

# 基于 Hamiltonian 三角剖分的三角网格多分辨率表示<sup>\*</sup>)

詹海生 周利华

(西安电子科技大学网络教育学院 西安 710071)

**摘要** 三角网格模型的多分辨率表示是几何模型绘制与传输的基础,本文通过三角形之间的拓扑相邻关系将三角网格划分为广义三角形带的集合,然后利用 Hamiltonian 三角剖分的性质构造三角网格的多分辨率表示。该方法统一了单分辨率网格和多分辨率网格的表示方法,当模型有  $c$  个不同分辨率表示时,其编码效率为  $(\log c + 5)$  bit/vertex。  
**关键词** 多分辨率表示, Hamiltonian 三角剖分, 三角网格, 计算几何

## Multiresolution Representation of Triangle Mesh Based on Hamiltonian Triangulation

ZHAN Hai-Sheng ZHOU Li-Hua

(School of Network Education, Xidian University, Xi'an 710071)

**Abstract** The multiresolution representation of a triangle mesh is base of rendering and distributing geometric model. In this paper, according to the topological adjacent relations among triangles, the triangle mesh is parted into a set of generalized triangle strips in which the dual graph of every strip is Hamiltonian path or cycle. The multiresolution representation of triangle mesh is constructed based on the Hamiltonian triangulation of polygon. By the way, the single and multi resolution representation have same codec methods. If representing a triangle mesh in  $c$  of resolutions, the code rate is bit/vertex.

**Keywords** Multiresolution representation, Hamiltonian triangulation, Triangle mesh, Computational geometry

### 1 介绍与相关工作

三角网格是计算机图形学中一种主要的几何模型表示方法,三角形是图形处理器可以直接处理的基本几何元素。三角网格模型的多分辨率表示在三维模型的实时绘制和在互联网中的传输中占有重要地位。在三角网格的多分辨率表示中,原始三角网格  $M$  通常被表示成基础网格  $M^0$  和不同精度的几何细节  $D^i$  的集合,即,

$$M = M^0 \oplus D^0 \oplus D^1 \oplus \dots \oplus D^{n-1}$$

关于多分辨率网格压缩的最早研究是 Hoppe<sup>[1]</sup> 的渐进网格方法,在他的方法中,通过边收缩计算简化网格,通过顶点分裂构造高分辨率网格,一个几何模型的渐进网格表示由一个基础网格和一系列顶点分裂操作构成,如下式所示:

$$M^0 \xrightarrow{vsplit_0} M^1 \xrightarrow{vsplit_1} \dots \xrightarrow{vsplit_{n-1}} M^n$$

其中,  $M^0$  是基础网格,  $vsplit_i$  是顶点分裂操作,  $M^n$  是原始网格。由于直接记录了分裂顶点的编号及其相邻顶点的标志,在该方法中,渐进网格的拓扑信息的存储量非常大,如果网格中有  $N$  个顶点,则编码率为  $(\log N + 5)$  bit/vertex。为了减少直接记录分裂顶点编号导致的存储量,相继出现了文<sup>[2~6]</sup>等改进算法。在这些方法中基础网格采用单分辨率编码方法,而集合细节采用其他方法,这样导致对基础网格和多分辨率网格的处理方法不一致,增加了系统复杂度。

本文算法首先利用三角形之间的拓扑相邻关系将三角网格表示成广义三角形带的集合,而每个三角形带为其边界形成的多边形的 Hamiltonian 三角剖分,利用 Hamiltonian 三角

剖分的性质构造多分辨率三角网格。本文算法统一了基本网格与多分辨率网格的拓扑关系压缩编码方法,可以直接生成所需分辨率的网格,解码时只需进行比较运算,效率较高,适用于对实时性要求较高的应用。同时,本算法的拓扑关系压缩率只与分辨率等级数量  $c$  有关,为  $(\log c + 5)$  bit/vertex。

