三维地形高质量实时矢量叠加绘制

李 融，郑文庭
（浙江大学 CAD & CG 国家重点实验室 杭州 310058）
（wtzheng@cad.zju.edu.cn）

摘 要：在高低起伏的三维地形上无缝叠加二维矢量数据是三维地理信息系统可视化的必须功能. 为了进一步提高矢量纹理的像素有效利用率，减轻走样，提出一种采用多级联的绘制方法来改进传统基于纹理的矢量叠加绘制过程的算法. 该算法采用深度剖分和屏幕空间分块 2 种策略对视域四棱锥进行分割，对子四棱锥内矢量进行级联绘制，并对绘制范围进行调整优化，整个算法过程完整地利用了现代 GPU 可编程硬件来实现. 实验结果表明，文中算法适用于大范围多分辨率地形上的矢量绘制，绘制过程达到了实时，绘制效果令人满意.

关键词：地理信息系统；矢量数据；三维地形；级联绘制；GPU 可编程管线
中图法分类号：TP391

Real-time Rendering High-Quality Vector Data on 3D Terrain

Li Rong and Zheng Wenting
(*State Key Laboratory of CAD & CG, Zhejiang University, Hangzhou* 310058)

Abstract: In the 3D geographic information systems (GIS), it is necessary to overlay 2D vector data on a 3D multi-resolution terrain model. This paper presents a precise and efficient texture-based overlaying algorithm. It generates vector textures from cascaded view-depended cameras to improve the utilization of texture pixels. The strategies of both depth splitting and screen space dividing are further adopted. All steps are based on programmable GPU pipeline. It runs in real-time and is suitable for displaying large-scale terrain with LOD algorithms.

Key words: GIS; vector data; 3D terrain; cascade rendering; GPU programmable pipeline

随着计算机技术和数据获取技术的迅速发展，以及在日益增长的三维空间信息需求的牵引和蓬勃发展的现代新兴技术的驱动下，现代地理信息系统(geographic information system, GIS)的发展正迅速从传统的二维 GIS 走向三维 GIS，甚至四维 GIS(动态三维). 从二维 GIS 到三维 GIS 虽然空间维数只增加了一维，但包容了几乎所有丰富的空间信息，也突破了常规二维表示对真实世界表达的束缚，可更好地洞察和理解现实世界.

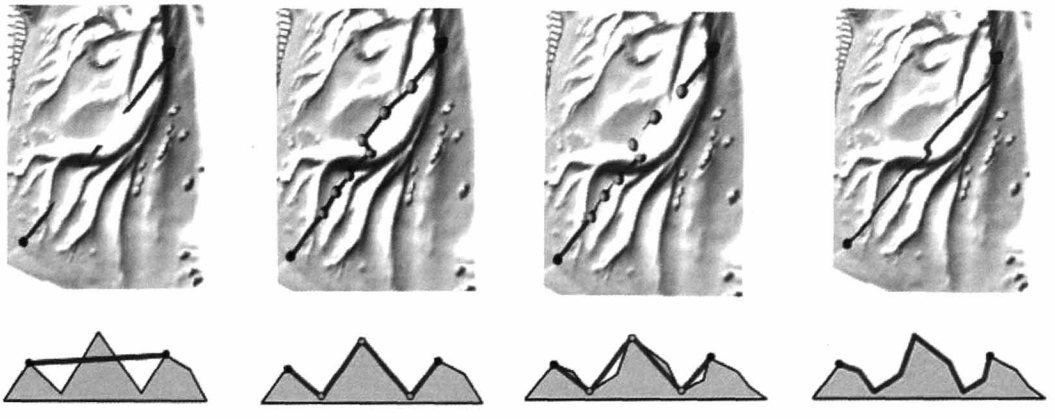
在三维 GIS 应用中，二维矢量数据[1]仍是一种必不可少的表达方式，它包括点、线、面(多边形)等，用于表达地物信息，如道路、湖泊、河流、地名等. 当前，越来越美观、逼真的多种分辨率多时相遥感影像数据和全球数字高程模型数据在网络环境下更易获取，二维矢量地图、遥感影像和三维地形表面模型的混合[2]及其应用已成为普通百姓习以为常的基本导航定位方式；同时在各种 GIS 分析功能中，二维矢量数据的可视化应用也非常普遍. 因此，如何在高低起伏的多层次细节的三维地形[3]上不断叠加二维矢量数据，并保证其高质量的实时可视化效果，是 GIS 领域普遍关注的问题.

收稿日期：2010-09-20；修回日期：2011-04-18. 基金项目：国家“八六三”高技术研究发展计划(2008AA121603，2009AA12Z229). 李 融(1985—)，男，博士研究生，主要研究方向为虚拟现实、计算机图形学；郑文庭(1974—)，男，博士，副研究员，主要研究方向为虚拟现实、计算机图形学.

1　相关工作

在三维地形表面上叠加绘制二维矢量数据的绘制算法，大体可分为基于几何的绘制方法和基于纹理的绘制方法两大类.

