

## 基于立体视觉掩蔽的自恢复非对称立体图像水印

骆挺<sup>1,2</sup>, 蒋刚毅<sup>2,3</sup>, 王晓东<sup>2</sup>, 郁梅<sup>2,3</sup>, 彭宗举<sup>2</sup>, 邵枫<sup>2</sup>

(1. 宁波大学 科学技术学院, 浙江 宁波 315000; 2. 宁波大学 信息科学与工程学院, 浙江 宁波 315000;  
3. 南京大学 计算机软件新技术国家重点实验室, 江苏 南京 210093)

**摘要:** 为了认证立体图像的完整性, 提出了一种基于立体视觉掩蔽的非对称立体图像水印方法。首先, 根据立体视觉特性, 建立立体视觉掩蔽计算模型, 设计了立体图像左右视点嵌入不同容量的非对称水印; 其次, 使用小波变换系数计算不同类型的恢复水印, 将其分别嵌入 2 个或 3 个不同映射块, 从而提高篡改恢复质量。仿真实验结果证明, 该方法能够有效检测和恢复、拼接、粘贴等不同类型的篡改, 并且恢复质量高于现有以及扩展的单图像水印方法。

**关键词:** 立体图像; 非对称立体图像水印; 立体视觉掩蔽; 篡改恢复

中图分类号: TN919.81

文献标识码: A

文章编号: 1000-436X(2013)06-0057-12

## Binocular suppression based asymmetrical stereo image watermarking with self-recovery

LUO Ting<sup>1,2</sup>, JIANG Gang-yi<sup>2,3</sup>, WANG Xiao-dong<sup>2</sup>, YU Mei<sup>2,3</sup>, PENG Zong-ju<sup>2</sup>, SHAO Feng<sup>2</sup>

(1. College of Science and Technology, Ningbo University, Ningbo 315000, China;

2. Faculty of Information Sciences and Engineering, Ningbo University, Ningbo 315000, China;

3. National Key Lab of Software New Technology, Nanjing University, Nanjing 210093, China)

**Abstract:** To authenticate the integrity of stereo image, an asymmetrical stereo image watermarking scheme based on binocular suppression was proposed. Firstly, according to the features of stereo vision, the computation of binocular suppression model was established. The concept of asymmetric stereo image watermarking was defined that two views of stereo image were embedded with different data payloads. Then, discrete wavelet transform was employed to generate different types of recovery watermark for each block, and their copies were embedded into two or three different mapping blocks to improve the quality of recovery. The experimental results demonstrate that the proposed method can detect and recover different kinds of tamper effectively, such as collage attack, and the recovered quality is better than those of other existing and extended mono-image watermarking methods.

**Key words:** stereo image; asymmetric stereo image watermarking; binocular suppression; tamper recovery

### 1 引言

随着网络与计算机技术的发展, 人们对视频服务的要求越来越高。三维电视(3DTV) 能够提供自然场景的深度信息, 给观众以身临其境的感觉, 越来越受到人们的欢迎<sup>[1,2]</sup>。同时, 立体图像作为三维视觉中典型的表现形式之一, 其安全需要有效的保

证<sup>[3]</sup>。如何保护立体图像的内容完整性已成为通信安全领域亟待解决的问题<sup>[4]</sup>, 数字水印作为解决该问题的有效技术之一, 正被学术界关注。

数字认证水印技术根据功能分为 2 类: 第一类只具有篡改的定位功能<sup>[5,6]</sup>, 然而更多的用户希望能够将篡改的区域近似地恢复, 因此既具有定位又具有恢复功能的第二类认证水印技术更受到人们的

收稿日期: 2012-08-21; 修回日期: 2013-03-04

基金项目: 国家自然科学基金资助项目(61071120, 61171163, 61111140392, 61271270, 612712700); 国家科技支撑计划基金资助项目(2012BAH67F01)

Foundation Items: The National Natural Science Foundation of China (61071120, 61171163, 61111140392, 61271270, 612712700); National Key Technology R&D Programme of China (2012BAH67F01)

注意。文献[7]提出了一种基于层次的水印方法，每一图像分块的恢复水印嵌入唯一的映射块。然而，当分块和其映射块被同时篡改时，该分块不能使用嵌入映射块的水印恢复篡改，因此该方法对面积稍大的篡改恢复质量较差。为克服文献[7]的缺陷，文献[8]和文献[9]将每个分块的恢复水印分别嵌入 2 个不同的映射块，当图像篡改区域稍大时，一定程度上改进了恢复的质量。此外，文献[7]和文献[8]的分块篡改的认证水印相互独立，难以有效抵抗拼接粘贴攻击(collage attack)<sup>[10]</sup>。

文献[7]只采用每一像素的 2 bit 最低有效比特(LSB)嵌入水印，因此没有足够多的比特位嵌入更多的恢复水印，从而无法保证篡改恢复的质量。而文献[8]采用 3 bit LSB 扩大嵌入水印的容量，从而取得较好的篡改恢复质量，但是同时降低了含水印的图像质量。为了平衡水印容量与水印不可见性，嵌入水印时以视觉掩蔽效应<sup>[11,12]</sup>为指导，保证水印透明性的同时提高水印容量。文献[13]提出了结合空域对比敏感度、亮度掩蔽和对比掩蔽的联合恰可觉察失真(JND, just noticeable distortion)模型，使嵌入水印容量最大化，但该方法主要针对单图像的版权保护。基于立体视觉掩蔽的立体图像水印技术鲜有报道。

立体图像是由 2 个摄像机在不同角度拍摄同一场景的 2 幅图像组成，即左右视点，因此该两视点具有内在联系<sup>[14]</sup>，如立体掩蔽、像素匹配等。文献[15]采用立体图像的深度，实现了一种基于面向对象的水印方法，此方法能够抵抗 JPEG 压缩。文献[16]利用左右视点离散余弦变换系数的顽健关系嵌入数字水印，从而保护立体图像的版权。文献[17]提出了基于感知的顽健立体图像水印，然而该感知模型基于单图像，未考虑立体视觉掩蔽效应。以上立体图像水印方法目的都是为了保护立体图像的版权，目前尚未见到关于立体感知的立体图像认证水印方法。

本文针对立体图像完整性认证问题，提出一种基于立体视觉掩蔽的自恢复非对称立体图像水印方法。根据人类视觉系统，建立立体视觉掩蔽计算模型。依据此模型，立体图像的一视点比另一视点嵌入更多容量的水印，定义为非对称水印。该方法具体步骤为：首先依据立体视觉掩蔽计算模型，计算左右视点每一像素的 JND 值，将图像分块分为敏感和非敏感类型，不同类型的图像分块提供不同数

目比特位和分配结构嵌入水印；其次根据图像分块的不同纹理，采用离散小波变换(DWT, discrete wavelet transformation)系数计算不同类型的恢复水印，并且每个分块的恢复水印分别嵌入 2 个或 3 个不同的映射块，从而提高篡改恢复质量。仿真实验结果证明，该方法有效抵抗 collage attack，图像受到不同比例的篡改时，都具有较好的恢复质量。

## 2 立体视觉掩蔽计算模型

文献[18]根据非对称立体图像失真的主观实验，提出了左右视点质量影响立体图像视觉感知的视觉特性，如图 1 所示，图中横坐标表示左视点质量的 PSNR (peak signal-to-noise rate, 峰值信噪比) 值，纵坐标表示左右视点 PSNR 差值的主观可察觉阈值。当左视点的图像质量 32 dB 以上时，若右视点的 PSNR 低于左视点可察觉阈值以内，即 2 dB，则相对于左右视点质量相同的立体图像，在主观感知质量上没有差异。结合文献[18]所提出的立体视觉特性和单图像 JND 模型<sup>[19]</sup>，本文建立如图 2 所示的立体视觉掩蔽计算模型，分别计算左右视点 JND 值。具体过程如图 2 所示。

