

运用层次纹理映射的基于图像绘制算法*

郑新¹, 王文成¹, 吴恩华^{1,2}

¹(中国科学院 软件研究所 计算机科学开放研究实验室, 北京 100080);

²(澳门大学 科技学院, 澳门)

E-mail: zhx-zhengx@263.net; {whn, weh}@ios.ac.cn

http://www.ios.ac.cn

摘要: 提出一种基于图像绘制(image-based rendering, 简称 IBR)的算法, 以发挥图形卡的纹理映射功能, 并能表现物体表面的三维凹凸细节. 首先将深度图像的像素按其相关深度分为多层纹理图像, 然后利用纹理映射的硬件支持, 将这些层次纹理图像依次投影到与视点相关的成像面上, 以得到所需目标图像. 为了避免目标图像上出现空洞, 在生成时, 将纹素在深度层次上进行扩展. 由于层次纹理图像需要较大的存储空间, 并且在装入纹理缓冲时要花费大量时间, 为此还提出一种基于压缩层次纹理图像的目标图像生成算法. 实验表明新算法是有效的, 尤其适于处理与整个物体大小相比深度层次不太多的三维景物, 如有表面凹凸纹理(浮雕、门窗等)的建筑物表面等.

关键词: 基于图像的绘制; 层次纹理; 纹理映射; 绘制; 透明度混合

中图法分类号: TP391 文献标识码: A

在基于几何的真实感图像生成技术中, 传统的纹理映射难以表现物体表面凹凸细节. 解决这一问题的方法通常有 3 种: 凹凸纹理映射(bump mapping)、位移映射(displacement mapping)及体纹理(volumetric texture). 凹凸纹理映射^[1]通过扰动表面法向来产生真实感图像, 但它的前提条件是物体表面起伏很小, 不会改变纹理映射后的轮廓线. 但这样无法处理凹凸表面在视点改变后所引起的自遮挡现象. 位移映射^[2]为每一个纹理像素赋予一个位移值, 并将每一个像素看做是一个多边形网格, 以便用几何图形学的方法来绘制, 因此计算量大. 三维体纹理^[3,4]是表现三维凹凸细节的另外一种方法, 且有硬件支持, 但运用它时需要巨大的存储量和计算量, 且价格昂贵, 目前在普通微机上还无法普及.

近年来, 基于图像的绘制(image-based rendering, 简称 IBR)技术已成为研究热点之一. 与传统的真实感图像生成技术相比, 其优点是无须构造复杂的场景造型及进行复杂的光照运算就可生成具有高度真实感的新视点目标图像, 而且其算法复杂度几乎与场景复杂度无关. 因而, 它在虚拟现实系统中被广泛采用^[5~8].

在 ACM SIGGRAPH'2000 上, Oliveira 等人将三维变换的图像绘制技术与传统的二维纹理映射相结合, 提出了一种称为“浮雕纹理映射”(relief texture mapping)^[8]的新视点图像生成算法, 实现了用传统纹理映射方法难以实现的三维凹凸细节效果. 该算法首先将三维物体正投影到它的包围盒表面上, 以得到带有深度的浮雕纹理(relief texture, 简称 RT); 然后通过一个可分离为两次一

* 收稿日期: 2000-12-05; 修改日期: 2001-05-14

基金项目: 国家自然科学基金资助项目(69873044; 69903009); Research Grant of University of Macau 基金资助项目(RG029/00-01S/WEH/FST)

作者简介: 郑新(1973-), 女, 山东苍山人, 博士, 主要研究领域为计算机图形学, 虚拟现实; 王文成(1967-), 男, 湖南双峰人, 博士, 副研究员, 主要研究领域为计算机图形学, 可视化, 虚拟现实; 吴恩华(1947-), 男, 江苏南通人, 博士, 研究员, 博士生导师, 主要研究领域为计算机图形学, 可视化, 虚拟现实.

