

http://game.watch.impress.co.jp/docs/series/3dcg/20090601_170531.html

西川善司为 3D 游戏粉丝的[生化危机 5]图形讲座（后篇）

MT Framework 1.x 的成熟图形魅力的新表现是什么？

2009 年 4 月收录



会场：卡普空总公司

在[生化危机 5]图形讲座的后篇中，要接触到的是最近在本连载中刊登次数增加的全局照明技术（Global Illumination）技术。还有，让大家看看现在的 3D 图形里的经典表现，在「BH5」中的[阴影生成]、[水面表现]等独特的实现方法，还有大家在意的 PS3 和 XBox360 版的区别等。

[生化危机 5]中的 Global Illumination 技术

在本世代的 3D 游戏图形中，开始倾向于重视环境光的处理。不仅有从设置在场景里的光源而得到的直接照明，还要考虑光源在复杂的反射后，布满空间的间接光照来实现照明的动感效果，使整体变得活跃。

这种复杂的照明效果被称作全局照明(Global Illumination)，通常被简称为 GI。[GI 方面的考虑]，说起来比较简单，但现在的实时 3D 图形的标准渲染管线中，只有直接光的照明(Lighting)手段。为了转换由 GI 来执行，必须要有特殊的方法来实现。

GI 最符合定义的实现是[光线追踪(Ray Trace)]等方法，但想要把这个在实时 3D 图形上实现，现在的 GPU 性能还不够，所以现在的主流是采用预先计算的模拟 GI 的方法。日本的游戏里，致力于采用这种模拟

GI 方法的有[索尼克世界大冒险]，这个在之前的连载里也有，在基础方面希望参考这篇。

在「BH5」中，玩家虽然可以体验到动感丰富的舞台（Stage），这些都是在关卡设计师（Level Design）（舞台设计）的巧手中完成的，实际上很意外的是做成了一个整体的“箱庭”（译者注：这里可以理解为把游戏里一章的所有舞台都做在一个大场景里）。也就是说，即使玩家是认为自己在自由行动，实际也只是在游戏系统的手掌上打转。虽然听上去不太好听，但这样构造的原因是为了让设计师和美工可以更容易控制。

场景是如此得接近玩家，为了让玩家的视觉印象感受深刻的地方，需要手动的进行环境光的设定。在表演舞台和影像的设置上，要在关键地方加入环境光。具体的，要先设定舞台中的重点位置的影响范围，再设定支配这个影响范围的环境光。

这个环境光本身，开始是在 MT Framework 的工具上，使用 CubeMap (6 面构造的的全方位纹理) 来进行设定，对进入影响范围的 3D 物体对象给予全方位 6 方向的环境光的形态。CubeMap 的构成，是把这个场景给予的动态和静态光的影响，还有精确的全局照明一起进行预先计算（也可以让设计师手工制作 CubeMap）。

[CubeMap]

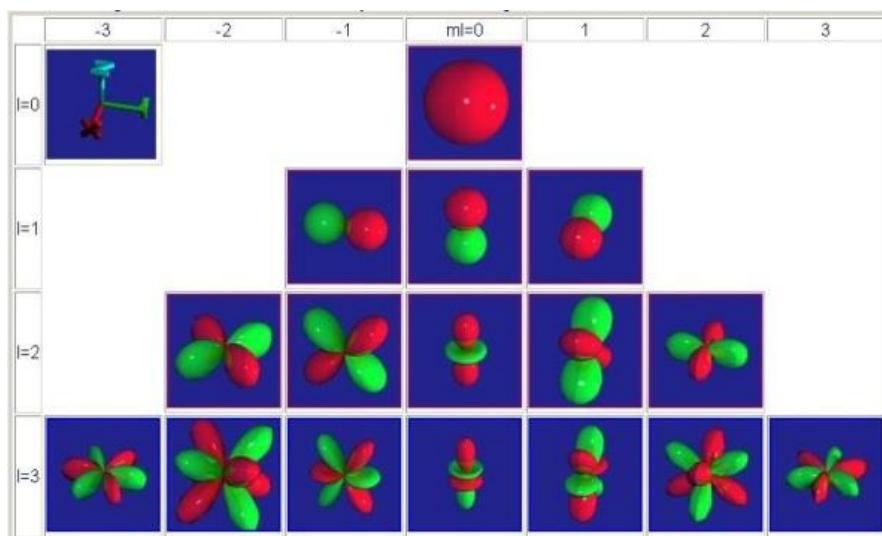




上图是 CubeMap 的影响范围的可视化截图，在范围内的部分表现有被特别设定的全方位环境光。
中图是最终效果截图。
下图是场景内设定的 CubeMap。

由于「BH5」舞台的箱庭结构，需要大量给予环境光的 CubeMap，而实际上这种 CubeMap 在运行时需要在内存管理上做大量辛苦的工作。所以在「BH5」中，把 CubeMap 用有 9 个系数的球谐波函数来做近似表现的实现。

关于球谐波函数的基本概念，可以参考本连载的[杀戮地带 2 (KILLZONE2)]。简单得说，就是把全方位的放射能量分布近似化的进行不可逆压缩的方法。「BH5」中，把 9 个系数分配到 RGB 来存放，虽然多少会粗糙点，但能够复原来自全方位的环境光。



