

一种鲁棒性的2D矢量图形水印算法

王伟¹⁾ 李岩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 为了解决空间信息网络传输和空间信息共享的版权保护问题,提出了一种具有鲁棒性的2D矢量图形的水印算法。在GIS中,矢量图形是其最基本的表达方式,常需针对2D矢量图形的大数据集进行水印嵌入处理,本方法先将空间矢量图层按照自身所含的多边形特征进行分解;对分解后的矢量多边形进行分析,选择合适的多边形的线段,在其顶点处嵌入水印;而提取水印的过程,则是对原矢量图形和水印图形进行分析,根据嵌入水印顶点的位置或坐标值得到水印信息的序列。实验结果证明,该方法对于空间信息的常规图形操作,如矢量图层的坐标变换、平移、旋转、缩放,以及图形的剪切,点的添加删除等,均具有较强的鲁棒性。

关键词 数字水印 矢量图形

中图法分类号: TP301.6 **文献标识码**: A **文章编号**: 1006-8961(2007)02-0200-06

A Robust Watermarking Algorithm for 2D Vector Graphics

WANG Wei¹⁾, LI Yan²⁾

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

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

Abstract In order to protect the copyright of the transmission and sharing of the geospatial information, this paper presents a robust watermarking algorithm for 2D vector graphics. In GIS system, vector graphics is a basic type of information and a large dataset of 2D vector graphics are often needed to perform watermark embedding for the security purpose. The processing method, includes: Firstly, decompose the vector graphics into the polygons that layer has; Then analyze these polygons and embedded the watermark in the start and end points of the selected line. In the procedure of extracting watermark, the original graphics and the watermarked graphics were analyzed and extracted the sequence of the watermark information according to the position or the coordinate of the vector. The experimental results show that the method is robust against the general operations of the spatial graphics such as transformation, translation, rotation, scaling, cropping and vertex insertion/removal etc.

Keywords watermarking, vector graphics

1 引言

随着多媒体技术和网络技术的迅速发展,对广泛应用于GIS中的空间信息数据的版权保护成为一个迫切需要解决的问题。

从空间信息模型两分法的理论出发,空间数据主要可分为栅格数据和矢量图形数据。与栅格数据相比,矢量数据有以下两个优点:(1)数据量小、精度较高;(2)可以任意地放大缩小而不失真。矢量

图形数据以其明显的优势被视为网络空间信息发布,或网络GIS的重要数据格式,而对矢量图形数据的知识产权保护则显得尤其重要。

近几年,数字水印技术作为一种在开放的网络环境中保护数字产品版权的新技术,逐步得到发展。但是,数字水印技术大部分都集中在数字图像和视频方面。近年来,逐渐推出一些有关图形数据数字水印方法研究的文章。例如:Vassilios^[1]针对多边形数据,使用1维离散傅里叶变换(DFT),将水印嵌入到DFT变换的幅值系数中,其结果会引起矢量图

收稿日期:2006-10-10;改回日期:2006-11-11

第一作者简介:王伟(1981~),男,华南师范大学计算机学院硕士研究生。从事空间信息理论与方法的研究。E-mail: wwdream2003@

163.com

形的轻微失真;李^[2]和张^[3]提出了基于小波变换的水印方法,其中,李^[2]通过离散小波变换将顶点序列分解成不同空间和频率上的复值系数,并根据水印的大小与小波系数之间的关系把水印嵌入到小波系数的幅值中;张^[3]提出了基于复数小波域的2维矢量图形的水印算法,利用 Dual Tree 复数小波变换的优点,将水印嵌入相对坐标线的复数小波中。这两种方法都具有良好的不可见性和对通常的图形几何变换具有较强的鲁棒性,但是都没有充分考虑对图形剪切的鲁棒性;Henry^[4]提出了在矢量图形的线段中插入点、改变线段的长度、方向和属性等方法作为水印信息,其插入点的位置和改变线段长度的位置太单一,使之不具有较强的鲁棒性,而改变线段的方向和属性的方法并不适合于精确的矢量数据;文献[5]~[7]都是利用坐标值的容差(Tolerance)嵌入水印,其中,Voigt^[5]采用由 $-tolerance$ 和 $+tolerance$ 的值组成的 PN-sequence (pseudo-noise sequence) 作为水印信息,将其嵌入到2D矢量图形的坐标数据中,通过 PN-sequence 和添加水印数据的关系来发现水印;此外,Voigt^[6]把矢量图形分为网格,将图形分为几个子区域,通过在容差(Tolerance)范围内移动坐标点来添加水印;Ohbuchi^[7]提出一种基于四叉树划分矢量地图的空域水印方法,根据顶点的密度将图形划分为不同大小的矩形框,在矩形内通过移动坐标点来嵌入水印信息,这3种方法都对在容差域内改变坐标值的攻击具有很好的鲁棒性,但这3种水印方法均改变了原始图形的坐标值。

