不可逆的 3 维网格模型数字水印算法

彭 伟[1),2),3)] 纪庆革[1),2),3)] 牟 宁[1),2),3)] 李桂清[4)]

[1)](中山大学信息科学与技术学院计算机科学系,广州 510275) [2)](中山大学信息安全技术广东省重点实验室,广州 510275)

[3)](中山大学数字家庭教育部重点实验室,广州 510275) [4)](华南理工大学计算机学院,广州 510640)

摘 要 提出一种基于奇异值分解的三角网格模型数字水印算法,该算法主要思想在于把水印信息嵌入到奇异值为零的地方,由于零值的特点使得算法不可逆,从而成功抵御了解释攻击。该算法简单,并可以快速嵌入到大规模三角网格模型里。通过几种不同的攻击实验,验证了算法具有很好的鲁棒性,最后还对构造不可逆算法的必要条件进行了总结。

关键词 3 维网格模型 数字水印 解释攻击 不可逆算法

中图法分类号: TP309 **文献标识码**: A **文章编号**: 1006-8961(2009)07-1418-08

Non-invertible Watermarking Scheme for 3D Meshes

PENG Wei[1),2),3)], JI Qing-ge[1),2),3)], MOU Ning[1),2),3)], LI Gui-qing[4)]

[1)](*School of Information Science and Technology, Sun Yat-sen University, Guangzhou* 510275)

[2)](*Guangdong Key Laboratory of Information Security Technology, Sun Yat-sen University, Guangzhou* 510275)

[3)](*Key Laboratory of Digital Life(Sun Yat-sen University), Ministry of Education, Guangzhou* 510275)

[4)](*School of Computer Science & Engineering, South China University of Tech, Guangzhou* 510640)

Abstract This paper presents a novel watermarking scheme based on singular value decomposition(SVD) for 3D meshes, it embeds the watermark into the singular values which equal zero. This watermarking scheme is non-invertible, since the singular value is zero. This algorithm can quickly embed the watermark to large-scale 3D meshes and is robust to various attacks. Finally we analyze some necessary conditions to construct the non-invertible scheme.

Keywords 3D meshes, digital watermark, interpretation attack, non-invertible algorithm

1 引 言

目前,越来越多的三角网格水印算法被提出,但是大都着重于数字水印的鲁棒性,对于算法是否真正能够证明水印的拥有权却很少提及。大部分的算法[1-4],在验证三角网格中存在水印的方法都是采用了一个相关系数(提取的水印与嵌入的水印的相似度)来实现,只要相关系数大于某一个阈值,那么就说证明了网格的归属权。但是这样是不够的,Craver 等人提出了解释攻击[5-6],该攻击通过对嵌入了水印信息的作品进行分析,伪造一个虚假的原始作品,而这个虚假的原始作品也能从嵌入了水印信息的作品和原始作品中检测出另一个水印信息(这在有一定鲁棒性的算法中是完全可以实现的),从而使得原作者失去了作品的拥有权。为了抵御这种攻击,Craver 提出了可逆数字水印算法和半可逆数字水印算法的定义[6],并认为只有不可逆的算法才可以成功抵御解释攻击。

基金项目:NSFC-广东省联合基金项目(U0735001);国家自然科学基金项目(60473109)

收稿日期:2008-09-07;**改回日期**:2008-10-31

第一作者简介:彭 伟(1984 ~),男。中山大学计算机软件与理论专业硕士研究生。主要研究领域为计算机图形学。E-mail:pw_windgod@163.com

通讯作者:纪庆革,E-mail:issjqg@mail.sysu.edu.cn

这方面的工作主要出现在图像数字水印方面,如文献[7]。而在三角网格模型数字水印方面的工作很少,主要还是在算法的鲁棒性方面着手。但是如果失去了安全性,鲁棒性也就失去了意义。而且解释攻击之所以可行,在很大程度上还是因为有鲁棒性的存在,因此很有必要重新审视鲁棒性。

本文的算法基于奇异值分解[8]。基于奇异值分解的算法[9]直接把水印信息嵌入到奇异值上,然后再利用嵌入水印后的奇异值重构原来网格,其算法是可逆的,在遭到解释攻击情况下,网格的归属权也就无法证明[5]。本文提出了一种不可逆的水印算法,通过把水印信息嵌入到奇异值为零的地方,使得攻击者无法从水印网格重构一个属于他的"原始网格",从而成功抵御了解释攻击。

自 Ohbuchi 在 1997 年提出了把水印信息嵌入到 3 维模型的概念后[10],3 维模型数字水印技术发展很快。最初的工作是直接把水印信息嵌入到 3 维模型的顶点[11]或法向[12]等几何信息上。但是这类方法不能抵御加噪、光顺等常见攻击。