译者注：杀戮地带 2 和生化 5 都是采用了 $l=2$ 的球谐波函数。

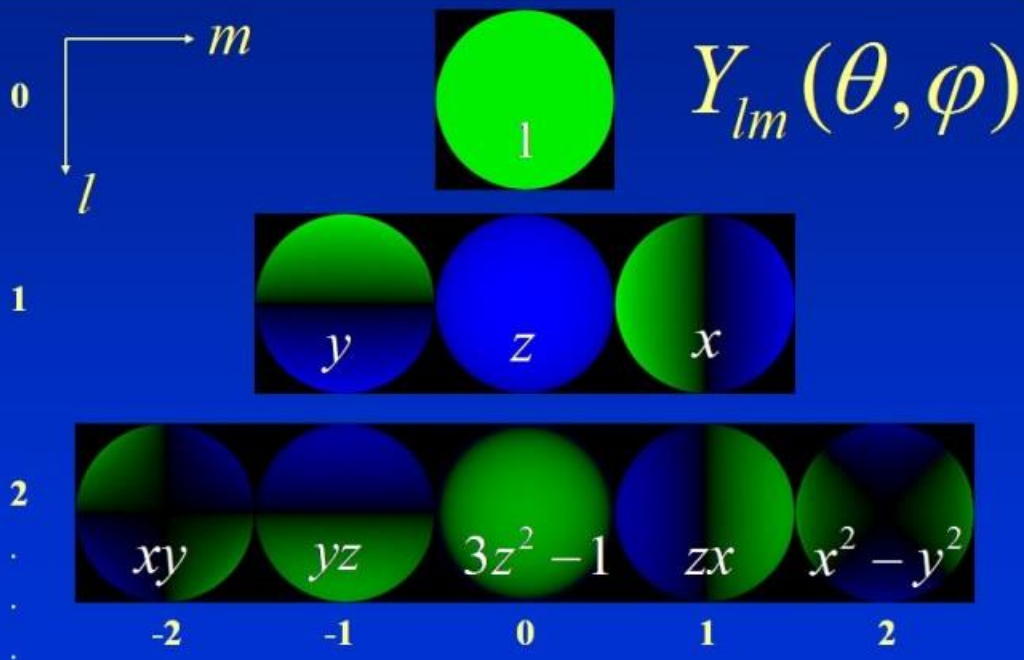
运行时上述的照明 (lighting) 是如何实现的呢？单纯的考虑，就是使用了 9 个系数的球谐波函数 (SH) 来复原环境 CubeMap，再通过这个，使用 IBL (Image Based Lighting) 的方法？

石田先生[不是这样，没有使用恢复为 CubeMap 的方法。使用的 SH Lighting 是比这个更简便的方法。原理上使用了斯坦福大学的 Ravi Ramamoorthi 的 [An Efficient Representation for Irradiance Environment Maps] 方法 (地址是 <http://graphics.stanford.edu/papers/envmap/>)。使用这个方法，从 3 次 (9 个系数的) 球谐波函数在 Pixel Shader 上复原全方位的环境光，就可以取得对应阴影处理对象法线的环境光。如果 9 个系数可以展开为 10 个左右的指令的话，那么恢复为 CubeMap 的方法在 Pixel Shader 上的成本更少。]

根据石田先生所说，可以分析出，就算使用 SPU 等芯片制作 CubeMap，也无法实现低 GPU 负荷的将 CubeMap 恢复的方法。顺便说一句，PS3 专用的 [杀戮地带 2]，采用了使用 SPU 来恢复 CubeMap 的方法。

[SH Lighting]

Spherical Harmonics



Hardware Implementation

$$E(n) = n^t M n$$

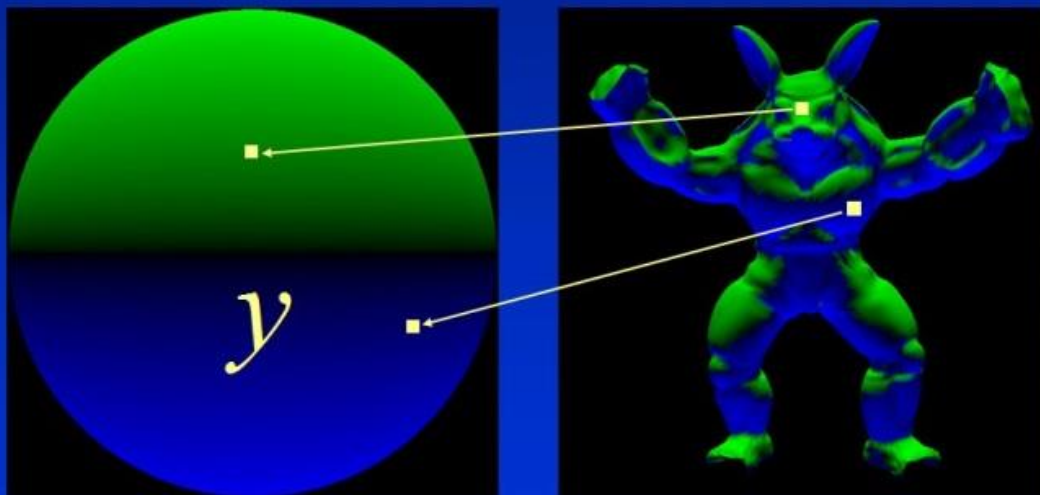
Simple procedural rendering method (no textures)

- Requires only matrix-vector multiply and dot-product
- In software or NVIDIA vertex programming hardware

```
surface float1 irradmat (matrix4 M, float3 v) {
    float4 n = {v, 1} ;
    return dot(n, M*n) ;
}
```

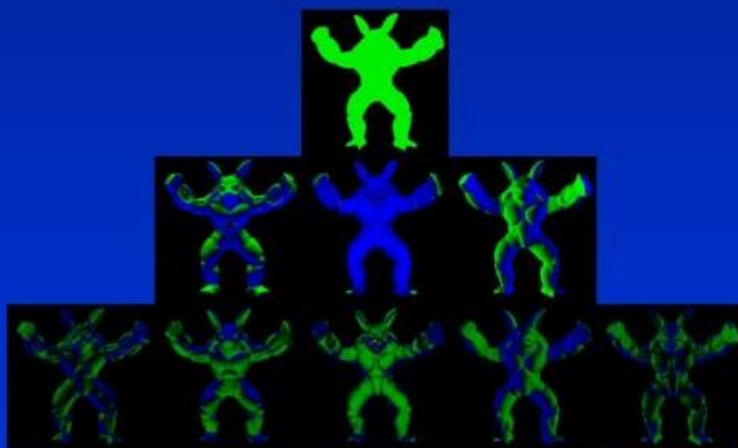

Complex Geometry

Assume no shadowing: Simply use surface normal



Lighting Design

Final image sum of 3D basis functions scaled by L_{lm}
Alter appearance by changing weights of basis functions



