

一种采用曲线分割的矢量图水印算法

张鸿生¹⁾ 李岩^{1),2)} 曹阳^{1),2)}

¹⁾(华南师范大学计算机学院, 广州 510631) ²⁾(华南师范大学空间信息技术与应用研究中心, 广州 510631)

摘要 针对矢量图形既需具有鲁棒性又要保证图形几何精度的问题,提出了一种针对矢量图形水印嵌入、提取和检测方法。水印的嵌入算法是将一幅矢量图形视为曲线的集合,按设定阈值进行曲线分割;再在容差范围内,使每条曲线对应一个水印位,且对曲线中每个结点嵌入一个含有用户证书信息的水印点。水印提取算法则对原图和含水印图进行对应的分割;按嵌入算法的逆过程逐曲线提取出水印信息;再通过计算原水印信息和所提取水印信息的相似度检测水印存在与否和符合程度。实验结果表明,该方法在无攻击和几何变形操作攻击下的水印检测相似度均接近1.00;且水印图即使在剪裁了2/3后,仍能保证大于0.9的相似度;同时,方法具有抗D-P压缩能力,并可借助所给出的相似度阈值经验曲线,控制水印点的偏离角度,或在水印信息检测的自动化处理流程中设定参数,起到既能保证矢量图形的精度,又具有较好的鲁棒性的作用。

关键词 矢量图形 数字水印 鲁棒性 相似度 曲线分割

中图法分类号:TP301.6 文献标识码:A 文章编号:1006-8961(2009)08-1516-07

A Watermarking Algorithm for Vector Graphics Using Curve Partition

ZHANG Hong-sheng¹⁾, LI Yan^{1),2)}, CAO Yang^{1),2)}

¹⁾(Computer School, South China Normal University, Guangzhou 510631)

²⁾(Spatial Information Research Center, South China Normal University, Guangzhou 510631)

Abstract This paper discusses a watermarking method towards keeping robustness and high accuracy of vector graphics. The graphics is treated as a set of curves in the embedding algorithm, and each curve is divided into several shorter curves under a given threshold. And then a watermark bit combined with user certificate is embedded into each segment around the node for every cutting curve. To extract the watermark, it also divides up the watermarked graphics according to the division of the original graphics. Then, it extracts the watermark bit from each curve with the inverse procedure of the embedding process. Finally, the method calculates the similarity between the original watermark bits and the extracted ones from the watermarked graphics, and determines whether the watermark exists or not. The experiment results showed that the method not only guarantees the accuracy of vector graphics but also possesses good robustness, such as it gives 1.00 similarities under no attack or only geometric transformation with the map; the anti-cropping ability gives 0.9 even for the graphics cropped 2/3. And it also has the ability of anti-compression of D-P approach and the experiment curve of the similarity threshold given which helps to control the anti-attack ability of watermark and set parameters for an automatic procedure of watermark detection.

Keywords vector graphics, digital watermark, robustness, similarity, curve partition

基金项目:国家自然科学基金项目(60842007);广东省百项工程项目(2002B32101);广东省重点引导项目(2004B32501001;2005B30801006)

收稿日期:2008-01-08;改回日期:2008-04-25

第一作者简介:张鸿生(1984~),男,华南师范大学计算机应用技术专业硕士研究生,主要研究方向为图像图形处理与空间信息技术。

E-mail: zhangstream@gmail.com

通讯作者:李岩, E-mail: yanli@sclu.edu.cn

1 引言

随着矢量图形的广泛应用,人们常以各种矢量格式进行信息表达和交流,如:CAD 图形绘制建筑设计图、规划方案、虚拟现实中的3维数字模型和GIS中矢量地图等。特别是空间信息共享已形成共识,作为空间数据主流格式之一,矢量空间数据的版权问题则显得更加重要,使得矢量图形数字水印研究迫在眉睫。

