

8-14-2020

## Research of Display Accuracy Improvement in Multi-scale 3D Scene

Pengfei Li

*Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001;*

Zhou Yang

*Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001;*

Jiwei Xu

*Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001;*

Xiaofei Hu

*Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001;*

*See next page for additional authors*

Follow this and additional works at: <https://dc-china-simulation.researchcommons.org/journal>



Part of the Artificial Intelligence and Robotics Commons, Computer Engineering Commons, Numerical Analysis and Scientific Computing Commons, Operations Research, Systems Engineering and Industrial Engineering Commons, and the Systems Science Commons

---

This Paper is brought to you for free and open access by Journal of System Simulation. It has been accepted for inclusion in Journal of System Simulation by an authorized editor of Journal of System Simulation.

---

## Research of Display Accuracy Improvement in Multi-scale 3D Scene

### Abstract

**Abstract:** In the multi-scale 3D scene, the single reference coordinate system has much application limitation due to the large spatial scale change, so the target will appear wobble phenomenon when the camera observes the target in a large distance from the reference coordinate. Due to the fact that the position and attitude of a spatial object change all the time under the unified time datum, the target will appear wobble phenomenon when the camera following a target observes a certain area near the target. *A setting method of reference coordinate system based on the observation of the inertial coordinate system was proposed, which could effectively improve the display accuracy when the space scale changed. What's more, a setting method of reference coordinate system based on the observation of the body-fixed coordinate system was proposed, which could effectively improve the display accuracy when the camera observed the target in a short distance.*

### Keywords

multi-scale three-dimensional scene, floating error, observed target, display accuracy

### Authors

Pengfei Li, Zhou Yang, Jiwei Xu, Xiaofei Hu, and Xianguang Xue

### Recommended Citation

Li Pengfei, Zhou Yang, Xu Jiwei, Hu Xiaofei, Xue Xianguang. Research of Display Accuracy Improvement in Multi-scale 3D Scene[J]. Journal of System Simulation, 2016, 28(9): 2275-2282.

## 跨尺度三维场景显示精度提高技术研究

李鹏飞, 周杨, 许继伟, 胡校飞, 薛现光

(信息工程大学地理空间信息学院, 郑州 450001)

**摘要:** 在跨尺度三维场景中, 由于空间尺度变化非常大, 单一的基准坐标系存在很大的应用局限性, 当相机观察距离基准坐标系远的目标时, 目标会出现抖动现象。空间目标在统一时间基准下位置和姿态不断发生变化, 当相机跟随目标近距离固定观察目标的某一区域时, 目标也会出现抖动。提出了基于关注目标惯性坐标系设置基准坐标系的方法, 在空间尺度变化时可以有效提高显示精度。提出了基于关注目标本体坐标系设置基准坐标系的方法, 当相机近距离固定观察目标时也可以有效提高显示精度。

**关键词:** 跨尺度三维场景; 浮点误差; 关注目标; 显示精度

中图分类号: TP391.41 文献标识码: A 文章编号: 1004-731X (2016) 09-2275-08

## Research of Display Accuracy Improvement in Multi-scale 3D Scene

Li Pengfei, Zhou Yang, Xu Jiwei, Hu Xiaofei, Xue Xianguang

(Institute of Geographical Spatial Information, Information Engineering University, Zhengzhou 450001)

**Abstract:** In the multi-scale 3D scene, the single reference coordinate system has much application limitation due to the large spatial scale change, so the target will appear wobble phenomenon when the camera observes the target in a large distance from the reference coordinate. Due to the fact that the position and attitude of a spatial object change all the time under the unified time datum, the target will appear wobble phenomenon when the camera following a target observes a certain area near the target. A setting method of reference coordinate system based on the observation of the inertial coordinate system was proposed, which could effectively improve the display accuracy when the space scale changed. What's more, a setting method of reference coordinate system based on the observation of the body-fixed coordinate system was proposed, which could effectively improve the display accuracy when the camera observed the target in a short distance.

**Keywords:** multi-scale three-dimensional scene; floating error; observed target; display accuracy

## 引言

随着航空航天技术的发展, 人类的活动已经不再满足于地面, 太空已成为各国利益争夺战的新据点。为维护空间利益、保证空间安全, 航天大国纷

纷建立了自己的空间态势感知系统<sup>[1]</sup>。由于跨尺度的三维场景仿真系统大到可以容纳整个太阳系, 小到可以展现米级的高分辨率地形, 所以在未来的军事强国对抗中, 建立一个基于统一时空基准建模并且还能对空间目标和空间环境进行展示分析的跨尺度三维虚拟仿真系统对于发展空间态势感知能力、快速获取战场信息和及时掌握作战主动权有着重要的意义<sup>[2]</sup>。现代主流的图形处理器 GPU 普遍对单精度浮点数的处理效率很高, 所以大型的仿真



收稿日期: 2015-06-14 修回日期: 2015-11-09;  
作者简介: 李鹏飞(1991-), 男, 河南, 硕士, 研究方向为三维仿真; 周杨(1974-), 男, 贵州, 博士, 教授, 硕导, 研究方向为数字摄影测量、三维地形可视化; 许继伟(1989-), 男, 安徽, 硕士, 研究方向为数字摄影测量。

