第 34 卷　第 5 期　　计　算　机　工　程　　2008 年 3 月
Vol.34　No.5　　Computer Engineering　　March 2008

·人工智能及识别技术·　文章编号：1000—3428(2008)05—0207—03　文献标识码：A　中图分类号：TP391

基于纹理贴图及高斯滤波的纹理反走样方法

周珍艮[1,2]，郭　立[1]，杨　毅[1]

(1. 中国科学技术大学电子科学与技术系，合肥 230026；2. 铜陵学院电气工程系，铜陵 244000)

摘　要：MIPmap 是当前各种硬件图形加速卡普遍支持的一种纹理映射方法，但它存在着过模糊和明显的走样现象。该文提出基于 MIPmaps 纹理数据结构的纹理映射算法，改进了纹理细节层次的计算方法，在存储带宽受限制的条件下尽量多存取纹理元素，并对存取的纹理采用高斯加权，从而改善了绘制图像的质量。

关键词：MIPmap 方法；纹理；走样；高斯滤波

Anti-aliasing Method Based on MIPmaps and Gaussian Filtering

ZHOU Zhen-gen[1,2], GUO Li[1], YANG Yi[1]

(1. Department of Electronic Science and Technology, University of Science and Technology of China, Hefei 230026;
2. Department of Electric Engineering, Tongling College, Tongling 244000)

【Abstract】In current graphics acceleration hardware, MIPmap is a common used filtering technique. But it exists blurring and aliasing artifacts. This paper describes a texture mapping technique, which is based on MIPmaps. With the restricted number of texels loaded from memory for real-time filtering, it improves the algorithms of LOD, and filters the texels with Gaussian filter. As the result, the visual quality is improved.

【Key words】MIPmap method; texture; aliasing; Gaussian filtering

在计算机图形和图像处理中，走样是由计算机的数字化特点决定的。对无限带宽的自然的连续图像取样为离散信号时，采样频率达不到原始信号最高频率的 2 倍，图像重建时就会产生走样。对计算机中已经数字化了的纹理图像，假设对纹理样本的数字化已经进行了前置滤波，或纹理样本的数字化的取样频率满足频率限制，那么在纹理映射的重新取样过程中，纹理映射后的图像产生了透视变形，侧影轮廓边和透视变换在图像空间产生高频细节，这是导致产生走样的原因。纹理反走样主要是针对重新取样过程。纹理映射反走样的关键问题是确定每个屏幕像素所对应的纹理像素，并用适当的滤波器对纹理像素进行卷积以获得最后的纹理颜色值。

1　纹理反走样的方法

纹理贴图最简单方法是以点取样方式从纹理图像中读取数据：将屏幕像素映射到纹理空间，读出最近的纹理颜色值来替换屏幕像素颜色值。该方法简单、快速，在屏幕像素与纹理的大小一致时，这种取样方法能够产生可以接受的图像质量，对纹理放大，则会出现色块，纹理缩小时产生波纹[1-2]。

基于分级细化的纹理贴图(MIPmap)[1]由于运算量小和易于硬件实现，成为最普遍的纹理滤波方法，当前各种图形加速卡都以各自的方式对它进行支持。在实时图形硬件中，先对原始的纹理图像进行分级细化，即对原始的纹理图像不断进行分辨率减半的采样，称为不同细节层次(LOD)，并按照金字塔方式存储在纹理存储器中。在进行纹理贴图时，先将屏幕空间的像素映射到纹理空间，找到分辨率最接近的纹理图像，然后对像素映射到该层次坐标点最近的 4 个纹理元素进行双线性插值。纹理空间所对应的细节层次可能介于 2 个层次之间，这时的双线性插值应在 2 个细节层次上进行，对 2 个双线性插值的结果再进行一次插值，即三线性插值，得到最终的像素颜色值。经典的MIPmap采用双线性或三线性滤波方法，每次访问的纹理元素数量为 4 或 8，这样解决了实时系统的效率问题，但由于欠采样，在解决走样时会导致图像的模糊。