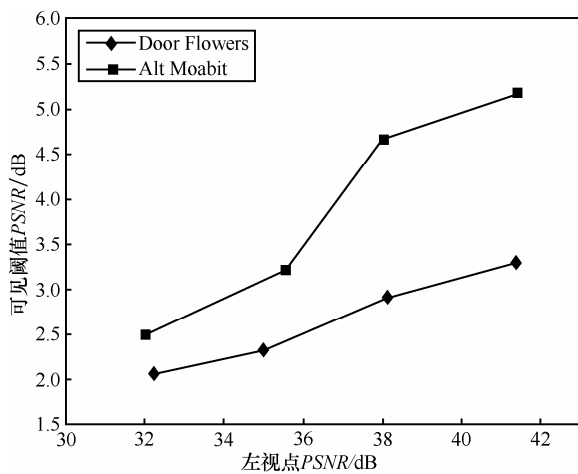


图 1 可见阈值 PSNR 与左视点 PSNR 关系

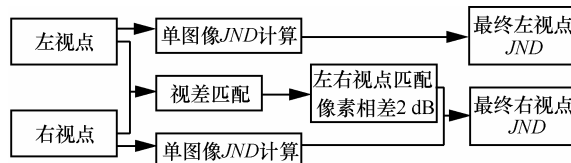


图 2 立体视觉掩蔽计算模型

1) 设原左右视点像素值分别记为  $I^L(i,j)$  和  $I^R(i,j)$ ，含水印的左右视点像素值分别记为  $I^L(i,j)$  和  $I^R(i,j)$ ，其中， $(i,j)$  表示为像素在视点中的位置

( $1 \leq i \leq N_1$  和  $1 \leq j \leq N_2$ ),  $N_1$  和  $N_2$  分别表示为图像的宽度和高度。含水印的左右视点的  $PSNR$  分别记为  $PSNR^L$  和  $PSNR^R$ 。左右像素每个像素的  $JND$  值分别记为  $JND^L(i,j)$  和  $JND^R(i,j)$ , 并计算为

$$JND^L(i,j) = I^L(i,j) - I^L(i,j) \quad (1a)$$

$$JND^R(i,j) = I^R(i,j) - I^R(i,j) \quad (1b)$$

设左视点为参考视点, 含水印的左右视点  $PSNR$  差值的上限为 2 dB, 建立公式, 并简化如下。

$$PSNR^L - PSNR^R \leq 2 \quad (2a)$$

$$10 \lg \left( \frac{255 \times 255 \times N_1 \times N_2}{\sum_i \sum_j [I^L(i,j) - I^L(i,j)]} \right) - 10 \lg \left( \frac{255 \times 255 \times N_1 \times N_2}{\sum_i \sum_j [I^R(i,j) - I^R(i,j)]} \right) \leq 2 \quad (2b)$$

$$10 \lg \left( \frac{\sum_i \sum_j [I^L(i,j) - I^L(i,j)]}{\sum_i \sum_j [I^R(i,j) - I^R(i,j)]} \right) \leq 2 \quad (2c)$$

由式(2c)计算得到

$$\frac{\sum_i \sum_j [I^R(i,j) - I^R(i,j)]}{\sum_i \sum_j [I^L(i,j) - I^L(i,j)]} \leq 1.58 \quad (3a)$$

$$\frac{\sum_i \sum_j JND^R(i,j)}{\sum_i \sum_j JND^L(i,j)} \leq 1.58 \quad (3b)$$

从上式可知当右视点像素的平均  $JND$  小于左视点像素的平均  $JND$  的 1.58 倍时, 其主观感知质量与两视点像素的平均  $JND$  相同的主观质量没有差异。

2) 人眼的立体视觉掩蔽效应意味一视点图像像素能够掩蔽其另一个视点图像的匹配像素的质量在一定程度上下降。因此根据式(3b), 定义左右匹配像素  $JND$  之间不大于 1.58 倍。设以左视点为参考视点, 在  $[0, \alpha]$  内搜索右视点的匹配像素, 计算立体图像像素视差图<sup>[20]</sup>, 表示为  $D$ , 其中设置不能找

到对应匹配像素的对应视差值为  $\alpha+1$ 。根据单图像  $JND$  模型, 计算左右视点每个像素的  $JND$ , 分别记为  $JND_f^L$  和  $JND_f^R(i,j)$ 。设立体图像由平行水平相机拍摄, 因此视差只为水平值, 在左视点每一像素  $JND$  不变的情况下, 更新右视点每一像素  $JND$  为

$$\hat{JND}_f^R(i-D(i,j),j) = \begin{cases} JND_f^R(i-D(i,j),j), & D(i,j) > \alpha \\ JND_f^L(i,j) \times 1.58, & D(i,j) \leq \alpha \end{cases} \quad (4)$$

3) 将两视点分别划分成不重叠的  $4 \times 4$  大小的图像分块, 计算每分块的  $JND$  均值, 定义为块  $JND$ 。左右视点图像分块的均值分别记为  $JND_b^L(x,y)$  和  $JND_b^R(x,y)$ , 其中,  $(x,y)$  表示图像分块在视点中的位置 ( $1 \leq x \leq N_1/4, 1 \leq y \leq N_2/4$ )。

### 3 基于立体视觉掩蔽的自恢复非对称水印方法

依据立体视觉掩蔽计算模型, 提出了自恢复非对称立体图像水印。其中, 非对称水印定义为立体图像的一视点比另一视点嵌入更大容量的水印, 从而提高篡改恢复质量。方法将分为 3 部分详细描述: 水印嵌入、篡改检测以及篡改恢复。

#### 3.1 水印嵌入

水印嵌入的主要过程如图 3 所示, 将左右视点分别划分成不重叠  $4 \times 4$  大小的图像分块, 并根据立体视觉掩蔽计算模型计算左右视点每分块的块  $JND$ 。定义当块  $JND$  大于等于 7 时, 为非敏感类型, 其像素提供 3 bit LSB 嵌入水印; 否则为敏感类型, 其像素提供 2 bit LSB。每一图像分块计算的水印分为认证和恢复水印。

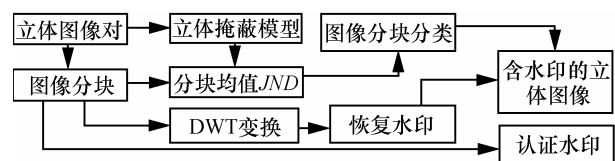


图3 水印嵌入过程

#### 3.1.1 分块认证水印计算

将图像分块像素的位置从左至右从上至下定义为  $1 \sim 16$ 。将每一像素值表示为 8 bit, 记为  $b_m(x,y) = \{b_{m,j}(x,y) | j=1,2,\dots,8\}$ , 其中,  $1 \leq m \leq 16$ ,  $j$  值越大表示有效位越高。每一图像分块计算 8 bit 的认证水印, 每比特记为  $A_s(x,y)$ , 其中,  $s$  表示第  $s$  个认证比特,  $1 \leq s \leq 8$ 。由于敏感和非敏感类型分块像素分

别提供 2 bit 和 3 bit LSB 嵌入水印, 因此采用 2 个像素的前高 5 bit 计算对应的认证水印比特, 即  $A_s(x, y)$  可表示为

$$A_s(x, y) = \text{mod} \left( \sum_{j=3}^8 (b_{m,j}(x, y) \& b_{n,j}(x, y)), 2 \right) \quad (5)$$

