索引

课程-01

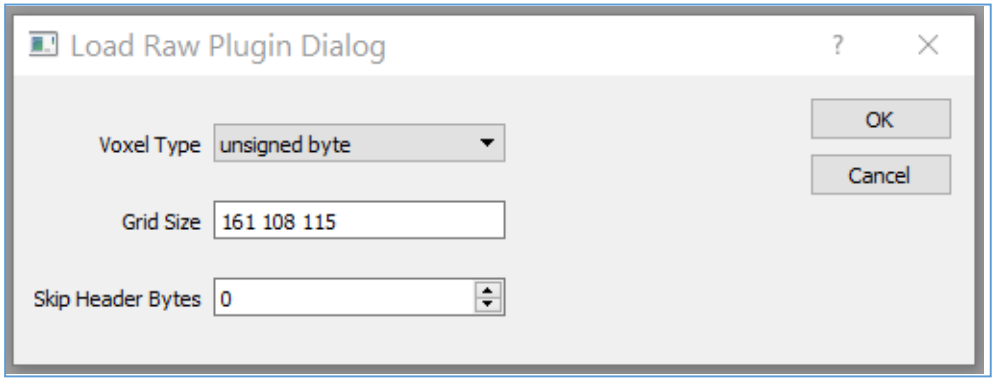
导入原始数据

打开 Drishti Import 导入。

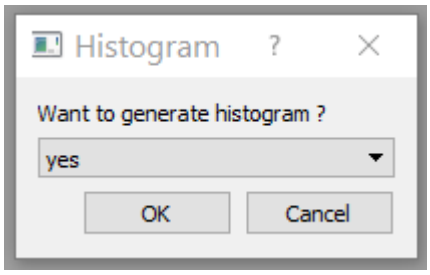
选择 tooth.raw 这个数据，然后拖入 Drishti Import。

从下拉列表的文件类型中选择选项 8：RAW Files（原始文件）。
您也可以使用另外一种加载数据的方法 - 文件 - >加载 - > 原始文件（Files->Load->RAW Files）

您将看到以下屏幕

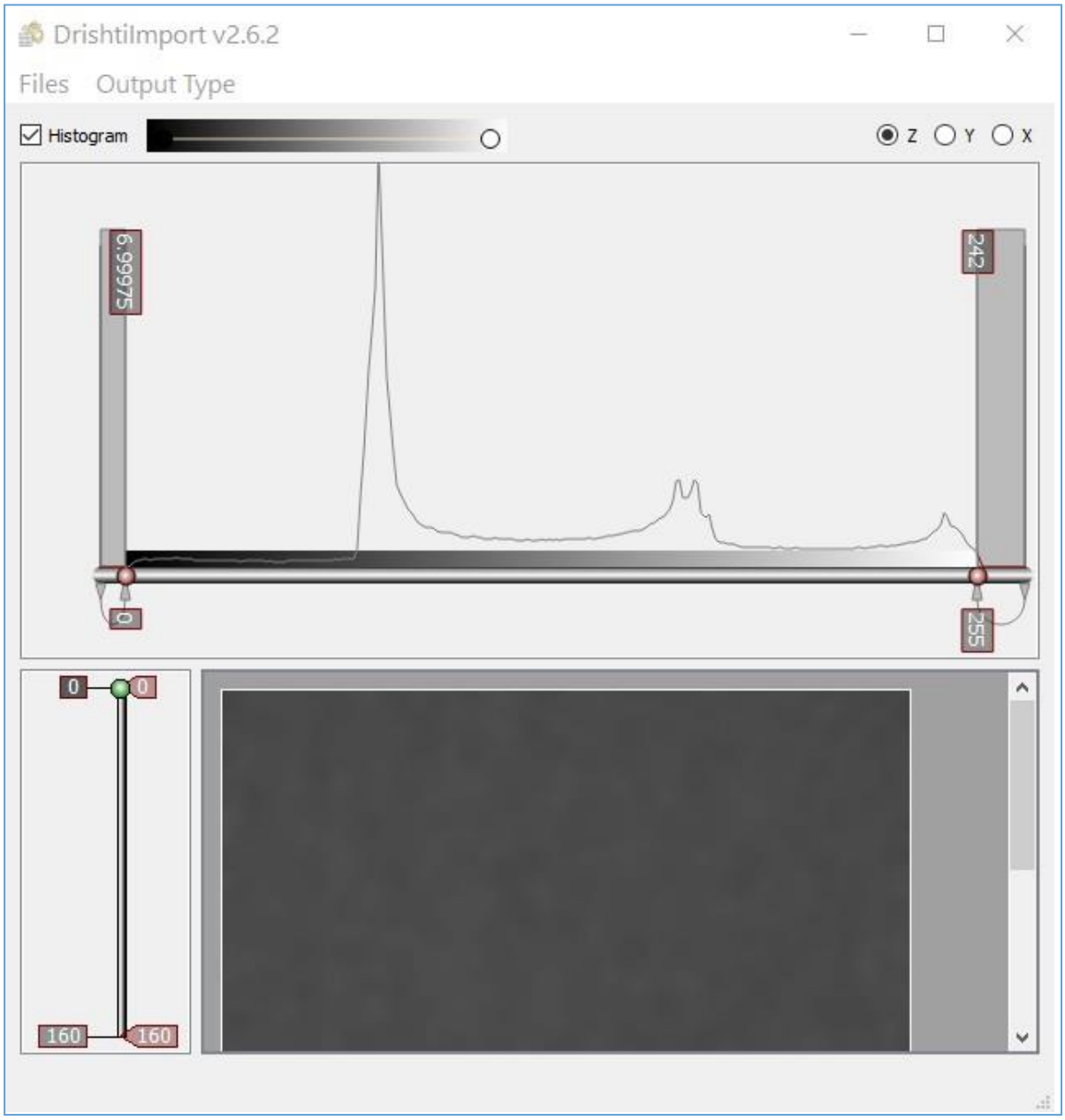


按 OK 继续下一步。



按 OK 生成直方图。

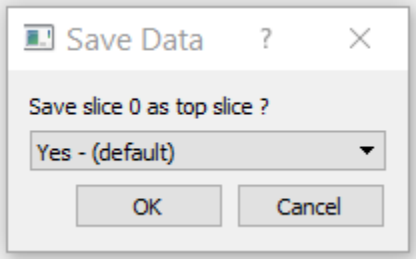
导入的数据将生成直方图并显示在窗口的顶部。底部将显示数据的第一个切片。



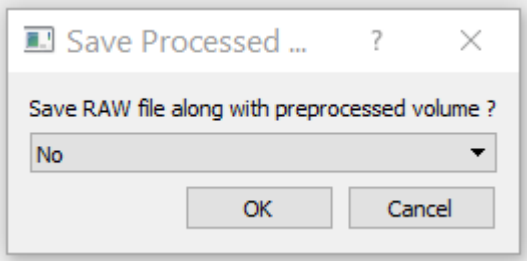
终点标记可以移动。所有超过左标记的值都映射到 0，超出正确标记的所有值都映射到 255。映射显示在顶部和底部的文本框中。 用户可以使用标记来获取映射过程。

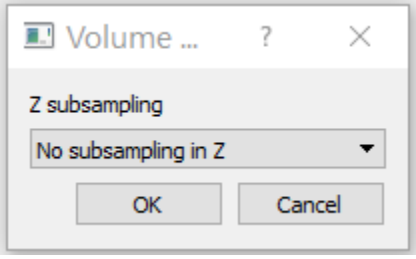
按 s 进行处理并保存卷轴。
文件 - >另存为（Files->Save As）进行另存。

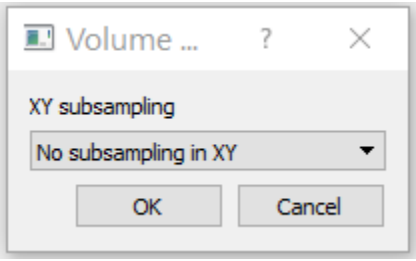
选择已处理的卷轴文件名为 tooth.pvl.nc

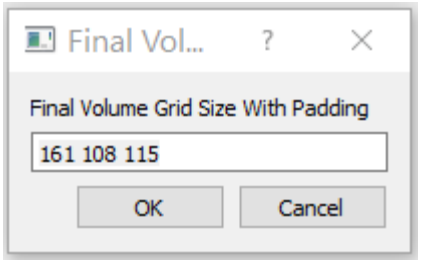

Save Data
Save slice 0 as top slice ?
Yes - (default)
OK
Cancel

连续按 OK 即可。

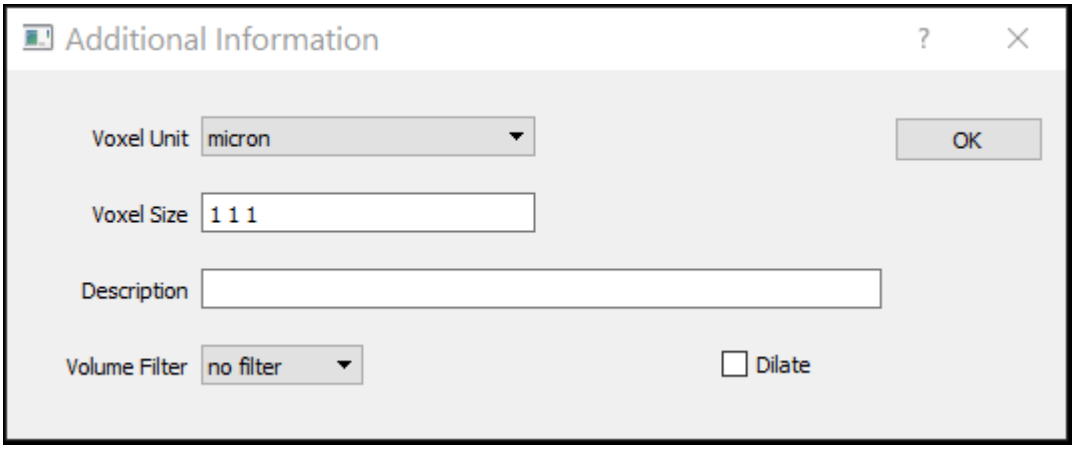

Save Processed ...
Save RAW file along with preprocessed volume ?
No
OK
Cancel


Volume ...
Z subsampling
No subsampling in Z
OK
Cancel


Volume ...
XY subsampling
No subsampling in XY
OK
Cancel



按 OK 继续下一个窗口。



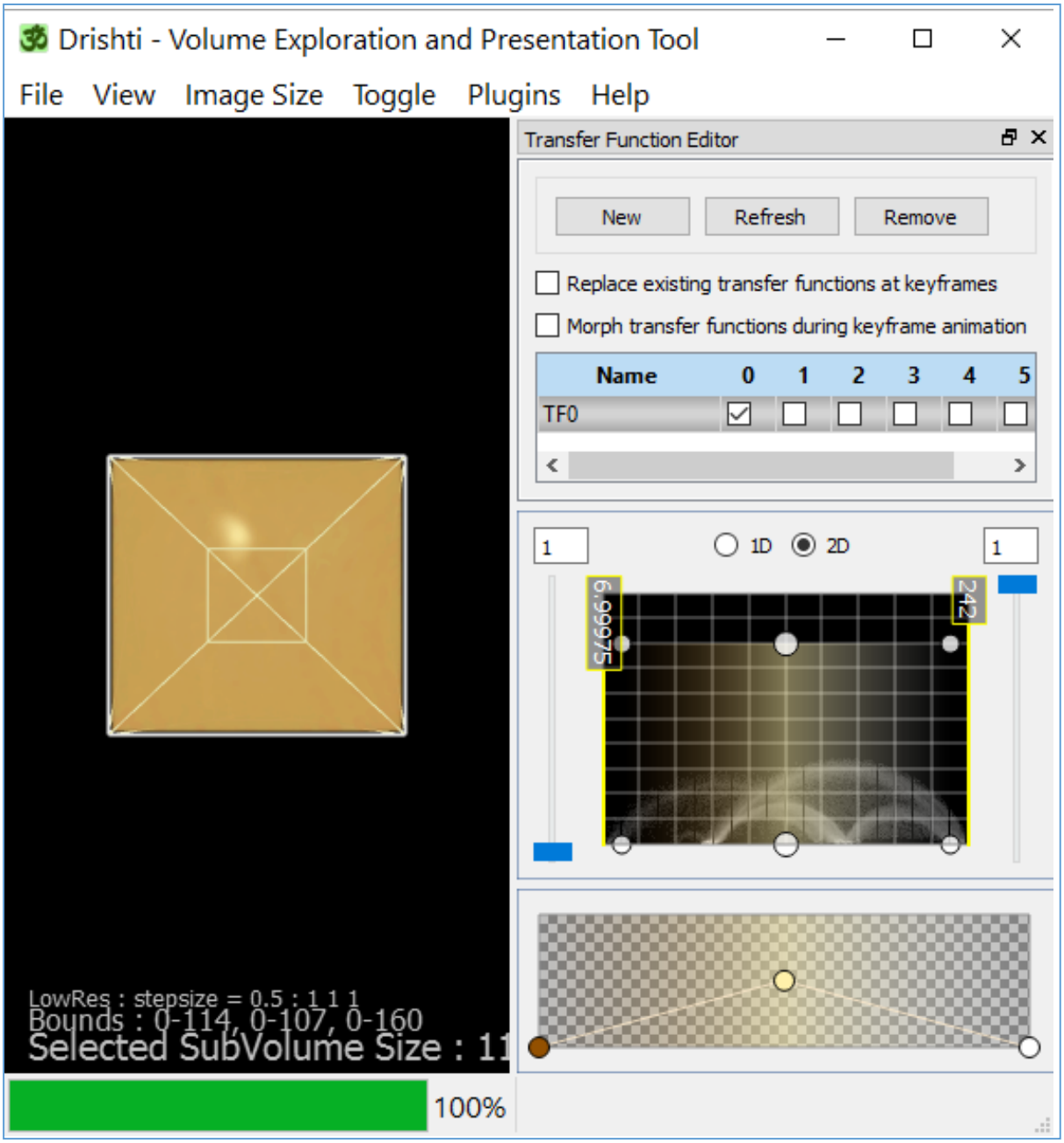
按 OK 并退出 Drishti Import。

基础渲染功能简介(运用 Drishti Render 进行可视化)

打开 Drishti Render。

将新创建的 tooth.pvl.nc 拖放到 Drishti Render 中。

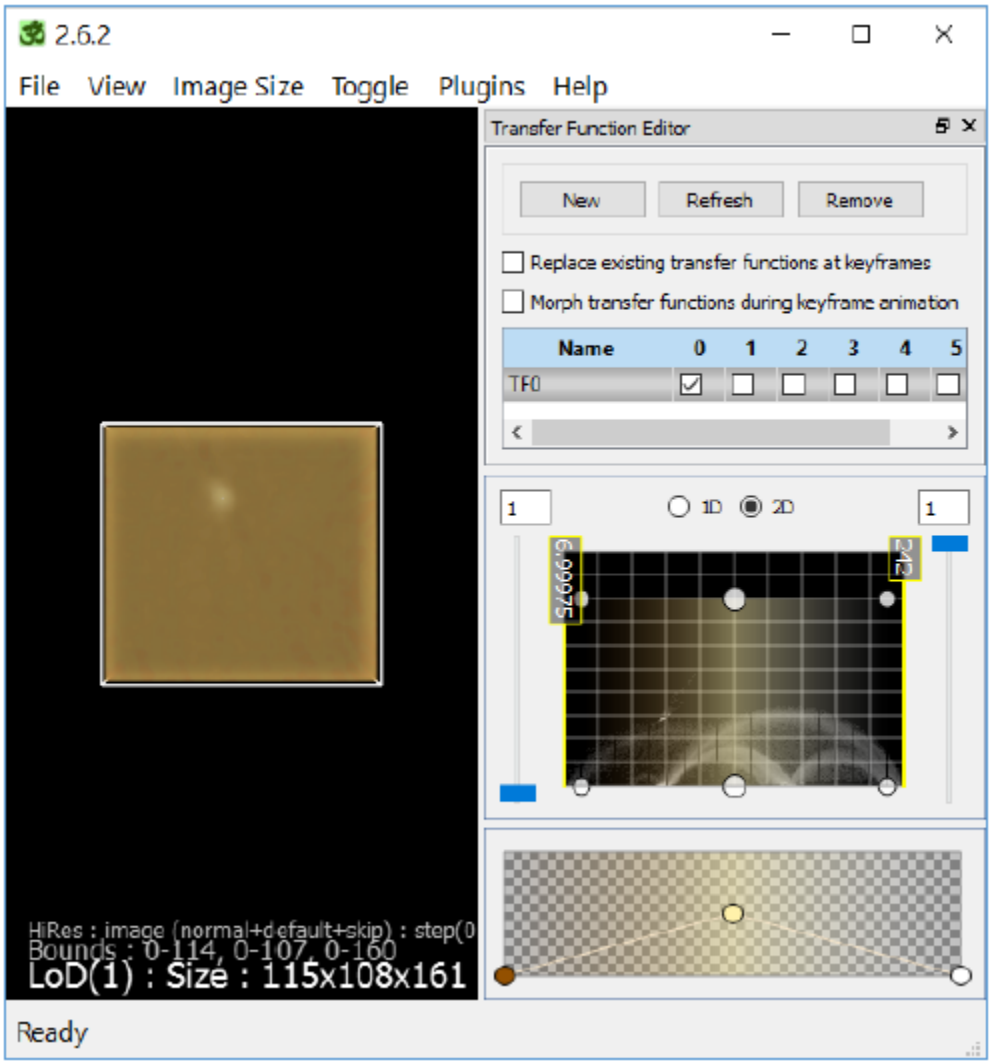
加载数据的另一种方法 - 文件 - >加载卷轴 - >加载 1 卷轴（Files->Load Volume->Load 1 Volume）您将看到以下屏



按 F2 可直接切换到 Hires Mode

或者： Toggle->Hires Mode (高清模式).

您将看到以下屏幕

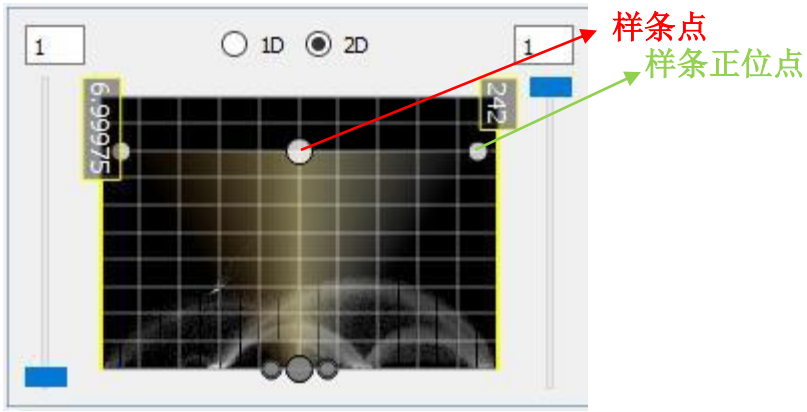

2.6.2
File
View
Image Size
Toggle
Plugins
Help
Transfer Function Editor
New
Refresh
Remove
Replace existing transfer functions at keyframes
Morph transfer functions during keyframe animation
Name
0
1
2
3
4
5
TF0
1
1D
2D
1
HiRes : image (normal+default+skip) : step(0
Bounds : 0-114, 0-107, 0-160
LoD(1) : Size : 115x108x161
Ready

将传递函数更改为如下所示 -

使用具有至少 2 个样条点的样条线创建传递函数。 每个样条点具有两个正位点，定义此时传递函数的宽度。 使用中间光标拖动样条点移动整个传递函数样条线。

左光标移动样条和样条正位点。

使用 Shift + 左鼠标将两个样条正位点移动到一起。



结果将出现以下画面 -

回到 Drishti Import 并练习调整样条点和样条正位点，同时使用直方图获取理想的渲染结果。

课程-02

数据信息面板简介和基础操作

打开 Drishti Render.

在 Drishti Render 中拖放 tooth.pvl.nc（在课程 01 中创建）。

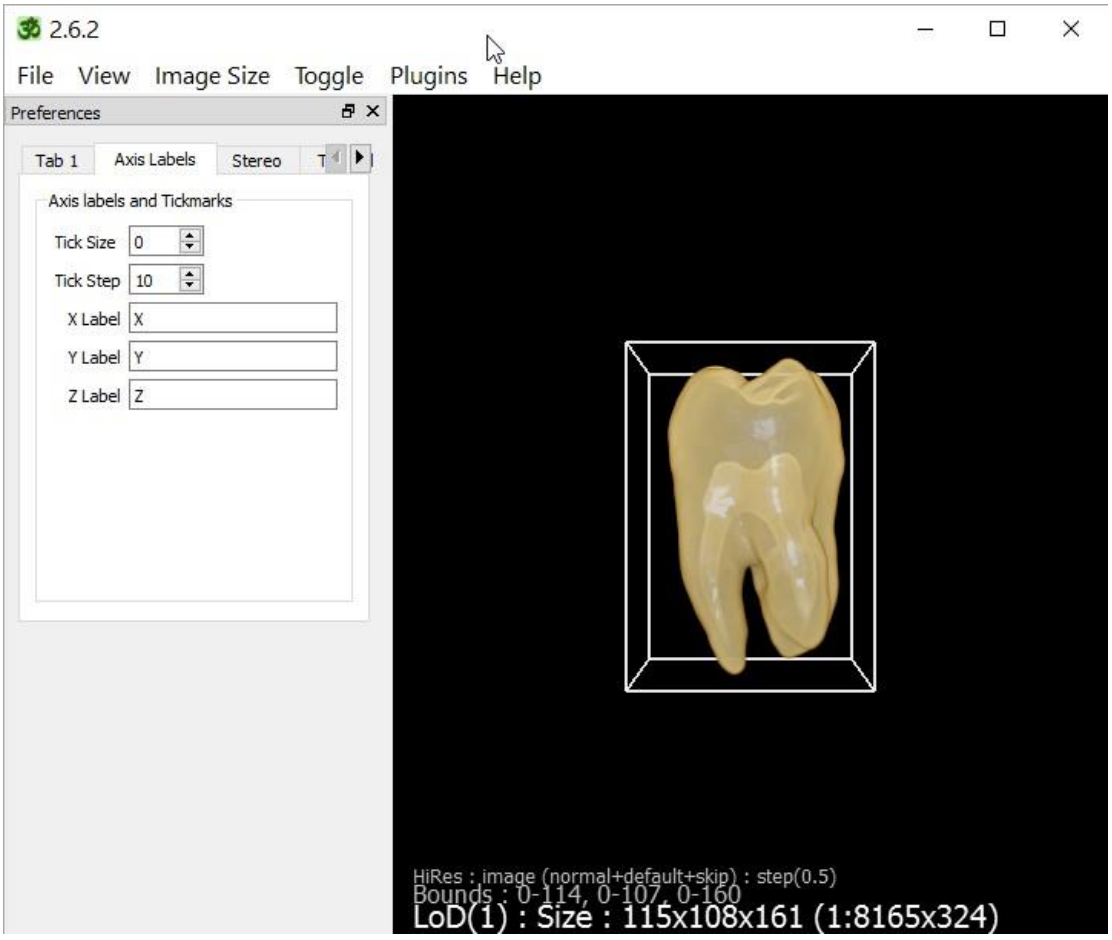
或者：文件 - >加载卷轴 - >加载 1 卷轴（Files->Load Volume->Load 1 Volume）

按 F2 切换到 HiRes 模式。

或者：Toggle->Hires Mode（F2）。

打开 Preferences panel : View->Preferences
打开首选项面板：查看 - >首选项（View->Preferences）

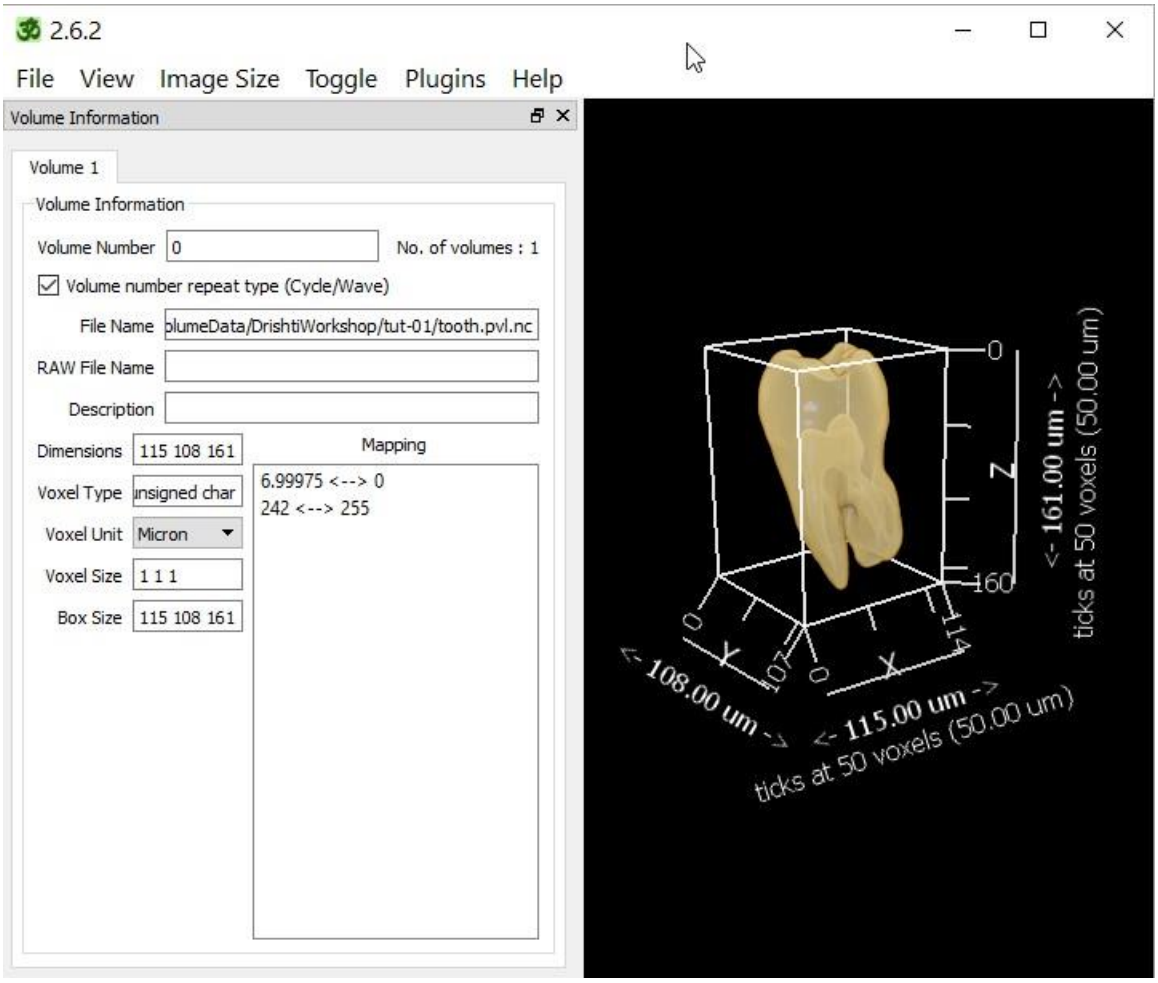
更改各种选项，并观察边界框上轴标签的更改。 可以用 b:Toggle-> Bounding Box（b）切换边界框的显示。



打开 Volume Information panel : View->Volume Information

打开卷轴信息面板：查看→卷轴信息（View->Volume Information）

更改体素单位 (voxel unit)，体素大小 (voxel size)和框大小 (Box size)，并观察渲染图像中的变化。



框的大小和体素的大小是相关的。

体素类型(Voxel Type) 是指每个体素-1（无符号字节）和 2（无符号）的字节数。

映射(Mapping) 显示有关从原始卷轴数据到处理卷轴映射的信息。

卷轴号(volume number) 表示当前卷轴号。 当处理时间序列数据（4D 卷轴）时，该数字可以大于 0。可以更改卷轴编号，以从 4D 卷轴加载不同的卷轴。

