

改进区域求和表算法

范波 吴慧中

(南京理工大学信息学院, 南京 210094)

摘要 分析了一般的纹理映射反走样算法,提出了对纹理映射几何变换过程简化的改进区域求和表算法。算法通过对几何变换的局部线性化使得映射后象素多边形为平行四边形,然后在区域求和表法的基础上对矩形包围盒进行矩形自适应分割来完成反走样处理。实验表明改进算法在计算量增加不大的基础上提高了图形质量。

关键词 纹理映射 区域求和表 反走样

0 引言

目前,纹理映射已广泛应用于三维计算机图形的生成以提高图形的真实感。纹理映射实际上是一个由屏幕上最终显示的象素到纹理图案的映射及采样过程,但是由于采样频率不能满足香农采样定理即高于原始信号中最高频率的2倍,故不能准确地从采样信号再现原始信号而造成走样。为此,人们提出了许多反走样算法。尽管这些反走样算法在实现上差异很大,但原理基本相同。

当对屏幕区域进行采样时,如果每个象素只有一个采样点,便会产生严重的图形走样。为了获得合适的反走样图象,必须考虑对每个图象元素产生影响的多边形区域部分,而不能只考虑单一的采样点中心,这通常可以通过两种途径来实现。

(1) 多点采样

就是在一个象素区域内设置多个采样点,对这些采样点的值进行加权平均来得到该象素的值。多点采样方法有两种,一种是采样范围限制在一个象素的区域内,也就是说对每个象素区域采用相同的点分布方式采样。这类似于采用子象素的反走样方式,通常可以用一个Box函数来实现。另一种是把采样范围限制在多个象素包围的区域内,可以应用更高级的采样函数来实现。在一般的算法中,通常把采样范围限制在包含9个象素的区域内。在该区域内可以使用高斯采样函数或其他采样函数用两种方式采样。第一种方法是用密度函数把采样点分布在各个象素中心,用高斯采样函数对各采样点的值进

行加权平均,这相当于使用了一个圆形滤波器,采样点距圆心越近其权值越大。第二种方法是根据高斯函数值的大小分布采样点的数目,对每个采样点赋予相同的权值,即按重要性分布采样点,在重要的象素区域设置较多的采样点,在不重要的象素区域设置较少的采样点。

(2) 区域采样

在区域采样中,象素被看成一个区域而不是一个单独的点,可计算被象素多边形覆盖的区域面积来计算象素的值。此法已被广泛应用于点和线段的反走样算法中。在文献[1]中William讨论了mipmap方法,提出了简单且有效的反走样纹理映射方法,他采用 2×2 金字塔数据模型,利用原始的纹理图案通过双线性插值方法,计算出的一系列不同分辨率下的纹理模型。采用这种算法,在纹理映射过程中,每个图象元素的处理时间基本相同。但该算法需要储存许多不同分辨率下的纹理模型,故需要大量的存储空间和执行查表操作;另一方面该算法假定:在低分辨率下的任何一个纹理元素的值都可在高分辨率下的一个矩形区域内的纹理元素的平均值来表示,并且在该区域内纹理图案的压缩是对称的,因此,在极个别的情况下,会出现图象模糊现象。尽管该算法以牺牲图形质量和存储空间为代价换取高的映射速度,但在许多场合下取得了较好的效果,已被广泛应用。

在文献[2]中Crow提出了区域求和表算法来计算纹理平面中与 u, v 坐标轴分别平行的矩形的纹理平均值,矩形即为象素多边形映射至纹理平面内多边形的包围盒。由于它计算的包围盒更加紧密故

效果也比 mip-map 法好。在该算法中,区域求和表代替了 William 算法中的多级纹理表。纹理空间中任何一个矩形区域内纹理元素的平均值,都可以通过简单的代数运算从该表中求出,从而减小了所需的存储空间,避免了查表操作,提高了纹理映射速度。mip-map 法和区域求和表法不能完全消除走样,因为它们处理的都不是象素映射后的多边形,还包括了多边形以外的区域,当包围盒比实际的多边形区域大得多的时候,会导致严重的走样。

