



# Vizard 3D建模原理和方法

2015年9月  
编译 李晓鸥



# 第一章 3D 模型介绍

Vizard可以直接读取多媒体文件并且制作互动效果。大多数情况下，需要用户自己建模。Vizard提供一些建模模块，例如3D图形，几何物体，2D和3D文字([3D shapes](#), [primitives](#), [2D/3D text](#))。但是创建一个内容丰富的虚拟现实环境需要用户使用第三方建模软件来创建三维模型。

## 推荐的建模渲染流程

WorldViz推荐用3ds Max建模，并导出Vizard原始OSG文件格式。Vizard对所有内置的OSG对象都支持，例如动画、材质、OSG辅助手段。OSG格式有二进制和文本版本。二进制的版本文件在Vizard里载入和渲染的会更快。并且该版本可以嵌入材质数据。文本版本可以在文本编辑器里编辑和调试。

OSG的导出插件可以在[WorldViz下载页面](#)里免费下载。

## 标准工业文件格式

下面的3D文件格式被广泛的用在工业界，Vizard可兼容：

格式
Collada (.dae)
FBX (.fbx)

## 支持部分特性的文件格式

下面格式的文件其特性智能部分支持。Vizard可以导入做渲染，但不是所有特性都能转化过来：

3D 格式类型	文件后缀
OpenFlight	.flt
VRML	.wrl
TerraPage (multi-threaded)	.txp
Carbon Graphics	.geo
Quake3 layers	.bsp
Quake animation models	.md2

AC3D	.ac
Maya	.obj
LightWave	.lwo / .lw
Performer	.pfb

## 建模工具

工具名称	推荐导出格式到Vizard
<a href="#">3D Studio Max</a> (commercial)	OSG
<a href="#">Maya</a> (commercial)	OSG Collada
<a href="#">SketchUp</a> (free/commercial)	Collada
<a href="#">LightWave</a> (commercial)	Collada
<a href="#">Blender</a> (free)	OSG Collada
<a href="#">AC3D</a> (commercial)	Collada

## 3D模型库

- [Turbo Squid](#) (付费)
- [3D Warehouse](#) (免费)

## 第二章 优化

对于所有虚拟现实即时渲染平台来说，优化都是必要的处理环节。这里不仅仅是针对Vizard平台。

到底需要多少优化工作？这取决于项目的目的和所提供的系统性能。对于预先渲染的3D图形，渲染一帧可以花几秒钟、几分钟甚至几个小时和几天。可以想象，在每秒60帧的速度下渲染一个场景收到很多因素的限制。重要的是必须要知道自己使用的系统的性能，并且能够高效的调用其性能。这里要重点介绍如何能够合理的分散这些对渲染速度有影响的因素，也就是做系统预算。

**注意：**渲染统计[Render stats](#) 可以在Vizard和Vizard的Inspector里显示出来。

### 绘制预算 Drawable Budget

一个可绘制的对象是共享材质的一个或者多个几何体物体。绘制时间和可绘制对象的数量相关。几千个绘制对象就会严重的影响绘制性能。可以通过在建模软件里将物体分组减少绘制对象数量。例如，一百个同样材质的盒子可以被归为一个组从而只产生一个绘制对象。这种方法的缺点就使在Vizard里对物体的位置改变是以组进行的，不能单个改动。下面几张在Inspector里的截屏图片演示了（A）两张长椅为一组（B）三张桌子为一组。B中的三张桌子即为一个绘制对象：



### 多边形预算 Polygon Budget

在即时3D渲染的应用中，大多数屏幕上的图形都被定义为一系列的三角形，而三角形来形成不同形态的物体。大多数游戏场景可以高效的渲染整个城市或者其他类型的大场景，而玩家在任何一帧都不会看到多于两百万的三角形在屏幕上。所以Vizard也需要遵循同样的建模和渲染规范。

- 如果一个三角形对于一个物体的轮廓没有任何贡献，那么你就不需要这个三角形。而应该将物体的细节由贴图过程来代替。但是也要避免过度修饰的细节化，取2-4米为一个参考距离，物体在这个距离范围之外就不需要额外的细节。
- 去掉物体内部和隐藏的几个体。使用者在屏幕上看不到的这些内容，计算机仍然会做渲染处理。甚至是一个隐藏在其他物体后面的物体，计算机也会处理。这些冗余内容都应该去除。精力应该放在可见物体上。
- 对于植物或者链锁篱笆墙这类物体，尽可能多的使用透明材质而不是用实际几何形状来建模。有些时候，一个小的植物就会用到上百万个三角形，但这么做却对整个场景没有增色的作用。所以最佳办法还是用透明材质来体现镂空的部分。有时间的话，可以多了解一下游戏中植物的建模方法。
- 在用户可达范围内，几何细节应该分布均等。而不应该有些物体表面几个体多余其他物体。这部分比材质分布处理要复杂些。
- 多出几何细节应该集中在用户关注的范围内。背景区域-用户从来都不会离进至50米的区域，几何细节没有必要过分修饰。

## 材质预算Texture Budget

控制材质预算相对简单。看一下你的显卡内存是多少，目标是控制在75%使用率一下。当材质加载到显卡的时候，他们以未解压缩的形式被存储。所以即使你有一个2048\*2048分辨率的图片，jpeg格式来讲那也只是几百k，而对于显卡来讲则是12兆。如果你超过预算，则渲染的帧率就会降低以便显卡能加载和卸载材质。所以你有三个选择：用更少的材质，用分辨率低的材质，或者用DXT压缩 ([DXT Compression](#))。

- 计算未压缩的图形文件大小很简单。大多数材质文件是每个像素为8比特。高度像素乘以宽度像素就能得到结果。8个比特为1个字节，你就知道一张图片为大了。再除以1024即为k，再除1024即为兆。所以2048\*2048的图片为12兆。
- 另外一个办法知道未压缩图片大小即在Photoshop里创建一个同样像素大小的空图片，在图片大小选项里即可显示值。
- 对于一些特殊的例子需要更高的显示发色数(bit-depth images).要计算16比特每像素的图片大小，就需要将像素数乘以2字节。对于32比特每像素的图片则乘以4。
- 材质的细节通常通过描述每厘米的像素数来决定。当像素用在材质中，他们就被称为材质像素 (texel)，每厘米的材质像素数被称为材质像素密度。对于一个场景的主要区域，理想的密度大于是2.56。也就意味着一个1米\*1米的分块地图 (tiling map) 分辨率为256\*256。
- 不是所有的场景都需要同等数量的近距离细节。一个远距离的100米的建筑物就可以用低材质密度来处理。
- 也要注意对于所有近距离处理的物体，他们的材质密度要相同。不然不同的密度会导致用户注意到不同的材质清晰度，会使用户偏离应该注意和交互的对象，影响体验感。

## 第三章 OSG导出器

OpenSceneGraph格式是Vizard本地格式。所以最好的输出3ds Max文件的方法就是通过OSG导出器，导出Vizard本地格式的模型文件。OSG的导出插件可以在[WorldViz下载页面](#)里免费下载。