课程-03

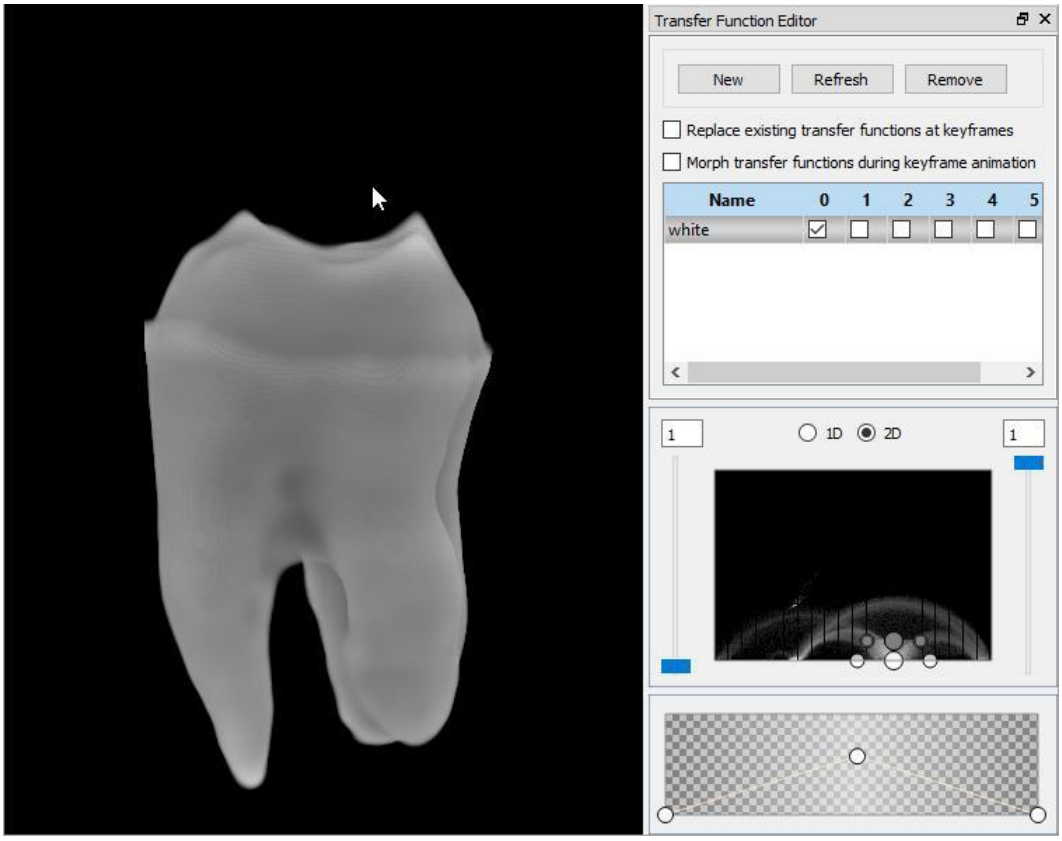
创建并运用传递函数过滤及更好的展示原始数据

打开 Drishti Render

在 Drishti Render 中加载 tooth.pvl.nc（导入课程 01 的创建结果）。
切换到 HiRes 模式。

删除现有的传递函数 TF0（使用删除 Remove 按钮）并使用新建 New 按钮创建新的函数。 通过双击传递函数名称将此传递函数的名称更改为白色。

改变白色传递函数，使其看起来像下面的图像。当键盘焦点位于绘图界面上时，可以分别按 g 和 o 切换传递函数绘图界面中的网格和映射信息。

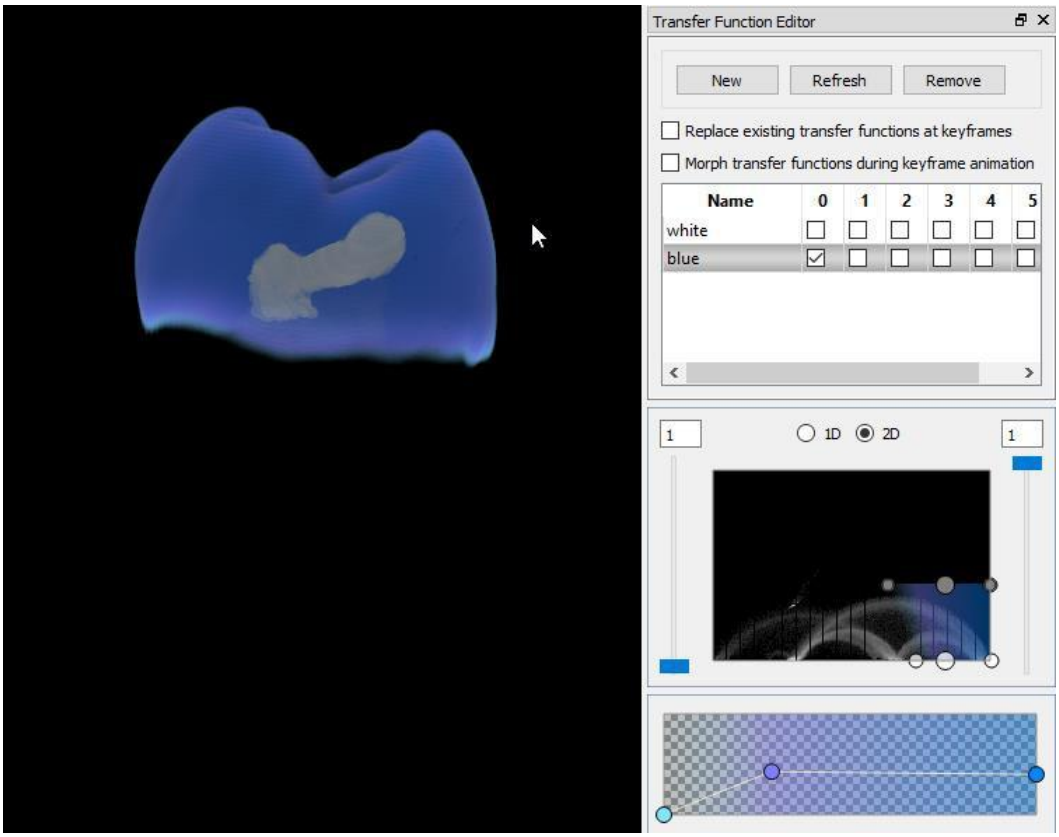


传递函数绘图界面

颜色不透明度梯度界面

使用 Shift +鼠标左键将样条正位点移动到一起。使用 Shift +鼠标左键限制沿垂直方向的样条点的运动。

添加另一个传递函数通过双击传递函数名称将此传递函数的名称更改为蓝色。更改传递函数形状（中间面板）和颜色和不透明度（底部面板），使图像如下图所示。当鼠标光标位于底部面板（颜色渐变面板）中时，按空格键可显示颜色弹出式菜单。



传递函数绘图界面

颜色不透明度梯度界面

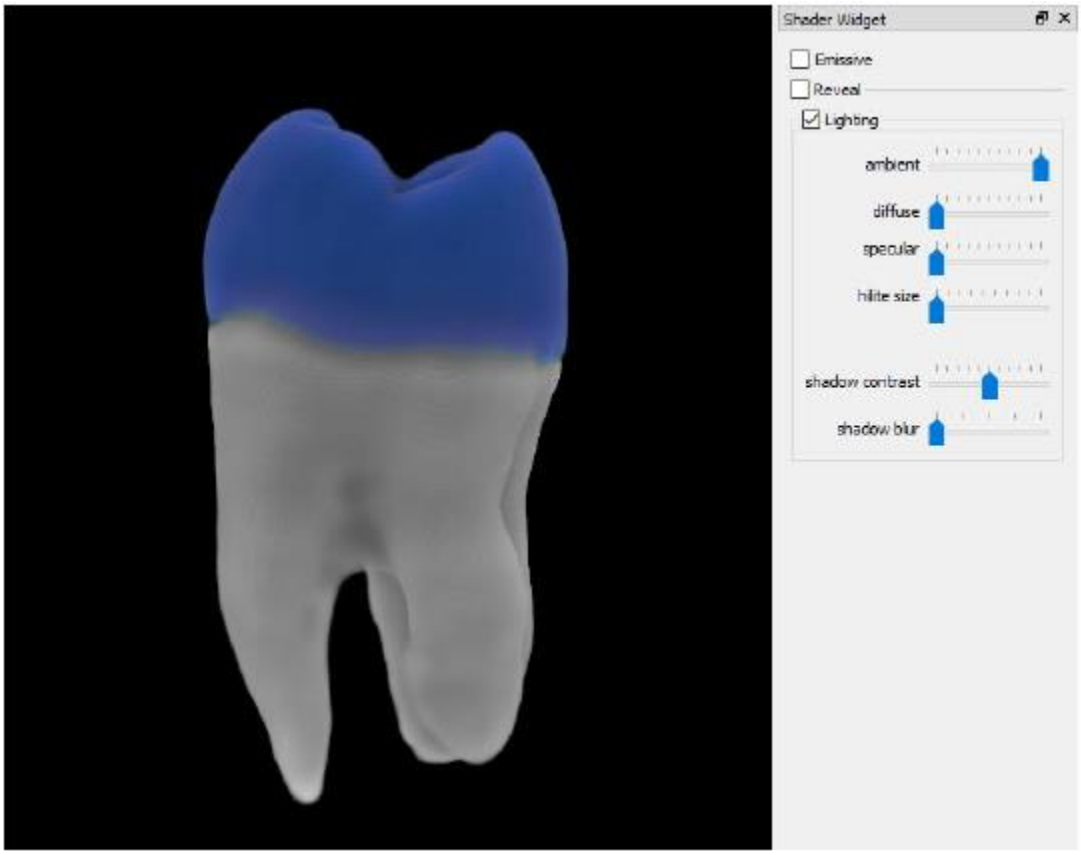
通过检查第 0 列（设置 0）下的方框可以激活/禁用传递函数。上述图像仅显示蓝色传递函数，白色函数被禁用。

当鼠标光标位于传递函数编辑器面板中时，按 Ctrl + H 显示传递函数管理器的帮助。

关闭传递函数编辑器。

打开 Shader Widget：View-> Shader Widget。

将 Specular 值设置为零，以减少蓝色牙质上的白色斑点。



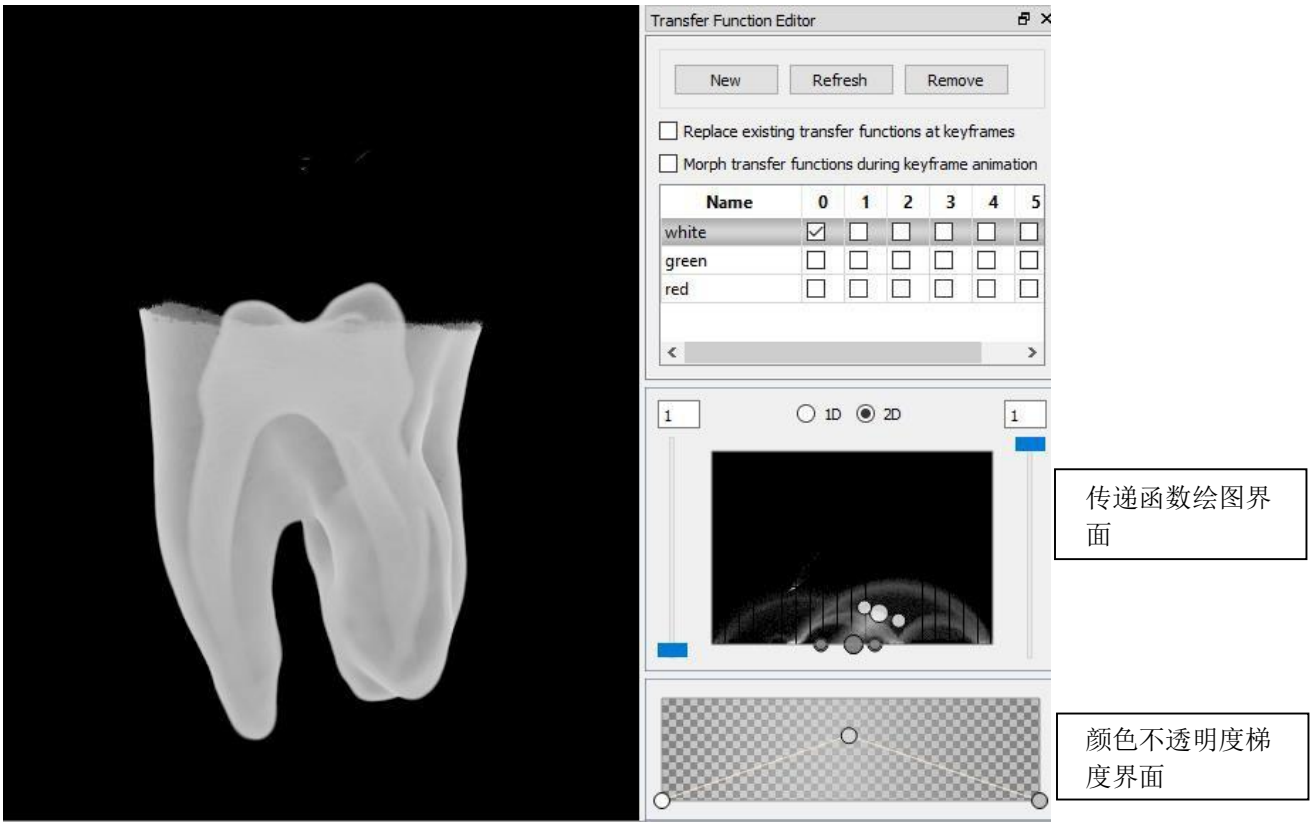
课程-04
传递函数进阶

打开 Drishti Render

加载 tooth.pvl.nc（课程 01 创建结果）。

切换到 HiRes 模式，然后添加三个传递函数，并命名为白

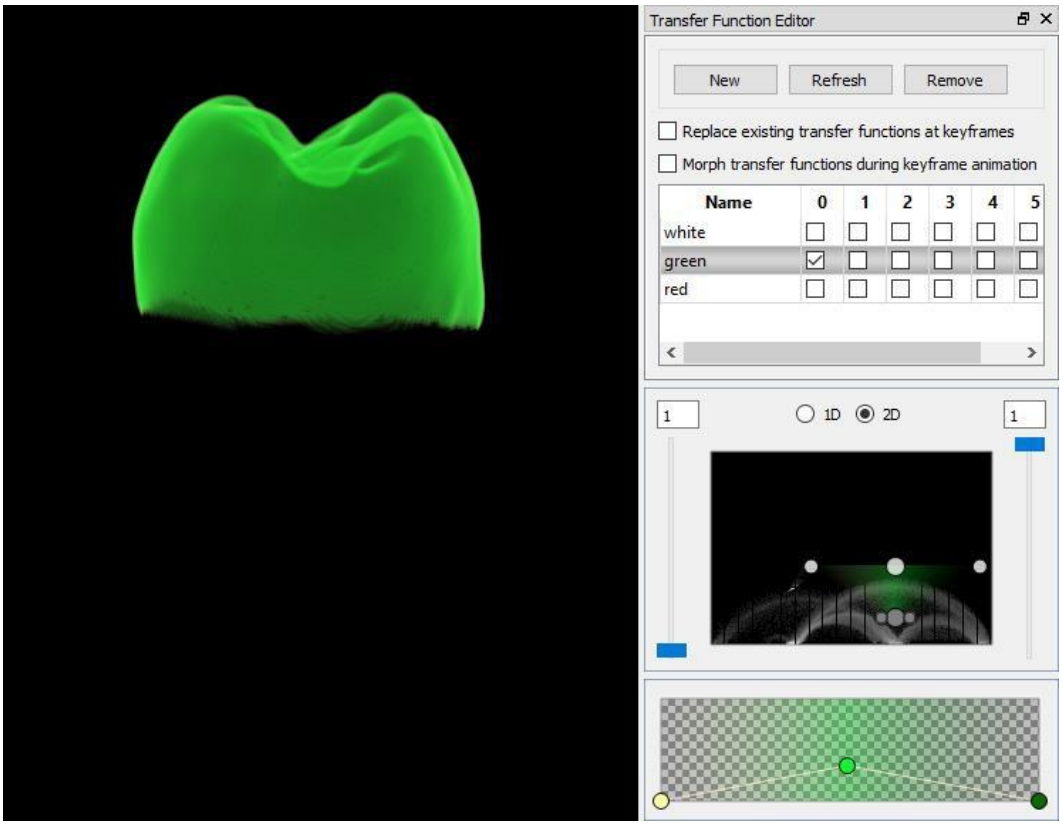
色、绿色和红色，如下所示。

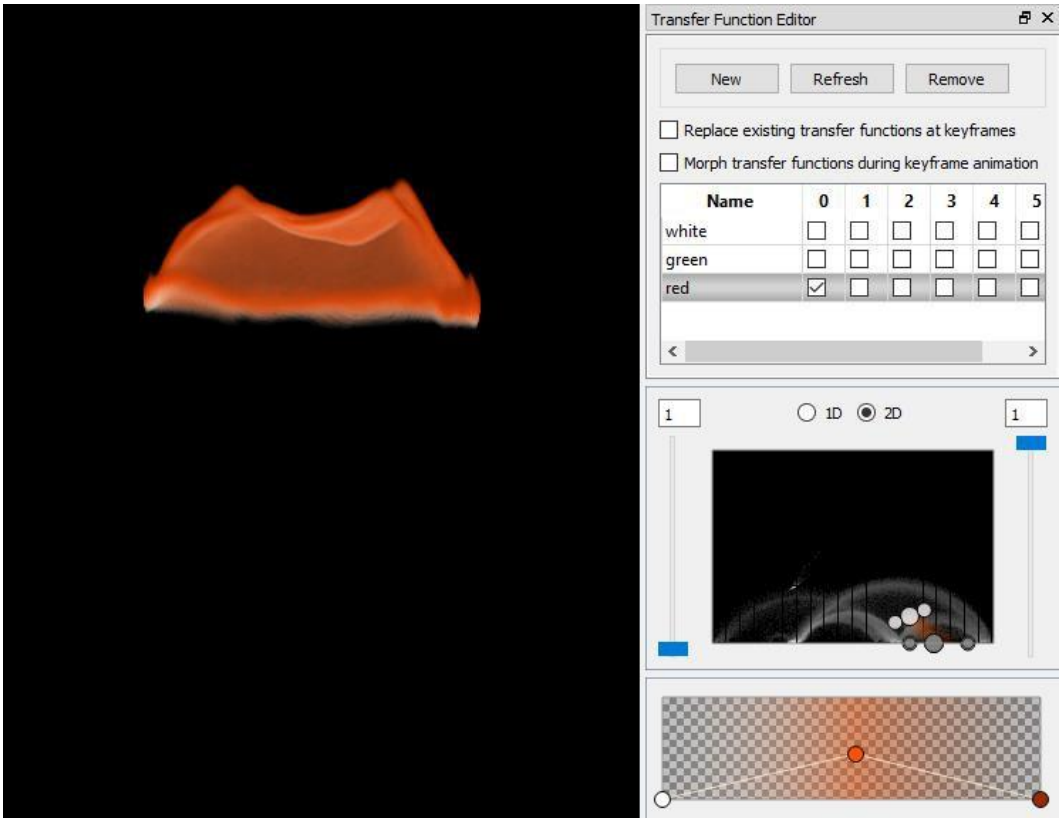


使用 Ctrl + Left 鼠标旋转样条正位点。

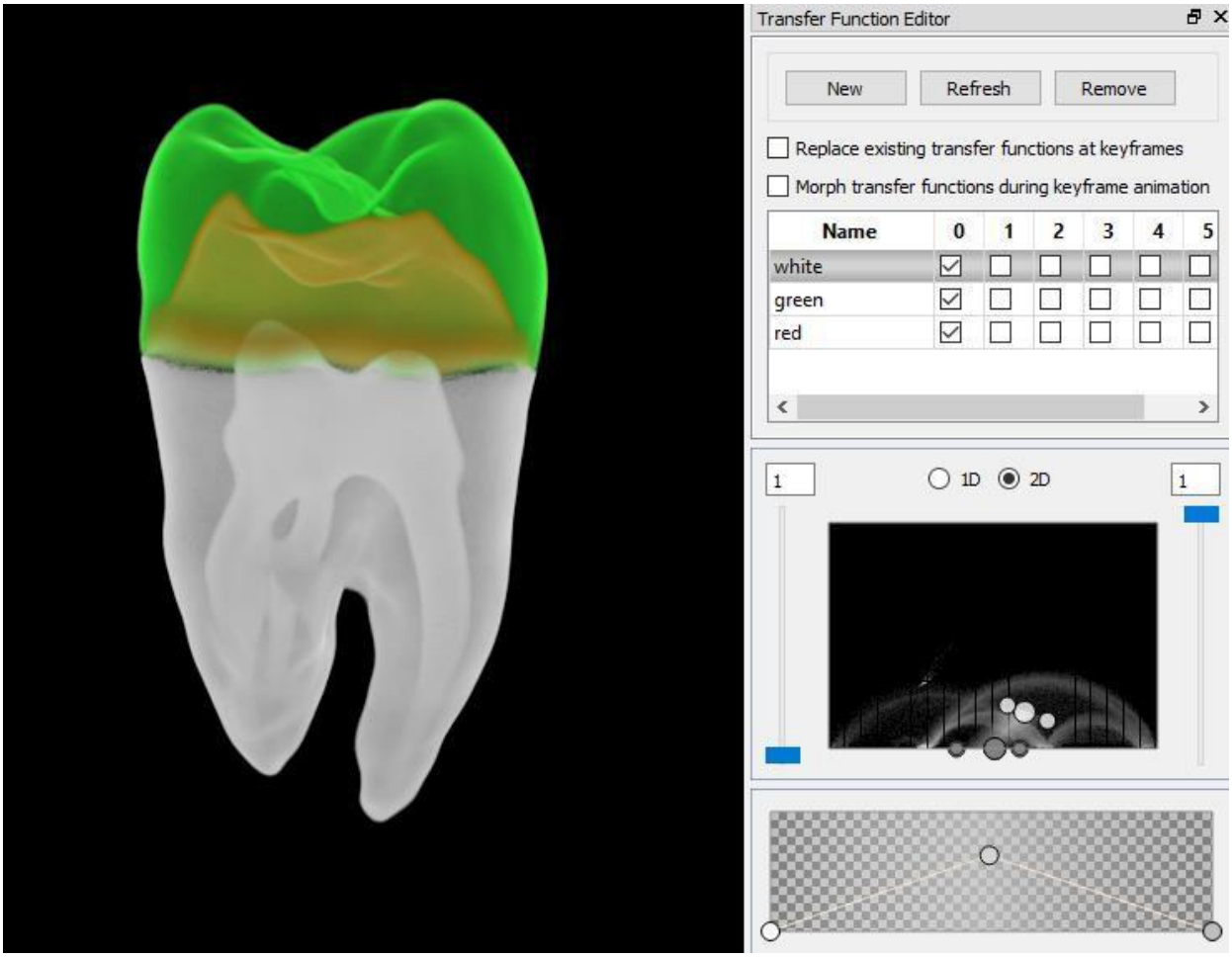
这些传递函数捕获数据中的边界层。 在课程-03 我们分类了数据中的同源区域。2D 直方图中的拱形通常表示两个同源区域之间的边界。

默认显示为 2D 直方图。
选择传递函数绘图界面上的 1D 开关，显示数据的 1D 体素强度直方图。


Transfer Function Editor
New
Refresh
Remove
Replace existing transfer functions at keyframes
Morph transfer functions during keyframe animation
Name
0
1
2
3
4
5
white
green
red
1
1D
2D
1


Transfer Function Editor
New
Refresh
Remove
Replace existing transfer functions at keyframes
Morph transfer functions during keyframe animation
Name
0
1
2
3
4
5
white
green
red
1
1D
2D
1

现在打开所有传递函数以显示三个区域。



关闭传递函数编辑器 （Transfer Function Editor）并打开 Shader Widget。

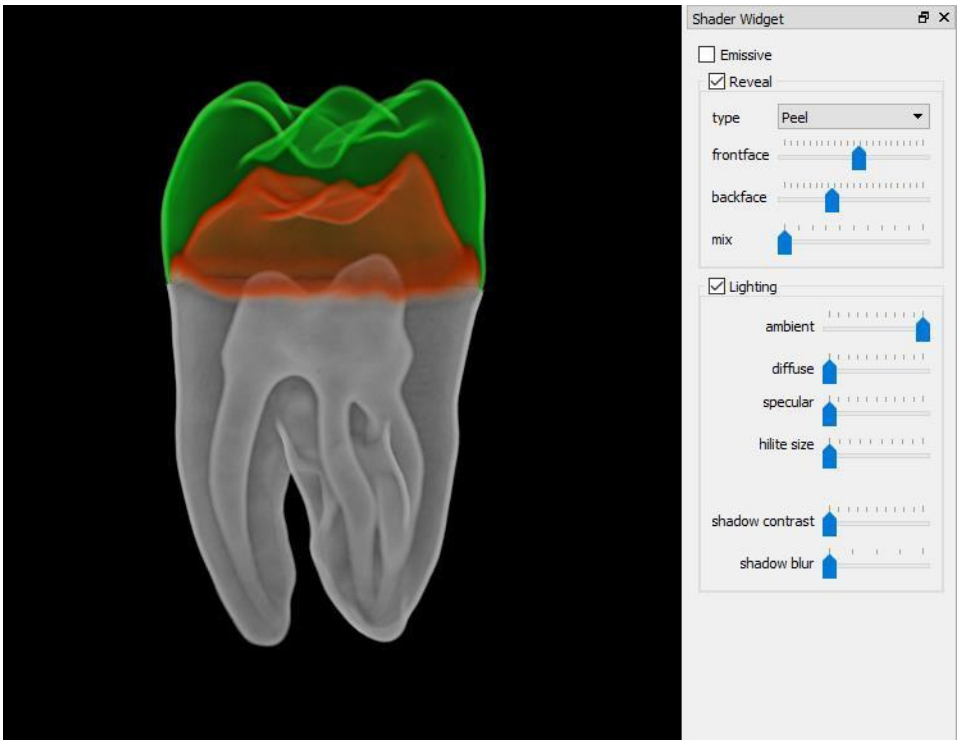
按 1 进行屏幕阴影渲染：Toggle-> Shadow Render（1）

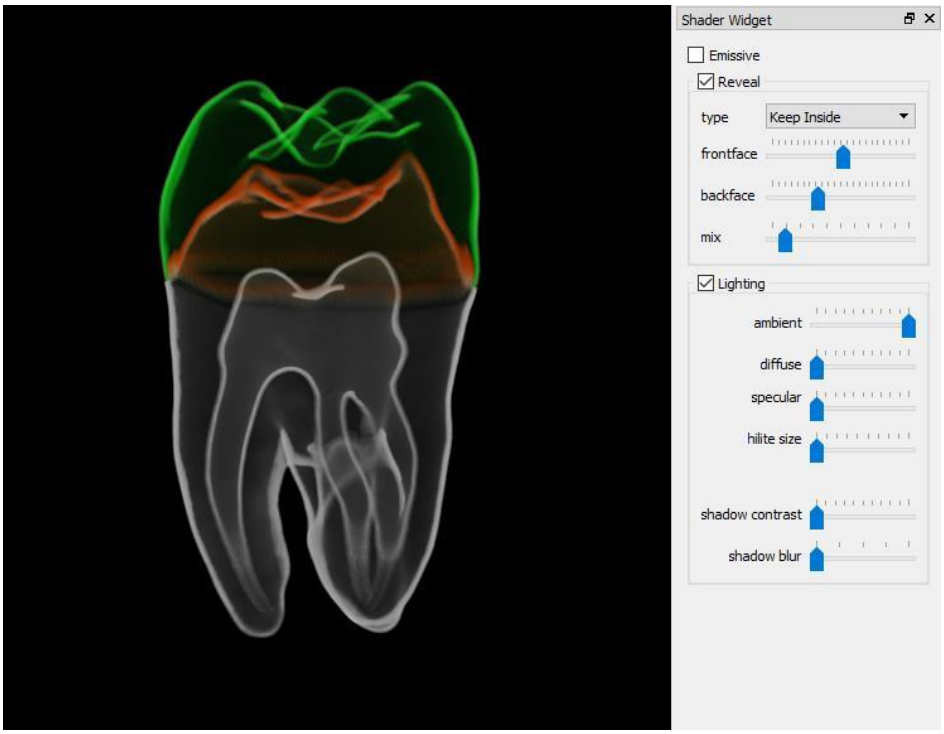
在 Shader Widget 中打开显示(Reveal)。有两种类型的显示 - 剥离和保持内部 Peel and Keep Inside。在更改滑块的前面、背面和混合时，请注意渲染图像的变化。