区域求和表法及椭圆形权重平均(EWA)[2]也是常用的纹理滤波方法，但由于它们所需的计算量太大，一直没有在实时系统中得到实现。区域求和表法的效果要好于分级细化的方法。EWA是软件实现的最好方法，已经成为评价各种滤波方法的基准。在纹理映射中，高质量和效率通常是相互冲突的。许多现有的方法焦点关注高质量而不重视效率(如区域求和表法和EWA法)，或者解决了效率而图像质量不佳(如MIPmap)。

对目前的图形硬件来说，提高纹理滤波的一种最常用方法就是利用现有的分级细化图形硬件，采用分级细化的数据结构，增加纹理采样数量，改进后置滤波方法。

2　分级细化与高斯加权方法

MIPmap 提取固定数量纹理元素进行滤波，纹理访问的时间是固定的，但存在着欠采样的问题。为解决这个问题，必须要访问更多有用的纹理，但实时系统中，带宽的限制一直是一个瓶颈。在带宽所允许访问的纹理数量 M 限制下来进行细节层次参数 λ 的选择，可在保证效率的同时提高纹理访问量。对所访问的纹理进行后置滤波时加权的方法有平均赋值法，也有基于子像素面积覆盖率赋值方法，这里采用基于每纹理像素的高斯赋值方法来提高图像质量及减小走样。高斯滤波器衰减更快，没有反向凸起，常用它来代替理论上最优的低通滤波器 sinc。

作者简介：周珍艮(1969－)，男，讲师、硕士研究生，主研方向：实时图形学，集成电路设计；郭　立，教授、博士生导师；杨　毅，博士研究生

收稿日期：2007-03-10　**E-mail：**zhgzhou@mail.ustc.edu.cn

2.1 分级细化层次选择

分级细化的层次 λ 是纹理在不同分辨率的图像金字塔中所处的层次，它决定了屏幕像素所要取样的纹理图像分辨率。

在MIPmap中，每个屏幕像素被看成是一个单位正方形，映射到纹理空间后为一任意四边形，每个像素采样 4 个纹理。这里将四边形的包围盒所覆盖的纹理数量定为存储带宽所限定的值M，它依赖于不同的系统。四边形的 4 个顶点是浮点位置，为了减小计算量，先要将这 4 个顶点转换到整数位置($P_0P_1P_2P_3$)。转换的方法是将 4 个顶点的坐标加 0.5 或减 0.5 后取整[3]。设u, v为纹理空间坐标，映射到原始纹理图像上的包围盒宽度和高度分别为D_U, D_V，$Width$及$Height$为原始纹理图像的宽度和高度，它们的积就是每个像素在原始纹理上所占的纹理数量，包围盒边长取最长边，对最长边缩小，使每个屏幕像素点在纹理图像层次λ上导入的纹理数量 为M，就得出最终的分级细化的层次：

$$\lambda = \left\lfloor \mathrm{lb}[\frac{\max(Width \times D_U, Height \times D_V)}{\sqrt{M}}] \right\rfloor \tag{1}$$

这样，在存储带宽允许的前提下，能从所对应的层次上存取最大限度的纹理像素来覆盖四边形，以减小图像欠采样所引起的模糊。

2.2 高斯滤波器

图 1 为屏幕空间的高斯滤波器，超过 1 的部分截断为 0，成为一个底部为正方形的形状，以此来对应屏幕上一个单位的正方形像素区域。图 2 为经过透视投影到纹理空间高斯滤波器形状，该滤波器底部形状即为像素正方形在纹理空间映射的四边形，包围盒纹理数量为 M，见图 3。滤波器的高是对应纹理像素在该位置的权值，距离投影中心点相等位置权重也相等，形成四边形等高线。

