

基于立方体全景图的虚拟场景 浏览技术研究及实现

韦 群^{1,3)} 高 丽²⁾ 龚雪晶³⁾

¹⁾(北京航空航天大学计算机科学系,北京 100083) ²⁾(总装备部情报所,北京 100063)

³⁾(装备指挥技术学院电子工程系,北京 101416)

摘 要 虚拟现实技术将计算机采集到的各类信息以全新的方式组织起来,提供给用户一个趋于人性化的虚拟信息空间,如今通常以全景图象的方式来表示的虚拟场景主要有球面全景图、柱面全景图和立方体全景图 3 种模式. 为了给用户浏览虚拟场景提供一种方便、自然的交互方式,提出了一种对立方体全景图进行实时交互浏览的基本重投影算法及其加速算法,并实现了一个立方体全景图浏览器,实验结果表明,该方法具有最大的观察自由度.

关键词 计算机图形学(520·6020) 虚拟现实 全景图 立方体全景图 浏览

中图法分类号: TP391.9 **文献标识码:** A **文章编号:** 1006-8961(2003)09-1061-06

The Technique of Browsing Virtual World Based-on Cubic Panorama

WEI Qun^{1,3)}, GAO Li²⁾, GONG Xue-jing³⁾

¹⁾(Department of Computer Science, Beijing University of Aeronautics and Astronautics, Beijing 100083)

²⁾(The Intelligence institute of general armament Department, Beijing 100063)

³⁾(The Electronic Department of the Armament College of Command and Technology, Beijing 101416)

Abstract Virtual reality technique organizes all kinds of information with a new mode in computer, and give a virtual information space of human nature. Usually, the virtual scene is denoted by panoramas, it has three kinds: spherical panorama, cylinder panorama and cubic panorama. Firstly this paper discusses the environment map briefly. From our point of view, the theory of environment map is a footstone to building virtual environment. Then it presents the basic and speedup arithmetic of browsing the cubic panorama, and introduces an anti-aliased arithmetic of the environment image to eliminate the distortion of the image. At last, the paper implements a cubic panorama browser. The browser has functions as follow: roaming in any direction in the virtual environment and the display speed is in direct ratio to the speed of the mouse; cruising the virtual environment with zooming in or zooming out. The browser also provides anti-aliased functions in following situation: real-time anti-aliased function, actionless anti-aliased function and no anti-aliased function. The browser can make user get the biggest freedom and give a kind of convenient and natural interactive mode.

Keywords Virual reality, Panorama, Cubic panorama, Browse

0 引 言

在传统的计算机图形学中,环境映照(Environment maps)技术^[1]是采用纹理图象来表示景物表面的镜面反射和规则透射效果.它是以景物中心为固定视点来观察整个场景,并将周围场景的图象记录在以该点为中心的环境映照球面或立方体

表面上,这种环境映照实际上是以全景图象的方式来提供其中心视点处的场景描述.在虚拟空间中,称给定视点的环境映照为全景图(Panorama).

全景图通常是全视角图象,在它所对应的视点处,通过透视投影可以得到任意观察方向的视平面图象,即它可以获得给定视点处的最大观察自由度.全景图的表示模式主要有球面全景图、立方体全景图和柱面全景图 3 种,而且不同全景图模式的获取

和控制的难易有很大的不同. 显然, 中心位于视点处的球面全景图是描述一个场景的理想选择, 不过, 球面全景图是一个非均匀采样表示, 其会导致场景扭曲变形, 在两极尤为严重, 且球面投影缺乏一种适合于计算机存储的表示方式; 采用柱面全景图来表示场景, 其优点是柱面全景图容易获得, 且可方便地生成任意视线方向的场景, 但在实际应用中, 这种无底面的有限柱面将对垂直方向的视域有一定的限制; 立方体全景图是由6幅广角为 90° 的画面组成, 它克服了球面全景图和柱面全景图的一些缺陷, 这种方式不仅易于全景图象数据的存储, 而且与屏幕像素对应的重采样区域边界为多边形, 非常便于显示, 另外, 这种立方体映射方式还可以实现水平和垂直方向的 360° 旋转, 即可以实现任意角度的旋转, 以便进行最大自由度的观察.