按 Ctrl + H 来阅读 Shader Widget 的帮助。

尝试获取以下渲染。

使用阴影对比度 (shadow contrast)和阴影模糊设置(shadow blur)来查看对渲染的影响。





课程-05

制作简单旋转动画

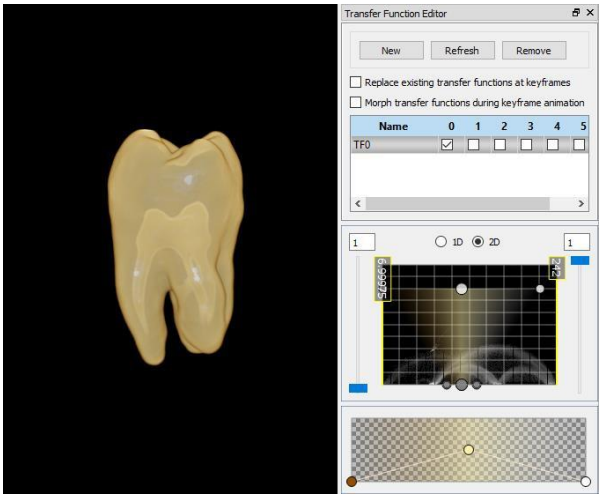
打开 Drishti Render

在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中的

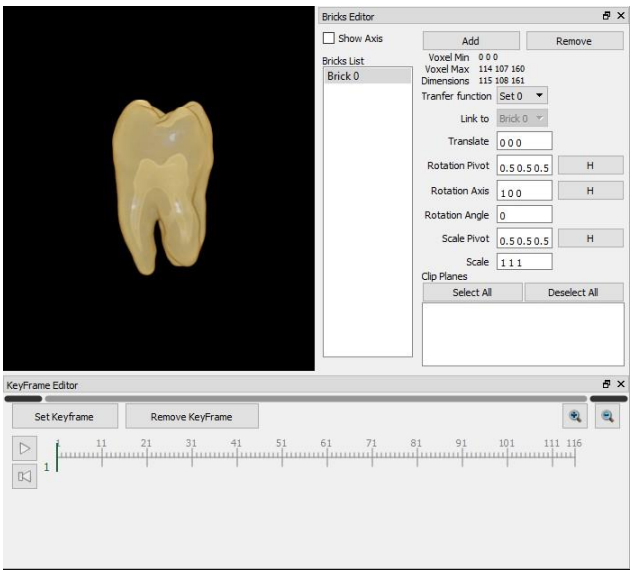
创建结果）。

切换到 HiRes 模式。

选择合适的传递函数来显示牙齿。



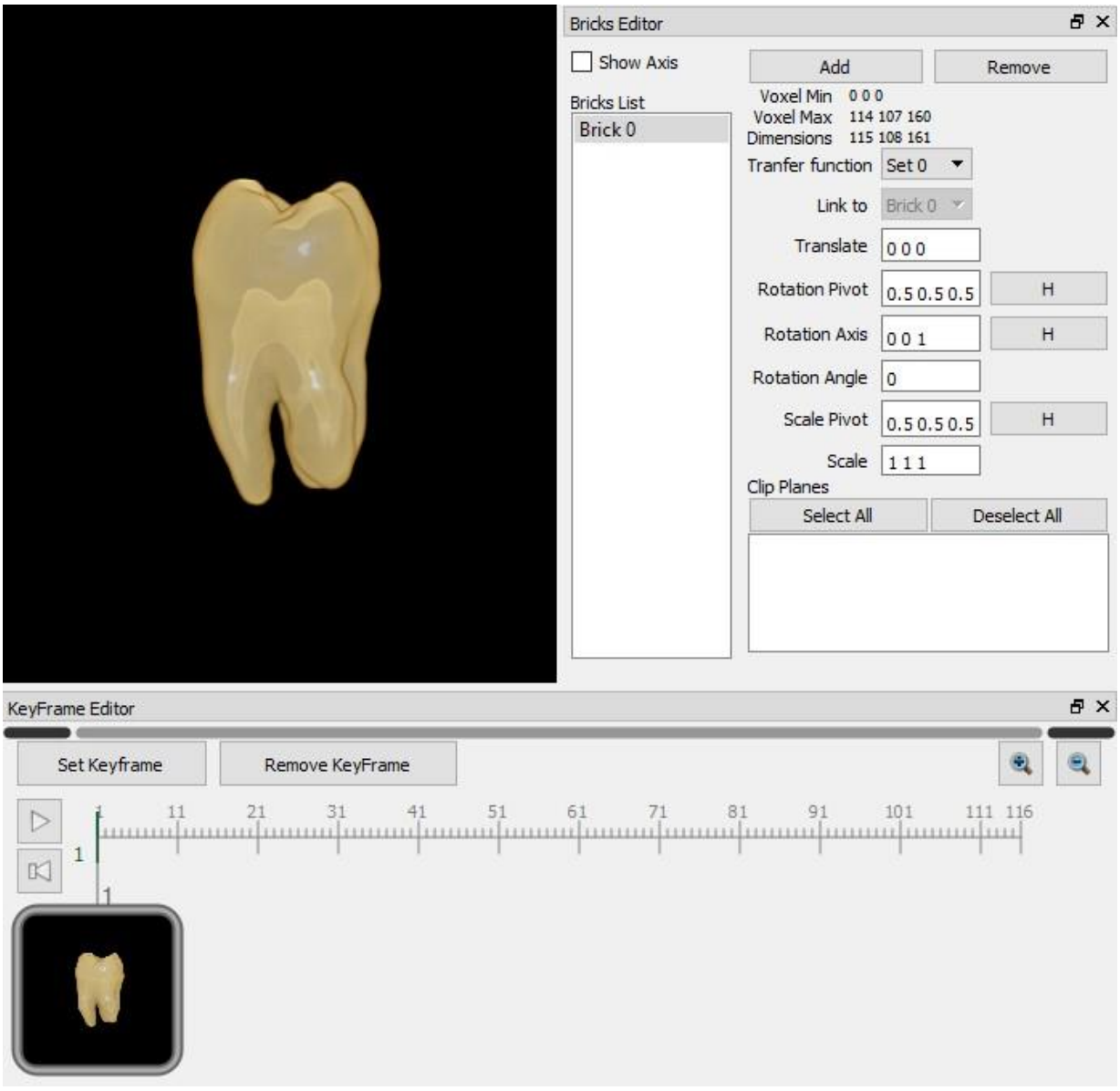
关闭传递函数编辑器 (Transfer Function Editor)。打开 Keyframe Editor 和 Bricks Editor.



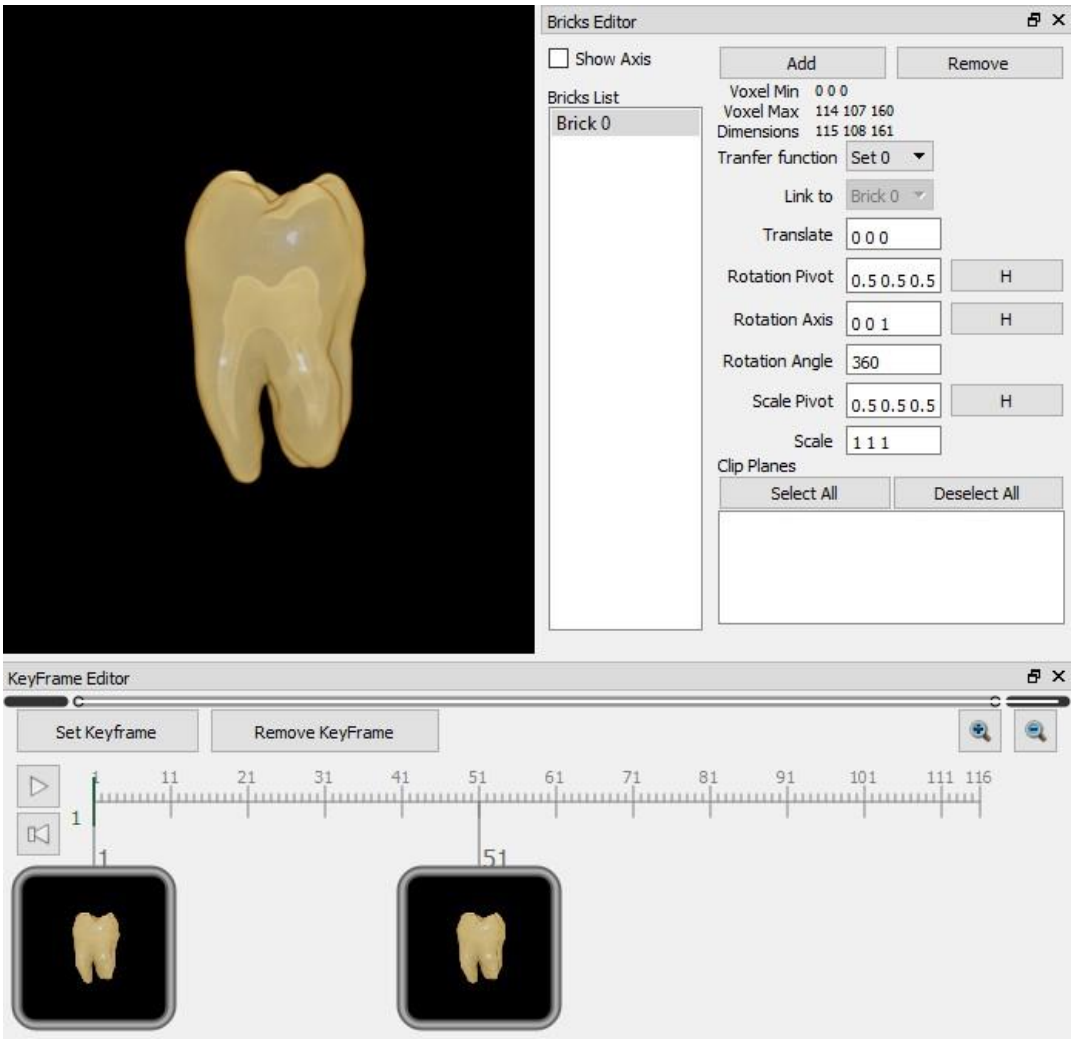
点击 Bricks Editor，然后按 Ctrl + H 显示帮助。
点击 Keyframe Editor，然后按 Ctrl + H 寻求帮助。

在 Bricks Editor 中选择显示轴(Show Axis)， 这将显示旋转轴。 将旋转轴从默认值 1 0 0（以 x 轴为参照做旋转）更改为 0 0 1（以 z 轴为参照做旋转）。

在 Keyframe Editor 中按设置关键帧（Set Keyframe）。 此操作将在 Keyframe Editor 中设置关键帧的时间表。



在框号 51 上单击鼠标左键。将 Bricks Editor 中的旋转角度(Rotation Angle)更改为 360，并在 Keyframe Editor 中设置关键帧(Set Keyframe)。



您现在已经设置了两个关键帧。这些关键帧可以移动。

内插关键帧之间的帧的各种参数，例如旋转角度(rotation angle)，摄像机参数(camera position/orientation)等。

在 Keyframe Editor 中按播放按钮(Play)，观看制作的牙齿旋转的动画。

可以在更改相机位置/方向，放大/缩小，以及更改 Bricks Editor 中的参数后加入更多的关键帧并查看其对动画的影响。

使用动画影片选项保存影片（仅限 Windows）保存：文件 - >保存影片 File->Save Movie。50 帧只有 2 秒的动画（假设每秒 25 帧）

课程-06
剪切平面入门

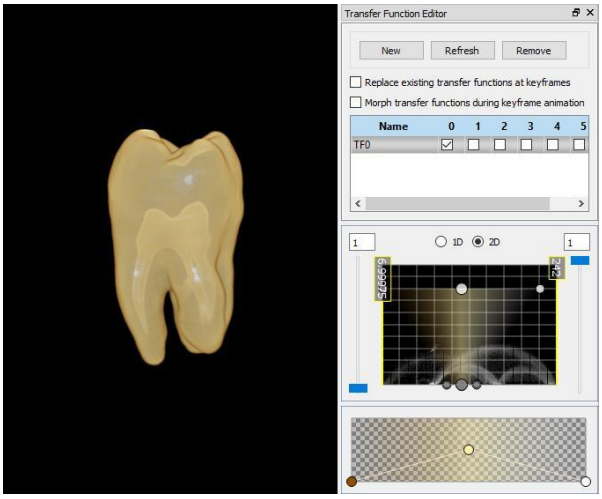
打开 Drishti Render.

在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中的

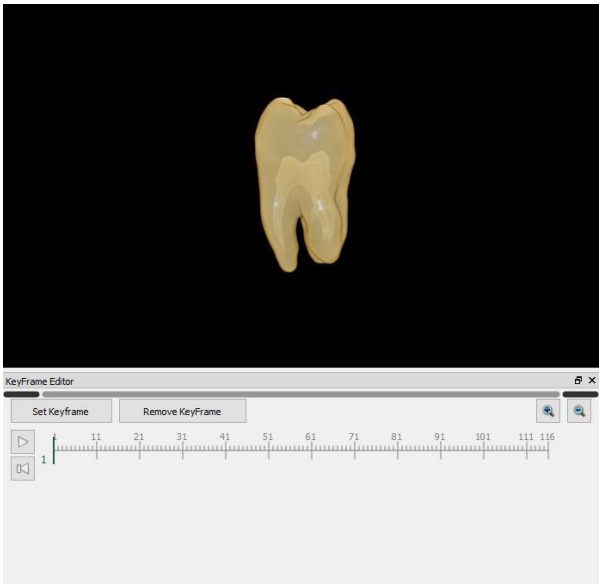
创建结果）。

切换到 HiRes 模式。

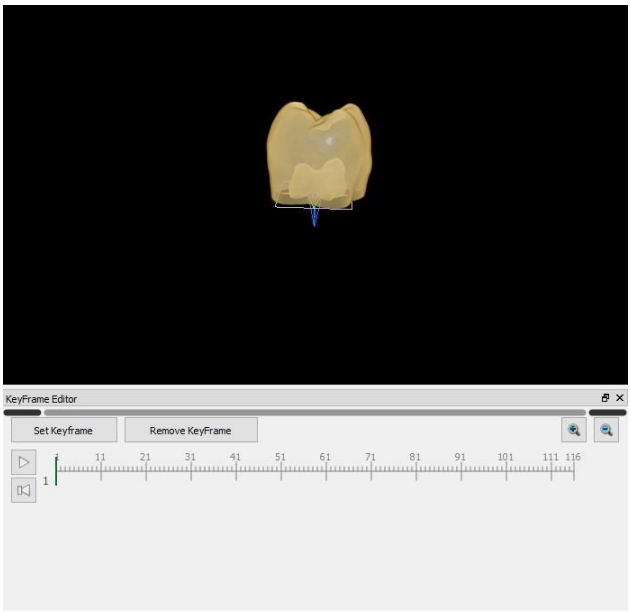
选择合适的传递函数来显示牙齿。



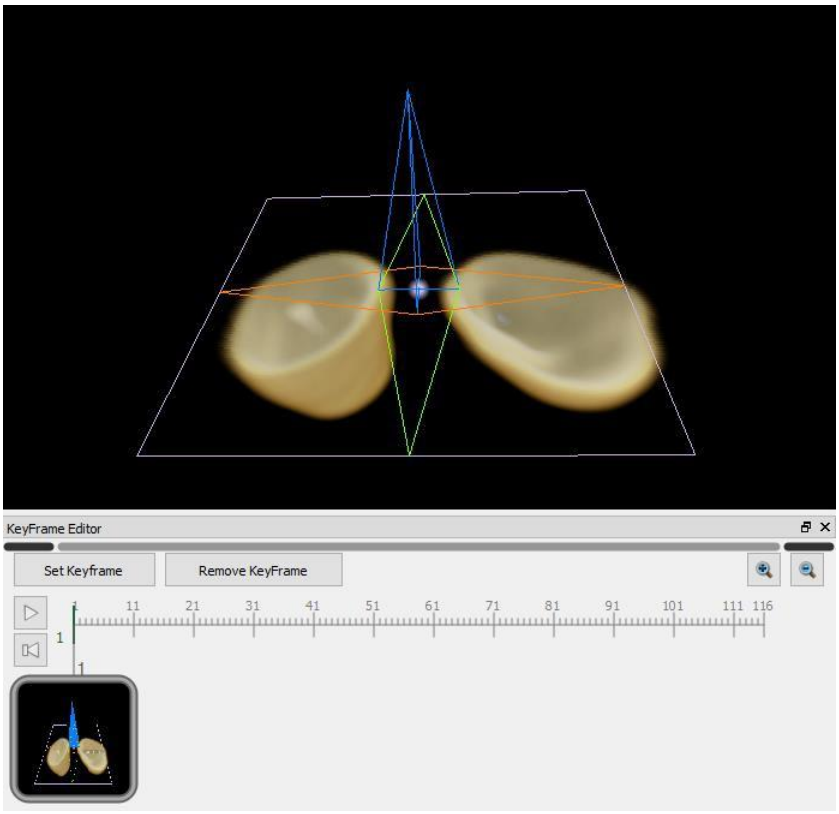
选择 传递函数编辑器。打开 Keyframe Editor。



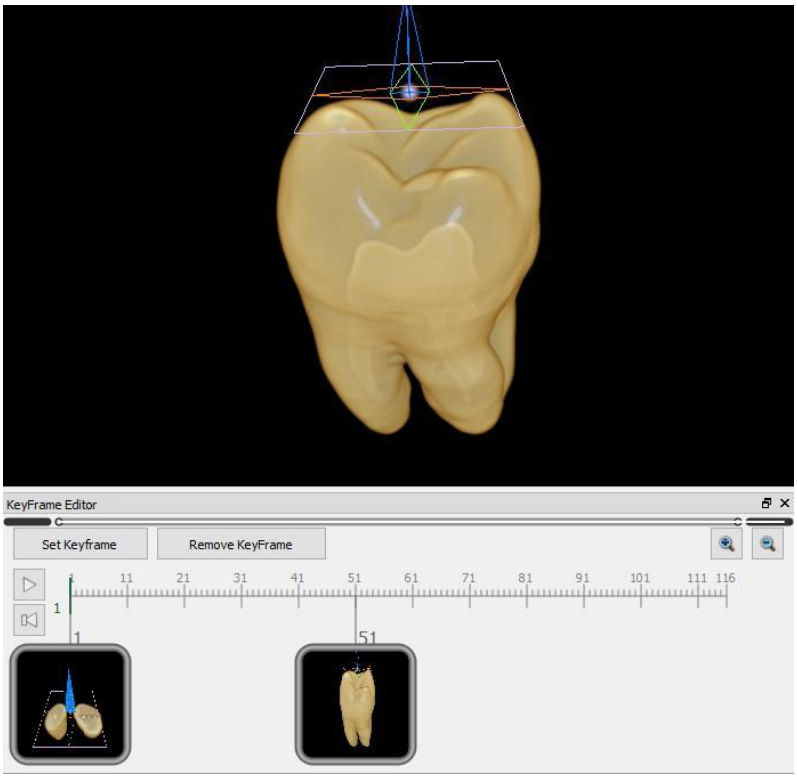
按 c 添加裁剪平面。悬停在裁剪平面(clipping plane)上，然后按空格键显示各种参数，并帮助裁剪平面。



操纵裁剪平面，使其现在位于牙齿底部附近。在 Keyframe Editor 中设置关键帧(Set Keyframe)。



现在移动裁剪平面，使其位于牙齿的顶部。 在 Keyframe Editor 中的第 51 帧设置关键帧。



播放动画，看到剪裁平面露出牙齿。

在这个动画中，裁剪平面从底部移动到顶部。如果不想显示剪切平面，则保存关键帧的时候应该隐藏剪切平面。

按 v 更改剪切平面的可见性：Toggle-> Visibility。

通过点击关键帧 1 选择关键帧 1。将鼠标移动到渲染窗口中，然后按 v 隐藏剪切平面。按设置关键帧 Set Keyframe 将关键帧重设为 1。

对第 51 帧的关键帧重复以上步骤。

一旦选择关键帧，剪裁的小部件(Clip widgets)将变得无法抓取，以确保用户不会无意中更改设置。要使小部件可抓取，应按下切换开关 g（Toggle-> Mouse Grab）。

保存动画影像。

使用 Bricks Widget 尝试使用旋转组合。
尝试多个裁剪平面。

课程-07
剪切平面对话框和改变传递函数参数入门

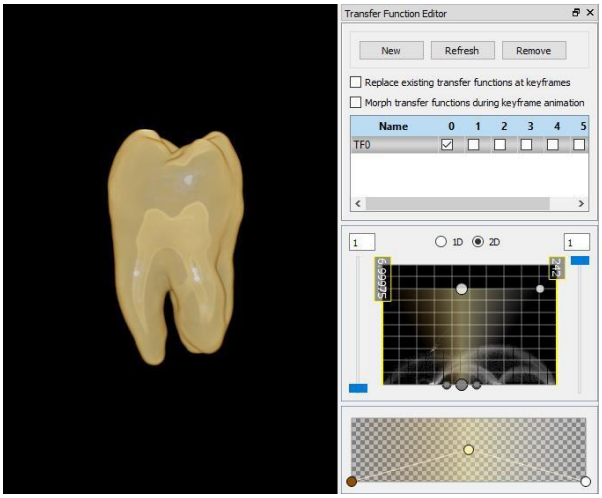
打开 Drishti Render.

在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中

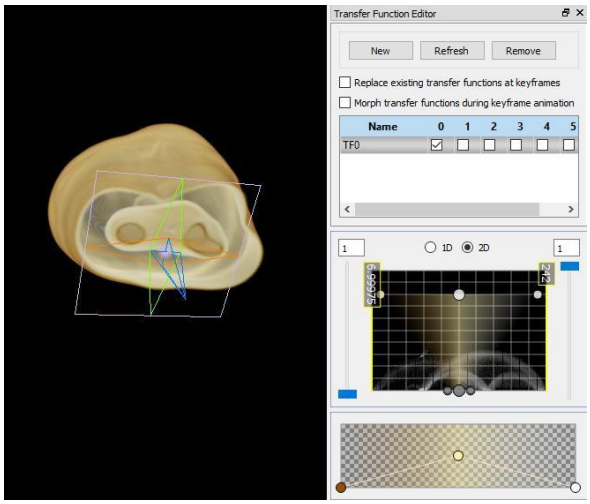
的创建结果）。

切换到 HiRes 模式。

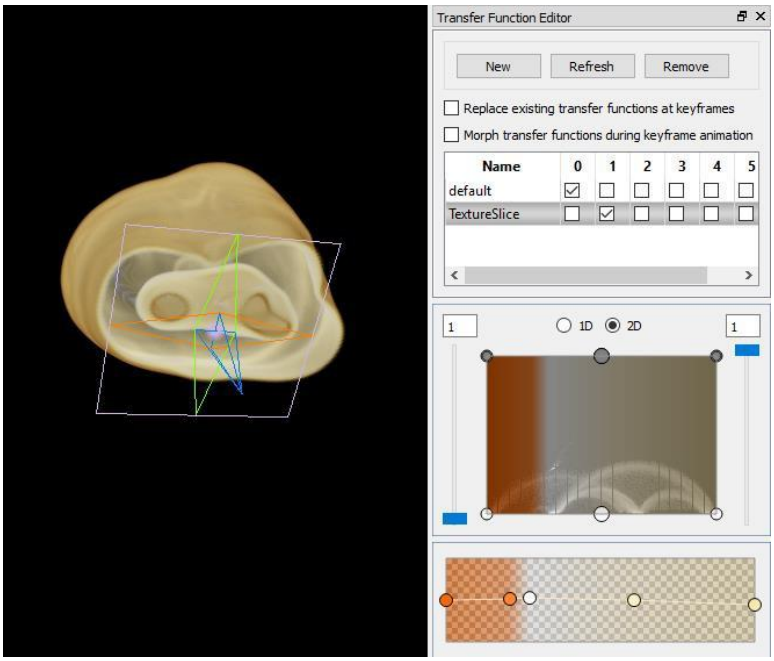
选择合适的传递函数来显示牙齿。



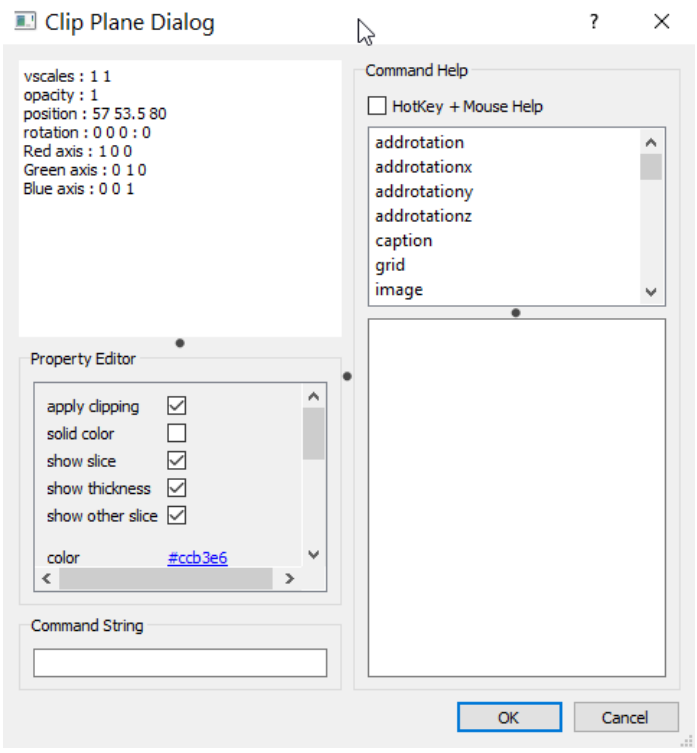
按 c 添加剪裁平面(clipping plane)。



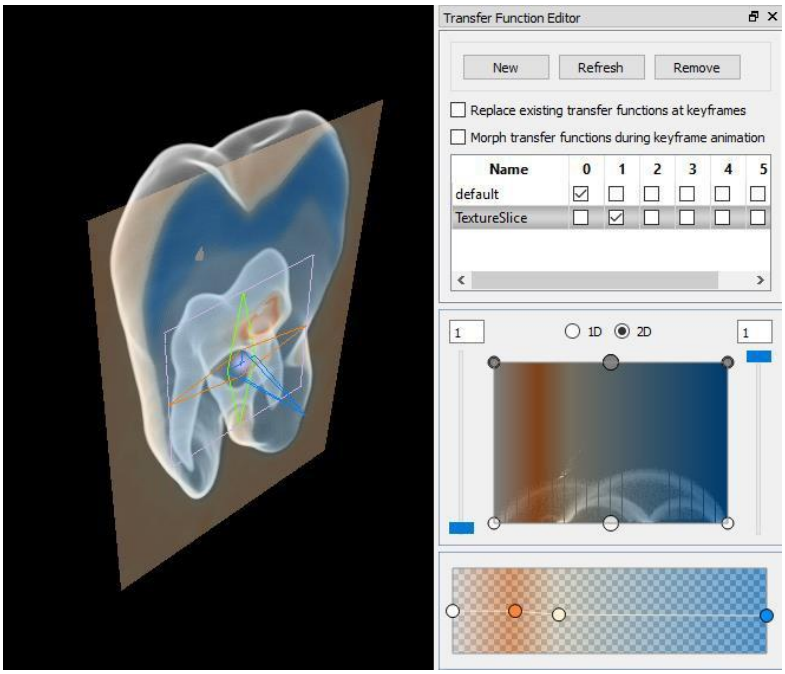
在传递函数编辑器(Transfer Function Editor)中使用新建按钮(New)添加新的合适的传递函数。 取消选择该传递函数的复选框 0，然后选择复选框 1。这意味着这个新的传递函数被添加到传递函数集 1。我们要将这个传递函数用作剪辑平面上的切片纹理。



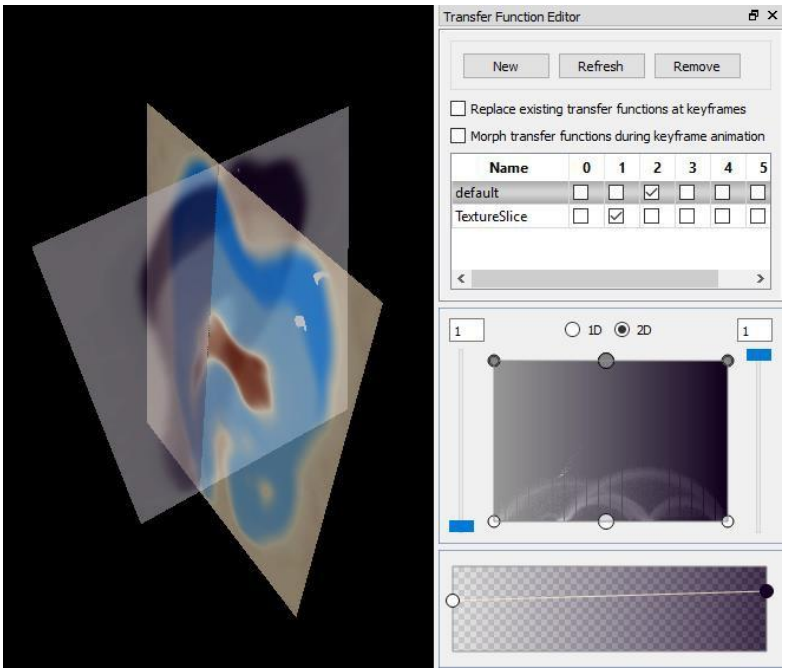
将鼠标悬停在裁剪平面上，然后按空格键弹出“剪辑平面 Clip Plane 对话框。



转到 tfset 参数并将其设置为 1（默认值为-1）。 tfset 参数指定传递函数用于纹理化 (texturing) 剪切平面切片的集合。 -1 的值取消激活剪切平面的纹理。 剪切平面可用于显示通过卷轴的纹理切片。



添加另一个剪切平面并将其 tfset 参数设置为 2。添加另一个传递函数并将其添加到 Set 2。关闭 Set 0 中的传递函数（Set 0 传递函数用于卷轴数据的 3D 渲染）。



尝试使用在课程 06 中学到的技术制作化纹理切片的动画。

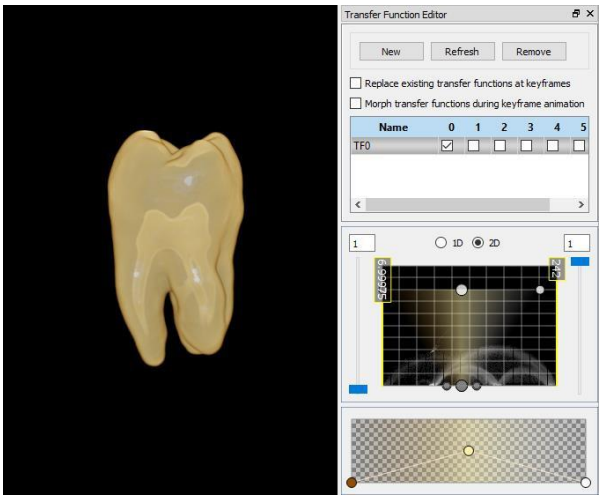
课程-08
剪切平面进阶（视角参数）

打开 Drishti Render.

