



(10) 授权公告号 CN 103123720 B

(45) 授权公告日 2015.09.23

17 期).

尚冠宇等. 基于采样预测和差值变换的医学图像可逆水印算法. 《计算机应用与软件》. 2012, 第 29 卷 (第 6 期).

熊志勇等. 基于预测误差差值扩展和最低有效位替换的可逆数据隐藏. 《计算机应用》. 2010, 第 30 卷 (第 4 期).

审查员 胡百乐

(72)发明人 陈志刚 邓小鸿 翦鹏 梁涤青

(74) 专利代理机构 长沙市融智专利事务所  
43114

代理人 黃美成

(51) Int. Cl.

*G06T 1/00*(2006. 01)

(56) 对比文件

CN 101449278 A, 2009. 06. 03.

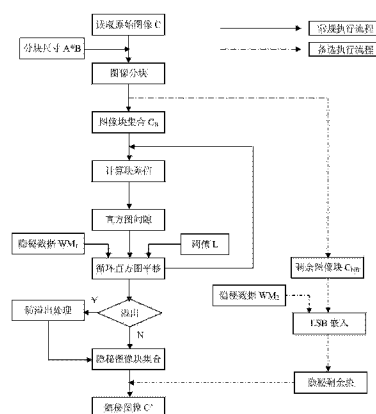
李晓博等. 基于图像块像素差的大容量无损信息隐藏算法.《计算机工程》,2012,第38卷(第

权利要求书3页 说明书10页 附图4页

# 基于差值图像直方图循环平移的无损数据隐藏方法

(57) 摘要

本发明公开了一种基于差值图像直方图循环平移的无损数据隐藏方法,包括:加密步骤:在发送端,将原图像分块为多个子图像块,利用子图像块差值直方图进行循环平移操作实现隐秘数据的逐级嵌入,形成隐秘图像  $C'$  并发送给接收端;密钥发送步骤:图像分块大小、嵌入数据长度和嵌入级别  $L$  作为密钥分发给接收端;数据恢复步骤:接收端通过嵌入过程的逆方法从隐秘图像  $C'$  中逐级提取出隐秘数据,并恢复块差值和原始图像数据,实现了基于图像载体的隐秘数据传输和无损恢复。本方法改进了原始直方图平移算法的容量依赖于单一峰值点的特点,利用循环嵌入方法增大了嵌入容量。实验结果表明本发明的方法很好满足无损数据隐藏方法的高容量、低失真和高效率的需求。



1. 一种基于差值图像直方图循环平移的无损数据隐藏方法,其特征在于,包括以下步骤:

加密步骤:在发送端,将原图像分块为多个子图像块,利用子图像块差值直方图进行循环平移操作实现隐秘数据的逐级嵌入,形成隐秘图像  $C'$  并发送给接收端;

密钥发送步骤:图像分块大小、嵌入数据长度和嵌入阈值  $L$  作为密钥分发给接收端;

数据恢复步骤:接收端通过嵌入过程的逆方法从隐秘图像  $C'$  中逐级提取出隐秘数据,并恢复块差值和原始图像数据,实现了基于图像载体的隐秘数据传输和无损恢复;

在所述的加密步骤包括:

步骤 1:读取原始载体图像  $C = \{C(i, j) | 1 \leq i \leq M, 1 \leq j \leq N\}$ ,  $M, N$  为载体图像的维数,即载体图像的长和宽;

步骤 2:将原始载体图像  $C$  分块,块尺寸为  $A*B$ ,块扫描顺序为 Zig-zag 方式,分块后图像块集合表示为  $C_b = \{C_b(i, j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A)*\text{floor}(N/B)\}$ , 这里  $\text{floor}(\cdot)$  为向下取整函数;

步骤 3:如果原始载体图像不能全部划分在图像块中,条件表示为  $\text{if } \text{mod}(M, A) \text{ or } \text{mod}(N, B) \neq 0$ , 这里  $\text{mod}(\cdot)$  为取余函数,则得到剩余部分的图像块的集合  $C_{NB} = C - C_b$ , 否则  $C_{NB}$  置为空,并且后续执行过程跳过步骤 12;

步骤 4:对  $C_b$  块中的元素计算图像块差值,即块中相邻的元素像素值差的绝对值:

$$D_b(i, j) = \text{abs}(C_b(i, j) - C_b(i, j+1)); \text{公式 1}$$

步骤 5:读取指定嵌入阈值  $L$ ,  $L$  为正整数;

步骤 6:形成图像块差值直方图,并在每块子图像的差值直方图中形成间隙,按照下式计算; $D_b'(i, j)$  为调整后得到的新的差值;

$$D_b'(i, j) = \begin{cases} D_b(i, j) + L + 1 & \text{如果 } D_b(i, j) > L \\ D_b(i, j) & \text{否则} \end{cases}; \text{公式 2}$$

步骤 7:读取待传输的隐秘数据  $WM = \{wm_k | 1 \leq k \leq WM\_Length\}$ ,  $wm_k \in \{0, 1\}$ ,  $WM\_Length$  为隐秘数据长度;

步骤 8:在每块差值直方图中嵌入数据,按照下式计算:

$$D_b''(i, j) = \begin{cases} D_b'(i, j) + P + 1, em\_count++ & \text{如果 } D_b'(i, j) = P \ \& \ wm = 1 \\ D_b'(i, j) + P, em\_count++ & \text{如果 } D_b'(i, j) = P \ \& \ wm = 0 \end{cases} \text{公式 3};$$

$D_b''(i, j)$  代表嵌入隐秘数据后的差值,  $wm$  代表隐秘数据的 1bit,  $em\_count$  记录差值直方图中可嵌入的隐秘数据位数,其初值为 0,该公式表明如果满足差值等于  $P$ ,根据要隐藏的信息是 1 或是 0,差值相应会增加  $P+1$  或者  $P$ ,并且可嵌入的隐秘数据位数增 1,即  $em\_count++$ ;  $P$  初值为  $L$ ,向下递减直到为 0,利用上式逐级嵌入;

步骤 9:产生每个图像块对应的隐秘图像块,首先按照公式 4 和 5 生成每个图像块中每行的前两个像素,然后按照公式 6 得到其他像素:

$$S_b(i, 1) = \begin{cases} C_b(i, 1) & \text{如果 } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) + D_b''(i, 1) & \text{否则} \end{cases} \text{公式 4};$$

$$S_b(i, 2) = \begin{cases} C_b(i, 1) + D_b''(i, 1) & \text{如果 } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) & \text{否则} \end{cases} \quad \text{公式 5 ;}$$

这里  $1 \leq i \leq A, 1 \leq b \leq \text{floor}(M \cdot N / A \cdot B)$

$$S_b(i, j) = \begin{cases} S_b(i, j-1) + D_b''(i, j-1) & \text{if } C_b(i, j-1) \leq C_b(i, j) \\ S_b(i, j-1) - D_b''(i, j-1) & \text{otherwise} \end{cases} \quad \text{公式 6 ;}$$

其中,  $1 \leq i \leq A, 3 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)$  ;

步骤 10、防止溢出处理, 在公式 4-6 执行的过程中会产生溢出现象, 公式 7 和 8 分别给出了上溢和下溢发生条件, 其中 Eq(4) 代表公式 4 ;

$$\text{overflow} = \begin{cases} C_b(i, 2) + D_b''(i, 1) > 255 & \text{in Eq(4)} \\ C_b(i, 1) + D_b''(i, 1) > 255 & \text{in Eq(5)} \\ S_b(i, j-1) + D_b''(i, j-1) > 255 & \text{in Eq(6)} \end{cases} \quad \text{公式 7 ;}$$

$\text{underflow} = S_b(i, j-1) - D_b''(i, j-1) < 0$  in Eq(6) 公式 8 ;