此图是实现不使用纹理(Texture)内存，直接从球谐波函数得到所需方向的环境光的方法的实现。



卡普空第一程序制作室 程序员 上田干夫先生

对了，根据场景的不同，特定的环境光 CubeMap 要可以设置在哪些区域(Area)产生影响和那些区域不产生影响。特别是屋外不能应用特定 CubeMap 的区域很多。这些区域，要选择使用全局设定的环境光 CubeMap。具体的，屋外情况使用全局环境光 CubeMap，要设定从天井方向而来的天球光和地面方向的地面反射而来的地面光。

还有，放在邻接的 CubeMap 之间的 3D 物体对象的情况，采用插值表现两者的球谐波函数系数，在通过球谐波函数，来计算出全方位环境光的方法。

上田干夫先生[环境光以外的静态光源，也要一起细致的设定影响范围。例如，从小屋向屋外看时，Bloom 的出现方式要由设计师来调整。在「BH5」中的情况，比起基于物理正确的照明，更重视为了在印象上使照明看上去更容易印象深刻的“基于设计师视觉正确的照明”。]

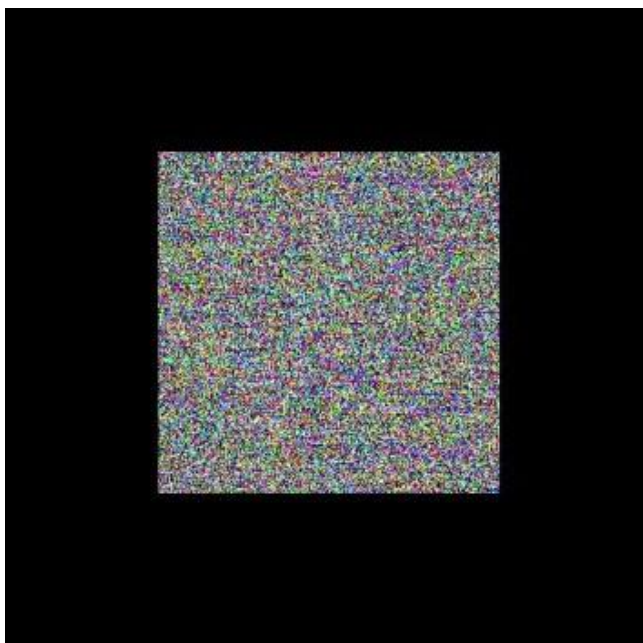
关于环境光，和连载里刊登的[合金装备 4(MGS4)]的手法非常相似。当然，虽然与[把预先生成 CubeMap 通过球谐波来获得]的实现等级不同，但舞台中环境光的配置，通过设计师来设定这些的影响范围等制作的风格上有相似的地方。虽然「MGS4」和「BH5」游戏类型上不同，但制作方对于画面要实现电影般的照明效果的重视，这方面是非常近似的，因此才有了一样的照明概念吧。

[由球谐波函数得到的环境光照明]



上图是没有球谐波函数的环境光，下图是根据球谐波函数得到的环境光（最终效果截图）

特意的让实时渲染看起来有预渲染风格？给予胶片质感的技术



[电视噪音 Television Noise]纹理的一种，与这个做 alpha 乘算合成后就会出现胶片颗粒(Film Grain)的效果。

在「BH5」里并没有把渲染的影像直接的显示，而是通过后处理（post process）添加了胶片的风味。这个是在电影等方面经常使用的手法，把色彩和渐变（gradient）用(艺术的风格)来调整的技术，「BH5」里，为了能做出电影的质感，把色彩过滤，渐变补正，对比补正，亮度补正等使用实时滤镜的形式实现并使用。

石田先生[在制作[失落的星球]时，实现了4个左右的独立滤镜，虽然不能太奢侈的使用，但在「BH5」中因为整合成一个来实现，所以把负荷减少到了失落的星球时的一成左右，这样就可以积极的利用了。]

平林先生[也加入了因为胶片的老化，在影像外围附近的红墨感，还有周边减光(影像的四个角附近会变暗，用镜头捕捉影像的特有现象)等效果。这些是将预先绘制好的上述效果的纹理，应用在 Sample 上。]

上篇有稍微介绍过，在「BH5」中，Movie 场景中会插入薄薄的胶片颗粒一样的噪声（银盐胶片引起的颗粒状的粒子噪声，虽然终归是“噪声”（译者注：正常考虑，噪声是不好的。），但为了胶片风味，有些情况要适当的加入）。这样，突出了胶片的质感，根据场合也可以看到好像动画压缩一样的噪声，故意的引起[这是预渲染影像吗？]这样的误导。

虽然笔者个人认为是胶片颗粒效果，不过在开发者里是把它称作[电视噪声]的效果来使用的。这里，把准备好的随机噪声纹理（Random Noise Texture），一边进行适当的纹理动画，一边做乘算的 alpha 合成的结果。虽然简单而普通，不过确实作出了前面所说的光学的老化效果和胶片老化效果等复合技术，作为视觉效果，风格上让人有相当类似的感觉。

在「BH5」中，附加内容（译者注：印象中是要通关后用奖励点数购买），不只是对应 Movie 场景，还有游戏本篇，准备了可以把这种电视噪声最大化的游戏模式，有兴趣的人可以试一试。

[电视噪声 Television Noise]





从上开始依次是

[胶片效果关闭/电视噪声关闭]

