

# 一种新的三维模型水印嵌入空域算法

廖学良 王瑀屏

(清华大学计算机科学与技术系 北京 100084)

**摘 要** 提出一种三维模型的仿射变换不变量——重心交点距离比(RBC),并基于该不变量提出了一种在三维模型中嵌入水印的空域方法——MICT方法.在嵌入过程中,该方法首先算出模型的重心,然后利用加速的顶点和重心连线与模型求交的方法,求得连线与模型的交点,从而计算出重心交点距离比.通过修改一个三角形面片3个顶点对应的RBC,依次嵌入Mark、Index、Content字符.为增强水印的鲁棒性,将水印字符串进行了多次嵌入.在提取过程中,为了提高字符提取的准确性,采取了投票的方法,对水印字符串的每一个字符,在256个字符中进行投票,得票最多的作为正确的字符.该方法是盲提取方法,简单实用,从理论上能够抵抗仿射变换、顶点乱序以及一定程度的噪声攻击等常见的三维模型水印攻击,并且在实验上得到验证.另外文章还对此方法的嵌入容量进行了分析和实验.

**关键词** 三维模型;数字水印;仿射变换;空域算法;MICT方法

**中图法分类号** TP391

## A New Spatial Domain Method for Watermarking in 3D Models

LIAO Xue-Liang WANG Yu-Ping

(Department of Computer Science and Technology, Tsinghua University, Beijing 100084)

**Abstract** An invariable of affine transformation in 3D models, the Rate of Barycenter and Cross-point (RBC) is proposed, and by using the invariable, a spatial domain method for watermarking in 3D models, called MICT method, is proposed. In the embedding process, this method first calculate the barycenter of the model, then after getting the line through the vertex and barycenter, it adopts the fast ray tracing method and gets the crosspoint of the line and the model. By modifying the RBC of the vertices, the Mark, Index, Content characters are embedded. In order to improve the robustness of the method, the watermark string is embedded multi-times. In the extracting process, we adopt the voting method to improve the accuracy of the extracted watermark, that is voting from 256 characters. For every character of the watermark string, they set the most voted character in the 256 characters as the right character. This method is a blind method, simple and useful. Theoretically, the embedded watermark is unchanged under some normal operations to the model such as affine transformation, vertex reorder and so on, and it can also resist some noise attack. The experiments have proved that it is robust under these operations. They also analyze embedding capacity by experiments.

**Keywords** 3D models; digital watermark; affine transformation; spatial domain method; MICT method

## 1 引言

随着多媒体技术和互联网的飞速发展,数字产品在网络上的传播变得非常普及,因此也带来了数字产品版权的问题. 三维模型作为数字产品的一种重要类型,应用越来越广泛,同样也存在版权保护问题. 在数字产品版权问题出现的同时,很多研究者利用数字水印来保护版权,对于三维模型的数字水印嵌入,也成为多媒体信息安全领域的研究重点.

针对三维模型的水印算法和其它数字媒体的水印算法一样,也大致分为空域算法和变换域算法. 空域算法是指直接修改顶点的坐标、拓扑结构或者其它数据来嵌入水印的方法,而变换域算法一般都先对原模型做某种变换,通过修改变换域中的数据来嵌入水印,然后进行反变换得到嵌入水印后的模型. 本文提出的水印算法是一种新的空域算法.

文献[1]被公认为是三维模型数字水印领域发表的第一篇论文. 在这篇论文中,作者开创性地提出了两种空域嵌入方法:TSQ 方法和 TVR 方法. 在 TSQ(Triangle Similarity Quadruple)算法中,适当选取两个无量纲量可以定义一组相似三角形. 比如图 1(a)中所示的  $\{a/b, b/c\}$  或者  $\{\theta_1, \theta_2\}$ ,只要两个三角形的  $a/b$  和  $b/c$  相等,或者  $\theta_1$  和  $\theta_2$  相等,则这两个三角形是相似的. 图 1(b)所示的相邻的 4 个三角形叫做一个嵌入单元. 在原始模型中找到没有使用过的合适的嵌入单元(我们认为“合适”的意思是比如避免两个无量纲量都太小的情况),通过修改三角形的顶点,改变三角形的无量纲量来嵌入  $\{subscript, mark, data1, data2\}$ . 这里, *subscript* 表示水印字节的下标, *mark* 是一个特殊的值,表示这里嵌有水印, *data1* 和 *data2* 是嵌入的水印,一般共同表示 *subscript* 代表的字节. 重复在嵌入单元中嵌入水印字节的过程,就能够嵌入所有的水印字节.

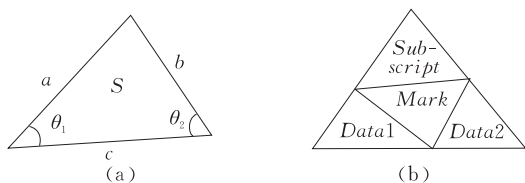


图 1 TSQ 方法示意图

在文献[2]中,作者提出了三种空域方法:VFA 方法、AIE 方法和 NBE 方法. 在 VFA 方法中,修改顶点的坐标,将水印 bit 嵌入到该顶点与一个指定的三角形(称为开始三角形, start triangle)的重心

的距离中. 该方法先根据到开始三角形重心的距离大小将所有的顶点分成  $N+1$  组,再将每一个组分成  $m$  个小组,这里  $m$  为嵌入的水印的 bit 数. 在 AIE 方法中,选择特定的两个共边的三角形作为 start triangles,并根据一定的规则选择和 start triangles 相邻的顶点集合  $G$ ,将  $G$  分成 4 个子集,每个子集嵌入 6bit 信息,这 6bit 信息包括标记是 4 个子集中的哪个子集的 2bit,另外还有 4bit 有用信息. 其嵌入方法利用了 Nielson 范数,修改顶点的坐标以达到修改顶点在 start triangles 的 4 个顶点坐标上的 Nielson 范数,从而嵌入水印.