维变换的二维“预变换”(pre-warp)过程,将 RT 图像变换为一个分辨率大小固定且与新视点相关的中间纹理图像;最后用传统的纹理映射技术将此中间纹理图像变换成新视点的图像。

与上述浮雕纹理映射方法不同,本文提出另外一种实现物体表面三维凹凸细节效果的 IBR 算法,且在成像时只需调用纹理映射的功能.该算法首先将纹理图像像素(简称纹素)按照其深度分别存放在不同的“层次纹理图像”(layered texture,简称 LT)之中,然后用纹理映射的方法将各 LT 图像由后往前映射到新视点相关的成像面上,生成目标图像.为了避免目标图像上出现空洞,该算法在生成 LT 图像时,将纹素在多个深度层次上进行扩展.同时,将包围盒中 6 个表面相关的 LT 图像合成一个关于三维物体的数据场表达.为了减少 LT 图像对存储空间的要求和装入纹理缓冲时所花费的大量时间,本文还提出了一种基于压缩层次纹理图像的新视点图像生成算法.实验表明,该算法是有效的,尤其适合于处理那些与整个物体相比深度层次不太大的三维景物,如有表面凹凸纹理(浮雕、门窗等)的建筑物表面等。

1 层次纹理映射

传统的纹理映射是将一幅图像作为纹理投影到一个平面上,难以表现物体表面的三维凹凸细节.本文提出的层次纹理映射方法是将带有深度的纹理图像按照深度分层存放,然后将这些纹理图像由后往前依次映射到成像面上,就能生成新视点的目标图像.此时,可利用纹理映射硬件来加速成像.下面,我们首先给出层次纹理图像的概念,然后详细介绍 LT 图像的生成以及基于它的绘制方法。

1.1 层次纹理图像

本文的参考图像为平行投影后得到的带有深度信息的纹理图像(本文沿用浮雕纹理映射中的名称,称其为浮雕纹理图像,简称 RT 图像).将此纹理图像上的像素按其深度分别存放到多层纹理图像上,就得到一个纹理图像序列,称为层次纹理(layered texture,简称 LT).为避免混淆,下面将 LT 图像上的像素称为纹素(texture pixel,简称 Tpixel).此时,每一个 LT 图像上的纹素都具有相同的深度,在某种意义上它与普通的纹理图像是相同的.本文的算法用 OpenGL 来实现 LT 图像到成像面的映射,这样,物体表面间的相互遮挡、投影变换、纹理映射等都可以方便地借助图形加速硬件来实现。

1.2 层次纹理图像的生成

在 IBR 技术中,为保证绘制结果的正确性,通常要进行两项工作:一项是空洞填补,另一项是保持正确的遮挡关系.鉴于产生空洞的原因主要是信息缺失,本文的算法在建立 LT 图像时从两个方面完备原始数据信息:首先将纹素在深度层次上进行扩展;然后将物体包围盒各表面所对应的各个 LT 图像综合成关于场景的一个三维数据场表示.至于遮挡问题,可以借助于 OpenGL 提供的深度检测功能来处理。

1.2.1 纹素在深度层次上的扩展

如果简单地将 RT 图像上的各个像素放到不同深度的 LT 图像上,那么,在生成新视点目标图像时就可能出现空洞.因为 RT 图像上两个深度差比较大的相邻纹素(如图 1 中的 A 和 B),当视点改变时,它们在屏幕上的投影(A'与 B')可能会分离,从而产生空洞。

为了填补这种空洞,本文的算法在生成 LT 图像时对纹素在深度层次上进行扩展.如图 1 所示的一维情况,为了填补 A'与 B'之间所出现的空洞像素,本文的算法将深度层次比较小的纹素 B 在

其RT视线方向上进行扩展,即从 LT_{B+1} 到 LT_A 各个层次的LT图像上,在与纹素 B 相同的位置都填上一个插值纹素,这些插值纹素的颜色值为纹素 A 与 B 颜色的线性插值.在真正的二维RT图像中,一个像素的扩展结果与它周围的8个相邻像素相关.