基于几何的绘制方法将二维矢量数据模型附着在三维地形表面，它需要解决二维矢量与三维地形的几何匹配问题. 如果矢量与地形不匹配，则可能出现矢量悬浮在空中或穿刺入地表等错误情况，必须根据相应地形几何模型，在矢量数据中动态地引入或者删除新的顶点或线段[4]，从而生成与地形几何表面相匹配的数据模型，如图 1 a 所示. 对采用连续层次细节绘制的地形模型则需要每帧都调整折线的几何节点[5]，这在大矢量数据时效率难以令人承受. 邹烷等[6]采取预计算并存储的方法来避免每帧几何重计算，但却大大增加了数据量. 基于模板阴影体的矢量数据绘制方法[7]需要先将矢量数据扩充成多面体，在地形绘制完成后将该多面体第一遍绘制到模板缓存形成一层掩模，再次绘制该多面体时即可完成矢量绘制；不同颜色的矢量需要分开依次绘制，这是一个高代价的算法.

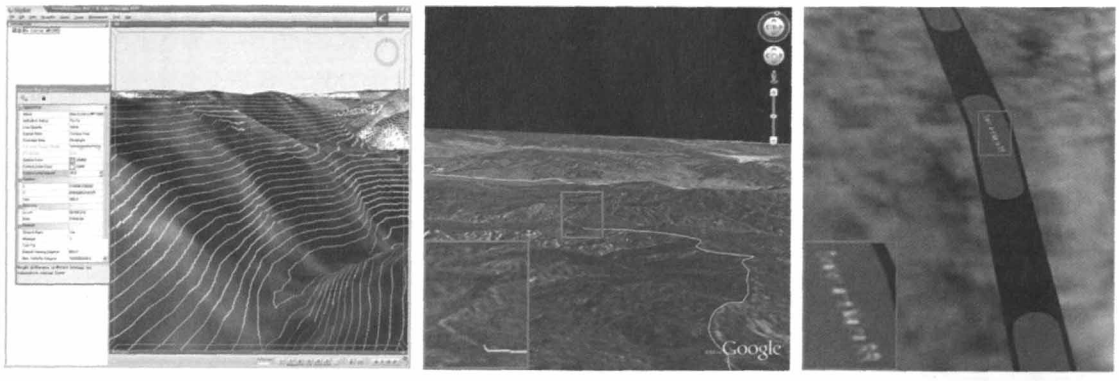


a 矢量几何模型以适应不同层次地形　　b 基于纹理绘制

图 1　基于几何和纹理方法对比

基于纹理的绘制方法首先用一个光栅化的过程将矢量数据转化为标准纹理；然后在绘制地形的图形流水线过程中该矢量纹理被映射到三维地形顶点上，从而实现矢量数据在地形表面的无缝叠加，如图 1 b 所示. 基于纹理的方法不需要调整矢量几何，其主要问题在于有限的纹理分辨率带来的绘制走样. 动态的矢量纹理生成方法[8]通过控制视点与地物距离，按需生成适当精度的矢量纹理，可以有效地降低绘制走样. 该方法在 Schneider 等[9]的方法中被进一步改进，其中使用类似透视阴影贴图方法[10]来分析走样原因，经过重参数化调整绘制矢量数据的视域四棱锥，提高了矢量绘制纹理的有效像素利用率. 陈鸿等[11]指出，当随视域方向和地形夹角变化时可近似选取局部地形高度来控制矢量纹理的绘制精度，并采取透视投影代替正交投影来产生矢量纹理，能进一步提高像素利用率. 以上算法虽然一步步地减少了基于纹理的方法的绘制走样程度，但是由于绘制过程与矢量纹理生成过程的采样密度不一致，以及局部地形可能起伏坡度过大，走样现象依然存在，特别在大规模地形中远距离部分走样现象往往比较严重.

在当今世界主流的 GIS 三维软件中，走样现象依然比较普遍. Skyline 软件使用的是基于纹理的方法，在绘制线型矢量时经常会在地形上变大放粗，且边缘粗糙不平滑，如图 2 a 所示. Google Earth 软件使用的是基于几何的方法，高空俯视时效果良好，但是接近地面近乎水平视角时依然会出现上文所描述的悬浮和穿刺地形等不良状况，如图 2 b 所示. 除此之外，对应逐渐精细级别地形时，矢量数据也需要不断更新级别，当符合当前地形精细高程级别的矢量数据顶点位置尚未加载完成或者缺失的情况下，通常采取隐藏不绘制的方法；所以当视点靠近地形时边界线等原本清晰的矢量信息就会消失. 当这些问题严重时，会大大影响到 GIS 领域很多分析功能的可视化反馈效果. 如图 2 c 所示，基于纹理的英文字母的绘制也容易存在走样现象，由于中文的复杂性远高于英文字母，在同等走样程度下中文比英文更难以辨认，在绘制中文文字矢量时做到精确的反走样显得尤为重要.



