

个人资料



cpyyin

访问：215987次

积分：2575

等级： 5

排名：第9493名

原创：38篇 转载：2篇

译文：0篇 评论：283条

文章搜索

文章分类

C/C++ (6)
DirectX (2)
Game (1)
Graphics (17)
OpenGL (5)
OpenKOK (6)
Visual Studio (2)
Windows (1)
心得体会 (1)

文章存档

2011年03月 (10)
2011年02月 (18)
2011年01月 (9)
2010年12月 (3)

阅读排行

从零开始重写KOK1(万王 (20370)
从零实现3D图像引擎：((19288)
当你的程序在朋友的机器 (18139)
VC2010中"Include Direc (14838)
从零实现3D图像引擎：((10604)
从零实现3D图像引擎：((8674)
从零实现3D图像引擎：((8571)

【专家问答】韦玮：Python基础编程实战专题 【知识库】Swift资源大集合 【公告】博客新皮肤上线啦 CSDN福利第二期

从零实现3D图像引擎：(14)背面消隐的三大陷阱

标签：引擎 存储 测试 blog c

2011-02-25 11:48

5449人阅读

评论(10) 收藏 举报

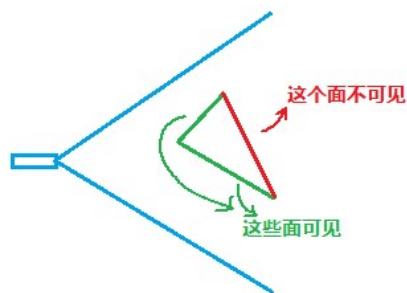
分类： Graphics (16)

版权声明：本文为博主原创文章，未经博主允许不得转载。

1. 为什么要背面消隐

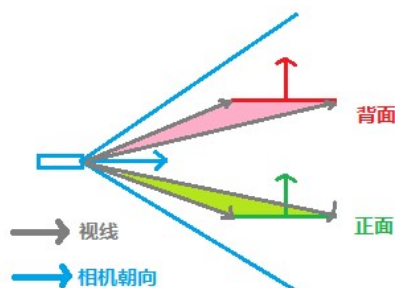
通过之前的DEMO，能够知道如果在渲染过程中多边形越多，那么要不处理的内容就越多，就越消费计算机的处理能力。对于物体来说，一般我们只看到它面对我们的面，可能不是正对着，但是肯定有很多面是完全背对我们的，我们就应该在渲染时跳过他们。这将减少大约一半的三角形渲染量。原理其实很简单，就是向量与平面的关系，这个原理我在参数化直线与平面的那篇文章中特意推导了整个来龙去脉。

我喜欢在2D俯视图中来分析问题，如下图：



2. 注意事项一：背面测试的是“视线与面朝向”的关系，而与相机朝向不直接相关

在刚接触背面消隐时，我以为是拿相机的朝向向量和面的法向量来求夹角，虽然也可以消除一些面，但发现是不对的，因为存在下面的这种情况：



从零实现3D图像引擎: (7380)

从零实现3D图像引擎: (7184)

从零实现3D图像引擎: (6647)

评论排行

裸辞三月之痒 (55)

从零实现3D图像引擎: (29)

从零实现3D图像引擎: (25)

从零实现3D图像引擎: (24)

从零开始重写KOK1(万王 (22)

从零实现3D图像引擎: (19)

从零开始重写KOK1(万王 (12)

从零开始重写KOK1(万王 (10)

从零实现3D图像引擎: (10)

从零实现3D图像引擎: (8)

推荐文章

*Android官方开发文档Training系列课程中文版: 网络操作之XML解析

*Delta - 轻量级JavaWeb框架使用文档

*Nginx正向代理、负载均衡等功能实现配置

* 浅析ZeroMQ工作原理及其特点

*android源码解析 (十九) -->Dialog加载绘制流程

*Spring Boot 实践折腾记 (三): 三板斧, Spring Boot下使用Mybatis

最新评论

裸辞三月之痒

ovenxp: 楼主能在乱流中捋出一丝精华, 最终摒弃杂念, 投身游戏, 很值得学习。我们好多人没能珍惜这个机会真是惭愧。...