<http://www.china-simulation.com>

• 2275 •

系统中的数据在进入 GPU 处理之前需要被转换为单精度浮点数。由于单精度浮点数舍入误差的限制,显示精度不足是类似星球仿真、海量三维地形仿真、物理计算仿真等大型虚拟仿真系统普遍存在的问题。单浮点数的舍入误差随着浮点数值的增长会逐渐增大<sup>[3]</sup>。也就是说,单浮点数越小数值精度越高,单浮点数越大数值精度越低。在跨尺度的三维场景中,最远的海王星与太阳的距离是  $4.5 \times 10^9$  km,单浮点数舍入误差达到 512 km。当场景经过连续的矩阵变换后最终渲染到屏幕上时,会出现三角网格破碎、裁剪不正确的现象,影响了可视化效果<sup>[4]</sup>。为了减小舍入误差、提高数值精度,目前普遍采用以局部坐标系代替全局坐标系作为场景基准坐标系的方法。现有的方法多是简单地将局部坐标系的原点定义在视点<sup>[2]</sup>或者观察目标上<sup>[5-6]</sup>形成单一的基准坐标系来解决浮点数过大的问题。由于在空间尺度上变化非常大,单一的基准坐标系在跨尺度三维场景中有很大的应用局限性。跨尺度三维场景不仅具有空间尺度,而且也具有时间尺度<sup>[7]</sup>。在统一的时间基准下,空间目标按照一定的规律随时间不断地变换空间位置和空间姿态。当相机跟随目标固定观察目标的某个区域时,相机需要实时更新位置和姿态。现有的方法基本上没有考虑时间对显示精度的影响。

本文采用基于相机关注目标设置基准坐标系的方法提高跨尺度三维场景的显示精度,根据不同的关注目标选择不同的基准坐标系原点,大幅度减小空间尺度带来的显示误差。当实现相机跟随目标固定观察时,本文还通过设置基准坐标系轴向与关注目标本体系轴向相同的方法消除了由于时间变化所带来的显示误差。

## 1 显示误差成因

主流的图形硬件 GPU 的单精度浮点运算能力是双精度的数十倍,所以在跨尺度的三维场景中采用单精度浮点数作为 GPU 着色器的数据源。但是由于硬件的限制,单精度浮点数不可能精确地表示

任意精度的浮点数值,即存在舍入误差。并且浮点数越大,舍入误差就越严重<sup>[3]</sup>。

### 1.1 浮点数舍入误差原理

计算机可以对非常大的数字、非常接近于 0 的数字以及任意的其它实数进行近似的计算。现有的计算机浮点数表示方法采用的是 IEEE-754 标准。IEEE 标准浮点数的表示数值  $V$  的方法为<sup>[8]</sup>:

$$V = (-1)^s \times M \times 2^E \quad (1)$$

式中:  $s$  是符号;  $M$  是尾数;  $E$  是阶码。在 32 位的单精度浮点格式中,  $s$  占 1 位,  $M$  占 23 位,  $E$  占 8 位,如图 1 所示。



图 1 单精度浮点数的位表示

根据阶码的二进制位表示,单精度浮点数被编码为三种情况:规格化值、非规格化值、无穷值。其中非规格化值阶码的二进制位全部为 0,无穷值阶码的二进制位全部为 1,阶码的二进制位既不全为 0 又不全是 1 的为规格化值。经过编码后的所有可能的浮点数值在数轴上的分布如图 2 所示。

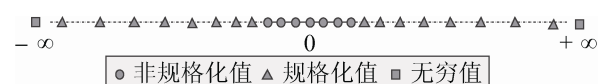


图 2 单精度浮点数在数轴上的分布

其中非规格化值的范围是  $(-2^{-126}, +2^{-126})$ ,在数轴上靠近 0 的位置附近均匀分布。规格化值的范围是  $[-(2-2^{-23}) \times 2^{127}, -2^{-126}] \cup [2^{-126}, (2-2^{-23}) \times 2^{127}]$ ,在数轴上不均匀分布,越是远离 0 的位置规格化值的分布就越稀疏。由于表示方法的限制,浮点数只能近似地进行计算,当浮点数数值不能被计算机精确表示时,就必须向上或者向下进行舍入调整。

非规格化值在数轴上靠近 0 的位置周围均匀分布,因此非规格化值的舍入误差是个常量:

$$\varepsilon = 2^{-129} \approx 1.469\ 37 \times 10^{-39} \quad (2)$$

规格化值在数轴上不均匀分布,规格化值的舍入误差是与数值大小相关的变量:

$$\varepsilon = 2^{E-23} \quad (3)$$

式中,  $E$  为浮点数的阶码。浮点数的舍入误差也会随着运算过程的累积而不断变大。设有浮点数  $x_1, x_2, \dots, x_n$ , 它们有如下关系:

$$z = k_1x_1 + k_2x_2 + \dots + k_nx_n + k_0$$

其中  $k_0, k_1, \dots, k_n$  为常数。设  $\varepsilon_1, \varepsilon_2, \dots, \varepsilon_n$  为浮点数  $x_1, x_2, \dots, x_n$  的舍入误差, 则浮点数舍入误差的传播公式:

$$\varepsilon_z = \sqrt{k_1^2\varepsilon_1^2 + k_2^2\varepsilon_2^2 + \dots + k_n^2\varepsilon_n^2} \quad (4)$$

