)

scale_v - texture scale for v-coordinate of shader mapping (implements tiling)

delta_u - texture offset of scaled texture along u-coordinate of UV map

delta_v - texture offset of scaled texture along u-coordinate of UV map

gamma - the power that is used to raise the value of the fetched texels before linear conversion to the output range.

out_rangeA - lower limit of the output range for texels. The value of 0 of the source texture is translated to **out_rangeA**.

out_rangeB - upper limit of the output range for texels. The value of 1 of the source texture is translated to **out_rangeB**.

File Reset - updates the texture file in the render if it is edited outside the app

The values scaled to the output range are used as texture shader results at the surface of the point wherein the shader is connected.

Notice

It is possible to use large positive and negative values for both out_rangeA and out_rangeB and hence implement inverted image or apply these texture driven results to the properties such as **displacement mapping amount**, **subsurface scattering radius** or **absorption distance** that are measured in scene units.

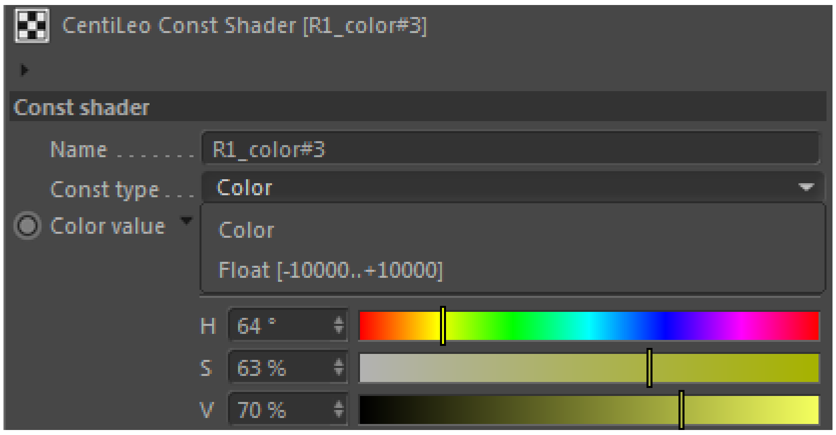
Notice for Bump maps

If the texture is used as in the shader tree connected to bump map material property then tex_type should be **tex_bump**.

Notice for Normal maps

If the texture is used shader tree connected to the bump map material property then tex_type should be **tex_normal**.

cntlConst (constant value)



CentiLeo Const Shader [R1_color#3]

Const shader

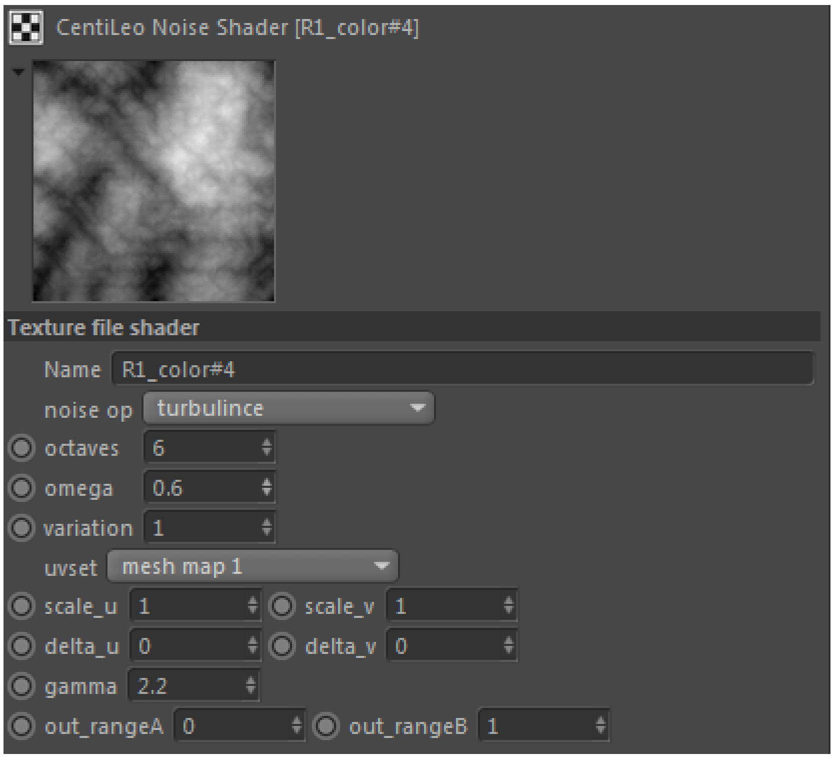
Name . . . . R1_color#3

Const type . . . Color

Color value

This shader produces the constant value, either the single component (arbitrary negative or positive) or RGB. It can be used to setup larger values to the properties wherein there is not enough range in UI like IOR with a range [1..20].

cntlNoise (procedural)


