

虚拟地形场景绘制中的实时 LoD 算法

杨崇源¹, 张继贤², 林宗坚²

(1. 华中科技大学 电子信息与工程系, 湖北 武汉 430074; 2. 中国测绘科学研究院, 北京 100039)

The Real-time LoD Algorithm for Rendering the Virtual Terrain Scene

YANG Chong-yuan¹, ZHANG Ji-xian², LIN Zong-jian²

(1. Department of Electronic and Information Engineering, Huazhong University of Science and Technology, Wuhan 430074, China; 2. Chinese Academy of Surveying and Mapping, Beijing 100039, China)

Abstract: In order to render the virtual terrain scene in real-time, we present a view-dependent continuous LoD algorithm in this paper. We analyze the regularity of the regular grid Digital Elevation Model and employ three view-dependent criterion to reduce the complexity of the terrain in real-time: ① screen space error; ② view-frustum intersection; ③ node orientation. The experiment results indicate that this scheme may attain a much higher frame rate in rendering the virtual terrain scene, as well as maintain the visual quality.

Key words: regular grid; view-dependent LoD; real-time simplification

摘要: 本文论述了基于视点的连续 LoD 算法在动态虚拟地形场景绘制中的应用。我们通过分析规则格网数字高程模型的规律性, 推导了 3 个与视点相关的判据来动态简化地形: ① 图像空间误差; ② 视锥台相交性; ③ 节点方向性。实验结果证明, 该算法在基本保持视觉效果不变的情况下, 极大地提高了虚拟地形场景的绘制速度。

关键词: 规则格网; 基于视点的 LoD; 实时简化

1 引言

实现虚拟环境的动态生成日益成为虚拟现实技术的一个重要研究方向。目前国内外研究热点是用细节层次模型 LoD(Level of Detail)对虚拟场景进行简化。LoD 技术由 Clark^[1]于 1976 年首次提出, 其主要思想是为虚拟场景中的物体建立多层次的细节模型, 在模型驱动时根据一定的规则选择相应的层次, 从而达到实时简化模型, 而又不影响视觉效果的目的。

文献[2]对用格网表示的地形模型提出了一种基于视点的连续 LoD 绘制方法。该算法的核心是利用格网的规律性, 设计了一种效率较高的算法来实现三角形对自下而上的合并, 从而实时生成与视点相关的连续 LoD 模型。合并的标准是新三角形与旧三角形对的误差在图像空间上的投影; 同时, 为加快绘制的速度, 采用了块作为地形模型大面积简化的基本单位。由于考虑了格网的规律性, 该算法在三角形对合并之后不需要重新

收稿日期: 1999-12-08; 修回日期: 2000-09-13

作者简介: 杨崇源(1972-), 男, 广西柳州人, 华中科技大学电子信息与工程系硕士研究生, 主要从事图像与图形方面的研究。

三角化,同时由于所采用的简化标准是基于视点相关的,因而所得到的简化模型既满足了视觉的要求,又在一定程度上节省了绘制时间。

当然,该算法还有许多需要改进的地方,主要体现在下面几个方面:① 格网的规律性利用不够充分;② 模型简化时没有考虑到区别对待视锥体之外的部分;③ 没有利用模型的前后向简化不可见面。

2 格网的规律性

图1为一个最简单的格网的层次模型。以图1(c)、图1(d)为例,图中的黑色节点连接着2个子三角形对(边界上的节点除外,它们只连接着一对三角形),三角形能否合并决定着节点的选取操作;在地形模型绘制时,我们将生成一个绘制节点队列(Render_Node_List),然后把无法合并的三角形组成的节点加入到队列中。我们定义图1(d)中的节点为偶类节点,定义图1(c)中的节点为奇类节点;对这两类节点来说,它的2个子三角形对必须同时存在,否则将产生T型三角形如图2所示(在后面的内容中将论述如何避免T型三角形的产生)。我们设定,最精细的图1(d)为第0层(偶数层),图1(b)为第1层(奇数层)……,依此类推。对于大小为 $(2^j+1) \times (2^j+1)$ 的格网来说,共可以分 $2j$ 层。绘制节点的选取从0层开始,按顺序进行。在偶数层中我们对偶类节点进行选取,奇数层则对奇类节点进行选取。从图1可以看出,上一层节点子三角形的合并状态将决定当前层相应节点子三角形能否合并。