### 1.2 空间尺度导致的显示误差

在图形硬件 GPU 上将三维目标渲染到二维屏幕窗口上需要经过顶点着色、细分着色、几何着色、图元装配、视口裁切、光栅化、片元着色、片元测试等一系列阶段。顶点着色又包括模型变换、视图变换、投影变换等 3 个阶段, 如图 3 所示。

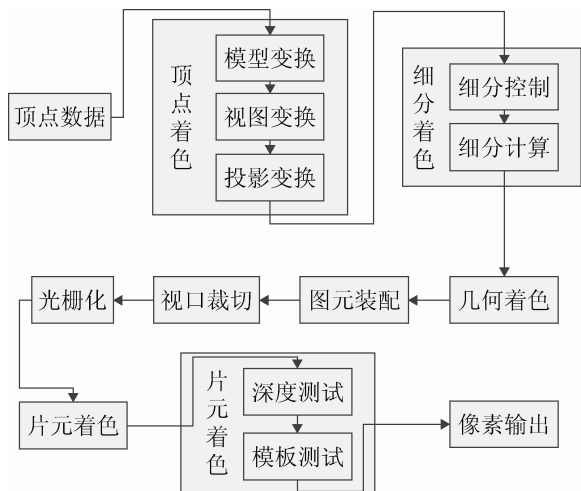


图 3 GPU 顶点渲染管线

其中, 模型变换负责将顶点坐标从局部坐标系转换至基准坐标系, 视图变换负责将顶点坐标从基准坐标系转换至相机坐标系, 投影变换负责将顶点坐标从相机坐标系转换至投影坐标系<sup>[9]</sup>。顶点着色坐标变换过程可由下式描述:

$$\mathbf{V}' = \mathbf{M}_{proj} \times \mathbf{M}_{view} \times \mathbf{M}_{model} \times \mathbf{V} \quad (5)$$

其中,  $\mathbf{M}_{model}$  为模型变换矩阵,  $\mathbf{M}_{view}$  为视图变换矩阵,  $\mathbf{M}_{proj}$  为投影变换矩阵。投影变换对显示误差的影响较小, 可以忽略不计。

考虑模型变换和视图变换对显示误差的影响: 顶点坐标  $\mathbf{V}$  (齐次坐标) 从局部模型坐标系转换至相机坐标系的坐标  $\mathbf{V}'$  (齐次坐标) 的公式有:

$$\mathbf{V}' = \mathbf{M}_{view} \times \mathbf{M}_{model} \times \mathbf{V} \quad (6)$$

$$\mathbf{M}_{model} = \begin{bmatrix} a_1 & a_2 & a_3 & x_{model} \\ b_1 & b_2 & b_3 & y_{model} \\ c_1 & c_2 & c_3 & z_{model} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{M}_{view} = \begin{bmatrix} a'_1 & a'_2 & a'_3 & -x_{view} \\ b'_1 & b'_2 & b'_3 & -y_{view} \\ c'_1 & c'_2 & c'_3 & -z_{view} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$\mathbf{V}_{model} = [x_{model} \ y_{model} \ z_{model}]^T$  为局部坐标系原点在基准坐标系中的位置  $\mathbf{V}_{view} = [x_{view} \ y_{view} \ z_{view}]^T$  为相机中心在基准坐标系中的位置。  $a_1, a_2, \dots, c_2, c_3$  为局部坐标系旋转至基准坐标系的旋转矩阵元素。  $a'_1, a'_2, \dots, c'_2, c'_3$  为基准坐标系旋转至相机坐标系的旋转矩阵元素。局部顶点坐标一般需要经过多次变换才能转换到基准坐标系中, 即  $\mathbf{M}_{model}$  一般是多个矩阵的连乘乘积。由于矩阵的元素也由浮点数表示, 为了避免矩阵连乘造成误差累积, 可以预先在 CPU 中使用双精度浮点数计算矩阵连乘结果, 然后再将计算结果传入 GPU 中对顶点坐标进行转换。由于旋转矩阵中的每个元素都小于 1.0, 舍入误差小于  $10^{-7}$ 。当  $\mathbf{V}_{model}$  和  $\mathbf{V}_{view}$  很大时, 它的舍入误差可忽略不计, 此时旋转矩阵的元素可以看作是常数项。

定义顶点  $[x \ y \ z]^T$  的误差为:

$$\varepsilon_V = \sqrt{\varepsilon_x^2 + \varepsilon_y^2 + \varepsilon_z^2} \quad (7)$$

则由公式(4)和公式(6)可以推出:

$$\varepsilon_{V'} = \sqrt{\varepsilon_V^2 + \varepsilon_{model}^2 + \varepsilon_{view}^2} \quad (8)$$

根据相机空间的顶点误差求出顶点的屏幕误差<sup>[10]</sup>error 的公式有:

$$error = \frac{height}{2 \times z' \times \tan\left(\frac{fov}{2}\right)} \varepsilon_{V'} \quad (9)$$