在国内外,有关矢量图形数字水印的研究较少,常受困于水印的鲁棒性、抗压缩能力和抗格式转换等方面的性能不能兼顾,使得许多水印方法难以在实际中应用,更无法实现商用化。当前,对矢量图形嵌入数字水印的算法按水印信息嵌入方式分为两大类:一是移动点,二是添加点。第一类算法又被分为频率域和空间域算法。频率域移动点算法把矢量图层变换到频率域,再把水印信息编码到频率域系数中,再将其变换回空间域。例如:文献[1]、[2]均将矢量图层分为若干个多边形,但前者是逐多边形对其结点进行1维离散余弦变换(DCT),通过修改变换后的高频信号系数而达到嵌入水印的目的;后者则将每个结点坐标看做复数,进行离散傅里叶变换,通过改变傅立叶描述符(Fourier descriptors)的大小来嵌入水印。此外,文献[3]、[4]的基于小波变换域和复数小波变换域的算法,均可归于此类算法。空间域移动点算法则通过直接移动矢量图层的结点位置,实现水印信息嵌入。文献[5]~[9]都是通过一定容差范围内移动每一个坐标点的位置来嵌入水印。其中:文献[8]先用二叉树对矢量图进行划分,再嵌入水印信息。文献[9]则在文献[8]的基础上,区分矢量地图中不同的地物,提出了一种双重嵌入的水印算法。显而易见,这类算法的主要问题在于不能确保图形精度,不适于空间数据的要求。第二类添加点算法,在国内外报道较少,它本着不改变矢量图层原有结点的原则,只在某些特定位置添加新的结点,从而实现水印信息嵌入,如:文献[10]在以多边形对矢量图层进行划分后,对每个多边形离质心最近的曲线,在两结点中嵌入水印点。这类算法的优点是可保证原图精度,但数据量比原图文件要大,需权衡是否影响应用和传输负载。同时,文献[10]虽具有较好的鲁棒性,但水印安全性仍存在不足,如:它在进行有损压缩操作后,水印信息易被压

缩掉。文献[11]曾针对常用DP压缩算法,提出一种抗压缩的水印算法。

本文针对这类对精度要求高的矢量图形,如SVG格式的矢量地图,为其探索新的基于曲线分割的添加点水印算法,旨在提出一种既适合于空间信息图形几何精度要求,又具有鲁棒性保证的方法,即:具有变形操作、抗裁剪和抗压缩的矢量图形水印算法,并提出相应的水印性能检测评价方法。

2 水印算法

2.1 水印嵌入算法

2.1.1 曲线的分割

本文算法是把每一个水印信息位(watermark bit)嵌入到矢量图层中图形的每一条曲线内,若某图层所包含的原始曲线数很少,且单条曲线又很长,会造成水印信息位数太少;或水印信息长度大于曲线数目,使水印信息在整个图层上分布不均匀,或水印信息不完整,从而降低水印的鲁棒性。因此,在嵌入水印前应先对整个图层的曲线进行长度和数目适中的调整,即:曲线段分割。

首先,按深度优先顺序排列好图形树中的多边形边界(曲线),并计算各曲线的长度(以结点数目计)。在分割曲线时,不仅需限定曲线的最大长度为 d_{\max} ,即大于 d_{\max} 的曲线应分割为多条曲线;还需限定曲线的最小长度为 d_{\min} ,小于 d_{\min} 的曲线弃为不嵌入水印点线段。另外, d_{\max} 可根据用户信息(M)和用户证书中的密钥计算水印信息的长度 n ,以字节位数计, $m = (m_1, m_2, \dots, m_n)$,其中 m_i 取0或1。

图形经过分割后,整个图层被分割成多曲线的集合 $\{P_i\}$,第 i 条曲线记为 P_i ,其结点数 $d_{\min} \leq |P_i| \leq d_{\max}$,并设原图形被分割后曲线的总数为 $L = |\{P_i\}|$;例如:取 $m = (1, 0)$, $d_{\min} = 2$, $d_{\max} = 3$,则 $|P_i|$ 为2或3,若svg中只有一条曲线,其坐标点为(7个点):

(113.202 184,22.281 367)(113.202 038,22.281 393)
(113.202 143,22.281 673)(113.202 056,22.281 709)
(113.202 065,22.281 79)(113.202 559,22.281 82)
(113.202 587,22.282 154)

则曲线被分割为3段,即:

P_1 :(113.202 184,22.281 367)(113.202 038,
22.281 393)(113.202 143,22.281 673)
 P_2 :(113.202 056,22.281 709)(113.202 065,

22. 281 79) (113. 202 559, 22. 281 82)

P_3 : (113. 202 587, 22. 282 154)

并且, 嵌入水印时只对前两条曲线段嵌入, 所以 $|\{P_i\}| = 2$ 。

2.1.2 水印信息生成

根据分割后曲线的数目, 对 n 位水印信息 m 进行调制, 使每一条曲线都对应一个水印信息位。通过 $c = \lceil L/n \rceil$ 次嵌入水印信息 m , 得水印序列 $b = (b_1, b_2, \dots, b_{nc})$ 这样, 嵌入到 P_i 的水印信息为 $b_i, b_i = m_j$ (其中 $i/c = j$), 并处理得 $\tilde{b} = (\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_{nc})$, 其中

$$\tilde{b}_i = \begin{cases} -1 & b_i = 0 \\ 1 & b_i = 1 \end{cases} \quad (1)$$