经过上述纹素扩展之后,可以保证在一定的视域内,目标图像上不会出现因像素分离而产生的空洞.在如图1所示的情况中,视线 VC' 与其可见纹素 C 所在深度平面夹角为 α .由图1可知,当夹角 α 满足以下条件时,

$$\alpha > \arctg(d/p), \tag{1}$$

视线 VC' 总会与纹素 C 相交,其中 d 为LT图像平面的深度间距, p 为纹素宽.因此,当所有视线与其可见纹素所在深度平面夹角满足条件(1)时,目标图像上就不会出现由于像素分离而产生的空洞.

当然,如果RT图像上两相邻像素是非深度连续的,那么在其对应纹素间做插值计算会引起“过多填补(skin)”的问题.为了避免此种情况的发生,本文的算法对RT图像上非深度连续的相邻像素做标记,以避免在这些地方进行纹素扩展的工作.本文的算法通过设置一个阈值 $D_{disconnect}$ 来区分相邻像素是深度连续的还是非深度连续的.虽然这样处理有可能使那些深度差大于 $D_{disconnect}$ 的深度连续像素间产生空洞,但在三维物体表面比较平滑的前提下,这些空洞可以由下面的图像合成过程来填补.

1.2.2 运用多幅图像的合成

由于单幅图像对三维物体的表达能力是有限的,当视点的改变使原本不可见的物体变为可见时,也会产生空洞.对于这种由于信息缺失而产生的空洞,本文将包围盒中6个表面各自生成的6个LT图像综合形成关于三维物体的三维数据场表示,以弥补绘制中的信息缺失.如图2所示,空间三维物体位于其最小立方体包围盒中,每个包围盒表面对应着一组深度扩展后的LT图像.在上述LT图像生成过程中,如果按照图像像素所表示的空间距离来进行深度层次划分,也就意味着将立方体包围盒均匀划分为 $n \times n \times n$ 的小体元(texel)(其中 n 为上述纹理图像的分辨率,在本文算法的实验中, n 取值为256).这样,6组包围盒表面所对应的LT图像都可以统一到同一个坐标系中,从而生成关于整个三维物体表面的一个三维数据场表示.此数据场包含了物体包围盒中6个表面的所有信息,可以根据新视点成像的需要,形成垂直于 X, Y, Z 轴方向的3组LT图像.

1.3 绘制

如前所述,将层次纹理图像依次投影到成像面上,就能生成新视点目标图像.因为每一组LT图像包含了三维物体表面的所有信息,所以在任意一个视点只需绘制其中一组纹理图像即可.从包围盒的中心点向包围盒各顶点连线,这些线与包围盒的边所确定的平面将三维空间划分为6个锥形区域,每个区域对应着包围盒的一个表面.当视点位于其中的一个区域时,只需绘制平行于此区

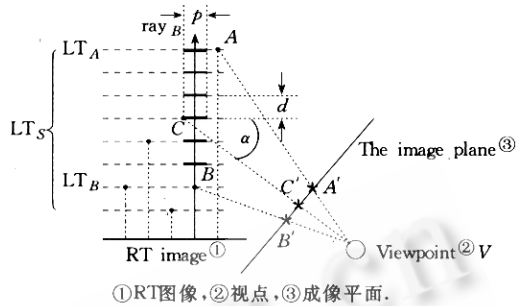


Fig.1 Tpixel extension along depth direction
图1 纹素在深度层次上的扩展

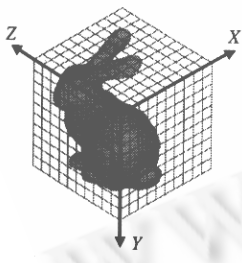
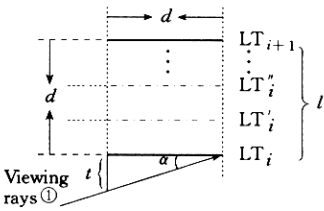


