基于改进零树编码的 ROI 图像联合压缩加密算法①

翟羽佳, 邓家先

(海南大学 信息科学技术学院, 海口 570228)

摘 要: 提出了一种能灵活控制感兴趣区域(Region Of Interest, R0I)的重建质量且将图像的压缩与加密同步实现的编码算法. 该算法将图像进行ROI划分后利用加权因子灵活控制其编码量, 再通过结合改进零数的压缩算法与算术编码, 使用密钥对压缩产生的原始上下文与原始判决进行修正, 从而实现 ROI 的联合压缩加密. 实验结果表明, 当密钥正确时, ROI的重建质量随着加权因子的变化而变化;当密钥出错时, ROI重建质量急剧下降, 图像实现分区域加密, 从而验证了本文所提出算法的可行性.

关键词: 图像压缩; ROI 编码; 改进零树; 联合压缩加密

Algorithm on Joint Compression and Encryption of Image's Region of Interest Based on Improving Zero-Tree Coding

ZHAI Yu-Jia, DENG Jia-Xian

(Institute of Telecommunication Engineering, Hainan University, Haikou 570228, China)

Abstract: In this paper, we propose a coding algorithm which could flexibly control the reconstruction quality of image's region of interest(ROI) and synchronously implement image compression and encryption. The algorithm uses the weighted factor to control the amount of coding after dividing ROI of image. Through combining improving zero-tree compression algorithm and arithmetical coding, the original context and original sentence from compression is amended by taking advantage of the key to realize joint compression and encryption of ROI. The results indicate that reconstruction quality of ROI changes with the change of weighted factor when the key is right. On the contrary, reconstruction quality of ROI reduces dramatically and image conducts regional encryption when the key is fault, which proves the algorithm we put forward is feasible.

Key words: image compression; ROI encode; improved EZW; joint compression and encryption

在信息量庞大和信息安全越来越受到重视的今天, 信息媒质的压缩和信息的安全通信都是不得不面临的问题. 目前, 图像压缩与加密往往分为两个独立过程[1]. 这种加密过程增加了系统开发、测试和使用成本. 在流媒体图像数据传输过程中, 为了保证解码同时兼顾数据安全, 往往只对头文件数据进行加密, 其数据安全性得不到充分保证. 所以近几年出现了一种新趋势, 即将加密算法融入到图像的压缩算法中[2-3], 在编码过程就进行加密处理, 可以不影响解码速度的同时又提高数据的安全性, 这种方法称之为图像联合压缩加密[4](Joint Compression Encryption). 而通常图像压缩和加密所使用的编码器是不同, 图像压缩中所广泛使用的是自适应编码器, 这种编码器更适合图像数据的压缩且相对复杂;而算术加密使用的又是较简单的算术编码模型和算法来进行研究的. 且现阶段的联合压缩加密算法主要是利用区间分裂算术编码器来实现, 它采用密钥对原始算术编码器的概率区间进行重新划分, 从而影响编码码字的概率分布, 使输出的压缩码流产生变化, 达到压缩加密的效果. 但由于区间分裂算法是基于传统的算术编码器, 所以它的

① 基金项目:海南省自然科学基金(613155)

收稿时间:2013-11-15;收到修改稿时间:2014-01-03

编码效率比较低，且不能满足自适应编码的要求，即这种基于区间分裂的算术编码器就不能满足现代压缩算法的要求. 所以本文提出了基于改进零数算法和引进自适应算术编码器修正的联合压缩加密算法. 该算法是在零数编码的基础上引入比特平面编码，并将产生的系数送入算术编码器中进行自适应的算术编码，实现对图像进行熵编码的同时进行数据加密.