CentiLeo Noise Shader [R1_color#4]

Texture file shader

Name: R1_color#4

noise op: turbulince

octaves: 6

omega: 0.6

variation: 1

uvset: mesh map 1

scale_u: 1 scale_v: 1

delta_u: 0 delta_v: 0

gamma: 2.2

out_rangeA: 0 out_rangeB: 1

This shader produces black and white value driven by noise generator based on Perlin noise procedure with parameters **octaves**, **omega** and **variation** which determine the number and pattern of noise details.

uvset - texture UV coords that will be applied for this texture map. Currently support only the single mesh_uvset and spherical mapping. More UV sets will be added later.

scale_u - texture scale for u-coordinate of shader mapping (implements tiling).

scale_v - texture scale for v-coordinate of shader mapping (implements tiling).

delta_u - texture offset of scaled texture along u-coordinate of UV map.

delta_v - texture offset of scaled texture along u-coordinate of UV map.

power - the power that is used to raise the value of procedural noise before linear conversion to the output range.

out_rangeA - lower limit of the output range for noise. The value of 0 of the source noise value is translated to **out_rangeA**.

out_rangeB - upper limit of the output range for noise. The value of 1 of the source noise is translated to **out_rangeB**.

Notice

This shader produces black and white values. Colored values of noise can be obtained using cntlBlend shader.

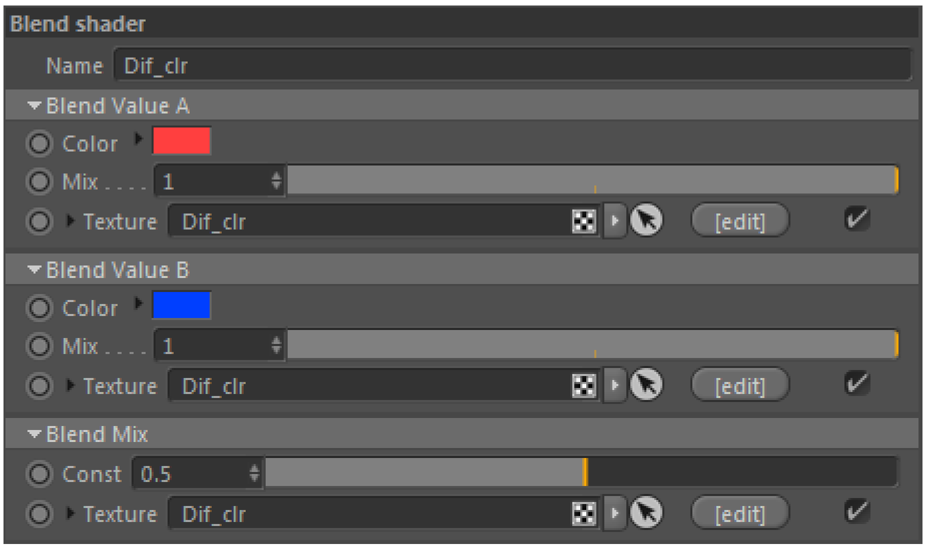
Notice

It is possible to use large positive and negative values for both out_rangeA and out_rangeB and hence implement inverted image or apply these results to the properties such as **subsurface scattering radius** or **absorption distance** that are measured in scene units.

Notice

Later this shader will be also implemented for displacement mapping.

cntlBlend (mix of textures)