从零实现3D图像引擎: (12)构建
麻子: @winmain0932:这里原点到平面上某个点的向量 就是这个点本身 $p - p$

从零实现3D图像引擎: (12)构建
麻子: 设视平面在各个象限的点分别为 其中 w 为视平面宽 h 为视平面高 $p1 = p2 = p3 = p4 = \dots$

从零实现3D图像引擎: (12)构建
麻子: 计算一下: , 我们可以把各分量缩放 $1/w$, 得: -----

从零实现3D图像引擎: (12)构建
麻子: @tktbxx:是的 这本书这么多年了 也不弄个勘误

从零实现3D图像引擎: (12)构建
麻子: @winmain0932:这里的所有 不包括远近裁剪面

从零实现3D图像引擎: (12)构建
麻子: 书中确实有不少问题

从零实现3D图像引擎: (2)画2D
小抱妖妖: 太厉害了。。

从零实现3D图像引擎: (4)三角
栗子冰激凌: 支持楼主给我们普及原理性的知识, 还附代码。而不用自己去啃书。有的人就是见不得别人好, 不用理他们就好了

从零实现3D图像引擎: (10)Hello
丁国华: 谢谢分享 学习了`(*^__^*)`

上面红色的面和绿色的面的法向量是相同的, 但是一个面是正面, 一个面是背面。因为相机朝向是固定的, 所以对于这种情况, 相机朝向与面法向量求夹角而得出的结论是不正确的。

在图中可以发现, 如果我们任取面上一点, 与相机的位置做连线, 那么这种情况就可以被区分出来。因为视线向量与法向量的夹角不同, 一个是锐角, 一个是钝角。(乍看都是锐角, 把他们的起始点都移到原点就能看出区别了)

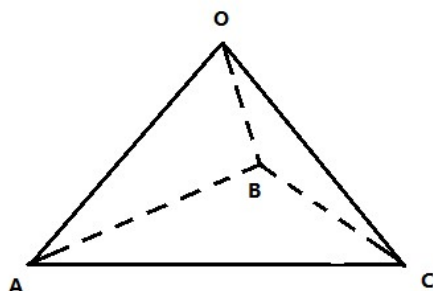
而且可以注意到, 无论取面上的哪一点来做连线都是正确的。

3. 注意事项二: 求面法向量的结果与“顶点的存储顺序”息息相关, 各面的顶点存储顺序必须一致

在做背面消隐时有一段时间怎么调都调不对, 我觉得肯定没问题的东西, 怎么会不对呢?

于是我在纸上推算了半天, 发现一个致命的错误——我每个三角形的顶点存储顺序是任意的, 造成法向量的不一致性。

看下图:



如果求一个面的法向量, 最简单的方法是得到两个向量, 然后把它们做叉乘。比如上面的三角形OAC来说, 可以用点A减去点O, 得到OA向量。点C减去点O得到OC向量, 然后用 $OA \times OC$, 得到法线向量, 这里隐含了一个重要的数学性质, 那就是叉乘不满足交换律。

即: $OA \times OC \neq OC \times OA$ 。

因为当我们拿到三个顶点的时候, 是不知道这三个顶点怎么排序的, 那么如果某个面1是逆时针来存储的, 如:O-A-C, 又有某个面是顺时针来存储的, 如O-C-A, 那么在求叉乘时, 结果会截然不同。

所以对于各个面的顶点存储时, 一定要约定好顺序, 比如都进行下面的步骤:

- 1) 从物体的外面来看这个面, 我们正对这个面
- 2) 逆时针来存储这些点(顺时、逆时无所谓, 但所有面应统一)

示例1:

对于面OAC, 我们首先在物体的外面来看这个面, 而不是在物体内部来看。然后逆时针来存储这些点, 则存储顺序为: O-A-C

示例2:

对于面ABC，**我们首先在物体的外面来看这个面**。然后逆时针来存储这些点，则存储顺序为：A-B-C，而不是A-C-B

正是因为这个原理，所以如果把3ds max的物体导出之后，就可以发现3ds存储的物体面中的顶点存储顺序都是满足上面一致性约定的。

4. 注意事项三：顶点存储顺序、向量叉乘顺序、视线向量的微妙关系

如果要写出正确的背面测试函数，就必须要把这三因素的关系搞明白，因为有一个不正确，你的结果就是相反的。(如果有两个搞错，反而会是正确的，有趣吧，但是可能计算的法向量是错误的，而影响以后的光照计算，所以我们还是要把这个搞清楚)。

这里给出一个正确的示例，可以有所体会：

- 1) 顶点存储顺序约定：从物体外面来看，逆时针存储
- 2) 在准备做叉乘时，可以拿到已经排好序的3个顶点（这里假设为 p_0, p_1, p_2 ），统一约定使用 $u = p_1 - p_0$ ， $v = p_2 - p_0$ ，也就是说从 u 转到 v ，也是一个逆时针旋转。因为 $p_0 \rightarrow p_1 \rightarrow p_2$ 是逆时针。
- 3) 求 $u \times v$ ，得到面法向量 n ，因为点的存储的统一性和 $u \rightarrow v$ 的转向相同，所以这时的面法向量是正确的，是朝向物体外面的。
- 4) 视线向量 $view$ 用相机坐标减去面上任意一点，这时这个向量是从三角面指向相机的。
- 5) 求 n 与 $view$ 的点积，如果点积大于0，则为锐角，是正面，如果点积小于0，则为钝角，是背面。这个性质的推导在向量函数库的那篇文章中有介绍。

5. 总结

至此我们的背面消隐函数终于是正确的了。在《3D大师》这本书中，虽然他代码是正确的，可是翻译出来的解释真的有时让人摸不着头脑，以后有时间还是看原版吧～这次如果我不是重新推导了一遍，光看那本书所说的文字真的是不会搞明白这里面的玄机。而且这次对写BLOG的认识又加深了，在写这篇文章的同时我又发现了新的认识，又发现了代码中隐藏在其中的逻辑错误，真爽。

6. 代码下载

这次的代码修改了上次的几个错误。

- 1) 所有的顶点使用统一逆序的存储
- 2) 使用视线而不是相机朝向来判断

项目完整源代码下载：>>[点击进入下载页](#)<<

效果和上次差不多，就不截图了。

顶 0 踩 0

上一篇 从零实现3D图像引擎：(13)把宽高比、透视投影矩阵、屏幕变换矩阵说透
下一篇 VC2010中"Include Directories" 和 "Additional Include Directories"的区别

我的同类文章

Graphics (16)

• 从零实现3D图像引擎：(15)...

2011-03-08 阅读 7386

• 从零实现3D图像引擎：(12)...

2011-02-21 阅读 7188

• 从零实现3D图像引擎：(10)...

2011-02-17 阅读 4099

• 从零实现3D图像引擎：(8)参...

2011-02-10 阅读 4051

• 从零实现3D图像引擎：(6)向...

2011-02-07 阅读 5562

• 从零实现3D图像引擎：(13)...

2011-02-23 阅读 4444

• 从零实现3D图像引擎：(11)...

2011-02-18 阅读 10611

• 从零实现3D图像引擎：(9)四...

2011-02-10 阅读 6362

• 从零实现3D图像引擎：(7)矩...

2011-02-09 阅读 3445

• 从零实现3D图像引擎：(5)3...

2011-02-07 阅读 5073

更多文章

猜你在找

深入浅出Unity3D——第一篇
iOS8-Swift开发教程
数据结构（C版）
软件测试基础
微信公众平台开发入门

转 第十七章 背面剔除与flash 3D 灯光 as30
画质直追照片新一代3D图像引擎
游戏引擎中三大及时光照渲染方法介绍以unity3d为例
游戏引擎中三大及时光照渲染方法介绍以unity3d为例
游戏引擎中三大及时光照渲染方法介绍以unity3d为例


坚持1个月，看美剧不用字幕

vipabc专业英美系外教,免费试听课程 每天45分钟,忙里偷闲学英语!