**Note:** 关于OSG导出器的完整文档和详细介绍，可参见OSG Exporter在SourceForge的[链接](#)

### 导出

在**File**菜单中选择 **Export**，选择 **OpenSceneGraph Exporter**。默认情况下，输出文件会保存为二进制的OSGB格式。要导出为OSG文本格式，需要手动输入文件的后缀，例如myModel.osgt

### 预览

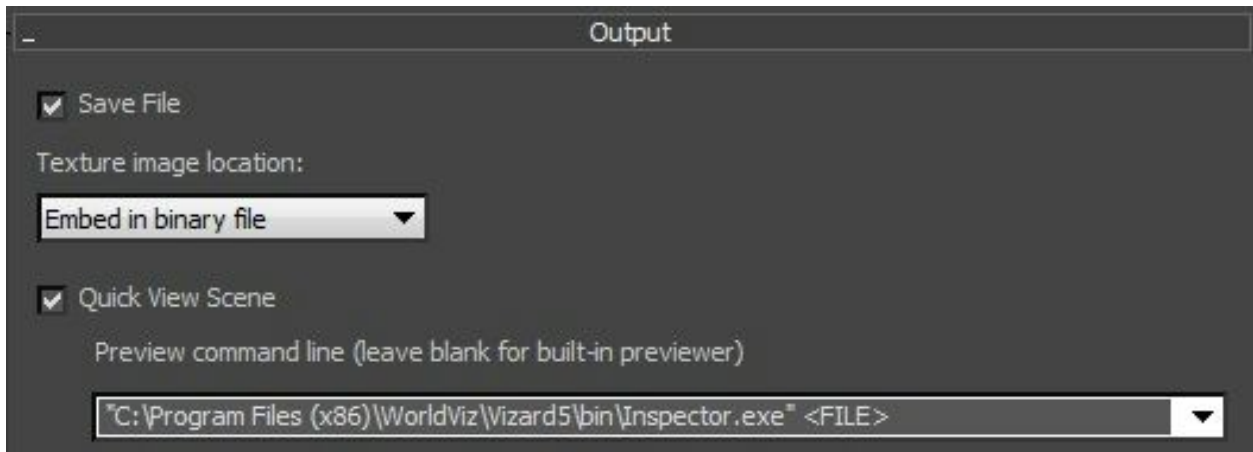
右键点击主工具栏的空白处打开自定义显示菜单，选择**Customize...**。在**Toolbars**选项中，从**Category**下拉菜单中选择 **OpenSceneGraph Exporter**：



拽托**actions** (例如**Preview Scene with OSG Exporter**)到工具栏中以便添加新的工具栏按钮：



打开**Export** 选项对话框  并设置Inspector为预览模式：



最后，点击预览按钮查看你创建的场景或选择的物体在Vizard中的样子。

## 所支持的文件格式

**osgb**: 比较新的OSG文件格式。它是二进制的格式。也意味着可以更快的加载并且支持内嵌图片信息。

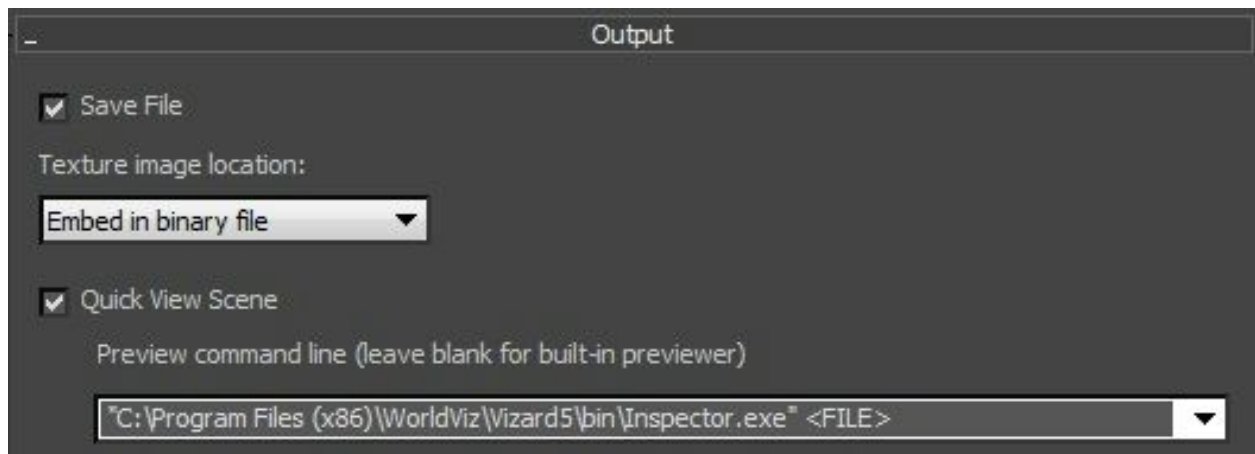
**osgt**: 这个格式是基于文本格式。导入的时间更长。但是这种文件可以在文本编辑器里编辑和修改。