a Skyline 等高线　　b Google Earth 边界线断裂　　c 文字绘制模糊

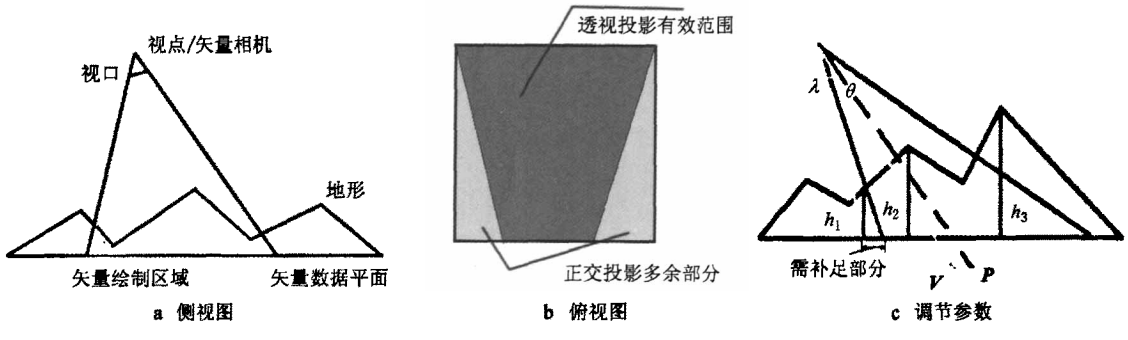
图 2 典型软件走样分析

2 本文算法

基于纹理的方法的优化重点在于如何减小甚至消除走样. 本文根据当前视点位置和方向，采用对当前视域棱锥按深度划分或者屏幕空间分块策略，对视域四棱锥进行分割；绘制地形时，在 GPU 片元着色器中根据对应深度计算查找采用与之相对应的级联纹理，以实现矢量数据在三维地形上的无缝贴合. 该方法能提高纹理的像素利用率，从而达到提高纹理分辨率减小走样的目的. 在绘制矢量时，基于级联绘制范围和矢量数据空间位置进行分析，通过优化绘制矢量的范围，提高矢量纹理的实际利用率，进一步减小走样. 在实际应用过程中，可以通过对大量的矢量信息根据视点距离进行宏观逻辑上的分类，同时在绘制时对当前绘制的矢量信息在每个子视域四棱锥内部进行基于包围盒的四叉树剔除，完成对海量矢量数据的高质量实时绘制.

2.1 视点相关矢量纹理的实时动态生成

本文算法需实时生成某一区域、某一分辨率级别的矢量纹理，但如果视口内矢量数据所占像素数远大于该区域数据在矢量纹理上对应的像素数，就会形成严重的采样失真. 本文采用了文献[11]方法，采用透视投影方式把视点位置作为生成矢量纹理的相机位置，如图 3 a 所示. 比起传统的正交投影方式，它可使矢量纹理在视点近处提高纹理分辨率，并部分排除正交投影冗余，从而有效地提高矢量纹理图像素利用率，如图 3 b 所示.



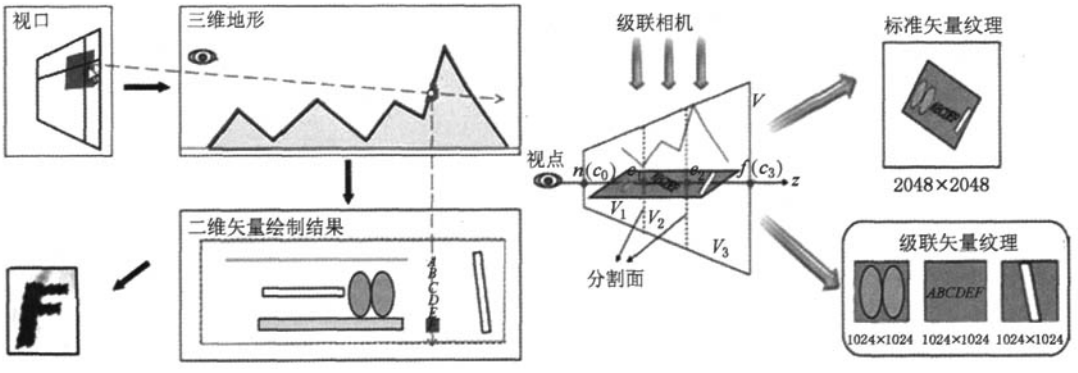
a 侧视图　　b 俯视图　　c 调节参数

图 3 视点相关的矢量纹理生成和调节张角

在绘制地形时，本文将矢量视域四棱锥的透视投影矩阵 M_t^{proj} 和模型视图矩阵 M_t^{model} 作为变量传入 GPU；在 GPU 顶点着色器中，利用地形顶点 $A(x,y,z)$ 与矢量数据的 $A'(x,0,z)$ 的一一对应关系，可快速并行计算出每个顶点的矢量纹理坐标；在 GPU 片元着色器中，使用得到的纹理坐标访问矢量纹理，将正确的颜色值混合到地形的片元上，以实现无缝叠加.

但实际应用中，该算法还是容易形成下采样走样现象. 如图 4 a 所示，当生成矢量纹理的零平面与当前地形高度差较大或者出现陡峭地形时，将使得视口内显示矢量所需的分辨率远超过矢量纹理所能提供的分辨率而形成下采样. 当视线方向近乎平行于地形时，走样现象将更为明显. 另外，为了覆盖所

有必要的矢量信息，在矢量纹理生成时需要适当扩大生成矢量纹理相机的张角大小并调节视线方向. 如图 3c 所示，其中 θ 为视点张角，λ 为 θ 张大一定角度后的生成矢量相机张角，为视线方向，$\boldsymbol{V}$ 为 $\boldsymbol{P}$ 调整一定角度后矢量相机视角. 理论上说，这些调节的变量应该由视域与地形的交点高度 h_1 决定，实时精确地计算 h_1 并不容易，可以用视域内最高点 h_2 替代，甚至用地块单位的最高点 h_3 来替代[11]；但随之矢量纹理多余的部分也在增加，使矢量纹理像素利用率下降，也会使算法控制变得复杂.



a 走样原因　　b 级联绘制

图 4　矢量纹理的级联绘制

2.2　矢量纹理的级联绘制

将视域内所有的矢量全部放在一个纹理绘制，在实践中证明难以保证在完整视域上的采样密度均匀，特别在大范围地形绘制时. 本文提出一种将矢量信息剖分成级联绘制的算法，根据视点位置和视线方向提供深度剖分和屏幕空间分块 2 种策略，可以自由地定义每一级所绘制的范围和纹理尺寸，将原本连续的矢量数据绘制到离散的级联纹理上. 这样可以更加优化每个矢量纹理的像素利用率，在整个视域内更好地控制采样密度，进一步减少下采样走样. 如图 4b 所示，将视域四棱锥按照深度剖分成 3 个部分，分别绘制对应深度级别的矢量纹理，采样密度比一张纹理要更合理，纹理显存消耗也更小. 同时，当应用级联绘制时无需考虑生成矢量相机的参数调节，也使得算法的鲁棒性加强. 具体的级联分割策略将在第 2.2.2，2.2.3 节讨论.