○ ○

查看评论


3楼 aa7629296 2011-03-02 09:27发表

 [e01] 虽然不是很彻底 但是已经明白了一些 正是我所很疑惑的


2楼 linjingtai 2011-02-25 14:58发表

 期待看到纹理的实现


1楼 农夫三拳 2011-02-25 14:31发表

 [e03]不错！不过用面的法向量只能判别凸多面体的后向面，对于凹多面体，可能要更多的测试，因为有的面是部分可见，部分不可见。


Re: cppyin 2011-02-25 18:53发表

 回复 fallStones：刚看了这个链接，讲的非常详细啊，赞


Re: cppyin 2011-02-25 18:48发表

 回复 fallStones：哦，我还没学到那儿呢，呵呵，多谢分享啦

Re: cppyin 2011-02-25 17:40发表

 回复 fallStones：消除被遮挡面的算法我还没有接触到，有没有相关链接指导一下哦，多谢

Re: 农夫三拳 2011-02-25 18:05发表

 回复 cppyin：http://wenku.baidu.com/view/0c21681c59eef8c75fbfb346.html
我用的是深度缓存算法。。正在写。还没搞定。。
给个联系方式，方便交流~~
cjian45@gmail.com

Re: [cppyin](#) 2011-02-25 17:39发表



回复 fallStones：看图了[e04]，这种面属于被其他的面挡住的情况(可能被自己，可能被别的物体)。背面消除没有去掉这些面的职责吧。虽然现在这种情况可以被画家算法修正，但我也觉得这些被遮挡的面也应该有函数去去除，应该是另外一种消除逻辑了吧？

Re: [cppyin](#) 2011-02-25 16:00发表



回复 fallStones：没太理解这种情况，我觉得不管是物体是凸是凹，对于一个三角形面，要不视线对着它正面，要不就是对着它背面吧。

Re: [农夫三拳](#) 2011-02-25 16:19发表



回复 cppyin：看图。。
http://hi.csdn.net/attachment/201102/25/2408546_12986219775m5N.jpg

您还没有登录,请[\[登录\]](#)或[\[注册\]](#)

* 以上用户言论只代表其个人观点，不代表CSDN网站的观点或立场

核心技术类目

全部主题 [Hadoop](#) [AWS](#) [移动游戏](#) [Java](#) [Android](#) [iOS](#) [Swift](#) [智能硬件](#) [Docker](#) [OpenStack](#)
[VPN](#) [Spark](#) [ERP](#) [IE10](#) [Eclipse](#) [CRM](#) [JavaScript](#) [数据库](#) [Ubuntu](#) [NFC](#) [WAP](#) [jQuery](#)
[BI](#) [HTML5](#) [Spring](#) [Apache](#) [.NET](#) [API](#) [HTML](#) [SDK](#) [IIS](#) [Fedora](#) [XML](#) [LBS](#) [Unity](#)
[Splashtop](#) [UML](#) [components](#) [Windows Mobile](#) [Rails](#) [QEMU](#) [KDE](#) [Cassandra](#) [CloudStack](#) [FTC](#)
[coremail](#) [OPhone](#) [CouchBase](#) [云计算](#) [iOS6](#) [Rackspace](#) [Web App](#) [SpringSide](#) [Maemo](#)
[Compuware](#) [大数据](#) [aptech](#) [Perl](#) [Tornado](#) [Ruby](#) [Hibernate](#) [ThinkPHP](#) [HBase](#) [Pure](#) [Solr](#)
[Angular](#) [Cloud Foundry](#) [Redis](#) [Scala](#) [Django](#) [Bootstrap](#)

[公司简介](#) | [招贤纳士](#) | [广告服务](#) | [银行汇款帐号](#) | [联系方式](#) | [版权声明](#) | [法律顾问](#) | [问题报告](#) | [合作伙伴](#) | [论坛反馈](#)

[网站客服](#) [杂志客服](#) [微博客服](#) webmaster@csdn.net 400-600-2320 | 北京创新乐知信息技术有限公司 版权所有 | 江苏乐知网络技术有限公司 提供商务支持
京 ICP 证 09002463 号 | Copyright © 1999-2014, CSDN.NET, All Rights Reserved