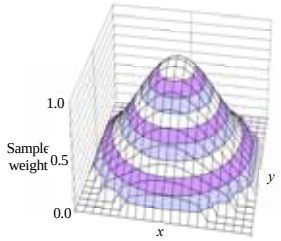


图 1 屏幕空间的圆形高斯滤波器

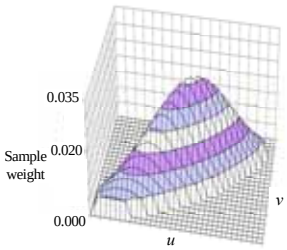


图 2 透视投影到纹理空间的高斯滤波器

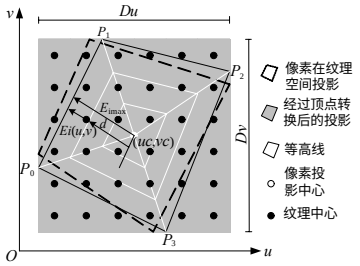


图 3 高斯权重等高线

2.3 高斯权重的计算

最终像素颜色值可由下式来计算[4]：

$$P(x,y) = \sum_l \left(T(l) \times W(l)\right) / \sum_l W(l) \tag{2}$$

式中，$P(x, y)$为滤波后像素的颜色值，l为所要读取的纹理，$T(l)$是四边形中的每个纹理颜色值，$W(l)$为每个纹理的高斯权重。高斯权重由下式表示[1]：

$$W(l) = \mathrm{e}^{-\alpha \cdot d \cdot d} \tag{3}$$

其中，a 为高斯权重系数；d 指从四边形投影中心到纹理的规范化距离(0 到 1)；投影中心为像素中心在纹理空间投影点；等高线有相同的规范化距离 d。

要计算每个纹理的权重$W(l)$，首先要判断该纹理被哪条等高四边形包围，并计算它与边的距离E_i，然后用投影中心到该边的最大距离$E_{i\max}$对E_i进行规范化处理，得出最终的d。

给边赋以方向，图 3 所示四边形边方向为顺时针方向，纹理中心到边P_0P_1的距离用线性函数即边函数E_i表示[5]：

$$E_i = (u - u_0)\mathrm{d}u + (v - v_0)\mathrm{d}v \tag{4}$$

其中，$\mathrm{d}u = \Delta V / (|\Delta U| + |\Delta V|)$；$\mathrm{d}v = -\Delta U / (|\Delta U| + |\Delta V|)$；$\Delta U = u_1 - u_0$；$\Delta V = v_1 - v_0$；d$u$和d$v$分别为纹理空间$u$和$v$方向的增量；$P_0$和$P_1$为边的起始点和终点；$\Delta U$和$\Delta V$为纹理中心$(u, v)$到边的坐标增量，并令 $\mathrm{d}u \times \Delta U + \mathrm{d}v \times \Delta V = 0$，见图 4。$E_i$的绝对值表示该点到边的距离，为正时，点在边的右边，为负在左边，点在边上则函数值为 0。