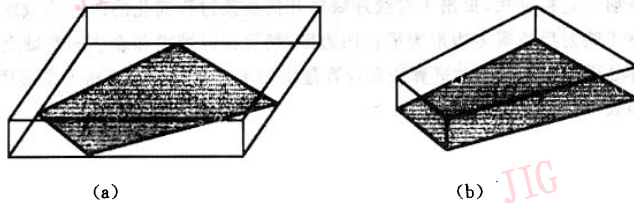


图1

在文献[3]中提出的自适应纹理映射算法,实际上是对映射后的多边形进行卷积,即一个在多边形内才为1的滤波器,见图1(b)。由于没有考虑映射的几何过程的简化,而只能把象素多边形映射至纹理平面的多边形的可能情况均加以考虑,如映射后的4个顶点所组成的可能会是四边形、三角形、共线或共点,也就造成了多种不同情况下的判断,使得算法较为繁琐。

为了对映射后多边形的各种情况进行简化,有必要对纹理映射的变换过程作一讨论。纹理映射将纹理平面 $\{(u, v)\}$ 变换至 3D 表面 $\{(x', y', z')\}$, 再利用投影将 3D 表面变换至屏幕 $\{(x, y)\}$ 。设 f 为 $\{(u, v)\}$ 到 $\{(x', y', z')\}$ 的变换, p 为 $\{(x', y', z')\}$ 到 $\{(x, y)\}$ 的投影变换。于是 $g(u, v) = p(f(u, v))$ 是从纹理平面到屏幕的映射函数。见图 2(a)。

$$\begin{bmatrix} x \\ y \end{bmatrix} = g(u, v) = \begin{bmatrix} g^1(u, v) \\ g^2(u, v) \end{bmatrix}$$

$$\begin{bmatrix} u \\ v \end{bmatrix} = h(x, y) = \begin{bmatrix} h^1(x, y) \\ h^2(x, y) \end{bmatrix}$$

但实际上我们需要的是 $h(x, y)$, 即从屏幕上象素位置得到对应的纹理值。对于一般的纹理映射过程,多采用反向映射技术,故 f^{-1} 是已知的,则 $h(x, y) = p^{-1}f^{-1}$ 。由于各种物体的建模方法不同,如可采用三角形面片拼接、圆锥曲面片拼接或参数样条曲面片拼接等方法来构造,而投影变换也有透视投影和平行投影之分,所以 $h(x, y)$ 会有多种形式而导致文献[3]中的对多种情况的处理。如果我们对 $h(x, y)$ 作以简化,则需要处理的情况将大大减少以

1 局部线性近似

纹理采样可以用一个卷积 $\sigma = \iint h(x - \xi, y - \eta) f(\xi, \eta) d\xi d\eta$ 来描述,其中 f 为纹理函数, h 为一卷积核。区域求和表法就是一个高度为 1 的盒形滤波器,即包围盒内 $h=1$, 而包围盒外 $h=0$, 见图 1(a)。

节省时间。下面讨论对 $h(x, y)$ 的局部线性近似来对纹理映射中的变换进行简化。

首先将屏幕 (x, y) 平面划分为正方形网格, 设每一象素的边长为 d , 取任一方格的中心为 (x_0, y_0) , 在 (x_0, y_0) 处 $h(x, y)$ 的泰勒展开式分别为:

$$h^1(x_0 - d, y_0) = h^1(x_0, y_0) - d \cdot h_x^1(x_0, y_0) + d^2 \cdot h_{xx}^1(x_0, y_0) + o(d^3)$$

$$h^1(x_0 + d, y_0) = h^1(x_0, y_0) + d \cdot h_x^1(x_0, y_0) + d^2 \cdot h_{xx}^1(x_0, y_0) + o(d^3)$$

$$a_{11} = \frac{h^1(x_0 + d, y_0) - h^1(x_0 - d, y_0)}{2d}$$

$$= h_x^1(x_0, y_0) + o(d^2)$$

$$a_{12} = \frac{h^1(x_0, y_0 + d) - h^1(x_0, y_0 - d)}{2d}$$

$$= h_y^1(x_0, y_0) + o(d^2)$$

$$a_{21} = \frac{h^2(x_0 + d, y_0) - h^2(x_0 - d, y_0)}{2d}$$

$$= h_x^2(x_0, y_0) + o(d^2)$$

$$a_{22} = \frac{h^2(x_0, y_0 + d) - h^2(x_0, y_0 - d)}{2d}$$

$$= h_y^2(x_0, y_0) + o(d^2)$$

于是我们可以得到 h 的线性近似函数 \hat{h} 为:

$$\hat{h}(x, y) = \begin{bmatrix} \hat{h}^1(x, y) \\ \hat{h}^2(x, y) \end{bmatrix} = \begin{bmatrix} h^1(x_0, y_0) \\ h^2(x_0, y_0) \end{bmatrix} + \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} d_1 \\ d_2 \end{bmatrix}$$

$$d_1 = x - x_0, d_2 = y - y_0$$

h 的线性近似实际上是一个屏幕空间和纹理

空间的仿射变换,也即将屏幕上的一正正方形象素区域映射为纹理平面上的一个平行四边形,反过来其逆映射纹理平面中的一矩形区域映射为屏幕上的一

个平行四边形,见图 2(b)。通过以上处理后,对我们下一步的工作将带来极大的方便。

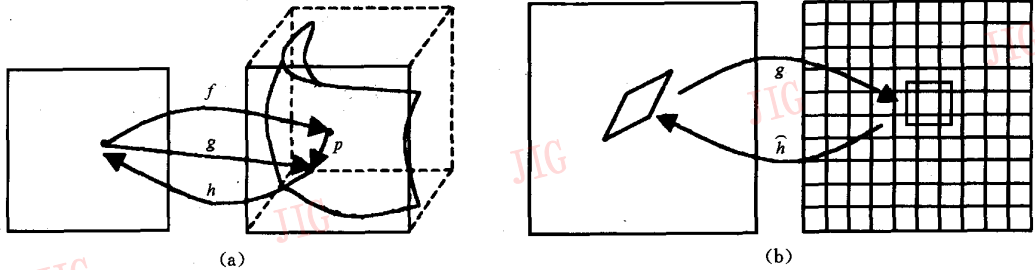


图 2 纹理映射过程及简化后的结果

2 区域求和表的矩形分割

通过局部线性化后,我们所需要处理的情况就变得非常简单,即仅对平行四边形而不是 4 个点任意构成的平面图形进行卷积计算。由于区域求和表法只能计算与 u, v 坐标轴平行矩形的纹理平均值,故对平行四边形外包围盒内的部分只能进行矩形分割。简化后平行四边形和包围盒的相对关系可分为两种情况:一是平行四边形有两条边与包围盒的边重合,见图 3(a)。二是平行四边形任一条边不与包围盒的边重合,见图 3(b)。

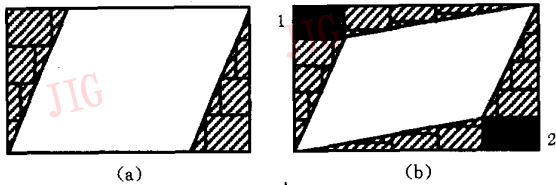


图 3 平行四边形和包围盒的关系

以下是算法具体的执行过程:

```

{
  计算包围盒角点坐标;
  计算平行四边形面积 S 和包围盒面积 T;
  if (S/T ≤ 常数 k) Sub=I else Sub=J;
  if (平行四边形 4 个顶点均在包围盒边上){
    对两个阴影三角形细分 Sub 次;
    计算各个小矩形纹理和值;
  }
  else {
    以不在包围盒边上的平行四边形顶点和包围盒对应顶点作矩形 1 和 2;
    对剩余 4 个阴影三角形矩形分割 Sub 次;
    计算 1 和 2 矩形纹理值与分割阴影三角形的小矩形纹理值的和值;
  }
}

```

象素纹理值 = (包围盒内纹理值 - 各小矩形内纹理值的和) / 平行四边形的面积;

(其中常数 k 可以根据生成图形速度和图形质量的要求来调节)