2.2.1　级联绘制的流程

级联绘制矢量策略的具体步骤如下：

Step1. 根据深度解剖分策略或者屏幕空间分块策略将视域四棱锥剖分或多个部分，并记录每个部分所见的地形投影到矢量信息平面的包围盒.

Step2. 构建一组级联相机，对 Step1 中包围盒内的矢量信息进行绘制，保存结果到纹理空间.

Step3. 绘制地形. 在 GPU 片元着色器上根据当前像素的深度值来查询对应的级联矢量纹理，取得对应的矢量颜色进行混合操作.

本文中要讨论的符号描述如表 1 所示.

表 1　符号描述

符号	描述
V	当前视点的视域四棱锥
V_i	V 的第 i 部分($1\leqslant i\leqslant m$)
W	矢量绘制的视域四棱锥
W_i	W 剖分后的第 i 部分
n 和 f	V 的近和远裁剪平面
m	V 剖分的数目
C_i	V 根据深度剖分 V_i 的阈值，$C_0=n$，$C_m=f$
T_i	对 V_i 内地形对应的矢量信息绘制到纹理空间
$CVM(m;[res])$	将 V 剖分成 m 个部分的剖分方案，res 是每个剖分的部分绘制的矢量纹理分辨率

2.2.2　深度剖分策略

按照深度进行剖分的策略，对视域四棱锥剖分的理念基于：处于不同视域四棱锥深度层的矢量数据需要不同的像素分辨率. 图 5 说明了 m 个深度剖分平面沿着深度方向位于 $\{C_i \mid 0\leqslant i\leqslant m\}$ 将 V 分割成 m 部分，其核心问题是如何指定剖分面 C_i 的位置. 本文采取了类似于图形学中对阴影贴图(shadow map)的走样分析过程[11]，对基于纹理的矢量叠加绘制方法的走样问题进行公式化.

由图 5 可见，当绘制一个深度为 dz 的地形表面时，对应矢量绘制纹理上的纹元大小为 $ds\times ds(s\in[0,1])$；同时，在视口上由地形表面投影出来的大小 dp 近似于 $n\,dy/z$. 图 5 中，ϕ 代表地形表面法向与到视口向量的夹角，θ 代表地形表面法向与矢量绘制面的夹角. 因为 $dy=dz\cos\phi/\cos\theta$，矢量绘制的走样错误 dp/ds 可以表示为

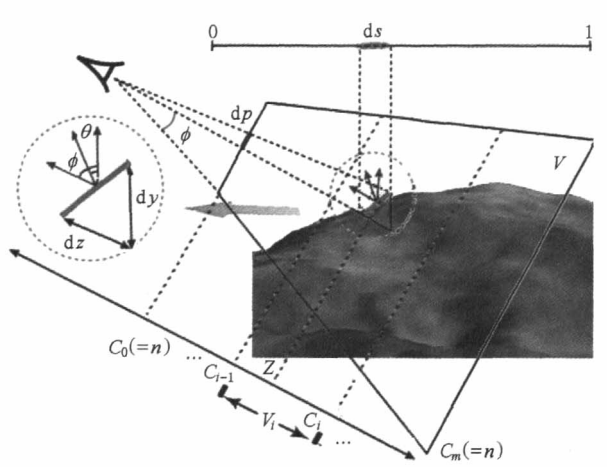


图 5 深度剖分视域四棱锥成 m 部分

$$\frac{\mathrm{d}p}{\mathrm{d}s}=n\frac{\mathrm{d}z}{z\mathrm{d}s}\frac{\cos\phi}{\cos\theta} \tag{1}$$

矢量绘制的下采样发生在式(1)的结果大于 1 时，这有可能由透视走样 $\mathrm{d}z/z\mathrm{d}s$ 和投影走样 $\cos\phi/\cos\theta$ 过大造成. 因为投影走样 $\cos\phi/\cos\theta$ 往往发生在地形与矢量数据面角度过大(比如竖面)，这类走样与地形几何细节密切相关，只能通过每帧的对地形几何特征的分析和复杂数据结构来完成，其代价非常高且不能通过现代硬件加速. 相反，透视走样 $\mathrm{d}z/z\mathrm{d}s$ 与场景无关，所以减轻透视走样不需要复杂的场景分析. 理论上最优化的透视走样分布要使得 $\mathrm{d}z/z\mathrm{d}s$ 在整个深度范围上保持一个定值，设 $\rho=\mathrm{d}z/z\mathrm{d}s$，则有

$$s(z)=\int_0^s \mathrm{d}s=\frac{1}{\rho}\int_n^z\frac{\mathrm{d}z}{z}=\frac{1}{\rho}\ln(z/n) \tag{2}$$

因为矢量参数 $s\in[0,1]$，所以可以得到 $\rho=\ln(f/n)$；进而解出 z(假设剖分的数量 m 非常大)，分割点应该按照指数排布，即

$$\{C_i\}:Z_i=n(f/n)^{i/m} \tag{3}$$

这种分割策略的缺点就是一切参数都是数学公式上推导的结果，而实际应用中深度 $z\in(n,f)$ 的矢量数据被完全绘制到归一化的 $s\in(0,1)$ 的纹理空间中没有浪费是不可能的. 只有当 $m\to\infty$ 时，才能保证每个级联上正交投影下的区域和视域完全重合不包含多余部分，满足推导的假设，而实际中级联的个数一般在 1～4 之间；同时，每个 i 级联上的纹理像素又要在总体矢量纹理像素的 i/m 也是难以做到的. 所以，这种分割策略并不适用于实际应用.