Fig.2
图2

域、对应于包围盒表面的那组 LT 图像即可。

此时,各 LT 图像平面间的深度间距 d 与纹素宽 p 等长,因此按照第 1.2.1 节中的方法对相邻



①视线.
Fig. 3
图3

像素做插值以后,满足目标图像上无空洞出现的条件便简化为新视点 V 到所有可见纹素的视线与纹素所在深度平面的夹角 α 大于 45° . 在视区划分的每一个锥形区域内,虽然大多数视点都满足上述夹角大于 45° 的条件,但当视点与区域间的分界面非常接近时,仍然会出现某些视线与可见纹素所在平面夹角小于 45° 的情况. 为了避免这种情况下在目标图像上出现空洞,本文应用一种增加采样的近似方法. 如图 3 所示, α 为当前视点与层次纹理图像 LT_i 所在深度平面的最小夹角,当 α 小于 45° 时,为了避免目标图像出现

空洞,将当前纹理平面 LT_i 在其深度层次和 LT_{i+1} 所对应的深度层次间重复绘制 l 次,每次绘制时都将 LT_i 在深度上平移一定的距离 d/l ,其相关计算如下:

$$t = d \cdot \text{tg}(\alpha),$$

$$l = \lceil d/t \rceil = \lceil \text{ctg}(\alpha) \rceil. \tag{2}$$

其中 d 为相邻 LT 图像的深度间距, $\lceil \cdot \rceil$ 为向上取整操作. 例如,图 3 中 LT'_i, LT''_i 等即为图像 LT_i 在三维空间中的重复. 由于 α 为当前视点与图像 LT_i 所在深度平面的最小夹角,所以最终绘制的层次深度间距必定小于 t . 此时,图像 LT_i 及其复制图像上所有可见纹素都会满足公式(1)中的条件,在目标图像上就不会出现空洞. 图 4 所示图像为应用此方法绘制的结果. 在参考图像和目标图像的分辨率均为 256×256 的情况下,其时间分布见表 1.

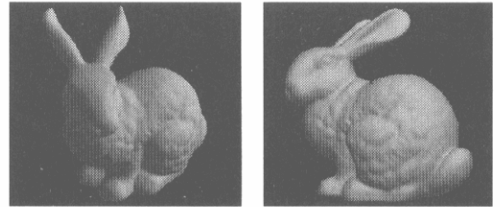


Fig. 4 Novel view generated from layered texture image

图 4 由层次纹理图像(LT)生成的目标图像

Table 1
表 1

	Install textures into the texture buffer ^①	Render space quadrangles ^②	Others ^③
Time ^④ (s)	1.031	0.228	0.015
Percentage ^⑤ (%)	80.9	17.9	1.2

①将纹理装入纹理缓冲区,②绘制空间四边形,③其他,④时间,⑤所占比例.

2 压 缩

如果纹理影射缓冲区足够大,可以同时装入所有的 LT 图像,那么上述算法的速度将足够快. 但是目前在普通微机上的图形加速硬件一般没有这么大的纹理缓冲区,因此只好每次装入 1 层或几层图像. 将纹理图像装入纹理缓冲区占用了绝大多数的绘制时间,如表 1 所示实例,它占用了全部绘制时间的 80.9%,这严重影响了绘制速度. 另外,分 n 层存放的 LT 图像也占用了大量存储空间. 因此,有必要对 LT 图像进行压缩处理.

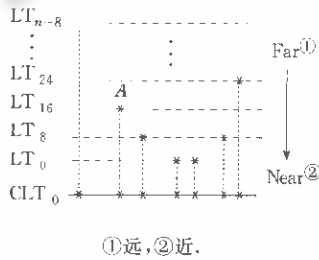
2.1 层次纹理图像的压缩

一个空间连续的三维物体,在其包围盒表面对应的 RT 图像上,绝大多数相邻像素的深度差都不大(在本文的实验中,99.3%的相邻像素深度差小于或等于 4). 因此,按照第 1.2.1 节中的方法对纹素在深度层次上进行扩展时,每一个纹素所扩展的长度都不大,而 6 个面合成的关于三维物体