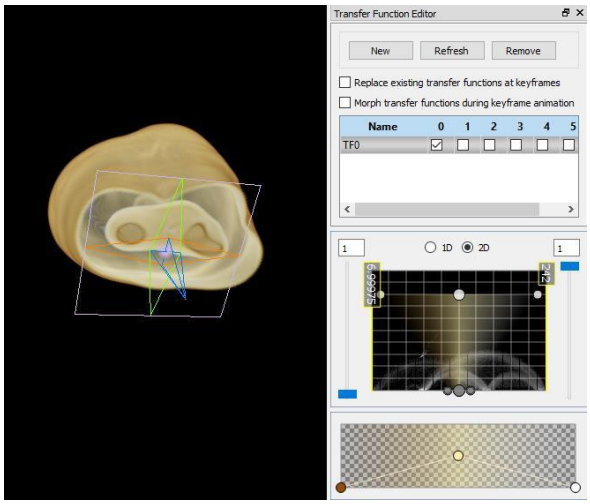
在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中的创建结果）。

切换到 HiRes 模式。

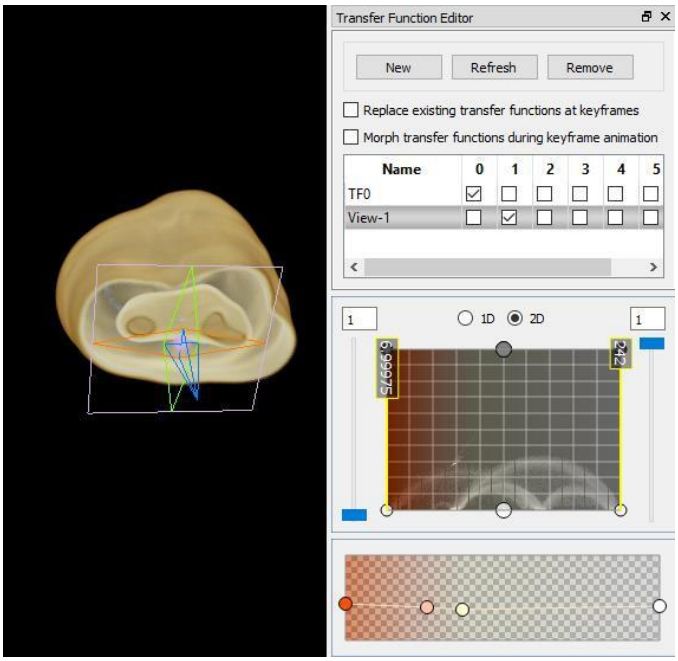
选择合适的传递函数来显示牙齿。



按 c 添加剪裁平面

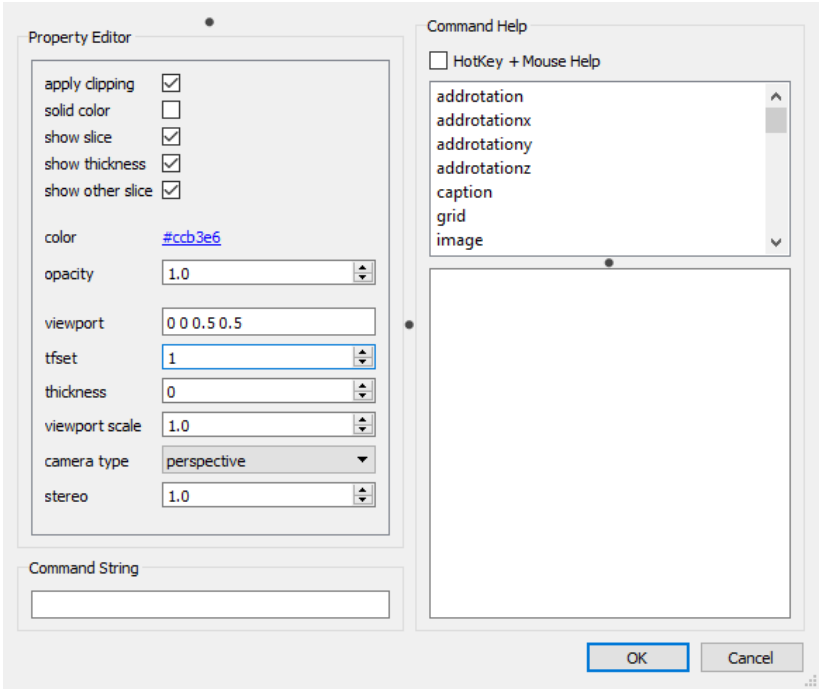


在传递函数编辑器中使用新建按钮添加新的合适的传递函数。取消选择该传递函数的复选框 0，然后选择复选框 1。这意味着这个新的传递函数被添加到传递函数集 1。我们要将这个传递函数用作剪辑平面上的一个切片。



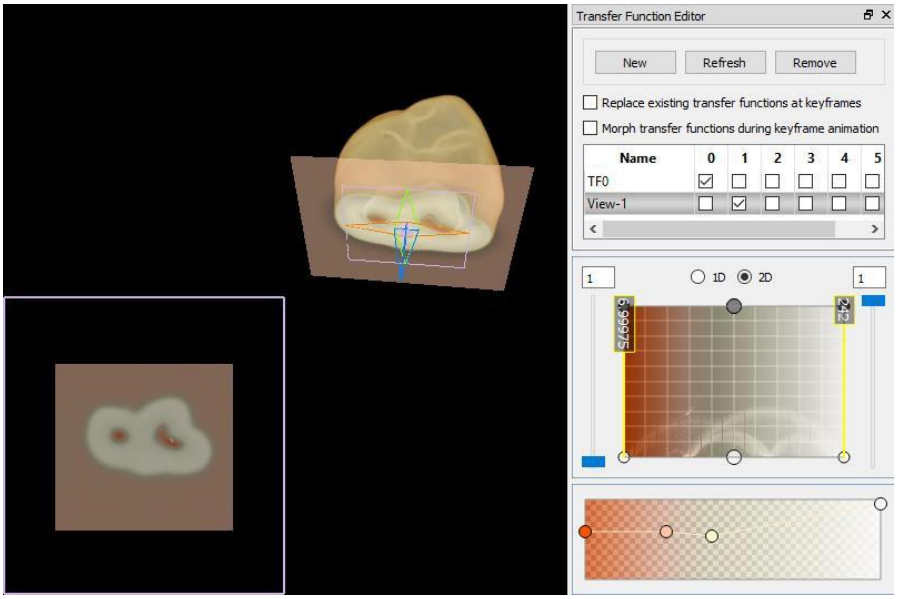
将鼠标悬停在裁剪平面上，然后按空格键弹出“剪辑平面 Clip Plane”对话框。 找到 tfset 参数，并将其设置为 1（默认值为-1），如有疑问请参见课程 07 中的操作。

另外将视口参数设置为 0 0 0.5 0.5 - 在左下角设置视口。

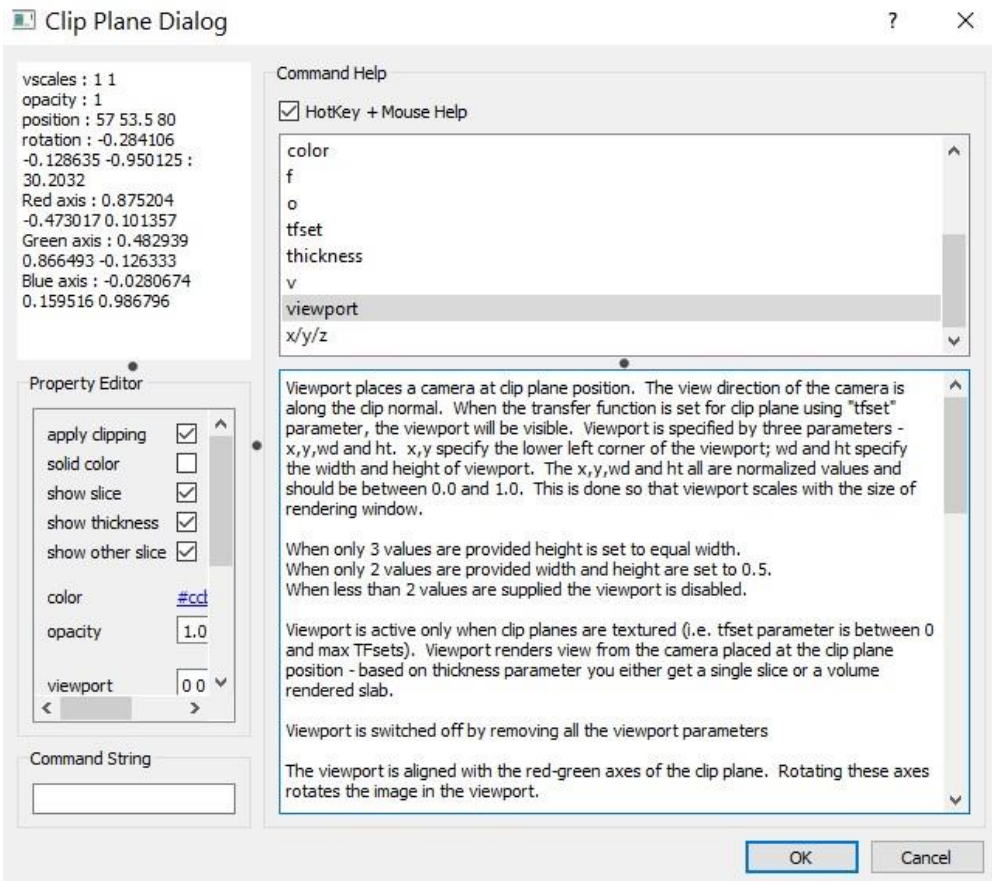


剪切平面可以用作虚拟相机，并通过卷轴纹理切片。剪切平面可以从其视口操纵。

视窗内的平面旋转通常被锁定，以防止意外的方向改变。 按 R（大写字母 r）切换此旋转锁定。平移和缩放可用于视口。

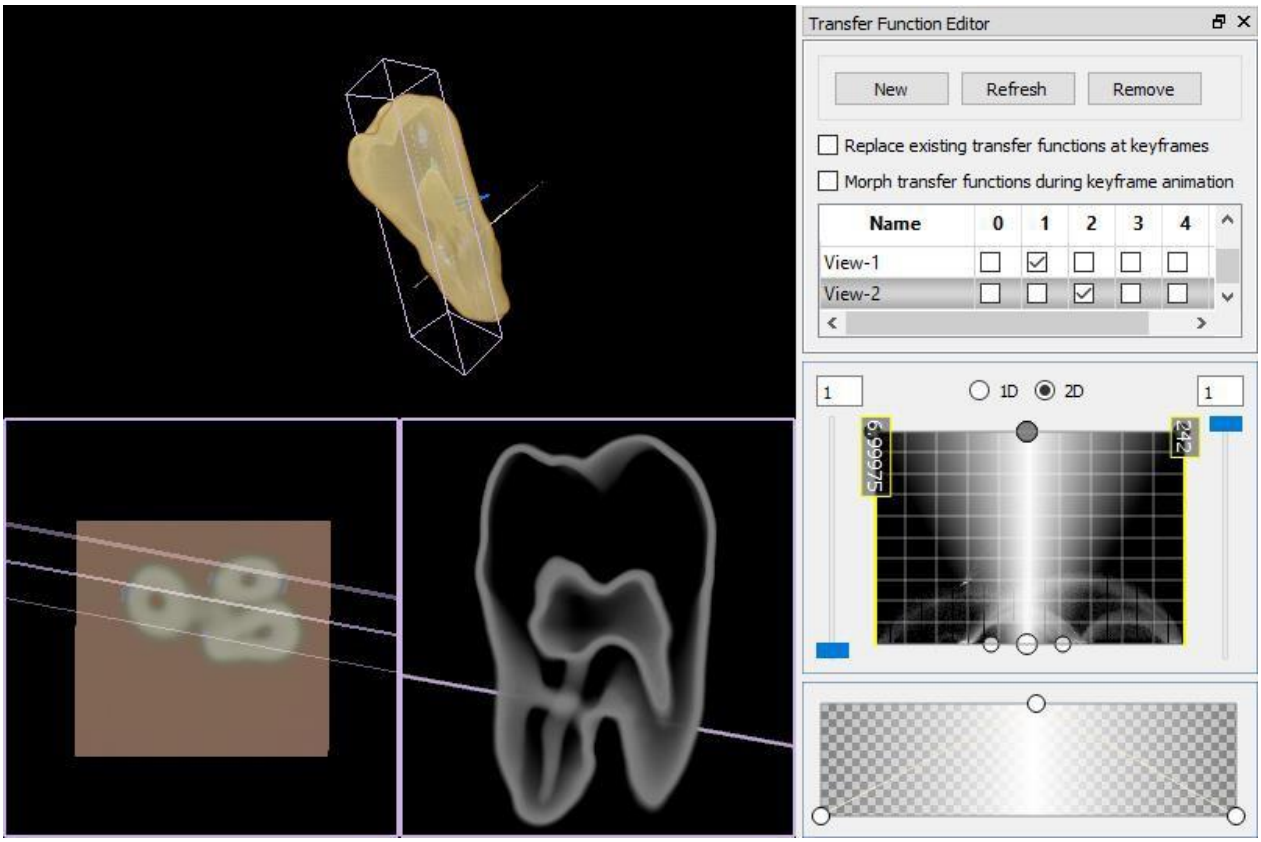


如果旋转被锁定，当尝试在视口内旋转时，MOP DRV（视口中的禁用旋转）将显示在顶部。在视口内按下 SPACE 会弹出该裁剪平面的对话框。 在 HotKey +鼠标帮助下，您将在帮助菜单中找到对视口 (Viewport) 的详细解释。



添加另一个剪切平面并将其 tfset 参数设置为 2。添加另一个传递函数并将其添加到 Set 2。将视口参数设置为 0.5 0 0.5 0.5 - 在右下角设置视口。

关闭对剪切平面应用裁剪(apply clipping) - 这将对卷轴禁用裁剪功能。 对于第二个剪切平面，将厚度参数(thickness parameter)增加为 15 - 这将从视口中的卷轴显示 15 个切片。



熟悉各种参数并观察其对渲染的影响并尝试动画化纹

理切片。

注意：

视口边框颜色由颜色参数控制。

显示切片 - 显示纹理切片

显示厚度 - 显示虚拟相机平截头体

显示其他切片 - 在视口尺度中显示其他切片 - 缩放因子

课程-09
简单的物体旋转–使用 Bricks Editor 选定旋转轴

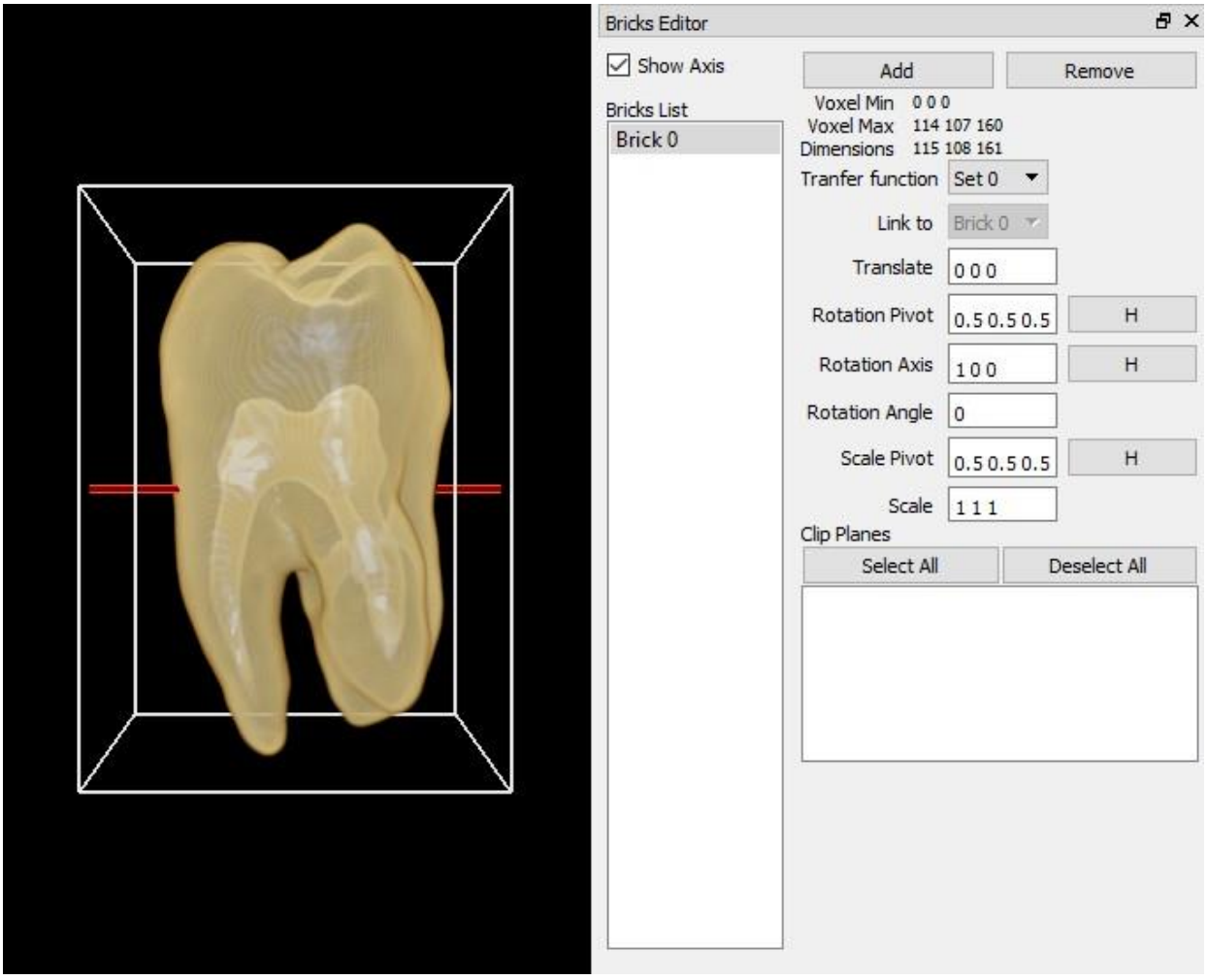
打开 Drishti Render.

在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中的

创建结果）。

切换到 HiRes 模式。

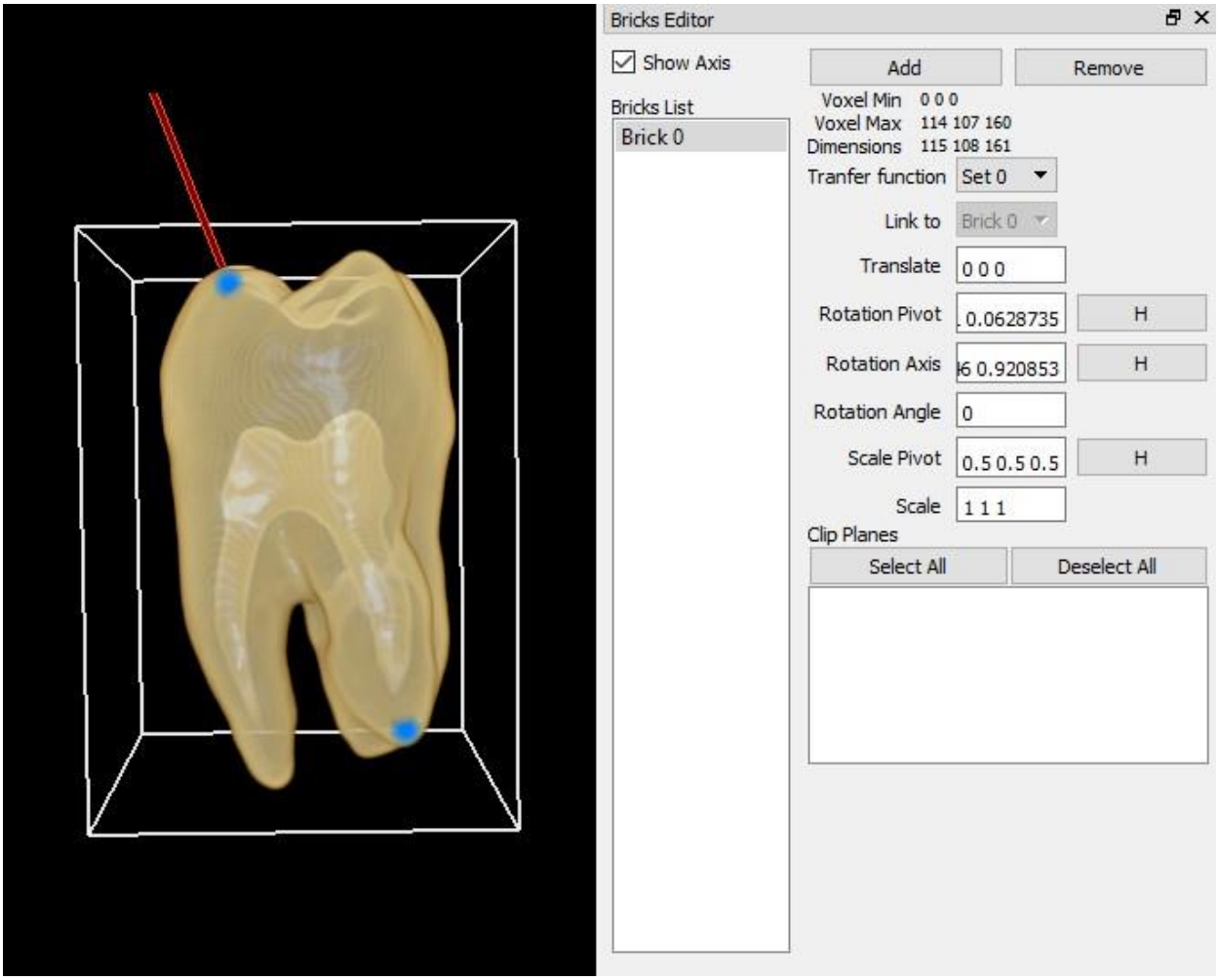
打开 Bricks Editor 并检查显示轴。显示旋转轴为红色。



默认的旋转枢轴(Rotation Pivot)是 0.5 0.5 0.5 - 方框的中心。

通过 Shift +左键在卷轴上添加一个点。 该点将显示为蓝点。在旋转轴旁边按 H 按钮 (命中点- hit point)，命中点选取以后将以其为旋转枢轴。旋转轴现在将以旋转枢轴为中心。

在卷轴上添加另一个点。在旋转轴旁边按 H 按钮，以改变旋转轴。 旋转轴将自动对齐，并通过当前添加的点。也可以理解为两点确定一条直线，即旋转轴。



砖盒 (Brick boxes) 具有标准的坐标 - 即盒限 (box limits) 为（0,0,0）至（1,1,1; 所以盒重心 (box centroid) 是（0.5,0.5,0.5）。

要删除点，请单独悬停在点上（点将变为绿色），然后按删除键 DEL。

在 Keyframe Editor 中以适当的旋转角度设置关键帧，以创建关于任意轴 (arbitrary axis)的旋转动画。

更改 Bricks Editor 中的各种参数，并观察渲染中的变化。

Bricks Editor 中的各种参数还可以通过输入文本来改变。

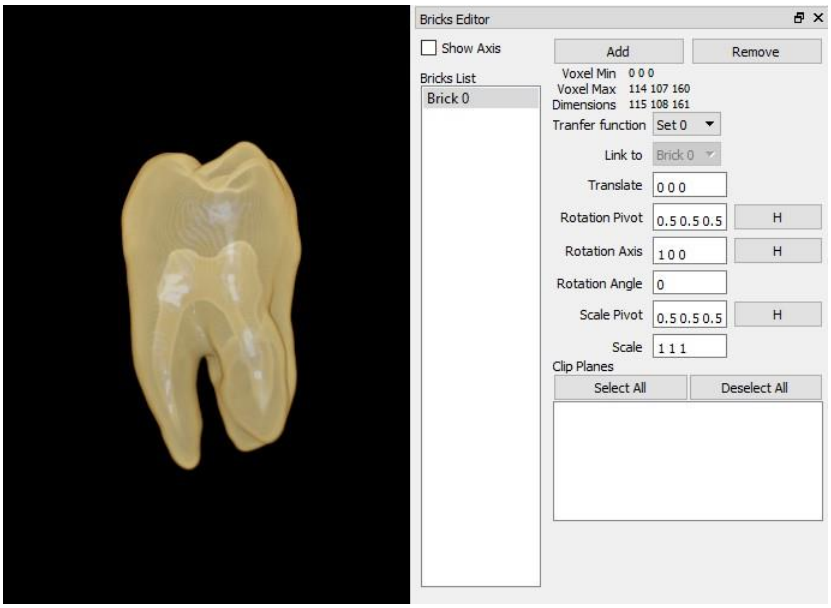
课程-10
结合 Bricks Editor 和 Keyframe Editor 制作简单动画

打开 Drishti Render.

