

Computergraphik II

Prüfungsvorleistung 03

Texturen und Cubemaps

Abgabetermin: 08.05.2019 11:00 Uhr

Die Aufgaben sind von jedem Teilnehmer eigenständig zu bearbeiten, Gruppenarbeit ist nicht zulässig!

Möglichkeiten zur Abgabe:

- elektronisch über das OPAL-Portal: mytuc.org/cljs

Hinweise zur Abgabe:

- In dem Projektverzeichnis befinden sich das Skript `abgabe.bat`¹ welches alle zur Abgabe relevanten Dateien in das Archiv `abgabe.zip` packt. Überprüfen Sie dessen Inhalt aber dennoch auf Vollständigkeit!
 - 在项目目录中是Skriptabgabe.bat，它将与传递相关的所有文件加载到存档abgabe.zip中。但请查看其内容以获取完整说明！
 - 请仅提交您已更改的文件。
 - 请提交ZIP档案！
 - 如果您上传多个版本，我们只考虑最新的交付。
- Bitte geben Sie nur Dateien ab, die Sie auch verändert haben.
- Bitte geben Sie ZIP-Archive² ab!
- Falls Sie mehrere Versionen hochladen, beachten wir nur die aktuellste Abgabe.

Sollten Sie Fragen haben können Sie sich gerne per Mail (timon.zietlow@informatik.tu-chemnitz.de) an mich wenden.

在上一个练习中，您首次接触到纹理。这PVL的目的是以通常的方式扩展我们的框架纹理，从而进一步完善框架。

In der vergangenen Übung haben Sie den ersten Kontakt mit Texturen gehabt. Ziel dieser PVL ist es, in gewohnter Manier, unser Framework um Texturen zu erweitern, und damit das Framework weiter zu vervollständigen.

对于纹理的抽象，添加了一个新类CG2Texture。由于纹理本身缺少使用它们的上下文（我们可以使用纹理作为颜色值，法线或其他参数的来源），因此类CG2Material已经完成了写出纹理和固定材质参数，以及之后相关的着色器程序管理。这两个类的对象，如CG2Geometry和CG2Object对象，都在CG2Scene中进行管理和存储。

Zur Abstraktion von Texturen wurde eine neue Klasse `CG2Texture` hinzugefügt. Da Texturen alleine der Kontext fehlt, in dem Sie eingesetzt werden (wir könnten die Textur als Quelle des Farbwerts, der Normalen oder anderer Parameter verwenden) wurde außerdem die Klasse `CG2Material` bereits fertig geschrieben, welche die Texturen und feste Materialparameter und später auch das zugehörige Shaderprogramm verwaltet. Objekte beider Klassen werden, wie die `CG2Geometry`- und `CG2Object`-Objekte, in der `CG2Scene` verwaltet und gespeichert.

Obwohl uns die Abstraktion der Shaderprogramme noch fehlt, können wir mit einer Repräsentation von Textur und Material die Sponza Szene laden und darstellen. Die Sponza Szene wurde ursprünglich von Marko Dabovic entwickelt, und war lange Zeit eine häufig verwendete Testszene, wenn es um Beleuchtungseffekte geht.

虽然我们仍然缺少着色器程序的抽象，但我们可以使用纹理和材质的表示来加载和显示Sponza场景。Sponza场景最初由Marko Dabovic开发，在照明效果方面长期以来一直是受欢迎的测试场景。

Da die verwendete Geometrie und besonders die Texturen relativ viel Speicher benötigen werden diese **nicht** teil der Übungs- und PVL-Frameworks sein. Sie können ein Archiv `scene_data.zip` mit den entsprechenden Daten im Opal - unter Übungsunterlagen - herunterladen, welches Sie dann in die jeweiligen Projektverzeichnisse entpacken müssen.