若上例中, $L=3, n=2, c = \lceil 3/2 \rceil = 1, b = (1, 0)$, 则 $\tilde{b} = (1, -1)$ 。

2.1.3 水印信息嵌入

本文算法对每一水印信息位对应嵌入多个水印点 (watermark point), 在对每条曲线 P_i 嵌入水印信息位时, 先在两点间嵌入一个水印点, 再对嵌入点以前一结点为中心旋转一个小角度 θ , 使嵌入的水印点和原曲线的两个结点间略有偏移, 以保证该水印点不易被移去。同时, 为了使嵌入的水印点在两点间的位置不唯一, 又引入了另一个随机因数 p_k , 该随机因数由用户密钥计算得出, 以避免攻击者通过检测每三个点间的比例系数来移除水印点。由此, 水印信息嵌入的算法流程如下:

(1) 首先, 对分割后的图形, 在第 i 条曲线上 P_i 中的第 k 个结点 (vx_{ki}, vy_{ki}) , $1 \leq k \leq |P_i|$, 嵌入水印点 (vx'_{ki}, vy'_{ki}) :

$$\begin{cases} dx_k = (vx_{k+1,i} - vx_{k,i}) \tilde{b}_i \alpha p_k \\ dy_k = (vy_{k+1,i} - vy_{k,i}) \tilde{b}_i \alpha p_k \\ vx'_{k,i} = vx_{k,i} + dx_k \\ vy'_{k,i} = vy_{k,i} + dy_k \end{cases} \quad (2)$$

式中, α 为调制幅值, 调节嵌入结点位置的幅度, $0 < \alpha < 1$; p_k 由用户密钥产生, 调节嵌入结点位置的系数。

(2) 然后, 对水印点 $(vx'_{k,i}, vy'_{k,i})$ 围绕前一结点 $(vx_{k,i}, vy_{k,i})$ 旋转角度 θ , 得:

$$\begin{bmatrix} vx'_{k,i} \\ vy'_{k,i} \end{bmatrix} = \begin{bmatrix} vx_{k,i} \\ vy_{k,i} \end{bmatrix} + \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \times \begin{bmatrix} dx_k \\ dy_k \end{bmatrix} \quad (3)$$

在本文实例中, $|P_i| = 3$, 所以 $1 \leq k \leq 3$, 假设 p_k 取 $(1, -1, 1)$, $\alpha = 1, \theta = 0.06$, 则对于第 1 条曲线 P_1 中的第 1 个点 (113. 202 184, 22. 281 367) 嵌入水印

点的计算过程如下:

$$\begin{cases} dx_1 = (113. 202 038 - 113. 202 184) \times 1 \times 1 \times 1 \\ \quad = -0. 000 146 \\ dy_1 = (22. 281 393 - 22. 281 367) \times 1 \times 1 \times 1 \\ \quad = 0. 000 026 \end{cases}$$

$$\begin{cases} vx'_{1,1} = 113. 202 184 - 0. 000 146 = 113. 202 038 \\ vy'_{1,1} = 22. 281 367 + 0. 000 026 = 22. 281 393 \end{cases}$$

旋转水印点则可通过下式计算得到:

$$\begin{bmatrix} vx'_{1,1} \\ vy'_{1,1} \end{bmatrix} = \begin{bmatrix} 113. 202 184 \\ 22. 281 367 \end{bmatrix} + \begin{bmatrix} \cos 0. 06 & -\sin 0. 06 \\ \sin 0. 06 & \cos 0. 06 \end{bmatrix} \times \begin{bmatrix} -0. 000 146 \\ 0. 000 026 \end{bmatrix} = \begin{bmatrix} 113. 202 036 7 \\ 22. 281 384 \end{bmatrix}$$

2.2 水印提取及检测算法

2.2.1 原图及水印图的曲线分割

按照嵌入算法中的分割方法对原图 and 嵌入水印图进行对应的划分, 得到两个曲线集合 $\{P_i\}$ 和 $\{\tilde{P}_i\}$ 。设 $L_1 = |\{P_i\}|$, $L_2 = |\{\tilde{P}_i\}|$, 对于完整的水印图, 必有 $L_1 = L_2$ 成立; 对于受到裁剪攻击的水印图, 则 $L_1 > L_2$ 。此时, 应根据原图对水印图进行校正, 把被裁剪掉的曲线用空曲线补充上去, 使得 $L_1 = \tilde{L}_2$, 并标明被裁剪掉的曲线所对应的水印信息位无效。

2.2.2 水印信息提取

水印信息提取是嵌入的逆向过程, 即:

(1) 对照原图, 对水印点进行旋转还原, 其中, $0 \leq k \leq h_i$, h_i 为第 i 条曲线的结点数, 则在水印图相应的曲线中 $(vx'_{2k+1,i}, vy'_{2k+1,i})$ 为水印点, 对所有水印点进行如下旋转运算:

$$\begin{cases} dx'_k = vx'_{2k+1,i} - vx_{k,i} \\ dy'_k = vy'_{2k+1,i} - vy_{k,i} \end{cases}$$

$$\begin{bmatrix} vx'_{2k+1,i} \\ vy'_{2k+1,i} \end{bmatrix} = \begin{bmatrix} vx_{k,i} \\ vy_{k,i} \end{bmatrix} + \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{bmatrix} \times \begin{bmatrix} dx'_k \\ dy'_k \end{bmatrix} \quad (4)$$

(2) 对每一条曲线 P_i 和 \tilde{P}_i , 分别计算其除去最后一个点后的重心坐标:

$$\begin{cases} \overline{vx'_i} = \frac{1}{h'_i - 1} \sum_{k=1}^{h'_i-1} vx'_k \\ \overline{vy'_i} = \frac{1}{h'_i - 1} \sum_{k=1}^{h'_i-1} vy'_k \end{cases} \quad \begin{cases} \overline{vx_i} = \frac{1}{h_i - 1} \sum_{k=1}^{h_i-1} vx_k \\ \overline{vy_i} = \frac{1}{h_i - 1} \sum_{k=1}^{h_i-1} vy_k \end{cases} \quad (5)$$

式中, h_i 为原图形被分割后第 i 条曲线的长度(结点数); h'_i 为该曲线嵌入水印后的长度, 并对原图的每条曲线 i 的结点计算以下两个偏移量的值:

$$\begin{cases} \Delta x_i = \frac{1}{2h_i - 2} \sum_{k=1}^{h_i-1} (vx_{k+1} - vx_k) p_k \\ \Delta y_i = \frac{1}{2h_i - 2} \sum_{k=1}^{h_i-1} (vy_{k+1} - vy_k) p_k \end{cases} \quad (6)$$

(3) 按照式(7)~式(10)分别计算每条曲线纵横坐标的水印点。

① $\Delta x_i \neq 0 \& \Delta y_i \neq 0$

$$\begin{cases} qx_j = \sum_{\{i|l/c=j\}} \frac{\overline{vx'_i - vx_i}}{\Delta x_i} = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \cdot \alpha \\ qy_j = \sum_{\{i|l/c=j\}} \frac{\overline{vy'_i - vy_i}}{\Delta y_i} = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \cdot \alpha \end{cases} \quad (7)$$

② $\Delta x_i = 0 \& \Delta y_i \neq 0$

$$\begin{cases} qx_j = \sum_{\{i|l/c=j\}} \sum_{k=1}^{h_i-2} \frac{\overline{vx_{k,i} - vx_{k+1,i}}}{(vx_{k+1,i} - vx_{k,i}) p_k} \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \\ qy_j = \sum_{\{i|l/c=j\}} \frac{\overline{vy'_i - vy_i}}{\Delta y_i} \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \end{cases} \quad (8)$$

③ $\Delta x_i \neq 0 \& \Delta y_i = 0$

$$\begin{cases} qx_j = \sum_{\{i|l/c=j\}} \frac{\overline{vx'_i - vx_i}}{\Delta x_i} \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \\ qy_j = \sum_{\{i|l/c=j\}} \sum_{k=1}^{h_i-2} \frac{\overline{vy_{k,i} - vy_{k+1,i}}}{(vy_{k+1,i} - vy_{k,i})} \cdot p_k \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \end{cases} \quad (9)$$

④ $\Delta x_i = 0 \& \Delta y_i = 0$

$$\begin{cases} qx_j = \sum_{\{i|l/c=j\}} \sum_{k=1}^{h_i-2} \frac{\overline{vx_{k,i} - vx_{k+1,i}}}{(vx_{k+1,i} - vx_{k,i}) p_k} \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \\ qy_j = \sum_{\{i|l/c=j\}} \sum_{k=1}^{h_i-2} \frac{\overline{vy_{k,i} - vy_{k+1,i}}}{(vy_{k+1,i} - vy_{k,i})} p_k \\ = \sum_{\{i|l/c=j\}} \tilde{b}_i \alpha = c \tilde{b}_j \alpha \end{cases} \quad (10)$$