其中,  $1 \leq n \leq 16$ , “&”表示二进制与操作, mod 为取模运算。图 4 表示计算每位认证比特的 2 个像素位置, “1”表示计算第一个认证比特的像素位置, 依次类推。例如当计算  $A_1(x, y)$ , 则  $m=1$  和  $n=11$ , 设  $b_1(x, y) = \{10101001\}$ ,  $b_{11}(x, y) = \{01110101\}$ , 则  $A_1(x, y) = 1$ 。  $A_s^L(x, y)$  和  $A_s^R(x, y)$  分别表示左右视点图像分块计算的认证水印比特。

1	5	8	3
7	2	4	6
8	3	1	5
4	6	7	2

图 4 认证水印比特计算的像素位置

### 3.1.2 分块恢复水印计算

对每个图像分块进行一层 DWT 变换, 对应的 LL 系数表示为  $D_k(x, y)$ , 其中,  $k$  表示系数值在子带的位置,  $k \in \{1, 2, 3, 4\}$ 。由于 LL 近似图像分块信息, 因此提取图像分块的 LL 量化值作为恢复水印。对 LL 的 4 个系数分别量化如下。

$$X_k(x, y) = \text{Round}(D_k(x, y) / Q) \quad (6)$$

其中,  $\text{Round}\{\cdot\}$  表示四舍五入函数, 采用 5 bit 保存 LL 量化值,  $Q$  为量化步长, 取值为 17。由于每个图像分块的纹理信息不同, 因此每个分块的 LL 量化系数之间差值不同。根据其差值, 采用不同比特

数计算分块的恢复水印, 具体分为 4 种类型。

$$\forall k [X_1(x, y) - X_k(x, y)] \leq \alpha, k \in \{2, 3, 4\} \quad (7)$$

当  $\alpha$  分别取值为 1、3、7, 并满足式(7), 恢复水印分别为第 1 至第 3 种类型, 否则为第 4 种类型。当  $\alpha=1$  时, 5 bit 表示 LL 的第一个量化系数, 分别用 2 bit 表示其他 3 个量化系数与第一个量化系数的正负关系和绝对差值, 其中, 第 1 比特表示正负, “0”表示正, “1”表示负; 另一比特表示绝对差值。因此共 11 bit 表示第一类恢复水印。图 5 为第 1 类恢复水印的计算过程: 将图像分块一层 DWT 变换的低频系数进行量化, “10000”表示第一个量化系数; 第二个量化系数小于第一个量化系数, 其绝对差值为 1, 因此“11”表示第二个系数, 同理第 3 个和第 4 个系数分别表示为“00”和“11”, 该分块的恢复水印为“10000110011”。当  $\alpha$  分别取值为 3 和 7 时, 分别由 14 bit 和 17 bit 表示第 2 类和第 3 类恢复水印。最后一类恢复水印采用 5 bit 记录每一个量化系数值, 因此 20 bit 表示对应恢复水印。

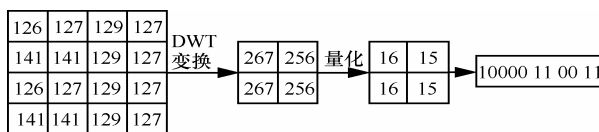


图 5 第一类恢复水印计算举例

### 3.1.3 水印比特分配

为了统计图像中不同恢复水印类型比例, 对不同纹理的立体图像计算, 采用“Alt Moabit”、“Art”、“Laundry”、“Bowling”、“DoorFlowers”、“Puppy”、“Rena”和“Akko”序列中的第一帧, 统计结果如表 1 所列。其中, 每一类的 2 个数据分别表示该类在左右视点的比例。对于不同纹理

表 1 立体图像的不同恢复水印类型比率/%

图像	第 4 类(20 bit)		第 3 类(17 bit)		第 2 类(14 bit)		第 1 类(11 bit)	
	左	右	左	右	左	右	左	右
Alt Moabit	3.29	2.77	10.93	11.06	18.08	18.30	67.70	67.80
Art	1.91	1.76	8.60	8.31	16.76	16.27	72.74	73.67
Laundry	3.28	3.49	9.15	9.09	19.05	18.91	68.52	68.51
Bowling	0.45	0.6	2.1	2.34	7.16	7.34	90.29	89.71
DoorFlowers	1.74	1.7	6.41	6.42	8.64	8.88	83.22	83.00
Puppy	1.67	1.94	7.65	7.81	15.37	15.05	75.32	75.20
Rena	0.11	0.12	0.86	0.88	7.18	8.80	91.85	88.33
Akko	0.45	0.64	4.29	5.70	14.46	15.88	80.80	77.79

立体图像，第 4 类恢复水印的比率最低，第 1 类恢复水印的比率最高。因此设计恢复水印位分配时，包含第 1 类恢复水印的方案最多，而包含第 4 类恢复水印的方案最少。

非敏感类型图像分块的每一像素提供 3 bit LSB，因此该类型分块共提供 48 bit 嵌入水印，将 48 bit 分别标号为 1~48，如图 6 所示。48 bit 由 5 部分或者 4 部分组成。第 1 部分表示分块本身类型，记  $C_t$ ，第 1 比特表示，其中，“0”表示非敏感，“1”表示敏感，因此对于非敏感类型分块  $C_t=“0”$ 。第 2 和第 3 部分定义嵌入恢复水印的类型，分别记  $C_n$  和  $C_r$ 。 $C_n$  表示恢复水印比特的分配类型代码，共 8 类，用 3 bit 表示，即第 2 至第 4 比特。表 2 显示每一类型代码对应具体嵌入的恢复水印类型，其中，2 个数字表示嵌入 2 个恢复水印，3 个数字表示嵌入 3 个恢复水印，每一数字值表示嵌入恢复水印的比特数。例如， $C_n=“000”$  对应 “20、17” 表示嵌入 2 个恢复水印，分别为第 4 和第 3 类型的恢复水印，所对应的比特数分别为 20 和 17。此时，图 6 的  $Z_2$  和  $Z_3$  值分别为 24 和 41，表示第 5 至第 24 比特嵌入 20 bit 的恢复水印，而第 25 至第 41 比特嵌入 17 bit 的恢复水印。同理，当  $C_n=“010”$  时，图 6 的  $Z_1$ 、 $Z_2$  和  $Z_3$  的值分别为 21、32 和 43，表示第 5 至第 21 比特嵌入 17 bit 的恢复水印，第 22 至第 32 比特嵌入 11 bit 的恢复水印，而第 33 至第 43 比特嵌入 11 bit 的恢复水印。



图 6 非敏感类型图像分块的水印比特分配

表 2 非敏感图像分块恢复水印比特的分配类型

$C_n$	详细恢复水印比特分配	$C_n$	详细恢复水印比特分配
000	20、17	100	11、11、17
001	17、20	101	14、11、11
010	17、11、11	110	11、14、11
011	11、17、11	111	11、11、14

此外，当嵌入 2 个恢复水印时，这 2 个分别为左右视点分块的恢复水印。当嵌入 3 个恢复水印时，前 2 个分别为左右视点分块的恢复水印，而第 3 个恢复水印所在视点由第 4 部分定义，记  $C_v$ ，其中，“0”表示左视点，“1”表示右视点，此时图 6 中  $Z_4=Z_3+1$ 。但是，表 2 的前 2 个类型，由于没有第 3

个恢复水印，因此没有第 4 部分，此时  $Z_4$  也不存在。最后一部分为分块认证水印，记  $C_d$ ，其嵌入的位置为第  $Z_5$  至第 48 比特。当存在第 4 部分时， $Z_5=Z_4+1$ ，而不存在时，则  $Z_5=Z_3+1$ 。由于嵌入的认证水印的剩余位置个数不同，因此每个分块的 8 个认证比特并不全部都嵌入。因此，每一分块的水印比特，记  $C$  为

$$C = C_t \parallel C_n \parallel C_r \parallel C_v \parallel C_d \quad (8a)$$

$$\text{或者 } C = C_t \parallel C_n \parallel C_r \parallel C_d \quad (8b)$$

其中，“ $\parallel$ ”表示比特连接符号。

敏感图像分块提供 32 bit 嵌入水印，比特分配的结构稍不同于非敏感分块，由于在该类型图像分块只嵌入 2 个恢复水印，因此没有  $C_v$ ，只有 4 部分组成，类似式(8b)。只是  $C_t$  的值为 “1”， $C_n$  和  $C_r$  如表 3 所列， $C_d$  为剩余比特。