本文在 Henry^[5] 和 Voigt^[6] 方法的基础上,提出了一种新的鲁棒性矢量图形水印算法,即借鉴矢量图形分区理念,首先按照图层自身包含的多边形进行分解,合并点的数目过于少的多边形,即将图层进行子区域分解。再将每个子区域中的多边形分解为多条边或多条曲线,为了提高水印的隐蔽性,只选取每一个子区域中的某一条合适的边或曲线用来嵌入水印信息,并且利用定义的 key 生成随机的水印序列,在所选取的边或曲线顶点的容差范围($-tolerance$, $+tolerance$)的 $-\alpha$ 或者 α 处嵌入新的点作为水印信息,最后得到嵌入水印信息的矢量图形。

2 水印算法

2.1 图形的分割

空间数据的每个矢量图层均可被称之为大数据

集,若要对其进行高效率的水印处理,则需对矢量图形进行分割处理。文献[6]、[7]分别利用网格和矩形框进行分割,可能造成一个多边形的水印和相邻的多边形有联系,如果攻击者只剪切其中某一个多边形区域的信息,便有可能造成水印的丢失。此外,在矢量图层中多边形的公共边问题是嵌入水印的一个重要障碍,它可能造成由于水印的重复嵌入,而导致水印提取的失败。因此,本文在讨论水印算法之前,先提出将矢量图层按照自身的内部多边形进行分割和基于多边形的最小边界矩形框(MBR, minimum boundary rectangle)处理公共边的方法,从而提高水印的鲁棒性。

首先,将矢量图层按照本身所包含的多边形分割成为若干个多边形区域,如果某一个多边形所包含的点数少于要嵌入水印的长度 n , 就将此多边形合并到点数最少的相邻多边形中。这样,原矢量图层就被分割为若干包含一个或者多个多边形的子区域;然后,求出每一个子区域的 MBR, 根据在 MBR 中包含的相邻多边形,获得这些多边形的公共边集合。图1虚线矩形框为多边形 A 的 MBR, 根据包含的多边形,获得多边形 A 的公共边 a_1 、 a_2 、 a_3 、 a_4 、 a_5 的信息。

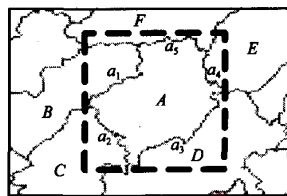


图1 图形分割

Fig. 1 Graphics division

2.2 水印的嵌入

第1个嵌入水印信息的子区域将会决定其他区域添加水印的情况,故选择整个矢量图层的质心所在的子区域作为第1个嵌入水印的区域。

为了提高嵌入和提取水印的效率,只对每一个子区域中一条边嵌入水印信息即可,它满足:

(1) 未添加水印的边,即相邻的多边形没有对其嵌入水印信息;

(2) 含有点的数目大于水印长度 n 。如果子区域所有边中点的数目少于 n , 逆时针或者顺时针合并相邻的边,直到满足点大于等于 n 时为止;

(3) 满足以上两个条件的边集合中,选取离子

区域质心 $(\bar{x} = \frac{1}{u} \sum_{i=1}^u x_i, \bar{y} = \frac{1}{u} \sum_{i=1}^u y_i)$ (其中 (x_i, y_i) 为子区域中多边形的点坐标值) 最远点所在的边。