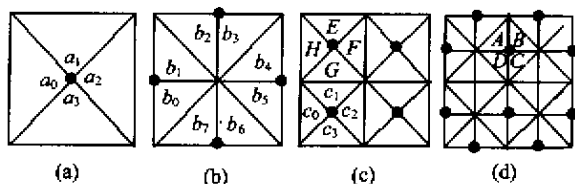


图1 格网的规律性

Fig.1 The regularity of lattice

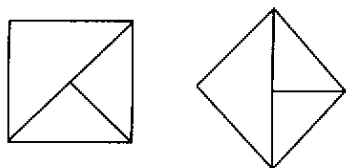


图2 两种类型的T型三角形

Fig.2 Two types of T triangle

3 基于视点的绘制节点选取准则

绘制节点列的生成是实时生成LoD的基本操作,因而生成算法的优劣直接影响到整个LoD算法的好坏。在文献[2]中,Lindstrom对节点的操作只采用了一种准则:图像空间误差 Screen_Space_Error(*Node);为在不增大误差的前提下提高算法的实时性,我们另添加了2个与视点相关的准则:视见区相交判断 View_Frustum_Intersection(*Node)和节点的方向性判断 Face_Orientation(*Node)。这3种判断准则组合在一起,构成了选取绘制节点模块:Node_Refine(*Node),其基本算法如下:

```

模块 Node_Refine(*Node)
{
    if 三角面与视锥台不相交 then 返回 0;
    //用函数 View_Frustum_Intersection(*Node) 判断节点的三角面是否在视锥台内
    else if // 在其内才进行下面的判断
        if 三角面背向观察者 then 返回 1;
        //用函数 Face_Orientation(*Node) 判断节点是否面向观察者
        else if //面向观察者才进行下面的判断
            if 子三角形图像空间误差都小于设定的阈值 then 返回 2;
            //用函数 Screen_Space_Error(*Node) 进行判断
        else if 返回 3;
}

```

在选取绘制节点时,若某节点计算值返回0,1,2,则表明该节点的子三角形需要参与进一步的合并;返回3,则可以将该节点加入绘制节点队列。(注:此算法的步骤设计只是针对节4(2)中的第②种情况。)

3.1 三角面与视锥台的相交判断

虽然我们利用OpenGL进行3维图形的绘制时,OpenGL自动进行了这样的判断,但这并非意味着我们的这部分工作是多余的,因为从Node_Refine(*Node)的基本算法中我们可以看出,经过View_Frustum_Intersection(*Node)的判断之后,我们有可能省却后面2步。

节点三角面与视锥台的相交的情况如图3所示,函数View_Frustum_Intersection(*Node)的主要功能就是判断三角面与由6个面组成的视锥

台是否相交; 相交则需要进行 Node Refine (*Node) 的下一步判断, 不相交则需要判断它应否参与简化合并。

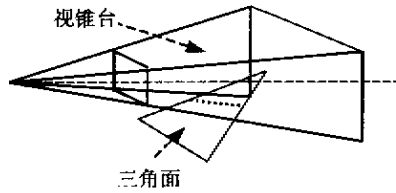


图3 三角面与视锥台的相交示意图

Fig. 3 The intersection of triangle face and view-frustum

当然, 在实际的算法中我们不需要计算每一个三角面的相交性, 因为整个数据将分块处理, 只有那些部分在视锥台之内, 部分在视锥台之外的子数据块内的三角面才需要逐个判断。