Blend shader

Name Dif_clr

Blend Value A

Color 🟥

Mix 1

Texture Dif_clr [edit] ✓

Blend Value B

Color 🟦

Mix 1

Texture Dif_clr [edit] ✓

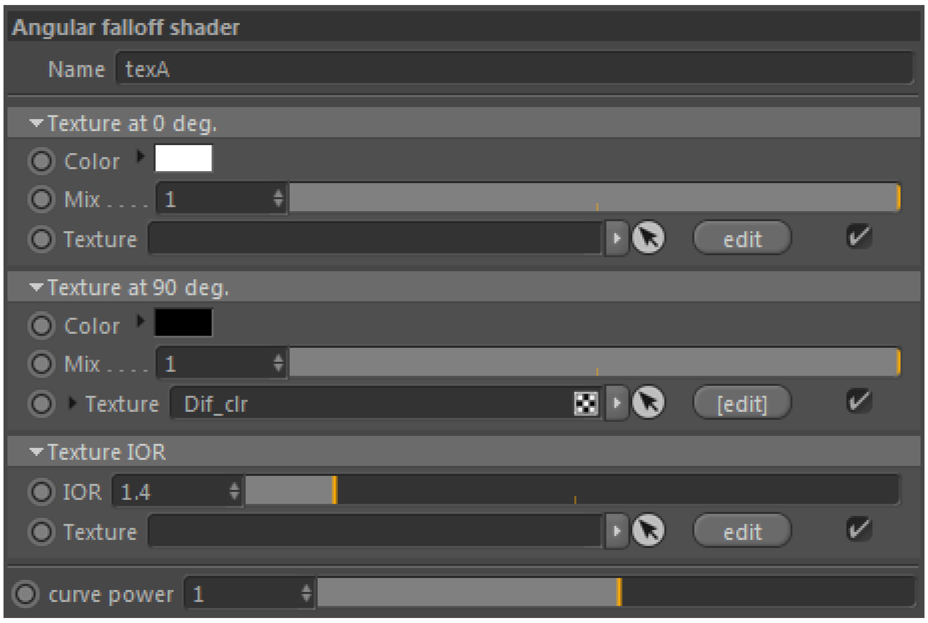
Blend Mix

Const 0.5

Texture Dif_clr [edit] ✓

This texture mixes results of two textures connected to **Blend Value A** and **Blend Value B** slot blending them together using the blend weight clamped to the range [0..1] and produced by the texture executed in **Blend Mix**.

cntlFalloff (angular mix of textures)



Angular falloff shader

Name texA

Texture at 0 deg.

Color ⬜

Mix 1

Texture edit ✓

Texture at 90 deg.

Color ⬛

Mix 1

Texture Dif_clr [edit] ✓

Texture IOR

IOR 1.4

Texture edit ✓

curve power 1

This texture defines the visual appearance of the point on the surface of object for the angle of view of 0° and for the angle of view of 90° using different textures. The visual appearance of the point for intermediate angles of view depends on these textures and parameters IOR and power.

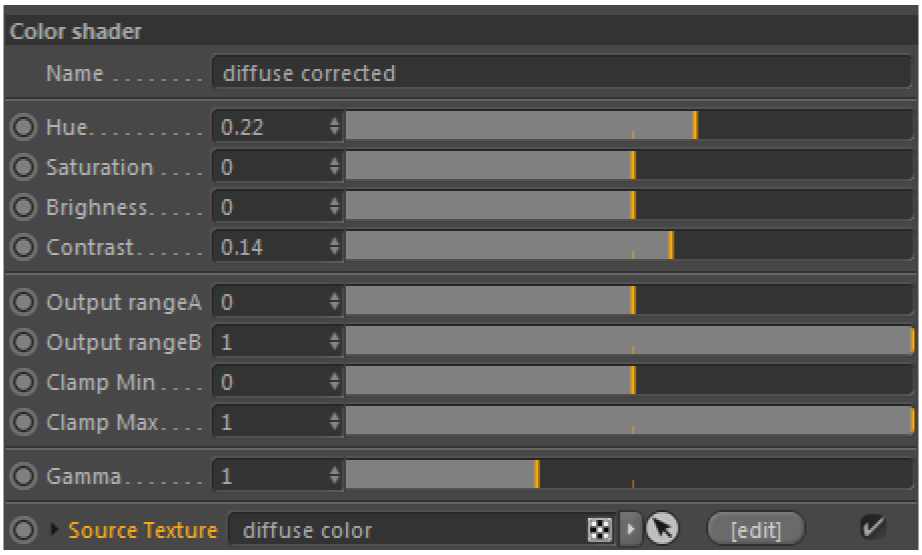
Texture at 0deg. – slot for texture that defines the visual appearance of the point for the angle of view of 0°.

Texture at 90deg. – slot for texture that defines the visual appearance of the point for the angle of view of 90°.

Textrure IOR - index of refraction that is used to compute Fresnel term that drives the angular falloff between **Texture at 0deg.** and **Texture at 90deg.**

Power - power that is used to raise the value of Fresnel term for reflection which is usefull to shift angular falloff.

cntlColorCorrection



Color shader

Name	diffuse corrected
Hue	0.22
Saturation	0
Brighness	0
Contrast	0.14
Output rangeA	0
Output rangeB	1
Clamp Min	0
Clamp Max	1
Gamma	1
Source Texture	diffuse color

[edit]

This shader is used to changed the value of any other CentiLeo shader.

shaderA – slot of shader which output is corrected here.

hue – hue shift from -1 to 1 wherein the value of 0 specifies no change.

saturation – saturation shift from -1 to 100 wherein the value of 0 specifies no change.

brightness – brightness shift from -1 to 100 wherein the value of 0 specifies no change.

contrast – constast change shift from -1 to 100 wherein the value of 0 specifies no change.

gamma - power that is used to raise the values of the source shader.

out_rangeA - lower limit of the output range. The value of 0 of already affected shader values is translated to **out_rangeA**.