假设满足上述 3 个条件的边有 u 个顶点, 用 $\bar{m} = (m_1, m_2, \dots, m_u)$ 表示坐标值。用 $w = (w_1, w_2, \dots, w_n)$ 表示 n 位水印的信息, 其中 $w_i \in \{-1, 1\}$, 取值由用户自定义的密钥 key 产生。用 $b = (b_1, b_2, \dots, b_u)$ 表示 u 个顶点的水印序列值, 因为 $u > n$, 故要重复嵌入水印 $c = \lceil u/n \rceil$ 次, 且满足 $b_i = w_j$, 其中, $jc \leq i \leq (j+1)c, i \in [1, u], j \in [1, n]$ 。

本文将在顶点的容差范围内, 嵌入一个新的坐标点作为水印。这样可以防止由图形坐标精度取舍的原因而造成图形的失真。用 $\bar{m}' = (m'_1, m'_2, \dots, m'_u)$ 表示增加点的坐标值, 与原图形顶点 $\bar{m}$ 关系如下:

$$m'_1 = m_1 + \alpha(m_2 - m_1) \quad i=1$$

$$m'_u = m_u - \alpha(m_u - m_{u-1}) \quad i=u$$

$$m'_i = \begin{cases} m_i + \alpha(m_{i+1} - m_i) & b_i = 1 \\ m_i - \alpha(m_{i+1} - m_i) & b_i = -1 \end{cases} \quad i \in [2, u-1]$$

其中, α 是坐标容差范围的绝对值。当 $b_i = -1$ 时, 在 m_i 的 $(-Tolerance, 0)$ 的 $-\alpha$ 处嵌入点 m'_i ; 当 $b_i = 1$ 时, 在 m_i 的 $(0, Tolerance)$ 的 α 处嵌入点 m'_i 。

经过以上操作, 就可对含有 u 个点的多边形的线段, 在容差范围内嵌入 u 个新的点作为该图形的水印信息。

2.3 水印的提取

水印的提取就是检验图层是否含有嵌入的水印信息, 从而证明对图层是否拥有版权。

为了能够成功从检验图层中提取出水印, 需要进行与原始图层间几何配准。首先, 对检验图层进行同样的分割, 利用嵌入水印取边的过程获取添加水印的边; 然后利用原始图层和检测图层中的边起始点、终点及质心集合作为几何配准的控制点, 则可利用式(1)进行几何校正:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = A \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (1)$$

其中, $\{\tilde{x}, \tilde{y}\}$ 和 $\{x, y\}$ 分别表示水印图层和原始图层中的坐标值, $\{a_{ij}\}$ 是图层的缩放、旋转和平移的参数。使用文献[7]的 IAM (iterative alignment-minimization) 算法, 进行精细的几何校正, 从而可检测原始图层与矢量图层点之间一一对应关系, 以便

检查出删除或者添加的坐标点。

随后, 就可提取水印信息。首先, 根据原始图层所确定地添加水印的序列 $\{w_i\}$; 然后, 对检测图层进行分析, 获取添加水印的边, 利用下式求出水印图层中的水印信息 $\{w'_i\}$

$$w'_i = \left(\frac{m'_{2i} - m'_{2i-1}}{m'_{2i+1} - m'_{2i-1}} \right) / \alpha \quad i \in [1, u] \quad (2)$$

最后, 通过相似度来检测原始水印和提取水印的相似性:

$$\text{sim}(w, w') = \frac{\sum_i (w_i, w'_i)}{\sqrt{\sum_i (w_i, w_i)} \sqrt{\sum_i (w'_i, w'_i)}} \quad (3)$$

其中, $i \in [0, u-1]$ 。

如果相似度大于某一阈值 T , 则认为提取的水印正确, 否则水印不存在。 T 取得经验值 0.75。

3 水印鲁棒性的讨论

水印必须具有一定的鲁棒性, 才会真正起到保护数据版权和来源认证的作用。因此, 需要讨论本文提出的水印算法是否在经过常用的图层几何变换, 如: 平移、旋转、剪切、缩放、添加和删除坐标点等操作后仍具有很好的鲁棒性。