文献[3]同样利用了 Nielson 范数来嵌入水印,并使用  $K$ -平均聚类法对 Nielson 范数进行聚类后,通过修改 Nielson 范数嵌入水印. 该方法在条件数、平滑、噪声等攻击下的比特错误率均低于 10%. 文献[4-6]利用了扩展高斯图像(EGI)和复扩展高斯图像(CEGI)来嵌入水印. 文献[7]通过修改顶点法向量的分布来嵌入水印,两种修改方法分别为修改法向量的均值和方差. 在顶点选取方法上,文献[8]中提出的算法从局部区域中选择一些对可见性影响不大的点,并使用两种限制修改机制:平行平面限制和包围椭球限制.

由上述分析可知,在空域水印算法中,一般都会先选择一个不变量作为嵌入位置,对这个量的选择成为水印嵌入效果的关键. 在这篇论文中,我们提出了一种新的仿射不变量——重心交点距离比,并利用这个仿射变换不变量来进行水印的嵌入.

本文第 2 节,我们将引入重心交点距离比的定义,并证明其仿射变换不变性,此外将介绍如何通过微调顶点坐标来修改该不变量的值;第 3 节介绍算法的原理,包括 MICT 方法的嵌入和提取算法;第 4 节给出 MICT 方法在鲁棒性和嵌入容量方面的实验结果;第 5 节总结全文.

## 2 一个新的仿射变换不变量及其修改方法

### 2.1 重心交点距离比(Ratio of Barycenter and Crosspoint, RBC)

首先,我们将给出重心交点距离比的定义.

**定义 1(重心交点距离比).** 如图 2 所示,设点  $A$  为三维模型表面上任意一点,  $O$  为该模型的重心. 通过点  $A$  和  $O$  的直线  $AO$  和三维模型交于点  $A'$ ,  $A'$  所在三角形面片的三个顶点分别为  $V_0, V_1, V_2$ . 设

$|\mathbf{AO}|=d_1$ ,  $|\mathbf{OA'}|=d_2$ , 则重心交点距离比定义为  $r(A)=d_1/d_2$ .

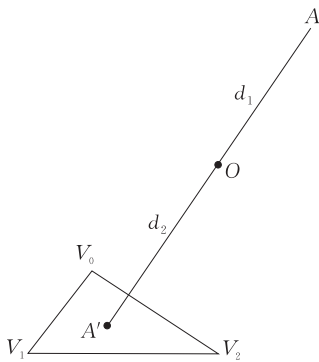


图 2 仿射变换不变量——重心交点距离比示意图

由定义我们可以证明, 在对整个模型做仿射变换的情况下, 比值  $r=d_1/d_2$  不变.

证明. 设点  $A$  的坐标为  $\mathbf{A}=[x_A\ y_A\ z_A]^T$ , 点  $O$  的坐标为  $\mathbf{O}=[x_O\ y_O\ z_O]^T$ , 点  $V_0, V_1, V_2$  的坐标为  $\mathbf{V}_0=[x_{V_0}\ y_{V_0}\ z_{V_0}]^T$ ,  $\mathbf{V}_1=[x_{V_1}\ y_{V_1}\ z_{V_1}]^T$ ,  $\mathbf{V}_2=[x_{V_2}\ y_{V_2}\ z_{V_2}]^T$ , 点  $A'$  的坐标为  $\mathbf{A'}=[x_{A'}\ y_{A'}\ z_{A'}]^T$ . 则

根据面  $V_0V_1V_2$  的方程有

$$|\mathbf{A'}-\mathbf{V}_0\ \mathbf{V}_1-\mathbf{V}_0\ \mathbf{V}_2-\mathbf{V}_0|=0.$$

根据直线  $\mathbf{AOA'}$  的方程, 有

$$\mathbf{A'}=k(\mathbf{O}-\mathbf{A})+\mathbf{O}.$$

设仿射变换的方程为  $\mathbf{V'}=\mathbf{TV}+\mathbf{b}$ , 其中  $\mathbf{T}$  为  $3\times 3$  的矩阵. 则

$$\begin{aligned} &|\mathbf{A'}-\mathbf{V}_0\ \mathbf{V}_1-\mathbf{V}_0\ \mathbf{V}_2-\mathbf{V}_0|=0 \\ \Rightarrow &|\mathbf{T}(\mathbf{A'}-\mathbf{V}_0\ \mathbf{V}_1-\mathbf{V}_0\ \mathbf{V}_2-\mathbf{V}_0)|=0 \\ \Rightarrow &|\mathbf{TA'}-\mathbf{TV}_0\ \mathbf{TV}_1-\mathbf{TV}_0\ \mathbf{TV}_2-\mathbf{TV}_0|=0 \\ \Rightarrow &|\mathbf{TA'}+\mathbf{b}-(\mathbf{TV}_0+\mathbf{b})\ \mathbf{TV}_0+\mathbf{b}-(\mathbf{TV}_0+\mathbf{b})\ \mathbf{TV}_2+\mathbf{b}-(\mathbf{TV}_0+\mathbf{b})|=0. \end{aligned}$$