### 2 Hamiltonian 三角剖分

**定义 1** 当且仅当一个多边形的三角剖分的对偶图含有一个 Hamiltonian 通路或者回路时,该三角剖分为 Hamiltonian 三角剖分。

**性质 1** 如果一个多边形的三角剖分为 Hamiltonian 三角剖分,则多边形边界中任意一边收缩后,所形成的新多边形仍然可以进行 Hamiltonian 三角剖分。

**证明:** 分 3 种情况,即三角剖分的对偶图为 Hamiltonian 通路、三角剖分的对偶图为 Hamiltonian 回路并且内部边界为环以及三角剖分的对偶图为 Hamiltonian 回路而内部边界为折线,如图 1 所示。对于第一种情况,每个多边形边界只与一个三角形对应,因此任何边界的收缩只减少三角剖分集合中的一个三角形,该三角形的一个边收缩后,剩余两边合并成一边,它前面和后面剩余的三角形集合的对偶图仍为 Hamiltonian 通路,并且通过该边可以合成为一个 Hamiltonian 通路,如图 1(a)、(d) 所示。对于第二种情况,无论收缩内边界还是外边界中的边,也只减少三角剖分集合中的一个三角形,该三角形的一个边收缩后,剩余两边合并成一边,新的三角剖分的对偶图仍为 Hamiltonian 回路,如图 1(b)、(e) 所示。对于第三种情况,收缩外边界中的边,也只减少三角剖分集合中

<sup>\*</sup> 论文的部分工作由国家“十五”军事预研基金资助项目(编号:413160501)支持。詹海生 讲师,博士研究生,主要研究方向为计算几何、几何压缩与传输、计算机动画以及多媒体技术等。

的一个三角形,该三角形的一个边收缩后,剩余两边合并成一边,新的三角剖分的对偶图仍为 Hamiltonian 回路。然而,如果收缩内边界中的一个边将减少三角剖分集合中的两个三角形,相当于将这两个三角形的公共边收缩,但是由于这两个三角形中剩余两边均合并成一边,新的三角剖分的对偶图仍为

Hamiltonian 回路,如图 1(c)、(f) 所示。因此,一个多边形的三角剖分为 Hamiltonian 三角剖分,收缩其任意一个边界,所形成的新多边形的三角剖分仍为 Hamiltonian 三角剖分。证毕。

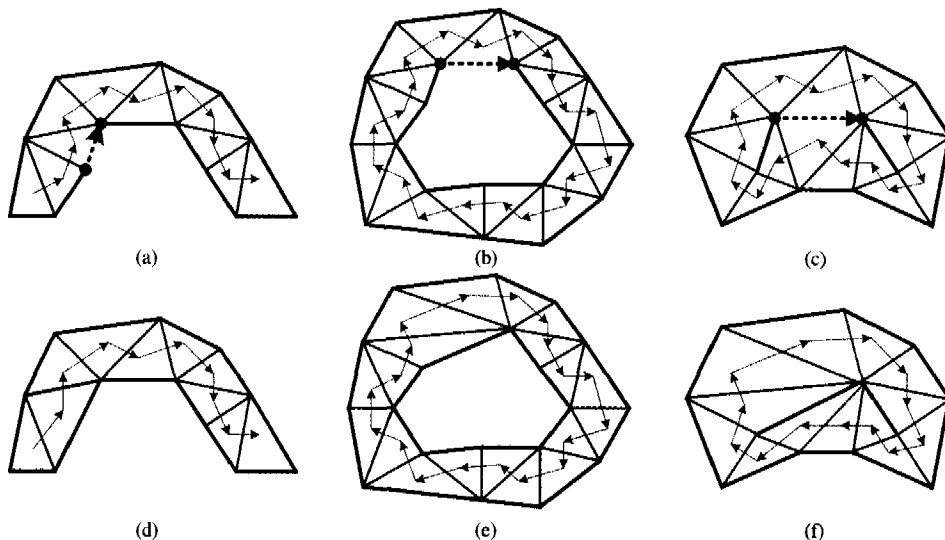


图 1 Hamiltonian 三角剖分及其性质

(a)对偶图为 Hamiltonian 通路和三角剖分,(d)为(a)中多边形边界中边收缩后的结果,  
 (b)对偶图为 Hamiltonian 加路并且内边界为环的三角剖分,(e)为(b)中多边形边界中边收缩后的结果,  
 (c)对偶图为 Hamiltonian 回路并且内边界为线的三角剖分,(f)为(c)中多边形边界中边收缩后的结果。