Kanai 于 1998 年提出了基于小波变换的三角网格水印算法,把水印信息嵌入到小波系数上,然后进行小波逆变换来重构水印网格。这种把水印嵌入到频谱域上的方法能够提高算法的鲁棒性,但是该算法不能抵御改变了网格拓扑结构的攻击。Praun 也于 1999 年提出了一个鲁棒的频谱域水印算法[3],通过构造一组标量函数把网格从空间域转化为频谱域上再嵌入水印信息。为了增加算法的鲁棒性,在水印提取过程中还特别加进了网格的重采样步骤,以抵御那些修改网格拓扑结构的攻击。Ohbuchi 随后也提出了自己的频谱域上的鲁棒水印算法[1-2]。Wu 和 Kobbelt 于 2005 年提出了一个快速的频域上的水印算法,该算法通过奇异值分解求出用来扩频的标准正交基,由于它只取一部分标准正交基用来分析网格的频域,所以算法很快,而且能够处理顶点数非常多的网格[4]。

近年来,还出现了很多新的水印算法:例如 Liu 和 Yang 的基于特征点的水印算法[13],通过选择特征点来构造距离图像(range image),然后对距离图像采用成熟的图像水印算法来嵌入水印信息。由于特征点能够在网格简化中保留下来,所以该算法能够抵御网格简化和网格重采样攻击。

以上提到的算法都是可逆的,因为他们都满足可逆的 3 个条件,即很容易在解释攻击下失效。尽管在图像数字水印方面也提出了很多不可逆的算法[7,14],但是在 3 维网格水印方面的文献还比较少。本文提出了一种快速的不可逆的算法,它利用零值奇异值的特点使得攻击者难以实施解释攻击。

2　水印算法

水印算法的总体流程类似于文献[1],如图 1 所示,它显示了对原始网格嵌入水印进行保护到水印网格遭受攻击后需要检测水印的过程,需要注意的是为了抵御各种对水印的攻击,需要在提取水印前进行一些预处理以减少攻击对水印的伤害。

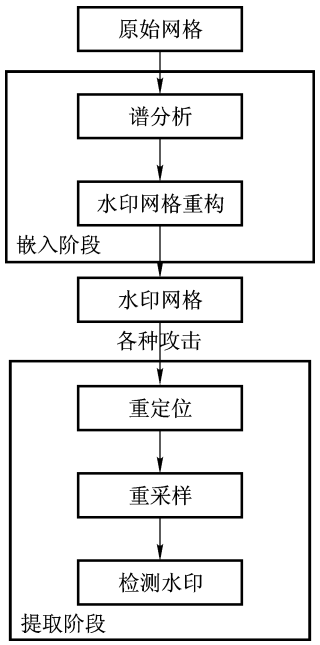

图 1　水印总体流程

Fig. 1　Watermarking process

本文提出的算法主要创新点在于水印嵌入过程中的谱分析,通过谱分析得到原始网格的频谱域后就可以嵌入水印,然后再根据修改过的频谱域重建网格得到水印网格。

当发生版权纷争的时候,就可以执行水印提取操作提取出水印以证明版权的归属权。首先把待检测的网格与水印网格进行对齐,以抵御相似变换攻击,然后进行重采样使得检测的网格与水印网格的拓扑结构一样。之后再进行谱分析就可以提取水印了,当提取到水印和原水印有很高的相似度时,就证明待检测网格含有本文提出的水印。

2.1 水印嵌入

3 维模型的表示方式有很多种,例如有三角网格、多边形网格、隐式曲面和参数曲面等。本文主要是在三角网格上嵌入水印。提出了一种不可逆的水印嵌入算法:首先在网格上选取一些顶点用来构成一个分析矩阵 $\boldsymbol{R}$,然后对 $\boldsymbol{R}$ 进行奇异值分解,得到奇异值,把水印信息嵌入到这些包含有零值奇异值中,由于奇异值为零这一特性,就会使得攻击者很难从水印网格重构一个属于他的“原始网格”,从而成功抵御了解释攻击。

2.1.1 谱分析

为了构造含有零值奇异值的分析矩阵,可以设计一个上半部分和下半部分相同的矩阵。首先,在网格上随机选取 $2\times N\times M$ 个顶点:$\boldsymbol{q}(x_1,x_2,\cdots,x_{2NM})$(只是顶点的序号)。其中 N 为 6 的倍数,$M=N/6$。然后让 $L=3M$,这样就可以构造一个分析矩阵 $\boldsymbol{R}$。选取随机顶点数组上的 $N\times M$ 对顶点:$q_{2k-1},q_{2k},k=1,\cdots,NM$,计算这些点对的相应中点:$Mesh(q)$ 表示网格的第 q 个顶点,$O_k=(Mesh(q_{2k-1})+Mesh(q_{2k}))/2,k=1,\cdots,NM$。令 $v_k=Mesh(q_{2k-1})-O_k=O_k-Mesh(q_{2k})$,$k=1,\cdots,NM$。于是可以构造如下 $6M\times N=N\times N$ 的分析矩阵:

$$\boldsymbol{R}=\begin{bmatrix}\boldsymbol{v}_1^{\mathrm{T}} & \cdots & \boldsymbol{v}_k^{\mathrm{T}} & \cdots & \boldsymbol{v}_N^{\mathrm{T}}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{v}_{(M-1)N+1}^{\mathrm{T}} & \cdots & \boldsymbol{v}_{(M-1)N+k}^{\mathrm{T}} & \cdots & \boldsymbol{v}_{MN}^{\mathrm{T}}\\ \boldsymbol{v}_1^{\mathrm{T}} & \cdots & \boldsymbol{v}_k^{\mathrm{T}} & \cdots & \boldsymbol{v}_N^{\mathrm{T}}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{v}_{(M-1)N+1}^{\mathrm{T}} & \cdots & \boldsymbol{v}_{(M-1)N+k}^{\mathrm{T}} & \cdots & \boldsymbol{v}_{MN}^{\mathrm{T}}\end{bmatrix}$$