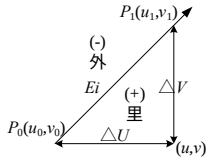


图 4 边函数

如果一个纹理中心到四边形每条边的E_i都为正，则纹理被该四边形包围。这样纹理权重的d值为

$$d = 1 - \frac{|E_i|}{|E_{i\max}|} \tag{5}$$

当纹理像素在四边形中心时，d=0，这时的权值最大。等高线离中心越远，d 值越趋向于 1，权值也越小。

权重的计算以查找表来实现，以加速运算，即将每个纹理的权重预计算并存储在 ROM 中，存储器的大小为 M×每个纹理权重所需的字节数。

2.4 反走样算法的步骤

纹理映射反走样算法过程描述如下：

Step1 计算从屏幕空间到纹理空间的映射矩阵；

Step2 将屏幕空间单位正方形映射到纹理原始纹理上，计算细节层次系数 λ；

Step3 扫描屏幕空间四边形，计算每个像素在λ层纹理投影中心坐标(u_c,v_c)及四边形各顶点坐标(u_i,v_i)，i=0,1,2,3；

Step4 扫描纹理四边形包围盒，计算每个纹理对应的 d；

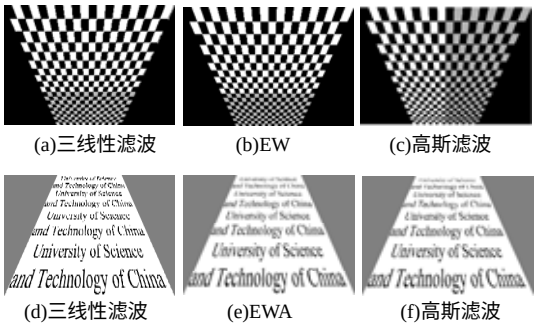
Step5 查找表读取纹理权重 $W(l)$；

Step6 将每个纹理权重与纹理值相乘并累加，再除以权重和；

Step7 将纹理颜色值取代屏幕像素的颜色值。

3 结果分析和比较

图 5 为 C++仿真结果。三线性滤波存在较明显的远处模糊、近处走样现象，见图 5(a)。而改进的基于分级细化数据结构及高斯滤波的方法，见图 5(b)，图像的清晰度和锯齿现象较三线性滤波方法有很大的改进，与 EWA 法结果相比较，图 5(c)在兼顾实时性的前提下，实验结果的主观评价还是比较理想的。它们的信噪比(SNR)在表 1 中列出，高斯滤波方法的 SNR 要比三线性滤波方法高 3 dB ~6 dB，但比 EWA 方法要低 7 dB ~9 dB。表中列出的是针对文字图案滤波的，棋盘格由于黑白交界变化更明显，其 SNR 较低。



(a)三线性滤波 (b)EW (c)高斯滤波

(d)三线性滤波 (e)EWA (f)高斯滤波

图 5　几种滤波方法的结果

表 1　3 种滤波方法的 SNR 比较　dB

	高斯滤波	三线性滤波	EWA
M=16	19.55		
M=32	21.32	15.87	29.06
M=64	21.41		

表 2 列出几种滤波方法所需的存储器资源及所要进行的运算。

表 2　几种滤波方法运算量及所需资源比较

		MIPmap	EWA	高斯滤波
每个采样纹理地址计算	+/-	1	没有限制	1
	×/÷	无	没有限制	无
后置滤波	+/-	5,12	36~100	$\leq 2M$
	×/÷	6,13	10~26	$\leq M+1$
存储器		1.33 倍	1.33 倍	1.33 倍+4M
读取纹理数量		4,8	9~25	$\leq M$

3 种滤波方法都采用透视投影，屏幕空间像素坐标映射到纹理空间所需的计算是相同的，表中未列出。对 MIPmap 算法，采样的纹理数量为 4 或 8 个，所以纹理地址计算需 4 或 8 次加，EWA 和高斯滤波方法都采用扫描线增量算法，但前者运算量最大。后置滤波时 MIPmap 的计算量是固定的，而 EWA 则没有最大限制，它取决于所用滤波方法及所要读取的纹理数量。另外，纹理数据都采用分级细化的方法存储，所需存储器为原始纹理图像的 4/3，但高斯滤波方法需要另外的权重表，M=64 时，所需额外的存储量为 256 B。

4　结束语

各种纹理映射的方案在解决走样问题时，都必须兼顾图像的清晰度和走样现象，实时系统同时还要考虑算法的效率。为了提高图像质量，本文利用 MIPmap 分级细化纹理存储数据结构，重新改进了双线性滤波中纹理图像层次的计算方法，在带宽允许的范围内，尽可能访问更多有用纹理，以解决欠采样，并对所存取的纹理采用高斯加权。另外考虑到带宽限制，高斯权重采用查找表来实现，减少了大量的实时计算，从而保证了系统的效率。

参考文献

[1] Heckbert P S. Fundamental of Texture Mapping and Image Warping[R]. Berkeley: Computer Science Division, University of California, Tech. Report: UCB/CSD 89/516, 1989-06.

[2] Heckbert P S. Survey of Texture Mapping[J]. IEEE Computer Graphics and Applications, 1986, 6(11): 56-67.

[3] Hüttner T, Straer W. Fast Footprint MIPmapping[C]//Proceedings of the 1999 EUROGRAPHICS/SIGGRAPH Workshop on Graphics Hardware. Napa, Californiaaaa: [s. n.], 1999: 35-44.