在发生溢出现象后, 将公式 4-6 分别用公式 9-11 替换 :

$$S_b(i, 1) = \begin{cases} C_b(i, 1) & \text{if } C_b(i, 1) \leq C_b(i, 2) \\ C * \text{floor}(C_b(i, 2) / C) + \text{mod}(C_b(i, 2) + D_b''(i, 1), C) & \text{otherwise} \end{cases} \quad \text{公式 9 ;}$$

$$S_b(i, 2) = \begin{cases} C * \text{floor}(C_b(i, 1) / C) + \text{mod}(C_b(i, 1) + D_b''(i, 1), C) & \text{if } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) & \text{otherwise} \end{cases} \quad \text{公式 10 ;}$$

$$S_b(i, j) = \begin{cases} C * \text{floor}(S_b(i, j-1) / C) + \text{mod}(S_b(i, j-1) + D_b''(i, j-1), C) & \text{if } C_b(i, j-1) \leq C_b(i, j) \\ C * \text{floor}(-S_b(i, j-1) / C) + \text{mod}(S_b(i, j-1) - D_b''(i, j-1), C) & \text{otherwise} \end{cases}$$

公式 11 ;

其中 C 取为具体载体图像的最大差值  $D_{\max} + L$  ;

步骤 11 : 合并所有的隐秘图像块  $S_b(i, j)$  形成隐秘图像  $C_b' = \{S_b(i, j) \mid 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)\}$  ;

步骤 12 : 对  $C_{NB}$  中的像素值, 依次提取所有  $C_{NB}$  中像素的最低有效位  $C_{NB\_LSB}$ , 采用无损压缩方法, 产生嵌入空间, 其长度为 LS, 将 LS bit 隐秘信息和  $C_{NB\_LSB}$  采用最低有效位 LSB 替换算法嵌入到  $C_{NB}$  中, 形成隐秘图像  $C_{NB}'$  ;

步骤 13、组合  $C_b'$  和  $C_{NB}'$  形成整幅隐秘图像  $C' = C_b' \cup C_{NB}'$ 。

2. 根据权利要求 1 所述的基于差值图像直方图循环平移的无损数据隐藏方法, 其特征在于, L 初始值设为 2。

3. 根据权利要求 1 所述的基于差值图像直方图循环平移的无损数据隐藏方法, 其特征在于, 在密钥发送步骤中, 将分块尺寸  $A \cdot B$ 、嵌入阈值 L 和隐秘信息长度  $\text{em\_count} + \text{LS}$  作为图像接收端的解密密钥, 传给接收方, LS 初值设为 0, 最终值为实际在  $C_{NB}$  中隐藏的数据位数 ;

利用公钥密码体制, 将图像分块大小、嵌入数据长度和嵌入阈值 L 作为秘密消息 M, 利用下式传输 :

$$\begin{cases} E_{PK_B}(M) = HM \\ D_{SK_B}(HM) = M \end{cases} ;$$

其中 E 代表加密操作, D 代表解密操作,  $PK_B$  为接收端公钥,  $SK_B$  为接收端私钥, HM 为公网上传输的密文。

4. 根据权利要求 1-3 任一项所述的基于差值图像直方图循环平移的无损数据隐藏方法, 其特征在于, 首先用公式 12 恢复隐藏了数据的块差值, 然后用公式 13 恢复公式 1 所产生的直方图间隙:

$$S_{Db}'(i, j) = \begin{cases} S_{Db}(i, j) - (P+1) & \text{if } S_{Db}(i, j) = 2P+1 \\ S_{Db}(i, j) - P & \text{otherwise} \end{cases} \quad \text{公式 12}$$

$$R_{Db}(i, j) = \begin{cases} S_{Db}'(i, j) - (P+1) & \text{if } S_{Db}(i, j) > 2P+1 \\ S_{Db}'(i, j) & \text{otherwise} \end{cases} \quad \text{公式 13};$$

其中公式 11 中的 P 从 0 开始逐步递增为 L; 公式 13 中的  $P = L$ ;  $S_{Db}(i, j)$  为  $C_B'$  中图像块的差值:  $S_{Db}(i, j) = \text{abs}(S_b(i, j) - S_b(i, j+1))$ 。

## 基于差值图像直方图循环平移的无损数据隐藏方法

### 技术领域

[0001] 本发明涉及一种基于差值图像直方图循环平移的无损数据隐藏方法,属于图像无损数据隐藏(加密)领域。

### 背景技术

[0002] 无损图像数据隐藏技术,也称为可逆图像数据隐藏,通过特定的无损数据嵌入方法将隐秘数据嵌入在载体图像中形成隐秘图像,隐秘图像在提取完数据后能恢复到原始状态。无损数据隐藏技术和先前的有损方法的最大区别就是载体本身的数据不会遭到任何变化,极大的扩展了数据隐藏的应用范围。比如在一些特定的医学、法律、军事(不涉及到军事机密,仅仅为了说明军事图像对完整性要求严格)和艺术等载体图像由于自身对完整性具有极高的要求,一方面要求能进行隐秘通信,另一方面要保证载体数据的完整性。

[0003] 图像载体具有较高的冗余性,特别适合于大容量的数据隐藏,常见的无损数据隐藏方法分为两大类,一类是基于空域的算法,此类算法直接对图像像素进行修改完成数据嵌入,算法嵌入容量大,计算简单,常用于图像的数据隐藏,但其最大的特点是鲁棒性差,即隐秘图像在受到任何程度的篡改时均会影响隐秘数据的提取和图像恢复;第二类是基于频域的算法,此类算法通过将图像像素转换成对应的变换域,然后对变换域的系数进行修改完成数据的嵌入,算法嵌入容量较小,计算复杂,常用于图像的版权保护,其优点就是鲁棒性较好,能抵抗一定程度的攻击。基于空域的算法中最常用的就是直方图平移和差值扩展方法,前者比后者计算更为简单,图像质量更好,但其缺点就是嵌入容量依赖于位于图像直方图中峰值点像素的个数。

[0004] 为了增大直方图方法的嵌入容量,提出利用图像中相邻像素相关性和冗余性高的特点,在差值直方图中嵌入数据,虽然这种方法大大增大了处于峰值点像素的个数,但采用单一峰值点嵌入仍然损失了很多可用来嵌入数据的像素。

[0005] 因此,有必要设计一种基于差值图像直方图循环平移的无损数据隐藏方法。

### 发明内容

[0006] 本发明所要解决的技术问题是提供一种基于差值图像直方图循环平移的无损数据隐藏方法,该基于差值图像直方图循环平移的无损数据隐藏方法,基于数字图像载体,利用小图像块中相邻像素差值收敛于一个较小的值的特点,利用循环直方图平移方法进一步增大嵌入容量,并保持隐秘图像较好的视觉质量,有效用于数字图像的隐秘通信。

[0007] 发明的技术解决方案如下:

[0008] 一种基于差值图像直方图循环平移的无损数据隐藏方法,包括以下步骤:

[0009] 加密步骤:在发送端,将原图像分块为多个子图像块,利用子图像块差值直方图进行循环平移操作实现隐秘数据的逐级嵌入,形成隐秘图像  $C'$  并发送给接收端;

[0010] 密钥发送步骤:图像分块大小、嵌入数据长度和嵌入级别  $L$  作为密钥分发给接收端;

[0011] 数据恢复步骤:接收端通过嵌入过程的逆方法从隐秘图像  $C'$  中逐级提取出隐秘数据,并恢复块差值和原始图像数据,实现了基于图像载体的隐秘数据传输和无损恢复。

[0012] 在所述的加密步骤包括:

[0013] 步骤 1:读取原始载体图像  $C = \{C(i, j) | 1 \leq i \leq M, 1 \leq j \leq N\}$ ,  $M, N$  为载体图像的维数,即载体图像的长和宽;