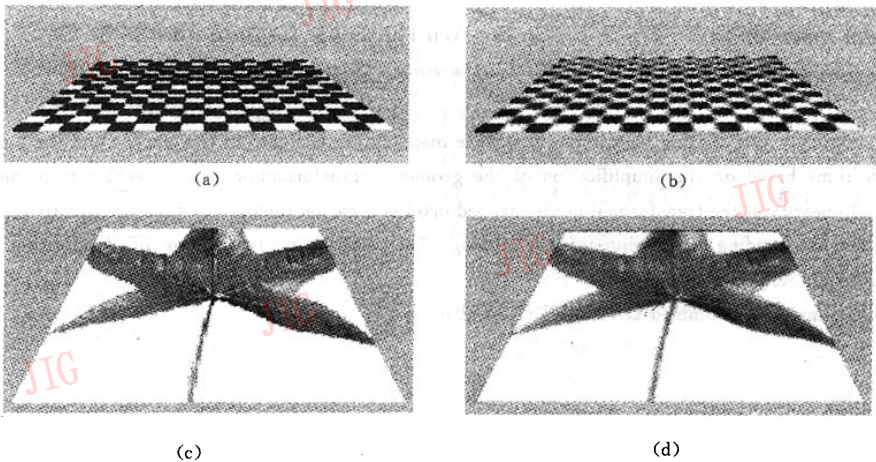


图 4 算法结果对比

由以上讨论可以看出,改进的区域求和表算法在数据预处理阶段即建立区域求和表的计算和原来完全一样。虽然由于对变换过程进行线形化和对包围盒内的矩形分割增加了计算量,但是改进算法通过第一次的线形化后,以后的象素变换到纹理平面的计算采用增量法可以显著减少处理时间,所增加的矩形分割的计算量为(16次纹理寻址时间+14次乘法+27次加法)×分割步数,当变换过程比较复杂时,两种算法的计算量相差不大,但效果却相差较大。图4中(a)为未改进前映射黑白象棋图案的算法结果,(b)为改进后的结果,可以看出在(a)中远处的走样在(b)中得到了很好的反走样处理,(c)、(d)分别为对叶子的 bmp 位图处理前和处理后的结果。

3 结 论

本文中的反走样算法仍采用区域求和表方法对原始的纹理图案进行预处理,但通过纹理映射过



范波 博士研究生。主要研究领域为计算机图形学和虚拟现实技术。



吴慧中 教授,博士生导师。主要研究领域为智能CAD,计算机图形学及虚拟现实技术等。

程的简化,使反走样算法计算的象素映射后多边形仅为平行四边形,并进一步通过矩形分割,使本算法在处理速度不比原始区域求和表算法降低太多的基础上图形质量大大提高。而且通过分割步数选取的不同,得到不同显示质量和速度的图形,可以适应不同的要求。

参 考 文 献

- 1 Williams L. Pyramidal parametrics. *Computer Graphics*, 1983, 17(3):1~11.
- 2 Crow F. Summed-area tables for texture mapping. *Computer Graphics*, 1984, 18(3):207~212.
- 3 Masaaki Oka. Real-time manipulation of texture-mapped surfaces. *Computer Graphics*, 1986, 21(4):181~187.
- 4 Franklin C. A comparison of antialiasing techniques. *IEEE Computer Graphics & Application*, 1981, (1):40~48.
- 5 Glassner A. Adaptive precision in texture mapping. *Computer Graphics*, 1986, 20(4):297~306.
- 6 张英杰,张铁昌. 快速纹理映射技术. *微电子学与计算机*, 1994, (4):18~21.

A Modified Summed-Area Tables Algorithms

Fan Bo, Wu Huizhong

(Department of Computer Science, Nanjing University of Science & Technology, Nanjing 210094)

Abstract The commonly anti-aliasing algorithms in texture mapping are analyzed in this paper, and the modified summed-area tables algorithms based on the simplification of the geometry transformation in the texture mapping are proposed. Through the local linearity of the transformation the mapped pixel become parallelogram, then the adaptive rectangle subdivision is proceeded on the bonding box to eliminate the aliasing. The result shows that the modified algorithm can improve the quality and reduce the calculation at one time.

Keywords Texture-mapping, Summed-area tables, Anti-aliasing