而在实际应用中，现在人们往往更多关心的是图像中某部分区域,我们称为感兴趣区域，例如远程医疗图像中的诊域，干涉多光谱图像中的包含谱信息的部分区域等等，而对图像中的背景则并不关心. 因此，感兴趣区域编码就是以牺牲一定背景区域(Background,BG)的解码质量为代价，保证感兴趣区域可以比背景区域得到更好的重建质量. 它能满足一定比特率下，重要图像信息的高质量重构. 所以本文在进行编码之前，先对图像进行 ROI 预处理,将其拆分成 ROI 图像及 BG 图像，在实现单独编解码的同时也实现了感兴趣区域和背景区域重构质量的灵活调整，再对零树编码进行改进，引入自适应算术编码，并利用给定密钥对原始上下文和判决进行修正，从而实现感兴趣区域联合压缩加密，或可以对感兴趣区域和背景区域使用不同的密钥，由于所使用的算术编码是自适应的，相对区间分裂的算术编码而言，所提出的算术加密算法的密文安全性更好，且与原算法有相当的压缩效率.

1 ROI 图像编码

通常，研究的感兴趣图像编码大多都是按比例增大感兴趣区域的小波系数，使感兴趣区域的小波系数能够位于较高的比特平面上. 在接下来进行的嵌入式编码的比特流里，这些处于较高平面上的系数被放在非感兴趣区域的小波系数前面. 这样感兴趣区域就会优先于非感兴趣区域进行编码和细化. 但是这样会使感兴趣区域与背景区域的相对质量难以调整，导致其重构质量效果较差，而且这些方法需要对图像感兴趣区域的形状信息进行编解码，增加了算法的复杂度. 这些 ROI 图像编码都是以严重牺牲图像的背景区域的质量为代价的,而并行编码解决了这种冲突. 如果仅仅直接将原始图像拆分成 BG 图像和 ROI 图像单独进行编解码， 就会产生两个信道，而编码量就会变成原来的两倍，即使图像质量好，也是以牺牲信道编码量为代价的. 且传统的算法没有考虑到人眼的视觉特性,也没有考虑感兴趣区域的编码优先性，不论是感兴趣区域还是背景区域均采用同样的编码方案. 如果对整幅图像采取同等级别的编码，则会在低比特率编码下得到整体重构质量都较差的图像. 所以文中采用了一种通过设置加权因子 α，来灵活分配 ROI 区域与 BG 区域的编码量，这样就可以满足在总编码不变的情况下，不仅可以得到理想的 ROI 区域，对于背景区域质量也不会受到较差影响. 解码之后，再对两幅重构图像进行简单像素值的叠加，就可以恢复原图像. 同时在感兴趣编码的实现过程中，本文算法无需对 ROI 形状进行编解码，降低了运算的复杂度. 流程图 1 所示.

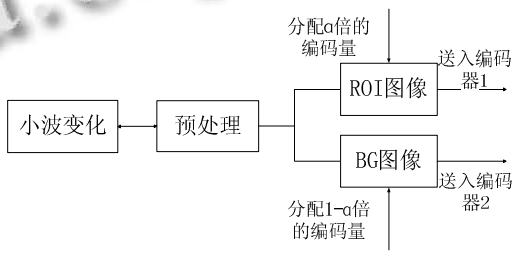

图 1 ROI 图像预处理流程图

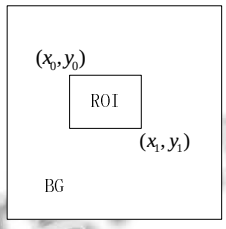

图 2 感兴趣区域坐标

首先在进行图像编码之前，仅先对原始图像进行小波变换，然后进行预处理来实现感兴趣区域划分，即确定在小波域中各子带对应的 ROI 区域，将原图像拆分成 ROI 图像和 BG 图像. 两幅图像的大小与原图像保持一致，在 ROI 图像中的非 ROI 区域的像素值均为 0，而对于 BG 图像中的正相反，既 ROI 区域的像素值均为 0. 本文采用的是 3 次小波变换，所以在模板生成之前，要先将 ROI 区域适当的扩大，使其边缘是 2^3 倍数. 然后寻找其模板. 设空间域的 ROI 区域的左上角和右下角的坐标分别是 (x_0,y_0) 和 (x_1,y_1)，如图 2 所示，则易知其小波变换后的重要系数 (LL_3) 的左上角和右下角的坐标为 $(x_0/2^3,y_0/2^3)$ 和 $(x_1/2^3,y_1/2^3)$，然后根据小波变换后确定其子孙节点的方法，来确定其 ROI 模板. 其他高频子带中所对应的 ROI 区域由公式