[0014] 步骤 2:将原始载体图像  $C$  分块,块尺寸为  $A*B$ ,块扫描顺序为 Zig-zag 方式,分块后图像块集合表示为  $C_b = \{C_b(i, j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A)*\text{floor}(N/B)\}$ ,这里  $\text{floor}(\cdot)$  为向下取整函数;

[0015] 步骤 3、如果原始载体图像不能全部划分在图像块中,条件表示为  $\text{if } \text{mod}(M, A) \text{ or } \text{mod}(N, B) \neq 0$ ,这里  $\text{mod}(\cdot)$  为取余函数,则得到剩余部分的图像块的集合  $C_{NB} = C - C_b$ ,否则  $C_{NB}$  置为空,并且后续执行过程跳过步骤 12;

[0016] 步骤 4:对  $C_b$  块中的元素计算图像块差值,即块中相邻的元素像素值差的绝对值:

[0017]  $D_b(i, j) = \text{abs}(C_b(i, j) - C_b(i, j+1))$ ;公式 1

[0018] 步骤 5:读取指定嵌入阈值  $L$ ,  $L$  为正整数;

[0019] 步骤 6:形成图像块差值直方图,并在每块子图像的差值直方图中形成间隙(间隙不是新的差值,间隙的形成是由于将所有大于  $L$  的差值进行了增加  $L+1$  操作,使得差值直方图中值为  $L$  的差值右边出现了连续的  $L+1$  个空白区域),按照下式计算; $D_b'(i, j)$  为调整后得到的新的差值;

[0020]  $D_b'(i, j) = \begin{cases} D_b(i, j) + L + 1 & \text{如果 } D_b(i, j) > L \\ D_b(i, j) & \text{否则} \end{cases}$ ;公式 2

[0021] 步骤 7:读取待传输的隐秘数据  $WM = \{wm_k | 1 \leq k \leq WM\_Length\}$ ,  $wm_k \in \{0, 1\}$ ,  $WM\_Length$  为隐秘数据长度;

[0022] 步骤 8:在每块差值直方图中嵌入数据,按照下式计算:

[0023]  $D_b''(i, j) = \begin{cases} D_b'(i, j) + P + 1, em\_count++ & \text{如果 } D_b'(i, j) = P \ \& \ wm = 1 \\ D_b'(i, j) + P, em\_count++ & \text{如果 } D_b'(i, j) = P \ \& \ wm = 0 \end{cases}$  公式 3;

[0024]  $D_b''(i, j)$  代表嵌入隐秘数据后的差值,  $wm$  代表隐秘数据的 1bit,  $em\_count$  记录差值直方图中可嵌入的隐秘数据位数,其初值为 0,该公式表明如果满足差值等于  $P$ ,根据要隐藏的信息是 1 或是 0,差值相应会增加  $P+1$  或者  $P$ ,并且可嵌入的隐秘数据位数增 1,即  $em\_count++$ ;  $P$  初值为  $L$ ,向下递减直到为 0,利用上式逐级嵌入;

[0025] 步骤 9:产生每个图像块对应的隐秘图像块,首先按照公式 4 和 5 生成每个图像块中每行的前两个像素,然后按照公式 6 得到其他像素:

[0026]  $S_b(i, 1) = \begin{cases} C_b(i, 1) & \text{如果 } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) + D_b''(i, 1) & \text{否则} \end{cases}$  公式 4;

[0027]  $S_b(i, 2) = \begin{cases} C_b(i, 1) + D_b''(i, 1) & \text{如果 } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) & \text{否则} \end{cases}$  公式 5;

[0028] 这里  $1 \leq i \leq A, 1 \leq b \leq \text{floor}(M*N/A*B)$

[0029]  $S_b(i, j) = \begin{cases} S_b(i, j-1) + D_b''(i, j-1) & \text{if } C_b(i, j-1) \leq C_b(i, j) \\ S_b(i, j-1) - D_b''(i, j-1) & \text{otherwise} \end{cases}$  公式 6;

[0030] 其中,  $1 \leq i \leq A, 3 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)$  ;

[0031] 步骤 10、防止溢出处理,在公式 4-6 执行的过程中会产生溢出现象,公式 7 和 8 分别给出了上溢和下溢发生条件,其中 Eq(4) 代表公式 4 ;

$$[0032] \quad \text{overflow} = \begin{cases} C_b(i, 2) + D_b''(i, 1) > 255 & \text{in Eq(4)} \\ C_b(i, 1) + D_b''(i, 1) > 255 & \text{in Eq(5)} \\ S_b(i, j-1) + D_b''(i, j-1) > 255 & \text{in Eq(6)} \end{cases} \quad \text{公式 7 ;}$$

[0033]  $\text{underflow} = S_b(i, j-1) - D_b''(i, j-1) < 0$  in Eq(6) 公式 8 ;

[0034] 在发生溢出现象后,将公式 4-6 分别用公式 9-11 替换 :

$$[0035] \quad S_b(i, 1) = \begin{cases} C_b(i, 1) & \text{if } C_b(i, 1) \leq C_b(i, 2) \\ C * \text{floor}(C_b(i, 2) / C) + \text{mod}(C_b(i, 2) + D_b''(i, 1), C) & \text{otherwise} \end{cases} \quad \text{公式 9 ;}$$

$$[0036] \quad S_b(i, 2) = \begin{cases} C * \text{floor}(C_b(i, 1) / C) + \text{mod}(C_b(i, 1) + D_b''(i, 1), C) & \text{if } C_b(i, 1) \leq C_b(i, 2) \\ C_b(i, 2) & \text{otherwise} \end{cases}$$

[0037] 公式 10 ;

[0038]

$$S_b(i, j) = \begin{cases} C * \text{floor}(S_b(i, j-1) / C) + \text{mod}(S_b(i, j-1) + D_b''(i, j-1), C) & \text{if } C_b(i, j-1) \leq C_b(i, j) \\ C * \text{floor}(-S_b(i, j-1) / C) + \text{mod}(S_b(i, j-1) - D_b''(i, j-1), C) & \text{otherwise} \end{cases}$$

[0039] 公式 11 ;

[0040] 其中 C 取为具体载体图像的最大差值  $D_{\max} + L$  ;

[0041] 步骤 11 :合并所有的隐秘图像块  $S_b(i, j)$  形成隐秘图像  $C_B' = \{S_b(i, j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)\}$  ( 把所有的隐秘图像块  $S_b(i, j)$  合并即形成了整个隐秘图像  $C_B'$  ) ;

[0042] 步骤 12 :对  $C_{NB}$  中的像素值,依次提取所有  $C_{NB}$  中像素的最低有效位  $C_{NB\_LSB}$ ,采用无损压缩方法,产生嵌入空间,其长度为 LS,将 LSbit 隐秘信息和  $C_{NB\_LSB}$  采用最低有效位 LSB 替换算法嵌入到  $C_{NB}$  中,形成隐秘图像  $C_{NB}'$  ;

[0043] 步骤 13、组合  $C_B'$  和  $C_{NB}'$  形成整幅隐秘图像  $C' = C_B' \cup C_{NB}'$  。

[0044] L 初始值设为 2。

[0045] 在密钥发送步骤中,将分块尺寸  $A*B$ 、嵌入阈值 L 和隐秘信息长度  $\text{em\_count} + LS$  ( 作为图像接收端的解密密钥,传给接收方,LS 初值设为 0,最终值为实际在  $C_{NB}$  中隐藏的数据位数 ;

[0046] 利用公钥密码体制,将图像分块大小、嵌入数据长度、和嵌入级别 L 作为秘密消息 M,利用下式传输 :

$$[0047] \quad \begin{cases} E_{PK_B}(M) = HM \\ D_{SK_B}(HM) = M \end{cases} ;$$