可见,  $\mathbf{TA'}+\mathbf{b}$  和  $\mathbf{TV}_0+\mathbf{b}, \mathbf{TV}_1+\mathbf{b}, \mathbf{TV}_2+\mathbf{b}$  在同一平面上. 另外,

$$\begin{aligned} \mathbf{A'}&=k(\mathbf{O}-\mathbf{A})+\mathbf{O} \\ \Rightarrow \mathbf{TA'}&=k(\mathbf{TO}-\mathbf{TA})+\mathbf{TO} \\ \Rightarrow \mathbf{TA'}+\mathbf{b}&=k(\mathbf{TO}+\mathbf{b}-(\mathbf{TA}+\mathbf{b}))+\mathbf{TO}+\mathbf{b}. \end{aligned}$$

上式说明仿射变换后的点  $A'$  还位于直线  $\mathbf{AO}$  上. 从而, 仿射变换后直线  $\mathbf{AO}$  和面  $V_0V_1V_2$  的交点就是仿射变换前的点  $A'$  在同样的仿射变换下得到的点.

$$r=\frac{\mathbf{AO}}{\mathbf{OA'}}=\frac{\mathbf{O}-\mathbf{A}}{\mathbf{A'}-\mathbf{O}}=\frac{\mathbf{TO}-\mathbf{TA}}{\mathbf{TA'}-\mathbf{TO}}=\frac{\mathbf{TO}+\mathbf{b}-(\mathbf{TA}+\mathbf{b})}{\mathbf{TA'}+\mathbf{b}-(\mathbf{TO}+\mathbf{b})}=\text{变换后的 } r.$$

可见, 重心交点距离比  $r$  在仿射变换下是不变的. 证毕.

2.2 重心交点距离比的修改

作为一种空域水印算法, 我们通过修改顶点坐标的方法来嵌入水印. 为了利用顶点的重心交点距离比的仿射变换不变性, 在修改顶点坐标时, 只允许顶点在顶点与重心连线上移动. 比如图 3, 点  $A$  的移动只在  $A$  与  $O$  所在的直线上移动.

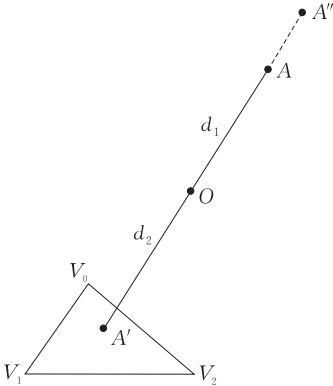


图 3 修改重心交点距离比的方法

如图 3 所示, 设点  $A$  的坐标为  $\mathbf{A}=[x_A\ y_A\ z_A]^T$ , 重心  $O$  的坐标为  $\mathbf{O}=[x_O\ y_O\ z_O]^T$ , 点  $A'$  的坐标为  $\mathbf{A'}=[x_{A'}\ y_{A'}\ z_{A'}]^T$ .

这样我们可以计算出  $d_1$  和  $d_2$  如下:

$$d_1=\sqrt{(x_O-x_A)^2+(y_O-y_A)^2+(z_O-z_A)^2}\tag{1}$$

$$d_2=\sqrt{(x_{A'}-x_O)^2+(y_{A'}-y_O)^2+(z_{A'}-z_O)^2}\tag{2}$$

根据重心距离比的定义, 我们可以得出  $r(A)$ .

为了达到嵌入水印的目的, 我们修改  $r(A)$ , 先将  $r(A)$  用一个 IEEE 标准下的双精度浮点数进行表示, 然后根据要嵌入的字符, 修改这个双精度浮点数的尾数的某些位(我们的方法是每次嵌入一个字节, 即修改 8 位), 使之变成  $r'(A)$ , 并按照如下方程修改  $A$  的坐标为

$$\mathbf{A''}=(x_{A''}, y_{A''}, z_{A''})=(\mathbf{A}-\mathbf{O})\cdot\frac{r'(A)}{r(A)}+\mathbf{A}\tag{3}$$

从式(3)可知,  $A''$  位于直线  $\mathbf{AO}$  上.

3 水印算法

3.1 算法概述

利用第 2 节介绍的重心交点距离比, 我们使用  $Mark, Index, Content$  一起嵌入的方法(简称 MICT 方法)来嵌入水印.

这里,  $Mark, Index, Content$  都是用字节表示的

数据. 字节 *Mark* 是一个嵌入标记, *Mark* 为某一事先设定的值时, 就表示可能在这里嵌入了水印; 字节 *Index* 表示嵌入的 *Content* 的是水印的第几个字节, 比如当 *Index* 为 3 时, 表示 *Content* 是水印的第 3 个字节; 字节 *Content* 表示水印内容的第 *Index* 个字节.