3.2 节点的方向性判断

所谓节点的方向性判断就是计算节点方向与视线方向的夹角。

方向性的判断多用到节点法线锥 (cone of normals) 的方法。该方法由 Shirman^[3] 提出, 文献 [4] 在基于 Merge_Tree 的 LoD 算法中, 进行了改进。如图 4 所示, 法线锥可由节点方向与法线锥半角确定。可做如下定义:

1. 节点法线锥为它包含的子三角面的法线的最小圆锥;
2. 定义节点法线锥中轴线方向 (如图 4) 为节点方向, 用向量 γ 表示;
3. 法线锥半角 (如图 4), 用 α 表示。

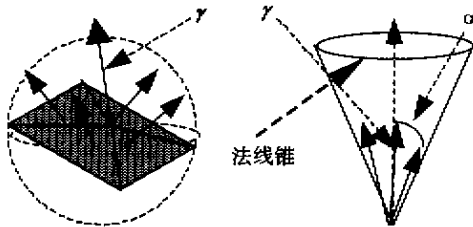


图4 节点法线锥

Fig. 4 The normal cone of node

根据该定义, 最精细的节点的 γ 与 α 可根据其所包含的最精细的三角面的法线求出, 如图 5(a), 节点 O 的 γ 与 α 可由子三角形 $\triangle AHO$ 与 $\triangle GHO$ 求出; 而高层节点的 γ 与 α 则可根据其所包含的、在上一层简化中得到的三角面的法线

求出。如图 5(b), 节点 O 的 γ 与 α 可由子三角形 $\triangle AFO$ 、 $\triangle AOB$ 、 $\triangle BOC$ 、 $\triangle COD$ 、 $\triangle DOE$ 、 $\triangle EOF$ 求出。

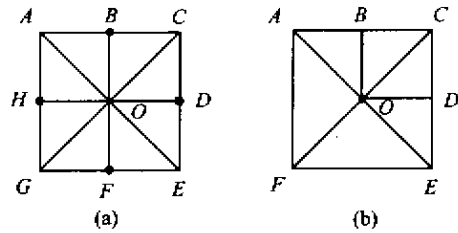
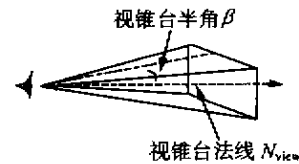


图5 节点与子三角形示意

Fig. 5 The relationship between node and sub-triangle

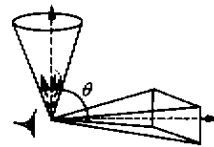
为进一步计算节点方向与视线方向的夹角, 我们做如下定义, 如图 6 所示:

1. 视锥台法线 N_{view} 为视锥台的中轴线方向, 即为视线方向;
2. β 为视锥台半角。



(a) 视锥台半角与视锥台法线示意图

(a) The relationship between half-angle of view-frustum and normal of view-frustum



(b) 视锥台法线与节点法线锥法线的夹角示意图

(b) The angle between normal of view-frustum and normal of node normal-cone

图6

Fig. 6

计算夹角的基本算法如下:

```

Face_Orientation(Node * node)
{
    N_cone = node -> γ;
    // 计算节点法线锥方向
    θ = arccos(N_cone * N_view); // 计算方向夹角
    if (θ - α - β > π/2)
        return Front_Facing; // 节点为前向
}

```

```

if( $\theta + \alpha + \beta < \pi/2$ )
    return Back_Facing; //节点为后向
return On_Silhouette; //节点在轮廓区域
}

```

根据计算出的方向结果,我们可进行如下3种处理:①当节点为前向时,需要进行图像空间误差的判断;②当节点为后向时,节点不需要进行图像空间误差的判断;③当节点在轮廓区域时,不仅需要进行图像空间误差的判断,而且其允许合并的误差阈值将小于为前向时的阈值,因为轮廓区域是最需要详尽表现的区域。

如上所述,对节点方向性的判断将使面向观察者的区域与轮廓区域得到充分的表现,而背向观察者的区域将在一定程度上得到省略。因而将进一步突出本文的 LoD 算法基于视点的特性。