[0048] 其中 E 代表加密操作, D 代表解密操作,  $PK_B$  为接收端公钥,  $SK_B$  为接收端私钥, HM 为公网上传输的密文。

[0049] 首先用公式 12 恢复隐藏了数据的块差值,然后用公式 13 恢复公式 1 所产生的直方图间隙 :

$$[0050] \quad S_{Db}'(i, j) = \begin{cases} S_{Db}(i, j) - (P+1) & \text{if } S_{Db}(i, j) = 2P+1 \\ S_{Db}(i, j) - P & \text{otherwise} \end{cases} \quad \text{公式 12}$$

$$[0051] \quad R_{Db}(i, j) = \begin{cases} S_{Db}'(i, j) - (P+1) & \text{if } S_{Db}(i, j) > 2P+1 \\ S_{Db}'(i, j) & \text{otherwise} \end{cases} \quad \text{公式 13 ;}$$

[0052] 其中公式 11 中的 P 从 0 开始逐步递增为 L ; 公式 13 中的  $P = L$  ;  $S_{Db}(i, j)$  为  $C_B'$  中图像块的差值 :  $S_{Db}(i, j) = \text{abs}(S_b(i, j) - S_b(i, j+1))$ 。

[0053] 本发明的简要步骤如下 :

[0054] a) 图像分块。

[0055] b) 数据嵌入。

[0056] c) 数据提取。

[0057] 上述图像分块方法进一步包含如下步骤 :

[0058] a1) 按照分块尺寸  $A*B$  获取每个子图像块 ;

[0059] a2) 按照 Zig-zag 扫描顺序组织每个子图像块, 然后依次进行数据嵌入。

[0060] 上述数据嵌入方法进一步包含如下步骤 :

[0061] b1) 计算每个子图像块差值, 形成差值直方图 ;

[0062] b2) 根据给定的嵌入阈值 L 形成直方图间隙 ;

[0063] b3) 设定调整幅度  $P = L$ , 扫描所有等于 P 的差值, 根据水印信息调整差值直方图嵌入水印 ;

[0064] b4) 依次减小 P 的值, 直到  $P = 0$ , 重复 b3 步骤 ;

[0065] b5) 根据调整后差值计算每个子图像块的像素值 ;

[0066] b6) 重复 b1 ~ b5 步骤, 形成隐秘图像。

[0067] 上述数据提取方法进一步包含如下步骤 :

[0068] c1) 将隐秘图像进行 a) 步骤的分块操作 ;

[0069] c2) 计算每个隐秘子图像块差值 ;

[0070] c3) 设定提取阈值  $P = 0$ , 扫描所有等于  $2*P$  和  $2*P+1$  的差值, 如果差值等于  $2*P$ , 提取数据 0, 并将差值减去 P, 否则提取 1, 并将差值减去  $P+1$  ;

[0071] c4) 依次增大 P 的值, 直到  $P = L$ , 重复 c3 步骤 ;

[0072] c5) 根据调整后差值计算每个子图像块的像素值 ;

[0073] c6) 重复 c1 ~ c5 步骤, 得到恢复后图像。

[0074] 有益效果 :

[0075] 本发明的基于差值图像直方图循环平移的无损数据隐藏方法, 本发明涉及无损图像数据隐藏技术, 提供一种在高完整性需求背景下的数字图像中无损隐藏大容量数据的方法及装置。所述方法及装置首先利用子图像块相邻像素之间的高相关性和冗余性, 产生块差值图像直方图 ; 然后, 根据给定的嵌入阈值 L, 在差值图像直方图中形成嵌入间隙 ; 其次, 逐步减小 L 的值直到 0, 利用直方图平移方法逐级嵌入数据, 最后, 根据调整后的差值和严格的防止溢出处理形成隐秘图像。本方法在隐秘图像没有遭到篡改时, 拥有解密密钥的图像接收端可以正确提取出所嵌入的数据, 并且无失真地还原出原始载体图像。本方法改进了原始直方图平移算法的容量依赖于单一峰值点的特点, 利用循环嵌入方法增大了嵌入容量。实验结果表明本发明的方法很好满足无损数据隐藏方法的高容量、低失真和高效率的

需求。

[0076] 本发明在图像的差值直方图中,采用循环直方图平移方法逐级嵌入数据,利用到了所有小于等于给定嵌入阈值点的差值,极大增大了差值直方图嵌入数据的容量;同时,本方法通过合理嵌入阈值的选择,在保证嵌入容量的同时,将隐秘图像的质量控制在合理的范围内。

[0077] 本发明的优点在于数据嵌入容量大,由于本方法是基于差值直方图平移方法,但传统的差值直方图平移方法只会用到单个或选定的差值嵌入,本方法利用了所有小于等于阈值的差值,增大了嵌入容量;其次,本方法采用 module-C 方法(C 为图像相邻像素差值,远小于 256)进行防止溢出处理,该方法比传统的 module-256 方法具有更高的图像质量;最后,本方法所有的操作均是简单的算术数学运算,具有较高的效率。图 5 和表 1 给出了本方法的部分实验结果,图 5 说明了嵌入阈值  $L = 2$  的情况下的隐秘图像质量,其峰值信噪比  $PSNR \geq 37dB$ 。表 1 对嵌入容量进行了说明,其中 Airplane 载体图像当嵌入阈值等于 3 时,其嵌入容量可达到 0.5bpp 左右。

#### 附图说明

[0078] 图 1 是本发明中涉及的数据嵌入执行过程图。

[0079] 图 2 是本发明中涉及的数据提取和图像恢复执行过程图。

[0080] 图 3 是本发明中子图像块嵌入数据实例图。

[0081] 图 4 是本发明中子图像块提取数据实例图。

[0082] 图 5 是本发明方法在 5 种测试图像和嵌入阈值  $L = 2$  时的嵌入情况下的隐秘图像对比图。

[0083] 在图 5 中,

[0084] (a) 代表原始 Lena 图像  $512 \times 512 \times 8bit$

[0085] (b) 代表嵌入 97315bit 数据和  $PSNR = 37.50dB$  的隐秘 Lena 图像

[0086] (c) 代表原始 Baboon 图像  $512 \times 512 \times 8bit$

[0087] (d) 代表嵌入 30898bit 数据和  $PSNR = 37.24dB$  的隐秘 Baboon 图像

[0088] (e) 代表原始 Boat 图像  $512 \times 512 \times 8bit$

[0089] (f) 代表嵌入 54303bit 数据和  $PSNR = 38.74dB$  的隐秘 Boat 图像

[0090] (g) 代表原始 Airplane 图像  $512 \times 512 \times 8bit$

[0091] (h) 代表嵌入 108696bit 数据和  $PSNR = 38.52dB$  的隐秘 Airplane 图像

[0092] (i) 代表原始 Pepper 图像  $512 \times 512 \times 8bit$

[0093] (j) 代表嵌入 68634bit 数据和  $PSNR = 38.13dB$  的隐秘 Pepper 图像

[0094] 附表说明

[0095] 表 1 是本发明中所用的测试图像在不同嵌入阈值  $L$  下的最大嵌入容量和峰值信噪比。

#### 具体实施方式

[0096] 以下将结合附图和具体实施例对本发明做进一步详细说明:

[0097] 虽然本发明是关于特定实施例进行描述的,但是本发明的原则,如所附的权利要

求书定义的,显然可以应用于超出在此特定表述的实施例的范围。而且,这里略去了一些细节为了不模糊本发明的发明方面。在本申请内没有描述的特定细节在领域内的任何普通技术人员的所知范围内。

[0098] 本申请的附图和其所附的详细描述仅用于示出本发明的示例实施例。为保持简洁性,使用本发明原理的本发明的其他实施例不在本申请中特定描述,且不由本附图特定说明。“示例”在此仅指“用作为例子、实例或说明”。