由于使用的几何体，尤其是纹理需要相对大量的内存，因此这些不会成为训练和PVL框架的一部分。您可以使用Opal中的相应数据下载存档 `scene_data.zip` - 在练习文档下 - 然后您必须将其解压缩到相应的项目目录中。

¹Für Linux-Nutzer existiert ein entsprechendes Skript: `abgabe.sh`

²Nein - RAR ist kein ZIP!

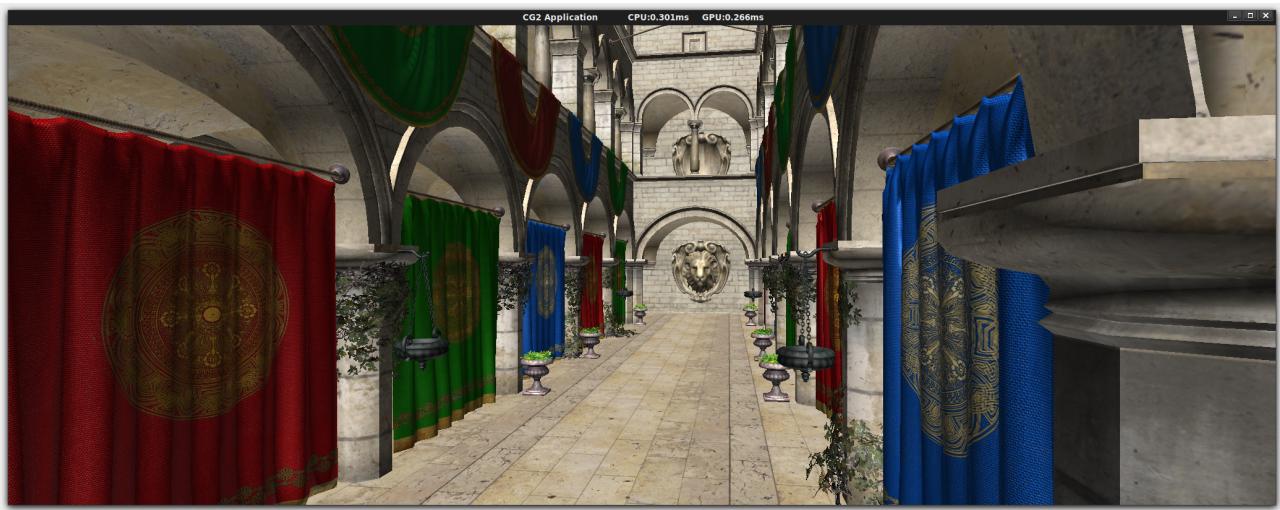


Abbildung 1: Sponza Innenhof

1. Texturen (2 Punkte)

如前所述，`CG2Texture`应代表纹理。为此，它存储纹理对象本身的名称，以及目标的名称，该名称对应于纹理类型（即第一次处于危险中的2D纹理，`GL_TEXTURE_2D`）。

Wie bereits angedeutet soll die `CG2Texture` eine Textur repräsentieren. Dafür speichert sie den Namen des Texturobjekts selbst, sowie den Namen des Targets, welcher dem Texturtyp entspricht (also für 2D-Texturen, um die es für's Erste geht, `GL_TEXTURE_2D`).

实现`CG2Texture :: createFrom (...)`方法，以便创建纹理对象并将图像信息从`img`复制到纹理。此外，应设置纹理参数以启用Mipmapping，并在必要时平铺纹理。

a Implementieren Sie die Methode `CG2Texture::createFrom(...)` so, dass das Texturobjekt erzeugt wird und die Bildinformationen aus `img` in die Textur kopiert werden. Außerdem sollen die Texturparameter so gesetzt werden, dass Mipmapping aktiviert ist, und die Textur, wenn nötig gekachelt wird.