表 3 敏感图像分块恢复水印位的分配类型

$C_n$	详细恢复水印比特分配	$C_n$	详细恢复水印比特分配
00	11、11	01	14、11
10	11、14	11	11、17

### 3.1.4 水印嵌入

设左右视点图像分块分别记为  $B^L(x,y)$  和  $B^R(x,y)$ 。将每个分块的恢复水印分别嵌入 2 个不同视点的映射块，并使部分图像分块的恢复水印嵌入第 3 个不同的映射块。为了有效防止 collage attack，各个图像分块的前 2 bit 认证水印比特分别嵌入前 2 个映射块。水印嵌入共分为 5 个步骤，其中，后 4 个步骤分别将各类型恢复水印配对，并和认证水印嵌入相应映射块，如图 7 所示。

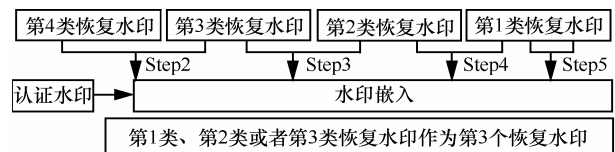


图 7 恢复水印嵌入步骤

**Step1** 首先将左右视点分成不重叠的  $4 \times 4$  大小的图像分块，根据立体视觉掩蔽计算模型，计算每一图像分块的  $JND$ ，将图像分块分为敏感和非敏感类型。敏感和非敏感类型图像分块分别提供 2 bit 和 3 bit LSB 嵌入水印。对每一图像分块进行一层 DWT 变换，计算各个分块的不同类型的恢复水印。

**Step2** 随机将左视点第 4 类恢复水印和部分右视点第 3 类恢复水印的图像分块一一配对，简称

为 $[B_{20}^L, B_{17}^R]$ 。左右视点图像分块分别记为 $B^L(x,y)$ 和 $B^R(x',y')$ , 恢复水印分别记为 $C_{r_1}^L(x,y)$ 和 $C_{r_1}^R(x',y')$ ,  $(x,y)$ 和 $(x',y')$ 分别表示分块在视点的位置。根据表 2, 随机将左视点非敏感类型图像分块作为它们的第一个映射块, 记为 $B^L(x'',y'')$ ,  $(x'',y'')$ 表示分块在视点的位置。将相应水印嵌入 $B^L(x'',y'')$ 的过程为 $C_i(x'',y'') = "0"$ 、 $C_{\pi}(x'',y'') = "000"$ 和 $C_r(x'',y'') = C_{r_1}^L(x,y) \parallel C_{r_1}^R(x',y')$ 。  $C_d(x'',y'')$ 为剩余的 7 bit 嵌入认证水印。其中, 前 2 位分别为 $B^L(x,y)$ 的第一认证比特和 $B^R(x',y')$ 的第二认证比特, 即 $A^L_1(x,y)$ 和 $A^R_2(x',y')$ 。剩余的 5 bit 为 $B^L(x'',y'')$ 的第 3 至第 7 认证比特, 即 $A^L_3(x'',y''), \dots, A^L_7(x'',y'')$ , 因此 $C_d(x'',y'') = A^L_1(x,y) \parallel A^L_2(x,y) \parallel A^L_3(x'',y'') \parallel \dots \parallel A^L_7(x'',y'')$ 。根据式(8b)计算 $C(x'',y'')$ , 并嵌入 $B^L(x'',y'')$ 每一像素的 3 bit LSB。用相同的方法随机将右视点非敏感图像分块作为它们的第 2 个映射块, 嵌入相应的水印, 不同的是 $B^L(x,y)$ 的第 2 认证比特和 $B^R(x',y')$ 的第 1 认证比特嵌入第 2 个映射块。重复该过程, 将每对 $[B_{20}^L, B_{17}^R]$ 的水印信息嵌入 2 个映射块。

同理随机组成 $[B_{17}^L, B_{20}^R]$ , 将它们的恢复水印和相应的认证水印随机嵌入 2 个视点的非敏感映射块, 此时映射块相应的 $C_{\pi} = "001"$ 。Step2 将所有左右视点第 4 类恢复水印和部分第 3 类恢复水印嵌入 2 个不同的映射块。

**Step3** 随机组成 $[B_{17}^L, B_{11}^R]$ 或 $[B_{11}^L, B_{17}^R]$ , 随机将相应水印嵌入左右视点的非敏感类型的映射块, 嵌入过程类似 Step1。此外将左右视点第一类恢复水印不重复并且交替地作为第 3 个恢复水印嵌入映射块中, 并且设置 $C_v$ 为“0”或“1”, 从而记录第 3 个恢复水印所在视点。当一视点所有非敏感类型分块已经完成水印嵌入, 则随机选取该视点的敏感类型分块作为映射块。则此时,  $C_i = "1"$ ,  $C_{\pi} = "11"$ 。Step3 将所有左右视点剩余的第 3 类恢复水印和部分第 2 类恢复水印嵌入 2 个不同的映射块。

**Step4** 随机组成 $[B_{14}^L, B_{11}^R]$ 或 $[B_{11}^L, B_{14}^R]$ , 随机将相应水印嵌入左右视点的非敏感或者敏感类型映射块, 方法与 Step3 相同, 只是对应 $C_i$ 、 $C_{\pi}$ 等值不同。该步骤将所有左右视点剩余的第 2 类恢复水印和部分第 1 类恢复水印嵌入 2 个不同的映射块。

**Step5** 随机组成 $[B_{11}^L, B_{11}^R]$ , 嵌入左右视点相应敏感或者非敏感类型映射块, 方法同 Step3。如果映射块为非敏感类型, 第 2 类或者第 3 类的恢复

水印作为嵌入映射块的第 3 个恢复水印, 类型选择的规则为先满足嵌入第 3 类恢复水印, 交替选择左右视点图像分块的恢复水印。该步骤将所有左右视点剩余的第 1 类恢复水印嵌入映射块。

左右视点各个分块嵌入水印后, 每个图像分块都有 2 个或 3 个映射块保存该分块的恢复水印, 记录每个图像分块的映射块位置, 记 $LT$ 。其具体形式, 每个图像分块都对应 3 个值, 每个值分别代表映射块的位置, 即 $(x,y)$ 。当部分图像分块没有第 3 个映射块时, 则保存的数据为(0,0),  $LT$ 将用于图像篡改的检测与恢复。

### 3.2 篡改检测

将待认证的左右视点分成不重叠的 $4 \times 4$ 大小的图像分块, 提取认证水印, 检测每一个分块是否被篡改。设 3 个左视点分块是否被篡改标志矩阵分别记 $F^L_1$ 、 $F^L_2$ 和 $F^L_3$ , 3 个右视点分块是否被篡改标志矩阵分别记 $F^R_1$ 、 $F^R_2$ 和 $F^R_3$ 。篡改检测详细步骤如下。

**Step1** 提取图像分块嵌入的认证水印比特, 该过程为认证水印嵌入的逆过程: 根据“分块类型”和“恢复水印比特的分配类型代码”, 提取左右视点每个分块的不同数目的认证比特, 分别记为 $A^L_s(x,y)$ 和 $A^R_s(x,y)$ 。根据式(5)重新计算各个分块的认证比特, 左右视点的认证比特分别记为 $A^L_s(x,y)$ 和 $A^R_s(x,y)$ 。