### 3 多分辨率三角网格的构造与编码

三角网格  $M$  可以表示为一个二元组:

$$M(V, F)$$

其中,

$$V = \{v_i | i = 0, 1, 2, \dots, n-1\},$$

$$F = \{f_i | i = 0, 1, 2, \dots, m-1\}$$

分别是三角形网格  $M$  的顶点集合和三角形集合。

定义 2 设  $v, t$  分别是三角网格  $M$  上的一个顶点和三角形,如果  $v$  是  $t$  的一个顶点,那么称  $v$  和  $t$  是相邻的。

定义 3 设  $e, t$  分别是三角网格  $M$  上的一个边和三角形,如果  $e$  是  $t$  的一个边,那么称  $e$  和  $t$  是相邻的。

定义 4 设  $v, e$  分别是三角网格  $M$  上的一个顶点和一条边,  $s, t$  分别是三角网格  $M$  上的两个三角形,如果  $v$  和  $s, t$  都是相邻的或者  $e$  和  $s, t$  都是相邻的,那么称  $s$  和  $t$  是相邻的。

#### 3.1 广义三角形带的构造

本文算法首先根据网格中三角形与三角形之间的相邻关系,即共点和共线的关系,由选定的种子三角形开始,求出包围该三角形并与该三角形相邻的三角形集合,将该集合用广义三角形带表示,然后再计算该三角形集合的外边界,用相邻关系计算包围该边界的所有三角形集合,用广义三角形带表示,依次类推,直到处理完整个三角网格。经过上述步骤整个三角网格被分解为广义三角形带序列的集合,本文的多分辨率表示就是针对每个三角形带序列进行的。

对于同一个广义三角形带序列,其中相邻的三个顶点序列  $B_{i-1} = \{v_0^{i-1}, v_1^{i-1}, v_2^{i-1}\}$ 、 $B_i = \{v_0^i, v_1^i, \dots, v_6^i\}$  与  $B_{i+1} = \{v_0^{i+1}, v_1^{i+1}, \dots, v_6^{i+1}\}$ 、 $B_{i-1}$  与  $B_i$  之间构成一个三角形带,  $B_i$  与  $B_{i+1}$  之间同样也构成一个三角形带,并且与边界  $B_i$  相邻的三角形都完全包含在这两个广义三角形带中,即在构造完这两个三

角形带后,边届  $B_i$  中的顶点将不再使用,如图 2 所示。在同一个广义三角形带中,首先被处理的边界为内边界,如  $B_i$  与  $B_{i+1}$  构成的三角形带中的  $B_i$ ,其次被处理的边界为外边界,如  $B_{i+1}$ 。按照广义三角形带中三角形的顺序,依次对组成该三角形的顶点来自哪个边界进行编码,如果来自内边界编码为 1,否则为 0。如图 2 所示,对于  $B_i$  与  $B_{i+1}$  构成的三角形带中的第一个三角形  $\Delta v_0^i v_0^{i+1} v_1^i$ ,顶点  $v_0^i$  和  $v_1^i$  来自内边界  $B_i$ ,  $v_0^{i+1}$  来自外边界  $B_{i+1}$ ,则该三角形的编码为 101,为了标志三角形带的方向,其中的第一个三角形的第一个顶点来自内边界,第二个顶点来自外边界,第三个顶点根据实际情况编码,而第三个顶点的与边  $v_0^i v_0^{i+1}$  的关系确定了三角形带的方向,在三角形带的解码过程中该方向决定三角形构造。 $B_i$  与  $B_{i+1}$  构成的三角形带的编码为:1010010011001001101001,如图 2 中所示。由于每个顶点被编码两次,每次用一个二进制位,整个网格模型的单分辨率编码效率为 2bit/vertex。

