非线性视域插值*

鲍虎军 陈 莉 王章野 彭群生

(浙江大学 CAD&CG 国家重点实验室 杭州 310027)

摘要 提出了一种新的基于图象的绘制技术.与线性视域插值方法不同,此方法能准确地模拟出漫游过程中的透视变换效果.为了加速视域插值算法,采用二叉剖分方法来优化源图象的分解,从而大大减少了分解后块图象的数目.对生成的中间画面上的黑洞,此方法分两个步骤来填充.算法首先根据深度信息来决定是否需要扩展与黑洞相邻的块图象的边界,然后采用多向视域插值方法插值这些扩展后的块图象来生成中间画面.理论和实验结果表明,此方法比传统方法在准确性和效率上均有显著提高.

关键词 视域插值,环境映照,透视变换,二叉剖分,虚拟现实.

中图法分类号 TP241

漫游是虚拟现实系统所必须具备的基本功能之一,当观察者在虚拟环境中改变视点和视线方向时,他的头盔式显示器上所显示的图象必须能随之实时地更新.传统的真实感图形生成方法均基于场景的几何模型,尽管近10年来图形硬件与绘制技术均有了极大提高,但是仍难以满足复杂场景的实时显示需要.

为了解决这个问题,一种基于图象的绘制技术应运而生.这种技术基本上放弃了场景的几何模型,通过变换、插值、变形附近视点或视线方向上预先生成好的场景画面来快速得到当前视点处的场景画面.最早的基于图象的绘制技术可追溯到环境映照方法[1~3],一个点处的环境映照可通过取该点为视点,将周围场景投影到一个中间面上来得到,中间面可取球面、立方体、圆柱面等.这样,当我们在该点处沿任何视线方向观察场景时,环境映照都可提供场景的完全、准确的视图.[4]基于这种策略,Eric Chen设计了一个虚拟现实系统QuickTime VR[5],通过在场景的离散采样点处预先建立环境映照,从而可使用户在虚拟环境中实时地从一个采样点漫游到另一个采样点处.这个系统的主要缺陷在于漫游过程中视点只能位于固定的网格点上,而不能提供场景的连续视图.为此,Eric Chen提出了基于视域插值的方法[6],它在建立环境映照的同时记录各象素处可见点的深度值,从而可恢复每个可见点的三维信息,将这些可见点投影到中间视点的视平面上就可得到中间画面.虽然重投影的过程在理论上很合理,但它的实现却很复杂,为了加速这个过程,Chen和Williams提出了线性视域插值方法来得到中间画面.

在Eric Chen等的线性视域插值方法中,事实上隐含着这样一个假设,即每个可见点在中间画面上的投影位置为一个线性函数,这显然忽略了一个可见点相对于不同的视点位置其深度值的变化,因而在中间画面中可能会产生显著的误差.而且,源画面中有可能有多个可见点投影到中间画面的同一个象素,同时中间画面上的一些象素还有可能得不到任何可见点的投影.对于前者,可按这些可见点相对于新的环境映照中心的距离进行排序,保留最近的可见点;而对于后者,空的象素则在中间图象上形成黑洞,Eric Chen等通过插值相邻象素的颜色或偏移向量来填补这些黑洞.为了提高算法的效率,还可将偏移向量均小于某一给定阈值的相邻象素组成一块图象,插值过程可基于块图象完成.Chen和Williams基于四叉树分割方法对源图象分块,所产生的块图象的数目是非常大的.