其中,  $height$  为当前显示窗口的高度,  $fov$  为相机的视场角。  $z'$  为顶点在相机坐标系中的  $z$  值。

相机视场角一般取  $60^\circ$ ，窗口高度的取值与系统的运行环境有关，一般为 900 像素。

假设太阳质心为跨尺度三维场景的基准坐标系原点，并且相机在地球表面附近时，计算地球表面顶点的屏幕误差：地球与太阳的距离  $s$  约为  $1.5 \times 10^8$  km， $s$  的舍入误差  $\varepsilon_s$  为 16 km。设当前相机中心与地球表面的距离为  $n=1\ 000$  km，并且  $n \ll s$ ，则  $t=n+s$  的舍入误差  $\varepsilon_t$  同样为 16 km。再假设地球质心在基准坐标系中的位置  $\mathbf{V}_{\text{model}}=[0\ 0\ s]^T$  并且相机中心在基准坐标系中的位置  $\mathbf{V}_{\text{view}}=[0\ 0\ t]^T$  时，有

$$\begin{bmatrix} \varepsilon_{x_{\text{model}}} & \varepsilon_{y_{\text{model}}} & \varepsilon_{z_{\text{model}}} \end{bmatrix}^T = [0\ 0\ 16]^T$$

$$\begin{bmatrix} \varepsilon_{x_{\text{view}}} & \varepsilon_{y_{\text{view}}} & \varepsilon_{z_{\text{view}}} \end{bmatrix}^T = [0\ 0\ 16]^T$$

可由式(8)计算出  $\varepsilon_{V_{\text{view}}}$  22.627 km。视场角设为  $60^\circ$ ，当前窗口高度设为 900 像素， $z'=n$  为 1 000 km，可由式(9)计算出屏幕误差  $error$  为 17.6 像素。一般只要屏幕误差超过 0.5 像素，场景就会抖动。这样的屏幕误差会导致场景抖动非常严重。

### 1.3 时间尺度导致的显示误差

在统一的时间基准下，空间环境和空间目标按照运行规律不断地变换各自的空间位置和空间姿态。如地球随着时间的推移进行绕轴自转和绕太阳公转。一般情况下，跨尺度三维场景中的相机坐标系的定义是基于惯性系的。例如，当需要对地球进行观察时，相机的位置和姿态都是相对于地球的惯性系定义的，对相机施加的各种操作控制也是在地球的惯性系中进行的。但是，当需要近距离观察地球局部区域高分辨率地形时，会导致由于地球转速太快而无法将相机焦点保持在兴趣区域。通常做法是在地球自转时将地球的旋转矩阵也同样作用于相机的位置和姿态，使相机和兴趣区域的空间关系保持不变。

设地球在时刻  $t$  相对于其惯性系的旋转矩阵为  $\mathbf{R}_t$ ，若使相机和兴趣区域的空间关系保持固定，则对相机的位置  $\mathbf{V}_{\text{view}}$  和相机的姿态  $\mathbf{R}_{\text{view}}$  有：

$$\begin{aligned} \mathbf{V}'_{\text{view}} &= \mathbf{R}_t \mathbf{V}_{\text{view}} \\ \mathbf{R}'_{\text{view}} &= \mathbf{R}_t \mathbf{R}_{\text{view}} \end{aligned} \quad (10)$$

$\mathbf{V}'_{\text{view}}, \mathbf{R}'_{\text{view}}$  分别为相机变换后的位置和姿态。

此时地球表面上的一点坐标  $\mathbf{V}$  从局部坐标系转换到相机坐标系中的坐标  $\mathbf{V}'$  时有：

$$\mathbf{V}' = \mathbf{R}'_{\text{view}}^{-1} \mathbf{R}_t \mathbf{V} - \mathbf{V}'_{\text{view}}$$

因为旋转矩阵存在舍入误差，所以  $\mathbf{R}_t^{-1} \times \mathbf{R}_t \neq \mathbf{E}$ ，即  $\mathbf{V}' \neq \mathbf{R}'_{\text{view}}^{-1} \mathbf{V} - \mathbf{V}'_{\text{view}}$ 。因此有：

$$\mathbf{V}' = \mathbf{R}'_{\text{view}}^{-1} \mathbf{R}_t^{-1} \mathbf{R}_t \mathbf{V} - \mathbf{V}'_{\text{view}} \quad (11)$$

从式(11)得知，顶点  $\mathbf{V}$  从局部坐标系变换到相机坐标系经历了三次旋转变换和一次平移变换，每次变换都会造成舍入误差的积累。由于相机位置和姿态的计算和场景的渲染过程是独立的，所以不能预先在 CPU 中以双精度浮点数的方式提前计算矩阵连乘结果。并且由于不同的时刻地球的惯性系到本体系的旋转矩阵  $\mathbf{R}_t$  不同，所以不同时刻计算  $\mathbf{V}'$  的结果也不一定相同。因此，当近距离固定观察地面高分辨率地形某一区域时，会导致随着时间的走动地形出现轻微的晃动。