## 导出器的选项

下面的这些图片显示的是针对Vizard的推荐默认设置。

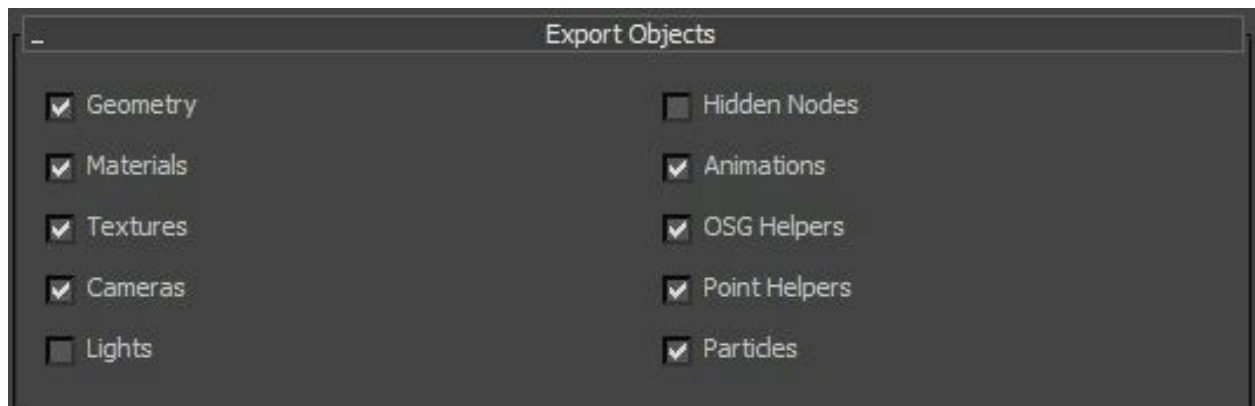


## 输出

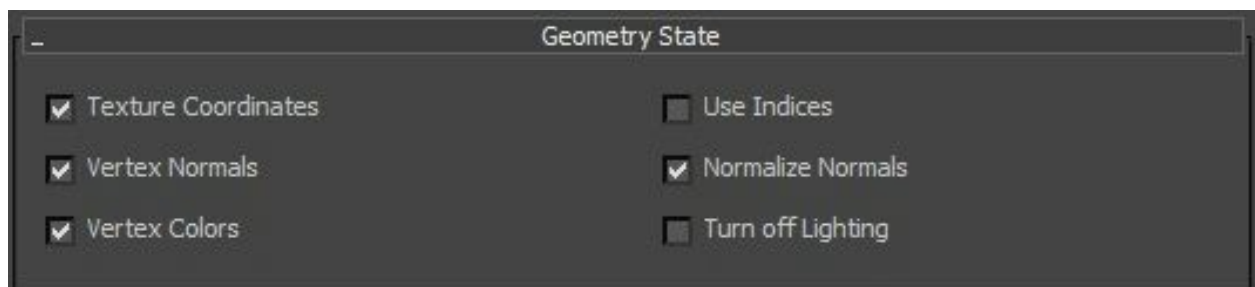


在修改过程中，这个选项可以更快地使用Inspector作为预览工具。

## 导出对象



## 几何状态

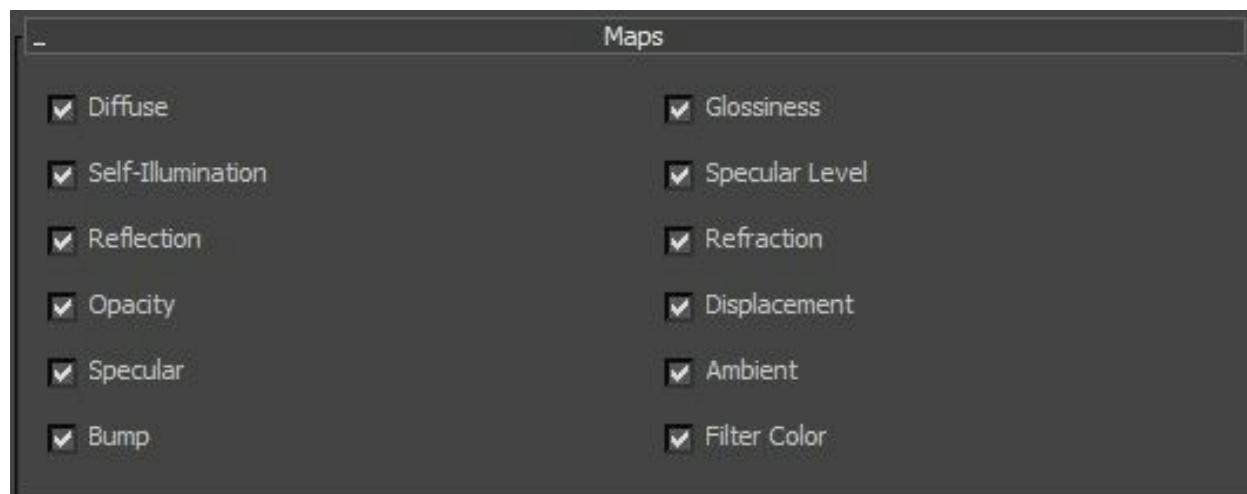


**运用指数Use Indices**：智能在特定情况下使用。可以造成速度变慢或其他问题。

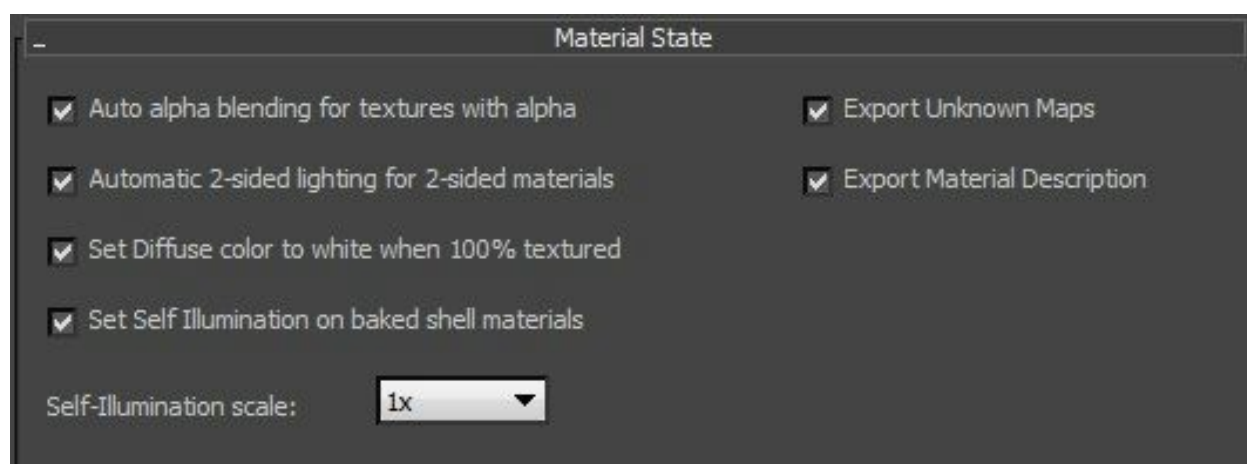


**关闭光源Turn off Lighting:** 关闭即时光源使用材质显示光源。当使用烘焙灯光时该选项需要勾选。

## 贴图

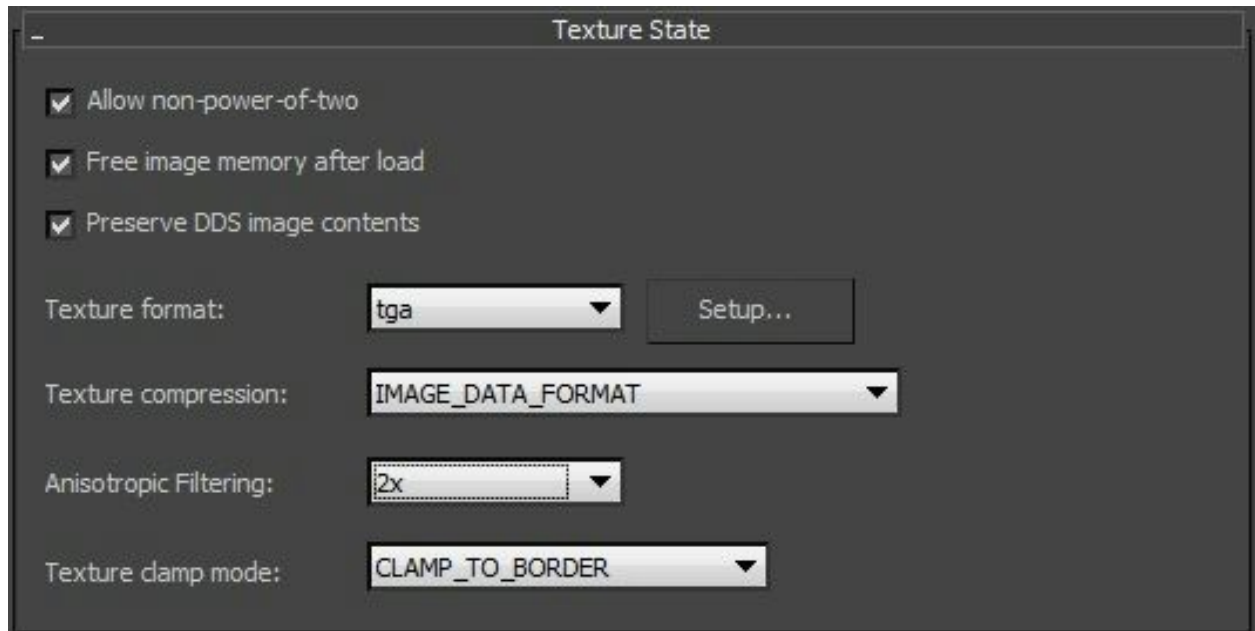


## 贴图状态



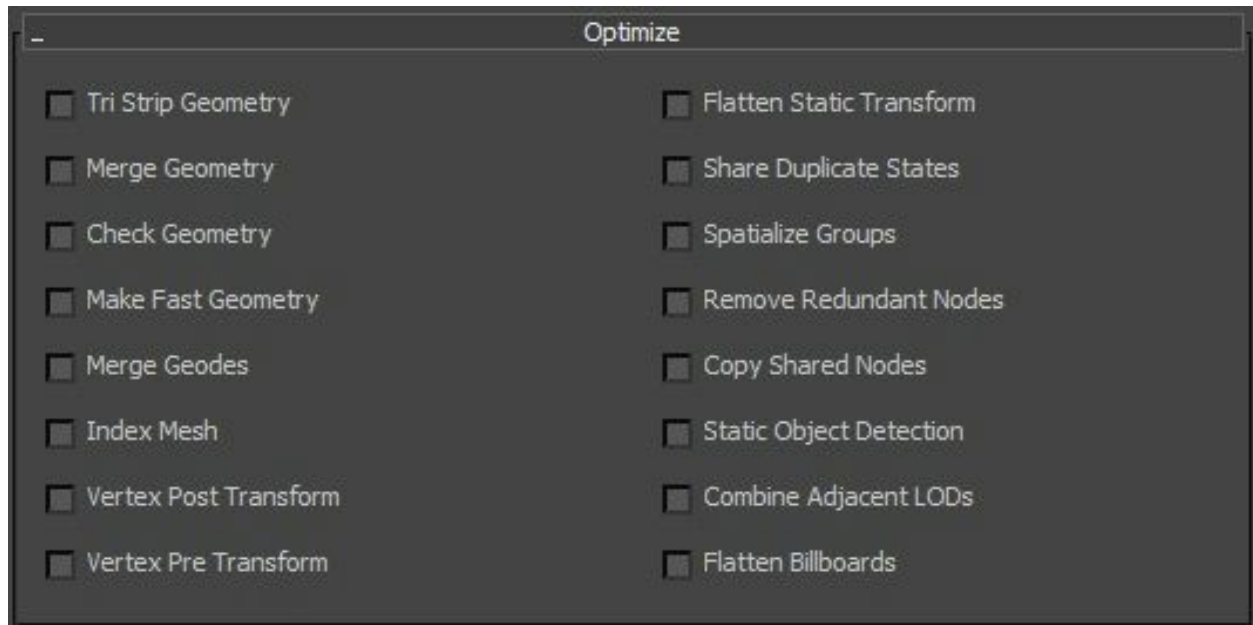