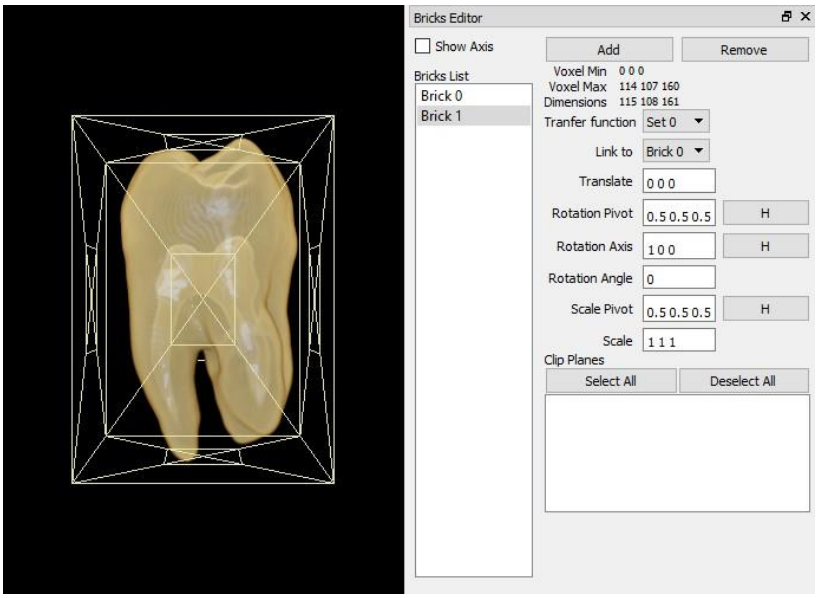
在 Drishti Render 中加载 tooth.pvl.nc（导入在课程 01 中的

创建结果）。切换到 HiRes 模式。按 b 隐藏边框。

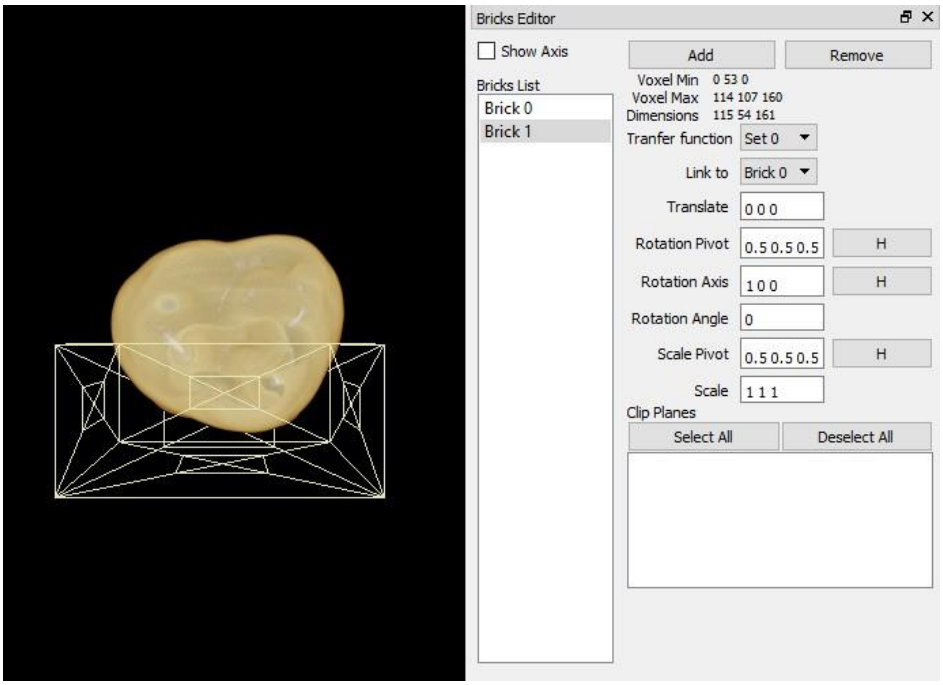
打开 Bricks Editor.



砖编辑器(Bricks Editor)中按添加 Add 按钮添加一个新的砖块。

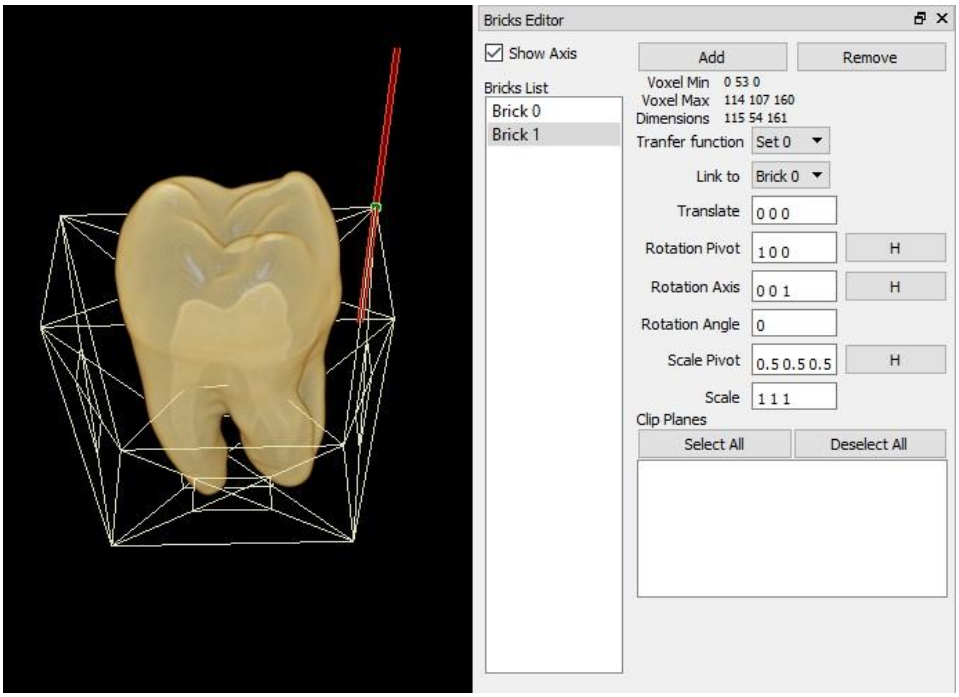


从砖列表中选择砖 1（Brick 1）。 通过移动框横栏来更改渲染窗口中砖 1（Brick 1）的大小。

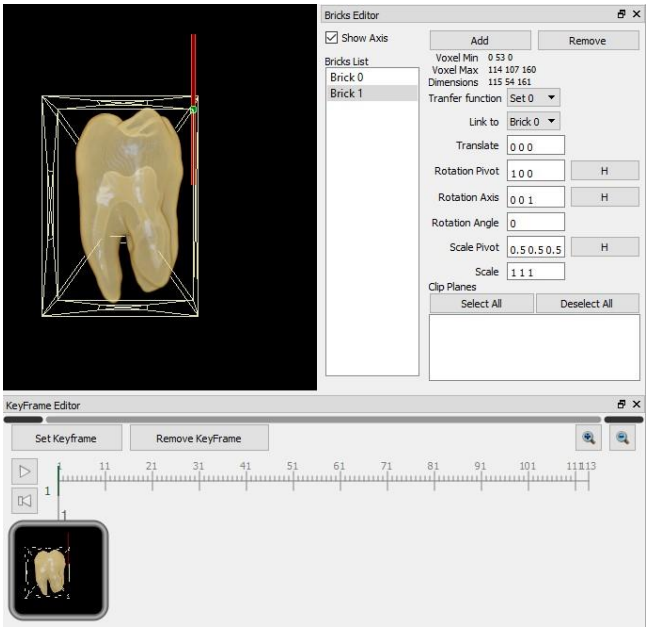


体素最小值(Voxel Min)、体素最大值(Voxel Max)和尺寸值(Dimensions)将在 Bricks Editor 的 Brick1 中更改。

检查显示轴(Show Axis)，显示旋转轴为红色。将旋转轴中心点设置为 1 0 0，旋转轴为 0 0 1。旋转轴现在位于砖盒的一角。

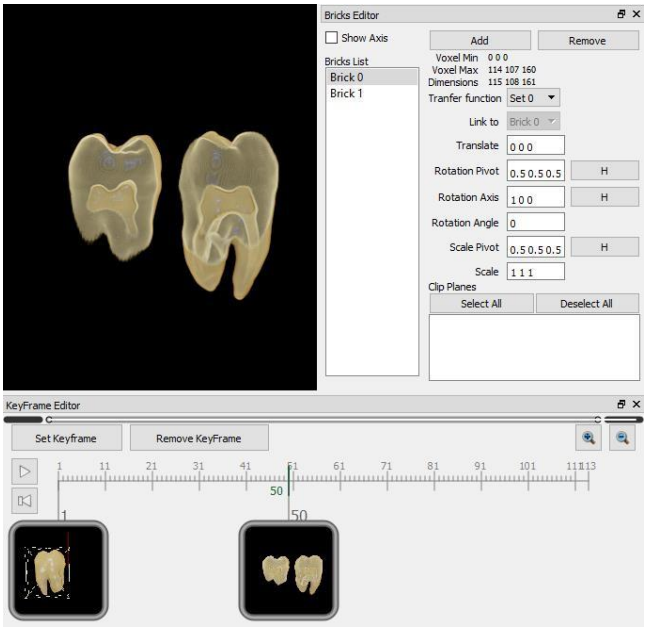


打开 Keyframe Editor 并设置关键帧。



将砖 1 旋转角度更改为-180。 取消选中显示轴。 选择砖 0，框显示将会被删除。带有交叉条的框只显示 Brick 0 以外的砖块。

在 Keyframe Editor 中设置另一个关键帧。



播放动画

您可以尝试更改传递函数参数（因为您可能需要在传递函数编辑器的相关集中添加传递函数）和使用多块砖等方式制作复杂的动画。请注意砖块是可以叠加的。

课程-11
通过对比度调试获得更清晰的数据

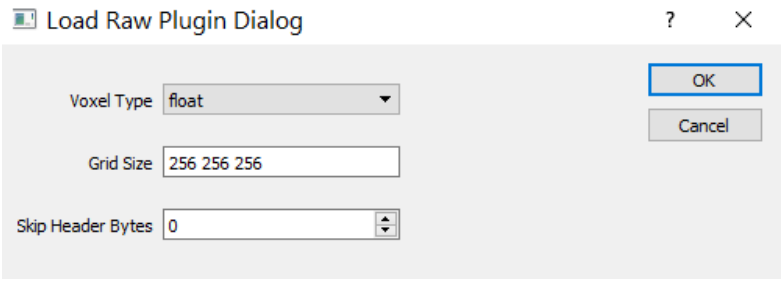
导入原始数据

打开 Drishti Import.

在 Drishti Import 中拖放 256_256_256.raw。

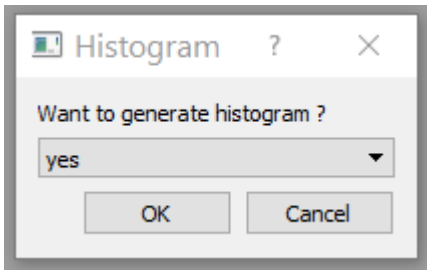
从下拉列表文件类型中选择选项 8：初始文件 RAW Files。 另一种加载数据的方法 - 文件 - >加载 - >初始文件 Files->Load->RAW Files

您将看到以下屏幕

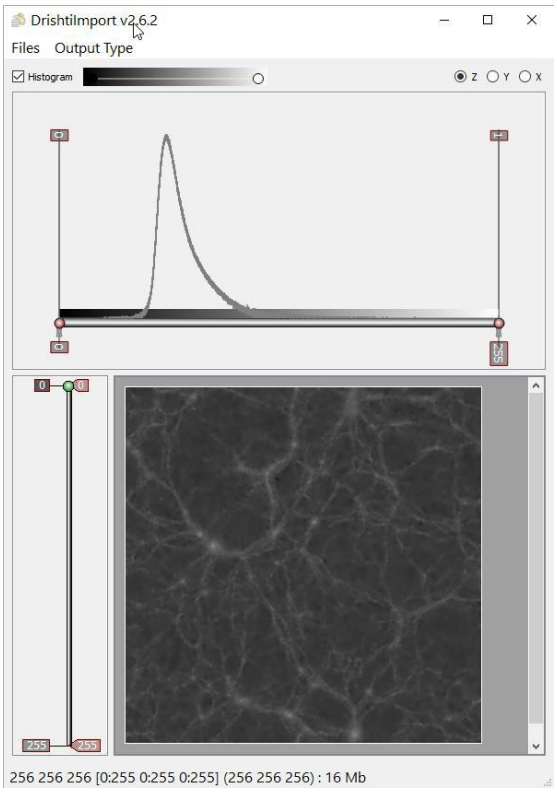


按 OK 继续下一步。

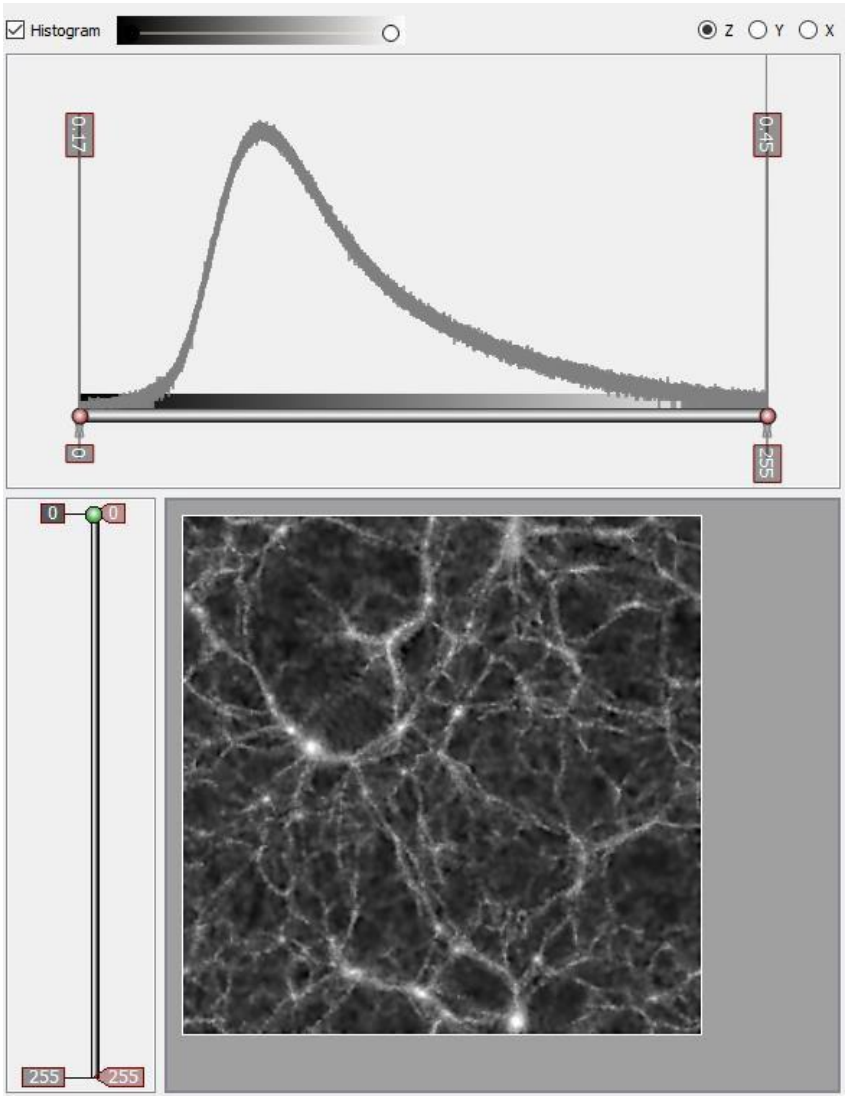
按 OK 可以生成直方图



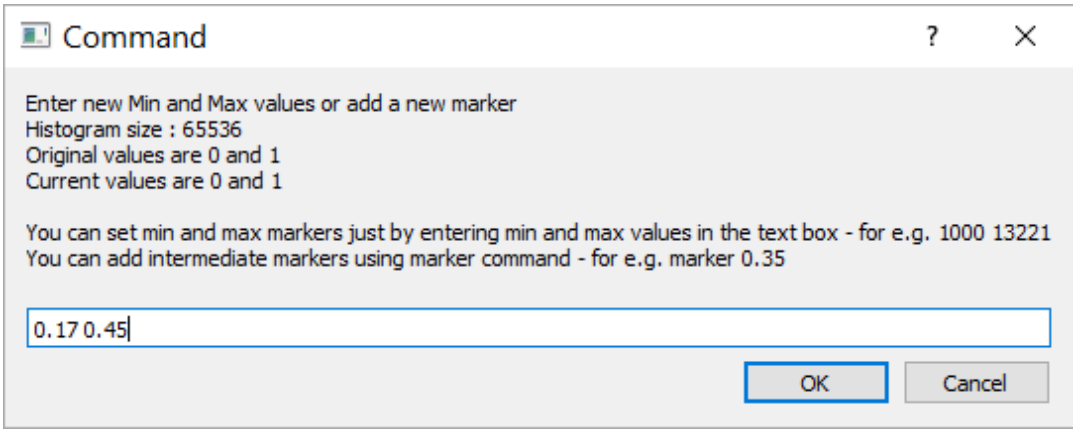
导入数据将生成直方图并将其显示在窗口的顶部。底部将显示导入数据的第一个切片。



移动终点标记，使下限约为 0.17，上限约为 0.45（如右边的图所示）。 这些值将分别映射到 0 和 256。我们之所以改变极限（limits）是为了增加数据的对比度。



还有另一种改变下限和上限的方法。将鼠标光标移动到直方图窗口并按空格键。随后会弹出一个对话窗口，然后在该文本框中输入必需的值。推荐输入 0.17 0.45，原因见上。



这些是终极限制标记。

您也可以添加附加标记。

附加标记可以通过标记命令添加，也可以通过左键单击直方图底线来添加。如果想要删除这些附加标记可以单击鼠标右键。移动标记可以增加/减少处理卷轴中的对比度。推荐大家多做尝试，找到最佳的对比度。

按 s（文件 - >另存为）处理和保存卷轴。

选择保存后的卷轴并命名为 cosmo.pvl.nc

请注意：运用 Drishti import 导出的卷轴或者数据均

为 .pvl.nc 格式，即导出的文件理论上来讲都是 Drishti-

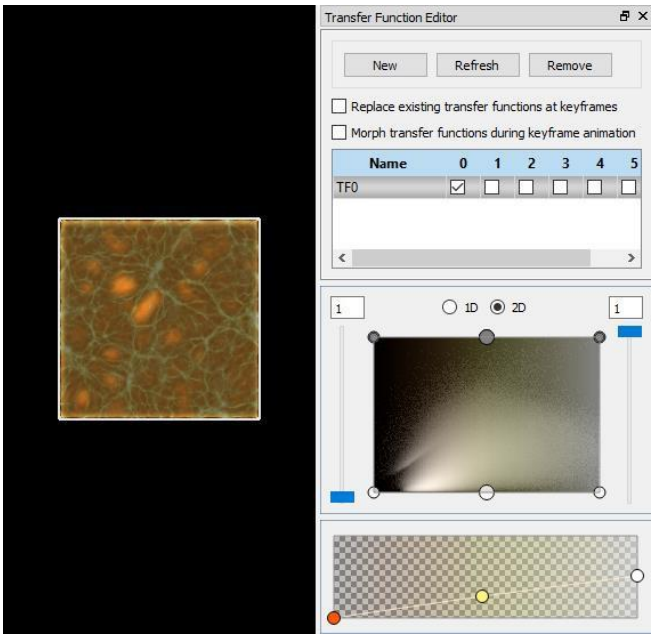
friendly 的。

课程-12
裁剪数据并结合 Keyframe Editor 制作动画

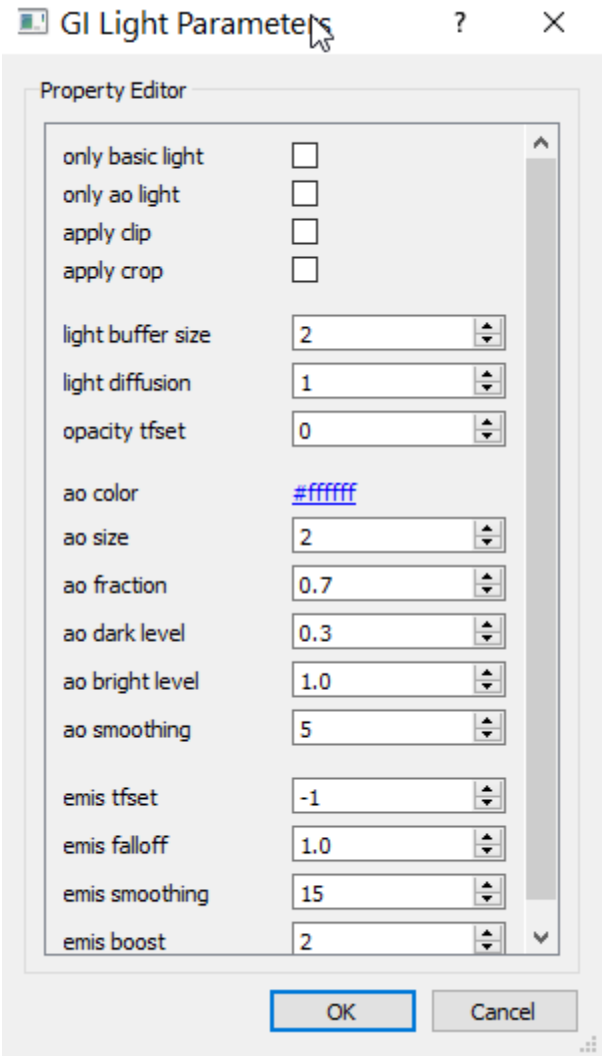
打开 Drishti Render 并加载在课程 11 中创建的 cosmo.pvl.nc。

按 F2 切换到 HiRes 模式。

选择合适的传递函数。



分离（或关闭）传递函数编辑器。

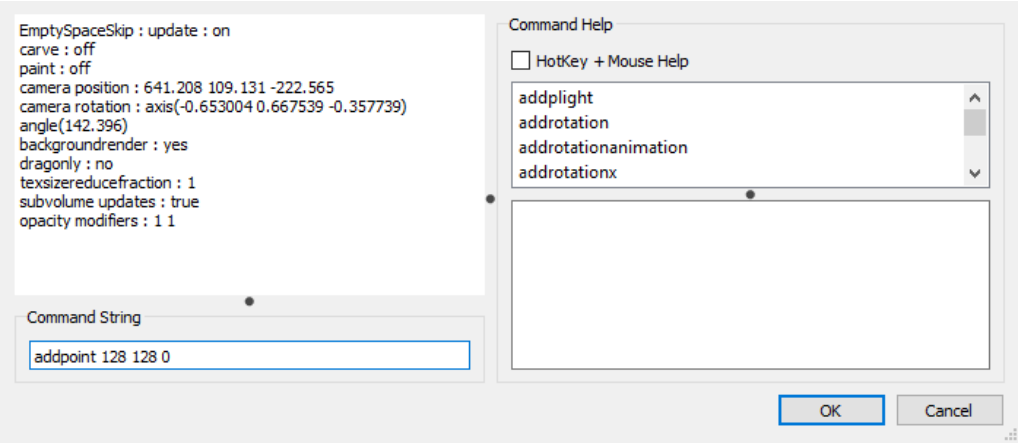


将鼠标光标移动到渲染窗口中，然后按 TAB 键弹出 GL Light Parameter 对话框。 将连接的面板分离到渲染窗口的原因是 TAB 的默认操作是更改键盘焦点，而我们想绕过该操作。

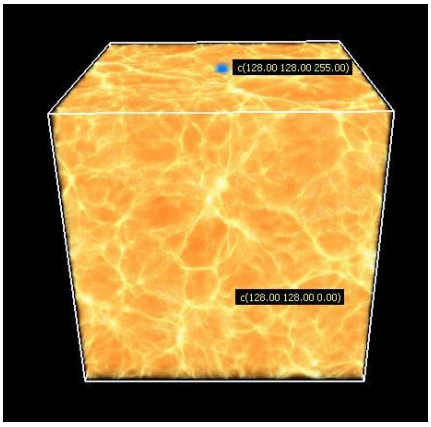
检查基本的光(only basic light) - 本课程只使用默认的照明。只有打开基本的光线，您才能看到更亮的渲染。

稍后将在本课程中切换阴影显示。

按空格键显示主命令对话框(Commend dialog)。 在命令字符串(Command String)文本框中输入 addpoint 128 128 0



在座标（128，128，0）处添加一个点。



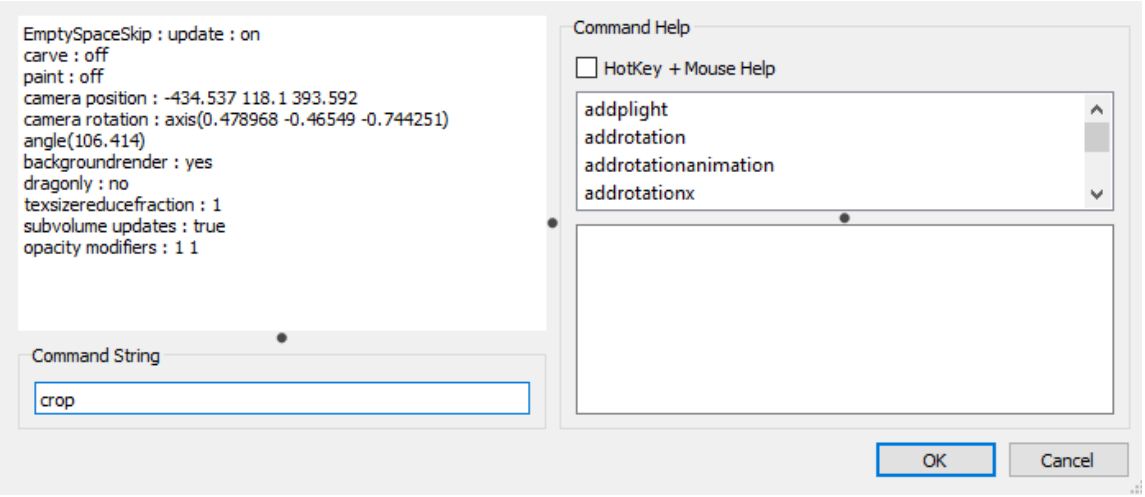
在（128，128，255）处添加另一个点。

【也可以通过 Shift +鼠标左键添加点】

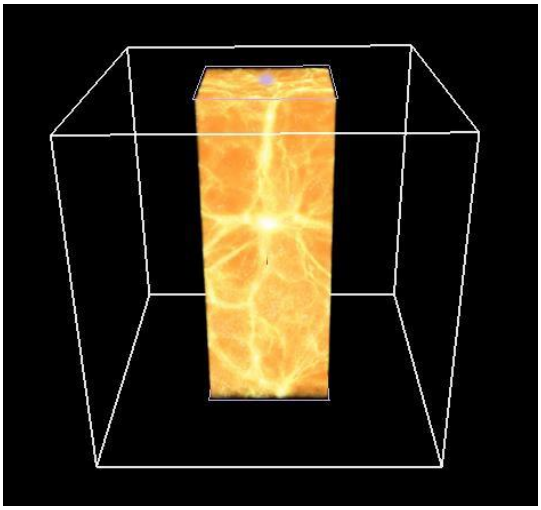
将鼠标悬停在其中一个点上（选中的点将变为绿色），然后按 c，以显示该点的坐标。

c 用作拨动开关 - 在悬停在一个点上时，再次关闭坐标显示。

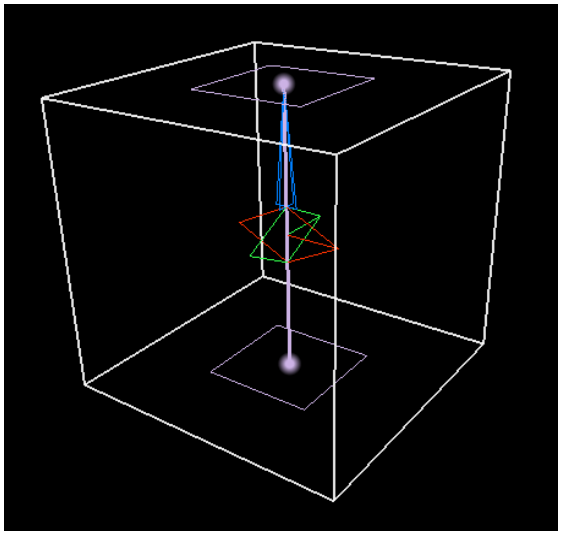
再次按空格键进入主命令对话框。 在命令字符串中输入 crop 即裁剪。



将出现以下渲染

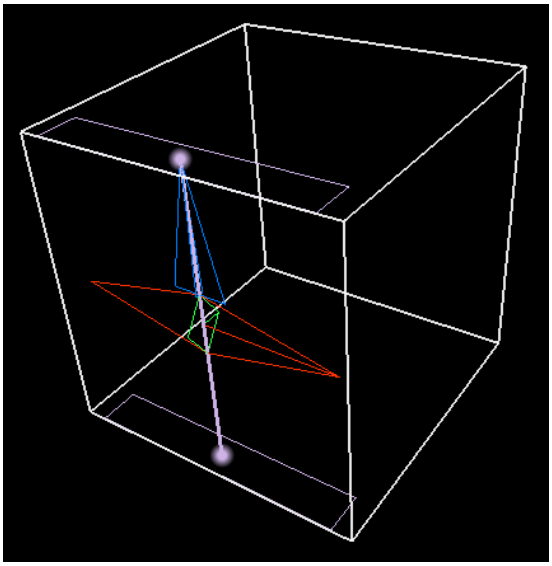


启动传递函数编辑器(Transfer Function Editor)并关闭设置 0 中 TF0 的复选框，以显示裁剪小部件(crop widget)。



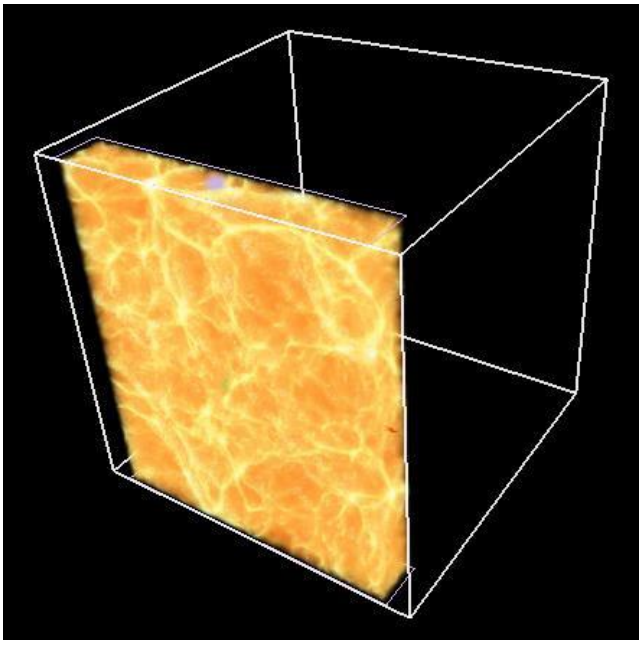
裁剪小部件上的红色、绿色和蓝色手柄可以使用户沿着这些轴移动、旋转和缩放小部件。悬停在红色/绿色/蓝色平面轴上，然后单击即可选择不同的平面轴。 选中平面轴以后，可以用鼠标左键拖动旋转平面，并用鼠标右键平移这些平面轴。