## 2 解决办法

空间尺度导致的显示误差是由于基准坐标系设置不合理造成的。由于场景空间尺度变化非常大，难以找到一个合适的坐标系作为场景统一的基准坐标系。因此，本文根据相机当前关注焦点的不同，动态地选择相应的基准坐标系以适应空间尺度的变化。当相机关注某一空间目标时，将基准坐标系设置在该目标的惯性系，其它空间目标的坐标可以通过平移、旋转等计算转换到基准坐标系中。当相机切换关注目标时，相应的基准坐标系也要进行切换到新关注目标的惯性系。

相机关注目标的顶点坐标从目标本体坐标系变换到相机坐标系由下式描述：

$$\mathbf{V}' = \mathbf{R}'_{\text{view}}^{-1} \mathbf{R}_t \mathbf{V} - \mathbf{V}'_{\text{view}} \quad (12)$$

$\mathbf{R}_t$  是目标惯性系到目标本体系的旋转矩阵。

首先将关注目标本体坐标系  $O-X_t Y_t Z_t$  旋转到该目标惯性系  $O-XYZ$  (与场景基准坐标系  $O-X_0 Y_0 Z_0$

重合), 再将  $O-XYZ$  旋转到与相机坐标系轴向相同的坐标系  $O-X_1Y_1Z_1$ , 最后将  $O-X_1Y_1Z_1$  平移到相机坐标系  $O-X'Y'Z'$ , 如图 4 所示。

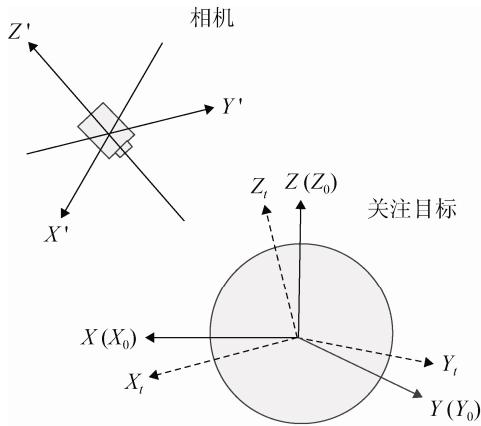


图 4 关注目标顶点坐标系变换示意图 (基准坐标系为惯性系)

非相机关注目标的顶点坐标从目标本体坐标系变换到相机坐标系由下式描述:

$$\mathbf{V}' = \mathbf{R}_{\text{view}}^{-1} (\mathbf{R}_0 \mathbf{R}_t \mathbf{V} + \mathbf{V}_0) - \mathbf{V}_{\text{view}} \quad (13)$$

$\mathbf{R}_0$  和  $\mathbf{V}_0$  分别是场景基准坐标系到非关注目标惯性系的旋转矩阵和平移向量。相机关注目标顶点坐标的变换也可由式(13)表达, 在这种情况下,  $\mathbf{R}_0$  为单位矩阵,  $\mathbf{V}_0$  为零向量。如图 5 所示, 图中基准坐标系为关注目标惯性系。

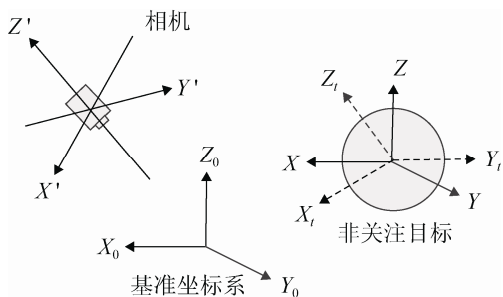


图 5 非关注目标顶点坐标系变换示意图

由式(8)可知, 顶点误差大小主要由目标坐标系原点在基准坐标系的位置  $\mathbf{V}_{\text{model}}$  和相机中心在基准坐标系的位置  $\mathbf{V}_{\text{view}}$  决定。采用动态基准坐标系后, 对于关注目标,  $\mathbf{V}_{\text{model}} = [0 \ 0 \ 0]^T$  不存在舍入误差,  $\mathbf{V}_{\text{view}}$  一般也不大, 舍入误差可忽略, 屏幕误差不会对画面质量产生影响。对于非关注目

标,  $\mathbf{V}_{\text{model}}$  和  $\mathbf{V}_{\text{view}}$  一般很大, 舍入误差也就比较大, 但是由于距离相机很远, 由式(9)知屏幕误差也会很小, 也不会对画面质量产生影响。

时间尺度导致的显示误差是由于空间目标的运动引起的。为了实现相机可以固定观察目标本体上的某个区域, 即保持相机和目标本体的空间关系不变, 必须不断地计算更新相机的位置和姿态。因为相机位置姿态的计算和场景渲染过程是独立的, 所以无法避免由于矩阵连乘带来的舍入误差。本文通过设置基准坐标系与关注目标本体系统相同的轴向可以避免相机位置和姿态的计算, 从而消除矩阵连乘产生的舍入误差。

非关注目标的顶点坐标从该目标本体坐标系变换到相机坐标系由下式描述:

$$\mathbf{V}' = \mathbf{R}_{\text{view}}^{-1} \mathbf{R}_t (\mathbf{R}_0 \mathbf{R}_t \mathbf{V} + \mathbf{V}_0) - \mathbf{V}_{\text{view}} \quad (14)$$