**导出材质描述Export Material Description:** 当使用Shader效果（[shader effects](#)）时该选项必须勾选。

## 材质状态



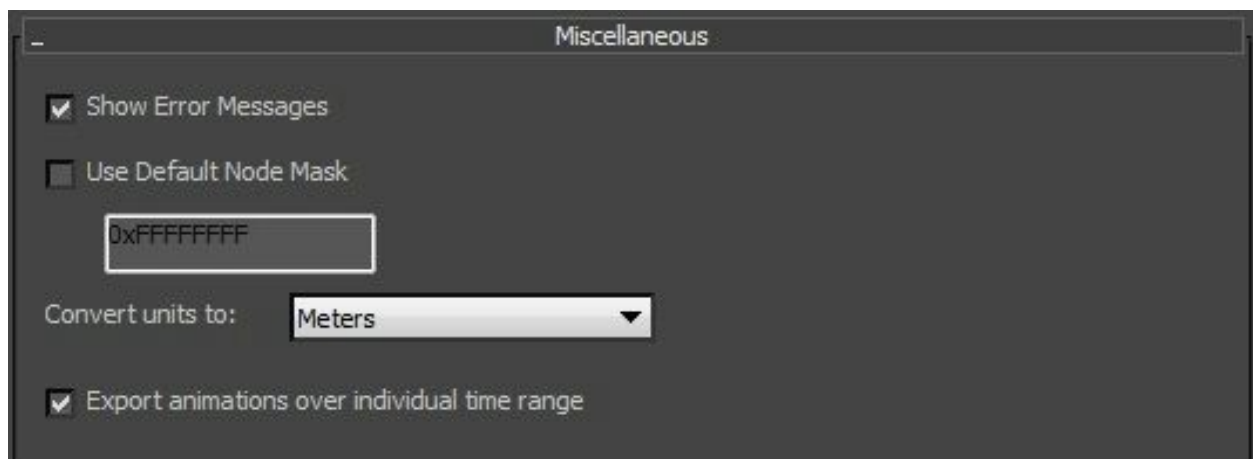
**材质压缩Texture Compression:** 使用材质压缩可以节省GPU资源。然而，压缩会有损失，会造成人为的渲染效应。当启用压缩时，图片会内嵌到二进制文件中，导出器会提前压缩图片并保存压缩信息到文件中。这个过程会减少导出文件大小，加速材质加载过程。

## 优化



**优化特性Optimization features:** 保持关闭直到需要的时候。许多特性会改变枢轴、层级和各种场景属性，可影响程序、材质和动画。

## 其他

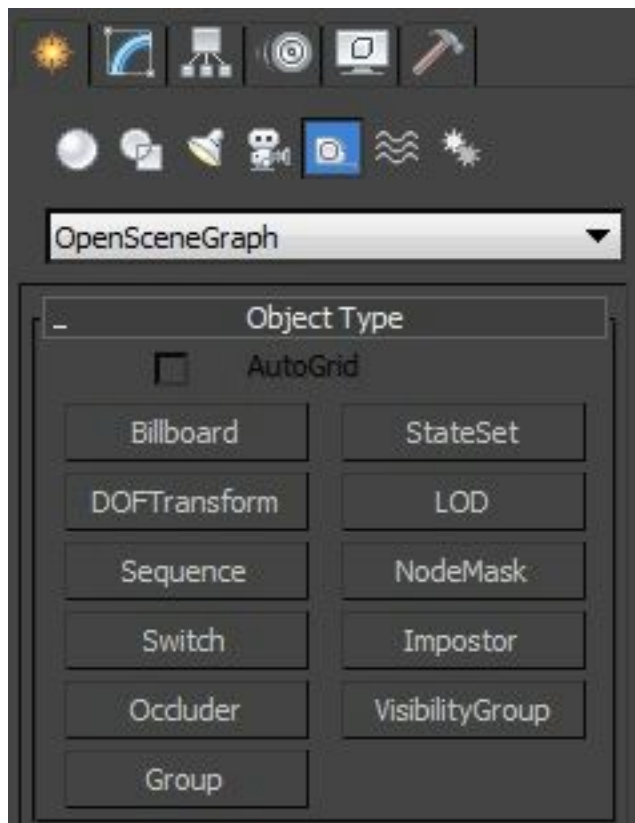


**转换单位到【米】 Convert units to: [Meters]:** 转换到Vizard中请使用米为距离单位。以此保证Vizard中显示模型大小的正确。

导出依据单个时距的动画**Export animations over individual time range**: 允许动画从初始帧到末帧的循环播放, 而不是使用全局时距。