如图 4 所示, 三角形 *ABC* 为模型上的一个三角形, *O* 是该模型的重心. 直线 *AO*, *BO*, *CO* 分别交模型表面于点 *A'*, *B'*, *C'*. 利用 2.2 节介绍的嵌入方法, 修改  $r(A)$  使之嵌入 *Mark*, 表示这里嵌有水印; 修改  $r(B)$  使之嵌入 *Index*, 表示这里嵌入的是水印的哪个字节; 修改  $r(C)$  使之嵌入 *Content*, 表示这个字节的内容. 点 *A*, *B*, *C* 和点 *A'*, *B'*, *C'* 所在面作为一个嵌入单元. 基于上述对一个嵌入单元的基本操作, 下面对水印的嵌入和提取算法做详细的说明.

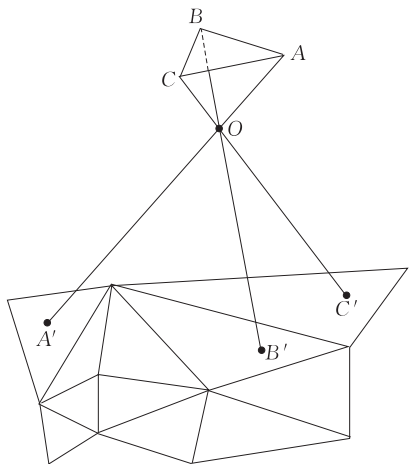


图 4 MICT 方法示意图

### 3.2 水印嵌入算法

我们的算法是根据 3.1 节中所述的对一个嵌入单元同时嵌入 *Mark*, *Index*, *Content* 的方法, 对模型的面片进行遍历, 以寻找这样的嵌入单元. 每找到一个符合条件的嵌入单元, 就嵌入一个字节. 这样, 为了大量嵌入水印, 需要对模型中的大部分顶点所对应的重心交点距离比进行修改, 而大量的修改会导致两个问题: 重心交点距离比的二次修改问题和重心偏移较大问题. 针对这两个问题, 我们提出了完整的解决方案. 此外, 为了增强水印的鲁棒性, 我们对水印字符串嵌入多个拷贝.

#### 3.2.1 重心交点距离比的二次修改问题及解决

**问题 1.** 每个顶点通过与重心连线对应一个三角形面片, 如果这个面片上的顶点位置发生变化, 原顶点与重心连线在该面片上的交点位置就会发生变化, 甚至可能出现不交于该面片的情况. 因此每修改

一个顶点的位置后, 其对应的三角形面片的三个顶点位置均不可进行修改.

为了解决问题 1, 对每一个顶点, 我们都定义一个叫作 *used* 的 bool 类型的属性, 以防止对该点的重心交点距离比进行二次修改. 在嵌入水印之前, 设置所有点的 *used* 属性为 false. 如图 3 所示, 只有点 *A* 满足如下的修改条件时, 才修改 *A* 的重心交点距离比.

**修改条件.** *A* 的 *used* 属性为 false.

在  $r(A)$  上嵌入水印之后, 设置点 *A*,  $V_0$ ,  $V_1$ ,  $V_2$  的 *used* 属性为 true. 这样, 在之后的嵌入过程中, 就不会对这 4 个点的坐标进行修改. 从而只要重心不变, 就不会改变嵌了水印后的  $r(A)$  值.

#### 3.2.2 重心偏移较大问题及解决

**问题 2.** 单独修改一个顶点的位置对重心位置的影响较小, 大量顶点位置的修改则对重心会有不可忽略的影响, 而重心位置的变化将对重心交点距离比产生重要的影响.

设顶点嵌入水印后的坐标平均改变量为  $\Delta$ , 嵌入的水印字节数为  $n$ , 则重心的最大改变量为  $\Delta_o = \frac{\Delta \cdot n}{N}$ , 这里  $N$  为模型的顶点数. 而  $n$  的上限为  $N/3$  (详细说明见 4.2.1 节), 所以  $\Delta_o$  的最大值为  $\Delta/3$ . 这个改变量是比较大的, 所以如果不修正重心, 则不能提取到正确的水印.

由于问题 2 的存在, 以及我们的提取算法是依赖于重心的, 所以在嵌入水印之后, 应该对重心进行修正.

修正重心的方法是消除嵌入水印之后的重心偏移量, 也就是将重心偏移量平均到那些没有使用的点 (即 *used* 属性为 false 的点) 上. 具体的算法如下:

1. 计算嵌入水印前后的重心偏移量  $\Delta_o$ ;
2. 统计嵌入水印之后所有 *used* 为 false 的顶点个数  $Num$ ;
3. 每个 *used* 为 false 的顶点  $V$  的坐标进行修改. 设  $V$  原来的坐标为  $\mathbf{V} = (x_V, y_V, z_V)$ , 修改之后的  $V$  的坐标为

$$\mathbf{V}' = \mathbf{V} + \Delta_o \cdot \frac{Num}{N} \quad (4)$$

这里  $N$  为模型的顶点个数.

#### 3.2.3 增加对噪声攻击的鲁棒性

鲁棒性对于水印来说是一个非常重要的效果衡量标准. 在鲁棒性方面, 由于水印攻击方法的多样性, 水印的嵌入和提取都只能针对某一些攻击, 即只能在某些固定的攻击下具有较好的鲁棒性. 我们的方法对仿射变换和顶点乱序等攻击, 具有很好的性

能.从理论上来说,该方法在遭受任意仿射变换和顶点乱序之后,仍然能够准确地提取到嵌入的水印.