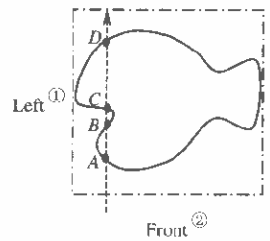
表面信息的一个数据场表示,其内部绝大多数体元是空的,这就为 LT 图像的压缩提供了空间。

在此,本文为包围盒中 6 个表面分别存储一组 LT 图像,并将每组压缩成 8 层存储,以便在绘制目标图像时可以将一组层次纹理图像一次装入纹理缓冲区,节省装入纹理所需的时间.此压缩过程的具体实现如下:按照由后往前的顺序,将每层纹理图像 LT_i 上非透明度为 1 的纹素(即非空纹素)存储到压缩层次纹理图像(compressed layered texture,简称 CLT)的第 s 层 CLT_s 上,其中 $s = i \% 8$,即 s 为 i 整除 8 后得到的余数;同时,将该纹素的深度层次 i 作为透明度分量存储到 CLT 图像中(此数值在绘制阶段用于在 CLT 图像中选择当前深度层次纹素).我们能够将 LT 图像压缩到 8 层存储的原因是,通过将非深度连续阈值 $D_{disconnect}$ 设为 8,本文的算法将纹素在深度层次上的扩展长度限制为 7,即在未合成到统一坐标系之前,每条垂直于 LT 平面的视线方向上最多有 8 个纹素,所以在压缩时不会发生重叠现象.如图 5 所示,为生成包围盒前表面所对应的一组 CLT 图像的二维情况。

在上述 LT 图像压缩过程中,每组 CLT 图像都保存了当前表面可见纹素的全部信息,但这会导致部分来自其他表面的像素信息的丢失.如图 6 所示,物体表面上的 4 个点 A, B, C, D 位于同一视线方向上,在相对于“Front”表面所建立的 LT 图像中,这些点位于不同深度图像的不同位置,每个纹素附近都有多个插值像素.在按照由后往前的顺序将纹素存储到对应的 CLT 图像的过程中,后面的点(如 D)有可能被前面的点(如 A 或其扩展纹素中的某个与 D 对应于同一压缩层次的点)所覆盖,但这些丢失的像素信息仍然可以由其他包围盒表面所对应的 CLT 图像得到,例如, B, C, D 这 3 点及其附近的插值纹素仍然保存在“Left”面所对应的 CLT 图像中,因此,6 个面的压缩纹理图像并没有造成信息的缺失。



①远,②近.
Fig. 5
图5



①左,②右.
Fig. 6
图6

可以由其他包围盒表面所对应的 CLT 图像得到,例如, B, C, D 这 3 点及其附近的插值纹素仍然保存在“Left”面所对应的 CLT 图像中,因此,6 个面的压缩纹理图像并没有造成信息的缺失。

2.2 绘制

在生成新视点图像时,单个表面所对应的一组 CLT 图像可能难以确保生成正确目标图像所需的全部信息,因而需要由多组 CLT 图像来合成.为此,对三维视区间重新进行划分,整个三维空间被 3 个分别平行于包围盒表面、互相垂直并经过包围盒中心的平面分割成 8 个区域.在每一个区域中,需要 3 组 CLT 图像来合成目标图像。

由实验可知,纹理缓存大于或等于 16M 的图形加速卡或显卡一次可以装入 8 幅分辨率为 256 的纹理图像.因此,对于每一组需要合成的 CLT 图像,只需一次纹理装入操作即可.假设这组 CLT 图像中所存储纹素的最大、最小深度层次分别为 d_{max} 和 d_{min} ,按照由后往前的顺序,从 CLT 图像中选择对应于当前深度层次的纹素,并且将这些纹素投影到与其包围盒表面平行的成像面上.这种挑选层次的操作可由 OpenGL 中的透明度(α)检测来实现。