3.3 图像空间误差判断准则

对应于图1,我们定义图7中的 O 为节点,三角形对 $\triangle ABO$ 与 $\triangle ACO$ 为子三角形, $\triangle ABC$ 为父三角形。根据格网的规律性,设 BC 的中点为 D ,定义有向线段 $\vec{DO}$ 为误差线段,长度为 δ 。绘制节点的选取(即子三角形对与父三角形的切换)将取决于 $\vec{DO}$ 在投影平面上的投影长度。下面我们将给出该投影长度 δ_{screen} 的计算过程。为此,我们先给出如下定义(设定在下面的推导过程中,下标 eye 与 screen 分别表示观察坐标系与投影坐标系,其余为世界坐标系):

1. 设世界坐标系的地平面为 XY 平面,垂直于 XY 平面向上的轴为 Z 轴;

2. 设定视点在世界坐标系中的坐标为 $E(e_x, e_y, e_z)$,观察坐标系中各坐标轴的单位向量在世界坐标系中表示为 $x(x_x, x_y, x_z), y(y_x, y_y, y_z), z(z_x, z_y, z_z)$;

3. 设定点 D 的坐标为 (x_d, y_d, z_d) ,则 O 点为 $(x_d, y_d, z_d + \delta)$,有向线段 $\vec{DO}$ 的中点 V 的坐标为 $(x_d, y_d, z_d + \delta/2)$;设定 Dis 为视点 E 到投影平面的距离。

同时,我们可作如下的近似:①对 $\vec{DO}$ 进行投影时,其中点沿 Z 轴的负向,位于视区的中央(详见文献[2]);②由于 $\delta \ll \|E - V\| = -V_{eye_z}$,因而可以设定 $D_{eye_z} \approx O_{eye_z} \approx V_{eye_z}$;世界坐标系到观察坐标系的变换矩阵为

$$P = \begin{bmatrix} x_x & x_y & x_z & 0 \\ y_x & y_y & y_z & 0 \\ z_x & z_y & z_z & 0 \\ -E \cdot x & -E \cdot y & -E \cdot z & 1 \end{bmatrix}$$

由于 x 与 y 始终垂直于 $E - V$,因而可得出如下等式

$$\vec{DO}_{eye} = \vec{DO} \cdot P = \delta[x_z, y_z, z_z, 0]$$

$$x_z^2 + y_z^2 + z_z^2 = 1$$

故误差线段在投影平面上的长

$$\delta_{\text{screen}} = \|\vec{DO}_{\text{screen}}\| = Dis \frac{\sqrt{\vec{DO}_{eye_x}^2 + \vec{DO}_{eye_y}^2}}{-V_{eye_z}} = Dis \frac{\sqrt{(\delta \cdot x_z)^2 + (\delta \cdot y_z)^2}}{\|E - V\|}$$

根据近似

$$z = (x_z, y_z, z_z) = ((E_x - V_x)/\|E - V\|, (E_y - V_y)/\|E - V\|, (E_z - V_z)/\|E - V\|)$$

因而

$$\delta_{\text{screen}} = Dis \frac{\sqrt{1 - (\frac{E_z - V_z}{\|E - V\|})^2}}{\|E - V\|} = Dis \frac{\delta \sqrt{(E_x - V_x)^2 + (E_y - V_y)^2}}{(E_x - V_x)^2 + (E_y - V_y)^2 + (E_z - V_z)^2}$$

为方便起见,我们采用 δ_{screen}^2 进行判断,对于用户指定的误差范围 μ ,如果 $\delta_{\text{screen}}^2 \leq \mu^2$,则认为节点的子三角形可以合并,否则将该节点加入到绘制节点队列中。