[0099] 参见图 1,图 1 为本发明中涉及的数据嵌入执行过程图。该数据嵌入过程包含如下步骤 (Em\_1 代表嵌入步骤的) :

[0100] Em\_1、读取原始载体图像  $C = \{C(i, j) | 1 \leq i \leq M, 1 \leq j \leq N\}$ ,  $M, N$  为载体图像的维数,即图像的长和宽;

[0101] Em\_2、将  $C$  分块,块尺寸为  $A*B$ ,块扫描顺序为 Zig-zag 方式,分块后图像块集合表示为  $C_b = \{C_b(i, j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A)*\text{floor}(N/B)\}$ , 这里  $\text{floor}(*)$  为向下取整函数;

[0102] Em\_3、如果原始载体图像不能全部划分在图像块中,条件表示为  $\text{if } \text{mod}(M, A) \text{ or } \text{mod}(N, B) \neq 0$ , 这里  $\text{mod}(*)$  为取余函数,则执行图 1 中“备选执行流程”,得到剩余部分的图像块的集合  $C_{NB} = C - C_b$ , 否则  $C_{NB}$  置为空,并且后续执行过程跳过步骤 Em\_12;

[0103] Em\_4、对  $C_b$  块中的元素计算图像块差值,即块中相邻的元素像素值差的绝对值:

[0104]  $D_b(i, j) = \text{abs}(C_b(i, j) - C_b(i, j+1))$  (1)

[0105] 这里,  $1 \leq i \leq A, 1 \leq j \leq B-1, 1 \leq b \leq \text{foor}(M/A)*\text{floor}(N/B)$ ,  $\text{abs}(*)$  为取绝对值函数;

[0106] Em\_5、读取指定嵌入阈值  $L$ ,  $L$  为正整数,通过大量实验分析,  $L$  取 2 时容量和峰值信噪比均达到较好的效果,本专利中不做特别申明,  $L$  初始值设为 2;

[0107] Em\_6、形成图像块差值直方图,并在每块子图像的差值直方图中形成间隙,按照公式 (2) 计算; $D_b'(i, j)$  为调整后得到的新的差值。

[0108] 
$$D_b'(i, j) = \begin{cases} D_b(i, j) + L + 1 & \text{if } D_b(i, j) > L \\ D_b(i, j) & \text{otherwise} \end{cases} \quad (2)$$

[0109] Em\_7、读取隐秘数据  $WM = \{wm_k | 1 \leq k \leq WM\_Length\}$ ,  $wm_k \in \{0, 1\}$   $WM\_Length$  为隐秘数据长度;

[0110] Em\_8、在每块差值直方图中嵌入数据,按照公式 (3) 计算:

[0111]

$$D_b''(i, j) = \begin{cases} D_b'(i, j) + P + 1, em\_count++ & \text{if } D_b'(i, j) = P \& wm = 1 \\ D_b'(i, j) + P, em\_count++ & \text{if } D_b'(i, j) = P \& wm = 0 \end{cases} \quad (3)$$

[0112]  $D_b''(i, j)$  代表嵌入隐秘数据后的差值,  $wm$  代表隐秘数据的 1bit,  $em\_count$  记录差值直方图中可嵌入的隐秘数据位数,其初值为 0, 公式 (3) 表明如果满足差值等于  $P$ , 根据要隐藏的信息是 1 或是 0, 差值相应会增加  $P+1$  或者  $P$ , 并且可嵌入的隐秘数据位数增 1;  $P$  初值为  $L$ , 向下递减直到为 0, 利用 (3) 逐级嵌入;

[0113] Em\_9、产生每个图像块对应的隐秘图像块,首先按照公式 (4) (5) 生成每个图像块中每行的前两个像素,然后按照公式 (6) 得到其他像素:

$$[0114] \quad S_b(i,1) = \begin{cases} C_b(i,1) & \text{if } C_b(i,1) \leq C_b(i,2) \\ C_b(i,2) + D_b''(i,1) & \text{otherwise} \end{cases} \quad (4)$$

$$[0115] \quad S_b(i,2) = \begin{cases} C_b(i,1) + D_b''(i,1) & \text{if } C_b(i,1) \leq C_b(i,2) \\ C_b(i,2) & \text{otherwise} \end{cases} \quad (5)$$

[0116] 这里  $1 \leq i \leq A, 1 \leq b \leq \text{floor}(M*N/A*B)$

$$[0117] \quad S_b(i,j) = \begin{cases} S_b(i,j-1) + D_b''(i,j-1) & \text{if } C_b(i,j-1) \leq C_b(i,j) \\ S_b(i,j-1) - D_b''(i,j-1) & \text{otherwise} \end{cases} \quad (6)$$

[0118] 这里  $1 \leq i \leq A, 3 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A)*\text{floor}(N/B)$  ;

[0119] Em\_10、防止溢出处理,在公式 (4) (5) (6) 执行的过程中会产生溢出现象,比如处于上界边缘的像素在加上一个差值时会超过 8bit 灰度图像的上限 255,公式 (7) (8) 分别给出了上溢和下溢发生条件 (注:Eq(4) 代表上述提到的第 4 个公式) :

$$[0120] \quad \text{overflow} = \begin{cases} C_b(i,2) + D_b''(i,1) > 255 & \text{in Eq(4)} \\ C_b(i,1) + D_b''(i,1) > 255 & \text{in Eq(5)} \\ S_b(i,j-1) + D_b''(i,j-1) > 255 & \text{in Eq(6)} \end{cases} \quad (7)$$

[0121]  $\text{underflow} = S_b(i,j-1) - D_b''(i,j-1) < 0$  in Eq(6) (8)

[0122] 在发生溢出现象后,将公式 (4) (5) (6) 分别用公式 (9) (10) (11) 替换 :

$$[0123] \quad S_b(i,1) = \begin{cases} C_b(i,1) & \text{if } C_b(i,1) \leq C_b(i,2) \\ C * \text{floor}(C_b(i,2)/C) + \text{mod}(C_b(i,2) + D_b''(i,1), C) & \text{otherwise} \end{cases} \quad (9)$$

$$[0124] \quad S_b(i,2) = \begin{cases} C * \text{floor}(C_b(i,1)/C) + \text{mod}(C_b(i,1) + D_b''(i,1), C) & \text{if } C_b(i,1) \leq C_b(i,2) \\ C_b(i,2) & \text{otherwise} \end{cases} \quad (10)$$

[0125]

$$S_b(i,j) = \begin{cases} C * \text{floor}(S_b(i,j-1)/C) + \text{mod}(S_b(i,j-1) + D_b''(i,j-1), C) & \text{if } C_b(i,j-1) \leq C_b(i,j) \\ C * \text{floor}(-S_b(i,j-1)/C) + \text{mod}(S_b(i,j-1) - D_b''(i,j-1), C) & \text{otherwise} \end{cases} \quad (11)$$

[0126] 这里 C 取为具体载体图像的最大差值  $D_{\max} + L$ , L 为指定的嵌入阈值 ;

[0127] Em\_1 : 合并所有的隐秘图像块  $S_b(i,j)$  形成隐秘图像  $C_B' = \{S_b(i,j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A)*\text{floor}(N/B)\}$  ( 将所有的隐秘图像块  $S_b(i,j)$  合并即形成了整个隐秘图像  $C_B'$  ) ;

[0128] Em\_12、对  $C_{NB}$  中的像素值,依次提取所有  $C_{NB}$  中像素的最低有效位  $C_{NB\_LSB}$ ,采用无损压缩方法, ( 如游程编码,先在  $C_{NB}$  中依次取 N 个像素的最低有效位,进行压缩,得到的数据长度为 CN,如果 N-CN 等于需要隐秘的数据长度 LS,则停止,否则增大 N 的值,直到能压缩产生 LS 个额外空间 ) 产生嵌入空间,其长度为 LS,将 LS bit 隐秘信息和  $C_{NB\_LSB}$  采用最低有效位 LSB 替换算法嵌入到  $C_{NB}$  中,形成隐秘图像  $C_{NB}'$  ;