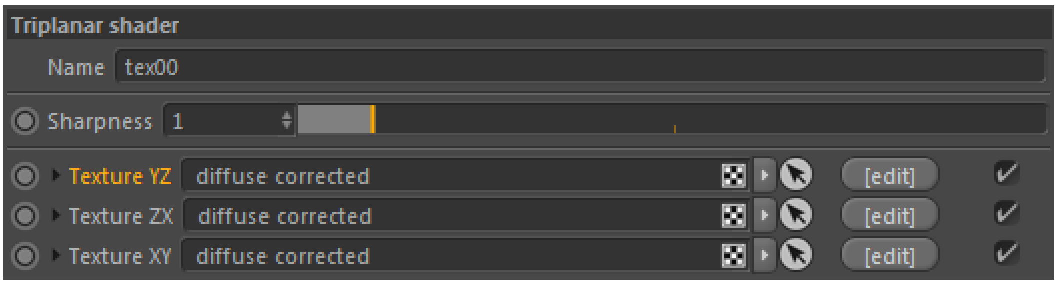
out_rangeB - upper limit of the output range. The value of 1 of already affected shader values is translated to **out_rangeB**.

Output value is clamped to the range specified by **clampMin** and **clampMax**.

Notice

E.g. for color inversion set out_rangeA = 1 and out_rangeB = 0

cntlTriplanar (UV projection)


Triplanar shader

Name tex00

Sharpness 1

Texture YZ diffuse corrected [edit]

Texture ZX diffuse corrected [edit]

Texture XY diffuse corrected [edit]

Triplanar UV mapping for any texture is implemented using this special shader.

This shader basically maps three textures (three results of evaluated textures' tree) to each of the base coordinate planes and can be used to apply the textures to the objects without dealing with explicit UV maps.

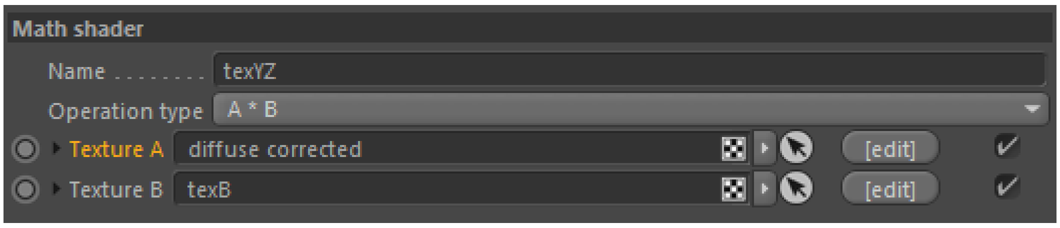
TextureYZ – slot of a texture which values are mapped along X axis to YZ plane of the local object coordinate system.

TextureZX – slot of a texture which values are mapped along Y axis to ZX plane of the local object coordinate system.

TextureXY – slot of a texture which values are mapped along Z axis to XY plane of the local object coordinate system.

Sharpness – is parameter that affects the blending among the mapped textures.

cntlBinary (math ops on textures)


Math shader

Name . . . . texYZ

Operation type A * B

Texture A diffuse corrected [edit]

Texture B texB [edit]

This texture performs math operation on **TextureA** and **TextureB**. Supported types of math operations are miltiplication (mul), addition (add), getting maximum (max), getting minimum (min), raising to the power (pow), substraction (sub), division (div).

9. Light sources

CentiLeo renderer supports native Cinema 4D lights such:

1) Environment -> Sky;
2) Omni Light
3) Spot Light
4) Target Light
5) Area Light
6) Infinite Light (up to 10 in one scene)
7) Sun Light (up to 10 with infinite lights in one scene)

From all properties of these light sources CentiLeo translates into the scene only:

- Coord. Tab
- General Tab: Color and Intensity, Type
- Details Tab: Size X/Size Y for Area light
- Details Tab: Inner/Outer Angle for Spot and Infinite light

Notice

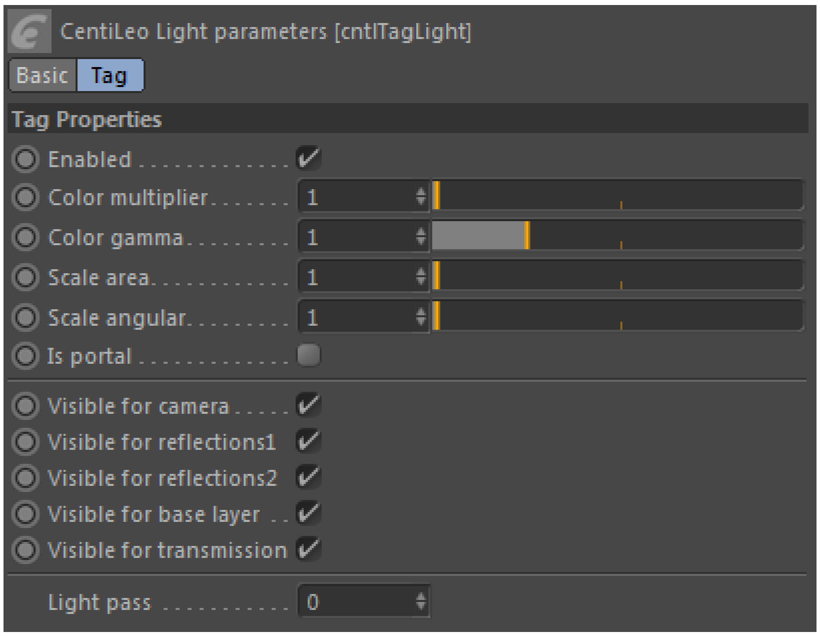
Issue to be solved: when Sun light expression tag has "Set Light Color" enabled it continuously restarts IPR render. That shouldn't be so. The issue will be solved soon.