$\boldsymbol{R}$ 的上半部分和下半部分完全相同,目的是保证它至少有 $L=N/2$ 个零的奇异值。记 $\boldsymbol{R}$ 的奇异值分解为

$$\boldsymbol{R}=\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{1}$$

式中,$\boldsymbol{\Sigma}=\mathrm{diag}(\lambda_1,\cdots,\lambda_L,0,\cdots,0)$。

假设要嵌入的水印信息是一串二进制序列 $\{b_i\}_{i=1}^{N}$,长度为 N。可用下式来嵌入水印信息:

$$\lambda_i'=\lambda_i+b_iA,\ i=1,\cdots,N \tag{2}$$

式中,A 是水印嵌入强度。得到修改后的 $\boldsymbol{\Sigma}'=\mathrm{diag}(\lambda_1',\cdots,\lambda_L',\lambda_{L+1}',\cdots,\lambda_N')$,再利用式(1)右端计算分析矩阵 $\boldsymbol{R}'=\boldsymbol{U}\boldsymbol{\Sigma}'\boldsymbol{V}^{\mathrm{T}}$。嵌入水印后,保存奇异值向量矩阵 $\boldsymbol{U},\boldsymbol{V}$,顶点序列 $\boldsymbol{q}(x_1,x_2,\cdots,x_{2NM})$ 及中点系列 $\{O_k\}_{k=1}^{NM}$。

2.1.2 水印网格重构

得到的分析矩阵 $\boldsymbol{R}'$ 如下:

$$\boldsymbol{R}'=\begin{bmatrix}\boldsymbol{v}_1'^{\mathrm{T}} & \cdots & \boldsymbol{v}_k'^{\mathrm{T}} & \cdots & \boldsymbol{v}_N'^{\mathrm{T}}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{v}_{(M-1)N+1}'^{\mathrm{T}} & \cdots & \boldsymbol{v}_{(M-1)N+k}'^{\mathrm{T}} & \cdots & \boldsymbol{v}_{MN}'^{\mathrm{T}}\\ \boldsymbol{v}_1''^{\mathrm{T}} & \cdots & \boldsymbol{v}_k''^{\mathrm{T}} & \cdots & \boldsymbol{v}_N''^{\mathrm{T}}\\ \vdots & \vdots & \vdots & \vdots & \vdots\\ \boldsymbol{v}_{(M-1)N+1}''^{\mathrm{T}} & \cdots & \boldsymbol{v}_{(M-1)N+k}''^{\mathrm{T}} & \cdots & \boldsymbol{v}_{MN}''^{\mathrm{T}}\end{bmatrix}$$

水印网格除了前面随机选中的顶点对位置有小的扰动外,其他顶点位置不变。这些扰动顶点对分别按如下方式计算:

$$\begin{aligned}Mesh(q_{2k-1})&=\boldsymbol{v}_k'+\boldsymbol{O}_k,\\ Mesh(q_{2k})&=\boldsymbol{O}_k-\boldsymbol{v}_k'',\ k=1,\cdots,NM\end{aligned} \tag{3}$$

需要注意,一般地,有 $v_k'\neq v_k''$,但是,只要它们对应相同的下标 k,就可以认为它们原来是一对,即可以采用式(3)求出修改后的网格顶点。

2.2 水印提取

为检测是否对某个作品拥有版权,需要提取水印来进行比较。考虑到嵌入水印的模型有可能受到相似变换攻击,因此,提取水印之前需要把待检测网格 $\boldsymbol{M}_{\mathrm{t}}$ 进行旋转、平移和缩放使得检测网格与未受攻击的水印网格 $\boldsymbol{M}_{\mathrm{w}}$ 尽量“对齐”。目前,这一工作需要用户交互。由用户定义 3 对匹配点,再利用匹配点计算出初始的旋转、平移、缩放量[15],然后采用经典的迭代近邻点(ICP)算法[16]来找到最佳匹配的旋转、平移、缩放量。经过这个过程后网格变为 $\boldsymbol{M}_{\mathrm{t}}'$。

此外,为了消解修改网格拓扑结构的攻击(还可以防止顶点重排列攻击),再用水印网格 $\boldsymbol{M}_{\mathrm{w}}$ 对受攻击网格 $\boldsymbol{M}_{\mathrm{t}}'$ 进行重采样:即对于水印网格的每一点,找到在 $\boldsymbol{M}_{\mathrm{t}}'$ 表面上的最近点,然后用最近点来代替这一点。当这一点与最近点的距离太大而超过一定的阈值后,要标记这一点,说明对应的点可能已被剪切。最后用经过重采样的水印网格代替 $\boldsymbol{M}_{\mathrm{t}}'$ 就完成了重采样的步骤,记为 $\boldsymbol{M}_{\mathrm{t}}''$。具体的做法请参考文献[3]。