## OSG帮助工具

共有11帮助工具:



1. [Billboard](#)
2. DOF Transform
3. [Sequence](#)
4. [Switch](#)
5. [Occluder](#)
6. Group
7. [StateSet](#)
8. [LOD](#)
9. [NodeMask](#)
10. [Impostor](#)
11. [Visibility Group](#)

## 第四章 Vizard光线阴影渲染方法

虚拟现实中真实的光效果很难实现实时渲染，会用到很多电脑资源。对于静态的物体也无需要实时渲染，提前在max烘焙灯光是推荐的方法。对于动态物体，实时灯光还是需要的并且能体现渲染的真实感。

### 完整贴图

完整贴图是用模型的原始材质和灯光做成的一个新的材质

#### 优点

- 能达到原始材质和灯光混合的最好效果
- 对于原始材质来说没有必要时标准贴图，只要新的烘焙的材料是标准的就可以

#### 缺点

- 烘焙必须在高分辨率下完成才能保持材质的细节。这么做的结果会导致模型文件很大。所以这种方法比较适合小场景例如室内，而不是打的户外场景。

### 光照纹理（lightmap）

光照纹理记录了光的亮度值。这些亮度值和漫反射材质结合来添加灯光和阴影效果。

#### 优点：

- 优化方面。原始材质的分辨率会被保存。文理的大小只会影响阴影和灯光的质量。灯光大部分由光滑梯度组成，所以不需要像漫反射贴图一样要特写的细节。
- 功能性。光照纹理可以将灯光和表面颜色分离开来，所以允许再不重新烘焙的情况下变换漫反射贴图。

#### 缺点

- 没有像完整贴图那么有真实感
- 需要所有的材质都必须为标准的

### 全局照明

在真实世界，直接光源会从物体反射回来，形成软性阴影，颜色混合和灰色区域，灰色区域还会被非直接光源照亮。在3D渲染中，全局照明技术可以计算光源反射来模拟这些效果。由于这些巨大的计算量，这种方法需要额外的渲染时间和计算机性能。不同的渲染工具（mental ray, v-ray）有自己不同版本的全局灯光。

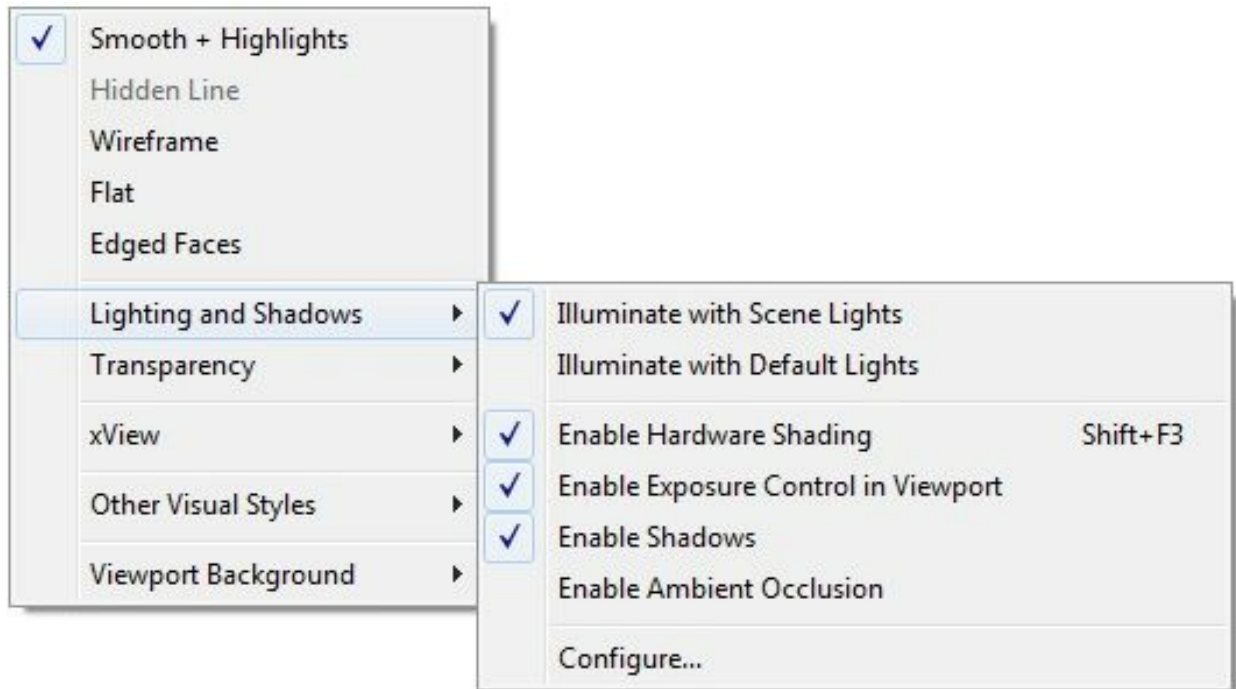
下面的教程用到了扫描器渲染器的默认设置。光源计算只计算初始光线如何碰到物体反射的。大部分用户对这种方法的渲染结果比较满意。如果需要更好的照片级的选人结果，就需要更好的高级光源方式。

## 设置视角渲染属性

Max硬件渲染器可以提供非常真实的光效果预览功能。这个功能确实会消耗电脑多余的资源，所以对于复杂场景不要使用这个预览方式。

右键点击视角标签来设置渲染方式：





### 添加漫反射光源

如果你寻求一个更迅速的烘焙方式而不是照片级的渲染效果，可以考虑使用简单的环境灯光而非物理上准确的全局照明。

没有环境灯光，会很难看到阴影区域的细节。在真实环境中，直接光源会有反射性，而且暗区域也会被非直接光源照亮也就是全局照明。计算整个效果会需要额外的时间和计算机性能。