[0129] Em\_13、组合  $C_B'$  和  $C_{NB}'$  形成整幅隐秘图像  $C' = C_B' \cup C_{NB}'$  ;

[0130] Em\_14、将分块尺寸  $A*B$ , 嵌入阈值 L 和隐秘信息长度  $\text{em\_count} + LS$  (LS 初值设为 0, 最终值为实际在  $C_{NB}$  中隐藏的数据位数) 作为图像接收端的解密密钥,通过安全传输机制传给接收方。

[0131] 参见图 2,图 2 为本发明中涉及的数据提取和图像恢复执行过程图。该数据提取和图像恢复过程包含如下步骤 (E&R\_1 代表提取和恢复过程中的第一步) :7

[0132] E&R\_1、读取隐秘图像  $C'$ ，其维数为  $M \times N$ ；

[0133] E&R\_2、读取解密密钥，获得分块尺寸  $A \times B$ 、嵌入级别  $L$  和隐秘数据长度  $em\_count + LS$ ；

[0134] E&R\_3、隐秘图像  $C'$  分块，分块顺序与嵌入阶段  $Em\_2$  相同，得到  $C'_b$ ；

[0135] E&R\_4、if  $\text{mod}(M, A) \parallel \text{mod}(N, B) \neq 0$ ，这里  $\text{mod}(\ast)$  为取余函数，得到  $C'_{NB}$ ，否则  $RC_{NB}$  设为空，并且后续执行过程跳过步骤 E&R\_12；

[0136] E&R\_5、按公式 (12) 计算所有  $C'_b$  中图像块的差值：

$$[0137] \quad S_{Db}(i, j) = \text{abs}(S_b(i, j) - S_b(i, j+1)) \quad (12)$$

[0138] 这里  $1 \leq i \leq A, 1 \leq j \leq B-1, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)$ ；

[0139] E&R\_6、提取隐秘信息并恢复部分差值，提取和恢复公式如 (13) (14) 所示：

$$[0140] \quad wm = \begin{cases} 0, ex\_count++ & \text{if } S_{Db}(i, j) = 2P \\ 1, ex\_count++ & \text{if } S_{Db}(i, j) = 2P+1 \end{cases} \quad (13)$$

$$[0141] \quad S'_{Db}(i, j) = \begin{cases} S_{Db}(i, j) - (P+1) & \text{if } S_{Db}(i, j) = 2P+1 \\ S_{Db}(i, j) - P & \text{otherwise} \end{cases} \quad (14)$$

[0142] 这里  $1 \leq i \leq A, 1 \leq j \leq B-1, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)$ ， $ex\_count$  为提取出隐秘信息的长度，初值为 0； $P$  从 0 开始递增直到  $L$ ；

[0143] E&R\_7、形成隐秘数据集  $WM'_1 = \{w'_k \mid 1 \leq k \leq count\}$ ；

[0144] E&R\_8、差值还原（将第  $Em\_5$  步中的间隙还原），按照公式 (15) 计算：

$$[0145] \quad R_{Db}(i, j) = \begin{cases} S'_{Db}(i, j) - (L+1) & \text{if } S_{Db}(i, j) > 2L+1 \\ S'_{Db}(i, j) & \text{otherwise} \end{cases} \quad (15)$$

[0146] E&R\_9、如果  $R_{Db}(i, j)$  中的差值小于等于  $C-L$ ，恢复  $C'_b$  中每块图像，按公式 (16) (17) 恢复每块每行的前两个像素，然后按照公式 (18) 恢复其他像素：

$$[0147] \quad R_{Cb}(i, 1) = \begin{cases} S_b(i, 2) + R_{Db}(i, 1) & \text{if } S_b(i, 1) \geq S_b(i, 2) \\ S_b(i, 1) & \text{otherwise} \end{cases} \quad (16)$$

$$[0148] \quad R_{Cb}(i, 2) = \begin{cases} S_b(i, 1) + R_{Db}(i, 1) & \text{if } S_b(i, 1) \leq S_b(i, 2) \\ S_b(i, 2) & \text{otherwise} \end{cases} \quad (17)$$

$$[0149] \quad R_{Cb}(i, j) = \begin{cases} R_{Cb}(i, j-1) + R_{Db}(i, j-1) & \text{if } S_b(i, j-1) \leq S_b(i, j) \\ R_{Cb}(i, j-1) - R_{Db}(i, j-1) & \text{otherwise} \end{cases} \quad (18)$$

[0150] 这里  $1 \leq i \leq A, 3 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)$ ；

[0151] E&R\_10、如果  $R_{Db}(i, j)$  中的差值大于  $C-L$ ，代表有溢出发生，则把公式 (16) (17) (18) 修改为公示 (19) (20) (21)；

$$[0152] \quad R_{Cb}(i, 1) = \begin{cases} S_b(i, 1) & \text{if } S_b(i, 1) < S_b(i, 2) \\ C * \text{floor}(S_b(i, 2) / C) + \text{mod}(S_b(i, 2) + R_{Db}(i, 1), C) & \text{otherwise} \end{cases} \quad (19)$$

$$[0153] \quad R_{Cb}(i, 2) = \begin{cases} C * \text{floor}(S_b(i, 1) / C) + \text{mod}(S_b(i, 1) + R_{Db}(i, 1), C) & \text{if } S_b(i, 1) \leq S_b(i, 2) \\ S_b(i, 2) & \text{otherwise} \end{cases} \quad (20)$$

[0154]

$$R_{Cb}(i, j) = \begin{cases} C * \text{floor}(R_{Cb}(i, j-1) / C) + \text{mod}(R_{Cb}(i, j-1) + R_{Db}(i, j-1), C) & \text{if } S_b(i, j-1) \leq S_b(i, j) \\ C * \text{floor}(-R_{Cb}(i, j-1) / C) + \text{mod}(R_{Cb}(i, j-1) - R_{Db}(i, j-1), C) & \text{otherwise} \end{cases} \quad (21)$$

[0155] E&R\_11、恢复图像  $R_{CB} = \{R_{cb}(i, j) | 1 \leq i \leq A, 1 \leq j \leq B, 1 \leq b \leq \text{floor}(M/A) * \text{floor}(N/B)\}$

[0156] E&R\_12、在  $C_{NB}'$  中采用 LSB 提取算法提取隐秘数据  $WM_2'$ ，并恢复  $C_{NB}'$  得到  $RC_{NB}$ ；

[0157] E&R\_13、重组提取出隐秘数据  $WM' = WM_1' \oplus WM_2'$  ( $WM_2'$  初值设为 0)， $\oplus$  表示拼接操作，如果  $WM'$  长度等于  $em\_count + LS$ ，数据提取正确；