经过上面两个步骤后,就可以使用在谱分析阶段保存的 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 来提取水印信息。由于 $\boldsymbol{M}_{\mathrm{t}}''$ 与未受攻击的水印网格有相同的拓扑连接,因此可按嵌入水印时生成的顶点对序列找到 $\boldsymbol{M}_{\mathrm{t}}''$ 中的相应顶点对,再结合保存的中点序列 $\{\boldsymbol{O}_k\}_{k=1}^{NM}$ 构造恢复分析矩阵 $\boldsymbol{R}''$ 如下:

$$\boldsymbol{R}''=\begin{bmatrix} \boldsymbol{v}_1'^{\mathrm{T}} & \cdots & \boldsymbol{v}_k'^{\mathrm{T}} & \cdots & \boldsymbol{v}_N'^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{v}'^{\mathrm{T}}_{(M-1)N+1} & \cdots & \boldsymbol{v}'^{\mathrm{T}}_{(M-1)N+k} & \cdots & \boldsymbol{v}'^{\mathrm{T}}_{MN} \\ \boldsymbol{v}_1''^{\mathrm{T}} & \cdots & \boldsymbol{v}_k''^{\mathrm{T}} & \cdots & \boldsymbol{v}_N''^{\mathrm{T}} \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ \boldsymbol{v}''^{\mathrm{T}}_{(M-1)N+1} & \cdots & \boldsymbol{v}''^{\mathrm{T}}_{(M-1)N+k} & \cdots & \boldsymbol{v}''^{\mathrm{T}}_{MN} \end{bmatrix}$$

式中,$\boldsymbol{v}_k'=Mesh(q_{2k-1})-\boldsymbol{O}_k$,$\boldsymbol{v}_k''=\boldsymbol{O}_k-Mesh(q_{2k})$,$k=1,\cdots,NM$。

然后计算矩阵乘积

$$\boldsymbol{\Sigma}_{\mathrm{w}}=\boldsymbol{U}^{\mathrm{T}}\cdot\boldsymbol{R}''\cdot\boldsymbol{V} \tag{4}$$

注意 $\boldsymbol{\Sigma}_{\mathrm{w}}$ 可能不是对角矩阵,但是可认为在一定的攻击下它还是接近对角矩阵的。利用下式来提取水印信息:

$$\boldsymbol{b}_i'=(\lambda_{wi}-\lambda_i)/A \tag{5}$$

式中,λ_{wi}是矩阵 $\boldsymbol{\Sigma}_{\mathrm{w}}$ 的第 i 个对角线上的值。

最后对提取到的水印信息和原来嵌入的水印信息相比较,确定它们之间的相关系数,可以采用下式来计算[17]:

$$C(\{\boldsymbol{b}_i\},\{\boldsymbol{b}_i'\})=\frac{\sum_{i=1}^{N}(b_i-\bar{b})(b_i'-\bar{b}')}{\sqrt{\sum_{i=1}^{N}(b_i-\bar{b})^2}\sqrt{\sum_{i=1}^{N}(b_i'-\bar{b}')^2}} \tag{6}$$

式中,$\bar{b}$ 是水印序列 $\{\boldsymbol{b}_i\}_{i=1}^{N}$ 的均值,$\bar{b}'$是水印序列 $\{\boldsymbol{b}_i'\}_{i=1}^{N}$的均值。

当相似度 $C(\{\boldsymbol{b}_i\},\{\boldsymbol{b}_i'\})$ 大于某一阈值(例如 0.5),就可以判断这个作品的归属。相似度的范围为$\{-1,1\}$之间。

2.3　算法的可行性

在分析算法的可行性时,先介绍一些基本概念。假定 $\boldsymbol{M}$ 为原始网格,$\boldsymbol{W}$ 为待嵌入的水印信息,$\boldsymbol{M}'$为嵌入水印的网格模型。参照文献[6],引入如下符号及定义(这里只考虑三角网格模型):$(\varepsilon,D,C)=(\varepsilon(\boldsymbol{M},\boldsymbol{W}),D(\boldsymbol{M}',\boldsymbol{M}),C(\boldsymbol{W}',\boldsymbol{W}))$表示一个水印算法,其中 $\varepsilon(\boldsymbol{M},\boldsymbol{W})$表示的是嵌入函数,$D(\boldsymbol{M}',\boldsymbol{M})$表示的是水印提取函数,$C(\boldsymbol{W}',\boldsymbol{W})$是求相关系数的函数。

水印算法(ε,D,C)称为可逆的当且仅当存在逆映射 ε^{-1}使得

(1)$\varepsilon^{-1}(\hat{\boldsymbol{M}})=(\boldsymbol{M}',\boldsymbol{W}')$

(2)$\varepsilon(\boldsymbol{M}',\boldsymbol{W}')=\hat{\boldsymbol{M}}$

(3)$C(D(\hat{\boldsymbol{M}},\boldsymbol{M}'),\boldsymbol{W}')=1$,否则的话这个水印算法是不可逆的。

除了可逆的定义外,文献[6]还定义了半可逆水印算法,主要区别就在于后者把第二个条件变为 $\varepsilon(\boldsymbol{M}',\boldsymbol{W}')\approx\hat{\boldsymbol{M}}$,也就是说由虚假原始网格 $\boldsymbol{M}'$嵌入水印信息后只是跟 $\hat{\boldsymbol{M}}$ 很相似。