虚拟空间^[2]的漫游过程实际上就是基于全景图的视景绘制过程, 这种全景图是采集到的场景视觉信息集合, 也是场景的一种表示模式. 视景绘制就是依据视点和视角方向对全景图进行重采样, 通常是通过将全景图的可见部分重投影到视平面上来实现.

获得立方体全景图的方式通常有如下3种: 一是使用三维软件来生成无缝拼接于一个立方体的6个面上的贴图; 二是使用特殊的拍摄装置, 依次在水平、垂直方向每隔 90° 拍摄一张照片, 获得6张可以无缝拼接于一个立方体的6个面上的照片; 三是使用图形软件来生成. 本文采用第3种方法来生成立方体全景图.

1 立方体重投影基本算法

为实现对立方体全景图进行交互式全方位的浏览, 必须根据用户的相应动作实时地将全景图的可见部分重投影到视平面上. 图1是视平面与立方体全景图的几何关系, 其中的粗线包围部分即是可见部分.

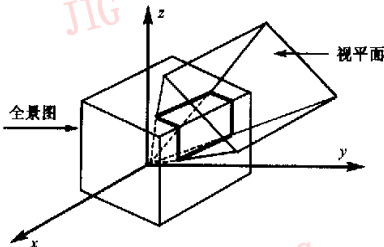


图1 视平面与全景图

1.1 视线与立方体求交

设立方体中心(即视点)为观察坐标系原点, 且视点到视平面的距离固定为 d . 为方便描述, 定义如下一些变量: a 为立方体的边长; w 为视平面宽; h 为视平面高; θ 为平转角, 又叫方位角 ($-\pi \leq \theta \leq \pi$); φ 为俯仰角 ($-\frac{\pi}{2} \leq \varphi \leq \frac{\pi}{2}$). 立方体6个面和各个顶点如图2所示, 6个面的方程分别为

$$\begin{aligned} \text{面1: } & \{x = \frac{a}{2}, (-\frac{a}{2} \leq y, z \leq \frac{a}{2})\} \\ \text{面2: } & \{x = -\frac{a}{2}, (-\frac{a}{2} \leq y, z \leq \frac{a}{2})\} \\ \text{面3: } & \{y = -\frac{a}{2}, (-\frac{a}{2} \leq x, z \leq \frac{a}{2})\} \\ \text{面4: } & \{y = \frac{a}{2}, (-\frac{a}{2} \leq x, z \leq \frac{a}{2})\} \\ \text{面5: } & \{z = \frac{a}{2}, (-\frac{a}{2} \leq x, y \leq \frac{a}{2})\} \\ \text{面6: } & \{z = -\frac{a}{2}, (-\frac{a}{2} \leq x, y \leq \frac{a}{2})\} \end{aligned} \quad (1)$$

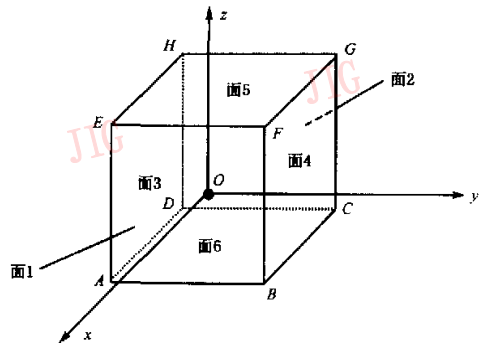


图2 立方体6个面和各顶点

因视平面任一点 (x_0, y_0, z_0) 与视点 $(0, 0, 0)$ 连线方程为

$$\frac{x}{x_0} = \frac{y}{y_0} = \frac{z}{z_0} \quad (2)$$

故只要能够判断出该线与立方体的相交面, 则能够很容易地得到视平面上的点与立方体各面上的对应点 (u_0, v_0, w_0) , 立方体各面上对应点的坐标分别表示如下:

$$\begin{aligned} \text{面1: } & (\frac{a}{2}, \frac{ay_0}{2x_0}, \frac{az_0}{2x_0}) \\ \text{面2: } & (-\frac{a}{2}, -\frac{ay_0}{2x_0}, \frac{az_0}{2x_0}) \\ \text{面3: } & (-\frac{ax_0}{2y_0}, -\frac{a}{2}, -\frac{az_0}{ay_0}) \end{aligned}$$