Max的全局灯光漫反射光源不能输出为OSG格式。所以需要在漫反射模式中添加泛光灯Omni light。如下图所示，这种方式可提升其他灯光的亮度。

**Note:** 你必须场景中有其他非漫反射光源，否则只有漫反射光的话不会看到阴影效果。

**Light Type:** 设置为 **Omni**

**Multiplier:** 使用低值

**Decay Type:** 设置为 **None**

**Affect Surfaces:** 启用 **Ambient Only**

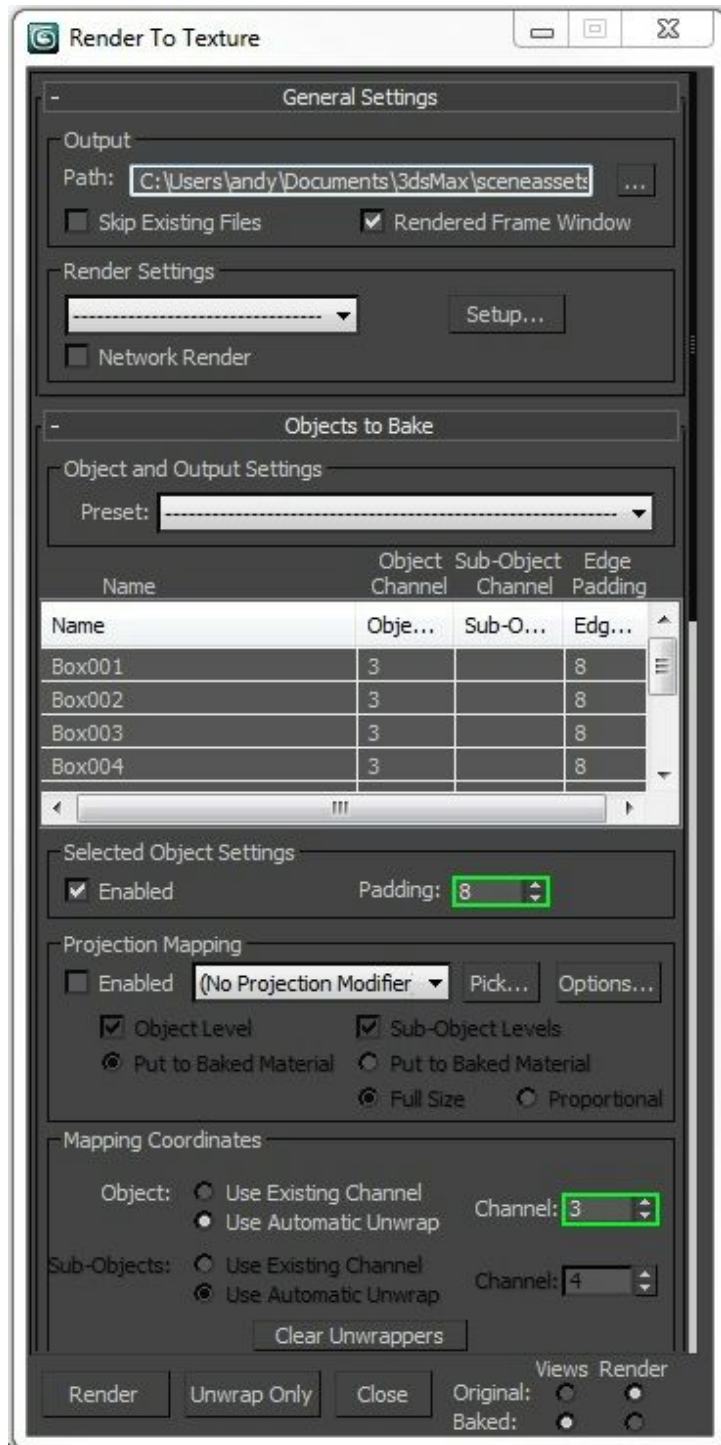


## 材质渲染 (第一步)

选择所有你想要烘焙的物体，这些物体会显示在Objects to Bake菜单下

为了保持一致性，对于烘焙的光照纹理和完整贴图，请使用贴图坐标系channel 3 (UVW Channel 3)