[胶片效果打开/电视噪声关闭]

[胶片效果打开/电视噪声打开]





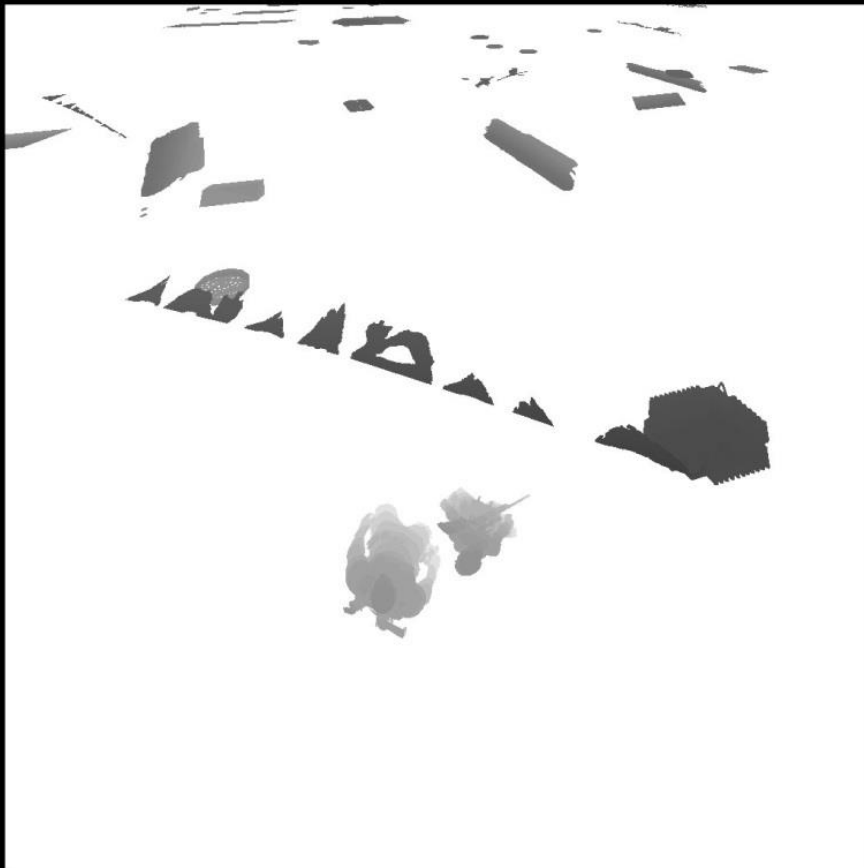
为了容易明白的局部放大图，带有明显的色彩和粗糙的附加点。

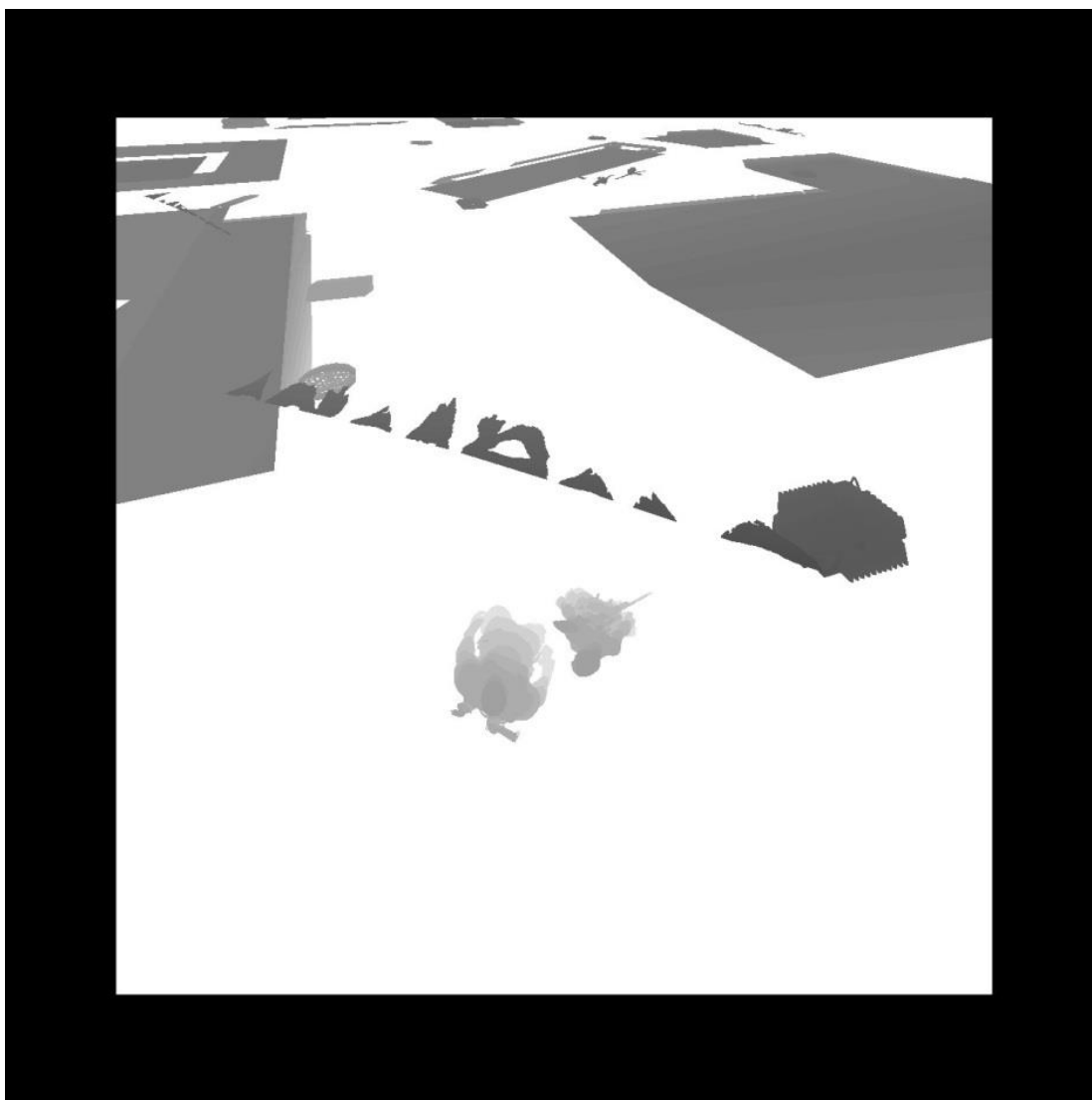
[持续进化中的「BH5」的阴影生成]