**Step2** 比较图像分块提取与重新计算的第一个认证比特是否一致。例如 $B^L(x',y')$ 第一个映射块为 $B^L(x,y)$ , 从 $B^L(x,y)$ 提取 $B^L(x',y')$ 的第一个认证比特为 $A^L_1(x',y')$ 。比较 $A^L_1(x',y')$ 和 $A^L_s(x,y)$ 是否相等, 如果不相等, 认定 $B^L(x,y)$ 和 $B^L(x',y')$ 被篡改, 分别设置 $F^L_1(x,y)$ 和 $F^L_1(x',y')$ 为 1, 否则, 都为 0。对于右视点图像分块, 从该图像分块的第 2 个映射块提取第一个认证比特与计算的认证比特比较, 从而设置 $F^R_1$ 相应的值。重复该步骤直至完成所有分块第一个认证比特的比较, 设置 $F^L_1$ 和 $F^R_1$ 相应值。

**Step3** 当图像分块是敏感类型, 并且“恢复水印位的分配类型代码”为“11”, 由于该图像分块只嵌入一个认证比特, 因此该图像分块的检测跳过 Step3 和 Step4。否则, 比较第 2 认证水印比特是否与重新计算的第 2 个认证比特一致。例如 $B^R(x,y)$ 是 $B^L(x',y')$ 的第 2 个映射块, 从 $B^R(x,y)$ 提取 $B^L(x',y')$ 的第 2 个认证比特为 $A^L_2(x',y')$ 。比较 $A^L_2(x',y')$ 和

$A^L_2(x',y')$ 是否相等,如果不相等,认定  $B^L(x',y')$ 和  $B^R(x,y)$ 同时被篡改,分别设置  $F^L_2(x',y')$ 和  $F^R_2(x,y)$ 为 1; 否则为 0。对于右视点图像分块,从该图像分块的第 1 个映射块提取第 2 个认证比特与计算的第 2 个认证比特比较,从而设置  $F^L_2$ 和  $F^R_2$ 相应的值。重复该步骤直至完成所有分块第 2 个认证比特的比较,设置  $F^L_2$ 和  $F^R_2$ 相应值。分别对  $F^L_1$ 、 $F^R_1$ 、 $F^L_2$ 和  $F^R_2$ 进行形态学的膨胀和腐蚀操作,从而更新  $F^L_1$ 、 $F^R_1$ 、 $F^L_2$ 和  $F^R_2$ 。

**Step4** 比较剩余的认证比特。从图像分块本身提取  $A^L_s(x,y)$ 或  $A^R_s(x,y)$ ,分别与  $A^L_s(x,y)$ 或者  $A^R_s(x,y)$ 进行比较,此时  $3 \leq s \leq 8$ 。当任一数值不相等,则认定相应的分块被篡改,设置  $F^L_3$ 或者  $F^R_3$ 相应的值为 1, 否则为 0。

**Step5** 对左右视点的标志矩阵拟合,拟合的结果分别表示为  $F^L_f$ 和  $F^R_f$ ,  $F^L_f = (F^L_1 \cap F^L_2) \cup F^L_3$ 和  $F^R_f = (F^R_1 \cap F^R_2) \cup F^R_3$ 。其中,  $\cap$ 和  $\cup$ 分别表示二进制与和或运算。当  $F^L_f$ 或者  $F^R_f$ 的系数值为 1 时表示相应的分块遭到篡改, 否则表示未被修改。

### 3.3 篡改恢复

根据  $F^L_f$ 和  $F^R_f$ , 若一图像分块认定篡改, 其恢复过程具体步骤如下。

**Step1** 根据查询表  $LT$ , 获取第 1 个同一视点的映射块。如果该映射块没有被篡改, 提取水印。根据该映射块的类型和恢复水印比特的分配类型代码, 提取嵌入映射块的恢复水印。如篡改的图像块位于左视点, 则提取第一个恢复水印, 否则为第 2 个恢复水印。将提取的恢复水印比特, 计算 4 个 LL 量化系数, 记  $X'_k(x,y)$ , 并计算 LL 系数如下。

$$\hat{D}_{k,l}(x,y) = X'_k(x,y) \cdot Q \quad (9)$$

令 LH、HL 和 HH 的系数为 0, 进行 DWT 逆变化恢复图像分块, 该分块恢复结束。如果映射块被篡改, 执行 Step2。

**Step2** 获取另一视点的第 2 个映射块。如果该映射块没有被篡改, 提取水印并恢复篡改的分块, 方法与 Step1 相同。否则, 执行 Step3。

**Step3** 根据  $LT$ , 提取第 3 个映射块的位置, 分别获取左右视点相应位置分块, 当提取的“图像分块类型”为“0”, “恢复水印位的分配类型代码”是表 2 的后 6 种类型, 并且“第 3 个恢复水印所在视点”符合篡改分块所在视点, 则该分块被认定为篡改分块的映射块。提取映射块的第 3 个恢复水印, 计算过程与 Step1 相同。如果没有第 3 个映射块或者映射块已经被篡改, 则停止该分块的恢复。

**Step4** 对所有篡改的分块进行 Step1~Step3 的操作, 部分篡改的图像分块仍不能恢复, 则采用像素插补法恢复, 即  $3 \times 3$  区域内的 8 个未被篡改或已恢复的领域像素均值恢复。

## 4 实验结果与讨论

采用  $640 \times 480$  的“Bowling”、“Door Flowers”、“Akko”、“Art”和“Laundry”序列的第一帧的灰度图像进行相关实验, 如图 8 所示为每幅立体图像的右视点, 视点每个像素点的灰度等级为 256。

### 4.1 水印容量

文献[7]和文献[9]由于每一像素提供 2 bit 嵌入水印, 水印容量为 2 bit/pixel。而文献[8]每一像素提供 3 bit, 水印容量为 3 bit/pixel。而本方法将图像分块分为敏感和非敏感类型, 其像素分别提供 2 bit 和 3 bit 嵌入水印, 因此本方法的水印容量的变化范围为 2 bit/pixel 与 3 bit/pixel, 敏感和非敏感类型图像分块所占比例决定了水印容量。根据 JND 模型, 像素处于纹理较丰富区域, 对应的 JND 值较高。因此对于纹理较复杂的图像, 非敏感类型的比率较高, 水印容量较高; 反之, 水印容量较低。表 4 列出了立体图像左右视点的水印容量, 其中, 第 2 行数据显示单图像 JND 模型指导下, 右视点的水印容



图 8 立体图像的原始右视点

量，而第 3 行为根据立体视觉掩蔽计算模型，计算的右视点水印容量。在立体视觉掩蔽指导下，相比单图像 JND 模型，右视点的水印容量提高较多，提高的范围为 0.21 bit/pixel 与 0.36 bit/pixel。此外，相对较平坦的“Bowling”，其水印容量较低，而纹理较复杂的“Door Flowers”则水印容量最高。

表 4 不同立体图像的水印容量 (单位: bit/pixel)

左/右	Bowling	Door Flowers	Akko	Art	Laundry
左	2.05	2.63	2.18	2.48	2.23
右(单)	2.04	2.63	2.17	2.48	2.24
右(立)	2.25	2.88	2.46	2.76	2.60

与水印容量相对应的是含水印立体图像的 PSNR，其水印容量越大，则 PSNR 越低，如表 5 所列。含水印的“Bowling”PSNR 最高，而含水印的“Door Flowers”最低。左右视点 PSNR 的差值约为 2dB，符合文献[18]的主观实验结论，水印主观不可感知。

表 5 含水印立体图像的 PSNR (单位: dB)

左/右	Bowling	Door Flowers	Akko	Art	Laundry
左	43.09	39.07	42.19	40.45	42.04
右	41.11	38.59	39.73	38.88	39.62