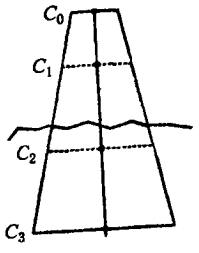
为了解决这个问题，实际应用中对于 C_i，可采取增加整个深度范围上的线性变化分割长度来取得折中平均策略，并由参数 $\lambda(0<\lambda<1)$ 来控制平均强度

$$\{C_i\}:Z_i=\lambda n(f/n)^{i/m}+(1-\lambda)(n+(i/m)(f-n)) \tag{4}$$

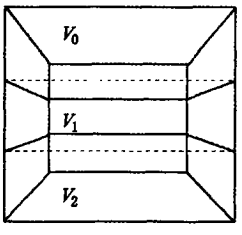
以上的级联位置确定方法是与场景无关的，因为每一帧根据地形位置和矢量数据进行准确的场景分析是高代价的. 经实验发现，按深度方向进行级联分割的策略可能造成距离视点近的区域空的比例过大. 在这种情况下，比较好的方法是直接延伸 C_0 放置的位置，并确保包含最近的矢量数据完整. 由于精确地求出每个矢量数据叠加在地形上对应的位置到视点的距离依然比较复杂，因此可以通过每个矢量数据在地形上的包围盒到视点的距离来近似计算.

2.2.3 屏幕空间剖分策略

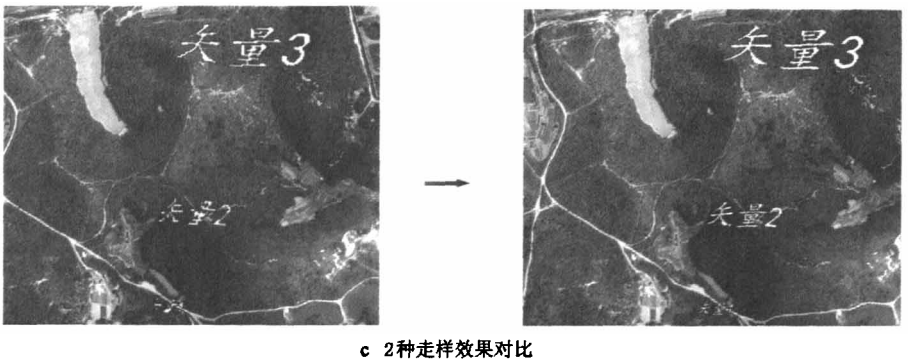
上述论证的按照深度进行剖分级联的策略，适



a 深度分割在该视点方向不合理



b 屏幕空间分割策略



c 2种走样效果对比

图 6 屏幕空间分割策略

用于多大数地形深度延伸比较宽广的漫游系统. 但在 GIS 的应用中经常需要鸟瞰局部地形，视线方向近乎于垂直于整个地形. 在这种情况下，由于在视域范围内的地形处于一个相对狭窄的深度范围内，基于深度的级联分割致使大面积地形集中在一个级联内，而其余级联没有任何价值，如图 6 a 所示. 因此，本文算法将采取另外一种更优化的剖分策略：将屏幕空间上直接切分成多个级联，从而将视域四棱锥切割成多个子四棱锥，如图 6 b 所示. 该策略过程与基于深度的级联算法过程基本类似，只是需要记录每个子四棱锥在矢量数据面上（$Z=0$）的范围，并在像素着色器上根据当前像素对应的地形顶点所处的范围来选择对应级联. 在漫游系统中，可以根据视点和地形夹角的阈值来灵活切换 2 种级联策略，实验证明在视线方向几乎垂直地面时，使用该策略反走样效果明显提升，如图 6 c 所示.

2.3　单独级联上范围绘制优化

本文把矢量绘制相机的视域锥 W 进一步分割成小的部分 $\{W_i\}$，W_i 可以通过计算 V_i 的包围盒在矢量数据面上的投影大小来构建. 每个 W_i 涵盖了 V_i 范围内的矢量信息，W_i 的集合覆盖了整个视域，它比单个 W 的像素使用率更高.

我们进一步对 W_i 进行优化，通过对矢量数据空间位置的分析，以剔除大量不包含任何矢量数据的绘制区域. 在初始阶段，对于所有的矢量数据可以构建它的二维包围框. 在 W_i 覆盖的矢量数据范围内部可以遍历所有矢量的包围框，与 W_i 在该区域的包围框一一求交；再把所有的求交后的包围框合并成为实际绘制所需要的包围框 B_i. 几乎所有的情况下，B_i 面积小于 W_i，其与矢量数据在 W_i 分布有关，越是数量小、小范围的矢量数据，则 B_i 越小，实际绘制到级联 i 上的矢量像素利用率越高. 图 7 中解释了对应级联 i 的 W_i 绘制全过程，B_i 和 B 分别代表了级联 i 的矢量正交视域包围框和整体视域包围框，显然 $\sum_{i=1}^{m} Area(B_i) \leqslant Area(B)$，从而使得矢量绘制的像素利用率更加提高.