3.1 平移、旋转、缩放

将图层分别沿 x 轴和 y 轴移动 T_x 和 T_y 的距离可以利用式(1)解决平移问题, 即

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (4)$$

2D 图层的旋转是指将 p 点绕坐标原点转动某个角度 θ 到新的点 p' 。同样可以利用式(1)解决旋转问题, 得到:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & \pm \sin(\theta) & 0 \\ \mp \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (5)$$

图层的缩放是对 p 点相对于原点沿 x 轴缩放 S_x , 沿 y 轴缩放 S_y , 也可利用式(1)解决缩放问题:

$$\begin{bmatrix} \tilde{x} \\ \tilde{y} \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad (6)$$

由此, 可以证明基于式(1)的平移、旋转和缩放等几何变换, 均不会对水印图层造成影响。

3.2 坐标点的添加删除

如前所述,首先获取原始图层要嵌入水印部分的坐标值 $\bar{m} = (m_1, m_2, \dots, m_u)$ 和检验图层水印部分的坐标值 $\bar{p} = (p_1, p_2, \dots, p_k)$ (其中由于攻击者可能添加或者删除坐标点,因此 k 可能大于也可能小于 $2u$),然后使用文献[7]的IAM算法可以检测处于原始图层与检测图层之间各点一一对应关系,从而检测出删除或添加的坐标点。

根据水印添加的方法过程,在未经破坏的水印图层中,所添加水印的线段上每一个节点处容差范围内应该有另外一个点(即作为嵌入水印信息的点),因此在检测图层中,如果存在与原始图层中的点 $m_i (i \in [1, u])$ 相对应的点 $p_j (j \in [1, k])$, 它的情况有两种:

(1) p_j 的容差的 α 处存在一点,例如存在的点为 p_i , 则使 $m'_{2i} = p_j, m'_{2i+1} = p_i$;

(2) p_j 的容差的 α 处没有点存在,即此水印信息可能被破坏,则使 $m'_{2i} = p_j, m'_{2i+1} = p_i$ 。

如果在检测图层 $\bar{p} = (p_1, p_2, \dots, p_k)$ 的坐标值中没有点与 m_i 相对应的情况,则使 $m'_{2i} = m_i, m'_{2i+1} = m_i$ 。

最终可以根据 $\bar{m} = (m_1, m_2, \dots, m_u)$ 和 $\bar{p} = (p_1, p_2, \dots, p_k)$ 获得有包含水印信息的 $2u$ 个坐标值的 $\bar{m}' = (m'_1, m'_2, \dots, m'_{2u})$, 然后根据式(2)求出水印图层中的水印信息 $\{w'_i\}$, 最后根据式(3)求出相似度。因此,适当的点的添加与删除并不影响水印信息的提取。

3.3 图形的剪切

如果攻击者只需要图层某一部分的图形数据,而不是整个图层的数据,就会对图层进行剪切操作。由于本文方法是对图层内每一个多边形嵌入水印信息,则无论剪裁图层的任何部位,只要剪切图层中含有至少一个多边形,均可很好地检测出水印信息。此外,如果矢量图层中包含某一条很长的河流或者很大的区域,即一个多边形含有很多的节点时,则可以将此线段或多边形再次进行分割,分割成一些相对较小的区域或较短的线段,对每一个分区或分段嵌入水印信息后,再将它们合并在一起组成原线段或多边形。

4 实验结果分析

SVG 作为新一代网络矢量数据发布的标准,正

在被广泛应用于 GIS 应用系统中。然而,由于 SVG 是以开放源的数据文档呈现在网络环境中,必然涉及到数据安全和数据版权的问题。因此,本研究以 SVG 矢量图形为例,进行算法的实验研究,以证明其实用性和有效性。

4.1 实验步骤

实验数据选择广东省县市级行政边界的 SVG 矢量图层做实验,具体实验步骤如下:

(1) 将原矢量图层(如图2(a))按照2.1节中图层分割的方法进行分割,原始图层含有132 185节点,共121个多边形。合并节点的数目少于水印长度 n 的多边形,而得到99个子区域。