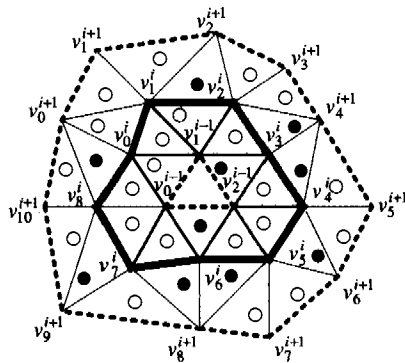


图 2 广义三角形带序列的编码  
 实心点表示 1,空心点表示 0。

### 3.2 多分辨率角网格构造与编码

模型简化是多分辨率表示的基础,边收缩是简化几何模型的一种有效的方法,该方法首先按照局部误差从小到大的顺序执行边收缩操作。收缩一条边的结果是把边两端的顶点合并为一个顶点。因此,在边收缩方法中,确定边收缩的顺序时,首先确定一个误差策略,即某一边收缩后对原模型的影响程度,按照影响程度的大小排序而得到边的优先级队列,合并误差最小的边的两个顶点,更新顶点合并后所影响的边的简化误差,同时更新边的优先级对列。因为误差度量过程常需要不少的计算量,而且还要对边的优先级序列进行更新,这要涉及到排序,所以高质量的几何简化算法的速度往往很慢。

本文的边收缩策略采用简单的边长准则为判定依据,利用边长准则对顶点的重要性进行排序,同时考虑广义三角形带的特点,来定义顶点  $v$  的重要度  $I(v)$ ,那些重要度低的顶点首先被删除,并且把被删除的顶点合并到与其重要度有关的顶点上。由于对于一个顶点来说,在不同分辨率网格中的重要度不断变化,本文重要度定义如下,

$$I(v) = \min_{i=1}^c \|v - v_i\|$$

其中,  $c$  为分辨率等级,  $v_i$  是顶点  $v$  在分辨率为  $c$  的网格的三角形带中的参考顶点,如图 3(a) 所示的灰色顶点,参考顶点包括外边界中位于它前面的一个顶点  $v_1$  以及内边界中它的相邻顶点,如  $v_2$  和  $v_3$ 。符合条件的顶点称为锚顶点,如图 3(b) 中  $v_1$  和图 3(c) 中的  $v_3$ ,锚顶点的重要度大于该顶点。参考顶点不包括外边界中位于其后的相邻顶点,如  $v_4$ ,也不包括与顶点  $v$  相邻的其他顶点,如  $v_5$  以及  $v_6$ 。在模型分辨率变化时,由于边收缩一些顶点的重要度相应变化,如果记录顶点在不同分辨率网格中的重要度,数据量将膨胀很多,本文只记录每个顶点的最大重要度,并记录最大重要度的锚顶点。按照重要度的计算方法,在同一个三角形带中内边界顶点的重