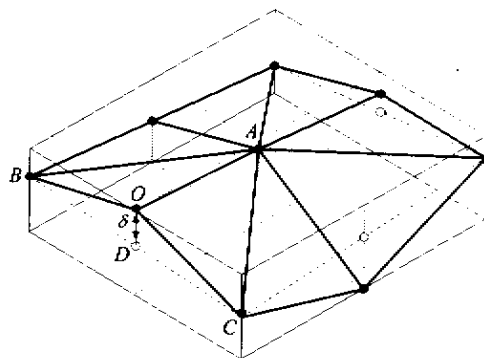


图7 格网的父三角形与子三角形对误差示意图

Fig. 7 The error between father-triangle and sub-triangle of lattice

在图8与图9给出了应用上述标准进行简化的相应示例。图8显示出随视点的变远,我们绘制的三角形数目呈现出一种连续的下降趋势。在图8中,(a)为 513×513 的原始格网,共绘制了

262 144个三角形;(b),(c),(d)用了图像空间误差简化, $\mu=0.0008$,随着视点逐渐变远,绘制的三角形数目逐渐下降,分别为 63 541,45 209,7 552。在图 9 的实验中我们假设视锥台与格网相交成一个三角形;图 9(b)显示,应用视锥台相交性判据进行判断后,所绘制的三角形的数目减少了约一半。当然,不出所料的是,地形模拟的视觉效果也保持得非常好,这一结论不难从图 8(a)与图 8(c)的比较中得出,它们的视点坐标是相同

的。在图 9 中,(a)为 $1\,025 \times 1\,025$ 的格网,只用图像空间误差简化, $\mu=0.0008$,三角数目 234 578;(b) 加上了视锥台相交判断, $\mu=0.0008$,三角形数目 128 529。

实际上,我们研究的基于视点的连续 LoD 算法是一种将地形的客观特征与视觉效果充分结合的算法。从示例中可以看到,它实际上融合了 TIN (Triangulate Irregular Networks) 模型的优势,在平坦的地方,它会用较稀的三角网来表现。