[0158] E&R\_14、重组恢复出图像  $RC = RC_B \cup RC_{NB}$ 。

[0159] 参见图 3，图 3 为本发明中涉及子图像块嵌入数据实例。

[0160] 在图 3 中，

[0161] (a) 表示原始的子图像块数据；

[0162] (b) 表示按照数据嵌入步骤中计算得到的差值，数据流向“→”上面（或右边）的数字标号代表计算所用公式，下面（或左边）的数字标号代表隐秘数据信息；

[0163] (c) 代表产生直方图间隙后的块差值；

[0164] (d) 代表  $P = L = 2$  时，嵌入第一级数据“10101”后的块差值；

[0165] (e) 代表  $P = 1$  时，嵌入第二级数据“0101”后的块差值；

[0166] (f) 代表  $P = 0$  时，嵌入第三级数据“10”后的块差值；

[0167] (g) 代表隐秘图像块数据

[0168] 参见图 4，图 4 为本发明中涉及子图像块数据提取和图像恢复实例。

[0169] 在图 4 中，

[0170] (a) 代表隐秘图像块数据；

[0171] (b) 代表按照数据嵌入步骤中计算得到的差值，数据流向“→”上面（或右边）的数字标号代表计算所用公式，下面（或左边）的数字标号代表隐秘数据信息；

[0172] (c) 代表  $P = 0$  时，提取第三级数据“10”后的块差值；

[0173] (d) 代表  $P = 1$  时，提取第二级数据“0101”后的块差值；

[0174] (e) 代表  $P = L = 2$  时，提取第三级数据“10101”后的块差值；

[0175] (f) 代表直方图间隙还原后的块差值；

[0176] (g) 代表恢复出的图像块数据。

[0177] 参见图 5，图 5 为本发明方法在 5 种测试图像和嵌入阈值  $L = 2$  时的嵌入情况下的隐秘图像对比图。

[0178] 表 1

[0179]

|      | L=0      |      | L=1      |      | L=2      |      | L=3      |      |
|------|----------|------|----------|------|----------|------|----------|------|
| 测试图像 | Capacity | PSNR | Capacity | PSNR | Capacity | PSNR | Capacity | PSNR |

[0180]

|          |          |         |          |         |           |         |           |         |
|----------|----------|---------|----------|---------|-----------|---------|-----------|---------|
| Lena     | 25273bit | 45.99dB | 66498bit | 40.41dB | 97315bit  | 37.50dB | 118905bit | 35.60dB |
| Baboon   | 6314bit  | 46.47dB | 18747bit | 40.58dB | 30898bit  | 37.24dB | 42719bit  | 34.94dB |
| Boat     | 11245bit | 47.65dB | 33282bit | 41.89dB | 54303bit  | 38.74dB | 74030bit  | 36.63dB |
| Airplane | 29479bit | 46.78dB | 77109bit | 41.33dB | 108696bit | 38.52dB | 128794bit | 36.67dB |
| Pepper   | 15119bit | 46.88dB | 43211bit | 41.19dB | 68634bit  | 38.13dB | 90326bit  | 36.10dB |

[0181] 本发明的具体实施过程中, 一些小的步骤用到了其他文献公开发表的算法和专利中声明的方法, 如图像分块方法、图像块差值计算、LSB 嵌入和提取方法在此不再详细说明。