使用 Ctrl +右键拖动可以更改该特定轴的裁剪尺寸。

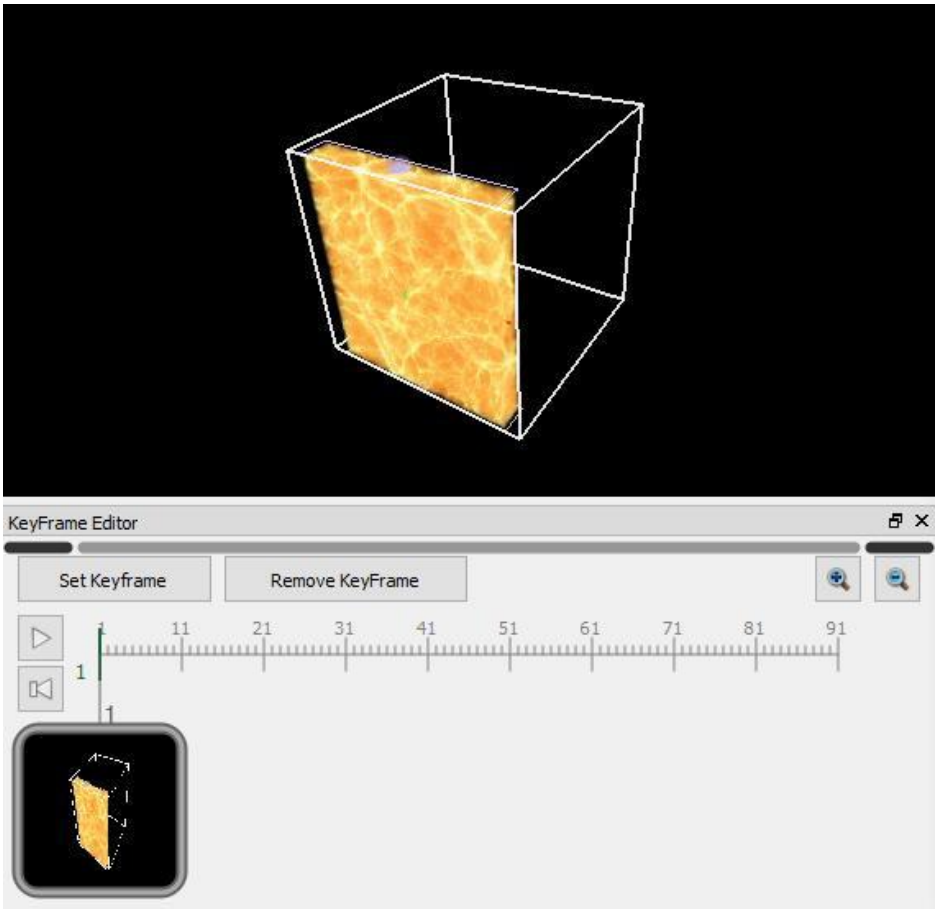


现在移动所有裁剪平面轴到盒子的边缘。薄板缩小红色和绿色的平面轴，最后得到一个。

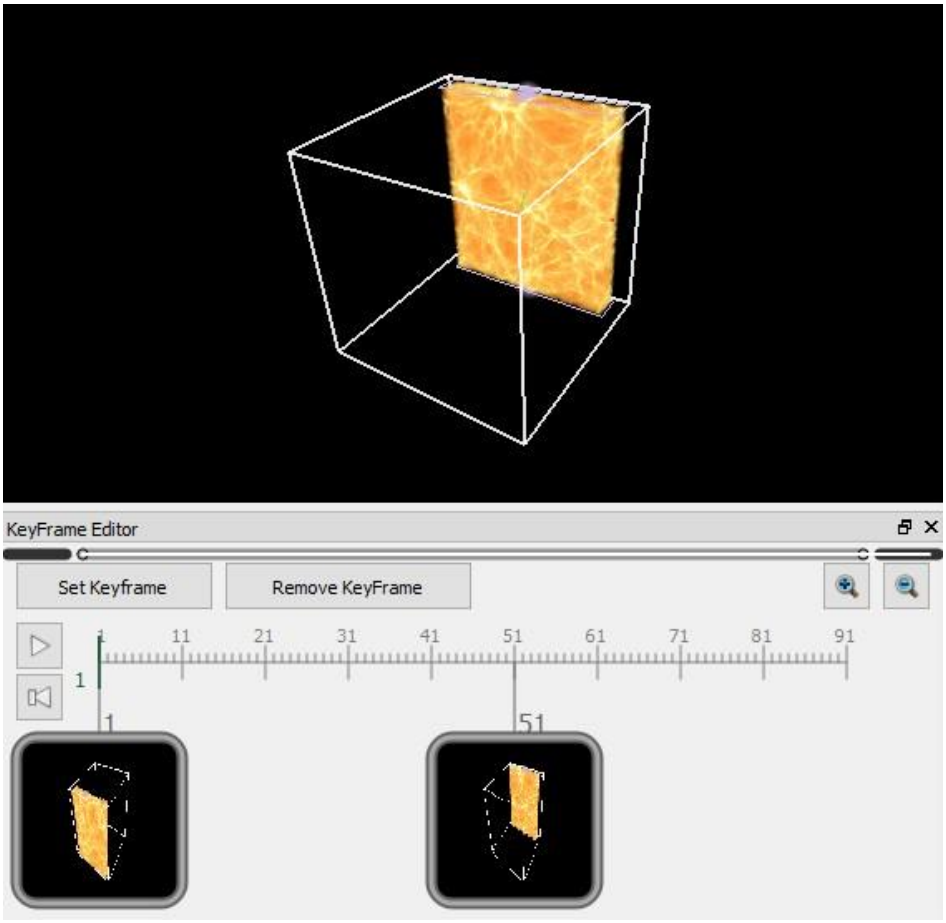
在 Set 0 中打开 TF0。



打开 Keyframe Editor 并设置一个关键帧



将裁剪移动到边框的另一端，并设置另一个关键帧。



按 v（切换 - >可见性 Toggle->Visibility）切换裁剪小部件的可见性。然后设置关键帧，并保持裁剪小部件不可见。

按 1（切换 - >阴影渲染 Toggle->Shadow Render）打开基于屏幕阴影。

播放剪裁动画。

将位于中间的帧插入剪裁参数。 也可以更改裁剪大小和旋转，然后保存改动的关键帧随后查看参数大小等改变以后对动画有什么影响。

注意：
一旦选择关键帧，剪裁小部件将变得无法抓取，以确保用户不会无意中更改设置。要使小部件可以抓取，请按下切换开关 g（Toggle-> Mouse Grab）。

课程-13
运用裁剪参数对话框进行部分裁剪

打开 Drishti Render 并加载课程 12 中使用的数据：cosmo.pvl.nc。

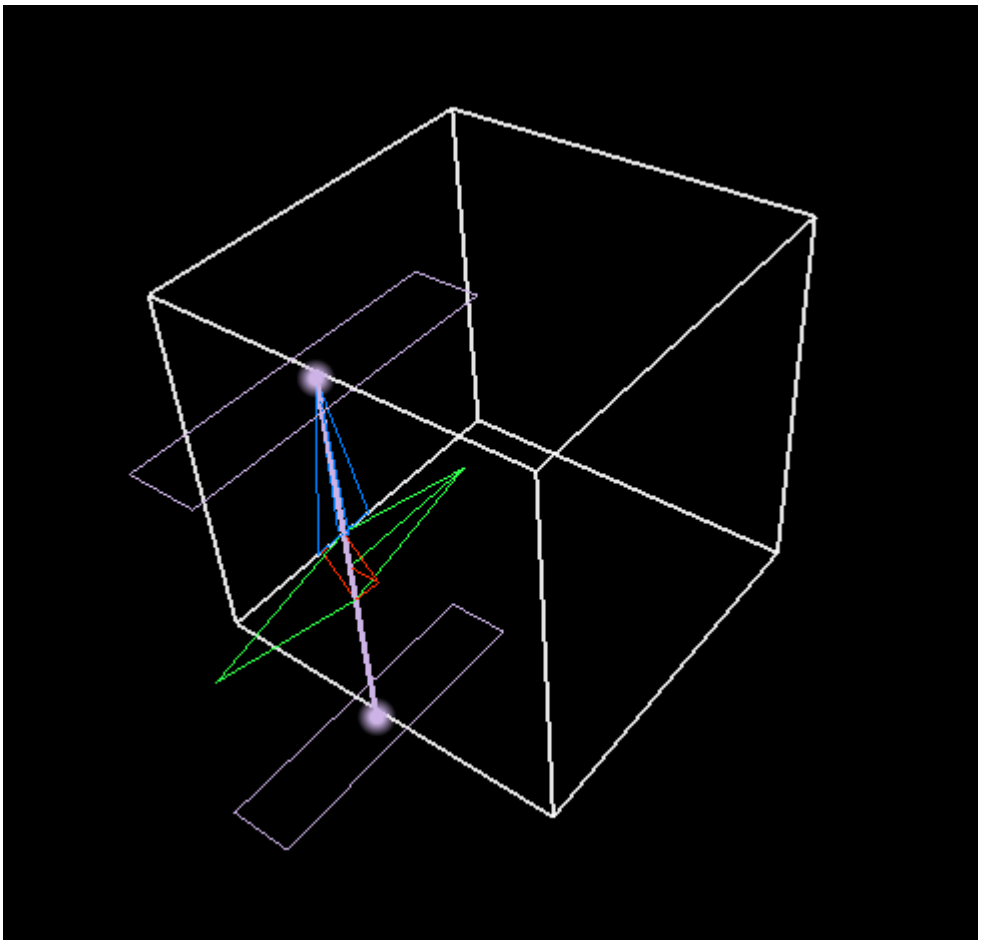
按 F2 并切换到 HiRes 模式。

如课程 12 所示，切换到基本照明 - 检查在 GI 灯参数框中只有基本灯光。

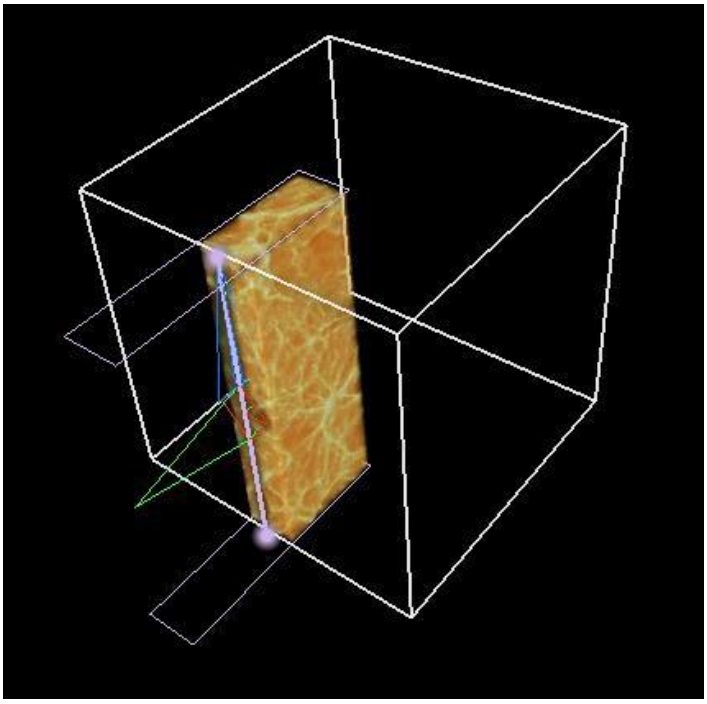
按照课程 12 的步骤添加一个剪裁。

在传递函数编辑器中关闭设置 0 中的 TF0，使得裁剪小部件清晰可见。通过 Ctrl + Left 鼠标左键拖动相应的侧面，操纵裁剪小部件，扩大绿色平面轴，并缩小红色平面轴。

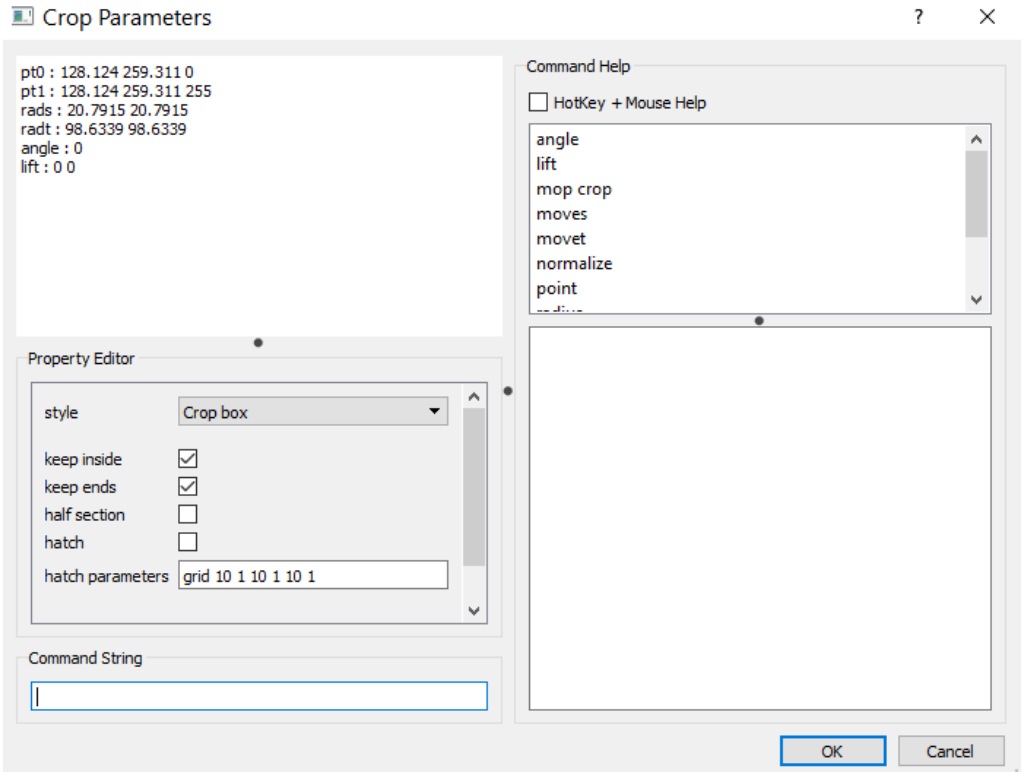
移动裁剪小部件，使其位于边框的边缘，如下图所示。



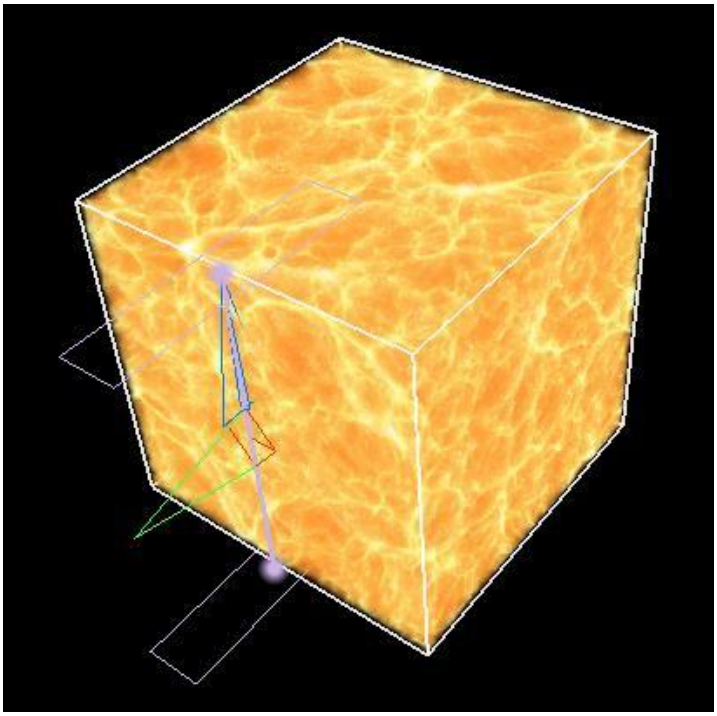
打开传递函数编辑器中的 Set 0 中的 TF0。



将鼠标悬停在裁剪窗口小部件上，然后按空格键，“裁剪参数”对话框就会出现。

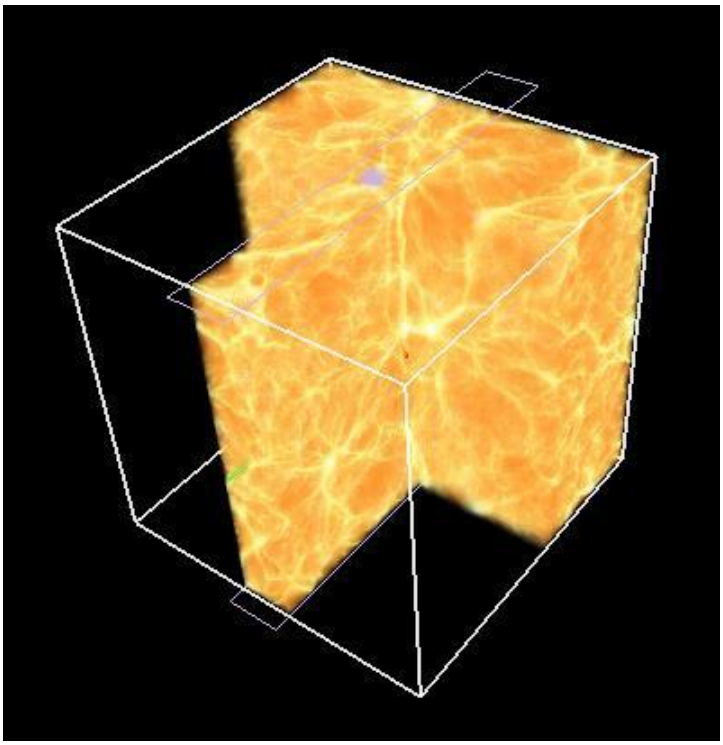


在 Property Editor 目录下，勾选半部分(half section)， 您将获得以下图像。



打开 Keyframe Editor 并添加关键帧。

将裁剪沿着绿色轴移动到后面，并保存此关键帧。



播放动画，您将看到从卷轴中挤出的板坯。

按 1（切换 - >阴影渲染 Toggle->Shadow Render）打开屏幕阴影(screen-based shadows)。

注意：以确保用户不会无意中更改设置，选择关键帧时/后，剪裁小部件将变得无法选取。如果要使小部件，请按下切换开关 g（Toggle-> Mouse Grab）。

在“裁剪参数”对话框中尝试改变不同的参数，可以看到不同参数对渲染结果的影响。

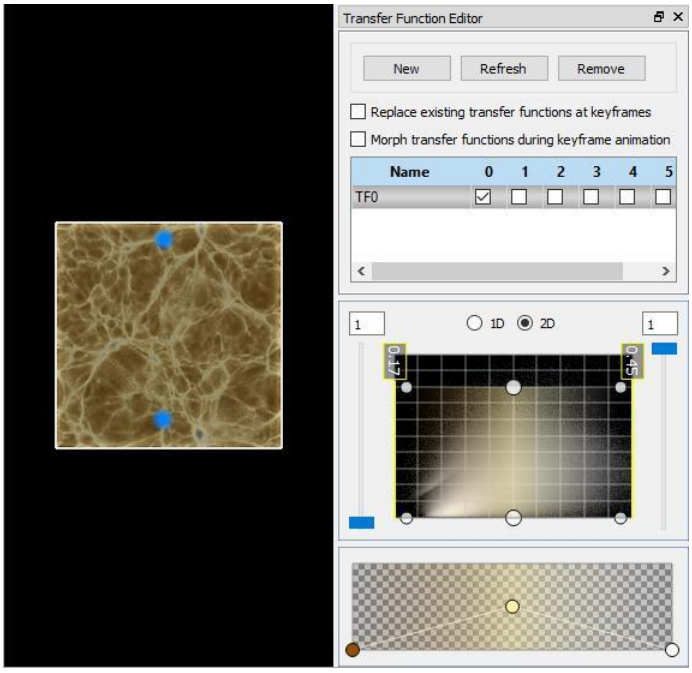
课程-14
可视化数据混合

打开 Drishti Render 并加载课程 12 中使用的数据：cosmo.pvl.nc。

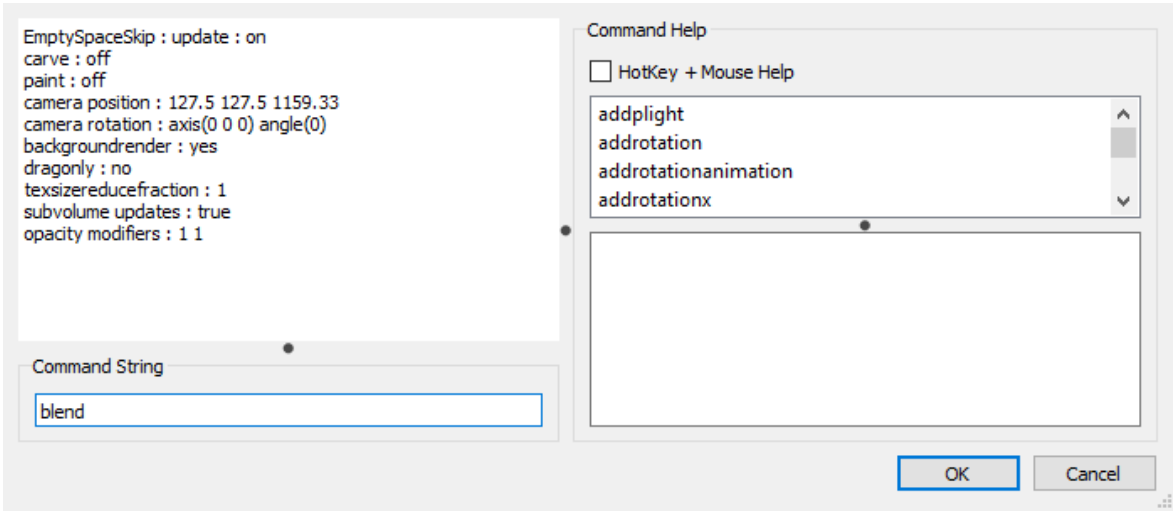
按 F2 并切换到 HiRes 模式。

如课程 12 所示，切换到基本照明 - 即在 GI 灯参数框中只显示基本灯光。

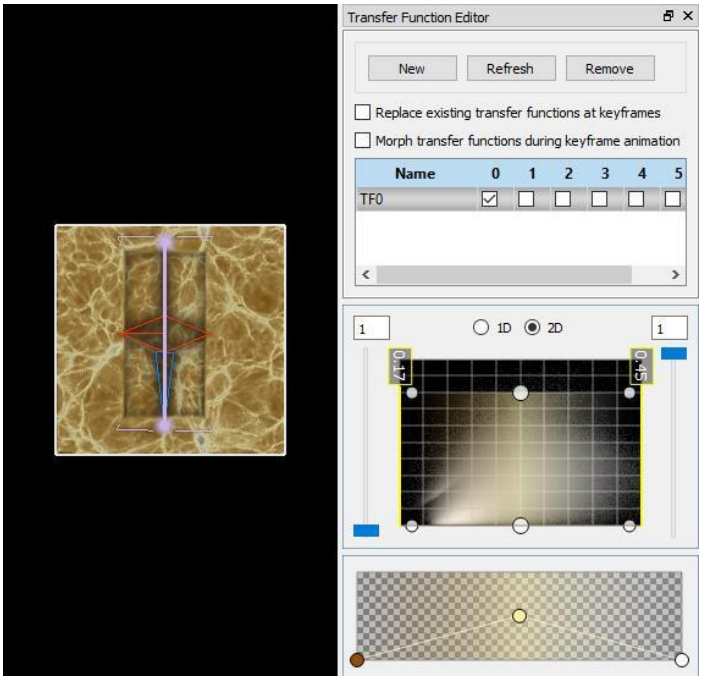
使用 Shift +鼠标左键添加两个点。



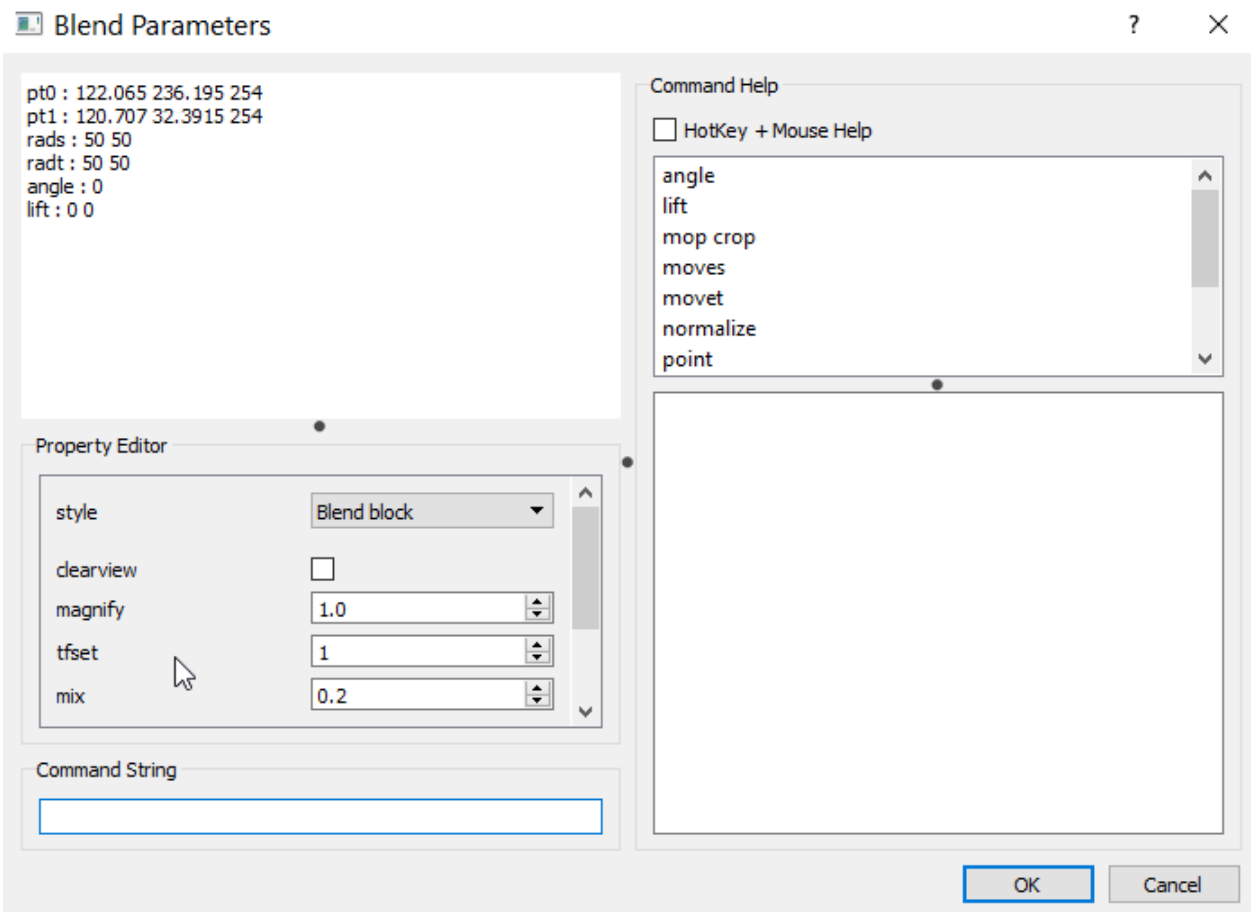
按空格键进入主命令对话框。 在命令字符串中输入混合（blend）。



按 1（切换 - >阴影渲染 Toggle->Shadow Render）打开屏幕阴影。

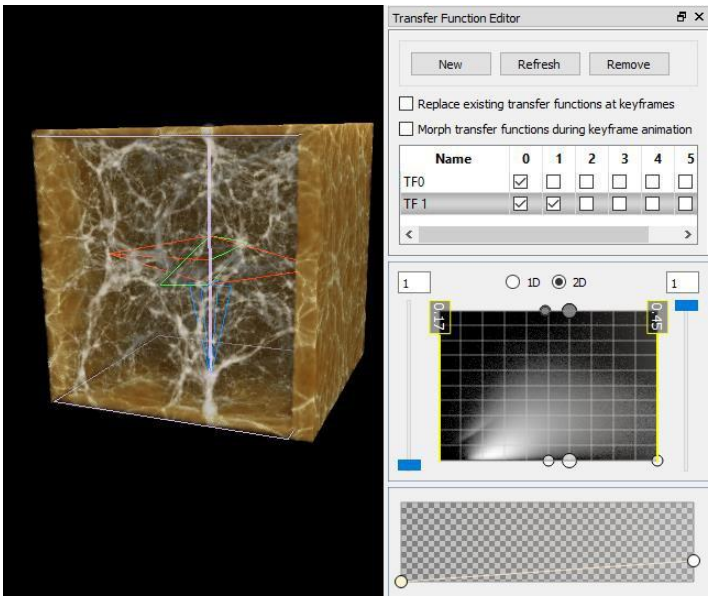


将鼠标悬停在混合小部件(blend widget)上，然后按空格键以显示“混合参数 Blend Parameter ”对话框。

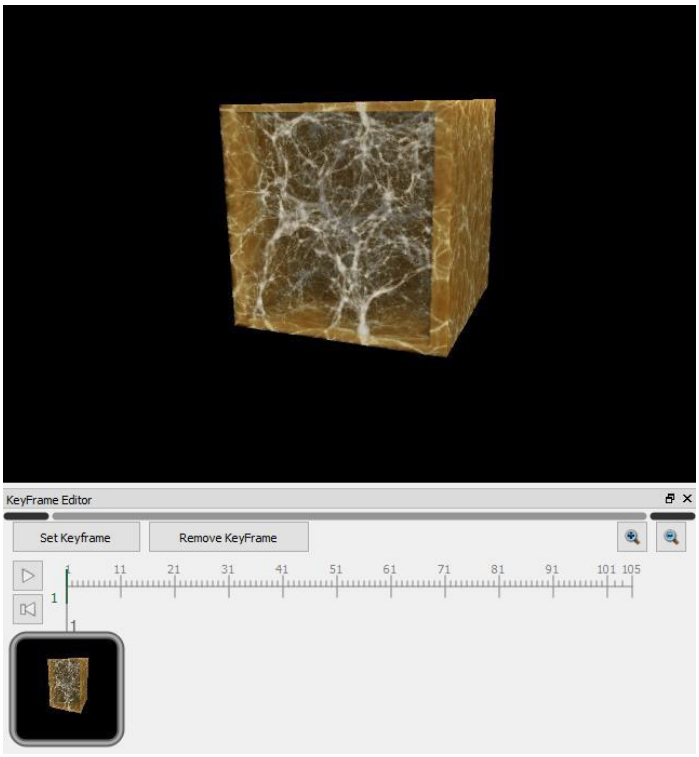


请注意，tfset 参数为 1（系统默认）在本课程中，混合操作将使用 Set 1 中的传递函数。目前在传递函数编辑器下没有将传递函数添加到 Set 1，因此混合区域目前是空的。