本文提出了一种非线性视域插值方法,可精确地计算出每个可见点在中间画面上的投影位置,还提出了一种对源图象进行优化二叉剖分的方法,使得分解的块图象的数目仅为Chen和Williams的方法的一半.同时,我们将中间画面上所产生的黑洞分成两类[7],一类是由于源图象的局部扩张所产生的,可通过扩展与黑洞相邻的块图象的边界来填充;另一类则是由于物体遮挡关系发生变化引起的,对这类洞,我们采用多向视域插值来填充.实验结果表明,我们的

* 本文研究得到国家自然科学基金和霍英东青年教师基金资助.作者鲍虎军,1966年生,博士,研究员,主要研究领域为计算机图形学,虚拟现实,计算机动画.陈莉,女,1968年生,博士,讲师,主要研究领域为计算机图形学,科学计算可视化.王章野,1965年生,讲师,主要研究领域为计算机图形学,红外成像技术.彭群生,1947年生,博士,教授,博士导师,主要研究领域为真实感图形基础算法,计算机动画,虚拟现实.

本文通讯联系人:鲍虎军,杭州 310027,浙江大学CAD&CG国家重点实验室

本文1998-02-28收到原稿,1998-04-20收到修改稿

方法在准确性和有效性上都比传统的方法有了很大改进.

1　视域插值的基本原理

当摄像机在场景中移动时，大部分物体的可见性在相邻帧中是保持不变的. 视域插值技术正是充分利用了相邻帧的这种相关性来加速中间画面的生成.

我们先来看一下一个可见点在相邻视平面上的投影位置关系. 设 $O_iU_iV_iW_i(i=1,2)$ 是两个相邻采样点处的摄像机坐标系(如图1所示)，其中 O_i 是视点的位置，W_i 是视线的单位方向向量，取 $W_i=1$ 作为视平面，设 (u_i,v_i) 为一个点 P 在坐标系 $O_iU_iV_iW_i(i=1,2)$ 中视平面上的投影位置，d_i 为点 P 相对于 O_i 的深度值，则

$$u_2=\frac{(R\cdot U_2)}{(R\cdot W_2)},\qquad v_2=\frac{(R\cdot V_2)}{(R\cdot W_2)},\tag{1}$$

其中 $R=O_1-O_2+d_1(u_1U_1+v_1V_1+W_1)$，$d_1=(P-O_1)\cdot W_1$.

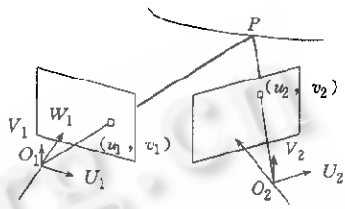


图1　一个可见点在相邻帧画面上投影位置的关系

由此，可得到点在两个相邻视平面上投影位置的差，即偏移向量，记为 (u_2-u_1,v_2-v_1). 将第1个视平面上所包含的所有象素的偏移向量组成一张图，称为偏移向量图，由它可确定出一对图象间的中间过渡图象.

2　非线性视域插值

由于透视变换是非线性的，用线性插值方法计算可见点在中间视平面上的投影位置显然仅是一个粗略的近似，尤其是当一个可见物体在两个相邻视点处深度值变化剧烈时会产生很大的误差. 本节给出一种非线性视域插值方法，它可以模拟出透视变换的效果，从而准确地计算投影位置.

在预处理中，我们先将场景采样并组织成一个一般的三维图的结构，设 $O_i(i=0,1,2,3)$ 是4个相邻的结点，则它们定义了一个局部空间. 在这个局部空间内的一点 O 可被表示为

$$O=O_0+\alpha_1(O_1-O_0)+\alpha_2(O_2-O_0)+\alpha_3(O_3-O_0),\tag{2}$$

其中 $\alpha_1,\alpha_2,\alpha_3$ 是这个局部空间的仿射变换系数.