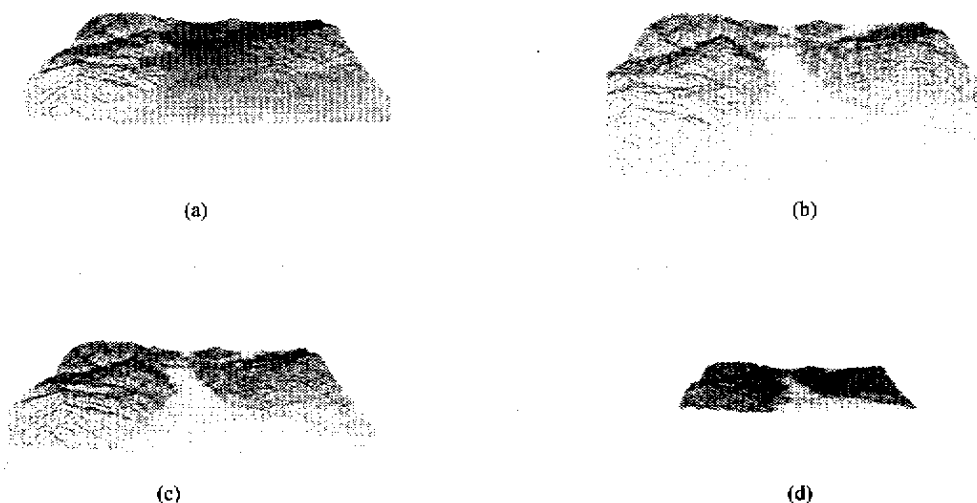


图 8
Fig. 8

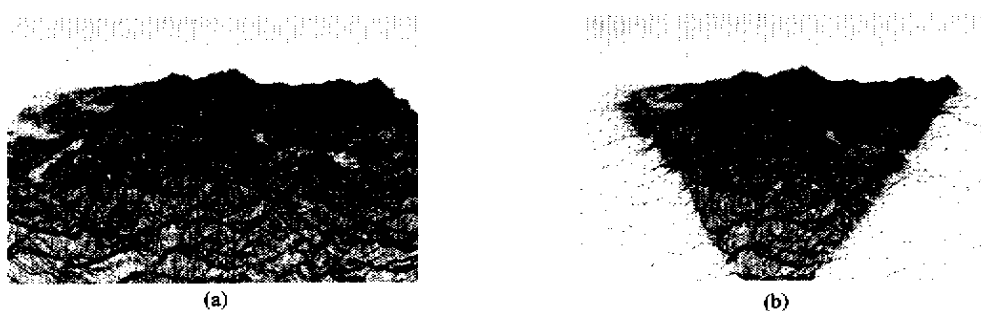


图 9
Fig. 9

4 实时 LoD 整体算法

在 2,3 节我们介绍了格网的规律性及选取绘制节点的准则,在本节中我们将阐述如何利用这

些规律性及准则来形成整体的实时 LoD 算法,从而实现对虚拟地形的实时 3 维漫游。实验中我们选取了 $1\,025 \times 1\,025$ 的格网数据块。为简化算法,我们将整个格网分成 4×4 的子数据块。下面是相

应的步骤:

1. 数据的预处理。在预处理的过程中主要是计算最精细节点的法线。

2. 基于子数据块的操作。如3.1节所述,基于子数据块的操作主要是确定它们的节点的可见性状态:①全部可见;②部分可见;③完全不可见。

3. 子数据块内的操作。我们将根据步骤(2)中确定的可见性状态,有针对性地对每一个节点进行处理。对于①我们只需用3.2节与3.3节的准则;对于②我们需用所有准则;对于③,不需对其节点的三角形进行任何判断,可直接合并。

4. 子数据块间的操作。仅按照上面的算法,子数据块边界上的节点在最终的绘制中将产生大量的T型三角形。为此,我们为边界上的节点建立了一个标志位,来保证对应的节点合并的同时性。

5. 绘制节点列的建立。对于不能继续进入到下一层参与合并的节点,可添加到绘制节点列中。最终在绘制虚拟地形场景时,我们只需绘制队列中的三角形。

5 实验结果

实验机器配置:PII 350,内存64 M,硬盘

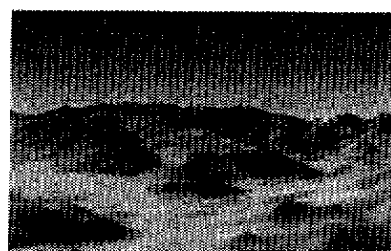
12 G,8 M显存。实验数据是大小为 $1\,025 \times 1\,025$ 的网格。我们选择了相同的漫游路线对下列4种情况作了比较:①不作任何地形简化;②仅用图像空间误差判据的LoD;③用图像空间误差判据及视见区相交判断的LoD;④利用了3种判据的LoD。其中 $\mu=0.000\,8$,表1列出了相应的实验结果。

表1 4种实验的结果

Tab. 1 Results of four experiments

实验序号	平均每秒的帧数	第1帧绘制的三角形数目
①	0.80	1 048 576
②	3.82	183 209
③	5.15	114 556
④	5.36	98 133

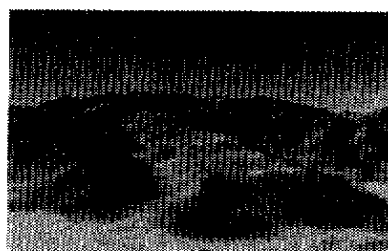
从表1我们可以看出,由于绘制三角形的减少,实验③比实验②的速度有了较大的提高;实验④比实验③有了一定的提高,但不是特别显著,这主要跟具体的地形有关,对于较陡峭的地形,实验④的效果将会更好。图10为实验④的前4帧场景图。