$\mathbf{R}_t$  是非关注目标惯性到该目标本体系的旋转矩阵,  $\mathbf{R}_t$  是关注目标本体系统到关注目标惯性系的旋转矩阵。首先将非关注目标从该目标本体坐标系  $O-X_iY_iZ_i$  旋转到该目标惯性系  $O-X_1Y_1Z_1$ , 然后将  $O-X_1Y_1Z_1$  旋转到与关注目标惯性系相同轴向的坐标系  $O-X_2Y_2Z_2$ , 再将  $O-X_2Y_2Z_2$  平移到关注目标惯性系  $O-X_3Y_3Z_3$ , 再将  $O-X_3Y_3Z_3$  旋转到关注目标本体系统(基准坐标系)  $O-X_0Y_0Z_0$ , 最后平移旋转到相机坐标系  $O-X'Y'Z'$  中, 最终完成顶点坐标的变换。如图 6 所示, 图中的基准坐标系  $O-X_0Y_0Z_0$  为关注目标的本体系统。

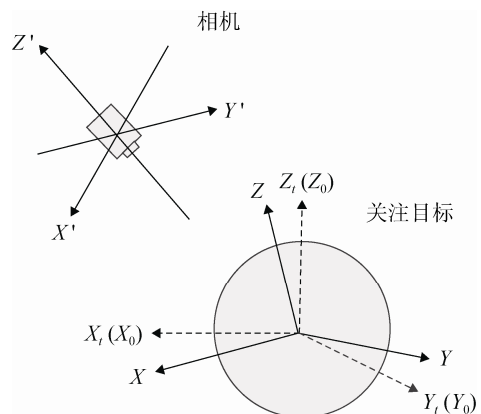


图 6 关注目标顶点坐标系变换示意图 (基准坐标系为本体系)

将式(15)改成齐次坐标的形式:

$$V' = M_{\text{view}} M_i M_0 M_t V \quad (15)$$

$M_{\text{view}} M_i M_0 M_t$  的连乘结果可以预先在 CPU 中以双精度浮点数的形式进行计算,然后就可以直接将矩阵计算结果传入 GPU 中对顶点进行批量转换,这样就避免由于矩阵连乘带来的浮点误差,而且可以大大提高转换速度。

### 3 实验结果

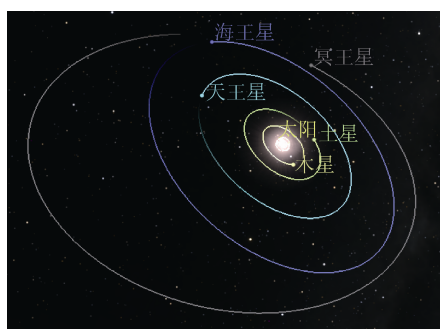
实验的硬件环境包括 Intel 酷睿 i5-3470 CPU; NVIDIA GeForce GT630 显卡; 1TB 硬盘; 4 GB 内存。实验的软件环境包括 Windows 操作系统; OpenGL 图形显示接口; Qt 跨平台 C++ 界面开发包。

在必备的软硬件环境的支撑下研发了跨尺度三维场景显示系统。系统包含太阳系的八大行星及其附属卫星。恒星背景采用国际上通行的依巴谷星表数据,该表包含了 2 539 913 颗星的精确位置数据。选用全球 30 m 分辨率的影像和 DEM 作为基础,局部地区采用 9 000×9 000 像素(5 m 分辨率)卫星正射影像和 1 024×1 024 格网点(5 m 格网间距)的 DEM 构建全球三维地形环境,如图 7 所示。

为了验证本文提出方法对提高空间尺度上显示精度的有效性,与太阳的距离从远到近分别以海王星、地球、水星为例对比实验,每组实验又分别设置太阳惯性系作为基准坐标系和设置关注目标惯性系作为基准坐标系进行对比。如图 8~10 所示。从图中可以看出,当设置太阳惯性系为基准坐标系

时,距离最远的海王星网格呈完全破碎状,距离次之的地球网格也有不同程度的破碎,距离最近的水星网格破碎程度较轻,因此可知,在单一的基准坐标系下,与基准坐标系距离越远的目标抖动就越严重。然而当设置关注目标惯性系为基准坐标系时,海王星、地球、水星均不存在抖动现象。从对比中发现,基准坐标系选择在关注目标的惯性系上时,显示精度明显提高。

当相机近距离固定观察所关注目标的某一区域时,为了验证本文提出方法对提高时间尺度上显示精度的有效性,分别将基准坐标系轴向设置为关注目标惯性系的轴向和设置为关注目标本体系轴向两种情况进行实验对比分析。在两种情况下分别按 1 s 间隔截取四帧图像进行分析。在不存在显示误差的情况下,这四帧图像应该是完全一样的。为了能更加有效地说明实验结果,对这四帧图像的相邻两帧进行逐像素的差值计算,生成三幅差值图。对这三幅差值图进行直方图统计,直方图分布越靠近 0 说明图像差值越小显示误差精度越高,如图 11 和图 12 所示。通过图中可以对比分析,第一组直方图统计中灰度值分布在[0, 20]区间,第二组直方图统计中灰度值几乎全部为 0。显然,第二组截取的四帧图像几乎完全一样不存在差值。由此可知,当相机跟随目标实现近距离固定观察时,基于目标本体系设置基准坐标系几乎不存在显示误差,而基于目标惯性系设置基准坐标系则存在显示精度不足的问题。