设在每个结点 O_i 处均已建立了环境立方体，并且都具有相同的朝向，其摄像机坐标系为 O_iUVW，则一点在两个坐标系 O_iUVW 和 O_jUVW 中的深度差 D_{ij} 可由下式得到.

$$D_{ij}=(O_j-O_i)\cdot W.\tag{3}$$

可见，任何一点在两个坐标系中深度值的变化与它的位置坐标无关. 由上面两个方程得

$$D=\sum_{i=1}^{3}\alpha_iD_{0i},\tag{4}$$

其中 $D=(O-O_0)\cdot W$. 如前所述，场景中一点 P 可被同时表示作

$$P=O_i+(u_iU+v_iV+W)d_i\quad(i=0,1,2,3),\tag{5}$$

$$P=O+(uU+vV+W)d,\tag{6}$$

其中 (u_i,v_i) 和 (u,v) 分别是 P 点在 O_i 和 O 坐标系中视平面上的投影位置，d_i 和 d 是其相应的深度值. 由这些方程，我们可得到

$$d_i=(P-O_i)\cdot W=d_0-D_{0i}\qquad d=(P-O)\cdot W=d_0-D$$

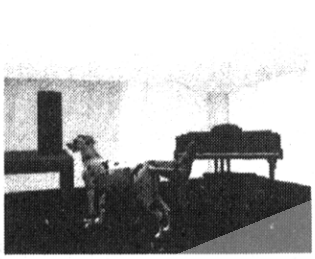
$$u=\frac{\left[u_0+\sum_{i=1}^{3}\alpha_i(u_i-u_0)\right]d_0+\sum_{i=1}^{3}\alpha_iu_i(d_i-d_0)}{d_0+\sum_{i=1}^{3}\alpha_i(d_i-d_0)}=u_0+\sum_{i=1}^{3}\frac{d_i}{d}\alpha_i(u_i-u_0).\tag{7}$$

$$v=\frac{\left[v_0+\sum_{i=1}^{3}\alpha_i(v_i-v_0)\right]d_0+\sum_{i=1}^{3}\alpha_iv_i(d_i-d_0)}{d_0+\sum_{i=1}^{3}\alpha_i(d_i-d_0)}=v_0+\sum_{i=1}^{3}\frac{d_i}{d}\alpha_i(v_i-v_0).$$

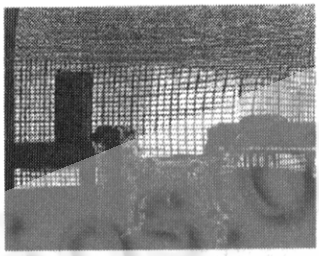
通过预先计算好的偏移向量图，我们可以很容易地计算出点 P 在坐标系 O 中视平面上的投影位置 (u,v). 容易看出，在推导这个新的插值方法的过程中，我们没有作任何假设，因而它的结果是精确的. 与线性插值方法相比，我们引进了一个非线性因子 $(1/d)$，故我们将它称为非线性视域插值方法. 当摄像机沿着垂直于 W 轴的平面移动时，P 点相

对于 O 的深度值保持不变，在这种情况下，这两种方法所得到的结果是一样的，也只有在这种情况下，线性插值方法得到的结果才是精确的.

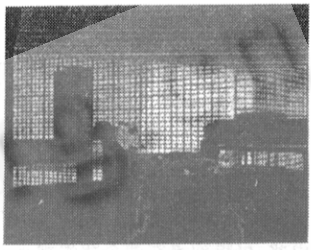
图 2 给出了一个室内场景的例子，其中，(a)和(d)分别是源图象和目标图象；采用本文的方法和线性插值方法分别得到(b)和(c)，当将这两幅图象与精确图象(e)进行比较时，容易看出，采用线性插值方法生成的图象误差较大，如(f)所示，而采用本文方法所生成的图象则效果较好，如(g)所示.