根据以上的定义,接下来介绍解释攻击是如何实现的。假设 Alice 创作了 3 维网格作品 $\boldsymbol{M}$,她采用了某一水印算法并用水印 $\boldsymbol{W}$ 产生水印作品 $\hat{\boldsymbol{M}}$。攻击者 Bob 通过分析 Alice 的算法去伪造虚假原始作品 $\tilde{\boldsymbol{M}}$。他首先根据算法对 $\hat{\boldsymbol{M}}$ 进行相同的处理,然后再减去自己的水印信息 $\tilde{\boldsymbol{W}}$,再重构网格就可以得到虚假原始作品 $\tilde{\boldsymbol{M}}$。上面的整个操作可以称为减法操作,原作者的方法可以称为加法操作。攻击者减去水印采用的公式是和嵌入水印时采用的公式刚好互逆的,例如对于本文的嵌入方法,攻击者减去水印信息可以通过下式来实现:

$$\tilde{\lambda}_i'=\tilde{\lambda}_i-\tilde{b}_iA\quad i=1,\cdots,N \tag{7}$$

式中,$\tilde{b}_i$ 是攻击者的水印信息,$\tilde{\lambda}_i$ 是攻击者对他自己的分析矩阵奇异值分解后的奇异值,$\tilde{\lambda}_i'$ 是嵌入攻击者水印后的奇异值。

通过这个方法伪造的虚假原始作品 $\tilde{\boldsymbol{M}}$,可以在作品 $\hat{\boldsymbol{M}}$ 中检测到攻击者的水印信息 $\tilde{\boldsymbol{W}}=\{\tilde{b}_i\}$(因为水印是从 $\hat{\boldsymbol{M}}$ 中减去的,这样就像是在 $\tilde{\boldsymbol{M}}$ 中加上了水印信息),而且由于鲁棒性的原因,在原作者的作品 $\boldsymbol{M}$ 中也能检测到 $\tilde{\boldsymbol{W}}$,使得原作者无法再证明发布的作品是他自己的[6,18]。

根据本文提出的算法,分析以下两种构造虚假原始网格攻击方式:

第 1 种,根据算法逆做一遍(如图 2 所示):当攻击者 Bob 用水印网格 $\hat{\boldsymbol{M}}$ 构造一个虚假原始网格 $\tilde{\boldsymbol{M}}$ 时,他首先选取自己的随机顶点序列 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$,得到分析矩阵 $\boldsymbol{R}(\hat{\boldsymbol{M}})$后用式(8)奇异值分解得到对角矩阵 $\boldsymbol{\Sigma}$。把 $\boldsymbol{\Sigma}$ 的对角线元素代入式(7)的 $\tilde{\lambda}_i$ 进行减法水印嵌入得到 $\tilde{\boldsymbol{\Sigma}}$(注意到此时 $\tilde{\boldsymbol{\Sigma}}$ 的对角线上会含有负数),然后用式(9)重构分析矩阵 $\boldsymbol{R}(\tilde{\boldsymbol{M}})$,最后再重构网格得到虚假原始作品 $\tilde{\boldsymbol{M}}$。但是 Bob 的虚假原始作品 $\tilde{\boldsymbol{M}}$ 有个缺点,他不能用 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 来作为检测水印的密钥,因为 $\tilde{\boldsymbol{\Sigma}}=\boldsymbol{U}^{\mathrm{T}}\cdot\boldsymbol{R}(\tilde{\boldsymbol{M}})\cdot\boldsymbol{V}$ 中含有负值,这在原始网格的奇异值分解中是不可能的。此时 Bob 只能再通过奇异值分解来得到用于检测水印的密钥 $\boldsymbol{U}'$和 $\boldsymbol{V}'$,即用式(10)再一次奇异值分解。因为此时密钥已经和 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 不同了,所以

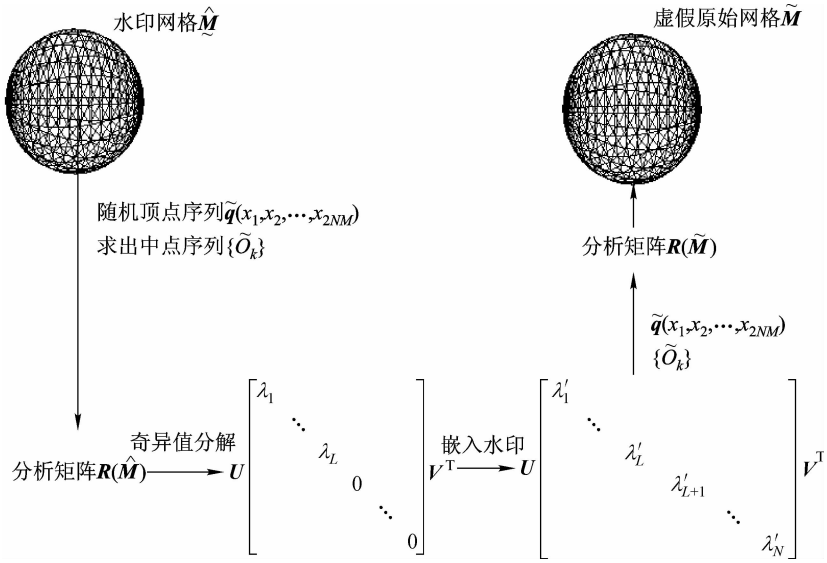

图 2 伪造虚假网格

Fig. 2 Fabricate fake mesh

检测出来的水印也不正确了。实验表明,此时的相关系数几乎都是负值。

$$\boldsymbol{R}(\hat{\boldsymbol{M}}) = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^{\mathrm{T}} \tag{8}$$