为了加强 MICT 方法对噪声攻击的鲁棒性,我们在嵌入时采用多次嵌入的方法,在提取时采用投票的方法.假设水印字符串的长度为  $L$ ,而利用 MICT 方法可以嵌入的字符数会远远大于  $L$ ,因此,我们将水印字符串重复嵌入多个拷贝,以加强水印的鲁棒性.

在嵌入时,我们使用了两个计数器,一个是  $Index$ ,用来标记要嵌入水印字符串中的是哪一个字符,另一个是  $m$ ,用来标记水印字符串重复嵌入了多少拷贝. $Index$  初始化为 0,当  $Index$  达到  $L-1$  时,就将其置 0,即重新从第 0 个字符开始嵌入,并将  $m$  增加 1,表示字符串嵌入了  $m$  遍.

在提取时,采用投票的方法.根据  $Index$  可以得知该字符是第几个字符,对每个  $Index$ ,选择票数最多的字符作为水印字符.

通过多次重复嵌入的方法,可以有效地增强对噪声等攻击的鲁棒性(关于鲁棒性的实验结果见第 4 节).

3.2.4 嵌入流程

在嵌入水印时,我们首先对每个顶点的  $used$  属性设置为 false,然后查找符合 3.2.1 节所述修改条件而且又位于同一个三角形面片的顶点  $A, B, C$ ,按照 3.1 节所述的 MICT 方法,嵌入  $Mark, Index, Content$ .

具体的嵌入算法如下:

1. 采用加速的直线与模型求交方法,获得每个顶点对应的面片,并计算每个顶点的重心交点距离比  $r$ ;
2. 设置每个顶点的  $used$  属性为 false;
3. 设置  $m$  为嵌入水印的拷贝数,初始为 0,  $Index=0, L$  为水印的长度;
4. 对模型中的每个面片,
  - 4.1. 如果该面片的 3 个顶点满足修改条件,则继续;否则转步 4,遍历下一个面片;
  - 4.2. 按照 3.1 小节中给出的方法,在该面片的顶点  $A, B, C$  处分别嵌入  $Mark, Index$ , 水印的第  $Index$  个字符;
  - 4.3.  $Index$  累加 1,指向下一个字节的水印;如果  $Index=L$ ,说明已经嵌入一个完整的水印拷贝,  $Index$  归 0, 拷贝计数  $m$  加 1;
  - 4.4. 设置该面片的 3 个顶点以及这 3 个顶点的对应面的 3 个顶点的  $used$  属性为 true;
5. 利用 3.2.2 小节中给出的方法修正重心.

从上面的过程可以看出,算法结束后,一共嵌入的字节数是  $m \times L + Index$  个,也就是将水印字符串嵌入了  $m$  个拷贝.

3.3 水印提取算法

在提取水印时,我们先找到重心交点距离比为  $Mark$  的顶点所在的三角形,在这些三角形上可能嵌入了水印,然后提取  $Index$  和  $Content$ . 遍历所有的三角形之后,由投票算法决定水印字符串的每个字符.

具体的水印提取算法如下:

1. 采用加速的直线与模型求交方法,获得每个顶点对应的面片,并计算每个顶点的重心交点距离比  $r$ ;
2. 对水印的每一个字符,定义一个长度为 256 的整形数组  $vote$ ,记录投票信息,并初始化为 0;
3. 对模型中的每个面片(其顶点分别为  $A, B, C$ ):
  - 3.1. 从顶点  $A$  中提取一个字节的值,如果该字节为  $Mark$ ,则表示可能嵌入水印;否则转步 3,遍历下一个面片;
  - 3.2. 从顶点  $B$  中提取出  $Index$ ,从顶点  $C$  中提取出  $Content$ ;
  - 3.3. 将第  $Index$  个字符的  $vote$  数组的第  $Content$  项计数加 1(由于  $Content$  的长度为 1byte,只有 256 种可能性,这就是  $vote$  数组的长度设置为 256 的原因);
  - 3.4. 对水印的每一个字符,取其相应的  $vote$  数组中 256 项计数值的最大值的下标,作为水印的该字符的内容.

4 实验结果

4.1 不可见性和鲁棒性实验

4.1.1 水印的不可见性

目前,三维模型的水印仍没有公认的不可见性评价标准,但一般的评价方法有两种:观察比较法和 Hausdorff 距离法<sup>[17]</sup>.前者以肉眼无法识别或不易识别嵌入前后的差异为评价标准,后者则定量计算嵌入前后两个模型的 Hausdorff 距离,距离越小,不可见性越好.下面就分别使用这两种方法证明 MICT 方法的不可见性.

使用观察比较法,图 5(a)是 chetah(5000 顶点)原始模型,图 5(b)是嵌入了水印“Tsinghua University Computer Science”后的模型.通过观察对比我们可以看出,嵌入水印之后的模型和原模型的差别用肉眼是无法识别的.

使用 Hasudorff 距离法,几个模型在嵌入前后的 Hausdroff 距离的计算结果如表 1 所示.