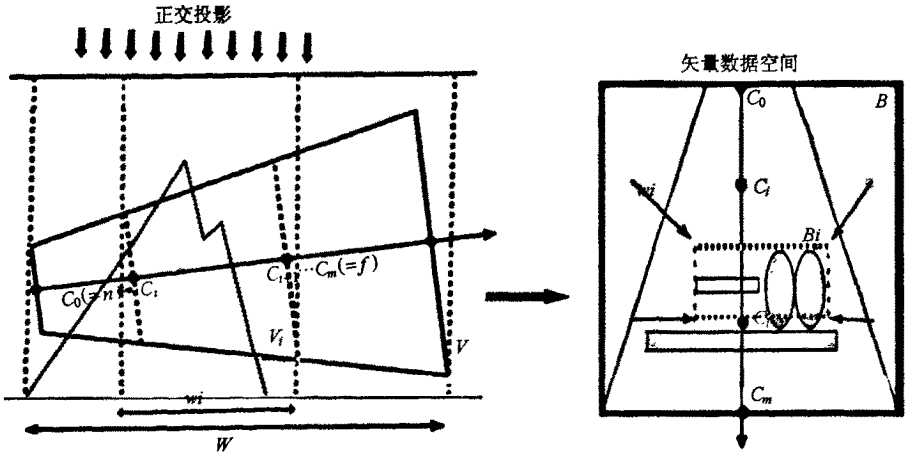
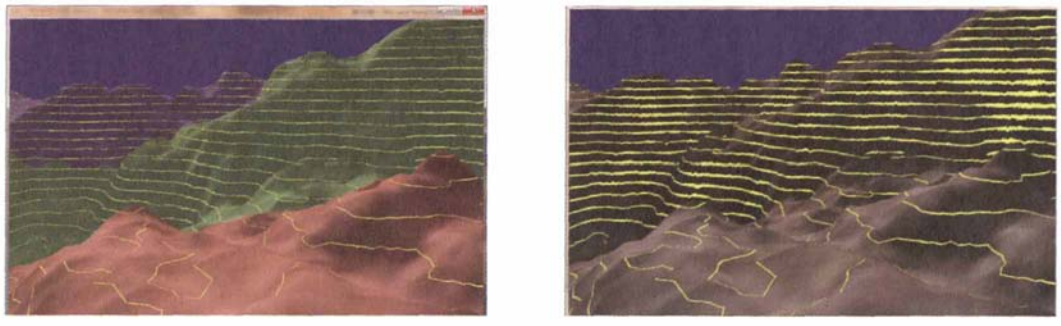


图 7　矢量绘制范围的第 i 级优化过程

3　实验结果

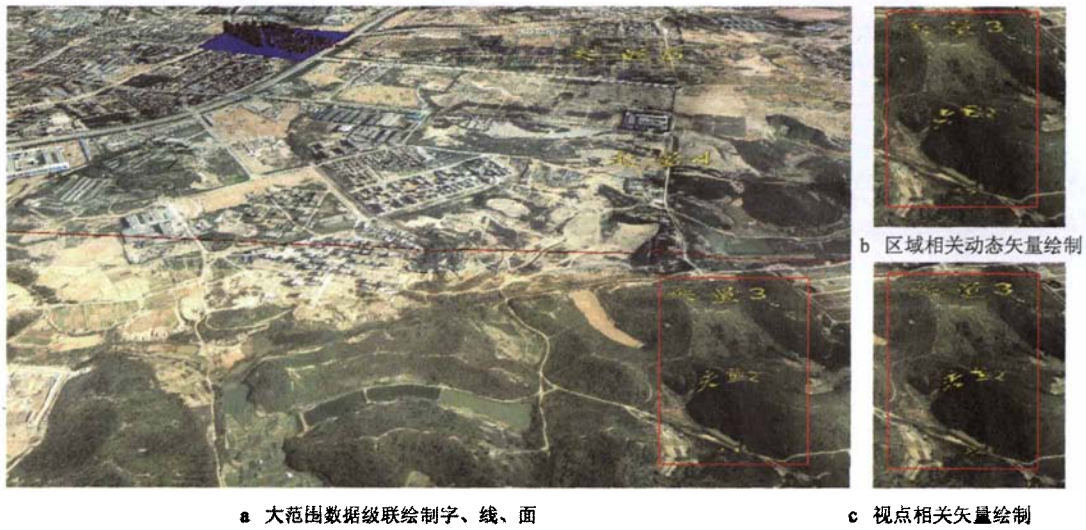
我们在 Win32 平台下使用 OpenGL2.0 实现了一个算法验证系统，该系统采用 OpenGL 的 TextureArray 扩展来加速动态生成矢量纹理数组，保证了绘制的高效实时. 作为典型的高密集矢量效果测试，我们对一个小范围（1 K×1 K）的地形高程测试数据进行了等高线绘制，并和如图 2 a 所示的该行业典型 Skyline 软件的等高线绘制效果进行比较. 图 8 b 所示为使用第 2.1 节描述的视点相关未启用级联的绘制方法，可以发现视点远处的等高线绘制变粗，走样较为明显，类似于图 2，已经很难完成 GIS 中要求的精准高程辨别要求. 图 8 a 所示为使用第 2.2 节描述的视点相关级联绘制算法，使用了 4 层级联绘制整个视域，图中红色、绿色、蓝色区域分别代表了前 3 层级联，第 4 层级联覆盖的深度范围在当前视点被遮挡不可见. 可明显发现，远处的等高线走样现象得到了良好控制，本文算法更适用于GIS的客观需求.