此时,视线与平面所成的夹角范围为 $(0^\circ \sim 90^\circ)$.当视线与平面的最小夹角 α 小于 45° 时,仍然用第 1.3 节中所提出的近似增加取样的方法来进行绘制.当 α 趋近于 0° 时,纹理平面所需重复的次数接近无穷,但在实际操作中,并不需要重复那么多次.本文的算法为重复绘制的面片数设置了一个最大值 L_{max} .在大多数情况下, $L_{max} = 10$ 就足够了。

3 实验结果

本文的算法的效率与图形硬件的纹理加速功能、最终目标图像的分辨率以及 LT 图像的深度层次相关. 为此, 以兔子和著名的澳门大三巴教堂遗址模型为例, 在参考图像的分辨率为 256×256 的情况下, 针对不同的目标图像分辨率, 我们在两种不同性能的机器上做了多组实验. 这两种机器及图形加速卡的型号和性能参数分别为: Dell Precision420 工作站, Pentium III 667 处理器, 带有 32M 缓存的 Matrox Millennium G400 Dualhead Max 显示卡; Dell Dimension4100 台式机, Pentium III 667 处理器, 带有 32M 缓存的 TNT2 显示卡. 实验结果数据见表 2.

Table 2
表 2

Computer ^①		Dell Precision420			Dell Dimension4100		
Resolution of the rendered image ^②		256×256	400×400	512×512	256×256	400×400	512×512
Speed (frame/s) ^③	Hare ^④ (LT)	1.16	1.12	1.09	0.80	0.78	0.73
	Hare (CLT)	2.82	1.80	1.42	3.88	1.74	1.10
	The facade of St. Paul Church ^⑤ (CLT)	28.6	25.0	21.7	71.4	28.6	17.5

①机器, ②目标图像分辨率, ③速率(帧/秒), ④兔子, ⑤大三巴.

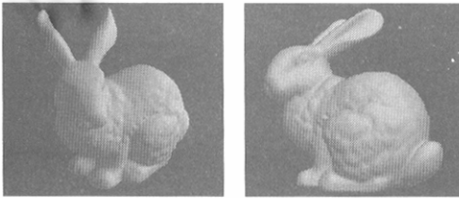


Fig. 7 Novel view generated from compressed layered texture image

图 7 由压缩层次纹理图像 (CLT) 生成的目标图像

由数据可知, 本文的算法尤其适合处理那些与整个物体大小相比深度层次不太大的三维景物, 如有表面凹凸纹理 (浮雕、门窗等) 的建筑物表面等, 对此完全可以达到实时. 图 4 和图 7 中两组图像分别为由 LT 图像和 CLT 图像生成的不同视点下的目标图像. 由图可以看出, 它们之间在视觉上并无明显差别. 与图 8 中无深度普通纹理映射得到的目标图像相比, 应用本文的算法由 CLT 图像生成的目标图像 (如图 9 所示) 的立体感和真实感更强 (由于视点改变, 凹洞中神像的一部分会被立柱遮挡, 可以注意到, 在图 9 中这种遮挡关系比较明显, 而图 8 中却没有).

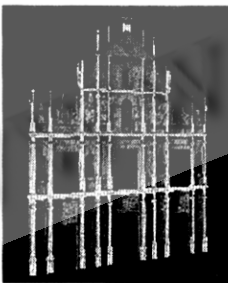


Fig. 8 Novel view generated by mapping a texture image without depth

图 8 无深度普通纹理映射得到的目标图像

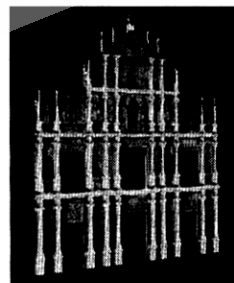


Fig. 9 Novel view generated from compressed layered texture image

图 9 由压缩层次纹理图像 (CLT) 生成的目标图像

4 结 论

本文提出了一种基于图像绘制的算法, 以充分利用图形硬件来加速成像操作, 并能表现物体表

面的三维凹凸细节效果. 该算法将深度图像按深度分层存放, 然后用通常的纹理映射方法将各 LT 图像由后往前依次映射到成像面上, 以生成目标图像. 物体表面间的相互遮挡、投影变换及纹理映射等都可以方便地借助图形加速硬件来实现. 实验表明该算法是有效的, 尤其适于处理那些与整个物体相比深度层次不太大的三维景物, 例如, 有表面凹凸纹理(浮雕、门窗等)的建筑物表面等, 此时可以达到实时效果.