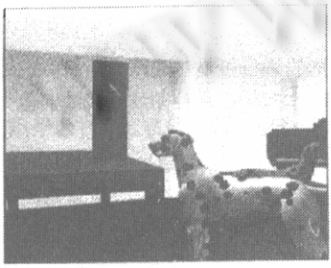
(a) 源图象



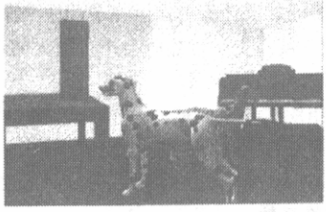
(b) 用本文方法生成的图象



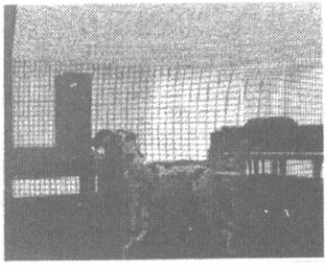
(c) 线性视域插值生成的图象



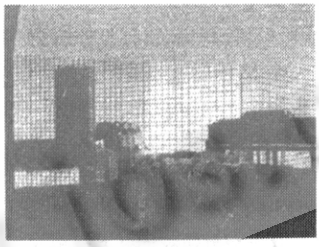
(d) 目标图象



(e) 精确图象



(f) (b)和(e)半透明叠加后的图象



(g) (c)和(e)半透明叠加后的图象

图 2 本文方法与线性视域插值方法的比较

3 源图象的分解

一种有效的加速视域插值过程的方法就是将具有相似偏移向量的象素组成一块，从而使得这一块图象中的象素可作为一个整体一起变换. 为处理一般情况，我们引进了一个新的误差矩阵

$$\Delta[i_1,i_2]\times[j_1,j_2]=\max_{1\leqslant k\leqslant n}\sqrt{(u^k_{\max\ off}-u^k_{\min\ off})^2+(v^k_{\max\ off}-v^k_{\min\ off})^2}\ , \tag{8}$$

其中 $n(n=1,2,3)$表示维数，$[i_1,i_2]\times[j_1,j_2]$表示源图象中左下角为(i_1,j_1)、右上角为(i_2,j_2)的一个矩形区域，参数 $u^k_{\max\ off}$，$v^k_{\max\ off}$，$u^k_{\min\ off}$和 $v^k_{\min\ off}$分别定义如下.

$$u^k_{\max\ off}=\max_{(i,j)\in block[i_1,i_2]\times[j_1,j_2]}(offmap[k][i][j].u) \qquad v^k_{\max\ off}=\max_{(i,j)\in block[i_1,i_2]\times[j_1,j_2]}(offmap[k][i][j].v)$$

$$u^k_{\min\ off}=\min_{(i,j)\in block[i_1,i_2]\times[j_1,j_2]}(offmap[k][i][j].v) \qquad v^k_{\min\ off}=\min_{(i,j)\in block[i_1,i_2]\times[j_1,j_2]}(offmap[k][i][j].v)$$

其中结构数组 off*map*[*k*]表示从源图象到第 *k* 维目标图象的偏移向量图，因此，图象分解的终止条件可取为 $\Delta_{[i_1,i_2]\times[j_1,j_2]}<Epixel$，其中 *Epixel* 由用户定义的阈值 ε 决定，其取值方法将在后面讨论.

因为我们是基于块对源图象进行视域插值的，分解后块图象的数目大小至关重要. Eric Chen 等人采用的四叉剖分方法，由于其严格的子分模式使得产生的块图象的数目很大，为此，我们提出了源图象的二叉剖分算法.

二叉剖分算法首先分别沿水平方向 *U* 和垂直方向 *V* 扫描当前块图象，设 *Ucut* 和 *Vcut* 是沿 *U* 方向和 *V* 方向最优的分割线位置，它们可由下式确定.

$$f(Ucut)=\min_{i_1\leqslant i\leqslant i_2} f(i)\ ,\qquad g(Vcut)=\min_{j_1\leqslant j\leqslant j_2} g(j)\ . \tag{9}$$