maxAnisotropy函数参数有效地描述了各向异性过滤器的最大质量。调整纹理参数，以便相应地启用各向异性过滤。各向异性纹理过滤器是GL扩展，仅适用于OpenGL 4.6标准。原因不是技术限制，而是该过程获得专利的事实，并且OpenGL标准没有提出任何需要专利许可的要求。然而，由于各向异性过滤器大大提高了场景的质量，大多数制造商都支持这种扩展。可以在此处找到扩展的规范

b Der Funktionsparameter `maxAnisotropy` beschreibt effektiv die maximale Qualität des anisotropischen Filters. Passen Sie die Texturparameter so an, dass anisotropisches Filtern entsprechend aktiviert ist. Der anisotropische Texturfilter ist eine GL Extension und wurde erst mit OpenGL 4.6 Standard. Der Grund ist aber keine technische Limitierung, sondern die Tatsache, dass das Verfahren patentiert war, und der OpenGL Standard keine Anforderungen stellt, welche eine Lizenzerwerbung von Patenten erfordern würden. Da der anisotropische Filter die Qualität der Szene aber stark verbessert, wird diese Erweiterung von den meisten Herstellern unterstützt. Die Spezifikation der Erweiterung finden Sie hier.

加载`CG2Image`的图像现在不再强制四个通道！您可以使用`CG2Image :: getChannelCount()`查询通道数。

- Bilder, die mit `CG2Image` geladen werden haben jetzt nicht mehr zwingen genau vier Kanäle! Sie können die Anzahl der Kanäle mit `CG2Image::getChannelCount()` abfragen.
要使mipmapping工作，您必须生成适当的mipmap级别。
- Damit das Mipmapping funktioniert, müssen Sie die entsprechenden Mipmapstufen generieren.

2. Cubemaps und Sky Box (3 Punkte)

到目前为止，场景已经准备好了，只有天空由一个晶红色的表面组成。这项任务的目标是改变这一点。我们希望在所谓的Cubemaps的帮助下将无缝的天空渲染到场景中。要做到这一点，我们首先要教CG2纹理来处理立方体贴图。您将在OpenGL Wiki中找到相应的信息

Bis hier her ist die Szene soweit fertig, nur der Himmel besteht aus einer einzigen magentafarbenen Fläche. Ziel dieser Aufgabe ist es das zu ändern. Wir wollen mit Hilfe von sogenannter Cubemaps einen nahtlosen Himmel in die Szene rendern. Um dies zu erreichen, müssen wir zuerst der `CG2Texture` beibringen mit Cubemaps umzugehen. Sie finden entsprechende Informationen im OpenGL Wiki.

- (a) Implementieren Sie die Methode `CG2Texture::setCubeMapSideFrom(...)`. Wobei der Parameter `side` die Seite der Cubemap (`GL_TEXTURE_CUBE_MAP_POSITIVE_X`, ...) ist, die mit dem Inhalt von `img` initialisiert werden soll. Sie können dabei im Wesentlichen analog zur Aufgabe 1 vorgehen.

Hinweise:

实现方法`CG2Texture :: setCubeMapSideFrom(...)`。参数ID是Cubemap的页面 (`GL_TEXTURE_CUBE_MAP_POSITIVE_X`, ...)，应使用`img`的内容进行初始化。他们可以基本上类似于任务1进行。

注意事项：

确保只创建纹理对象一次 (`glGenTextures`)，这样在设置每个页面时就不会创建新的纹理，最后一次创建六个纹理，只有一个页面。

- Achten Sie darauf, das Texturobjekt selbst nur einmal zu erzeugen (`glGenTextures`), damit Sie nicht beim Setzen jeder Seite eine neue Textur anlegen und zum Schluss sechs Texturen mit nur jeweils einer Seite erzeugt werden.

- Als Wrap-Modus bietet sich hier `GL_CLAMP_TO_EDGE` an. 包装模式为`GL_CLAMP_TO_EDGE`。

不要忘记将成员 `text_target` 设置为 `GL_TEXTURE_CUBE_MAP`。

- Vergessen Sie nicht den Member `texture_target` auf `GL_TEXTURE_CUBE_MAP` zu setzen.