对于某些三维物体, 由于其 6 个包围盒表面投影图像难以获得物体表面足够的信息, 因此, 从某些视点得到的目标图像上仍然会有空洞存在. 对于这种情况, 本文的算法和浮雕纹理映射方法目前都难以处理, 但借助于 LDI^[6]的思想可以有效地加以解决.

致谢 本文实验中所使用的场景是由费广正博士提供的, 我们衷心地表示感谢.

References:

- [1] Cohen, M. F., Wallace, J. R. Radiosity and Realistic Image Synthesis. Academic Press, 1993.
- [2] Logie, J. R., Patterson, J. W. Inverse displacement mapping in the general case. Computer Graphics Forum, 1994, 14(5): 261~273.
- [3] Kajya, J. T., Kay, T. L. Rendering fur with three dimensional textures. Computer Graphics, 1989, 23(3): 271~280.
- [4] Meyer, A., Neyret, F. Interactive Volumetric Textures. In: Drettakis, G., Max, N., eds. Proceedings of the 1998 Eurographics Rendering Workshop. Springer-Verlag, 1998. 157~168.
- [5] Mark, W. R., McMillan, L., Bishop, G. Post-Rendering 3D warping. In: Proceedings of the 1997 Symposium on Interactive 3D Graphics. New York: ACM Press, RI, 1997. 7~16.
- [6] Shade, J., Gotler, S., He, L., et al. Layered depth images. In: Cohen, M., ed. Proceedings of the SIGGRAPH'98. New York: ACM Press, 1998. 231~241.
- [7] Liu, Xue-hui, Sun, Han-jiu, Wu, En-hua. A hybrid method for image synthesis in IBR for novel viewpoints. In: Kwang, Yun Wahn, ed. Proceedings of the ACM Symposium on Virtual Reality Software and Technology. New York: ACM Press, 2000. 55~63.
- [8] Oliveira, M. M., Bishop, G., McAllister, D. Relief texture mapping. In: Akeley, K., ed. Proceedings of the SIGGRAPH 2000. ACM Press, 2000. 359~368.

Image-Based Rendering Through Layered Texture Mapping*

ZHENG Xin¹, WANG Wen-cheng¹, WU En-hua^{1,2}

¹(Laboratory of Computer Science, Institute of Software, The Chinese Academy of Sciences, Beijing 100080, China);

²(Faculty of Science and Technology, University of Macau, Macau, China)

E-mail: zhx_zhengx@263.net; (whn, weh)@ios.ac.cn

http://www.ios.ac.cn

Abstract: An efficient IBR (image-based rendering) method is presented in this paper to take full advantage of the graphics card, which can successfully simulate the 3D bump details on surfaces of object. In this new approach, an original texture with a displacement at each pixel is firstly decomposed into a series of new textures with each one having a same displacement value, called layered textures, or LT. Then, these LTs are used to render the novel view by conventional texture mapping. To avoid gaps in the rendered image, some pixels are to be interpolated and extended in the LTs according to the depth difference between neighbor pixels in the original texture. As these LTs ask for much storage and need a lot of time to be installed into the texture buffer, a method is given to compress these LTs and the corresponding rendering method. Experimental results show the new method is efficient, especially in rendering those objects with a smaller depth range compared with their size, such as relief surfaces of buildings.

Key words: image-based rendering; layered texture; texture mapping; rendering; transparency blending

* Received December 5, 2000; accepted May 14, 2001

Supported by the National Natural Science Foundation of China under Grant Nos. 69873044, 69903009; the Research Grant of University of Macau of China under Grant No. RG029/00-01S/WEH/FST