(2) 在每一个区域中选取满足条件的边界线段,按照2.2节中水印嵌入的方法对其嵌入水印,嵌入水印后的矢量图形如图2(b)。



(a) 原矢量图层

(b) 嵌入水印后的矢量图形

图2 原始图形与嵌入水印后的图形

Fig. 2 Original and watermarked graphics

(3) 对添加水印后的图形进行一系列的攻击,按照2.3节中水印提取的方法检验水印的鲁棒性。

在水印嵌入过程中,需要注意嵌入水印的长度 n ,如果嵌入的水印信息太短或很少不利于提取,但若太长,水印的鲁棒性就会变差;同时,水印太长,会有很多线段所含的节点数目少于 n ,需进行大量的合并操作,从而降低添加水印和检测水印的效率。水印的长度 n 可根据图层的情况而定,此次实验中选取 $n=64$,即64位用户定义的key生成的随机的由 $\{-1, 1\}$ 组成的序列。

4.2 鲁棒性验证

验证是对实验数据中的子区域进行分析,图3为图2(a)矢量图层中的一个子区域,图3(a)为含有1552个点未嵌入水印的原始图形,分割为6条边界线段,满足选择边的3个条件后,选择了一条有224个节点的边来嵌入水印;图3(b)为嵌入水印后的图形,图形上的虚线矩形框是添加水印信息的区域,右边大的矩形框是嵌入水印区域放大后的部分区域;其中A、B、C为原始图形的点,a、b、c为添加的

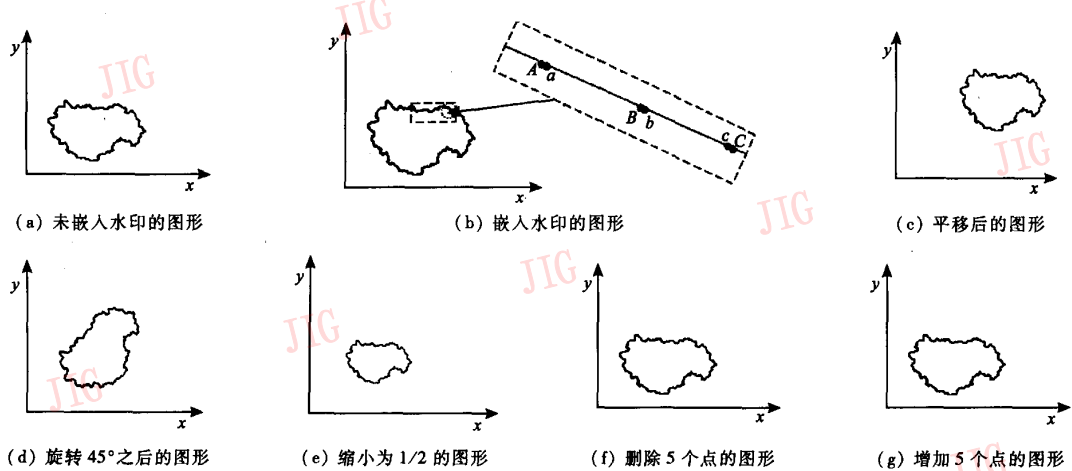


图 3 嵌入水印子区域图形经过攻击的图形
Fig. 3 The watermarked graphics after being destroyed

点,经过水印处理后此多边形图形有1776个点,其中224个为水印信息;图3(c)为水印图形平移后的图形;图3(d)为水印图形旋转45°角后的图形;图3(e)为水印图形缩小1/2后的图形;图3(f)为水印图形经过删除5个点攻击后的图形;图3(g)为水印图形经过增加5个点攻击后的图形。

针对上述实验结果,表1给出了在3种情况下按照提取水印的方法所得到的相似度,阈值取0.75,由表中的数据可以看出,本算法对水印图形平移、旋转和转移具有很好的鲁棒性。