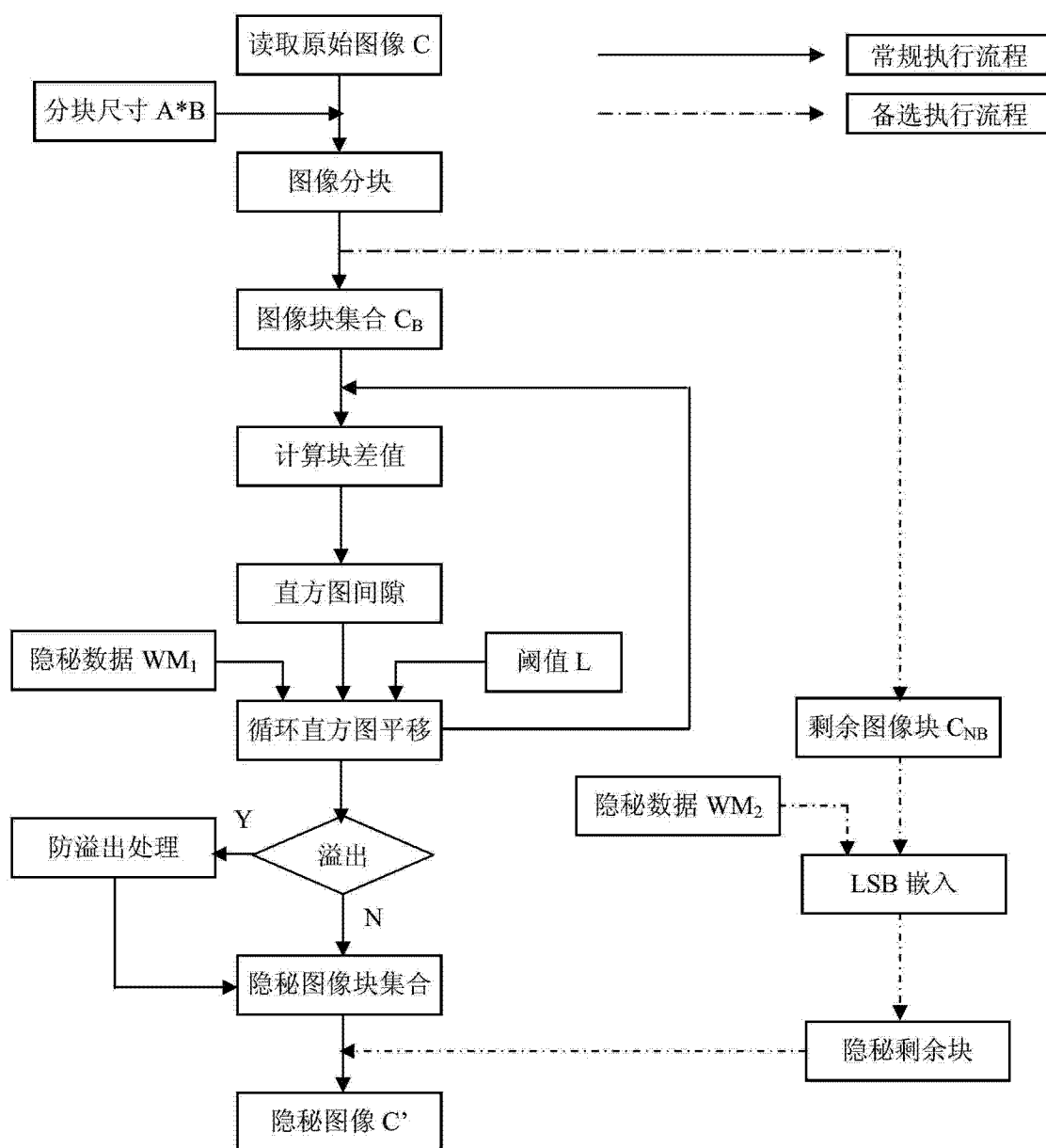


图 1

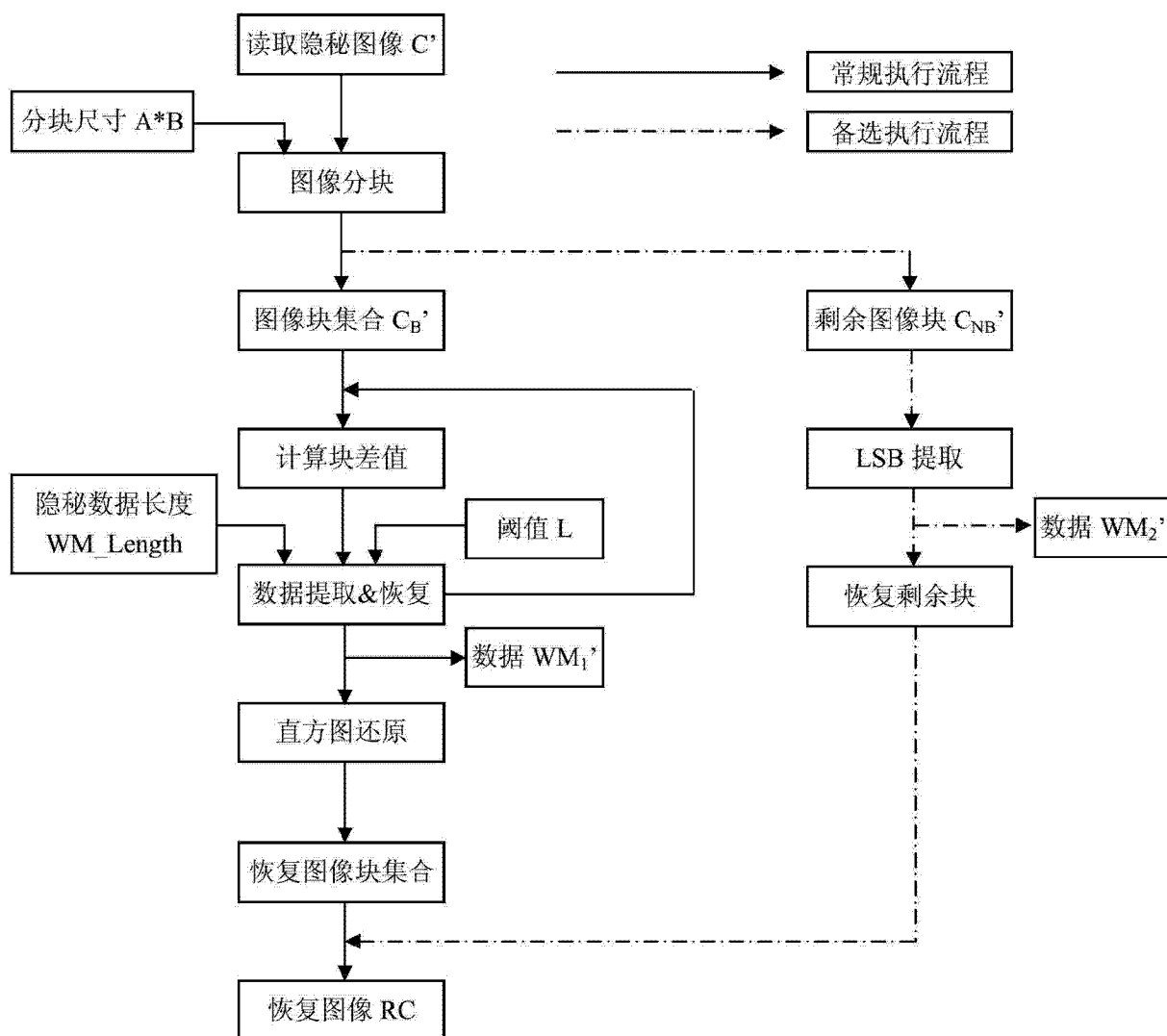


图 2

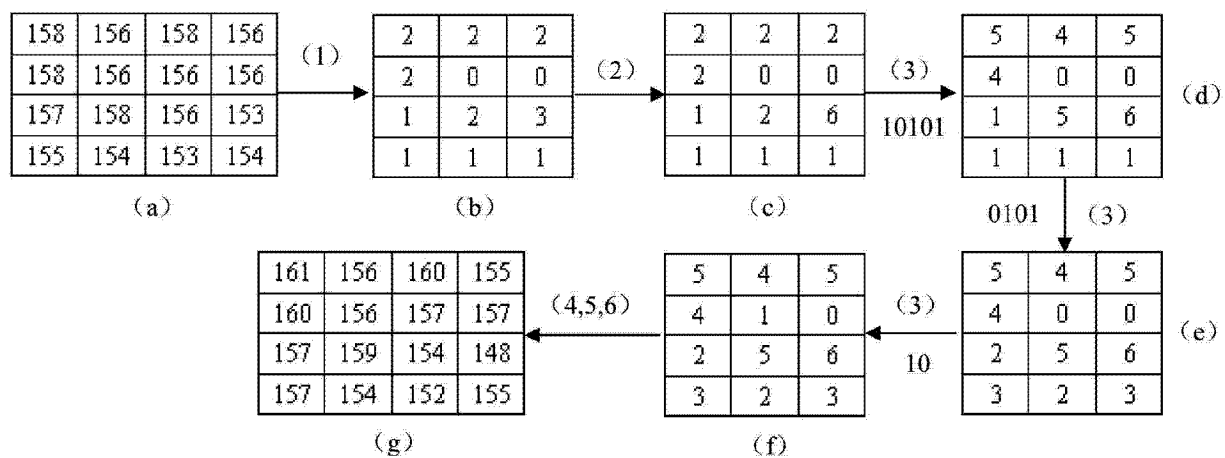


图 3

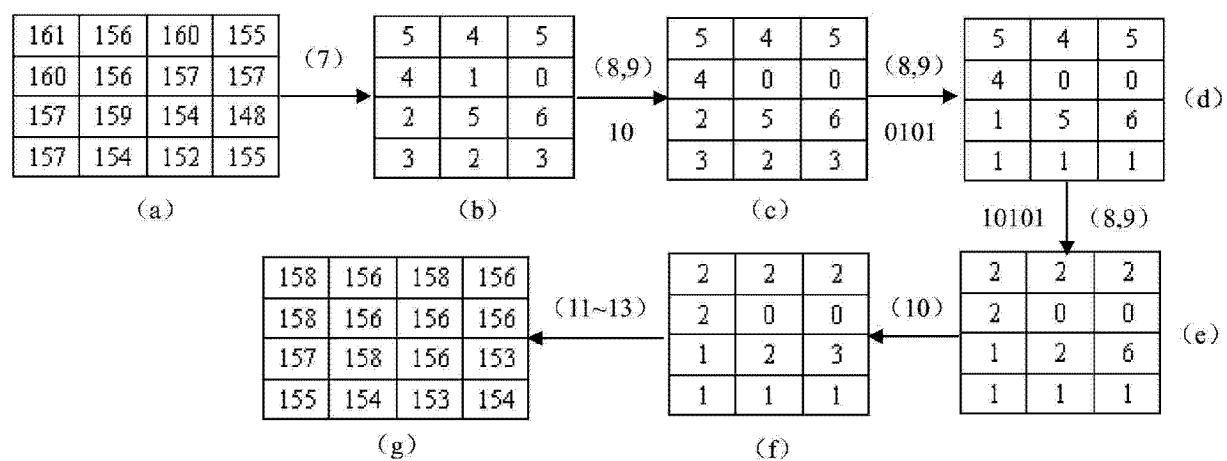
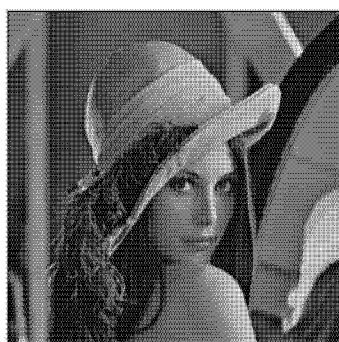


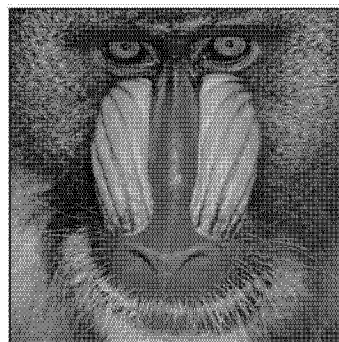
图 4



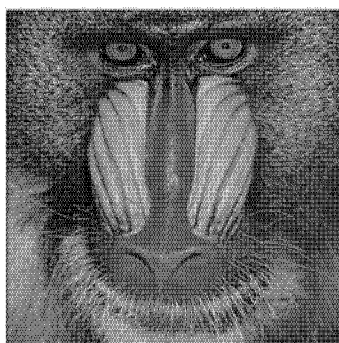
(a)



(b)



(c)



(d)



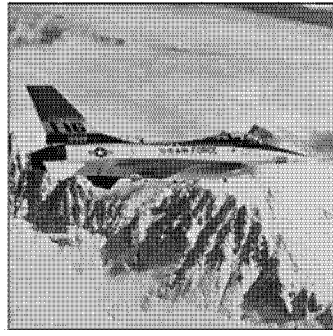
(e)



(f)



(g)



(h)



(i)



(j)

图 5