要度已知,只要计算外边界中的顶点的重要度,对于外边界中的顶点其重要度要么由与外边界中的相邻顶点的距离决定,要么由与内边界中顶点的距离决定,这两种情况分别对应一种边收缩类型,对于前者,当对应的边被收缩后,根据 Hamiltonian 三角剖分的性质,剩余部分仍为 Hamiltonian 三角剖分,仍可以表示成广义三角形带,并且不影响其他顶点的拓扑关系编码,这是因为拓扑关系编码只是标志该顶点位于内边界还是外边界,如图 3(b) 所示。当外边界中的顶点合并到内边界上时,广义三角形带中两个相邻三角形被删除,而剩余部分仍为广义三角形带,满足 Hamiltonian 三角剖分的性质,其他顶点的拓扑关系编码不变。与第一种情况不同的是,需要用内边界中的锚顶点替代外边界中的顶点,如图 3(c) 所示,需要用锚顶点  $v_3$  替代外边界中的顶点  $v$ 。为了解决这个问题,需要记录锚顶点位置,在参考顶点中锚顶点用二进制 1 表示,其它顶点用 0 表示,对于锚顶点以后的顶点的编码不记录,图 3(b) 所示的锚顶点  $v_3$  的编码为 001。由于三角网格的平均权值为 6 的特点,因此参考顶点编码的平均最大位数为 3。另外,根据重要度的范围分成不同等级,对这些等级编码,如果等级数为  $c$  则重要度编码为  $\log c$  个二进制,本文取  $n_c = 16$ 。该编码策略保证边收缩操作只在同一个三角形带中进行,并且边收缩操作不影响未收缩边的相邻顶点的编码,保证了拓扑关系编码的一致性,从而保持了广义三角形带的 Hamiltonian 三角剖分特点。广义三角形带的多分辨率编码的过程与单分辨率编码过程类似,只是在编码过程要增加与顶点的重要度相关的编码,包括重要度等级和重要度的参考顶点的标志,因此,本文的多分辨率编码方法中,每个顶点的编码包括顶点的位置,即位于内边界还是外边界,占 2 个比特,顶点重要度等级编码,占  $\log c$  个比特,锚顶点编码,平均 3 个比特,因此总的编码效率为  $(\log c + 5) \text{ bit/vertex}$ 。

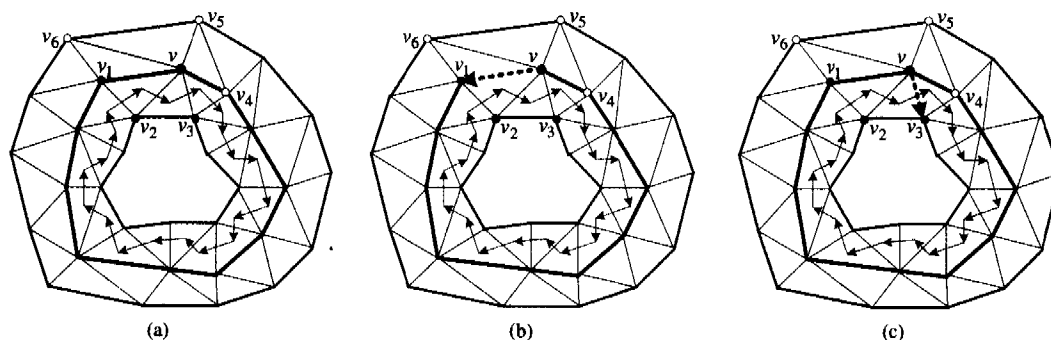


图 3 按顶点重要度进行的边收缩对广义三角形带的影响

(a) 顶点的重要度的计算以及参考点,灰色顶点所示; (b) 广义三角形带的外边界中的边收缩的影响; (c) 广义三角形带中内部边的收缩的影响。

## 4 实验结果

图 4 所示为 Nefertiti 模型的三个不同分辨率的表示,其中(b)、(c)中的模型均以(a)中模型的广义三角形带为基础构造而得,表 1 为主要结果数据的比较。从中可以看出随着分辨率的降低,三角形带的数量以及每个三角形带中顶点的数量同时降低。

**结论与未来工作** 本文研究了三角网格模型的多分辨率绘制与渐进式传输的问题,以广义三角形带为基础,利用 Hamiltonian 三角剖分的性质,给出了多分辨率三角网格的表示方法,该方法统一了单分辨率网格与多分辨率网格的表示

形式,便于用同一的算法实现对二者的操作,总的编码效率为  $(\log c + 5) \text{ bit/vertex}$ 。进一步的工作包括对该法在实时应用中改进算法以及高精度的简化算法的实现。

表 1 Nefertiti 模型的多分辨率表示的结果

模型	顶点	三角形带 (单个三角形)	最小边长	最长三角形 带的顶点数
(a)	299	33(24)	2mm	54
(b)	203	25(18)	4mm	47
(c)	153	21(16)	6mm	39