在「BH5」中动态阴影的生成，采用了和[失落的星球]同样的 Depth Shadow 技法改良型中的 1 种，LSM: Light-Space Shadow Maps 的技术。

在[失落的星球]中，实现方式是使用 1024 x 1024 纹理大小的 Shadow Map，分为远，中，近三个区域后生成 Cascade 的扩展。「BH5」中，根据场景不同采用 Cascade LSM 技法和失落的星球是一样的，但很多的场景基本上是单一的 Shadow Map。对比失落的星球那种一眼看去很广阔的场景，「BH5」是以遮蔽物比较多的场景为主体的，因为和后述的静态阴影烘培组合起来，就成了现在这种样子。顺便说一下，「BH5」的 Shadow Map 分辨率基本上是 1024 x 1024 纹理，Moive 场景中也有些情况使用的是 512 x 512 纹理。

[阴影生成]





在BH5中把Shadow Map按用途分为2张生成,中间和下面的图分别是用途1的Shadow Map和用途2的Shadow Map。

虽然不是全部,但很多不活动的静态背景物体对象,并不是动态的生成阴影,而是提前烘培到顶点上和作为光照贴图烘培到纹理里,两种方法一起使用来预先生成阴影。

那些本身不活动的房屋和墙壁的影子,虽然是事前生成烘培的,但这么做情况下,如果那个影子附近有活动的角色进入,房屋的影子就动态生成,很好的落在角色上。实际要让对应的房屋生成自身动态的Shadow Map,再向动态角色身上投射。影的生成在引擎部分,也控制了谁对谁投落什么样子的影子,反过来,静态的背景对象生成动态影子时,不会在烘培的影子进行二次投射。

【阴影生成】



上图是只有顶点颜色的画面，没有任何影子。

下图是只有顶点颜色和光照贴图的画面。可以看到基于纹理烘培的影子。



上图是在顶点颜色和光照贴图，加入了生成的动态影子的画面。
下图是最终画面。



上图是把静态物体对动态角色生成阴影关闭的截图，脚下的影子是事前生成的烘培影子。
下图是静态物体对动态对象生成影子投射的最终效果截图。

笔者对于「BH5」的阴影生成，感受最深刻的就是影子的颜色，开发组虽然说明了是和[失落的星球]的阴影生成使用了基本相同的处理。不过如果仔细得看，实际上「BH5」的影子在悄悄的进化着。

在[失落行星]时，因为在 LSM 上只描画乘算生成的影子和透射位置的影子颜色，所以影子“只是变暗了”，然而在「BH5」上，影子的部分中就像是很好的进行了环境光照明一样。正因为是软影子，虽然说影子的边通常是模糊的影子，但在「BH5」中影子的颜色是真实的浅色。

动态生成的影子，是由这个场景所的代表的光源所生成的。这个影子是通过遮蔽了从代表光源而来的光得到的，所以也应该受到间接光(即环境光)的影响。到现在为止的很多实时 3D 游戏图形中，明亮的部分

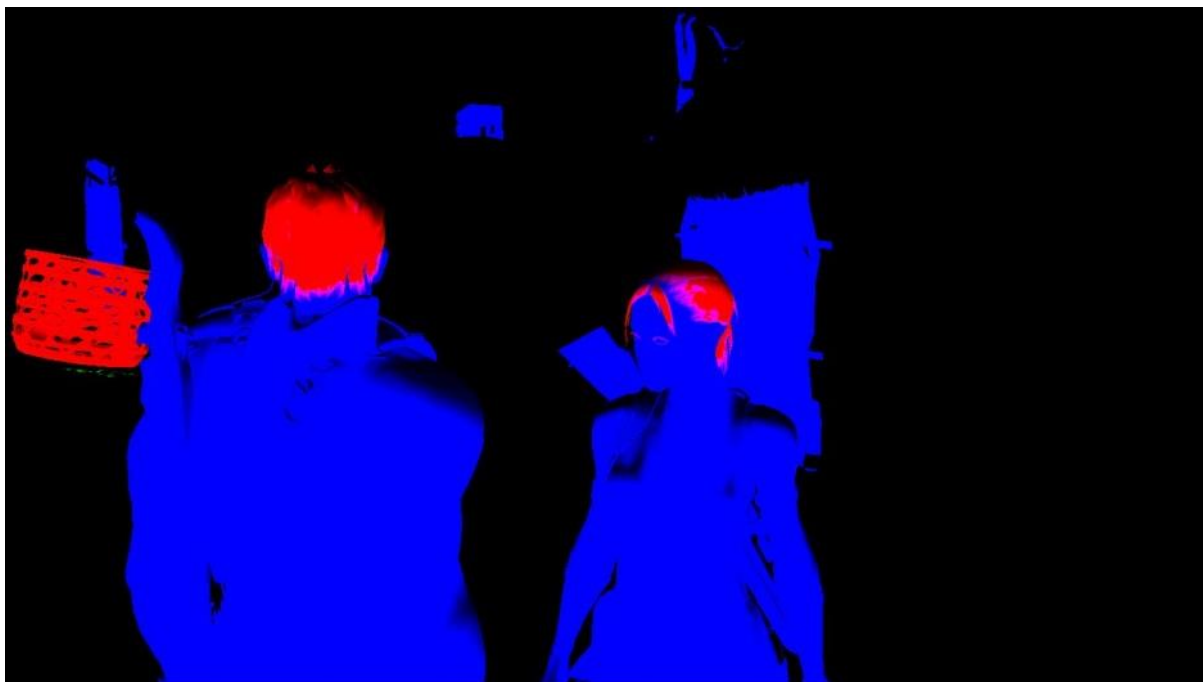
是由直接光和环境光的照明生成的，引擎通过直接光生成动态影子时，在光无法照到的地方，也会成为“影子”。作为案例，最基础的实现就是[漆黑的影子]。稍好一些方法，是对光照结束后的结果把影子部分作暗，因为环境光的照明的结果会一起变暗，所以要可以控制影子的颜色。