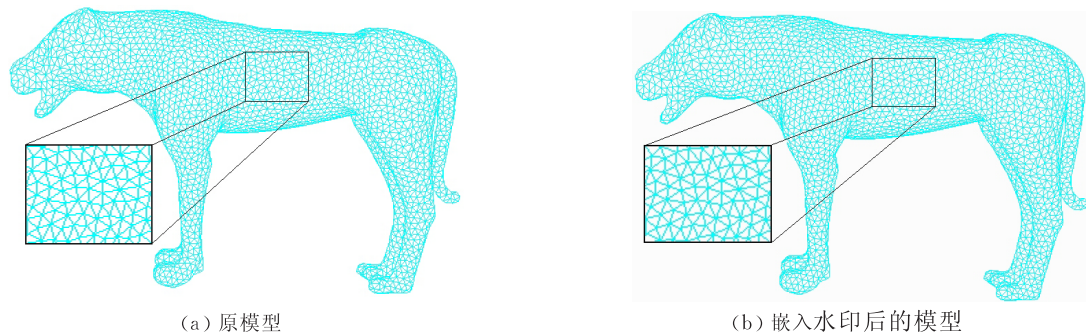


图 5 原模型和嵌入水印后模型的对比

表 1 嵌入水印前后对比

模型	Haus(1E-3)	Mean(1E-3)
bunny	13.5	0.26
cheta h	7.5	0.10
dino	5.3	0.10
hema	4.8	0.12
horse	7.7	0.12

表 1 中, Haus 表示 Hausdorff 距离, Mean 表示顶点距离的平均值. 可以看到嵌入前后两模型的

Hausdorff 距离值较小.

综合以上两点, 我们可以认为, MICT 方法嵌入水印的不可见性非常好.

4. 1. 2 抗仿射变换

图 6(a)~(c)是我们在嵌入了水印“Tsinghua University Computer Science”后的模型分别做缩放、旋转、一般仿射变换攻击之后得到的模型. 在做了这些攻击之后, 我们仍然能够提取到嵌入的水印.

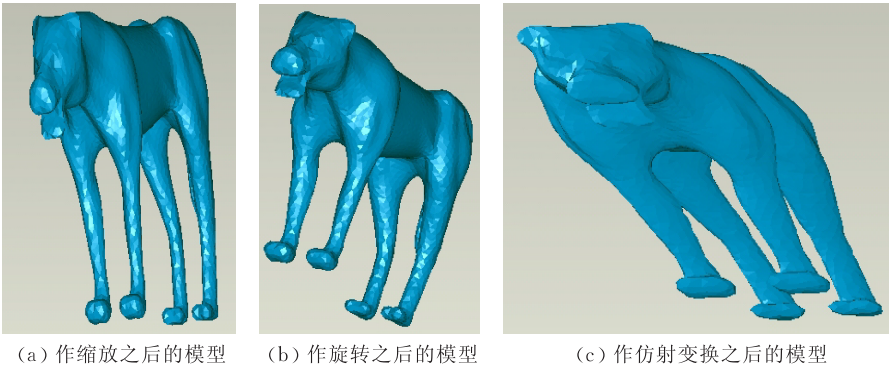


图 6 作仿射变换后的模型

4. 1. 3 抗顶点乱序

由于 MICT 方法提取水印时并不依靠顶点的顺序, 所以我们的方法是能够抵抗顶点乱序攻击的. 图 7(a)是嵌入了水印“Tsinghua University Com-

puter Science”之后的模型, 图 7(b)是对图 7(a)所示的模型做顶点乱序之后的模型. 实验证明, 我们在顶点乱序之后的模型中顺利地提取到了水印“Tsinghua University Computer Science”.

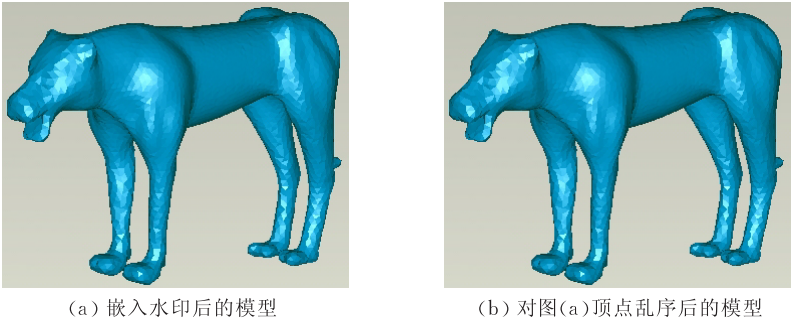


图 7 嵌入水印后的模型和顶点乱序后的嵌入水印后的模型

4. 1. 4 抗噪声攻击

我们测试了 5 个模型在噪声攻击下的鲁棒性,

采用的鲁棒性指标是比特错误率. 比特错误率指的是提取的水印和实际水印相比, 不同的比特在实际

水印中所占的比例. 在 MICT 方法中嵌入的是 ASCII 字符,但是这丝毫不影响我们用比特错误率来评测噪声攻击下的鲁棒性. 我们对模型采用的攻击是均匀分布的随机噪声攻击,并控制噪声大小在一定的范围内,使噪声不致引起模型的较大变化. 我们对每个模型攻击和提取水印 50 次,取这 50 次提取的 BER 的平均值为最终结果,见表 2.

表 2 噪声攻击下的比特错误率

模型	顶点数	面数	比特错误率/%
bunny	5000	9906	2.22
cheta h	5000	9966	6.39
dino	5032	10043	6.71
hema	5033	10062	4.25
horse	5014	10024	5.24

由表 2 可以看出,在加入一定的噪声之后,比特错误率都在 10% 以下,因此,可以认为,我们的方法具有抗一定程度噪声攻击的能力.