(4) 计算每条曲线的水印信息值:

$$\tilde{m}_j = \begin{cases} 1 & \frac{qx_j + qy_j}{2} > 1 \\ 0 & \frac{qx_j + qy_j}{2} < 1 \end{cases} \quad (11)$$

2.2.3 水印信息检测

在提取出水印信息后, 需检测水印的真伪。此过程是通过相关系数 $cor(m, \tilde{m})$ 来检测原始水印和提取水印的相似性, 具体算法见式(12)。

$$cor(m, \tilde{m}) = \frac{\sum_{i=0}^{n-1} (m_i \tilde{m}_i)}{\sqrt{\sum_{i=0}^{n-1} m_i^2} \sqrt{\sum_{i=0}^{n-1} \tilde{m}_i^2}} \quad (12)$$

式中, \tilde{m} 是从加水印后的图形中提取出来的水印信息; $cor(m, \tilde{m})$ 是水印信息 m 和 \tilde{m} 的相关系数。

2.2.4 水印鲁棒性测试理论与方法

在文献[10]中已从理论角度证明了各种图层的常规操作, 如: 平移、旋转或缩放等变形操作将不会影响水印图的恢复。本文水印鲁棒性测试方法主要讨论其是否具有抗裁剪和抗压缩性。

由于本算法在嵌入水印时, 之所以将一组水印信息多次嵌入到图层中, 使其遍布在整个图层中, 则考虑无论图层经历多少次裁剪, 只要分散嵌入的水印位有一个位存在, 仍可检测出水印信息。在考虑抗压缩性能时, 主要考虑如何抗击道格拉斯-普克压缩算法对矢量图形压缩的情景, 当对其阈值 T 取 0 为无损压缩, 大于 0 为有损压缩。

3 实验及结果分析

为了进一步验证本文提出的水印算法的性能, 特设计了测试实验方法与流程, 具体步骤如下: 首先, 选取多幅不同的 svg 地图作为矢量图形的实验样本, 对其用本文所述算法进行水印嵌入, 再按照相应的水印提取及检测算法进行测试。然后, 特以其中一幅地图为例, 对其进行各种裁剪攻击的模拟实验。与此同时, 定量评估如何确定算法中的旋转角度 θ 和相似度的阈值, 以及它们对整个算法性能的影响。在实验中, 嵌入的用户信息为“空间信息中心”6 个字, 再利用从用户公钥证书中提取出的用户密钥对其进行加密, 构成了 1 152 位的水印信息, 按照第 2.1 节中所述方法嵌入到矢量地图中。

3.1 没有任何攻击下的水印检测

如上所述, 水印提取算法是嵌入算法的逆过程,

在没有任何外部攻击的情况下,理论上应该是能够完整地提取出嵌入的原始水印。然而,常会因数据计算的误差,实际计算的相似度难以达到 100%。表 1 是 6 幅完整水印地图的检测结果,后 3 个样例相似度系数达到 1,前两个接近 1,只有第 3 个样例相似度系数仅为 0.911 5,存在近 9% 的误差,原因是地图 city_10. svg 曲线数目较少,而且每条曲线较短,使分割后的曲线总数较少,而被丢弃的线段也较多,造成水印分布不均,且嵌入次数较少,故相似度较低。

表 1 6 幅水印图的检测结果

Tab. 1 Detected results of 6 watermarked graphics

| 编号 | 矢量图文件名 | 分割前 曲线数 | 分割后 曲线数 | 水印相 似度 |
|----|---------------------|------------|------------|-----------|
| 1 | GD. svg | 512 | 14 908 | 0.998 3 |
| 2 | bndcity. svg | 448 | 7 649 | 0.998 3 |
| 3 | city_10. svg | 1 372 | 1 725 | 0.911 5 |
| 4 | sgl_riv_rank_8. svg | 3 933 | 5 863 | 1.0 |
| 5 | sgl_riv_rank_5. svg | 671 | 1 021 | 1.0 |
| 6 | city_14. svg | 22 | 317 | 1.0 |

3.2 几何攻击下的水印检测

几何攻击实验是以地图 GD. svg 为例,对其嵌入水印图进行旋转、平移和缩放等 3 种几何变形操作,再对其进行水印提取和相似度检测。由于 svg 对图形几何变形的操作均是利用 transform 函数实现,而该操作不会改变含水印图中的坐标点值,故不会对所嵌入的水印信息有任何改变。对地图 GD. svg 的横、纵坐标向右平移 500 000 个单位,再以地图为中心顺时针旋转 30 度,并相对于原图缩小 85%。两幅图所提取水印的相似度均为 0.996 527,说明上述几何变换操作所造成的几何攻击不会对水印信息提取有影响。