在「BH5」中，这个影子的描绘部分正在进化。具体的实现是，引擎生成影子时，对于生成影子的区域，还要输出 Mask (Shadow Mask)，引擎的光照处理通过这个 Shadow Mask 来判断哪里被看作影子(译者注：这里的影子是指直接光照不到地方的阴影)，并只使用环境光进行照明。

石田先生[因为「BH5」是影子特别多的游戏，所以一旦和[失落的星球]使用同样的做法就会完全变的漆黑。而且，这个方法不能作出基于物理的正确照明。Shadow Mask 是分为对应半透明材质 Mask 和不透明材质的 Mask，在进行适合各自的照明。]

[Shadow Mask]





青色的部分是通常的 Shadow Mask，红色部分是半透明用的 Shadow Mask。

根据这个，「BH5」的影子。不会只描绘出一样黑暗的影子，而是在影子中也表现出丰富的阴影。这种过分的真实和自然不知道是否会看不出来，但要非常注意的观察主人公角色的 Self Shadow。投射在身体上的手臂的影子不只是变暗，而是可以看出这个场景里环境光照出的柔和的阴影。比起没有阴影的位置，虽然确实是变暗了，但可以注意到伴随着非常丰富的颜色明暗。

上田先生[关于影子边缘的 Soft Shadow 化，基本上实施了 2X2 或 3X3 的 PCF：Percentage Closer Filtering 的模糊处理。还有使用 Depth Shadow 方法的影子会有阶梯装的锯齿的问题，使用了传统的在 Shadow Map 里加入 Bias 的改良方法。]

Depth Shadow 技法是把光源到遮挡物的距离记录在 Shadow Map 来生成，在渲染时，把像素的深度值（要转换到在计算光源距离的空间）和记录在 Shadow Map 上的值做比较，来判断这个像素是在遮蔽物的里面还是外面。要做[内侧是影子，外侧是被光照]的判断。但是，精度问题会使影子边界附近的判断有误差，描绘的结果也会有失真发生。传统的方法，是在生成 Shadow Map 时稍微加入一些 bias，这样使得在这些危险地方的判断明确化。

石田先生[虽然是普通的方式，但是这样的感觉[shadowmap $z < z_0 ? 1 : 0$]。根据和 Shadow Map 的深度的比较是在外侧还是在里面来生成只有 0、1 二值的信息。在「BH5」中，Shadow Map 的深度和渲染时像素的深度，要有一定程度的距离偏差。这个感觉就是[shadowmap_z $< z_0 : \text{satuate}((z - \text{shadowmap_z}) / \text{weight})$]，用 0~1 之间的实数值来表示。这个公式表示的是 1 个 Sample 的，让这些组合成 4 (2x2) Sample 和 9 (3X3) sample 的 PCF。]

虽然原理和实现不同，但把判断是否是影子用实数来返回，生成顺滑的影子边缘的概念，和「Unreal Engine 3」采用的 VSM：Variance Shadow Maps 非常相似。

石田先生[这个处理因为使用 RSX 的 4 点同时比较的采样机能不能高速化，所以在 PS3 上省去了。]

上田先生[在 PS3 上为了减少影子的阶梯状 Noise，只能使用增加 Bias 的方法来对应，所以细微凹凸的影的描绘有一部分被省略了。作为 Soft Shadow 的处理，XBOX360 不只是 4 点，而是 9 点的 PCF。结果生成影子的实际形态，XBox360 版变得相当的顺滑。]

总的来说，通常技法（PS3 版），把是否是影子的判断信息作成[是否被遮蔽的二值判定]，但 XBox360 版中实现的是把遮蔽用百分比来返回。使用这个方法，影子轮廓附近从影子最暗部分到没变成影子的明亮部分，可以用渐变顺滑的连接起来。

这里要用 PCF 组合出模糊，如果用 2X2 的 4 点 PCF 案例来比喻，PS3 版是把 4 个为 0 或 1 的 2 值影子变模糊，XBOX360 版是把 4 个 0~1 的实数的影子变模糊，模糊结果的光滑度完全不一样。

[影子边缘的 Soft Shadow 化]

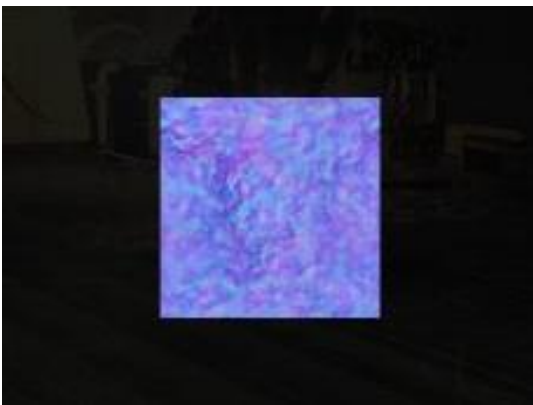




上图没有 PCF。锯齿明显。

下图是有 PCF，出现了降低锯齿的 Soft Shadow 效果。

不是很显眼但很厉害的「BH5」的水面。高级技术和低级技术的最高协作



水波的法线贴图的可视化截图

「BH5」中，在被寄生物污染的湿地地区非洲原住民的村落里，有个主人公们进入鳄鱼栖息的沼泽场景。这个即使在剧本里也和水面有很强交互的场景，水面的表现，比之前任何用 MT Framework 为基础的游戏都更受到重视。「BH5」的水面表现，基本上采用了近代 3D 游戏图形的标准实现方法。