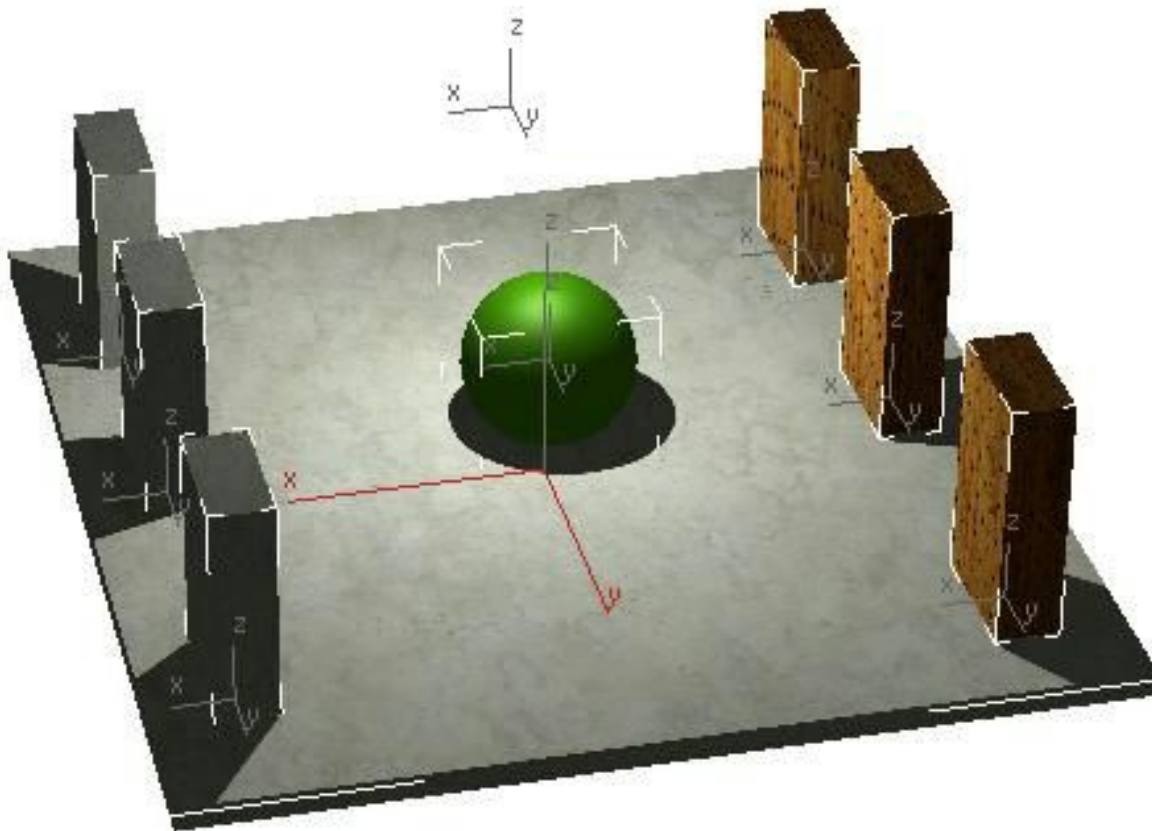
**Output Path:** 选择烘焙的图片要保存的路径。



**Objects to Bake:** 选择该视角下的物体将其放置在烘焙目录下

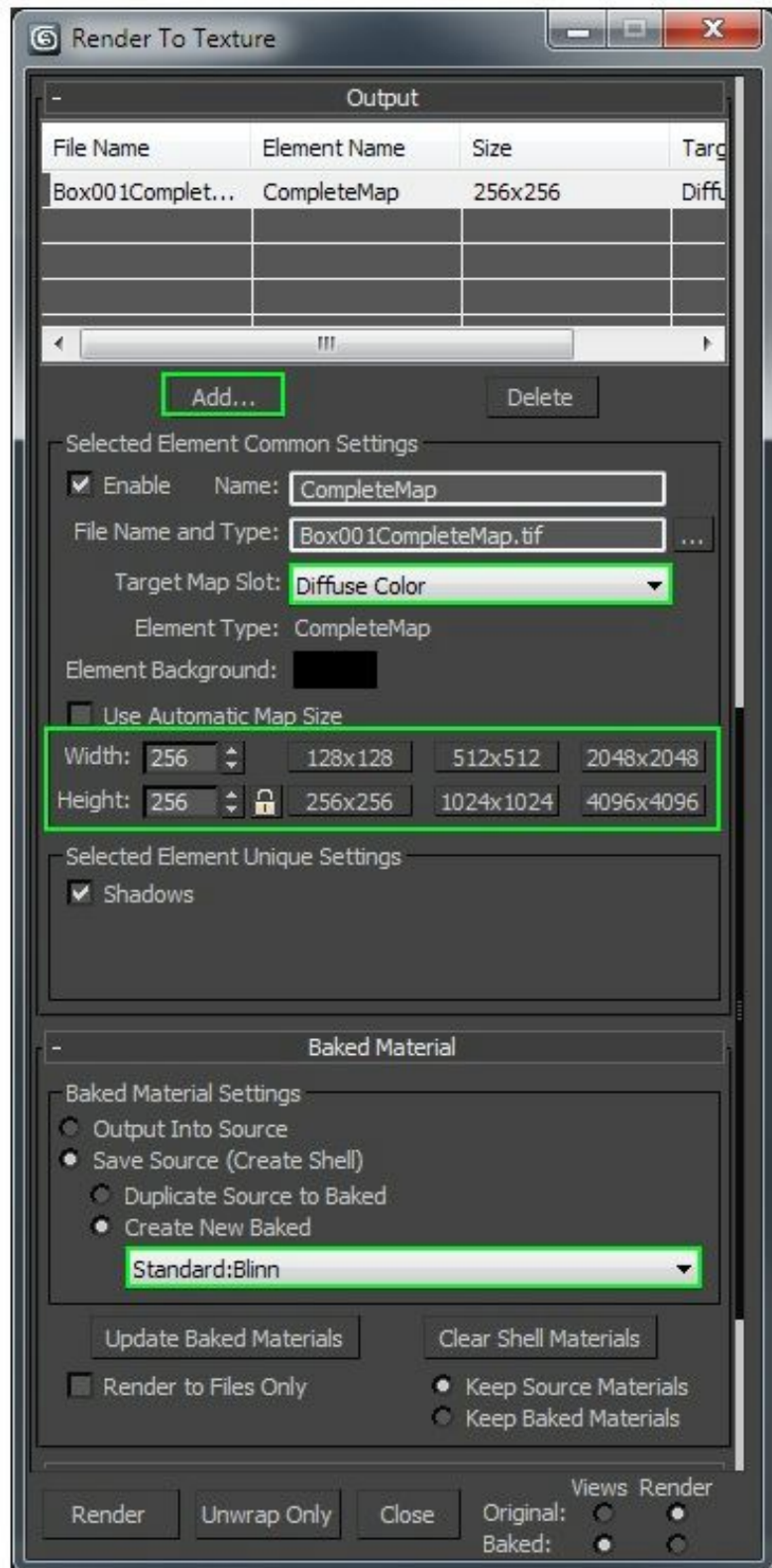
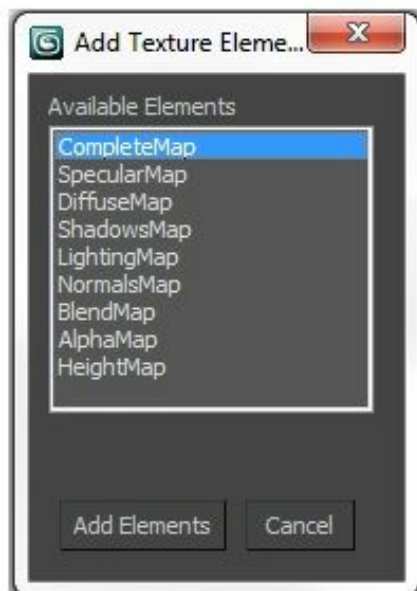
**Padding:** 推荐8或更高像素值