$$\begin{aligned} \text{面 4: } & \left(\frac{ax_0}{2y_0}, \frac{a}{2}, \frac{az_0}{2y_0} \right) \\ \text{面 5: } & \left(\frac{ax_0}{2z_0}, \frac{ay_0}{2z_0}, \frac{a}{2} \right) \\ \text{面 6: } & \left(-\frac{ax_0}{2z_0}, -\frac{ay_0}{2z_0}, -\frac{a}{2} \right) \end{aligned} \quad (3)$$

1.2 判断视线与立方体的相交面

因立方体的长、宽、高相等,且坐标原点O(图 2)与立方体每个顶点的射线上任一点的 3 观察坐标值相等,所以该射线与三坐标轴的夹角均为 $\pi/4$ 。而任一点 (x, y, z) ,其 3 个坐标中哪个绝对值最大,则该点和原点连线就与哪一个轴的夹角小于 $\pi/4$,即射线一定会与该轴穿过的面相交,简单地说,视平面上某一点的观察坐标值中绝对值大的轴穿过的面就是视线与立方体相交的面;当 3 个坐标绝对值相等时,则与顶点相交;当两个坐标绝对值相等,且大于另一个坐标时,则与边相交;再根据坐标值的正负,即可判断出视线与立方体的相交处。

例如,当 $|x| = \max(|x|, |y|, |z|)$ 且 $x < 0$ 时,视线必与面 2 相交;当 $|x| = |y| = |z|$ 且 $x > 0, y > 0, z > 0$ 时,交点为顶点 F,依此类推。这样,就得到了视平面上的任一点与立方体各面上相应点的对应关系。

1.3 基本算法

为了能够交互式地浏览全景图,必须根据用户的动作来实时计算出视平面上任一点的观察坐标,以便根据以上算法求得全景图上所对应的可见部分,并显示出来。

用户的交互式浏览,是对俯仰角 φ 和方位角 θ 的改变,反映到坐标系中就是绕水平轴 y 和垂直轴 z 的旋转,三维绕 y, z 轴旋转方程可用齐次坐标形式表示如下:

$$\begin{aligned} \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} &= \begin{bmatrix} \cos\varphi & 0 & \sin\varphi & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\varphi & 0 & \cos\varphi & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \\ \begin{bmatrix} x'' \\ y'' \\ z'' \\ 1 \end{bmatrix} &= \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} \end{aligned} \quad (4)$$

由式(4)可得任意时刻视平面上任一点的观察坐标为

$$\begin{aligned} x'' &= x \cos\varphi \cos\theta - y \sin\theta + z \sin\varphi \cos\theta \\ y'' &= x \cos\varphi \sin\theta + y \cos\theta + z \sin\varphi \sin\theta \\ z'' &= z \cos\varphi - x \sin\varphi \end{aligned} \quad (5)$$

根据式(3)可得到对应的立方体各面上的对应点。

1.4 设备坐标系与观察坐标系之间的变换

在上述算法中,首先要将视平面中的点由设备坐标 (i, j) 变换到观察坐标 (x, y, z) ,如图 3 所示, u, v 为设备坐标系, x, y, z 为观察坐标系。

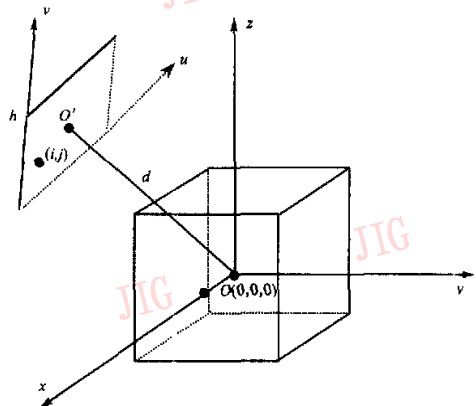


图 3 某时刻坐标系之间的位置关系

为了便于推导运算,假定视平面绕视点旋转运动,视线(视点与视平面中心连线 OO')始终垂直于视平面。设视平面中心初始观察坐标为 $(d, 0, 0)$,且视平面平行于立方体面 1,则视点、面 1 中心、视平面中心 3 点共线,因此,初始位置视平面方程为: $x=d$ 。