水波，要进行基于纹理的波动模拟，动态的生成法线贴图来表现。自然发生的波浪，是在波动模拟用纹理里写入波纹等“种子”特效的方法来使其发生的、动态角色和水面的交互也是一样。光照上采用了水面反射模型里经典的菲涅尔（Fresnel）反射，对应视线和水面相交的角度，把周围的情景作为镜像显示，并把水底的样子光滑后合成渲染。

还有，镜像反射是动态生成的环境贴图，以主人公角色为首，各种动态对象物体都可以很好的映入。这个动态镜像的表现在[失落的星球]中是被省略的部分。在沼泽场景中，为了制作混浊污水的气氛，在水底方向升起了雾气，虽然很难看到水底的样子，但在一边拿着手电筒一边前进的洞窟场景中，应该可以很清楚的看到水底。

[水面表现]



没有水面的状态，可以看到水底原来的样子。



只有水面多边形的状态



在多边形的水面上应用法线贴图的水波和菲涅尔反射的状态



从上图进一步有效的做出折射的状态



在水面上有效的反映情景（镜像）的状态



表现混浊度很深的水中烧起雾气的状态，完成画面。



汽艇的波浪是由设计工作室预先制作的特效。如果不说能看出来么？

登上敌人乘船的两个主人公被追击的水上追逐场景中 (Boat Chase Scene)，生成了由于船体的运动，将水面分开的漂亮几何体等级的高度很高的波浪，不过这是怎样的东西呢？

上田先生[实际上这些是由程序和美术师共同作出的顶点动画之类的效果(笑)。总之，在船的周围让其作为特效来发生。这个最终的形态，就是水上 Boss[Irving 决战]。这个场景有着暴风雨和大波浪的印象，实际是把 100 个以上的骨骼放入水面，把这些进行反复的活动就成了 Bone Skin 的波浪了(笑)。]

「BH5」中几何体等级的波浪全都不是波动模拟作出的，而是手工制作提前生成的顶点动画。虽然怎么看也只看出[暴风雨的大浪]、实际是美术师预先做好的波浪，不论是电影还是游戏，看上去和真的很像就是正确的，那么就有理由“胜出”了，这个原始的手法很有趣。再次玩游戏的用户如果有意识的去仔细观察水面，也许就会体会这个意图而偷笑起来，真是个有趣的秘密。

[折射贴图]

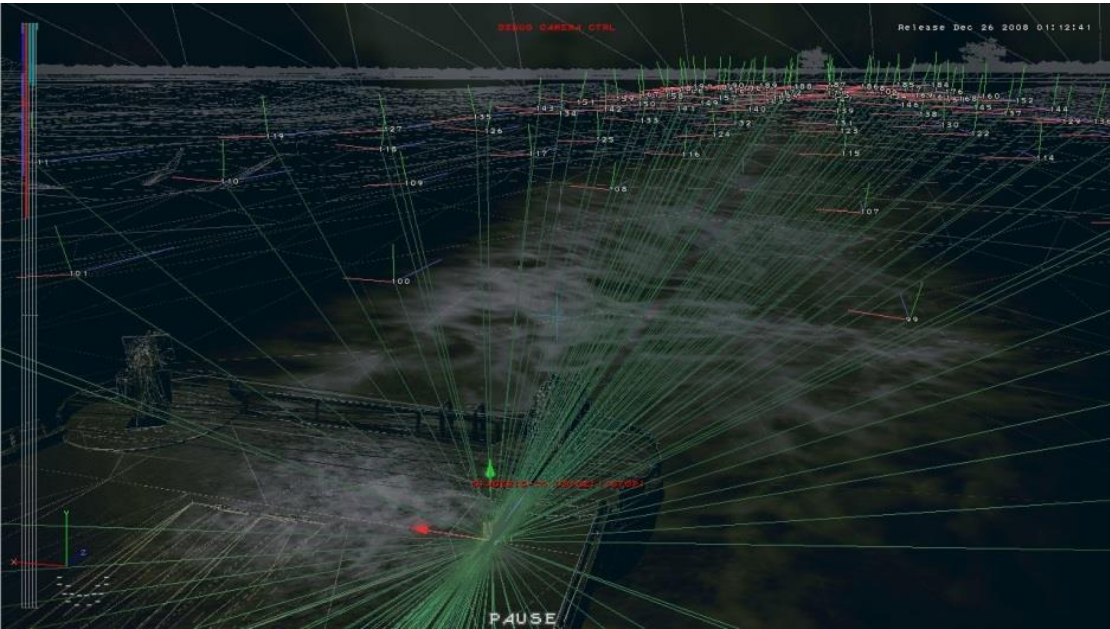


上图是关闭折射的状态。

下图是开启折射的状态。

因为把视线用很好的弯曲方式处理折射，所以注意下水面下的角色们的脚看起来短了。

[Irving 决战]





Irving 决战的水面线框显示。在下图里隐藏了冲击的实时，实际上波浪使用骨骼来控制的。

让人在意的 PS3 版和 XBox360 版的区别是什么？



卡普空技术研究室室长伊集院胜先生

「BH5」的 PS3 版和 Xbox360 版虽然是以[相同的视觉效果]作为基本理念开发的，不过就像对于阴影生成使用了不同的方式那样，因为硬件上的不同而存在有细节部分的不同。