(a) 太阳系全局效果

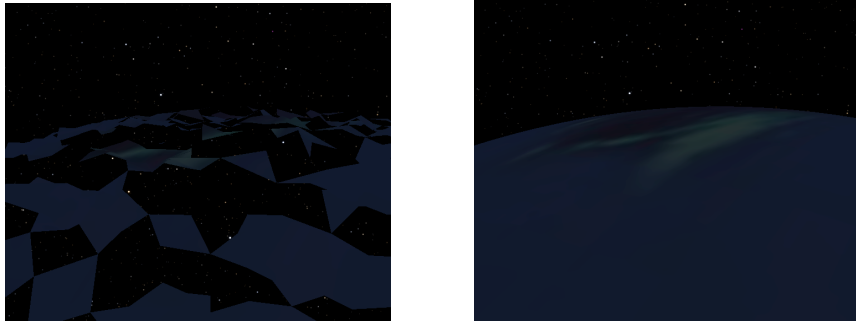


(b) 土星效果

图 7 跨尺度三维场景系统运行效果图

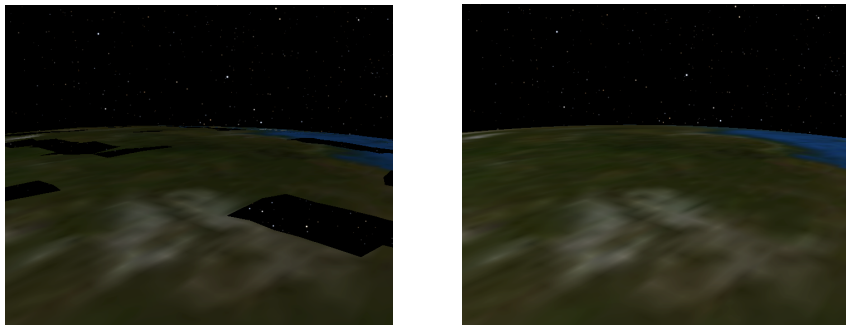
<http://www.china-simulation.com>





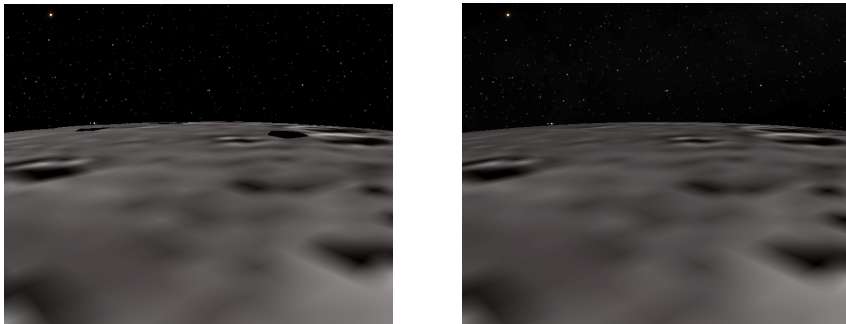
(a) 基准坐标系原点设置在太阳质心时观察海王星 (b) 基准坐标系原点设置在海王星质心时观察海王星

图 8 基准坐标系分别设置在太阳质心和海王星质心时观察海王星的效果(海王星距太阳约  $4.5 \times 10^9$  km)



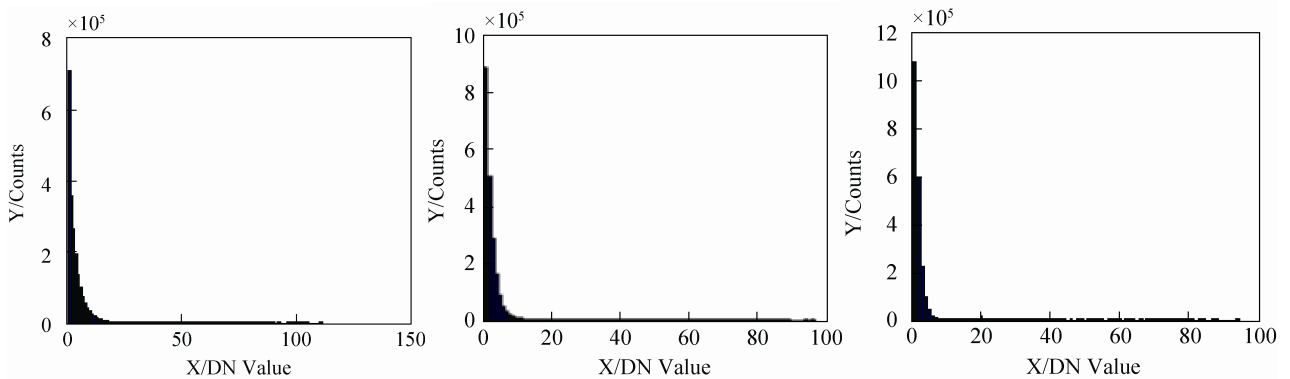
(a) 基准坐标系原点设置在太阳质心时观察地球 (b) 基准坐标系原点设置在地球质心时观察地球