在设备坐标系下,对于视平面上初始位置任一点 (i, j) ,根据中心点的初始观察坐标,可得到其观察坐标为 $(d, j - \frac{w}{2}, i - \frac{h}{2})$,随着观察俯仰角 φ 和方位角 θ 的改变,由 1.3 节算法即可实时计算出视平面上任一点的观察坐标,再由 1.2 节和 1.1 节算法求得立方体各面上对应点的观察坐标。

然后,根据立方体各面上各对应点的位置,先取得其像素值,再将其重投影到视平面上,并显示出来,即实现了交互实时显示功能。

2 立方体重投影的加速算法

加速算法从设备坐标系与观察坐标系之间的变换以及判断视线与立方体的相交面两个方面改进浏览速度。

2.1 设备坐标系与观察坐标系之间的变换

即视平面中的任一点由设备坐标 (i, j) 变换到

观察坐标 (x, y, z) 时, 加速算法是将视平面的一条扫描线作为变换单位, 一条扫描线 i (起止点分别为 $(i, 0), (i, w)$) 经过设备坐标系与观察坐标系之间的变换后仍为一条直线, 其参数方程为

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} + \begin{bmatrix} m \\ n \\ p \end{bmatrix} \cdot t \quad (6)$$

其中, 参数 t 的取值范围为 $0 \sim w$; 观察坐标系下直

线起始点 $\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}$ 的坐标取值为 $\begin{bmatrix} d \\ -\frac{w}{2} \\ i - \frac{h}{2} \end{bmatrix}$; $\begin{bmatrix} m \\ n \\ p \end{bmatrix}$ 为直线

的方向向量取值为 $\begin{bmatrix} d \\ \frac{w}{2} \\ i - \frac{h}{2} \end{bmatrix} - \begin{bmatrix} d \\ -\frac{w}{2} \\ i - \frac{h}{2} \end{bmatrix} = \begin{bmatrix} 0 \\ w \\ 0 \end{bmatrix}$.

随着观察俯仰角 φ 和方位角 θ 的改变, 由式(5) 可得到旋转后的直线参数方程, 形式与式(6) 相同, 其中直线起点坐标为

$$\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} = \begin{bmatrix} d \cos \varphi \cos \theta + \frac{w}{2} \sin \theta + \left(i - \frac{h}{2}\right) \sin \varphi \cos \theta \\ d \cos \varphi \sin \theta - \frac{w}{2} \cos \theta + \left(i - \frac{h}{2}\right) \sin \varphi \sin \theta \\ \left(i - \frac{h}{2}\right) \cos \varphi - d \sin \varphi \end{bmatrix}$$

方向向量为

$$\begin{bmatrix} m \\ n \\ p \end{bmatrix} = \begin{bmatrix} -w \sin \theta \\ w \cos \theta \\ 0 \end{bmatrix}$$

扫描线上的每一点坐标变换应先由变换后的直线方程求出, 以避免对视平面每一点进行坐标变换, 这样就提高了算法的执行速度。

2.2 判断视线与立方体的相交面

由于每条扫描线经坐标变换后投射到立方体上可分为几段, 因此每一线段同属一个立方体面(如图 4 所示)。

由于扫描线 i 投射到立方体上分为两段, 并分别属于立方体的面 1 和面 3, 因而在判断相交面时, 可将每条扫描线按照其投射到不同的立方体面划分为几段, 如图 4 中的扫描线 a 段和 b 段。这样每段扫描线中的点就与立方体相交于相同的面, 这样就可避免对每条视线进行判断, 即 a 段中的点与立方体的相交面都为面 1, b 段中的点与立方体的相交面为