在 `cg2application.cpp` 中，已经使用 `setCubeMapSideFrom` 方法创建并绑定了一个纹理。着色器 `sky.vs.glsL` 和 `sky.fs.glsL` 用于渲染天空盒。In `cg2application.cpp` wird bereits eine Textur mit der Methode `setCubeMapSideFrom` erzeugt und gebunden. Die Shader `sky.vs.glsL` und `sky.fs.glsL` werden zum Rendering der Skybox verwendet.

为此，有效地绘制了远平面上的屏幕填充矩形。这确保了碎片仅在天空实际可见时才通过深度测试。要对立方体贴图进行采样，每个片段需要一个方向向量。这应该对应于 `Worldspace` 中可视光束的方向。

Dafür wird effektiv ein bildschirmfüllendes Rechteck auf der Far-Plane gezeichnet. So wird sichergestellt, dass Fragmente den Tiefentest nur dann bestehen, wenn auch wirklich der Himmel zu sehen ist. Zum sampeln der Cubemap benötigen wir einen Richtungsvektor pro Fragment. Dieser soll der Richtung des Sichtstrahls im `Worldspace` entsprechen.

- (b) Passen Sie den Vertexshader so an, dass in dem Varying `v_view_ray_ws` der Sichtstrahl für diesen Vertex gespeichert wird. 调整顶点着色器，以便在 `Varying v_view_ray_ws` 中为此顶点存储视图光线。

使用片段着色器中的变化 `v_view_ray_ws` 来对立方体贴图进行采样。

- (c) Nutzen Sie das Varying `v_view_ray_ws` im Fragmentshader um die Cubemap zu sampeln.

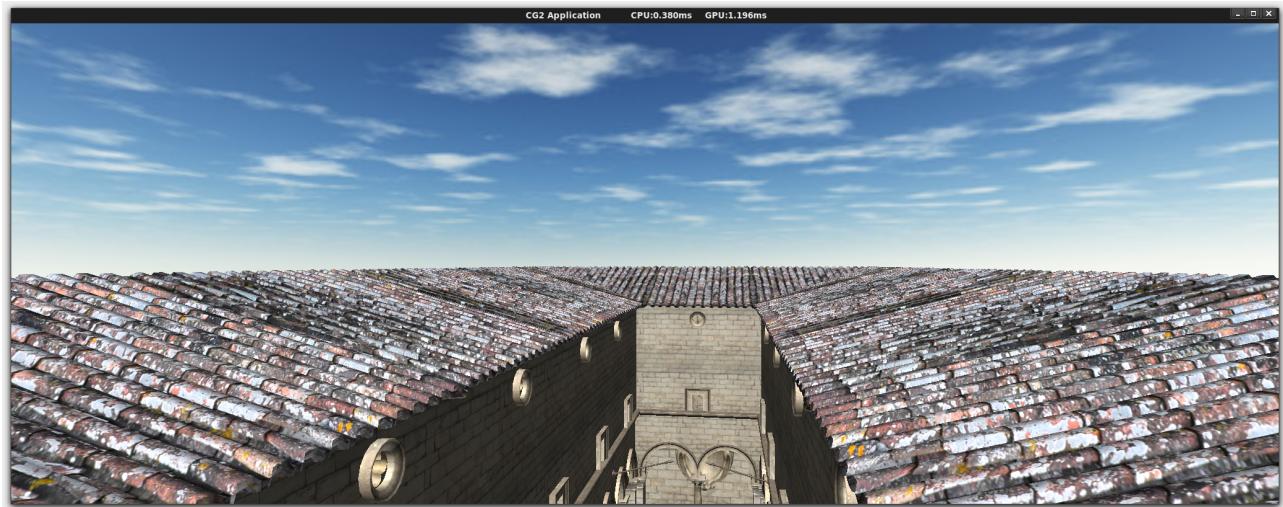


Abbildung 2: Über den Dächern von Sponza