Notice

Currently only up to 10 directional light sources are supported in CentiLeo renderer.

Tag cntlTagLight for light sources (and Portal accelerator)

More light properties can be assigned to supported Cinema light sources in CentiLeo translated scene using the tag **cntlTagLight** (except to the Environment Sky because it has different Tag).



CentiLeo Light parameters [cntlTagLight]

Basic | Tag

Tag Properties

Enabled	✓
Color multiplier	1
Color gamma	1
Scale area	1
Scale angular	1
Is portal	
Visible for camera	✓
Visible for reflections1	✓
Visible for reflections2	✓
Visible for base layer	✓
Visible for transmission	✓
Light pass	0

Color and Intensity translated from Cinema light properties are multiplied to give the light emission during render.

Enabled – switches on/off the light in CentiLeo scene.

Color multiplier additionally multiplies the color emission which will be more relevant when textured light angular distribution will appear among these setting.

Light emission is raised to the power of **Color gamma**.

Scale area scales the light source shape in addition to the Cinema light shape scaling are which helps creating soft shadows from this light source.

Scale angle scales the light source emission spread angle for Spot Light or Infinite Light/Sun in addition to the Cinema light angular scaling.

Light pass number determines the AOV image wherein the contribution of this light source should be accounted. In order to enable rendering of specific AOV corresponding to selected light pass it must be checked in CentiLeo Renderer settings -> Tab AOV.

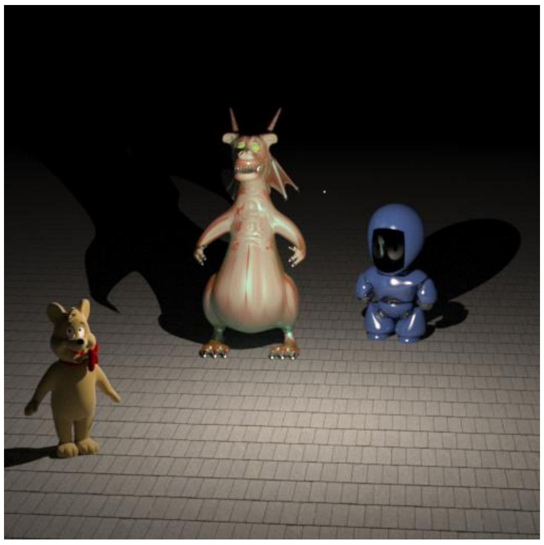
Visibility checkboxes determine the light source visibility for camera or particular CentiLeo material layers.

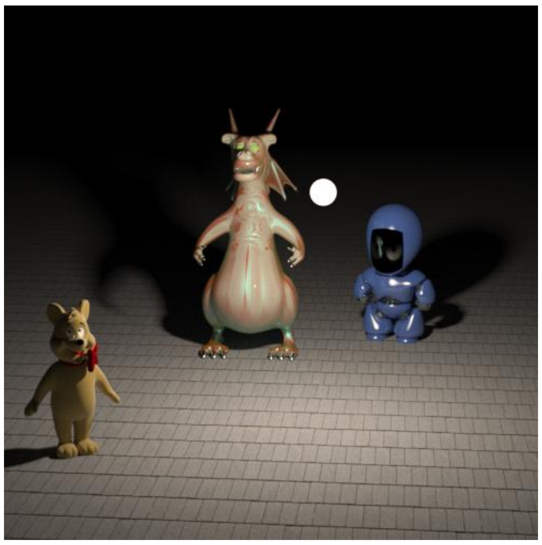
Is Portal checkbox should be enabled for interior renderings wherein the light is coming from the outside (using HDRI environment map) through the window to the room.

Notice

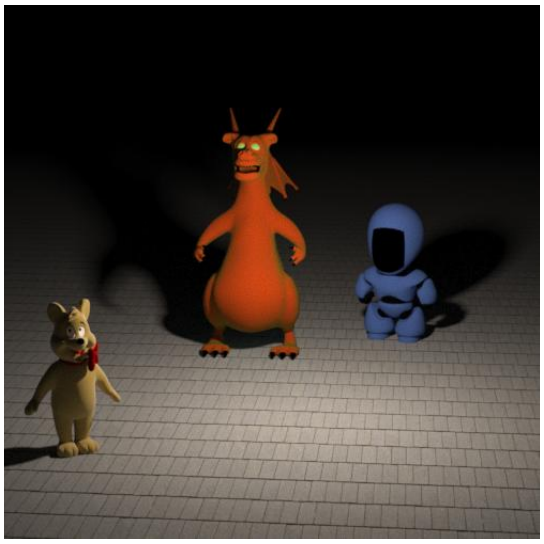
If **Is Portal** checkbox is enabled then this is not a light source anymore but a rectangle which helps the render engine to accelerate rendering task. This rectangle area should slightly overlap the window hole for better performance (however few experiments show how is it better to make).