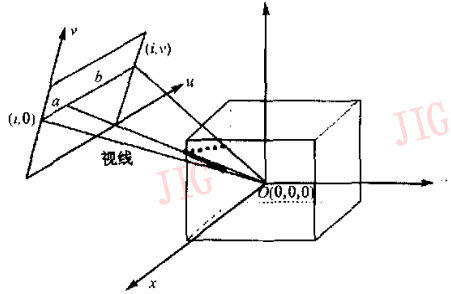


图 4 扫描线投射到立方体的分段示意图

面 3。扫描线段的划分算法可使用折半查找法来提高执行效率。

3 变焦算法

连续变焦是指相机焦距和视野(Field of View)的改变, 主要用于对场景进行不同尺度的观察, 其可分为拉长(Zoom in)和缩短(Zoom out)两种改变。在透视变换中, 焦距对应于视点 to 视平面中心的距离 d 。当 d 变化时, 重复上述重投影算法, 然后刷新显示; 当视平面大小不变时, 若 d 拉长, 则视野缩小, 图象放大; 若 d 缩短, 则视野放大, 图象缩小。

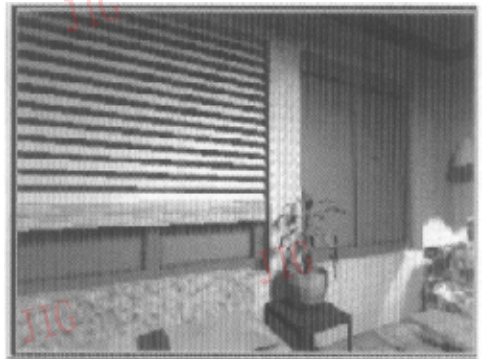


图 5 绘制场景时出现的走样现象

4 反走样(Anti-aliasing)算法

由于计算机生成的图形是由离散像素组成的数字化图象, 因而屏幕上的图形常常与真实景物之间存在误差。这种误差表现为: 直线或光滑的曲线在屏幕上呈现锯齿状、彩色花纹失去原有的形态和色彩、细小的物体在屏幕上丢失等等, 在动态的画面中, 则表现为部分像素的闪烁现象。如图 5 所示, 图中的百

叶窗有非常明显的变形现象,从而影响了视觉效果.反走样,也称抗混叠,它是一种提高图象质量的绘制技术,常用来减少或消除图象的变形走样现象.反走样的方法^[3]有很多种,例如提高分辨率,区域采样等.本文采用将源图象中相邻像素按不同权值进行累加来得到显示像素值的方法.

在对全景图进行浏览的过程中,通过透视变换得到的采样像素坐标值,由于是离散化成整数像素坐标的,且只取了其中一个像素,而忽略了相邻像素对它的贡献,这就造成了因图象的不连续而出现走样现象.

要解决这个问题,就要考虑相邻像素的影响.如果将每个像素看作是一个单位正方形区域,而将相邻像素对采样像素的影响系数作为权值来对相邻像素进行加权累加,那么即可得到采样像素值.

如图 6 所示,图中 O 点为采样像素中心,因为其坐标不是整数化的,必然会落在 $A、B、C、D$ 这 4 个像素间的某一位置.显然,越靠近像素 O ,相交的面积越大,权值越大;相反,权值越小.例如 B 距离 O 最远,其相交面积(阴影块)最小,权值最小.由于根据像素坐标之间的距离可以得到各个权值的大小,且权值之和正好为单位正方形的面积,从而可以得到采样像素的像素值

$$\bar{x} = \sum_{i=0}^3 x_i w_i$$

其中, x_i 为相邻像素的像素值; w_i 为相邻像素的权值; \bar{x} 为采样像素的像素值.

这样,就比较精确地获得了采样像素值.图 7 是将图 5 结果进行反走样后的效果图,由图 7 可见,已明显地消除了锯齿等变形现象,图象清晰自然.