表 1 鲁棒性测试结果(阈值取 0.75)
Tab.1 The results of robust

	水印图形	平移	旋转	缩放
相似度	0.98	0.975	0.95	0.953

同时,表2给出了对嵌入水印图形分别进行了添加和删除5个和30个点的鲁棒性检测结果。因为攻击者要获取图形的数据,因此对水印图形添加和删除点的攻击一般不会造成图形的畸变。由于添加删除节点位置和数目不同,提取水印的相似度和效率则会有很大差异。经过多次不同数量点的删除和添加的实验,得到的如下结论:如果添加的点恰在

表 2 添加删除点鲁棒性测试结果(阈值取 0.75)

Tab.2 The robust after appending and deleting points

水印图形	添加 5 个点	添加 30 个点	删除 5 个点	删除 30 个点
相似度	0.98	0.95	0.915	0.893

α 处,水印提取相似度和效率就会随着添加点的增多而降低,但可成功提取水印信息;如果是在非水印区域,就可利用文献[7]的IAM算法删除这些多余的点,可以成功提取水印信息;如果删除的点在水印区域,水印提取的成功率会随着删除点的增多而降低。

对矢量图层剪切的方式有很多种,实验只选取其中一部分,图4是对图3(b)的两种剪切的方式,经实验证明只要剪切后的图层包含一个完整的多边形或线段图形,就可以成功的提取水印信息。

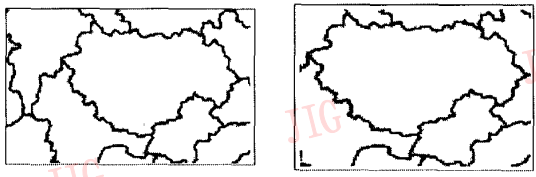


图 4 嵌入水印子区域图形经过剪切攻击的图形
Fig.4 The watermarked graphics after being cropped

5 结 论

本文尝试了一种新的矢量图层数字水印算法。该算法将矢量图层按照自身的多边形分割为子区域,对每个子区域中某一条边顶点的容差范围内嵌入新的点作为水印信息。实验结果证明,此算法对图形的几何变换、点的添加、删除和剪切都具有较好的鲁棒性,可以很好的起到矢量图形数据的版权保护作用。然而,本文算法是非盲水印算法,如何实现

矢量图形的频域中嵌入水印的算法和盲水印算法将是今后工作的重点。

参考文献 (References)

- 1 Solachidis Vassilios, Nikolaidis Niikos, Pitas Ioannis. Watermarking polygonal lines fourier descriptions [A]. In: Processings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'2000) [C], Istanbul, Turkey, 2000, 5: 1955 ~ 1958.
- 2 Li Yan-yan, Xu Lu-ping. Vector graphical objects watermarking scheme in wavelet domain [J]. ACTA Photonica Sinica, 2004, 1(33): 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, 1(33): 97 ~ 100.]
- 3 Zhang Qin, Xiang Hui, Meng Xiang-xu. Watermarking vector graphics based on complex wavelet transform [J]. Journal of Image and Graphics, 2005, 4(10): 12 ~ 15. [张琴, 向辉, 梦祥旭. 基于复数域的图形水印方法[J]. 中国图象图形学报, 2005, 4(10): 12 ~ 15.]
- 4 Sonnet Henry, Isenberg Tobias, Dittmann Jana, et al. Illustration watermarks for vector graphic [A]. In: Proceedings of 11th Pacific Conference on Computer Graphics and Applications (PG2003) [C], Canmore, Canada, 2003: 73 ~ 82.
- 5 Michael Voigt. Christoph busch watermarking 2D-vector data for geographical information system [A]. In: Processings of IS&T/SPIE Electronic Imaging [C], Washington, America, 2002, 4675: 621 ~ 628.
- 6 Michael Voigt, Christoph Busch. Feature-based watermarking of 2D-vector data [A]. In: Processings of SPIE [C], Santa Clara, 2003, 5020: 359 ~ 366.
- 7 Ryutarou Ohbuchi, Hiroo Uedda, Shuh Endoh. Robust watermarking of vector digital maps [A]. In: Processings of IEEE International Conference on Multimedia and Expo 2002 (ICME 2002) [C], Lausanne, Switzerland, 2002, 8: 1 ~ 4.