首先，Xbox360 版采用了 30FPS 的双缓冲（double buffer）。固定上限 30FPS，除了在高负荷场景不能达到之外，基本上都维持在 30FPS。PS3 版本是可变帧率（Frame Rate），因为使用了三重缓冲（Triple Buffer），有着即使在帧率降下来时粗看也不显眼的点。

双缓冲由显示用的前置帧缓冲和渲染用的后置帧缓冲两个构成的渲染方法。三重缓冲是把后置缓冲再增加一个的方法。双重缓冲是把通过两个缓冲的前后关系，使得显示器的刷新率可以同步的显示模式，在等待同步的时间里无法进行下一帧的渲染。如果无视显示同步的来显示，就会在显示途中被新的一帧替换，显示器显示出来的就是[上面是前一帧][下面是后一帧]这样的状态。这就是所谓的[Tearing]现象。

三重缓冲，即使让显示器的刷新率同步后显示，但后置缓冲因为空出一个来，系统不需要等待同步，就可以着手下一帧的渲染了。PS3 版的平均刷新率不及 Xbox 360 版，但在刷新率降低时也很难看到缺点。还有，Xbox360 因为有 10M 的 EDRAM 的硬件配置的制约，不能使用三重缓冲。

抗锯齿（anti-aliasing）处理，在 PS3 版采用的是关闭～2×的可变 MSAA（Multi Sampled Anti-Aliasing），Xbox360 版采用了 2X～4X 的可变 MSAA。关于 PS3 的 CELL 处理器，以 SCE 提供的对应 SPE（Synergistic Processor Element）的程序库的主题为例。根据这个，每个 SPE，能以比线程更小的单位（Fiber）自发的进行上下文交换（context switch），就像矩阵的计算等级一样，把更小粒度的大量工作，在 SPE 上分配，也能得到期待的性能。布料的模拟等，都可以从这个新的程序库里得到好的效果。还有，用 SPE 作几何体加速的[Playstation Edge]，在「BH5」上没有被利用，那是为什么呢？

伊集院先生[虽然加入应该会稍微提高响应速度。不过 PS Edge 本身会消耗内存，必须把顶点数据用独立的格式放在主内存里，所以这次因为容量问题没有加入。作为替代，在「BH5」中利用了被称为[SPU Patch]的 RSX 的加速程序库。因为可以利用 SPE 上用 RSX 进行 Shader 的组织，所以变的相当的快了。]

在世嘉的[索尼克世界大冒险]那篇文章中曾经提到过，想要让一个游戏在多个平台上展开，在本时代中，会有各种各样的问题。游戏盘的媒体等级的容量就不一样，PS3 是双层的 50GB 的蓝光，相对的 XBox360 采用的是传统的双层 8.5GB 的 DVD-ROM。

关于把[生化危机 5]在 PS3 和 XBox360 上展开，开发时还是以装在 XBox360 的 1 张 DVD-ROM 中为最优先的设计吧。顺便说一下，虽然作为 DVD-ROM 最大是 8.5GB，但在作为 XBox360 的游戏媒体时，因为系统数据等的关系，游戏资料的数据能收录的大约有 6.8GB 左右。

石田智史先生「BH5」的容量是 5.8GB 吧。在通常的游戏项目中，把全部的数据放在一个大的文件里来实现，为了在预估的读盘时不会发生没有准备好的寻道操作，放入了重复的数据。但在「BH5」当中，5.8GB 是没有重复数据的（笑）。有 1G 是为了在读取模式中减少寻道操作分配到了高速化文件里。全部数据平均的被可逆的压缩到了一半，所以实际容量是 12G 左右。]

伊集院先生[虽然很有意识的放在了 DVD-ROM 中，但在 DVD-ROM 的容量限制内，并没有放弃什么，虽然也可以采用多张 DVD-ROM 的方案。不过，在开发接近完成时，用一张盘也可以了吧。那么，PS3 版那边就相当宽裕了，因为是世界首次的游戏和 Movie 的混合 BD，在制作水平上也很辛苦。]

总结——MT Framework 的进化没有停止！



「BH5」中支持和植物的交互

卡普空夸耀的 MT 架构的游戏，有[丧尸围城]、[迷失行星]、[鬼泣 4]和[生化危机 5]这 4 个作品，设计师的阵容，也因为经过了这 4 个作品的开发，变的能够熟练自如的使用 MT 架构了，在最近的作品中，能够远远的表现出凌驾于当初预想的品质，也出现了很多让技术阵营方面吃惊的局面。

以这次的「BH5」来说，皮肤的质感和水面的表现等，这些可以说是真正的熟练的使用 MT Framework

的技术。

MT Framework 在所述的 4 个游戏里依次的升级版本，技术阵营一方引擎也反复改良，在提升品质上的关系很大。如果用「BH5」来说，脸部表现等的法线再计算部分等也是，如果没有这个改良，「BH5」中就不会实现现在这样真实的脸部表现吧。

所以对于[以游戏引擎为基础的游戏开发，就要完全停止表现的进化]的畏惧，虽然被多次说到，但关于 Framework，设计人员和引擎技术人员的配合很强，而且自制引擎有着可以和公司内部灵活进行机能强化的[地利]，这样做完全没有负面影响。

还有，以通用引擎为基础的游戏，虽然说可以得到[开发成本减少]和[开发周期缩短]的直接好处，但实际并不是这样。优秀的通用游戏引擎，用同样的预算开发费用，想全新的领域进行研究，因为尝试和新技术的融合，会产生预算和时间的富余。「BH5」中和好莱坞合作的实现，也可以说是 MT Framework 优秀的间接效果吧。

作为日本游戏开发动向的试金石般的存在，CAPCOM 传家宝刀[MT Framework]的最新动向，和以它为基础的游戏，在今后也必定受到注目。所以，关于下一次作为[生化危机 5]特别篇的次世代 MT Framework 的秘密也会尽快报道。

[植物互交作用]



和植物的互交作用的演示