$$\boldsymbol{R}(\tilde{\boldsymbol{M}}) = \boldsymbol{U}\,\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}^{\mathrm{T}} \tag{9}$$

$$\boldsymbol{R}(\tilde{\boldsymbol{M}}) = \boldsymbol{U}'\tilde{\boldsymbol{\Sigma}}\boldsymbol{V}'^{\mathrm{T}} \tag{10}$$

总之,上面的方法构造的虚假原始网格只满足了可逆算法条件的(1)和(2)。

为了构造出一个虚假原始网格满足可逆性质的 3 个条件。接着介绍第 2 种方法:Bob 采用估计中点序列的方法来进行攻击,一开始选取自己的随机顶点序列 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$,不过这次用来构造分析矩阵 $\boldsymbol{R}(\hat{\boldsymbol{M}})$ 的中点序列不是由随机顶点序列 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$ 来生成,而是 Bob 自己通过估计而得(比如由随机顶点序列 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$,根据嵌入时的算法在 $\hat{\boldsymbol{M}}$ 中求出中点序列 $\{\hat{O}_k\}_{k=1}^{NM}$,然后稍微进行扰动),设为 $\{\hat{O}'_k\}_{k=1}^{NM}$。由 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$ 和 $\{\hat{O}'_k\}_{k=1}^{NM}$ 生成分析矩阵 $\boldsymbol{R}(\hat{\boldsymbol{M}})$ 后,其他做法和上面的差不多,只是用式(7)进行减法水印后要使得 $\tilde{\boldsymbol{\Sigma}}$ 的后半部分为零,最后得到虚假原始网格 $\tilde{\boldsymbol{M}}$。但是这个攻击成立的前提是根据随机顶点序列 $\tilde{\boldsymbol{q}}(x_1,x_2,\cdots,x_{2NM})$ 在 $\tilde{\boldsymbol{M}}$ 上取得的中点序列 $\{\tilde{O}_k\}_{k=1}^{NM}$ 要和 $\{\hat{O}'_k\}_{k=1}^{NM}$ 吻合,而这是很困难的。因为在这个算法的谱分析阶段里中点序列的选取和网格的顶点坐标有关系,而且在重构网格的过程中也要这个中点序列来恢复分析矩阵,当嵌入水印信息使得顶点坐标发生变化时,就很难在一开始估算 $\{\hat{O}'_k\}_{k=1}^{NM}$,使得它就是虚假原始网格的中点序列 $\{\tilde{O}_k\}_{k=1}^{NM}$。

上面的攻击是完全通过对算法进行逆处理而得到的,水印嵌入的分析过程如图 3 所示。

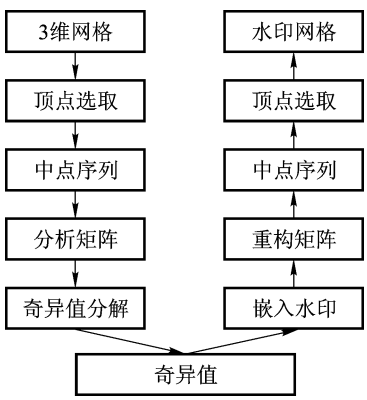

图 3 水印嵌入过程

Fig. 3 Watermark embedding

根据上面的攻击分析和图 3 可以看出,影响算法可逆的难度主要在于顶点选取后的中点序列,这是因为中点序列是跟原始 3 维网格的顶点坐标有关的,而且在嵌入水印后网格的顶点坐标会发生变化。它不同于顶点选取时的随机顶点序列,尽管随机顶点序列也是跟原始 3 维网格有关,但是在嵌入水印后随机顶点序列并没有变。

在上面提供的两种攻击中,第 1 种攻击不成功

主要是因为在嵌入水印过程中密钥 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 改变了,而第 2 种攻击不成功是因为中点序列在嵌入过程中改变了。这些密钥都是由原始网格的顶点坐标决定的。

根据以上分析,可认为要使算法不满足可逆性的 3 个条件,那么用做嵌入水印时的密钥要由原始网格来决定。

这种密钥越多越使得攻击者难以成功构造虚假的原始模型,例如本例就有两条这样的密钥:奇异值分解时的 $\boldsymbol{U}$ 和 $\boldsymbol{V}$、构造分析矩阵的中点序列 $\{O_k\}_{k=1}^{NM}$。随机顶点序列不算在内,因为它在水印嵌入前后并没有发生变化。