其中

$$f(i)=(i-i_1+1)(j_2-j_1+1)\Delta_{[i_1,i]\times[j_1,j_2]}+(i_2-i)(j_2-j_1+1)\Delta_{[i+1,i_2]\times[j_1,j_2]},$$

$$g(j)=(j-j_1+1)(i_2-i_1+1)\Delta_{[i_1,i_2]\times[j_1,j]}+(j_2-j)(i_2-i_1+1)\Delta_{[i_1,i_2]\times[j+1,j_2]}.$$

当前块图象最优的子分方向和子分位置则可由下式得到.

$$CutDirection=(f(Ucut)<g(Vcut))?\ U:V\ .$$

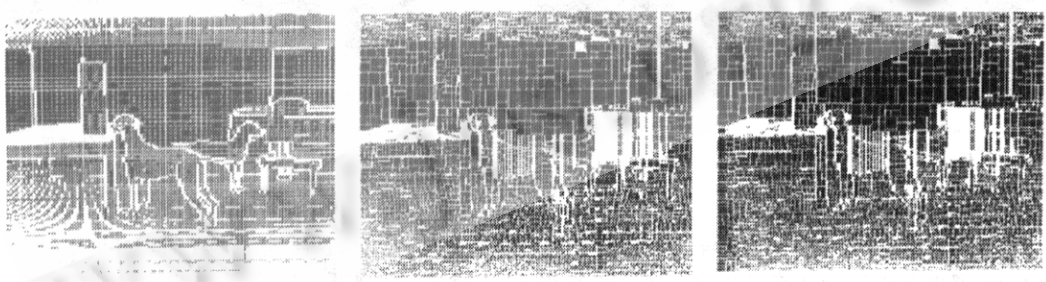
$$CutPosition=(f(Ucut)<g(Vcut))?\ Ucut:Vcut\ .$$

对源图象的每一块图象动态确定这两个参数，就可用一简单的递归过程完成对源图象的分解. 为了进一步优化算法，在分解过程中可动态地由下式决定阈值.

$$Epixel=\begin{cases}\varepsilon & \text{如果}[i_1,i_2]\times[j_1,j_2]\text{不会移出中间画面}\\ \beta\varepsilon & \text{如果}[i_1,i_2]\times[j_1,j_2]\text{将会移出中间画面}\ .\end{cases}$$

其中 $\beta=\max\{\frac{voff}{height-j_1},\frac{uoff}{width-i_1},-\frac{voff}{j_2+1},-\frac{voff}{i_2+1},1.0\}$，uoff，voff 是源图象中最小的块图象偏移向量的两个分量，*width* 和 *height* 是图象的分辨率，阈值的动态选取可避免在偏移向量差别很大的区域产生太多的小块图象，这些区域一般位于源图象的边界附近和中间视图视域锥体之外.

虽然我们的方法在预处理阶段比四叉剖分方法费时，但它并没有增加视域插值的计算耗费. 为了提高可见性测试的效率，可预先对源图象二叉树的所有叶结点进行深度排序，然后建立优先级块表. 实验结果充分表明了我们提出的二叉剖分方法的优势，如图 3 所示，其中，(a)是采用四叉剖分生成的结果，将源图象分成了 33 316 块，(b)是采用我们提出的二叉剖分方法且没有动态决定阈值的结果，只需分成 14 493 块，而采用动态确定阈值，其块数更降低到 13 948 块，结果见(c). 以上图象剖分的阈值为两个象素(ε=2).



(a)　源图象的四叉剖分　　(b)　没有动态确定阈值的二叉剖分　　(c)　动态确定阈值的二叉剖分

图 3　四叉剖分与本文的二叉剖分方法的比较

4　黑洞填充算法

最精确的黑洞填充方法是采用光线跟踪算法重新计算黑洞所在象素的光亮度值[8]，但其计算量是很大的. Chen 和 Williams 采用了插值相邻象素颜色或偏移向量的方法填充黑洞[6]，但由于该方法逐个象素地填充黑洞，其时间耗费也是比较大的. 我们分两个步骤对黑洞进行高效填充.