(1)确定, 高频子带中所对应的 ROI 区域和低频子带的 ROI 区域构成零树结构[5].

$$\begin{cases} x_1' = \lfloor x_1/2^3 \rfloor, y_1' = \lfloor y_1/2^3 \rfloor, \\ x_2' = \lceil x_2/2^3 \rceil, y_2' = \lceil y_2/2^3 \rceil, \end{cases} \quad (1)$$

其中3为小波变换级数;$\lfloor a \rfloor$表示下取整运算, 既不大于 a 的最大整数;$\lceil a \rceil$ 为上取整运算, 即不小于 a 的最小整数.

通过预处理获得 ROI 图像, 图像中的背景部分的像素值均已设为0, 再进行编码时, 由于ROI区域像素的小波系数幅值相对 BG 区域较大, 则 ROI 区域图像的重构质量就会更好. 编码加权因子α, 该参数反应了对 ROI 图像编码的优先权. 相应的 BG 图像的编码优先系数为$1-\alpha$. 通过调节编码加权因子, 可以调整 ROI 与 BG 图像重构图像的相对质量. 本文用较多的比特(α倍的总编码量) 对 ROI 图像进行编码, 用较少的比特$(1-\alpha)$倍的总编码量)对 BG 图像进行编码. 设原始图像大小为$H \times W$,ROI 大小为$H_1 \times W_1$(其中$H>0$、$W>0$、$H>H_1>0$、$W>W_1>0$), 压缩倍数为c, ROI 占原始图像的比例为β ($\beta=(H_1 \times W_1)/(H \times W)$), 总的输出码流长度为 Len. ROI 的输出码长为$Len_1 = Len \times \alpha$, BG 的输出码长为$Len_2 = Len \times (1-\alpha)$, 则有如下关系:

$$\begin{aligned} Len_1 &= Len \times \alpha = (H \times W/c) \times \alpha \\ &= (H_1 \times W_1/c) \times (\alpha/\beta) \\ &= (H_1 \times W_1)/(\beta c/\alpha) \end{aligned} \quad (2)$$

则 ROI 压缩倍数:

$$c_1 = \beta c/\alpha \quad (3)$$

当满足$1>\alpha>\beta$时, $c_1=\beta c/\alpha<c$, 即 ROI 的压缩倍数c_1小于总的图像压缩倍数c, 可以保证ROI的传输质量好于 BG. EZW 解码后将两幅图像进行简单的系数相加, 便可恢复小波域系数, 再进行小波逆变换, 最终得到解码图像. 既避免了对小波系数的提升, 又避免了对 ROI 区域形状的编码. 这样的处理方法, 即使是双信道传输, 总的编码量也没有增加. 并且可以让 ROI 区域的图像获得优先编码, 得到较背景区域更好的重构质量.

实验中验证了该运算的正确性. 随着因子α的增大, 图像 ROI 区域的重构质量也越来越好, 相应的是 BG 区域的质量越来越差. 与传统的算法相比, 本文算法不仅能够保留无需单独对感兴趣区域进行编解码的优点, 而且还能够调节背景区域与感兴趣区域的相对重构的质量, 在灵活性上有较好的改进.

2 基于零数的 ROI 图像联合压缩加密

在进行外预处理后, 原始图像将被拆分成两幅图像送入两个编码器进行编码, 在编码的同时进行数据加密. 本文在零树编码的基础上进行改进, 使得图像不仅可以设定感兴趣区域进行编码, 还可以对 ROI 进行单独加密. 当密钥出错时, 感兴趣区域质量会严重下降且背景图像则不受影响.