3.3 裁剪攻击下的水印检测

矢量图在网络传输或应用过程中,按需剪裁是常见的操作。因此,图形嵌入水印信息的抗剪裁能力非常重要。本文设计了 7 种剪裁攻击模式,再以嵌入水印的地图 GD. svg 为例分别进行了图 1 中的实验。其中,P1 为完整水印图,其他则以所设计的各种剪裁攻击模式进行处理而得到的结果。

通过对图 1 中 8 幅图分别进行水印提取及相似度的计算和分析,从表 2 可以看出:当剪裁攻击在 1/2 以上时,所提取水印信息的完整性或相似性均

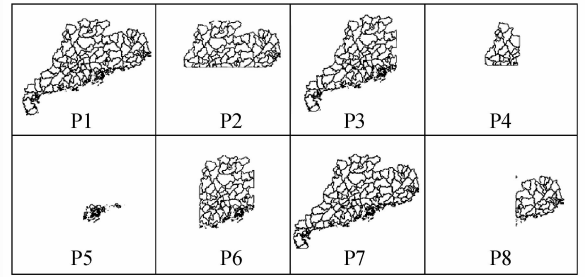


图 1 水印图 7 种剪裁攻击模式(P1 为完整水印图)

Fig. 1 The 7 kinds of cropping mode for watermarked graphics (P1 is the complete graphics)

可达到 90% 以上;而剪裁攻击在 1/3 以上时,则低于 50%,特别是模式 P5。尽管如此,通过第 3.6 节中阈值确定的方法计算相似度,仍可判断水印是否存在。

表 2 水印图 GD. svg 在 7 种剪裁攻击下的检测结果

Tab. 2 Detected results of GD. svg under the 7 kinds of cropping mode

| 裁剪模式 | 总结点数 | 被裁剪掉的 结点数 | 有效水印 位位数 | 水印相似度 |
|------|---------|--------------|-------------|-------------|
| P1 | 249 462 | 0 | 1 152 | 0.996 527 |
| P2 | 249 462 | 124 416 | 591 | 0.928 934 |
| P3 | 249 462 | 132 076 | 885 | 0.945 762 |
| P4 | 249 462 | 210 076 | 191 | 0.643 979 |
| P5 | 249 462 | 203 051 | 150 | 0.48 |
| P6 | 249 462 | 105 198 | 641 | 0.906 396 2 |
| P7 | 249 462 | 2 096 | 1 140 | 0.989 473 6 |
| P8 | 249 462 | 169 053 | 377 | 0.909 814 3 |

3.4 抗压缩能力实验

实验基于经典的道格拉斯-普克压缩算法(DP)算法。实验只对其进行了简单修改,用于测试水印的抗压缩能力,并与文献[10]中所提出的添加点的水印算法进行比较。实验样本采用矢量图 GD. svg,压缩算法的阈值 T 为相连的两条线段的斜率之差,即在连续的 3 个结点中,若中间结点偏离两边结点的连线达到指定的斜率差,则认为中间的结点是可压缩的结点。分别用文献[10]中的算法和本文算法对 GD. svg 嵌入水印(本文算法的旋转角度 θ 设置为 0.4,水印信息的长度为 1 152 位),设置不同的阈值 T ,对水印图进行有损压缩,再分别检测水印信息。从表 3 给出的实验结果可知,文献[10]中算法

嵌入的水印点随着压缩阈值的增大,压缩量迅速增大,检测的相似度也迅速地降低,本文算法嵌入的水印点由于经过了旋转操作,具有较好的抗压缩能力,压缩阈值增大到 0.3 时,仍保持 0.644 10 的水印信息相似度。

压缩能力越好,但地图精度损失越大;反之亦然。因此,需在应用中根据地图的精度要求适当选择 θ 的大小。在算法设计中, θ 角的大小是可调节变量,可视应用的精度需求确定之。换言之,确定 θ 角的大小,既要考虑矢量图对精度的要求,又要考虑其对水印信息性能的影响。

表 3 文献[10]算法与本文算法抗压缩能力的比较

Tab. 3 Comparison on anti-compression between the two algorithms in literature [10] and this paper

| 阈值 T | 被压缩掉的结点数 | | 水印相似度 | |
|--------|----------|--------|----------|----------|
| | 文献[10]算法 | 本文算法 | 文献[10]算法 | 本文算法 |
| 0.02 | 83 503 | 2 790 | 0.455 37 | 0.718 75 |
| 0.04 | 102 170 | 4 487 | 0.290 73 | 0.708 33 |
| 0.08 | 114 349 | 7 852 | 0.165 78 | 0.680 55 |
| 0.16 | 121 996 | 14 033 | 0.090 65 | 0.661 46 |
| 0.20 | 123 659 | 17 124 | 0.073 70 | 0.659 72 |
| 0.30 | 125 753 | 25 550 | 0.053 30 | 0.644 10 |