#### 4.2 一般粘贴实验

为了验证本方法在篡改检测以及恢复的有效性，通过一系列相关实验与现有的单图像水印方法进行比较和讨论。将文献[7]、文献[8]和文献[9]的水印方

法应用于立体图像，分别表示为 Lin’s、Lee’s 和 Li’s，该 3 种方法将左右视点作为相互独立的视点。

第 1 个实验在含水印的“Laundry”的背景纹理区域粘贴一个“卡通狗头”，如图 9(a)和图 9(b)所示，左右视点的篡改约为 6.47%。本方法的篡改检测率达到 99%以上，恢复的图像近似于原始图像，视觉不可感知，如图 9(c)和图 9(d)所示。如表 6 所示，Lin’s、Lee’s 的篡改检测率也都 99%以上，然而，本方法左右视点恢复相对于含水印图像的 PSNR 分别为 37.40 dB 和 37.41 dB，比 Lin’s 和 Lee’s 约高 1 dB。Li’s 的篡改检测率最低，其篡改绘制质量也较差。本实验表明对于纹理复杂区域的篡改，本方法利用 DWT 系数的恢复水印方法优于 Lin’s 和 Lee’s 的像素均值方法。

第 2 个实验为含水印的“Laundry”有多处修改，篡改率约为 11.15%，具体为“一块肥皂”、“一只小包”和“一个洗衣篮”粘贴于 3 个不同纹理的区域，平坦区域较多，如图 10(a)和图 10(b)所示。Lin’s、Lee’s 和本方法的篡改检测率和恢复 PSNR 都高于 Li’s 的方法，如表 6 所列。此外，本方法恢复的图像如图 10(c)和图 10(d)所示，主观不易察觉。与 Lin’s 和 Lee’s 方法比较，本方法恢复的 PSNR 较高，但是相比第一个实验，PSNR 增加的幅度有所减少，因此实验表明本方法在纹理区域篡改的恢复能力较强。

第 3 个实验扩大篡改比例，含水印的“Akko”右边部分大面积替换为其他图像，如图 11(a)和图 11(b)所示，其篡改率达到约 51.31%。Li’s 的篡改检测率最低，其他 3 种方法的篡改检测率都达到

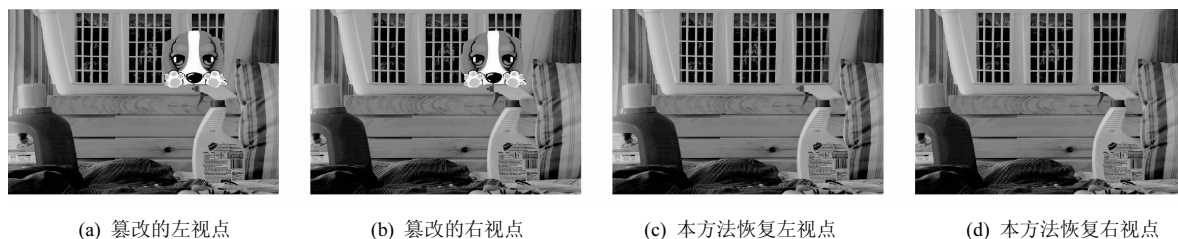


图 9 “Laundry”一处篡改以及恢复

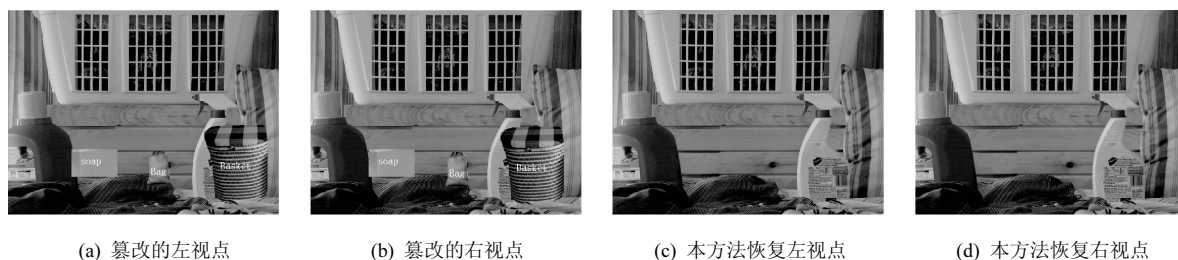


图 10 “Laundry”多处篡改以及恢复

99%以上，如表 6 所列。由于本方法部分图像分块的恢复水印保存于 3 个不同映块因此恢复的 *PSNR* 远远高于另外 3 种方法，达 5 dB 多。在视觉上，本方法恢复的左右视点不可感知如图 11(c)和图 11(d)所示。通过右视点的局部恢复比较，本方法恢复的纹理较清晰，Lin’s 和 Lee’s 方法具有明显的块效应，而 Li’s 的方法由于较低的篡改检测率，其部分内容无法被恢复，如图 11(e)~图 11(i)所示。

以上 3 个实验证明本方法在图像一处或多处篡改的情况下，都能进行有效检测和恢复。相对平坦区域的篡改，纹理区域的恢复优势较明显。此外，较大比例篡改的恢复优于较小篡改比例的恢复。

### 4.3 collage attack 实验

collage attack 是对自恢复水印最威胁的攻击之一，很多方法无法抵抗该类攻击。将如图 12(a)和图 12(c)所示的含水印“Laundry”标注的“衣服”

复制，并粘贴于含水印的“Bowling”中，如图 12(b)和图 12(d)所示。由于 Lin’s 和 Lee’s 方法的分块篡改检测相互独立，只能检测部分边缘篡改，而无法识别大部分篡改内容，如图 12(e)~图 12(h)所示。Li’s 的方法的篡改检测率都达到 90%以上，但是仍旧低于本方法，如表 7 所列。图 12(i)和图 12(j)显示本方法篡改恢复的左右视点不可感知，其 *PSNR* 远远大于 Lin’s 和 Lee’s，比 Li’s 的高 4 dB。

此外，复制含水印的“DoorFlowers”部分区域，如图 13(a)矩形标注，覆盖同一图像的“雕塑”，如图 13(b)所示，左视点被类似篡改。本方法能够准确检测篡改，并能有效恢复，如图 13(c)和图 13(d)，篡改检测率和恢复 *PSNR* 都高于 Li’s 的方法，如表 8 所列。其他 2 种方法仍旧不能检测该类篡改，Lee’s 的方法的篡改检测率甚至达到 0，因此它们都无法恢复篡改。实验证明本方法对 collage attack 检测的有效性，并能实现篡改区域的近似恢复。

表 6 不同方法对不同篡改的检测以及恢复比较

图像	检测及恢复	本方法		Lin’s		Lee’s		Li’s	
		左	右	左	右	左	右	左	右
Laundry(一处)	篡改检测率/%	99.12	99.23	99.91	99.92	99.11	99.03	94.02	94.01
	恢复 <i>PSNR</i> /dB	37.40	37.41	36.39	36.41	36.47	36.42	34.96	34.95
Laundry(多处)	篡改检测率/%	99.85	99.12	99.95	99.91	99.32	99.78	91.21	91.34
	恢复 <i>PSNR</i> /dB	36.83	37.01	36.68	36.73	36.21	36.21	33.12	33.26
Akko	篡改检测率/%	100	99.77	99.58	100	99.98	99.76	61.11	59.84
	恢复 <i>PSNR</i> /dB	34.55	34.15	28.82	29.08	25.22	25.12	18.19	17.89