4.1　扩展块图象

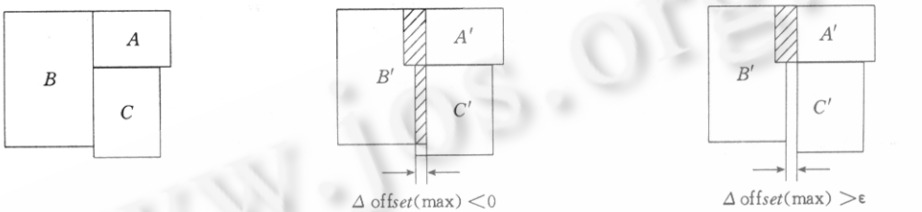
我们首先填充由于局部图象扩展而产生的黑洞，其基本思想是，在与洞相邻的块图象中，根据它们的深度值及其位置确定出相应块并扩展其边界来将黑洞覆盖.

因为深度值低的块图象通常遮挡深度值高的块图象，我们先将所有具有低深度值的块图象组成一个候选块表，然后对于候选块表中块图象的每条边界，决定是否将该边界往外扩展一个象素. 不失一般性，以块图象 B 的右边界为例(如图 4 所示)，首先找到与 B 的边界相邻的所有块图象，将它放入块表 $Blist$ 中，然后将这些块图象与块 B 沿 U 方向的偏移向量进行比较.

$$\Delta offset(\max)=\max_{k=1,2,3}\{\max_{b\in Blist}(b.offmap[k].u-B.offmap[k].u)\},$$

其中 $offmap[k].u$ 表示一个块图象从 O 处的源图象到 O_k 处的目标图象其偏移向量沿 U 方向的分量，这时可能会出现下面 3 种情况：

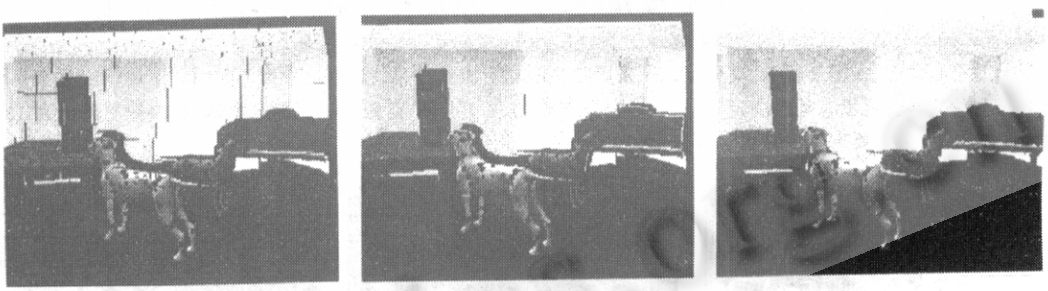
- $\Delta offset(\max)<0$，块图象 B 在变换到新的位置后将沿着这条边覆盖相邻块图象(见图 4(b))；
- $\Delta offset(\max)>\varepsilon$，变换后块图象 B 与相邻块图象之间沿着这条边有洞出现(见图 4(c))；
- 否则，块图象 B 与其相邻块图象之间在变换后由于局部图象扩展出现黑洞.



(a) 原始块图象 (b) 变换后块图象间出现覆盖 (c) 变换后块图象间出现分离

图4 变换到中间画面后块图象间的关系

显然，只有在第 3 种情况下，块图象 B 右边界需往外扩展一个象素. 对块图象 B 其他边界的检测可采用类似方法进行. 当检测完候选块表中所有块图象后，黑洞填充方法的第 1 步就执行完毕. 图 5(b)说明了第 1 步的填充结果.