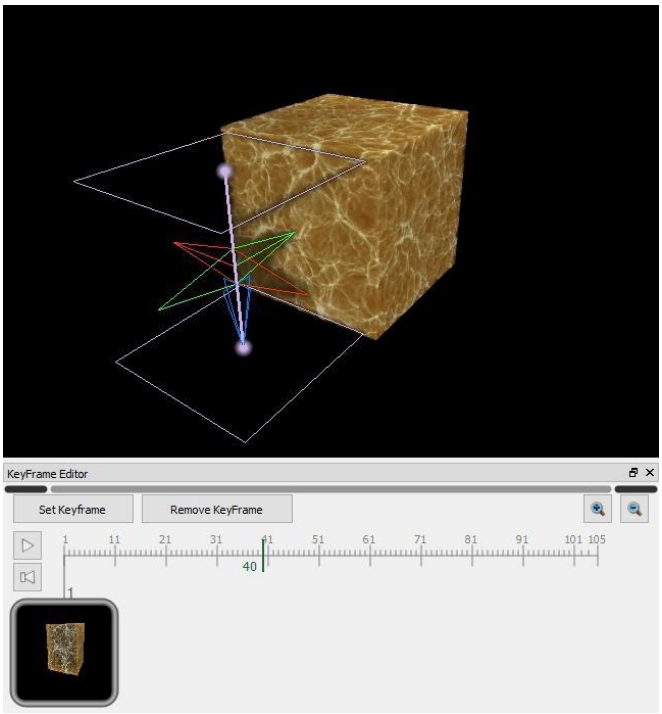
使用传递函数编辑器添加一个新的合适的传递函数并将其添加到 Set 1。新的传递函数可以在两个集合中设置 0 以及设置 1。修改混合窗口小部件（操作类似于裁剪窗口小部件），以便占用一个较大的区域。



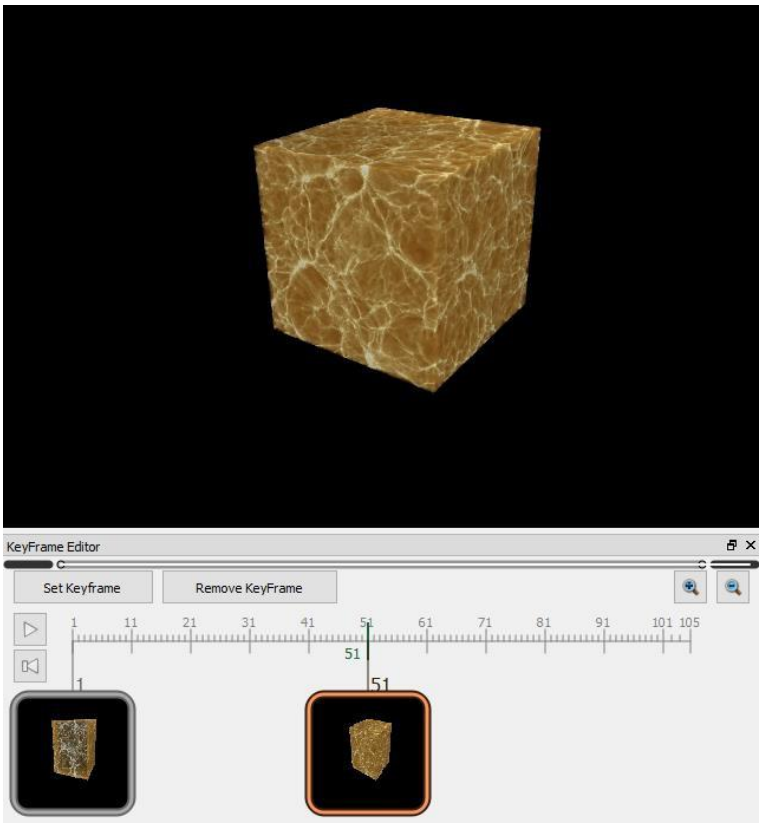
打开关键帧编辑器。隐藏混合小部件（按 v）并保存关键帧。



按 v 显示混合小部件。将混合小部件移出卷轴所占据的边界框。



隐藏混合小部件并保存另一个关键帧。



将两个关键帧移动到较近的位置，以便混合操作在卷轴中打开一个孔，以显示混合传递函数。将位于 1 的关键帧移动到 61 帧，而位于 51 的关键帧移动到第 1 帧。

然后播放动画

注意：以确保用户不会无意中更改设置，选择关键帧时/后，剪裁小部件将变得无法选取。如果要使小部件，请按下切换开关 g（Toggle-> Mouse Grab）。

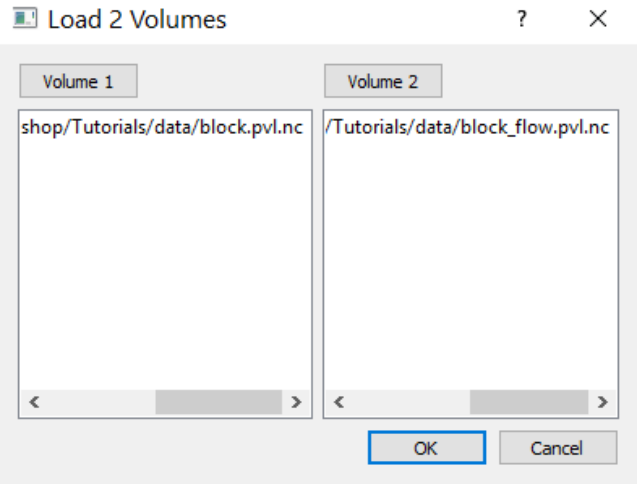
在“裁剪参数”对话框中尝试改变不同的参数，可以看到不同参数对渲染结果的影响。

课程-15
融合两组数据并进行可视化处理

打开 Drishti Render.

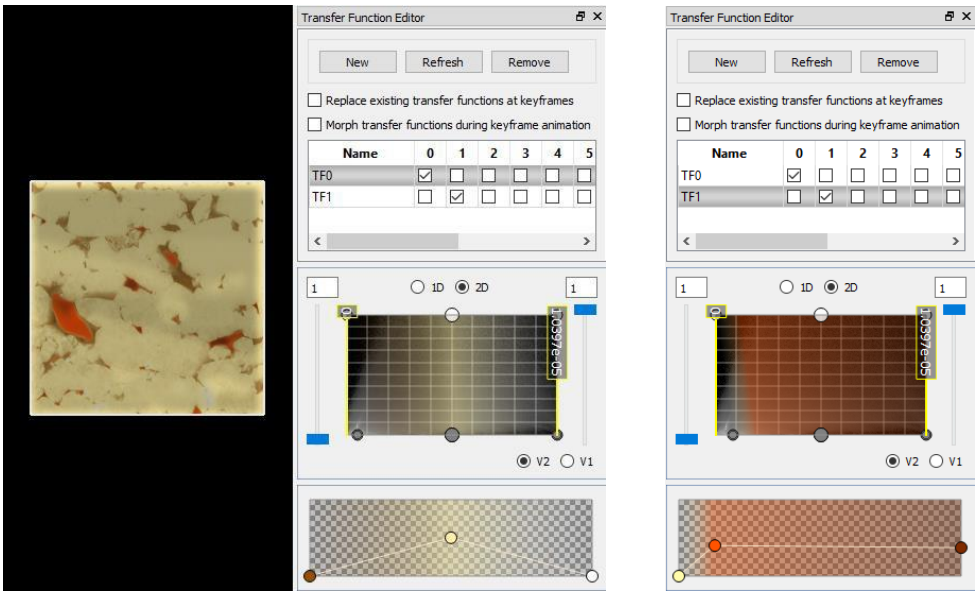
File->Load Volume->Load 2 Volumes
（文件 - >加载卷轴 - >加载 2 个卷轴）

将 block.pvl.nc 作为第一个卷轴，将 block_flow.pvl.nc 作为第二个卷轴。

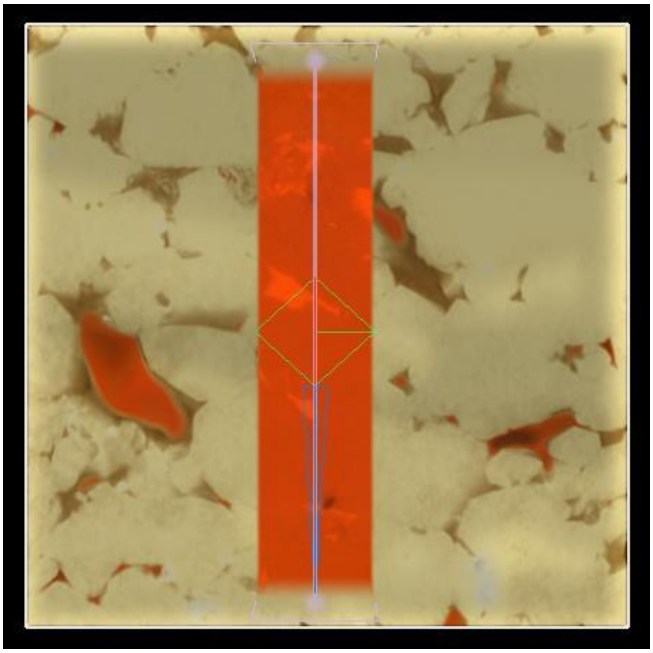


按 F2 并切换到 HiRes 模式。

为两个卷轴自动设置两个传递函数。Set 0 中的传递函数将作用于第一卷轴，Set 1 中的传递函数在将作用于第二个卷轴。视需要修改传递函数。

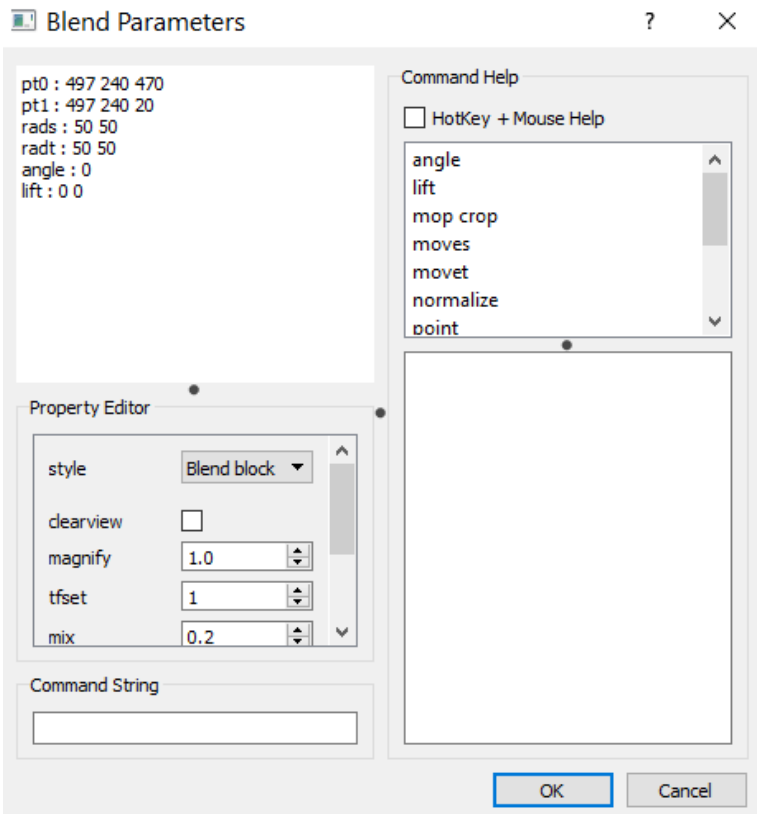


使用 Shift +鼠标左键，单击并添加两个点，并按照课程 14 的步骤完成创建混合小部件。

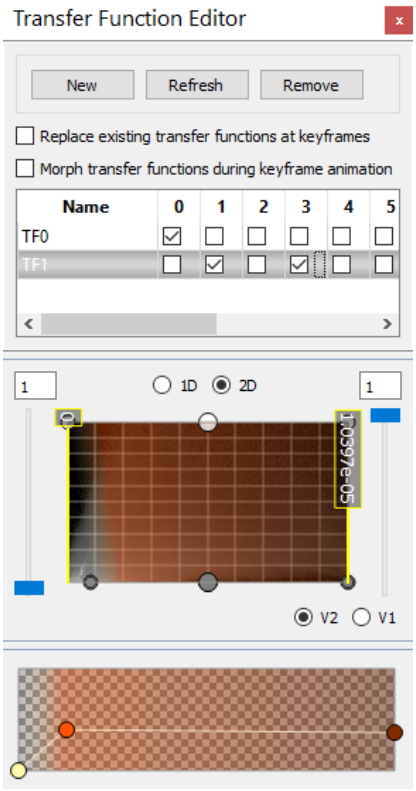


再按照课程 12 的步骤，切换到基本照明 - 可以通过按 TAB 来查看 GI Light 参数框中的仅选择了基本灯光。

将鼠标悬停在混合小部件上，然后按空格键以显示“混合参数 Blend Parameter”对话框。



将 tfset 参数设置为 2（系统默认值为 1）。由于没有为 Set 2 和 3 设置传递函数，所以混合区域将为空。



将传递函数 TF1 设为 Set 3。保持 Set 2 为空 - 即无传递函数。

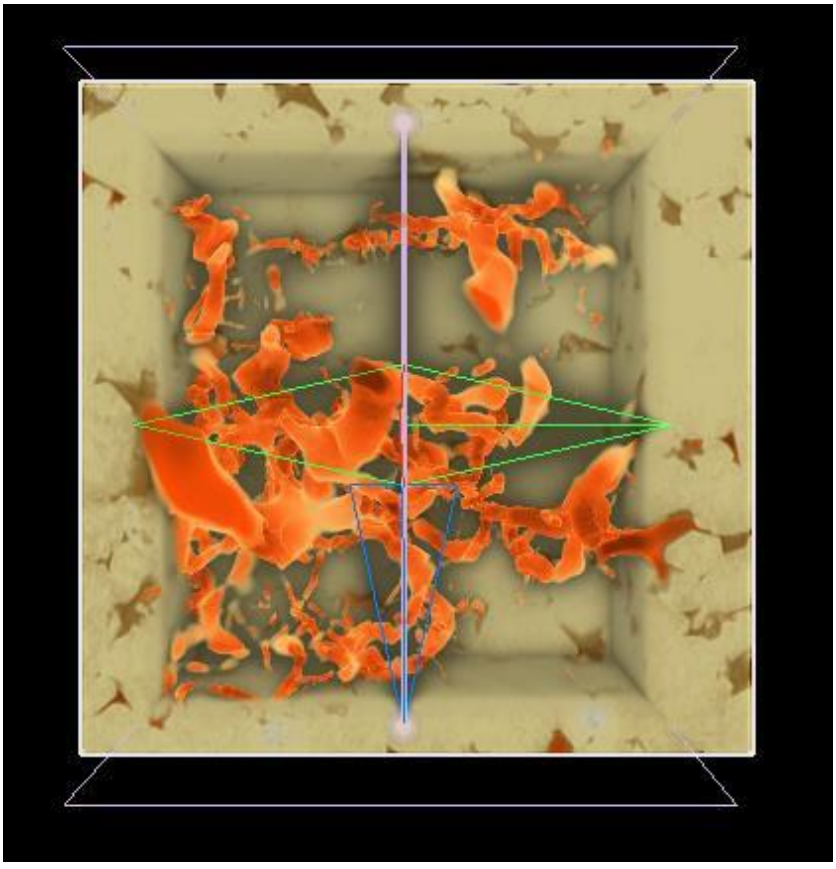
使用 tfset 参数并将混合小部件设置为 2，混合区域将 Set 2 传递函数应用于第一个卷轴，并将 Set 3 传递函数应用于第二个卷轴。

由于在 Set 2 中没有任何内容，所以在混合区域中只能看到第二个卷轴，而看不到第一个卷轴。

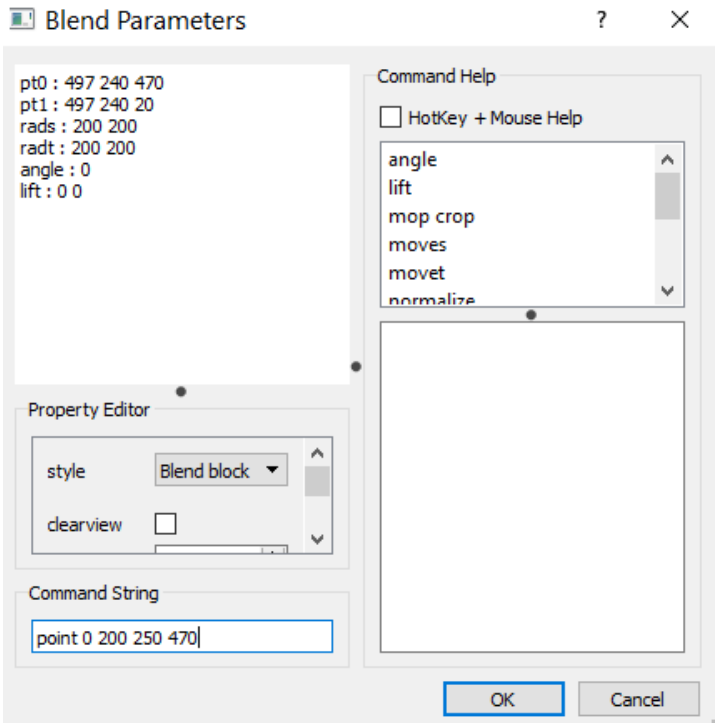
将鼠标悬停在混合小部件上，然后按空格键。

在混合参数对话框(Command String)的命令字符串(Blend Parameter)中输入半径 200 (radius 200)。此操作可以增加红色和绿色轴上的混合区域的大小。

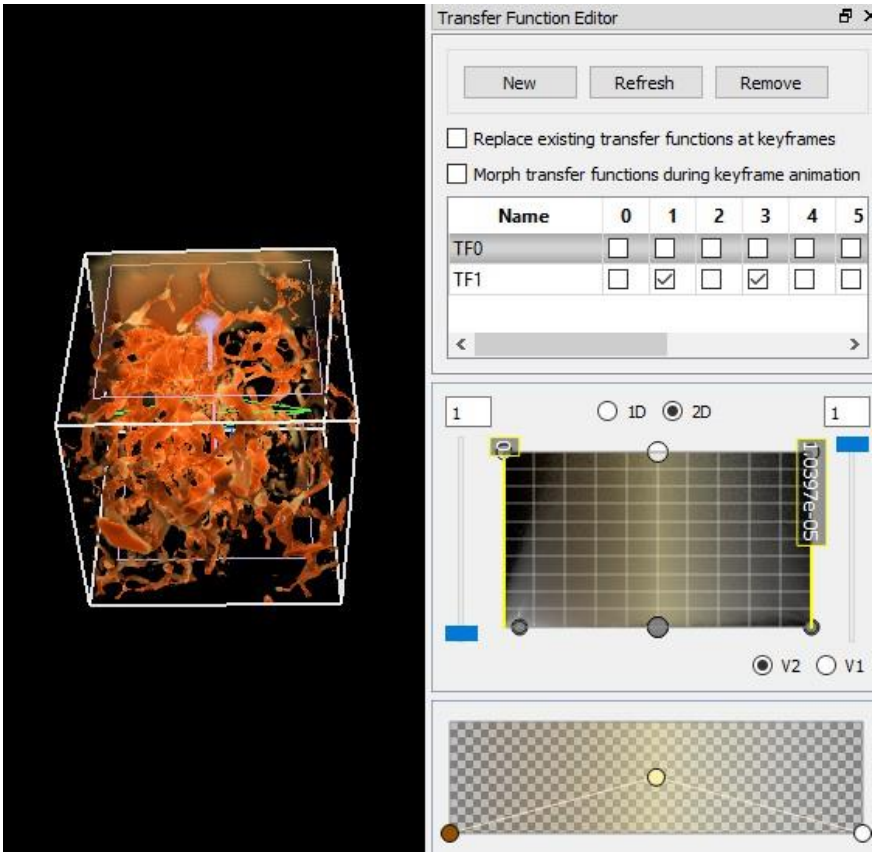
按 1 可切换基于屏幕的阴影（Toggle-> Render Shadows）。



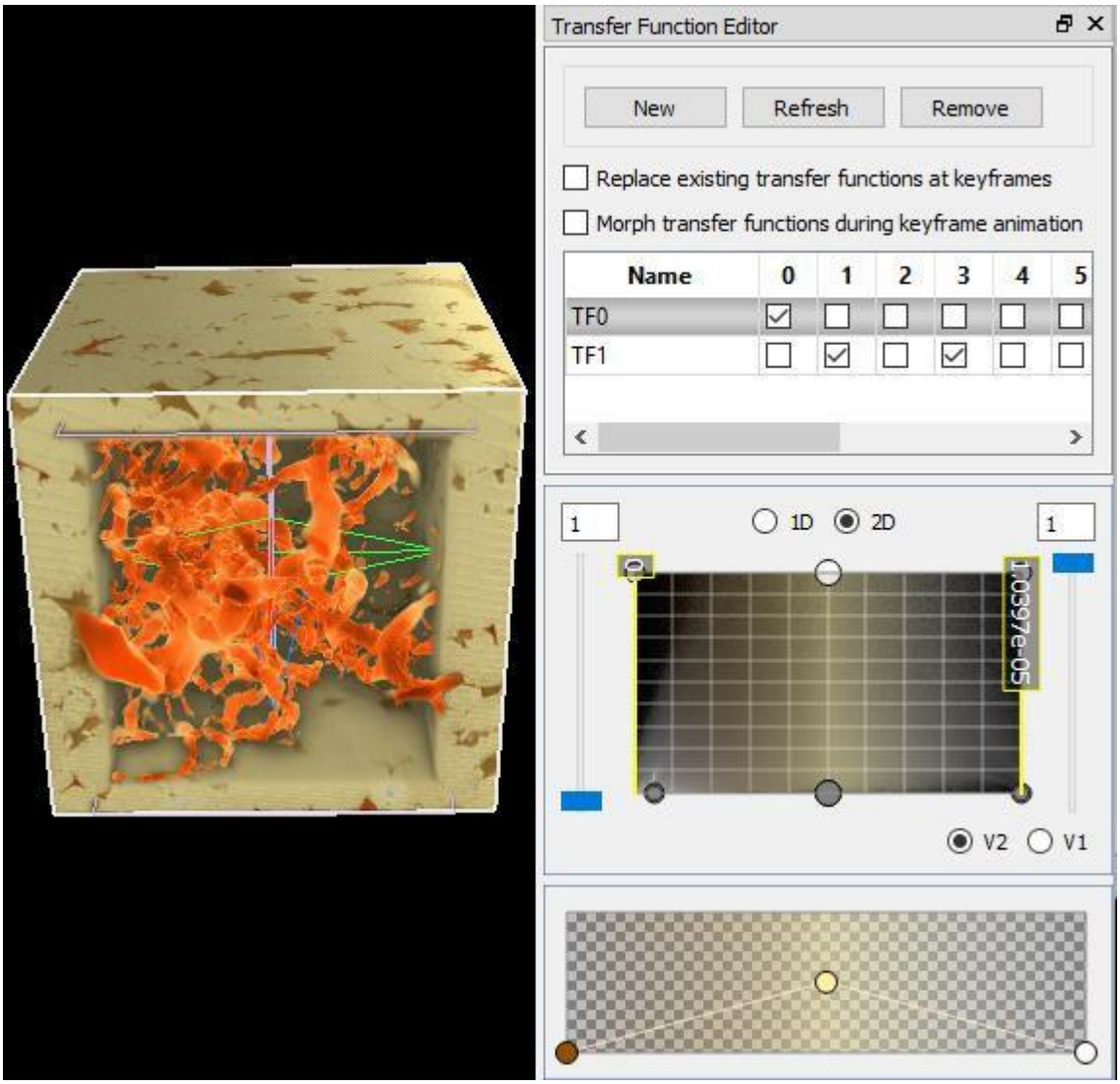
启动混合参数对话框并在命令字符串框中输入点 0 200 250 470。



此操作首先将两个混合区域的点移动到位置（200,250,470）。类似地，将第二个点移动到（200,250,20） - 在命令字符串中输入点 1 200 250 20。混合区域将右移到卷轴的中心。 如要查看混合小部件，请关闭 Set 0 中的 TF0。