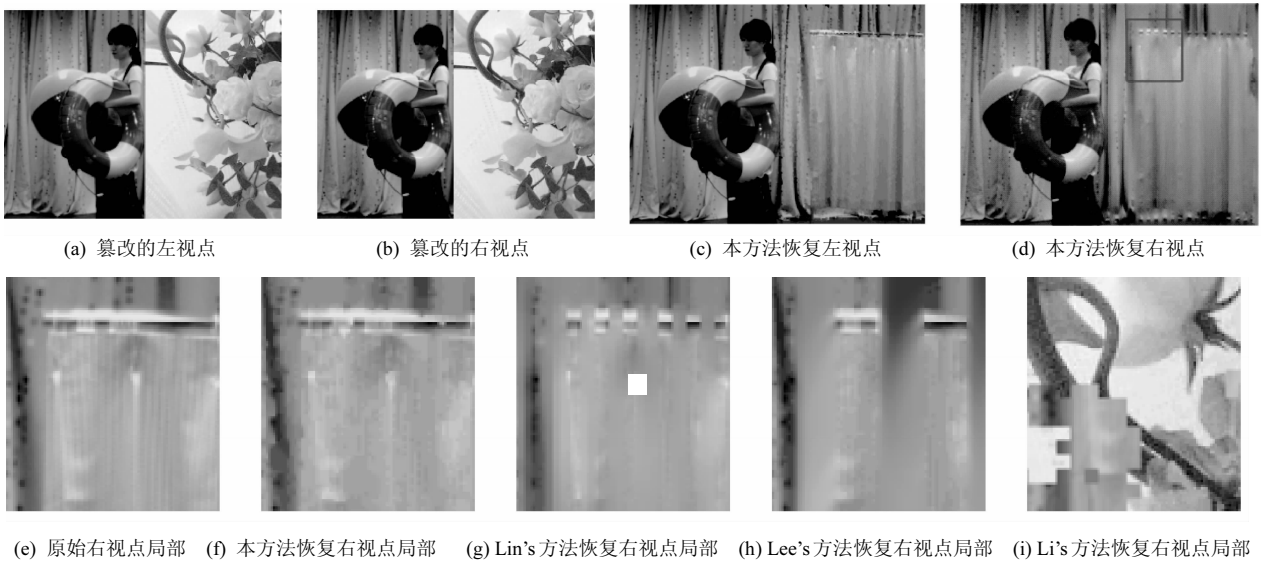


图 11 “Akko”大面积篡改以及恢复

表 7 不同方法对“Bowling”篡改检测以及恢复的比较

检测及恢复	本方法		Lin's		Lee's		Li's	
	左	右	左	右	左	右	左	右
篡改检测率/%	99.72	99.77	4.21	4.69	3.11	2.96	93.14	92.86
恢复/dB	45.57	45.11	17.89	18.09	17.62	17.44	41.50	41.03

表 8 不同方法对“Door Flowers”篡改检测以及恢复的比较

检测及恢复	本方法		Lin's		Lee's		Li's	
	左	右	左	右	左	右	左	右
篡改检测率/%	94.27	94.44	9.72	9.39	0	0	86.32	85.43
恢复/dB	45.43	45.24	34.77	34.46	32.58	32.20	44.92	44.78

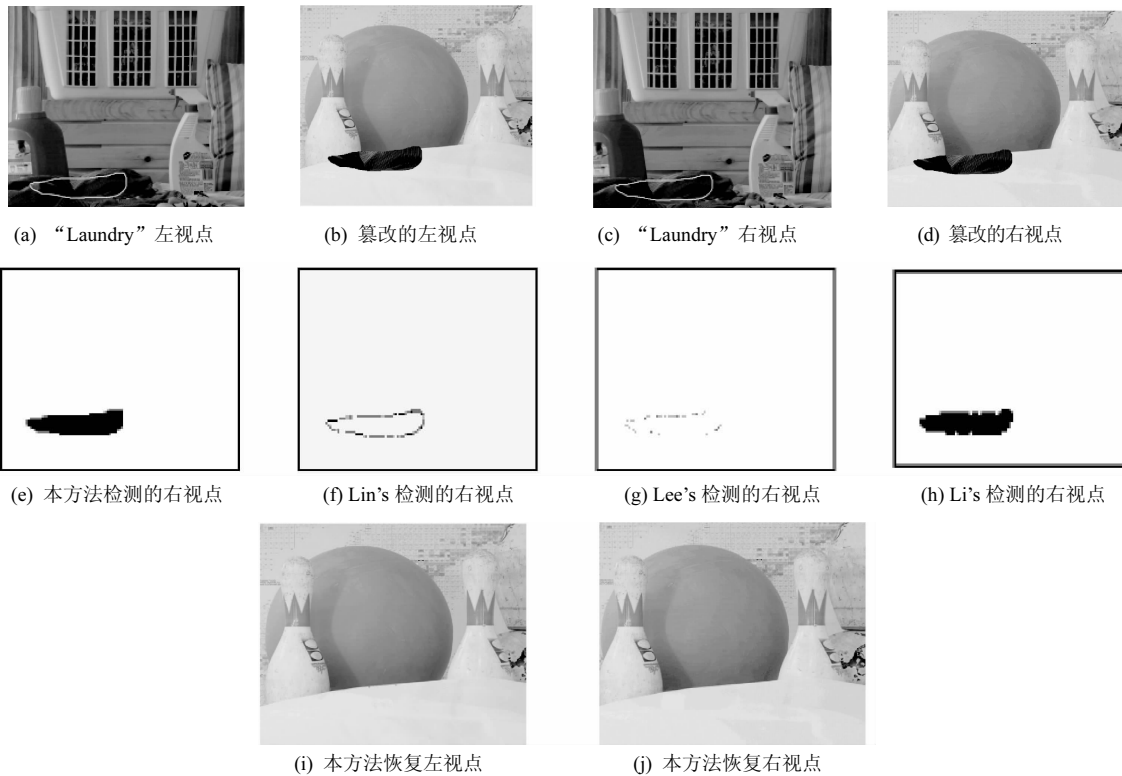


图 12 “Bowling”的篡改、检测以及恢复

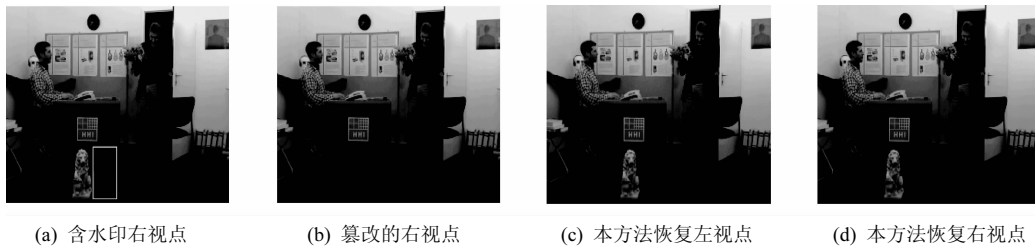


图 13 “Door Flowers”的篡改以及恢复

#### 4.4 不同面积剪切实验

为了再次证明本方法的有效性，将文献[7]、文献[8]和文献[9]的 3 种方法进一步扩展，利用左右视

点的视差关系改进篡改恢复。具体扩展方法为：在篡改恢复过程中，使用水印恢复篡改后，如仍旧有部分图像分块未恢复，再通过视差图  $D$  对左右视点

表9 Art不同比例剪切恢复的PSNR比较(单位: dB)

比例	本方法		Lin's		Lee's		Li's		Lin's+D		Lee's+D		Li's+D	
	左	右	左	右	左	右	左	右	左	右	左	右	左	右
10%	40.56	40.79	39.56	39.21	39.56	39.44	38.32	38.45	41.18	40.93	39.62	39.42	40.19	40.05
20%	37.11	37.29	35.82	36.10	36.18	36.23	32.04	32.42	37.09	37.17	36.23	36.23	36.22	35.59
30%	34.21	34.56	32.23	32.27	33.74	34.01	27.88	28.01	33.94	33.64	33.77	34.09	31.91	31.38
40%	32.57	32.74	29.69	29.68	31.94	31.93	24.59	24.21	31.25	30.82	32.13	32.07	30.63	29.12)
50%	30.99	31.04	27.46	27.53	28.94	28.64	22.53	22.42	28.73	28.39	29.92	29.43	26.77	26.01
60%	29.44	29.71	25.63	25.65	26.11	26.14	20.12	20.03	26.18	25.97	27.49	25.22	24.00	23.56
70%	27.51	27.80	24.04	24.09	23.59	23.52	18.08	18.23	24.21	24.07	25.18	24.91	20.84	20.06