(a) 没有填充黑洞时的图象 (b) 扩展块图象填充后的结果 (c) 四向视域插值填充后的结果

图 5 黑洞填充方法的比较

4.2 多向视域插值

按照上节给出的方法对黑洞填充后，中间画面上也许还有一些黑洞存在，这种类型的洞往往很难填充，因为它们中的大部分是由于遮挡关系改变而引起的. Chen 和 Williams[6]建议通过加密采样来减少这些洞的存在. 这里，我们给出一个简单、有效的方法，不需增加任何采样点.

不失一般性，以一维情况为例，设 A 和 B 是两个相邻采样点处的图象，按上节中给出的方法，分别取 B 作为目标图象、A 为源图象和以 A 作为目标图象、B 作为源图象来生成两幅图象 C_A 和 C_B. 由于 C_A 和 C_B 中大部分黑洞通常是互补的，因此，中间画面 C 可由图象 C_A 叠加到图象 C_B 或 C_B 叠加到 C_A 上而得到，其合成顺序可根据 C 点到 A 点和 B 点的距离来决定. 当然，这样处理后，图象 C 中可能还有少量黑洞存在，这时一个较好的解决方法是直接插值相邻象素颜色. 因为这类洞所占比例很小，插值相邻象素的结果还是可以接受的. 图 5(c)是由四向视域插值生成的，图象中的黑洞被自动填充.

5 结 论

本文提出了一种非常有效的视域插值算法. 与传统的视域插值方法相比，本文提出的非线性视域插值方法能够准

确地模拟出透视变换效果.我们还提出了对源图象进行二叉剖分的方法,极大地减少了分解后块图象的数目.为了更有效地填充中间画面上的黑洞,我们分两步进行填充,首先在中间画面上与洞相邻的块图象中,根据深度信息选择适当的块扩展其边界,然后通过多向视域插值来得到中间画面.实验结果表明,本文提出的方法结果更精确,速度也很快,能在微机上实现复杂场景的漫游.今后我们准备探索对空间采样和对图象反走样的更有效的方法,从而使视域插值方法生成的图象效果更好.

参考文献

1 Blinn J F, Newell M E. Texture and reflection in computer generated image. Communications of ACM, 1976,19(10):542～547

2 Greene N. Environment mapping and other applications of world projections. IEEE Computer Graphics and Applications, 1986, 6(11):21～29

3 Greene N, Kass M. Approximating visibility with environment maps. Technical Report No. 41, Apple Computer Inc., 1993

4 Watt A, Watt M. Advanced animation and rendering techniques. New York: ACM Press, 1992

5 Chen S C. QuickTime VR: an image-based approach to virtual environment navigation. Computer Graphics, 1995,29(4):29～38

6 Chen S C, Williams L. View interpolation for image synthesis. Computer Graphics, 1993,27(2):279～288

7 FU Sheng, BAO Hu-jun, PENG Qun-sheng. An accelerated rendering algorithm for stereoscopic display. Computers & Graphics, 1996,20(2):223～229

8 Levoy M, Hanrahan P. Light field rendering. In: Holly Rushmeier ed. Computer Graphics, Annual Conference Series, Addison-Wesley Publishing Company, 1996. 31～42

Nonlinear View Interpolation

BAO Hu-jun　CHEN Li　WANG Zhang-ye　PENG Qun-sheng

(*State Key Laboratory of CAD & CG　Zhejiang University　Hangzhou* 310027)

Abstract　A new image-based rendering technique is presented in this paper. Unlike the linear interpolation scheme, this method can exactly simulate the perspective viewing transformation during the walkthrough. To accelerate the view interpolation, the algorithm employs a binary subdivision scheme to optimize the decomposition of the source image so that the number of the resultant blocks is greatly reduced. Holes on the intermediate image are filled by two steps, namely enlarging the transferred blocks at the side adjacent to holes and retrieving the local image within the holes from the destination images by multiple-directional interpolation. Experimental results demonstrated the algorithm is much more accurate and efficient than the traditional one.

Key words　view interpolation, environment map, perspective transformation, binary subdivision, virtual reality.