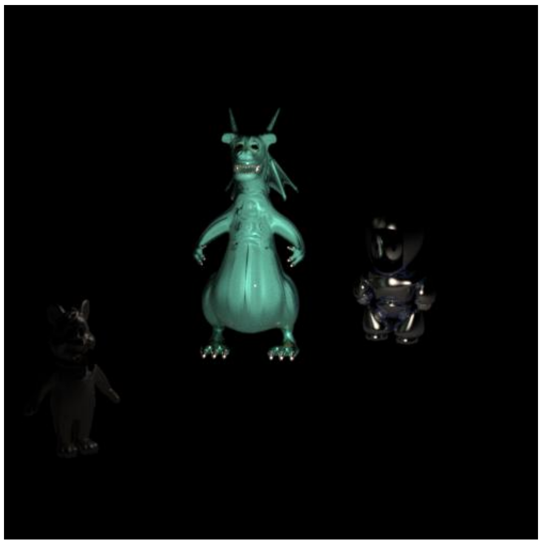
Examples of point light source with zero and large radius (see soft shadow):



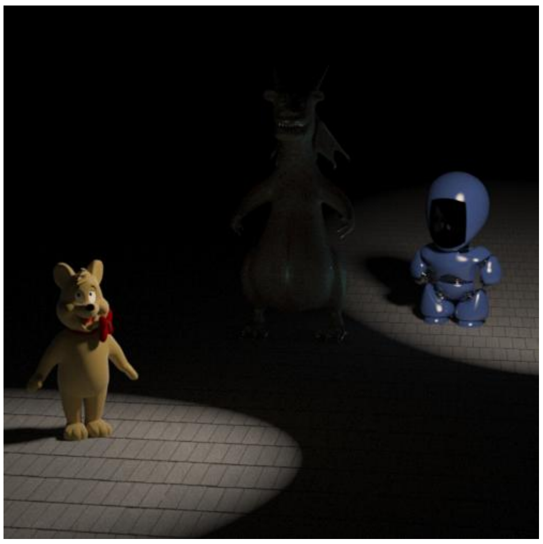


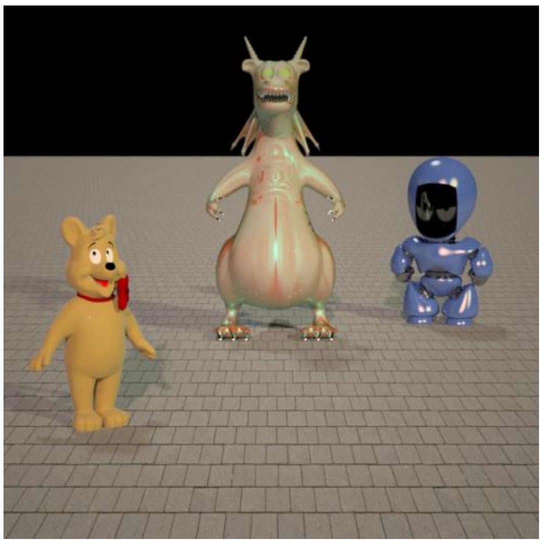
Examples of point light source visible to only base layer or only reflection1/reflection2:





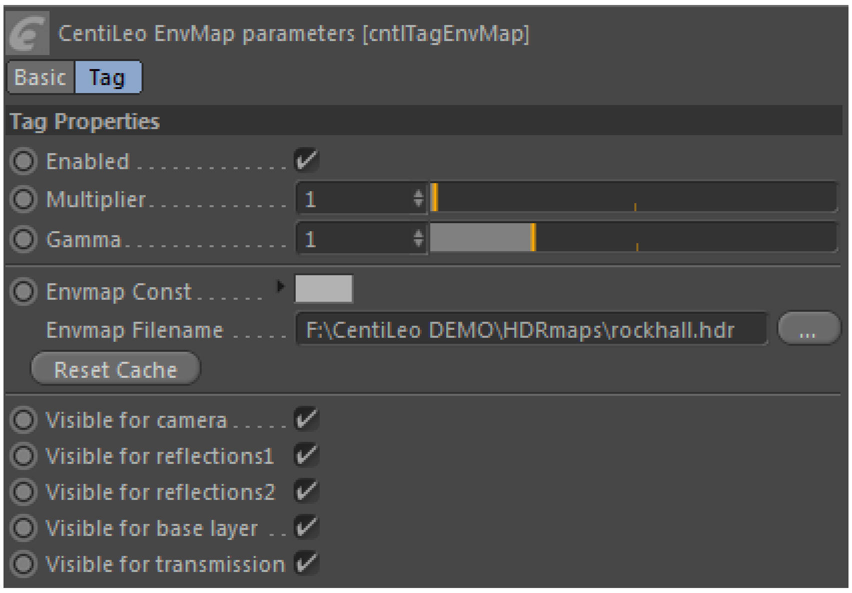
Examples of spot lights and directional infinite lights:





Tag cntlTagEnvMap for Environment Mapping

Several Environment Sky objects can be added to Cinema 4D object manager and they can be supported in CentiLeo rendered scene. The Sky objects at the beginning of the objects list override the Sky objects below them in CentiLeo scene for each lighting visibility set determined by **Visibility parameters**.



CentiLeo EnvMap parameters [cntlTagEnvMap]

Basic Tag

Tag Properties

Enabled ✓
Multiplier 1
Gamma 1
Envmap Const
Envmap Filename F:\CentiLeo DEMO\HDRmaps\rockhall.hdr
Reset Cache
Visible for camera ✓
Visible for reflections1 ✓
Visible for reflections2 ✓
Visible for base layer ✓
Visible for transmission ✓

Envmap Filename specifies the path to file of environment texture.

Reset Cache upon click recalculates the version of Environment Map that is considered in the rendered scene.

Color multiplier multiplies the color emission given by Environment Map texture.

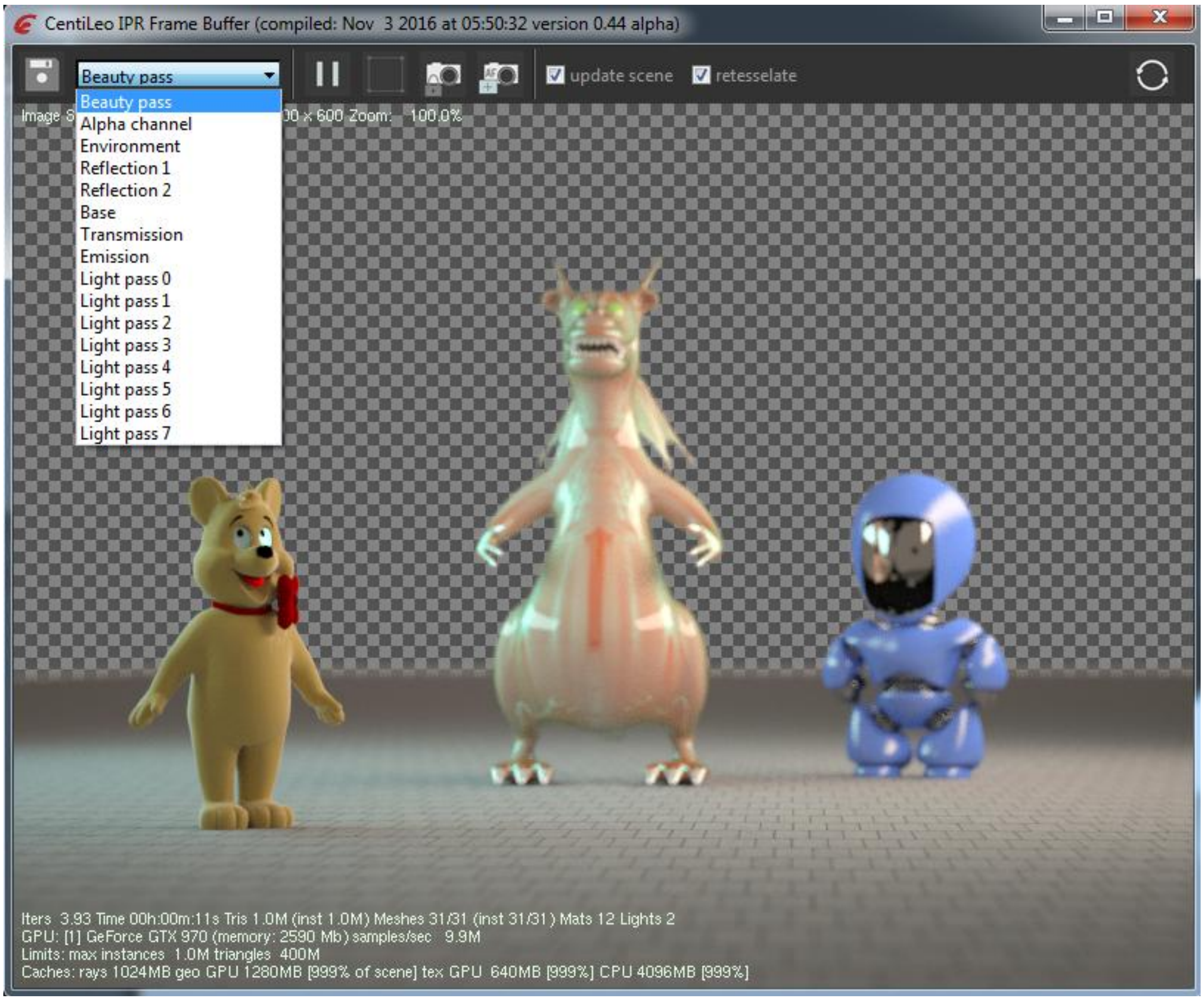
Brightness of each pixel of Environment map is raised to the power of **Color gamma**.