[4] Akenine-Möller T. 实时计算机图形学[M]. 2 版. 普建涛, 译. 北京: 北京大学出版社, 2004.

[5] Schilling A. A New Simple and Efficient Antialiasing with Subpixel Masks[J]. ACM Computer Graphics, 1991, 25(4): 133-141.

（上接第 173 页）

表 1　各参数设置下的结果(每组数据实验 10 次)

(N_{pop},M_{gen},P_c, P_m)	最优值	最差值	平均收敛时间/s	平均收敛代数	平均收敛率/(%)
(10,100,0.7,0.10)	1574.96	1604.93	15.2	56.3	90
(10,100,0.8,0.15)	1574.96	1604.93	14.7	49.2	100
(20,100,0.7,0.10)	1574.96	1574.96	11.8	87.2	100
(20,200,0.8,0.15)	1574.96	1574.96	13.4	66.5	100
(30,200,0.7,0.10)	1574.96	1574.96	12.5	70.0	100
(30,200,0.8,0.15)	1574.96	1574.96	12.0	68.9	100

最优定位和最优流如图 2 所示，线上圆圈中的数字是运输量。椭圆中的数字是最优生产量和接受的返回容器数量，加深圆圈表示生产地，加深椭圆中的数字分别表示生产产品和接收的返回品数量。最优定位方案中有 5 个生产地，分别是 A,C,F,J,K，目标函数值为 1 574.96。图中 F 和 K 已经达到最大的生产能力，C 已达到接收返回品的最大能力限制。

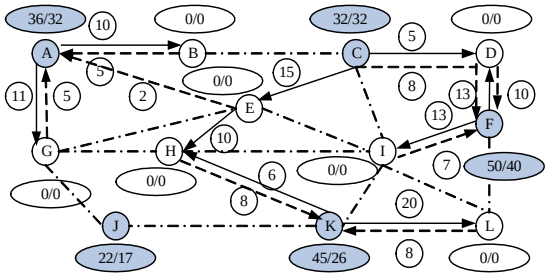


图 2　最优方案

4　结束语

本文提出了逆向物流直接再利用网络的能力受限模型，使用遗传算法作为启发式算法进行求解。通过该约束处理，以自适应选择、交叉和变异概率设计，使 DRN 问题既能确定最优方案，又能避免传统算法必须求上下限的繁琐迭代过程。仿真和比较结果表明，该算法具有较好的收敛性和搜索能力。

参考文献

[1] Rogers D S, Tibbens-Lembke R S. Going Backwards: Reverse Logistics Trends and Practices[M]. Pittsburgh, PA: RLEC Press, 1998.

[2] Barros A I, Dekker R, Scholten V. A Two-level Network for Recycling Sand：A Case Study[J]. European Journal of Operational Research, 1998, 110(2): 199-214.

[3] Bloemhof-Ruwaard J M, Salomon M, Van Wassenhove L N. The Capacitated Distribution and Waste Disposal Problem[J]. European Journal of Operational Research, 1996, 88(3): 490-503.

[4] Del Castillo E, Cochran J K. Optimal Short Horizon Distribution Operations in Reusable Container Systems[J]. Journal of Operational Research Society, 1996, 47(1): 48-601.

[5] Fleischmann M, Bloemhof-Ruwaard J M, Dekker R, et al. Invited Review, Quantitative Models for Reverse Logistics: A Review[J]. European Journal of Operational Research, 1997, 103(1): 1-17.

[6] Thierry M C, Salomon M, Van Nunen J, et al. Strategic Issues in Product Recovery Management[J]. California Management Review, 1995, 37(2): 114-35.

[7] Dolgui A, Sdek J, Zaikin O. Supply Chain Optimization——Product/Process Design, Facility Location and Flow Control[M]. [S. l.]: Springer, 2005: 151-166.

[8] Daskin M S. Network and Discrete Location, Models, Algorithms, and Application[J]. Wiley: Wiley-Interscience Publication, 1995, 4(4): 302-303.

[9] 胡运权. 运筹学教程[M]. 北京: 清华大学出版社, 2003.