移动混合小部件从一边伸出红色的轴，然后在 Set 0 中打开 TF0。

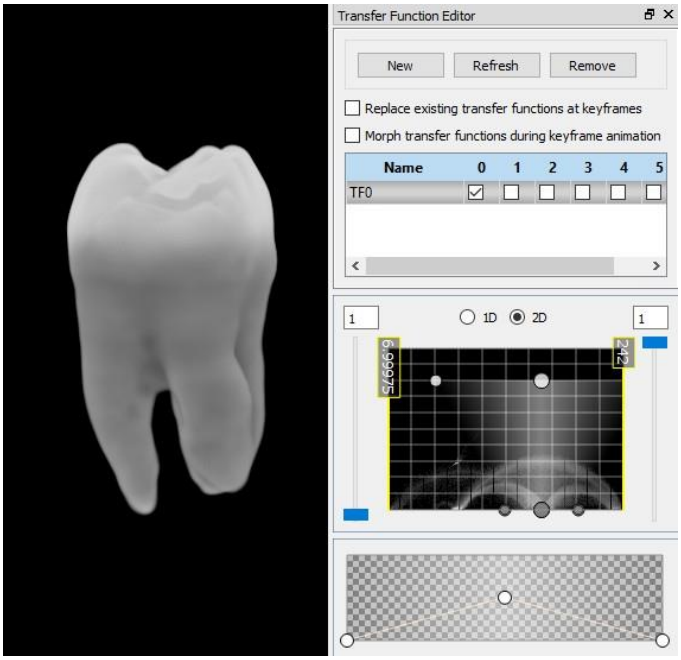


现在我们已经学习了怎样融合两组数据并进行简单的可视化处理，您可以参考之前的课程然后设置关键帧和制作动画。

课程-16
灯光参数运用

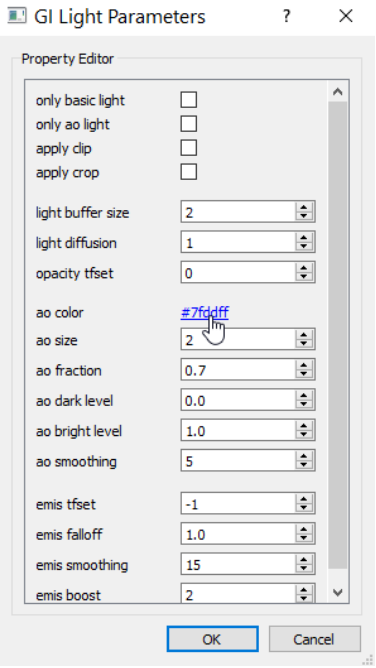
打开 Drishti Render
加载 tooth.pvl.nc 并切换到 HiRes 模式。

将传递函数全部设置为白色。要改变颜色，双击颜色渐变点(color gradient point)打开颜色编辑器(color editor)。



将传递函数编辑器分离。按 TAB 打开 GI 灯参数(GI Light Parameters)，只打开基本的光线(only basic light)。现在的渲染效果是纯白色的 - 所以看不见任何的表面特征。

再按下 TAB 键并关闭基本指示灯。将黑暗水平(ao dark level) 改为 0，然后可以看到牙齿变暗了。与透明的结构相比，牙齿结构中的实心部分现在变得更暗了。



现在更改 GI 光参数中的 ao color。要更改颜色，请单击蓝色文本（系统默认为白色，文本字符串默认为**#ffffff**）以显示颜色对话框。

观察对渲染的影响。牙齿现在将采取 ao color。

将 ao 黑色水平 (ao dark level) 更改为 1.0，将 ao 亮度级别 (ao light level) 更改为 0.0，并查看效果。 透明区域的牙齿应该更暗。而牙齿更厚、更坚实的区域将看起来则更亮。

将 ao 黑色水平和 ao 亮度分别切换回 0.3 和 1.0，并将 ao 颜色恢复为白色。

添加剪切平面（c）。 因为 GI 照明未被重新计算，并且使用了先前的照明条件，所以现在在剪裁侧看起来牙齿很暗。

通过以下步骤进行修复：
打开 GI Light 参数对话框并打开应用剪辑（类似的切换可用于 crop / blend / dissect - 应用裁剪）。 剪辑区域应正确显示。

移除裁剪平面 - 将鼠标悬停在裁剪平面上，然后按 DEL。

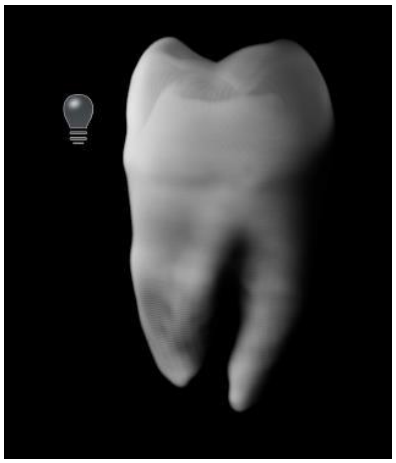
Shift +左键单击可在牙齿上添加点。

按空格键显示主命令对话框。

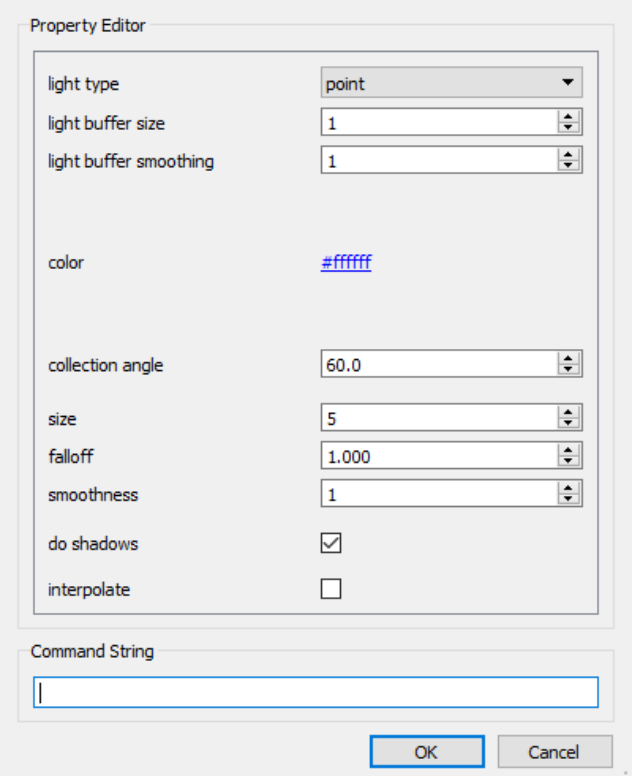
在命令字符串文本框中输入 addplight
以添加一个光点。

再次按 TAB 调出 GI 灯参数，并将 ao 颜色设为黑色（＃000000）。

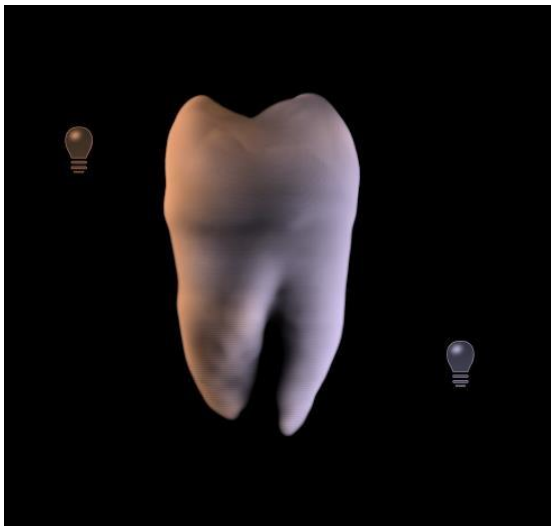
移动灯点让点远离牙齿，并且不能太靠近表面，这样可以看到正常照明下的牙齿。



悬停在光图标上，按空格键显示点光源的参数框。



更改点光的颜色 - Color，并查看对渲染的影响。



添加另一个光点并改变它的颜色。

尝试改变 ao 颜色，可以浏览不同的效果。

课程-17
光点的移动和增加

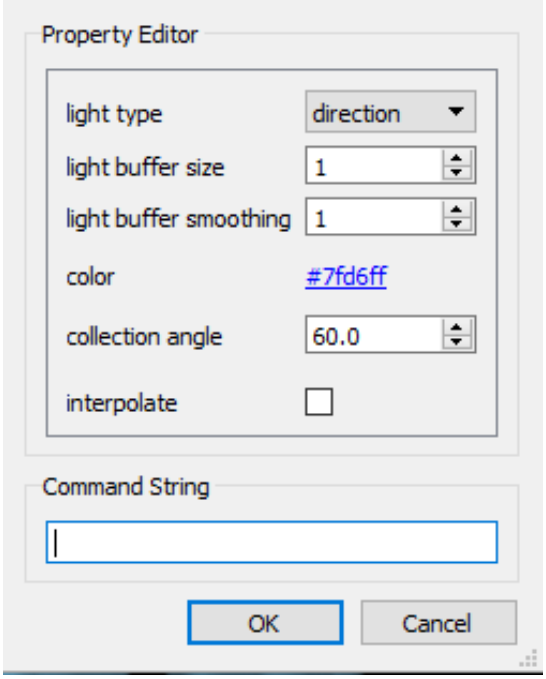
打开 Drishti Render

加载 tooth.pvl.nc 并切换到 HiRes 模式。

参考课程 16，将传递函数全部设置为白色。

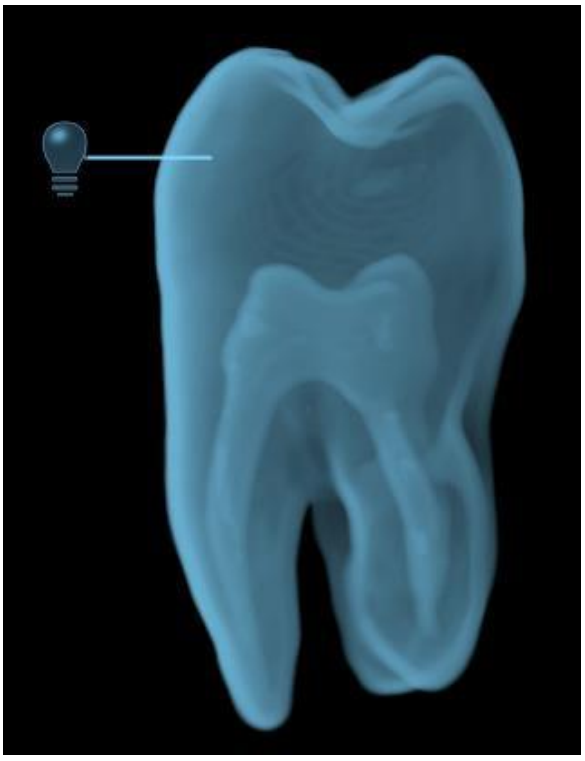
按照课程 16 的步骤，添加光点，并将 GI Light Parameter 中的 ao 颜色设置为黑色（＃000000）。

将鼠标悬停在光图标上，然后按空格键显示该指示灯的参数，将灯光类型(light type)更改为方向(direction)，并改变颜色。

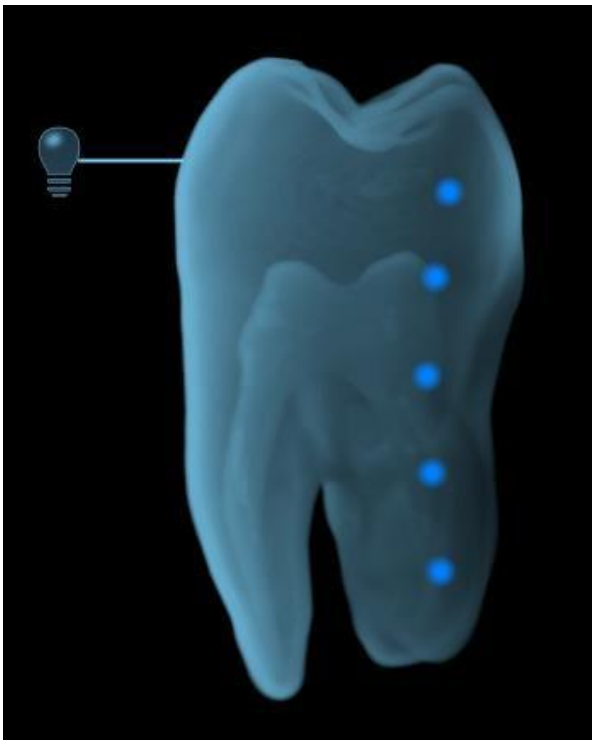


抓起灯图标并移动它，同时会从灯图标衍射出一条线， 该线显示了目前光线的方向。

移动线的任一端点将改变光线方向。可以按住鼠标中键移动整条线（光的方向不变）。



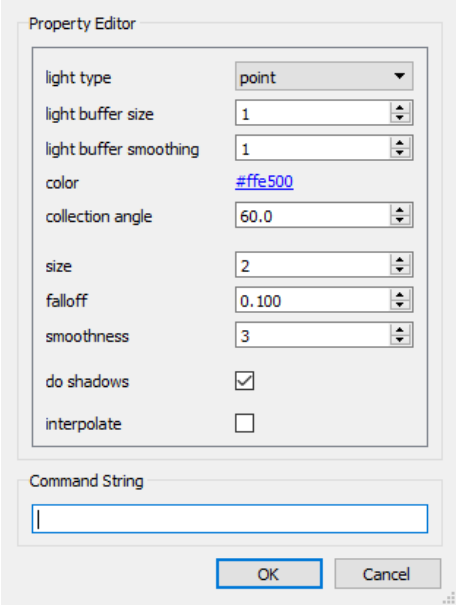
再添加几个点（用 Shift +鼠标左键单击可增加点）。



按空格键显示主命令对话框。 在命令字符串文本框中输入 addplight。所有的灯点会连成一条线。

将鼠标悬停在新添加的灯光上，然后按空格键显示其参数框。改变光的颜色。

将大小设置 (size) 为 2，衰减 (falloff) 为 0.1，平滑度(smoothness)为 3。按 OK 关闭参数框。

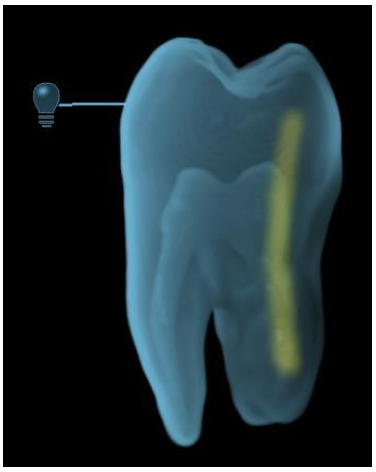


falloff 控制灯光的衰减。

smoothness 通过在用户指定的点光之间添加附加点光来控制样条的平滑度 - 值为 1 表示采用原始的灯串。

将鼠标悬停在灯串上，然后按 v 隐藏这些灯标。

灯串的影响范围较小，您会看到如下所示的粗亮线。



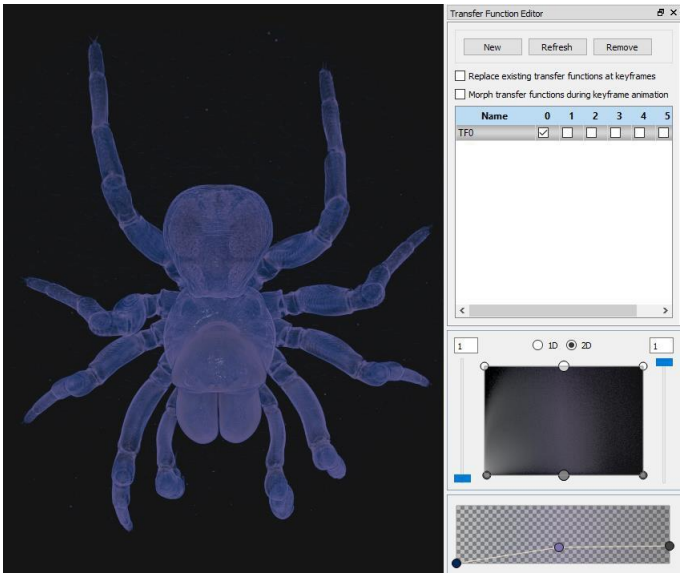
您可以随意尝试各种参数并查看渲染效果。

课程-18
发射光基础运用

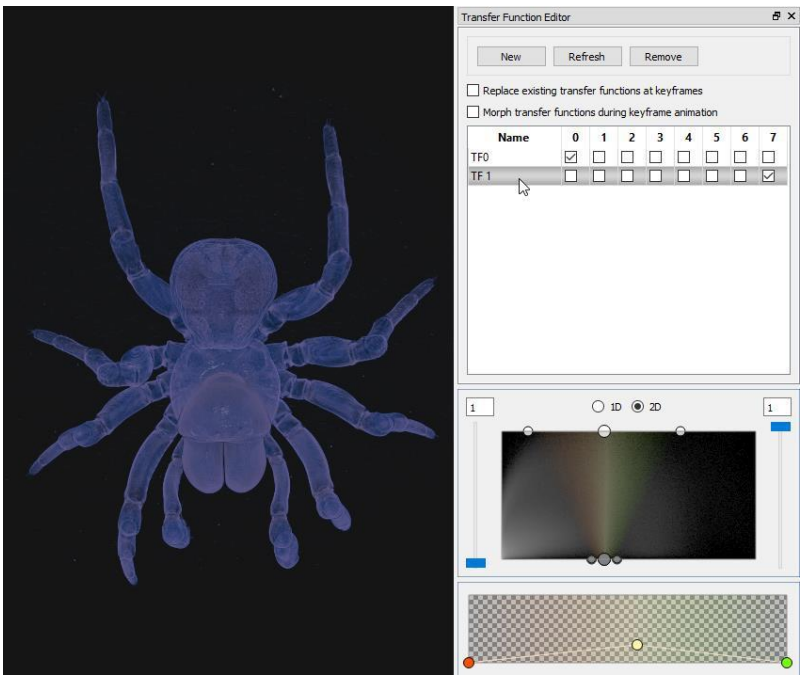
打开 Drishti Render.

加载蜘蛛数据：mousespider.pvl.nc 然后切换到 HiRes

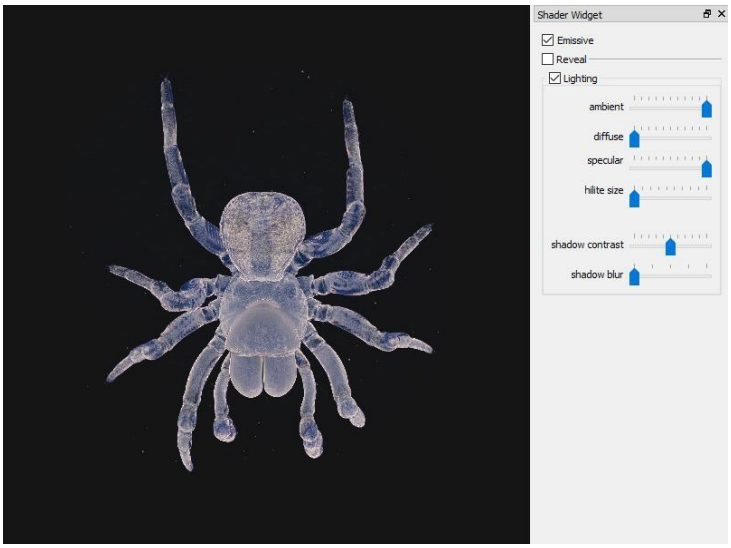
模式，设置一个较暗的传递函数，如图所示。



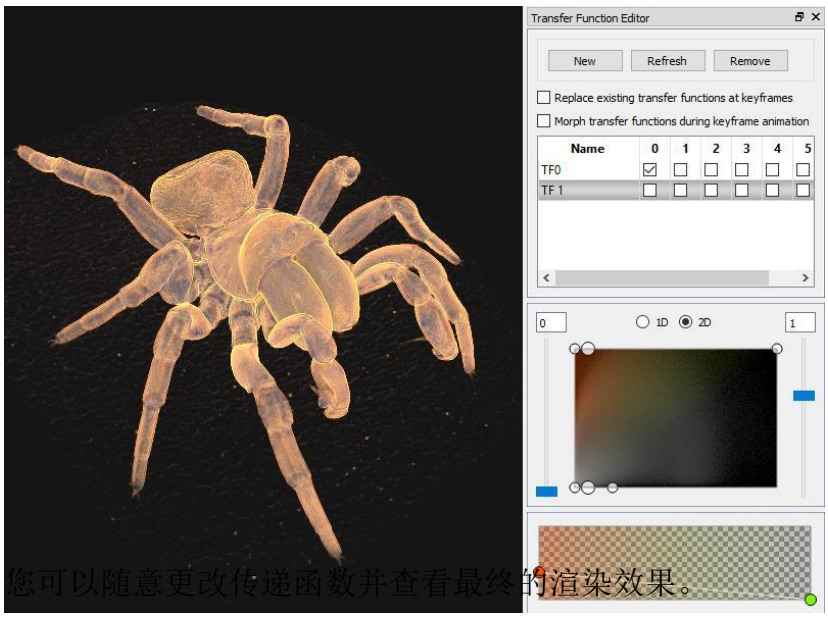
添加一个新的传递函数。 此函数将用于放射式呈现。关闭第 0 传递函数集，并打开最后一个传递函数集 - 在从课程中为第 7 传递函数集。



打开 Shader Widget（View-> Shader Widget）并打开 Emissive。您会看到蜘蛛的一些部分与其他部分不同。



受影响的区域是发射传递函数与用于确定不透明体素(opaque voxels, 在此课程中为 tfset 0) 的传递函数的重叠区域。该重叠区域增加了发射传递函数(emission transfer function) 所以更加明亮了。



您可以随意更改传递函数并查看最终的渲染效果。

*发射传递函数 (emissive transfer function) 的不透明度 (opacity) 和颜色会影响渲染。

*同时使用灯光和阴影对渲染效果并不会产生影响，发射光的渲染效果也是可见的。

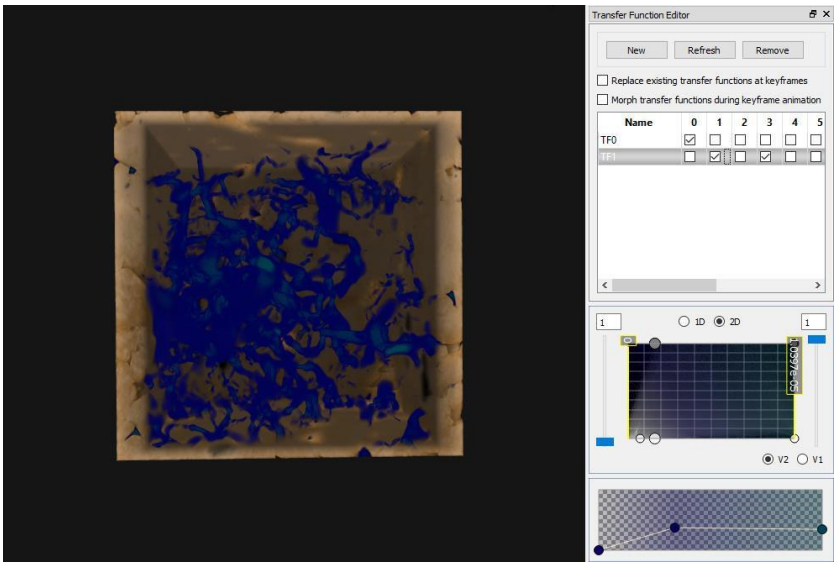
课程-19
发射光运用（针对多组数据）

打开 Drishti Render.

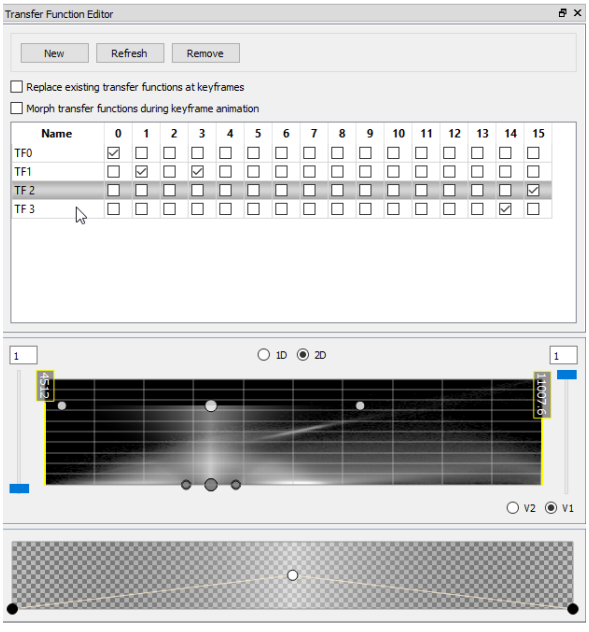
File->Load Volume->Load 2 Volumes（文件 - >加载卷轴 - >加载 2 卷轴）

将 block.pvl.nc 作为第一个卷轴，将 block_flow.pvl.nc 作为第二个卷轴。

按照课程 15 的步骤，创建一个混合块，以显示岩石内的流动。



添加两个新的传递函数，用于放射式渲染。将它们从 tfset 0 切换掉。将其中一个传递函数切换到最后一个 tfset - 在此课程中为 Set15。将另一个切换到倒数第二个 tfset - 在此课程中为 Set 14。

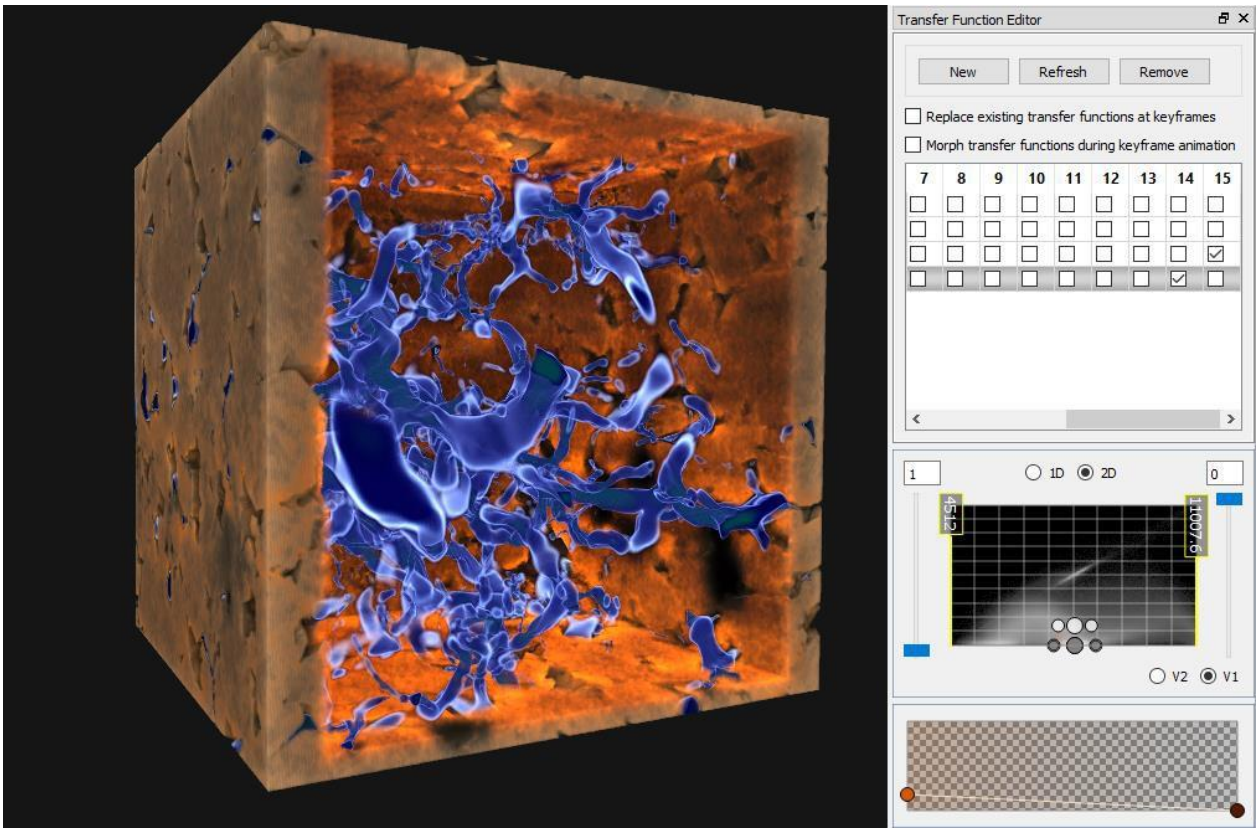


打开 Shader Widget（View-> Shader Widget）然后打开 Emissive。您会看到有一部分的数据的亮度与其他部分不同。

在倒数第二个集合中打开的传递函数将作用于第一个卷轴。
在最后一个集合中打开的传递函数将作用于第二个卷轴。（加载三或四个卷轴时，可以重复以上步骤。）

受影响的区域是发射传递函数与用于确定不透明体素（在本课程为 tfset 0,1 和 3）的传递函数的重叠区域。该重叠区域增加了发射传递函数，所以更加明亮了。

修改传递函数，以使两个卷轴中的所需区域都被点亮。



您可以随意更改传递函数并查看最终的渲染效果。

*发射传递函数 (emissive transfer function) 的不透明度 (opacity) 和颜色会影响渲染。

*同时使用灯光和阴影对渲染效果并不会产生影响，发射光的渲染效果也是可见的。