a 级联绘制等高线　　b 未使用级联

图8 等高线绘制对比

除去测试地形外，本文还使用了深圳地区的16K×16K的高程图和同比例影像数据图构建了可适应的地形绘制系统，配合5层影像金字塔，在这个基础上对无缝拼接的矢量信息绘制质量进行对比. 如图9a所示，上端大图为空中视点俯视图，其上绘制的矢量包括标准的线、面和文字，使用未加矢量数据空间分析的4层级联绘制；图9b所示为非严格意义的视点相关的动态绘制算法结果，即当视点进入预划分好的区域后用该区域中心高点对该区域矢量数据绘制到纹理空间. 图9b所示为第2.1节描述的视点相关的绘制算法，3幅图中红色框标注区域为走样情况的对比区域. 不难发现，视点相关的矢量绘制比区域相关的动态绘制在下采样所导致的走样上有了减轻，而级联相关的矢量绘制能表现出更强的细节表现力. 这3种方法在图示中均采用1024×1024的纹理，按图9b→9c→9a顺序绘制代价递增，反走样效果递增.



b 区域相关动态矢量绘制

a 大范围数据级联绘制字、线、面　　c 视点相关矢量绘制

图9 大范围矢量绘制反走样对比

观察图9(右下)发现，实际应用中，在每层的级联绘制纹理里若矢量数据覆盖率低，则往往大量区域被浪费，实际有效矢量占面积率不高，容易产生下采样. 第2.3节中描述的基于矢量空间分布的分析处理策略，在每层级联绘制中可以快速剔除掉浪费的、昂贵绘制纹理的像素值. 如图10所示，在经过矢量空间分析后，矢量绘制范围剖分更加完善，像素利用率大大提高，更进一步降低了下采样的可能性. 上部左右图为场景反走样效果对比，下部两端为4层级联绘制矢量优化前后内容对比，下部中间为图中红框内局部细节的放大对比图，其中反走样效果提升明显.

由于本文算法是基于纹理的，所以在不同分辨率的地形快速变化下并不需要重构任何矢量信息，依然可以完成上文表述的反走样效果. 由于现代显卡具有高速的面片绘制能力，基于纹理方案的矢量信息并不需要根据地形的分辨率来新增顶点，绘制的矢量几何顶点数相对于地形数据本身几何少很多，

所以在时间消耗上对实时影响不大. 同时我们可以根据视点距离对矢量数据进行宏观逻辑上的分类，在加速绘制时，加速对当前绘制的矢量信息在每个子视域四棱锥内部进行基于包围盒的四叉树剔除. 通过对不同分辨率下的大量矢量信息进行实验，在硬件平台为 CPU i7 和 NVIDIA GTX480 图形卡环境下，图 11 a 所示在当前地形分辨率下绘制矢量多边形 1 个、矢量文字 9 个、矢量线 653 条，使用 4 级级联，速度 38.2 帧/s；图 11 b 所示在更精细分辨率下绘制矢量文字 3 个、矢量线 31 条，使用 4 级级联，速度 39.5 帧/s，其结果满足实时要求.

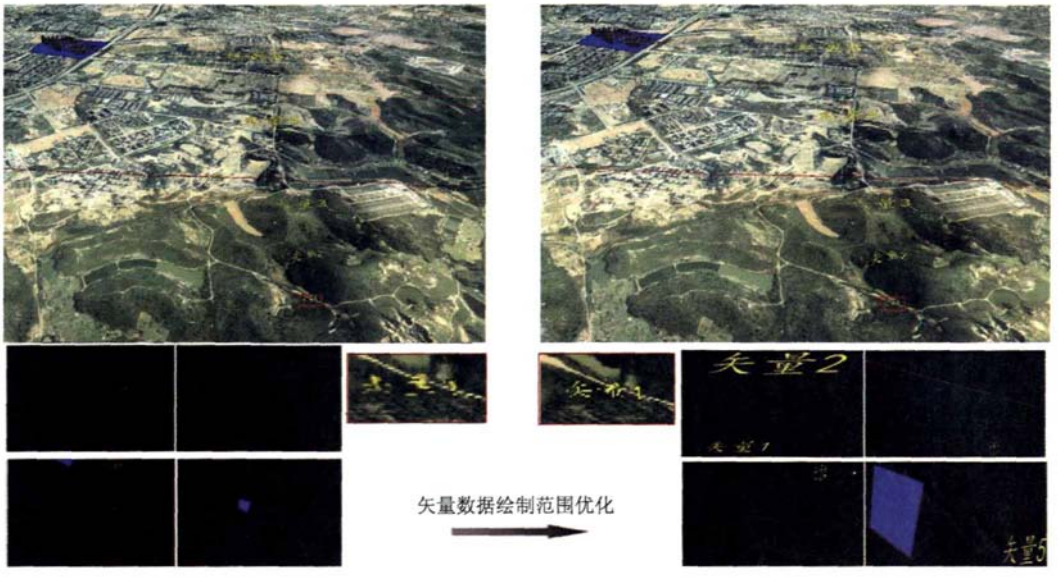
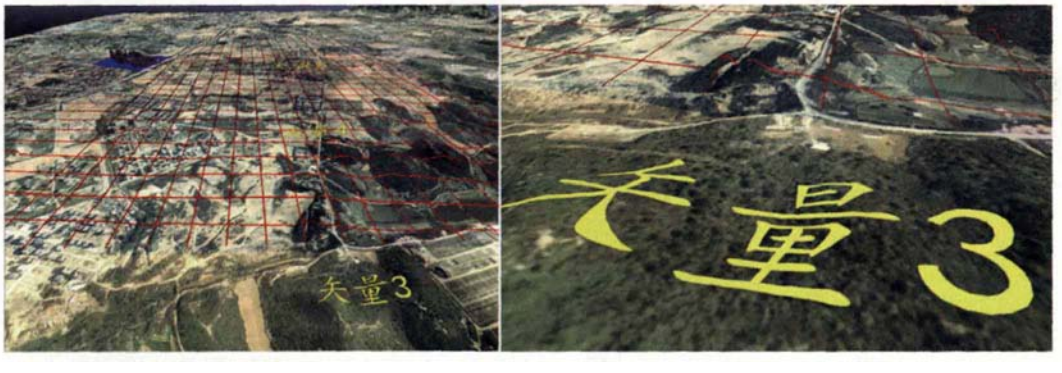