分别进行恢复, 扩展的3种方法分别表示为Lin's+D、Lee's+D和Li's+D。

剪切是另一种常用篡改攻击, 将含水印“Art”的中间区域从10%~70%分别进行剪切。7种方法的都能对篡改进行有效检测, 因此都能对其篡改进行一定的恢复。Lin's+D、Lee's+D和Li's+D方法的恢复质量分别都不同程度好于原始的3种水印方法(Lin's、Lee's和Li's), 如表9所列。尤其当篡改区域等于10%时, Lin's+D方法左右恢复视点的PSNR分别为41.18 dB和40.93 dB, 比Lin's方法提高了1.5 dB多, 并且高于本方法。但是, 当篡改比例增大时, 本方法视点恢复的PSNR都高于Lin's+D。Li's+D方法相比Li's尽管较大幅度提高了恢复质量, 但是仍旧远低于本方法的恢复质量。当篡改比例50%及以上时, 本方法的恢复质量比Lin's+D和Lee's+D方法都高1 dB多, 比Li's方法高3 dB多。因此本实验证明本方法篡改恢复能力较优于现有的以及扩展的单图像水印方法。

## 5 结束语

提出的基于立体视觉掩蔽的自恢复非对称数字水印方法能够较准确识别并较高质量恢复篡改。该方法具有以下几个特点。1) 以立体视觉掩蔽计算模型为指导嵌入水印, 平衡了水印容量和水印不可见性。2) 根据分块纹理, 计算基于DWT系数的不同类型的分块恢复水印。3) 图像分块的认证水印比特相互联系, 因此有效抵抗collage attack。4) 分块的恢复水印分别嵌入2个或3个不同映射块, 有效提高了篡改恢复质量。实验证

明在不同比例篡改下, 本方法的恢复质量好于现有以及扩展的单图像水印方法。由于本方法需要一张查询表查询图像分块恢复水印保存位置, 因此需要额外的比特传输该查询表, 下一步工作重点将减少传输比特。

## 参考文献:

- [1] TANIMOTO M, TEHRANI M, FUJII T, *et al.* FTV for 3-D spatial communication[J]. Proceedings of the IEEE, 2012, 100(4):905-917.
- [2] SMOLIC A, KAUFF P, KNORR S, *et al.* Three-dimensional video post-production and processing[J]. Proceedings of the IEEE, 2011, 99(4):607-62
- [3] HUR N, LEE H, LEE G, *et al.* 3DTV broadcasting and distribution systems[J]. IEEE Transaction on Broadcasting, 2011, 57(2):395-407.
- [4] 杨义先, 钮心忻. 数字水印理论与技术[M]. 北京: 高等教育出版社, 2006.  
YANG Y X, NIU X X. Theory and Applications of Digital Watermarking[M]. Beijing: Higher Education Press, 2006.
- [5] QI X, XIN X. A quantization-based semi-fragile watermarking approach for image content authentication[J]. Journal of Visual Communication and Image Representation, 2011, 22(2):187-200.
- [6] 叶天语, 钮心忻, 杨义先. 多功能双水印算法[J]. 电子与信息学报, 2009, 31(3):546-551.  
YE T Y, NIU X X, YANG Y X. A multi-purpose dual watermark algorithm[J]. Journal of Electronics & Information Technology, 2009, 31(3):546-551.
- [7] LIN P, HSIEH C, HUANG P. A hierarchical digital watermarking method for image tamper and recovery[J]. Pattern Recognition, 2005, 38(25):2519-2529.
- [8] LEE T, LIN S. Dual watermark for image tamper detection and recovery[J]. Pattern Recognition, 2008, 41(11):3497-3506.
- [9] LI C, WANG Y, MA B, *et al.* A novel self-recovery fragile watermarking scheme based on dual-redundant-ring structure[J]. Computer & Electrical Engineering, 2011, 37(6):927-940.
- [10] FRIDRICH J, GOLJAN M, MEMON N. Cryptanalysis of the Yeung-Mintzer fragile watermarking technique[J]. Journal of Electronic Imaging, 2002, 11(2):262-274.

[11] 李动节, 朱仲杰, 王玉儿. 结合视觉感知的对象基嵌入式图像编码中的码率控制方法[J]. 通信学报, 2012, 33(4):115-120.  
LI D J, ZHU Z J, WANG Y E. HVS-based rate-control scheme for object-based embedded image coding[J]. Journal on Communications, 2012, 33(4):115-120.

[12] 张维纬, 张茹, 刘健毅等. 基于纹理特征的 H.264/AVC 顽健视频水印算法[J]. 通信学报, 2012, 33(3):82-89.  
ZHANG W W, ZHANG R, LIU J Y, *et al.* Robust video watermarking algorithm for H.264/AVC based on texture feature[J]. Journal on Communications, 2012, 33(4):82-89.

[13] NIU Y, KYAN M, KRISHNAN S, *et al.* A combined just noticeable distortion model-guided image watermarking[J]. Signal, Image and Video Processing, 2011, 5(4):517-526.

[14] ELLINAS J, SANGRIOTIS M. Stereo image compression using wavelet coefficients morphology[J]. Image and Vision Computing, 2004, 22(4):281-290.

[15] CAMPISI P. Object-oriented stereo-image digital watermarking[J]. Journal of Electronic Imaging, 2008.

[16] LUO T, YU M, JIANG G, *et al.* Novel DCT-based blind stereo image watermarking algorithm[A]. Future Wireless Networks and Information Systems[C]. Macau, China, 2012. 297-304.

[17] NIU Y, SOUIDENE W, BEGHDADI A. A visual sensitivity model based stereo image watermarking scheme[A]. Third European Workshop on Visual Information Processing[C]. Paris, France, 2011. 211- 215.

[18] WANG X, JIANG G, ZHOU J, *et al.* Visibility threshold of compressed stereoscopic image: effects of asymmetrical coding[J]. The Imaging Science Journal, 2013, 61(2):172-182.

[19] YANG X, LIN W, LU Z, *et al.* Motion-compensated residue preprocessing in video coding based on just-noticeable-distortion profile[J]. IEEE Transactions on Circuits System Video Technology, 2005, 15(6):742-750.

[20] BOYKOV Y, KOLMOGOROV V. An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2004, 26(9): 1124-1137.

作者简介:



骆挺 (1980-), 男, 浙江慈溪人, 宁波大学博士生、讲师, 主要研究方向为信息隐藏、多媒体信号处理等。



蒋刚毅 (1964-), 男, 浙江绍兴人, 博士, 宁波大学教授、博士生导师, 主要研究方向为立体/多视点视频信号处理、3DTV/自由视点视频系统编码与传输等。



王晓东 (1970-), 男, 浙江上虞人, 硕士, 宁波大学副教授、硕士生导师, 主要研究方向为通信协议、网络安全等。



郁梅 (1968-), 女, 江苏无锡人, 博士, 宁波大学教授、博士生导师, 主要研究方向为图像视频信息过程编码压缩与通信等。



彭宗举 (1973-), 男, 四川南部人, 博士, 宁波大学副教授、硕士生导师, 主要研究方向为多媒体信号处理与编码等。



邵枫 (1980-), 男, 浙江杭州人, 博士, 宁波大学副教授、硕士生导师, 主要研究方向为三维视频信号编码、质量评价等。