基于差分拓展的近似缩略图保持加密技术

一．摘要：

二．关键词：差分拓展，缩略图保持，图像加密

三．介绍

1.背景介绍

随着移动网络以及摄像头的飞速发展，再加上人民生活水平的不断提高，越来越多人利用电子设备记录生活的点点滴滴，并将其上传到第三方例如阿里、华为、谷歌等提供的云平台存储空间中。这些第三方平台在一方面为我们存储管理照片提供了便利，使我们的电子设备可以腾出更多的空间来拍摄新的照片，但从另一方面来说，却将我们的个人隐私置于危险之中。

目前，个人隐私保护也越来越得到国家和社会的关注，很多平台在数据存储方面都新增了加密功能，更好地保存了用户的隐私。但在在线图片预览方面却存在着隐私性与可用性的矛盾，隐私性要求我们要将图片加密存储，但可用性又要求我们需要为用户提供在线缩略图预览，以便用户管理自己的图片，若用户的每次预览访问都需要平台先解密图片再生成缩略图，这将为平台造成很大的计算负担。为了均衡隐私性和可用性，缩略图保持加密技术越来越重要。

2.已有方法

目前已有各种加密方式来达到此要求，这些方法基本都是以块为单位，独立地对每个块进行操作，使其加密后的块平均像素仍然和原图块平均像素近似甚至达到一样的效果。例如特殊的加密方法TPE【1】；利用对图像块内的像素进行伪随机洗牌来达到缩略图保持加密【2】，这种方法操作简单，且能将缩略图完全保持，但又暴露了平均像素甚至块内像素列表，且不好进行压缩。或者是通过动态区间保留加密【3】方法，只加密低LSB位，使加密后的块内像素仍然保持在一定范围内，从而使加密后平均像素与原平均像素相差无几以达到加密前后缩略图基本一致的效果，但此种方法存在解密失败的可能性。还有利用LSB嵌入【3】来进行调整的方法，将加密后的图像各像素高低位反转，通过调整高位值使得块平均像素基本与加密前保持基本一致以达到加密前后缩略图近似的效果，然而，由于改变了加密后的最低位，在解密时，像素的低位无法恢复，导致解密后的图像与原图不完全一致。如今，这方面的技术越来越多，但也都有优有劣。

3.我们的方法

在了解了图像信息嵌入的方法后，我们考虑将信息嵌入与图像加密相结合，以达到加密的特殊要求——缩略图保持。文献【4】介绍了在加密前进行腾空间，在加密后进行信息嵌入的方法，既保护了图像的隐私，又能满足信息嵌入效果。文献【5】介绍了基于差分扩展的信息嵌入，既简单方便，又有大容量的空间。我们的基于差分扩展的近似缩略图保持加密方法采用腾空间、加密、调整格式，以块为单位，块内分调整区以及嵌入区，在嵌入区利用差分扩展将调整区的MSB位嵌入保存，然后进行常规的图像加密，加密后通过调整调整区像素的MSB位以达到其缩略图与原图像缩略图基本一致的效果。

我们的方法没有使加密前后的缩略图完全一致，一方面保护了用户的隐私，对图像的细节部分模糊化，另一方面降低了计算复杂度以及腾空间强度。当然我们的近似化程度是可以通过变化MSB大小调整的，MSB越大，得到的缩略图与原缩略图越一致，但同时所需要的空间也越大。与LSB嵌入方法不一样的是，我们的方法是完全可逆的，恢复后得到的图像与原图像完全一致，是无损的。