4.2 嵌入容量的实验

4.2.1 容量的理论分析

为分析嵌入容量,我们首先观察一个嵌入单元所包含的顶点个数. 从图 4 可以看出,每嵌入一个字符,△ABC 的 3 个顶点是包含在嵌入单元中的,另

外点 A',B',C'可能位于同一个三角形上,也可能位于 2 个或者 3 个三角形上. 此外,考虑到这些三角形的相邻情况,点 A',B',C'所在的三角形所包含的顶点个数范围是 3~9,所以每个嵌入单元所包含的顶点数为 6~12 个. 由于△ABC,点 A',B',C'所在的三角形仍然可以被其它的顶点当作对应面,即某一嵌入单元中的顶点仍然可以包含在其它的嵌入单元中,所以平均每个嵌入单元所用到的字节为 3~12 个. 在每个这样的嵌入单元中,算法嵌入了一个字节的水印信息.

假设模型的顶点个数为 N,那么模型能够嵌入的水印字符数 n 有一个上限: N/3. 当然,在实际的嵌入过程中,很难达到这个上限.

4.2.2 实验结果

对于表示同一个物体的模型来说,顶点个数越多,能够嵌入的水印就越长. 我们对不同顶点数的球模型进行实验. 图 8 所示的 5 个模型,是顶点个数分别为 111,300,500,1000,2000 的球模型. 我们在每个模型上进行嵌入容量的实验,得到的实验结果如图 8(括号内的数字表示嵌入的水印字节数)所示.

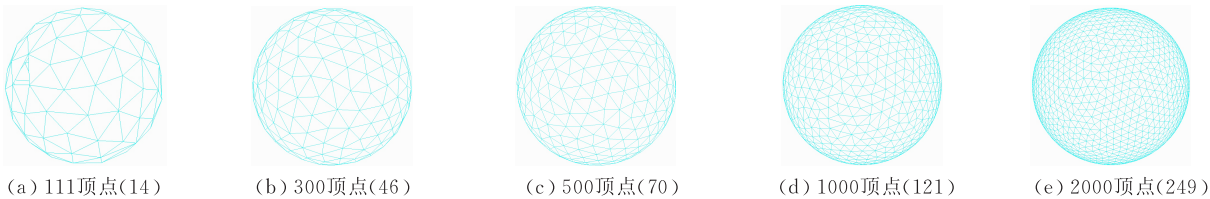


图 8 不同顶点数的球模型

为了考查顶点个数和嵌入字节数的关系,我们将上述结果在图 9 中表示.

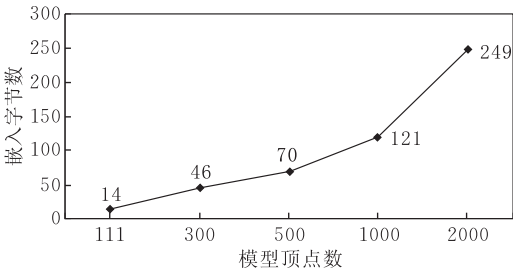


图 9 球模型的顶点个数和嵌入容量的关系图

从图 9 可以看出,对于都表示一个球的 5 个模型来说,嵌入容量是随着模型的顶点个数的增加而增加的,而且基本上是直线关系. 嵌入水印的长度约为顶点个数的 1/8,与顶点个数的 1/3 这一上限相差很远. 而且,对于不同的模型,嵌入容量也相差很大. 例如,对于顶点数为 2000 的球模型来说,能嵌入

的水印长度为 249 字节,而对于顶点数为 5000 的 cheta h 模型来说,嵌入的容量可达到 1124 字节,接近顶点数的 1/4.

4.3 结果评价及比较

由以上的实验结果可以看出,我们的方法能够抵抗仿射变换、顶点乱序攻击,并能够抵抗一定程度的噪声攻击. MICT 法和文献[3]比较,都能够抵抗仿射变换、顶点重排,并且在噪声攻击下的比特错误率也在 10% 以下,抵抗噪声能力相当. 和文献[6]中提出的方法相比,MICT 法虽然不能处理网格简化和剪切攻击,但是能很好地抵抗仿射攻击,对噪声的抵抗能力也相当. 和文献[7]中的方法相比,MICT 具有抗仿射变化的优势,虽然在噪声方面的数据不如文献[7],但是这和噪声大小很有关系,文献[7]中的噪声大小最大为 0.5%,而我们的实验噪声大小都在 1% 以上.

## 5 结 论

本文利用重心交点距离比的仿射变换不变性和 MICT 方法在三维模型中嵌入水印,使三维模型能够抵抗仿射变换以及顶点乱序的攻击,实验结果显示该方法能够很好地抵抗这些攻击.对于噪声攻击,我们的方法也能进行一定程度的抵抗.此外,本文对 MICT 方法的嵌入容量也进行了理论分析和实验验证.