零树编码是由 Shapiro 提出的一种基于小波变换和比特平面编码方法[6], 利用小波子带系数之间的相似性进行高效的数据压缩, 是最早基于小波变换的图像压缩算法之一. 以三级小波变换为例, 图像经 3 级小波变换后, 按其频带从低到高形成一个树状结构. 树根位于最低频子带内, 它有 3 个子女, 分别位于次低频子带的相应位置;最高频子带除外, 其余子带的结点都有 4 个子女, 位于更高一级子带的相应位置. 这样图像经过 3 级小波变换后形成了深度为 4 的树. LL3 子带的一个系数对应LH3、HL3、HH3三级子带系数, 除了LL3子带有3个子节点之外, 其他每个子带系数对应 4 个子节点(LH1, HL1, HH1 子带中的节点没有子节点). 从而构成一棵树, 称之为零树. 零树编码按照比特平面的先后顺序进行排序, 在同样的比特平面内按照 LL3 子带系数的顺序对相应的树进行编码, 这种逐次逼近的编码算法能够取得好的编码效率.

零数编码算法本质上是一种变换编码, 其小波分解后信号的低频部分对应于原信号的一个平滑版本, 而高频部分对应于这两个信号的差别信息. 一幅图像经过若干级小波分解后, 在不同子带的相同位置的变换系数应该是相关的, 这种相关性形成了零树结构. 即一幅经过小波变换的图像按其频带从低到高形成一个树状结构. 树根是最低频子带的节点. 它有 3 个孩子, 分别位于 3 个次低频子带的相应位置, 其余子带(最高频子带除外)的节点都有4个孩子, 位于高一级子带的相应位置(由于高频子带分辨率增加, 所以一个低频子带结点对应 4 个高频子带节点). 这样一个 N 级小波分解就形成了深度为 N+1 的树, 以 3 级小波变化为例, 如图 3 所示.

研究表明, 在图像的低比特率编码中, 用来表示非零系数位置的开销远远大于用来表示非零系数数值的开销. 零树结构正是一种描述图像经过小波变换后

非零系数位置的有效方法. 在零树中共有 4 种符号, 零树根(ZTR)、孤立零点(IZ)、正值(POS)&和负值(NEG). 其中 ZTR 表示自己的值为 0, 并且其所有子孙的值也均为 0;IZ 表示自己的值为 0, 但有非零的子孙结点;POS 和 NEG 分别表示自己的值为正的或负的. 零数的编码思想是不断扫描变换后的小波系数, 生成多棵零树来对应图像编码. 一棵零树的形成需要对图像进行两次扫描. 在生成第一棵零树时, 首先找出变换后图像的最大绝对值系数 L, 则初始阈值为 $T_1 = 2^{\log_2 \lfloor L \rfloor}$, "$\lfloor \rfloor$"是向下取整. 对图像进行第一次扫描, 把变换图像中绝对值小于阈值的系数都看作 0, 然后按前面的符号定义形成零树. 在第二次扫描中, 对那些绝对值大于阈值的重要节点(POS 和 NEG)按其绝对值是否超过阈值的 1.5 倍附加一个比特 1 或 0 来描述其精度, 这样做的目的是减小非零节点系数值的变化范围, 使其适应下一次阈值减半后的比特附加. 而后将阈值减半, 再经两次扫描生成第二棵零树, 在第一次扫描生成零树时, 以前已经大于阈值的结点不再考虑, 而第二次扫描附加比特时则要考虑以前数值较大的结点以保证精度. 这个过程被称为连续近似量化(successive approximation quantization,SAQ). 如此往复下去, 不断生成零树, 直至压缩文件中的码流数据能够完全无损地还原出原图像为止. 零树编码按照比特平面的先后顺序进行排序, 在同样的比特平面内按照 LL3 子带系数的顺序对相应的树进行编码, 这种逐次逼近的编码算法能够取得好的编码效率.