3.6 相似度阈值的确定

如上所述,本文的水印检测算法采用矢量的夹角余弦值来表示两水印序列间的相似度。将其用于矢量图形数字水印检测时,检测对象是可能受过各种攻击的水印图形,也可能是未包含水印的图形。例如:受过剪裁攻击的水印图形中提取的水印信息可能只是原水印的一部分,甚至只是一小部分,理想的检测算法应能最大程度地检测部分提取水印与完整原始水印对应部分的相似性。

3.5 旋转角度的定量分析

在本文的水印嵌入算法中,水印点嵌入后需围绕其邻近的结点旋转角度 θ ,该角度的大小是否对嵌入水印图的精度有影响,且对水印图的抗压缩和抗剪裁攻击能力有何影响?实验仍以地图 GD.svg 为例,不断改变角度 θ 的大小,对其完整水印图形和剪裁了约 50% 后的图形进行水印信息提取及相似度计算。结果表明(如表 4 所示),旋转角度 θ 对完整嵌入水印信息的图形检测几乎无影响,而对裁剪攻击的影响也不是十分明显,其角度 θ 的改变只对有损压缩和地图精度有较大的影响。即 θ 越大,抗

为了解决部分水印信息检测问题,则利用确定水印相似度阈值的方法提升检测算法的应用价值。分析噪音信息和受攻击剩余水印信息均为随机序列,则可通过计算所提取的水印与大量随机生成水印的相似度来确定阈值。从概率论的角度看,通过以上方法计算的阈值应当与水印序列的长度有关。因此,试验以地图 GD.svg 为例,假定嵌入的水印为 W ,提取的水印为 W' , W 的长度为 L ,按照不同的长度 l 截取 W' ,对应生成长度为 l 的 999 个随机水印序列(W_i),计算 W 与 W' , W_i 的相似度,共有 100 个相似度, ms 为最大者,表 5 是 l 取不同的值及对应的 ms 的大小。

表 4 旋转角 θ 对水印信息检测及抗裁剪能力的影响

Tab. 4 Infection of the angle θ to watermarked detection under cropping attacks

| 旋转角度 θ (弧度) | 裁剪前的 相似度 | 裁剪掉的 结点数 | 裁剪后的 相似度 |
|-----------------------|-------------|-------------|-------------|
| 0 | 0.996 53 | 56 158 | 0.945 76 |
| 0.12 | 0.998 26 | 56 156 | 0.948 02 |
| 0.24 | 0.996 53 | 56 154 | 0.945 76 |
| 0.36 | 0.998 26 | 56 155 | 0.943 5 |
| 0.48 | 0.996 53 | 56 155 | 0.948 02 |
| 0.6 | 0.996 53 | 56 156 | 0.945 76 |
| 0.72 | 0.998 26 | 56 158 | 0.950 28 |
| 0.84 | 0.996 53 | 56 158 | 0.945 76 |
| 0.96 | 0.996 53 | 56 159 | 0.943 5 |
| 1.08 | 0.998 26 | 56 159 | 0.945 76 |

表 5 不同长度的水印与 999 个随机水印的最大相似度

Tab. 5 The maximum similarity for the extracted watermark and other 999 random ones at different length

| L | 25 | 50 | 100 | 200 | 400 | 800 | 1 000 |
|------|-----|-------|------|------|------|------|-------|
| ms | 0.5 | 0.499 | 0.39 | 0.28 | 0.23 | 0.21 | 0.198 |

实际上,水印信息的长度不应太小,而受到裁剪攻击的水印图形若有效水印位过少,则已无实际应用价值。因此,考虑设计提取最小量和最大量水印信息位长度设定在 25 到 1 000 之间。在对以上的数据进行插值处理后,可根据每一个水印长度对应的相似度阈值做出一条阈值曲线(threshold curve),如图 2 所示,可在水印检测时,作为不同长度的水印信息相似度阈值的参考。