**Mapping Channel:** 推荐Channel 3



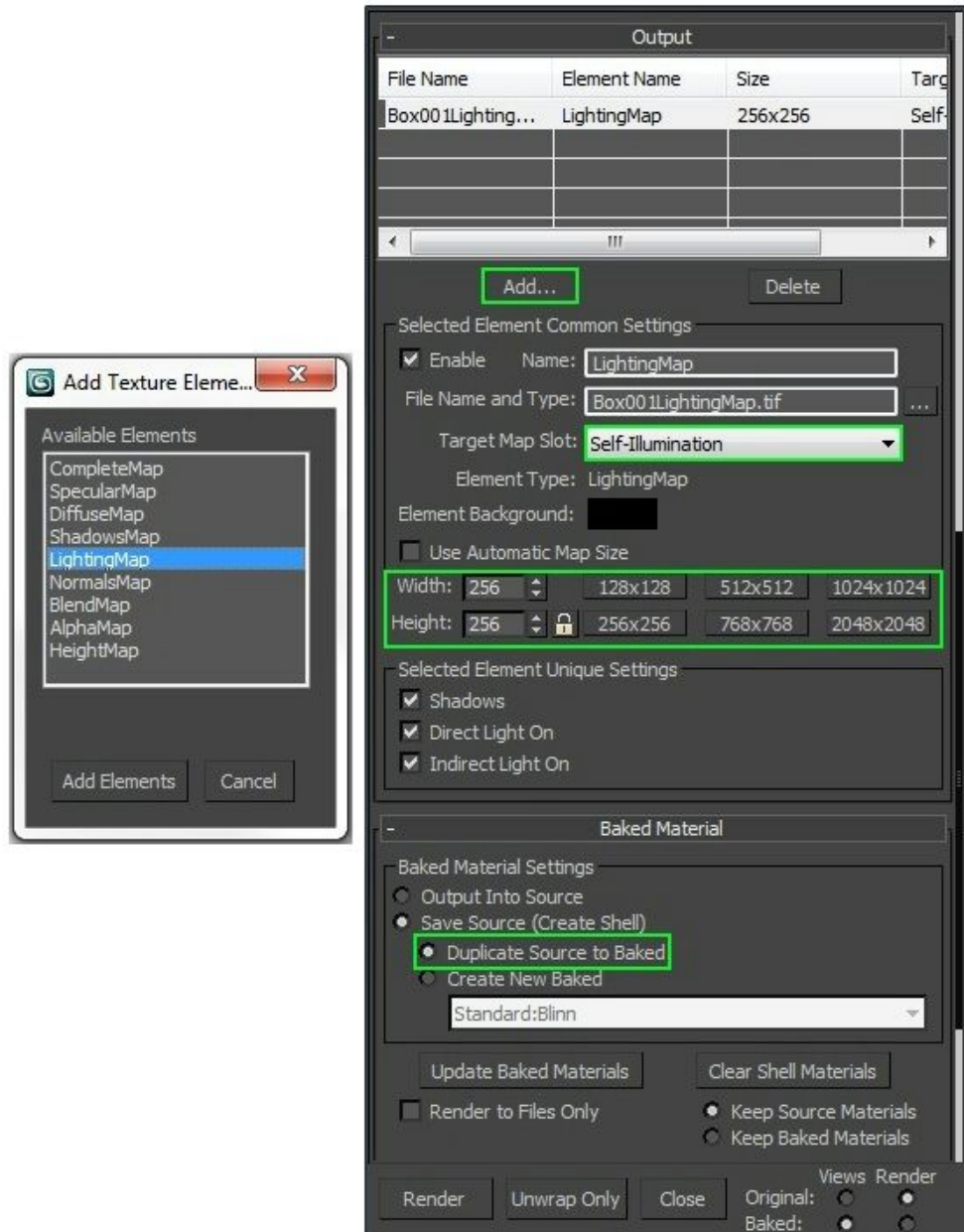
## 材质渲染（第二步）：完整贴图方式

1. 点击 **Add...**
2. 选择 **CompleteMap**
3. 选择一个贴图大小
4. 在 **Create new baked** 下选择 **Standard: Blinn**
5. 为**Target Map Slot** 选择 **Diffuse Color**



## 材质渲染 (第二步): 光照纹理方法

1. 点击 **Add...**
2. 选择 **LightMap**
3. 选择一个贴图大小
4. 选择 **Duplicate Source to Baked**
5. 为 **Target Map Slot** 选择 **Self-Illumination**

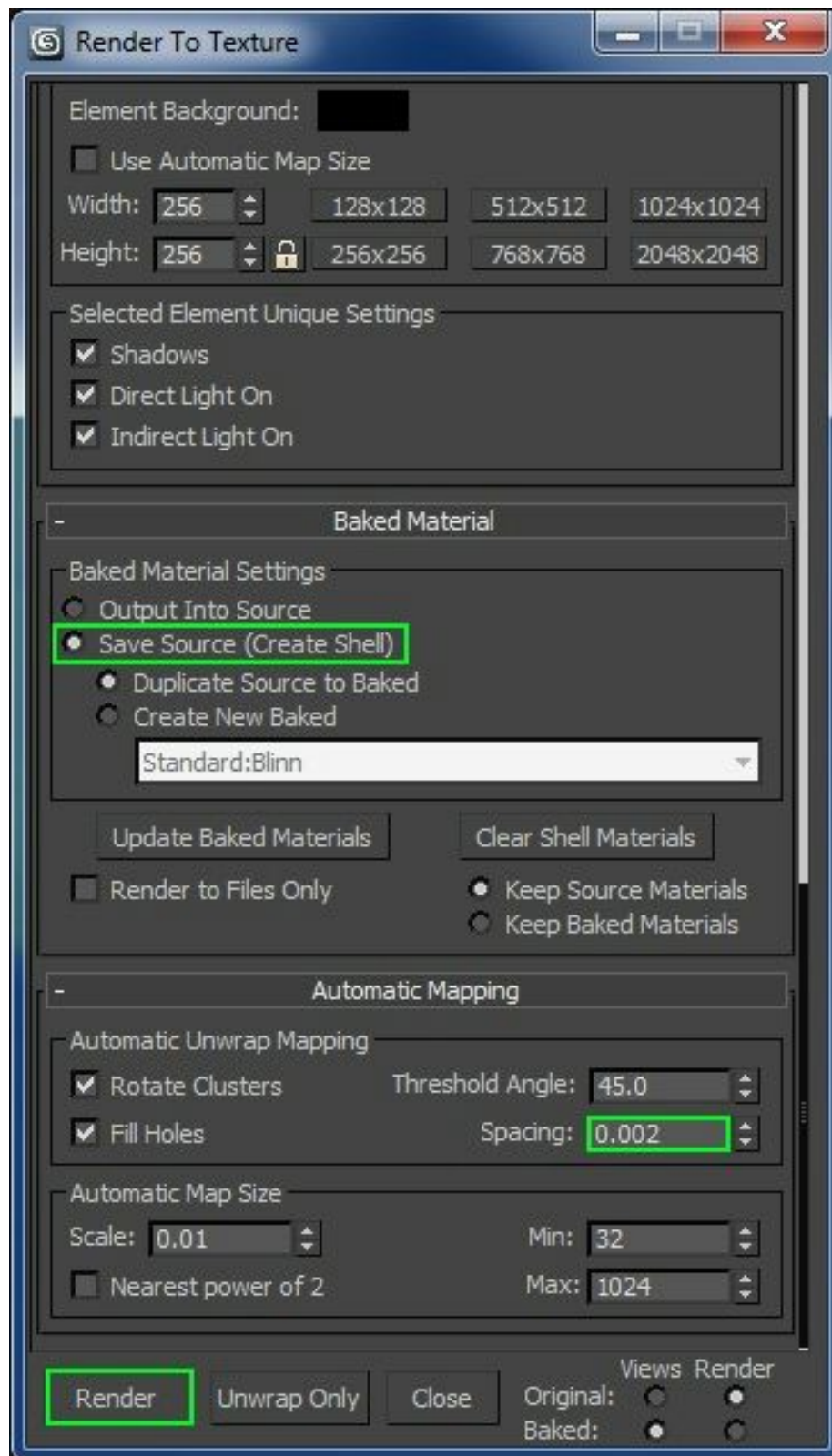


### 材质渲染 (第三步): 烘焙材质设置

在上面的两个方法中**Save Source**这个选项都提供了。这个选项将渲染结果保存为一个新的外壳材质。当你将场景输出到Vizard中，OSG导出超贱可以选择热河一个烘焙的外壳材质来覆盖原始材质。

通过减少在**Automatic Mapping**中**Spacing**的值可以帮助增加烘焙场景的分辨率。尽管贴图分辨率已经在之前设定了，贴图仍然会有没有用过的空间。通过减少这个值，更多的材质细节可以挤进到这个没有用的空间中。一个负面效应是多重贴图（在Vizard中会自动处理）会在缝隙空间中产生一些不理想的颜色。

1. 减少 **Spacing**的值以便获得更高效的空间和更高分辨率的材质
2. 点击 **Render**





## 输出设置

在OSG输出插件的对话框中：

- 在 **Geometry State**中启用 **Turn off Lighting**
- 确保所有贴图类型在**Maps** 选项中都被启用

## 在Vizard的Inspector中预览

Inspector是最快捷的预览烘焙效果的方式