## 参 考 文 献

- [1] Ohbuchi R, Masuda H, Aono M. Embedding data in 3D models//Proceedings of the European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services'97. Darmstadt, 1997: 1-10
- [2] Benedens O, Busch C. Towards blind detection of robust watermarks in polygonal models//Proceedings of the Eurographics. Interlaken, 2000: C199-C208
- [3] Jin J, Wang X. A novel 3D mesh blind watermarking algorithm based on nielson affine invariant norm//Proceedings of the CAD/Graphics. Beijing, 2007: 294-297
- [4] Benedens O. Geometry-based watermarking of 3D models. IEEE Computer Graphics and Applications, 1999, 19(1): 46-55
- [5] Benedens O. Two high capacity methods for embedding putlic watermarks into 3D polygonal meshes//Proceedings of the SPIE: Security and Watermarking of Multimedia Contents, 1999: 329-340
- [6] Lee S, Kwon K. A watermarking for 3D mesh using the path CEGIs. Digital Signal Processing, 2007, 17(2): 396-413
- [7] Cho J, Prost R et al. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. IEEE Transactions on Signal Processing, 2007, 55(1): 142-155
- [8] Bors A. Watermarking mesh-based representations of 3-D objects using local moments. IEEE Transactions on Image Processing, 2006, 15(3): 687-701

- [9] Kanai S, Date H, Kishinami T. Digital watermarking for 3D polygons using multiresolution wavelet decomposition//Proceedings of the International Workshop on Geometric Modeling. Tokyo, 1998: 296-307
- [10] Uccheddu F, Corsini M, Barni M. Wavelet-based blind watermarking of 3D models//Proceedings of the 2004 Multimedia and Security Workshop. Magdeburg, 2004: 143-154
- [11] Jin J, Dai M, Bao H et al. Watermarking on 3D mesh based on spherical wavelet transform. Journal of Zhejiang University: Science, 2004, 5(3): 251-258
- [12] Zhou K, Bao H, Shi J. 3D surface filtering using spherical harmonics. Computer-Aided Design, 2004, 36(4): 363-375
- [13] Ohbuchi R, Takahashi S, Miyazawa T et al. Watermarking 3D polygonal meshes in the mesh spectral domain//Proceedings of the Graphics Interface. Ottawa, 2001: 9-17
- [14] Chen Yang, Medioni Gerard. Object modeling by registration of multiple range images//Proceedings of the 1991 IEEE International Conference on Robotics and Automation Sacramento. California, 1991: 2724-2729
- [15] Yu Zhi-Qiang, Ip Horace H S, Kwok L F. A robust watermarking scheme for 3D triangular mesh models. Pattern Recognition, 2003, 36(11): 2603-2614
- [16] Pottmann Helmut, Huang Qi-Xing, Yang Yong-Liang, KÄolpl Stefan. Integral invariants for robust geometry processing. Geometry Preprint Series, Vienna University of Technology: Technical Report, 2005
- [17] Cignoni P, Rocchini C, Scopigno R. Metro: Measuring error on simplified surfaces. Computer Graphics Forum, 1998, 17(2): 167-174
- [18] Pan Zhi-Geng, Sun Shu-Sen, Li Li. An overview of 3D model watermarking. Journal of Computer-Aided Design & Computer Graphic, 2006, 18(8): 1103-1110(in Chinese)  
(潘志庚,孙树森,李黎. 三维模型数字水印综述. 计算机辅助设计与图形学学报, 2006, 18(8): 1103-1110)
- [19] Zhou Xin. Research on digital watermarking techniques and algorithms of three-dimensional models [Ph. D. dissertation]. Hangzhou: School of Information, Zhejiang University, 2002 (in Chinese)  
(周昕. 三维几何模型数字水印技术及算法研究[博士学位论文]. 杭州: 浙江大学信息学院, 2002)



**LIAO Xue-Liang**, born in 1984, M. S. candidate. His current research interests include content security, computer graphics.

**WANG Yu-Ping**, born in 1984, Ph. D. candidate. His current research interests include content security, computer graphics, digital geometry processing.



Background

As the multimedia technology and internet develops very fast, more and more digital products are used over the internet, then the copyright of these products appears. As an important, 3D models begin prevailing recently. Digital watermarking has been studied over many years for the copyright protection of digital products, and ten or more years ago, most attention was paid on the audio, image, video. In 1997, the first article on 3D model watermarking was published by Ohcuchi. After then, a lot of algorithms were developed by more and more researchers.

Generally speaking, there are two categories of algorithms for watermarking: spatial domain methods and frequency domain methods. Spatial domain methods embed watermark by modifying the coordinate of vertices, the color of texture points, the topology of the model, or other elements representing the model. The frequency domain methods first make transformation to the model and get the coefficients, then modify the coefficients to embed the watermark.

In order to extract the watermark, the extracting algorithm should be designed as the embedding algorithm is designed, that is, a watermark system includes an embedding subsystem and an extracting subsystem corresponding to the embedding subsystem. Some extracting algorithms need the

original model, they are called blind algorithm, and the other algorithms, which don't need the original model, are called non-blind algorithm.

There is no authoritative standard to evaluate which algorithm is better. For a given application, we can consider the following properties: (1) Robustness: The watermark should withstand different types of attacks; (2) Invisibility: The model should be modified just a little bit so that the embedded model be similar with the origin one; (3) Validation: The watermark should be extracted entirely; (4) Capacity: The mount of information that can embedded in the model.

In this paper, the authors propose an invariable of affine transformation: The Rate of Barycenter and Crosspoint (RBC). Based on this invariable, they propose a new spatial domain method, and it is also a blind algorithm. In the embedding subsystem, the authors calculate the barycenter of the model, then the RBC of every point. By modifying the RBC of the vertices, they embed the information. In the extracting subsystem, they adopt the voting method to improve the accuracy of the extracted watermark. Theoretically, the embedded watermark is unchanged under some normal operations to the model such as affine transformation, vertex reorder and so on, and it can also resist some noise attack.