四．差分拓展

差分拓展是两个像素为一组，利用两像素之间关系的冗余性来进行数据嵌入的技术。假设我们现在有两个像素值，现将一个比特嵌入其中。

首先，计算平均像素以及两个像素的差值：

符号表示不大于括号内的数值的最大整数，且可能为负。其次，我们将表示为二进制形式，然后我们将嵌入到的最低位（LSB）中。在数学上，等价于：

最后我们基于新的差值以及原来的平均值，计算新的像素值：

现在假设我们已知嵌入信息后的像素组，并从中获取嵌入的信息以及恢复原来的像素组值。首先仍计算平均像素以及像素差：

对于现在的，我们将其化为二进制模式，将最低位的值作为嵌入的信息进行提取，并将差值还原为。在数学上，等价于

最后我们基于平均像素以及恢复后的差值还原两个像素：

由上面的式子可以看出，想要对像素组进行差分扩展，必须要使得差分扩展后的像素组仍旧处于像素变化范围内，即0~255区间内，由此可得范围：

其中。对上述式子进行化简得到关于的范围：

我们称满足上述不等式的像素组为可拓展组，若像素组不满足以上不等式，但满足不等式

的，我们称其为可变化组。对于可变化组，其。

我们利用差分扩展的简便性和完全可逆性来将图像恢复所需要的数据事先进行嵌入，使得我们的加密过程也是完全可逆的。

五．基于差分扩展的缩略图保持加密技术

1.腾空间

该步骤主要为后面的第三步调整服务， 通过事先腾出空间存储调整时可能会改变的值从而保证在调整后的图像仍旧是可完全恢复的。

*以块为单位*：首先我们先对块内进行处理，将块分为调整区与嵌入区两部分，具体分割法如图1。

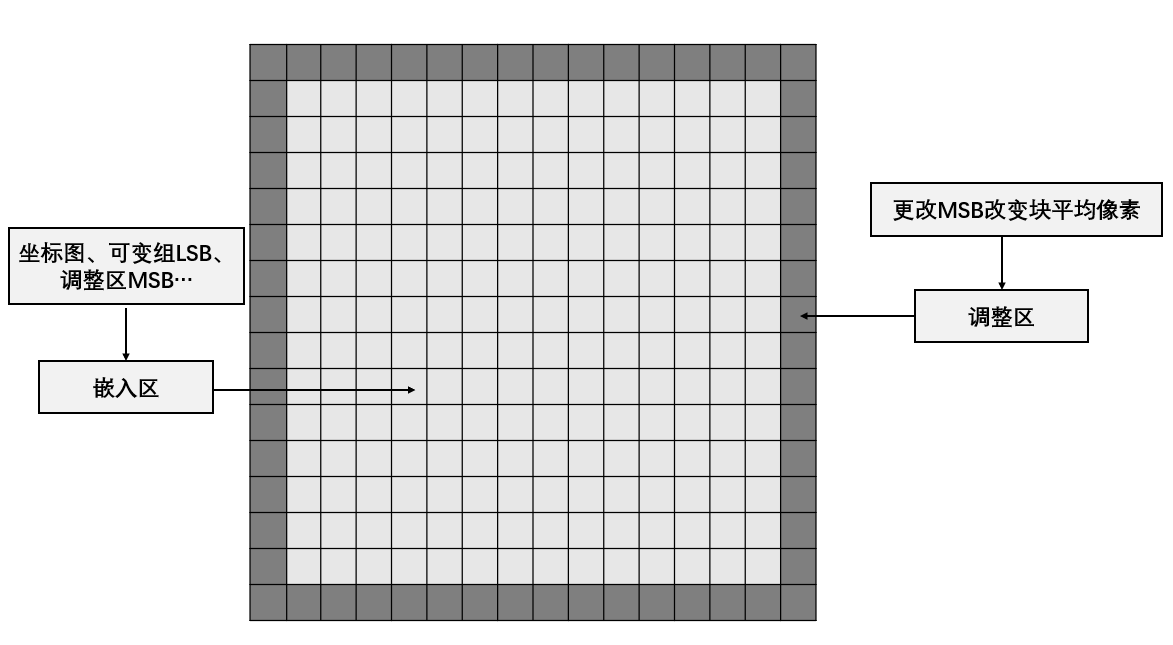


图1.块内功能区划分情况

记录调整区各像素的MSB部分的比特值为集合。然后我们将嵌入区的像素进行划分，每两个像素为一组，根据章节四中提到的不等式将像素组划分为4组，EZ，EN，CN和NC：