(下转第 242 页)

一个状态序列上的,因此这是一个线性特性,不能用 TCTL 公式来表示,而要用时间化的线性时序特性规格说明语言(参见定义 5)来描述。为了验证的方便,由于活性的否定形式是安全性<sup>[14]</sup>,因此,我们验证系统中的该特性的否定形式是不可满足的。liveness 的否定可用图 2 中的 TBA  $B_2$  来表示,这样一来,该特性就可以通过对  $B_2$  建模为 PVS 的 TBA 理论,然后与建模 TGC 系统的 TBA 求积得到一个新的 TBA,利用判断该 TAB 所包含的语言是否为空来验证 liveness 是否成立。

根据定义 5,若所有的运行都不能进入到 TBA 的接受位置(Reject),则不存在可接受的运行,则该 TBA 所接受的语言为空。这验证简单起见,可以先证明 Reject 不可达:

strong-liveness: THEOREM  
 $\forall (r: (lv\_init\_run?), i: nat); r(i)'loc'2 \neq Reject$

其中,lv 是 TA TGC 与 TBA  $B_2$  的积自动机的理论的名称。该引理的证明与上面的安全性证明方法类似<sup>[19]</sup>,这里不再赘述。应用引理 TBA 理论中的引理 accept-empty 可验证活性:

liveness; THEOREM lv\_lang-empty。

## 6 相关工作

关于在 PVS 验证系统中建立时序逻辑的语法和语义形式规格说明的工作,S. Rajan 和 N. Shankar 等给出了模型检查和定量证明相结合的技术,给出了 CTL 在 PVS 中的  $\mu$ -演算理论和相关的引理,并在 PVS 证明器引入了 (bddsimp)、(model-check) 等命令支持有关公式的化简<sup>[15]</sup>。Paul. Gloess 中给出了 LTL 语言及某些性质的 PVS 规格说明<sup>[17]</sup>。T. Arons 和 A. Pnueli 给出了 LTL 在参数化公平系统 PFS(Parameterized Fair System) 框架中的 LTL 验证系统 TLPVS<sup>[16]</sup>,这是一个涉及众多演绎规则的、功能强大的验证系统。本文的工作受到了如上工作的启发,但上面这些系统都是建立在离散序列之上的时序逻辑,而本文给出的时序逻辑系统是带有稠密时间(时间域是非负实数集合)扩充的,解释在 TA 形式模型上的时间化的时序逻辑,并且时间化的线性时序逻辑 TLTL 用 TBA 来表示,使其验证变得简单易行。