3　实验结果与分析

本文用这个方法对多种模型进行了实验,水印信息和选取的顶点都是随机选取的,其中分析矩阵的大小 N 要是 6 的倍数且要满足 $2\times N\times M$ 要小于模型的顶点数。该算法对于相似变换、加噪和光顺(平滑化)等攻击都有很高的鲁棒性,但对简化攻击鲁棒性较弱。可以通过控制水印强度来增加或减少鲁棒性,当水印强度较大时,鲁棒性较强;而当水印强度较小时,鲁棒性较弱。

3.1　相似变换

如图 4(顶点数为 891,面数为 1 704)所示,把嵌入水印的模型进行相似变换,然后对水印进行提取(在提取之前可能需要自定义 3 个重定位校准点)。所得的相关系数为 1,表 1 显示了实验结果:

图 4　相似变换攻击实验结果

Fig. 4　Similarity attack experimental results

表 1　相似变换攻击实验

Tab. 1　Similarity attack experiments

模型	顶点数	攻击方式	N(6 的倍数)	相关系数
马	3 000	平移 + 缩放 + 旋转	90	1.000 0
马	3 000	平移 + 缩放 + 旋转	60	1.000 0
苹果	891	平移 + 缩放 + 旋转	12	1.000 0
苹果	891	平移 + 缩放 + 旋转	36	1.000 0

3.2　添加噪声

添加噪声会随机地根据顶点法向修改顶点的坐标,实验表明本文提出的算法对添噪攻击有鲁棒性。

如图 5 所示(face 模型, 12 530 个顶点),在各个顶点的法向量上添加以下两种基准的随机扰动:

(1)选取平均边长的 0.1 倍为随机扰动的基准,添加 1 次。

(2)选取最短边长的 0.5 倍为随机扰动的基准,循环添加 5 次。

图 5　噪声攻击实验结果

Fig. 5　Noise attack experimental results

噪声攻击实验结果如表 2 所示。

表 2　添加噪声攻击实验

Tab. 2　Noise attack experiments

模型	攻击方式	水印强度	N(6 的倍数)	相关系数
face	(1)	奇异值的十分之一	90	0.93
face	(1)	奇异值的五十分之一	90	0.88
face	(2)	奇异值的五十分之一	90	0.75
face	(2)	奇异值的十分之一	120	0.94
苹果	(1)	奇异值的五十分之一	36	0.95
苹果	(1)	奇异值的一百分之一	36	0.87
苹果	(2)	奇异值的五十分之一	36	0.99
苹果	(2)	奇异值的一百分之一	36	0.94

3.3 平滑处理

本文对模型进行了平滑处理,例如图 6(恐龙三角网格模型,顶点数 2 832、边总数 8 490、面总数 5 660,攻击均采用平滑系数 $\lambda = 0.630\ 7$,通带频率 kPB = 0.1 的参数设置[19]),实验结果表明,该算法对平滑攻击有一定的鲁棒性,对不同的水印强度嵌入水印如表 3 所示。

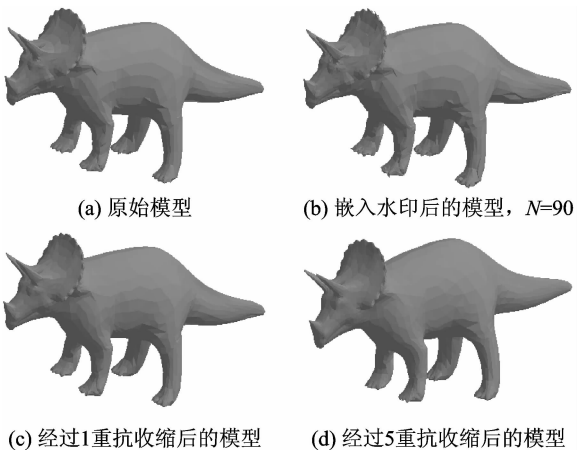

图 6 平滑攻击实验结果

Fig. 6 Smoothing attack experimental results

表 3 平滑攻击实验

Tab. 3 Smoothing attack experiments

攻击次数	水印强度	N(6 的倍数)	相关系数
1	奇异值的百分之一	90	0.526
1	奇异值的十分之一	90	0.98
5	奇异值的五十分之一	90	0.76
5	奇异值的十分之一	90	0.92

从表 3 可以看出水印强度越大,其鲁棒性越高,而水印强度越小,鲁棒性越弱。在没有明显改变模型形状的时候,应该尽量选择比较大的水印强度。

3.4 二次水印

对 mannequin 模型(顶点数 11 704、边总数 35 103、面总数 23 402)进行了二次水印攻击,通过改变二次水印的 N 值来改变攻击强度,然后通过提取第 1 次嵌入的水印来测试相关系数。二次水印攻击一般会对原水印造成一定的影响,表 4 表明本文提出的算法可以很好地抵御这种攻击。

最后还对没有嵌入过水印的模型进行了提取操作,得到的相关系数总是在零左右,说明该算法不会在没有水印的模型中检测出水印。

表 4 二次水印实验

Tab. 4 Twice-watermarking experiments

第 1 次水印的 N 值	第 2 次水印的 N 值	第 1 次水印相关系数
60	90	0.931 966
90	90	0.965 069
120	90	0.994 260
120	120	0.966 680