图 9 基准坐标系分别设置在太阳质心和地球质心时观察地球的效果(地球距太阳约为  $4.5 \times 10^8$  km)



(a) 基准坐标系原点设置在太阳质心时观察水星 (b) 基准坐标系原点设置在水星质心时观察水星

图 10 基准坐标系分别设置在太阳质心和水星质心时观察水星的效果(水星距太阳约为  $5.8 \times 10^7$  km)



11 基于关注目标惯性系设置基准坐标系直方图统计

<http://www.china-simulation.com>

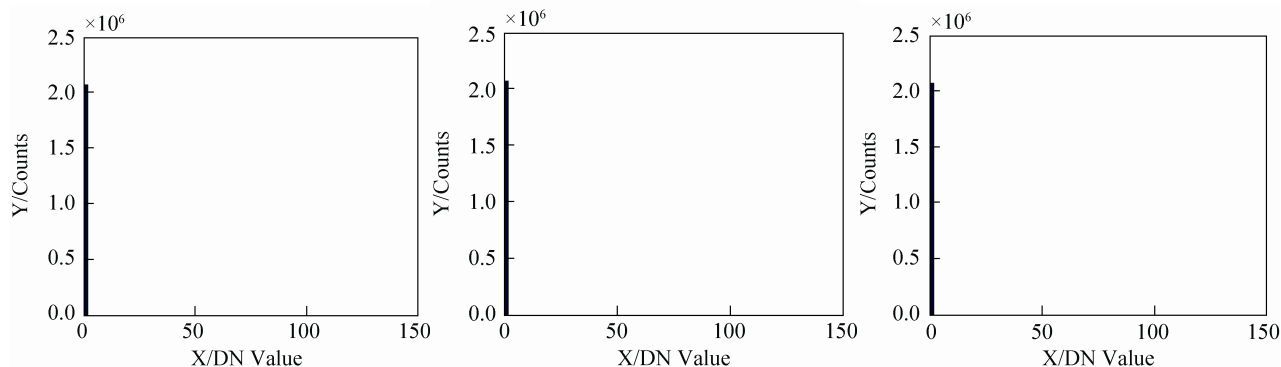


图 12 基于关注目标本体设置基准坐标系直方图统计

## 4 结论

跨尺度三维场景具有空间尺度和时间尺度。在空间尺度上,大到能容纳整个太阳系,小到能展现米级的高分辨率地形。在时间尺度上,空间目标在统一时间基准下按照规律不停地变化各自的空间位置和姿态,相机可以跟随关注目标本体实现固定观察。本文分析了两种尺度下显示误差的成因,提出了基于相机关注目标惯性系设置基准坐标系的方法,有效解决了空间尺度上显示精度不足的问题,提出了基于相机关注目标本体设置基准坐标系的方法,有效解决了在时间尺度上相机跟随目标实现近距离固定观察时显示精度不足的问题。本文方法对于解决跨尺度三维场景中存在的显示精度不足问题有着重要的作用。然而,本文的研究工作是在假设场景目标相互距离比较远、同一时刻只可能关注一个目标的基础上开展的,因此,当若干个目标距离较近而又需要同时关注这几个目标时,我们还需要判断选择哪个目标的坐标系作为场景基准坐标系的效果最好,这也是我们下一步的工作方向。

## 参考文献:

- [1] 吕亮. 空间态势图构建及可视化表达技术研究 [D]. 郑州: 信息工程大学地理空间信息学院, 2014.
- [2] 施群山, 徐青, 蓝朝桢, 等. 空天地一体化态势表达引擎的设计与实现 [J]. 系统仿真学报, 2014, 26(7): 1570-1576.
- [3] Bryant, O'Hallaron. Computer Systems: A Programmer's Perspective [M]. 2nd ed. Upper Saddle River, NJ, USA: Pearson Education, 2011.
- [4] 罗飞雄, 钟耳顺, 刘海志, 等. 一种解决全球地形绘制抖动问题的方法 [J]. 系统仿真学报, 2011, 23(3): 506-510.
- [5] 陈学文, 张炎. 载人飞行器在轨视景仿真图像抖动研究 [J]. 系统仿真学报, 2013, 25(7): 1592-1597.
- [6] 徐勇波, 王行仁, 贾荣珍. 飞行仿真器联网仿真视景图像抖动研究 [J]. 系统仿真学报, 2005, 17(2): 414-420.
- [7] 徐青, 施群山, 蓝朝桢, 等. 空天地一体化态势表达引擎及其视点跨尺度无缝切换方法 [P]. 中国: 201310533634, 2014-02-27.
- [8] ANSI/IEEE Standard 754-2008, IEEE Standard for Binary Floating Point Arithmetic [S].
- [9] Dave Shreiner, Graham Sellers, John Kessenich, *et al.* OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 4.3 [M]. 8th ed. Upper Saddle River, NJ, USA: Pearson Education, 2013.
- [10] 赵友冰, 石教英, 周骥, 等. 一种大规模地形的快速漫游算法 [J]. 计算机辅助设计与图形学学报, 2002, 14(7): 624-628.