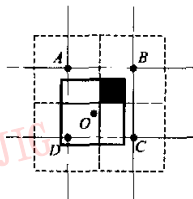


图 6 采样像素值的获取

考虑到交互性和实时绘制视图画面的要求及投影变换算法的效率,应当依据用户的交互状态选用不同时机进行反走样处理:当用户对场景的整体布局进行浏览时,则要求动态生成满足实时交互约束的视图画面,由于这时用户并不关心细节信息,因而

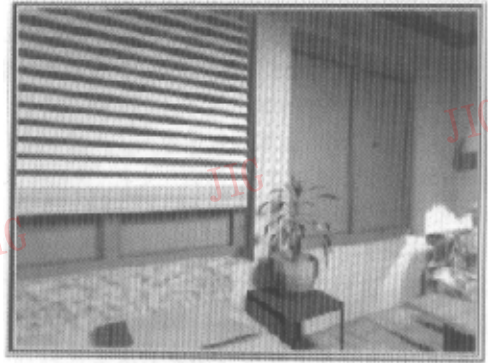


图 7 反走样后的场景图

可以不加处理;当交互的视角变化缓慢或不变时,则细节信息就变得醒目,这时就要进行反走样处理,以得到低帧数或静止帧的反走样图象.

5 实验结果

图 8 是在 VC6.0 环境下采用上述基本的和加速的立方体重投影算法实现的立方体全景图浏览器.在浏览立方体全景图时,由实验结果可见,两种算法显示的图象质量完全一致,但加速算法在浏览速度上则有明显提高,其提高程度与浏览时的转动方向相关.经测试,在相同的运行环境下,当俯仰角度一定时,水平环视一周的平均速度提高 50%.

如图 8 所示,立方体浏览器的主要功能包括:

(1)当鼠标位于显示窗口区时,按下鼠标左键,随着鼠标的移动,可以完成左右环视、俯视、仰视、斜视等任意方向的操作,速度与相对于鼠标初始位置的距离成正比.

(2)按下显示窗口右边的方向按钮,可以以一定的角速度完成左右水平环视、垂直俯视、垂直仰视等



图 8 一个立方体全景图浏览图

操作.

(3)变焦距观察.按下 ZOOM₊按钮可进行拉长焦距观察,按下 ZOOM₋按钮可进行收短焦距观察, RESIZE 按钮可恢复初始焦距.

(4)根据用户对图象显示质量的不同要求,可以选择不同的反走样处理,如实时反走样、静止图象反走样和不进行反走样.

实验结果表明,该浏览器可以随着用户的动作实时地显示出相应的视景画面,使用户可以进行任意方向观察,即可获得给定视点处的最大观察自由度,并可进行变焦距观察和图象的反走样处理.另外,它还能够提供方便、自然、符合人的心理特点的交互方式.

6 结 语

建立虚拟空间的最终目的是提供给用户一个能够沉浸于其中,进行自由交互观察与操纵的虚拟信息环境^[4],因此虚拟空间系统必须紧随用户的行为实时生成场景画面.

本文提出并实现了一种对立方体全景图进行重投影的基本算法,在此基础上又研制了加速算法,从而明显地提高了立方体全景图浏览器的浏览速度,并对图象进行了反走样处理,使得场景画面清晰、自然.该方法简单、实用,已应用于中国建筑科学研究院建筑设计系统中,即为该系统提供了一个强大、方便的软件工具.

参 考 文 献

- 1 彭群生,鲍虎军,金小刚.计算机真实感图形的算法基础[M].北京:科学出版社,1999:221~245.
- 2 李洪举,吴恩华.一种用于室内虚拟环境的新表示[J].软件学报,1998,9(9):641~645.
- 3 Weiman, Carl F.R. Continuous anti-aliased rotation and zoom of raster image[J]. Computer Graphics. 1980,14(3):286~293.
- 4 Mauro Figueiredo, Klaus Bohm, Jose Teixeira. Advanced interaction techniques in virtual environments[J]. Computer & Graphics. 1993,17(6):655~661.



韦 群 1964 生,1991 年获国防科技大学信息工程专业硕士学位,现为北京航空航天大学在职博士研究生,教授.主要研究领域为基于图象的虚拟现实技术、网络信息系统及其开发方法等.发表论文 20 余篇.

高 丽 1975 年生,2001 年获装备指挥技术学院管理信息系统专业硕士学位.主要研究领域为基于图象的虚拟现实技术.



龚雪晶 1970 年生,讲师,2001 年获装备指挥技术学院科学技术可视化专业硕士学位.主要研究领域为科学计算可视化.