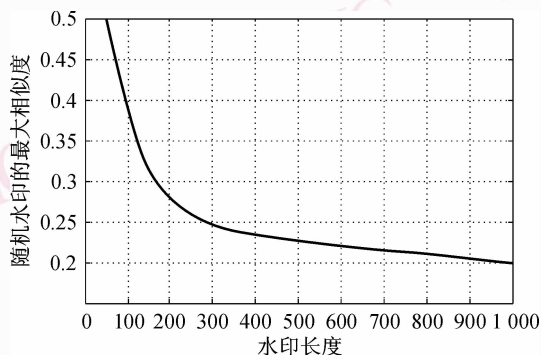


图 2 阈值曲线

Fig. 2 The threshold Curve

4 结 论

提出了一种针对矢量图形的基于曲线分割的数字水印算法,算法设计主要为适应图形几何精度要求高的应用,故基于添加点算法,对图形中每条曲线进行分割后,将水印位和用户信息嵌入到矢量图中。实验结果表明,该算法的主要特点在于:

(1)在水印信息嵌入后,又对其进行 θ 角度的旋转,选择适当的旋转角度既能加强算法抗压缩的鲁棒性,又可保证矢量图形的精度;

(2)为增强水印信息检测和提取的抗攻击能力,实验建立了水印信息长度与相似度阈值经验曲线,便于水印信息检测的自动化处理流程中设定参数。

然而,该算法仍有需改进的问题:当对每条曲线除起始点和结束点外的所有结点都嵌入一个水印点后,使图形所嵌入的水印位过多,以致矢量图形文件增大较多;若采用只对一条曲线的特征点嵌入水印位时,又需以降低水印鲁棒性为代价。另外,算法还需进一步考虑抗有损压缩的能力,以及盲水印算法的应用。

参考文献 (References)

1 Michael Voigt, Bian Yang, Christoph Busch. Reversible watermarking of 2D-vector data [A]. In: Proceedings of the 2004 Workshop on

Multimedia and Security [C], Magdeburg, Germany, 2004: 160-165.

2 Doncel Victor R, Nikolaidis Nikos, Pitas Ioannis. Watermarking polygonal lines using an optimal detector on the fourier descriptors domain [EB/OL]. <http://www.ee.bilkent.edu.tr/~signal/defevent/papers/cr1493.pdf>.

3 Li Yuan-yuan, Xu Lu-ping. Vector graphical objects watermarking scheme in wavelet domain [J]. Acta Photonica Sinica, 2004, 33(1): 97-100. [李媛媛, 许录平. 矢量图形中基于小波变换的盲水印算法 [J]. 光子学报, 2004, 33(1): 97-100.]

4 Zhang Qin, Xiang Hui, Meng Xiang-xu. Watermarking vector graphics based on complex wavelet transform [J]. Journal of Image and Graphics, 2005, 10(4): 494-498. [张琴, 向辉, 孟祥旭. 基于复数小波域的图形水印方法 [J]. 中国图象图形学报, 2005, 10(4): 494-498.]

5 Michael Voigt, Christoph Busch. Watermarking 2D- vector data for geographical information systems [A]. In: Proceedings of the 4th Conference on Security and Watermarking of Multimedia Contents [C], San Jose, California, USA, 2002: 621-628.

6 Michael Voigt, Christoph Busch. Feature-based watermarking of 2D-vector data [A]. In: Proceedings of the 5th Conference on Security and Watermarking of Multimedia Contents [C], Santa Clara, California, USA, 2003: 359-366.

7 Gerrit Schulz, Michael Voigt. A high capacity watermarking system for digital maps [A]. In: Proceedings of the Workshop on Multimedia and Security [C], Magdeburg, Germany, 2004: 180-186.

8 Ohbuchi Ryutarou, Ueda Hiro, Endoh Shul. Robust watermarking of vector digital maps [A]. In: Proceedings of IEEE Conference on Multimedia and Expo [C], Lausanne, Switzerland, 2002: 577-580.

9 Wang Xun, Lin Hai, Bao Hu-jun. A robust watermarking algorithm for vector digital mapping [J]. Journal of Computer Aided Design & Computer Graphics, 2004, 16(10): 52-56. [王勋, 林海, 鲍虎军. 一种鲁棒的矢量地图数字水印算法 [J]. 计算机辅助设计与图形学学报, 2004, 16(10): 52-56.]

10 Wang Wei, Li Yan. A robust watermarking algorithm for 2D vector graphics [J]. Journal of Image and Graphics, 2007, 12(4): 200-205. [王伟, 李岩. 一种鲁棒性的 2D 矢量图形水印算法 [J]. 中国图象图形学报, 2007, 12(4): 200-205.]

11 Zhu Chang-qing, Yang Cheng-song, Li Zhong-yuan. An anti-compression watermarking algorithm for vector map data [J]. Journal of Zhengzhou Institute of Surveying and Mapping, 2006, 23(4): 50-52. [朱长青, 杨成松, 李中原. 一种抗数据压缩的矢量地图数据数字水印算法 [J]. 测绘科学技术学报, 2006, 23(4): 50-52.]