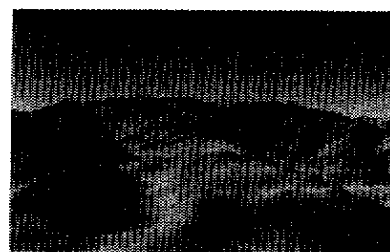
(a) 第1帧
(a) The first frame



(b) 第2帧
(b) The second frame



(c) 第3帧
(c) The third frame



(d) 第4帧
(d) The fourth frame

图10 实验④的前4帧场景图

Fig. 10 The prior four scenes of experiment ④

6 结论及今后的工作

我们将基于视点变化的连续 LoD 算法用于虚拟地形的实时简化,取得了一定的成果。在算法中,我们结合格网的规律性应用了3种判据进行地形简化:图像空间误差;视锥台相交性;节点方向性。该算法在保持视觉效果的情况下,使得虚拟地形场景的绘制速度有了较大的提高。在今后的工作中我们将在以下3个方面努力以使我们的算法能更快更好地绘制虚拟场景:①利用帧间的相关性优化算法;②利用三角条(triangle strip)技术以提高绘制的速度^[3];③研究相应的纹理映射的算法,加快纹理映射速度。

参考文献:

- [1] CLARK J H. Hierarchical Geometric Models for Visible Surface Algorithms[J]. Communication of

the ACM, 1976, 19(9): 547-554.

- [2] DEHAEMER J R, MICHAEL J, MICHAEL J Z. Simplification of Objects Rendered by Polygonal Approximations[J]. Computer and Graphics, 1991, 15(2): 175-184.
- [3] SHIRMAN L, ABI-EZZI S. The Cone of Normals Technique for Fast Triangle Meshes [A]. Proceedings of Eurographics'93[C]. Amsterdam: Elsevier Science Publishers B V, 1993. 261-272.
- [4] JULIE C X, JIHAD E S, AMITABH V. Adaptive Real-time Level-of-Detail-Based Rendering for Polygonal Models[J]. IEEE Transactions on Visualization and Computer Graphics, 1997, 3(2): 91-105.
- [5] FRANCINE E, STEVEN S, AMITABH V. Optimizing Triangle Strips for Fast Rendering [A]. Proceedings of Visualization'96[C]. New York: IEEE Computer Society Press. 1996. 319-326.

全球定位系统技术应用协会第六次年会征集学术论文

中国全球定位系统技术应用协会第六次年会将于2001年8月召开,为推动GPS事业发展,鼓励广大会员、专业技术人员更好地交流经验,总结科研、应用成果,面向西部大开发,发挥GPS技术的作用,中国全球定位系统技术应用协会办公室将组织编辑学术论文集。在年会上将评选优秀论文,并对获奖者进行奖励。

论文范围包括技术、应用与管理。技术方面:空间数据基础设施(定位数据、数据的获取、存储更新、交换;数据网络;数据的管理和维护),高精度DGPS、WAAS和LAAS, GPS/GNSS/GLONASS卫星系统研究,大气层影响,时间同步, GPS天线与接收机,蓝牙(Bluetooth)技术, WRC-2000, GSM与GPS的集成。

应用与管理方面:GIS与GPS集成应用,陆地、航天、航海指挥调度、动态监控、自主导航等安全系统, GPS与精细农业, GPS与环境监测,工程(地壳、建筑物)形变监测, GPS现代化, GPS政策阐述。

要求参加评选的论文立论正确、有创新、文字通顺、图表清晰。已在公开刊物上发表的文章不得参加评选。今年拟评出一等奖1名,奖金2000元,二等奖2名,奖金1000元,三等奖3名,奖金500元,获奖论文均发给奖状,并安排在年会上作报告,向有关公开刊物推荐发表。

论文征集截止日期2001年6月10日。有关论文格式及其他要求,请与中国全球定位系统技术应用协会办公室联系,欢迎各界人士踊跃投稿。

地址:北京复外三里河路50号 中国全球定位系统技术应用协会 邮编:100045

联系人:王丽 韩丽华 联系电话:(010)68512952 E-MAIL:wangli@sbsm.gov.cn

中国全球定位系统技术应用协会