If no filename is specified then **Envmap Const** is used to determine the environmental constant color.

Notice

By default the light pass of each Environment Sky object is 0.

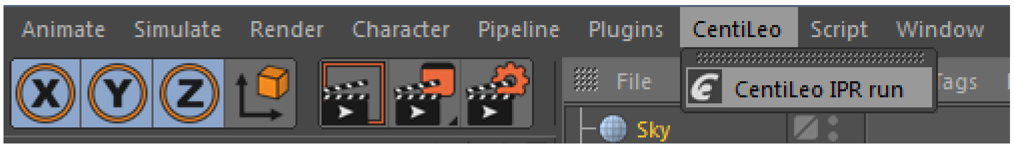
10. Interactive Preview Rendering (IPR)



```
Iters 3.93 Time 00h:00m:11s Tris 1.0M (inst 1.0M) Meshes 31/31 (inst 31/31 ) Mats 12 Lights 2
GPU: [1] GeForce GTX 970 (memory: 2590 Mb) samples/sec  9.9M
Limits: max instances  1.0M triangles  400M
Caches: rays 1024MB geo GPU 1280MB [999% of scene] tex GPU  640MB [999%] CPU 4096MB [999%]
```

GUI for this Frame Buffer was developed using wxWidgets library (www.wxwidgets.org)

Interactive Preview Render can be started by clicking the button "**CentiLeo IPR run**" in the Cinema 4D top menu -> CentiLeo menu item. At first time it waits for full scene load into CentiLeo Renderer memory.



When scene is loaded a window titled "CentiLeo IPR Frame Buffer" appears and show the current scene. All the Cinema 4D edits appear in this window which displays the render in process wherein the noise is gradually reduced for the scene currently edited.

The dropdown list shows supported render system AOVs. When the item from this list is selected the Frame Buffer shows selected AOV. In order to display selected AOV correctly the corresponding checkbox must be enabled in CentiLeo Renderer settings -> Tab AOV.

Transfers the Beauty pass and all checked AOVs in CentiLeo Settings to Picture Viewer

Pause/Resume the rendering process

Enables/disables Render Region draw mode (a render region accelerates rendering in selected image area)

Locks the current camera position. It will not be automatically updated if the camera position in the viewport is updated

Enables/disables the mode of selection of autofocus point in screen space. The respective value of Camera Depth of Field is updated in Camera DOF settings

Reloads the scene from scratch

update scene – this flag indicate whether automatic scene updates are enabled or not

retesselate – this flag indicate whether automatic scene retesselation is enabled or not

Notice

If the checkbox "**retesselate**" is enabled then all the scene objects with applied displacement mapping are retesselated automatically for each given viewpoint change. Sometimes it can be a long process until we accelerate our tesselator.

Uncheking "**update meshes**" and "**retesselate**" may accelerate the interactive rendering experience while material and shading edits are performed.

"**focus pick**" button enables a mode when by clicking to certain image regions will place them to the camera depth of field focus. In order to make this effect enabled look for the settings in CentiLeo Renderer settings -> rollout Camera.

IPR status bar

The status bar of the Frame Buffer shows the various statistics about the scene including the number of objects, instances, materials and lights.

It also shows the number of passed render iterations and execution time.

The Limits line shows limitations of the scene for selected GPU given the amount of allocated GPU memory. These limits are introduced and displayed for robustness and allow operating with the scenes of size much larger than the GPU may contain in VRAM. These limitations depend on the memory overhead described in Special note on GPU memory consumption.

samples/sec is renderer performance parameter that can be affected by render settings Render settings, scene complexity and Hardware settings.

The last line of status bar shows used scene cache parameters and the portion of the scene that fits to cache. Changing the Hardware settings affects these parameters and can change rendering performance (see **samples/sec**).

11. User Manual changelog

cntlc4d 0.44 alpha

1) Added in Section 3: Special note on GPU memory consumption
2) Added in Section 6: Tab Render Parameters -> Firefly killer parameter has no effect
3) Added in Section 6: Tab IPR
4) Updated (a lot) Section 8: "Materials and Shaders behavior"
5) Updated Section 10: "Interactive Preview Rendering (IPR)"