4 结 论

本文提出了一个不可逆的 3 维数字水印算法,它利用了零值奇异值使得攻击者不能使用相同的算法伪造一个虚假的原始网格,从而成功抵御解释攻击。但是如果攻击者使用另外一种鲁棒性算法进行减法水印嵌入他自己的水印,从而构造了一个虚假原始网格,由于鲁棒性的原因,攻击者利用这个虚假的原始网格很可能通过比较原始网格而提取出自己的水印信息,从而使得原作者无法证明他的拥有权。所以即使是不可逆的算法,仍然还会存在危险性,要真正解除这种威胁,可以采用以下两点:

(1) 原作者可以把他的原始作品拿到认证机构去注册;

(2) 建立了统一的水印算法机制,在统一的不可逆的算法下可以实现不需要认证中心而保证能够证明版权的归属性。

如果采用第 1 种方法,将会使得数字水印技术的使用受到限制。但是第 2 种方法也很难统一现在的数字水印算法,因为在相当一段时间内还很难提出一个相当完美的算法。

从上面可以看出,单靠数字水印技术是无法保证版权的唯一性,需要一定的机制来实现认证的过程,水印技术与现代密码学的结合将会是未来的研究方向。而对鲁棒性的要求也应该在不同的应用领域上有所不同,否则很难将水印技术推广到实际应用中。

参考文献(References)

1 Ohbuchi R, Mukaiyama A, Takahashi S. A frequency-domain approach to watermarking 3D shapes[J]. Computer Graphics Forum, 2002, **21**(3): 373-382.

2 Ohbuchi R, Mukaiyama A, Takahashi S. Watermarking a 3D shape model defined as a point set[A]. In: Proceedings of the International Conference on Cyberworlds[C], Tokyo, Japan, 2004: 392-399.

3 Praun E, Hoppe H, Finkelstein A. Robust mesh watermarking[A]. In: Proceedings of the ACM SIGGRAPH[C], Los Angeles, CA, USA, 1999: 49-56.

4 Wu Jian-hua, Kobbelt L. Efficient spectral watermarking of large meshes with orthogonal basis function[J]. The Visual Computer, 2005, **21**(8-10): 848-857.

5 Craver S, Memon N, Yeo B L, Yeung M M. Can invisible watermarks resolve rightful ownerships[J]. SPIE Electronic Imaging: Storage and Retrieval of Image and Video Databases, 1997, **12**(2): 310-321.

6 Craver S, Memon N, Yeo B L, *et al*. Resolving rightful ownerships with invisible watermarking techniques: limitations, attacks, and implications[J]. IEEE Journal on Selected Areas in Communications, 1998, **16**(4): 573-586.

7 Liu Rui-zhen, Tan Tie-niu. An SVD-based watermarking scheme for protecting rightful ownership[J]. IEEE Transactions on Multimedia, 2002, **4**(1): 121-128.

8 Press W H, Teukolsky S A, Vetterling W T, *et al*. Numerical Recipes in C: The Art of Scientific Computing, Second Edition[M]. New York: Cambridge University Press, 1995: 59.

9 Kohei Murotani, Kokichi Sugihara. Watermarking 3D polygonal meshes using the singular spectrum analysis[A]. In: Proceedings of the IMA Conference on the Mathematics of Surfaces [C], Heidelberg, Germany: Springer Berlin, 2003: 85-98.

10 Ohbuchi R, Masuda H, Aono M. Embedding data in 3D models [A]. In: Proceedings of the European Workshop on Interactive Distributed Multimedia Systems and Telecommunication Services'97 [C], Heidelberg, Germany: Springer Verlag, 1997:1-10.

11 Ohbuchi R, Masuda H, Aono M. Watermarking three-dimensional polygonal models through geometric and topological modifications[J]. IEEE Journal on Selected Areas in Communications, 1998, **16**(4): 551-559.

12 Benedens O. Geometry-based watermarking of 3d models[J]. IEEE Computer Graphics and Applications, 1999, **19**(1):46-55.

13 Quan Liu, Liu Yang. Watermarking of 3D polygonal meshes based on feature points[A]. In: Proceedings of the 2nd IEEE Conference on Industrial Electronics and Applications[C], Harbin, China, 2007: 1837-1841.

14 Qiao L T, Nahrstedt K. Watermarking schemes and protocols for protecting rightful ownership and customer's rights[J]. Vis. Commun. Image Represent, 1998, **9**(3): 194-210.

15 Berthold K P Horn. Closed-form solution of absolute orientation using unit quaternions[J]. Journal of the Optical Society A, 1997, **4**(4): 629-642.

16 Besl P, McKay J. A method for registration of 3-d shapes[J]. IEEE Trans actions on Pattern Recognition and Machine Intelligence, 1992, **14**(2): 239-255.

17 Mrak M, Grgic S, Grgic M. Picture quality measures in image compression systems[A]. In: Proceedings of the IEEE Eurocon[C], Ljubljana, Eslovenia, 2003: 233-236.

18 Sun Shui-fa. Study of Digital Watermarking[D]. Hangzhou: Zhejiang University,2005.[孙水发. 数字水印技术研究[D]. 杭州:浙江大学,2005.]

19 Taubin G. A signal processing approach to fair surface design[A]. In: Proceedings of the SIGGRAPH'95[C], New York NY, USA: ACM Press, 1995: 351-358.