1. EN: 包括所有可拓展组，并且不与集合EZ有交集.
2. CN：包含所有可变化组，并且不与集合EZ及EN有交集.
3. NC：不属于上述任一集合的像素组.

根据分组情况，我们将生成坐标图用以记录集合EZ以及EN中像素组所在位置（这里采用的是高度与嵌入区高度一致，宽度为嵌入区宽度一半的图来记录，若像素组属于集合EZ或EN，则坐标图对应位置值为1，否则值为0），并且记录集合CN中像素组差值的最低位（LSB）为集合(其中若h=1或h=-2，则不需要进行记录)。最后我们将待嵌入数据进行整合形成一个比特流

其中，是比特流data的比特长度。由于在一个块内像素的高位往往一致，所以此处可以对集合bits进行压缩后在进行拼接（若MSB>1，此处可以将每个像素的同一位一起存储，例如MSB=2时，可将每个像素的最高位收集在一起，将次高位收集在一起存在所有最高位之后，然后再进行压缩），且经实验可知，集合locate-map中1出现概率极高，故也可以先行进行压缩，这样拼接后的比特流data长度大大简短。同时我们将对各集合中的像素组进行预处理，便于后面的信息嵌入，伪代码如下：

*以图像整体为单位*：我们将每个块根据上述步骤得到的待嵌入比特流data进行整合，其中块与块数据之前需插入分隔符以便区分。整合完毕后我们对整幅图像的各个嵌入区进行操作，首先将像素组分为可变组CH和不可变组NC两组：

1. EN: 不满足上面不等式的像素组

然后对于集合CH中的每个像素组进行信息嵌入：

其中i为待嵌入比特流下标。至此腾空间部分便完成。这当中涉及到的分组等情况详情参见文献【5】。由于我们新增了调整区，此处我们相较于文献【5】进行了改进，不再将块产生的待嵌入信息直接嵌入到当前块中，这样有效地避免了某些块需嵌入信息过多容量不够而某些块空间有大量冗余的不均衡情况。

2.加密

该方法中的加密部分并无严格要求，只需要加密后的像素值仍旧在0~255范围内即可，再投入实际应用时可以根据安全等级需求采用相对应的加密方法。这里我们直接用的最简单的比特流加密。

3.调整

该过程以块为单位，通过调整调整区的各像素的高MSB位来使得所处块的平均像素值尽可能接近原图像对应块的平均像素值，以达到近似缩略图保持的效果。MSB取的值越大，调整后图像的缩略图就越接近于原缩略图，但是相对应的，用于存储覆盖的MSB位比特所需的空间就越大，即对腾空间过程要求就更高。

4.恢复

首先我们先对图像进行解密。然后我们正式开始图像恢复工作。

*以图像整体为单位*：与腾空间部分的整体处理过程相似，我们也先将嵌入区的像素组分为两个集合：可变组CH以及不可变组NC，然后我们将嵌入在可变组集合中各像素组的数据进行提取：

将所有嵌入数据提取出来之后，我们便可通过之前插入的分隔符分割出各子块的嵌入信息。

*以块为单位*：在得到子块的嵌入信息后，我们便可以将调整区被覆盖各像素的高MSB位还原。其次，通过还原坐标图locate-map，我们可知晓可变区CH中哪一部分是原来的可拓展组，哪一部分是原来的可变化组，从而进一步还原嵌入区的各像素，该过程的伪代码如下：

经过上述四大步骤便可实现近似缩略图保持加密以及无损恢复，同时，若MSB选取较小，还能有多余空间用于其他信息如图片的描述等的嵌入。

六．实验结果

这里我们使用大小的“Lena”的灰度图作为实验样本，表1显示的是块大小、MSB大小与待嵌入比特量、可嵌入负载量之间的关系。