图 10　矢量绘制范围剖分优化策略



a 低细节地形大量矢量　　b 高细节地形少量矢量

图 11　不同地形分辨率大量矢量数据绘制效果

4　结束语

本文改进了基于纹理的矢量数据绘制方法，根据视点位置和视线方向提供深度剖分和屏幕空间分块 2 种策略，进而提出级联绘制矢量的详细算法过程，可以将矢量数据实时绘制到地形纹理上. 分析和实验结果表明，本文算法能够提高矢量纹理的像素利用率，在一定程度上抑制走样的发生. 在实践中，该算法能与目前大多数的多分辨率地形绘制算法以及遥感影像金字塔很好地结合在一起使用，矢量在地形上完成了无缝叠加，反走样效果提高明显，绘制效果增强显著，为其他 GIS 的场景分析功能（如等高线绘制，道路中间线绘制，小区光照分析，区域水淹模型分析）提供了优良的分析可视化反馈效果.

参考文献(References)：

[1] Gong Jianya, Xia Zongguo. An integrated data model in three dimensional GIS [J]. Journal of Wuhan Technical University of Surveying and Mapping, 1997, 22(1): 7-15 (in Chinese)

(龚健雅，夏宗国. 矢量与栅格集成的三维数据模型[J]. 武汉测绘科技大学学报，1997，22(1)：7-15)

[2] Zhu Qing. A survey of three dimensional GIS Technologies [J]. Geomatics World, 2004, 2(3): 8-12 (in Chinese)
(朱 庆. 3 维地理信息系统技术综述[J]. 地理信息世界，2004，2(3)：8-12)

[3] Duchaineau M, Wolinsky M, Sigeti D E, *et al*. ROAMing terrain: real-time optimally adapting meshes [C] // Proceedings of the 8th Conference on Visualization. Los Alamitos: IEEE Computer Society Press, 1997: 81-88

[4] Rui X P, Zhang Y M. Overlaying vector data on 3D terrain [C] //Proceedings of IEEE International Geoscience and Remote Sensing Symposium. Los Alamitos: IEEE Computer Society Press, 2004: 4560-4563

[5] Wartell Z, Kang E, Wasilewski T, *et al*. Rendering vector data over global, multi-resolution 3D terrain [C] // Proceedings of the Symposium on Data Visualization. Aire-la-Ville: Eurographics Association Press, 2003: 213-222

[6] Zou Wan, Fang Jinyun, Liu Jingang. Research about visualization of hybrid multi-resolution terrain and vector data [J]. Journal of System Simulation, 2006, 18(Suppl1): 324-325 (in Chinese)
(邹 烷，方金云，刘金刚. 混合多分辨率地形与空间矢量数据的可视化研究[J]. 系统仿真学报，2006，18(增刊 1)：324-325)

[7] Schneider M, Klein R. Efficient and accurate rendering of vector data on virtual landscapes [J]. Journal of WSCG, 2007, 15(1/3): 59-65

[8] Kersting O, Döllner J. Interactive 3D visualization of vector data in GIS [C] //Proceedings of the 10th ACM International Symposium on Advances in Geographic Information Systems. New York: ACM Press, 2002: 107-112

[9] Schneider M, Guthe M, Klein R. Real-time rendering of complex vector data on 3D terrain models [C] //Proceedings of the 11th International Conference on Virtual Systems and Multimedia. Budapest: Archaeolingua, 2005: 573-582

[10] Wimmer M, Scherzer D, Purgathofer W. Light space perspective shadow maps [C] //Proceedings of the Eurographics Symposium on Rendering/Eurographics Workshop on Rendering Techniques. Aire-la-Ville: Eurographics Association Press, 2004: 143-152

[11] Chen Hong, Tang Xiaoan, Xie Yaohua, *et al*. Rendering vector data over 3D terrain with view-dependent perspective texture-mapping[J]. Journal of Computer-Aided Design & Computer Graphics, 2010, 22(5): 754-761 (in Chinese)
(陈 鸿，汤晓安，谢耀华，等，基于视点相关透视纹理的矢量数据在三维地形上的叠加绘制[J]. 计算机辅助设计与图形学学报，2010，22(5)：754-761)