**结论** 本文在时间自动机和时间 Büchi 自动机的基础上,建立了解释在 TA 上的、带有稠密时间扩充的线性分支时序逻辑的语法和语义模型,并在定理证明器 PVS 中实现解释在 TA 形式模型上的规格说明。本文还对一个建模为 TA

的经典的实时系统的特性进行了形式验证。下一步的工作应考虑对时间化的时序逻辑进行形式验证的通用证明规则,以使实时特性的形式验证可以较为容易地进行。

## 参考文献

- Alur R, Dill D L. A theory of timed automata. Theoretical Computer Science, 1994, 126: 183~235
- Clarke E, Grumberg O, Peled D. Model Checking. Cambridge: MIT PRESS, 1999
- Qingguo XU, MIAO Huaikou. Modeling Timed Automata Theory in PVS: [上海大学计算机学院内部技术报告]
- Owre S, Shankar N, Rushby J, et al. PVS system guide version 2.4. Computer Science Laboratory. SRI International, 2001. 1~97
- Owre S, Shankar N, Rushby J, et al. PVS language reference version 2.4. Computer Science Laboratory. SRI International, 2001. 1~102
- Alur R, Henzinger T A. Really temporal logic. Journal of the ACM, 1994, 41: 181~204
- Shankar N, Owre S, Rushby J, et al. PVS prover guide version 2.4. Computer Science Laboratory. SRI International, 2001. 1~127
- Alur R. Timed Automata. In: 11th International Conference on Computer-Aided Verification. LNCS 1633, Springer-Verlag, 1999. 8~22
- Archer M. TAME: Using PVS Strategies for Special-Purpose Theorem Proving. Annals of Mathematics and Artificial Intelligence, 2000, 29(1-4): 139~181
- Kolano P Z. Proof assistance for real-time systems using an interactive theorem prover. Theoretical Computer Science, 2002, 282(1): 53~99
- Hooman J. Timed Automata in PVS. Summer School Zhengzhou, 2004
- Owre S, Shankar N. Abstract Datatypes in PVS. Computer Science Laboratory, SRI International, 1997. 1~52
- Owre S, Shankar N. The PVS Prelude Library. Computer Science Laboratory, SRI International, 2003. 1~31
- Berard B, Bidoit M, Finkel A. System and Software verification Model-Checking Techniques and Tools. Springer-Verlag, 2001
- Rajan S, Shankar N, Srivas M K. An integration of model-checking with automated proof checking. In: Wolper P, ed. Computer Aided Verification, CAV '95 volume 939 of Lecture Notes in Computer Science, Liege, Belgium, Springer-Verlag, June 1995. 84~97
- Arons T, Pnueli A. TLPVS: A PVS-based LTL verification system, Verification: Theory and Practice, Essays Dedicated to Zohar Manna on the Occasion of his 64th Birthday, volume 2772 of Lecture Notes in Computer Science, Springer-Verlag, 2004. 598~625
- http://www.enseirb.fr/~gloess/pvsdir/pvs/temporal\_logics/
- Tripakis S. The analysis of timed systems in practice. Universite Joseph Fourier, Grenoble, France, December 1998
- XU Qingguo, MIAO Huaikou. Formal Verification Framework for safety of Real-time System Based-on Timed Automata Model in PVS. accepted by the IASTED International Conference on Software Engineering (SE 2006), Innsbruck, Austria, 2006

(上接第 216 页)

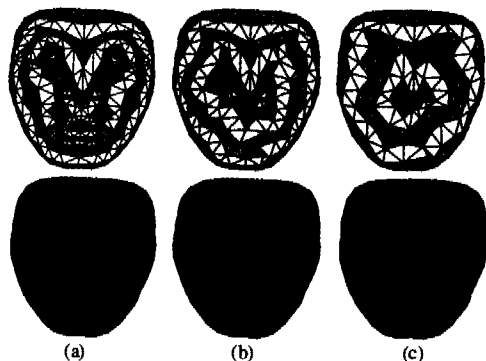


图 4 Nefertiti 模型的多分辨率表示  
 下为着色后的效果图

## 参考文献

- Hoppe H. Progressive meshes. In: Rushmeier H, ed. Proceedings of SIGGRAPH'96. New Orleans; ACM Press, 1996. 99~108
- Pajarola R, Rossignac J. Compressed Progressive Meshes. IEEE Transactions on Visualization and Computer Graphics, 2000, 6(1): 79~93
- Cohen-Or D, Levin D, Remez O. Progressive Compression of Arbitrary Triangular Meshes. In: Ebert D, Gross M, Hamman B, eds. Proceedings of IEEE Visualization '99, New Orleans; ACM Press, 1999. 67~72
- Alliez P, Desbrun M. Progressive Encoding for Lossless Transmission of 3D Meshes. In: Fiume E, ed. Proceedings of SIGGRAPH'01. New Orleans; ACM Press, 2001. 198~205
- Karni Z, Bogomjakov A, Gotsman C. Efficient Compression and Rendering of Multi-Resolution Meshes. In: Moorhead R, Gross M, Joy K I, eds. Proceedings of IEEE Visualization '02. New Zealand; IEEE Computer Society Press, 2002. 347~354
- Taubin G, Gueziec A, Horn W, et al. Progressive Forest Split Compression. In: Cohen M, ed. Proc. Proceedings of SIGGRAPH'98. New Orleans; ACM Press, 1998. 123~132