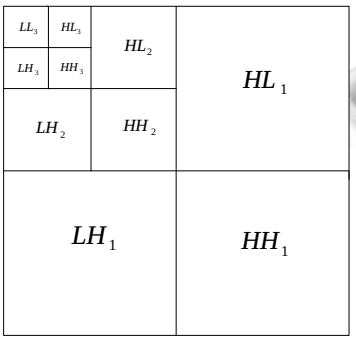

图 3 3 级小波分解

为了利用算术编码对数据进行进一步的压缩, 对零树编码进行改进, 改进后的编码器结构如图 4 所示. 本文中是对 ROI 图像和 BG 图像并行进行编解码, 这里以 ROI 图像为例. 在同一比特平面内, 按照图 2 中的子带顺序进行编码, 编码后产生的上下文和判决一起送往算术编码器进行下一步压缩. 同理, 将 HL3、HH3 系数及其子孙构成各自集合, 2 级子带的系数及其子孙构成各自子集合送入编码器. 例如 HL3 分为 HL2_1、HL2_2、HL2_3、HL2_4 共四个集合, 三级子带系数及其集合一起编码, 形成码流送入同一算术编码器. 对于 BG 图像, 其原理与 ROI 图像相同, 在此就不再进行重复.

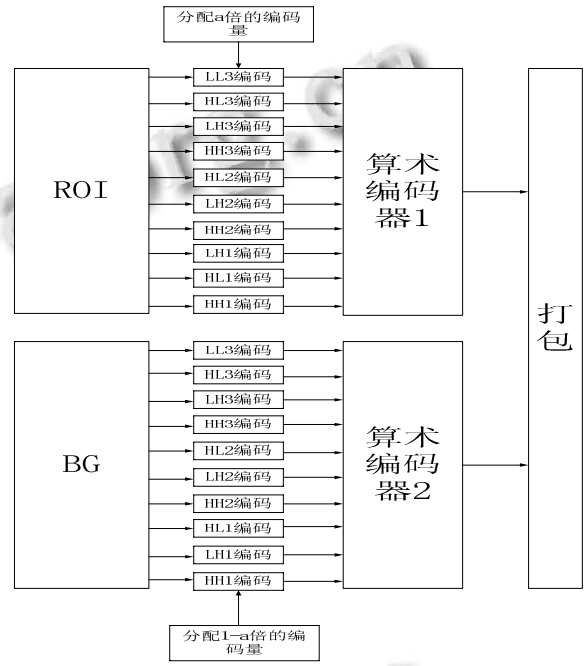

图 4 编码器结构

集合上下文是根据相邻集合的重要性产生, 经过合并形成 4 种集合上下文. 图 5 为集合相邻关系, X 表示当前集合重要性, E0, E1, E2, E3 表示对角邻居集合重要性, H0, H1 表示水平方向邻居集合重要性, V0, V1 表示垂直方向邻居, 其中 0 表示该集合不重要, 1 表示该集合重要. 集合上下文 CXS 为

$$CX_S = (V0|V1|H0|H1)\times 2 + (E0|E1|E2|E3) \quad (4)$$

E0	V0	E1
H0	X	H1
E2	V1	E3

图 5 集合邻居

其中|表示逻辑或运算, 显示, 当所有邻居集合都不重要时, CXS=0;水平、垂直都不重要, 且对角集合至少有一个重要, 则 CXS=1;水平、垂直集合至少有一个

重要, 且对角集合都不重要, 则 CXS=2;水平、垂直集合有一个重要, 且对角也至少一个重要, 则 CXS=3. 与 EBCOT 算法一样, 系数上下文进一步细划分为零编码上下文、幅值细化上下文、符号编码上下文, 其上下文计算采用 EBCOT 中的方法[7,8], 这里不再赘述.

3 ROI 图像联合压缩加密

算术编码器是将给定序列映射为一个概率子区间, 编码器输出的码字就是对应概率子区间的一种描述. 对于简单的基于区间分裂的算术编码而言, 输入的二进制判决 0、1 的概率是固定的, 通过改变判决就可以实现算术加密. 而对于自适应算术编码器而言, 其输入的包括上下文和判决两部分, 不同上下文对应判决的初始分布不完全相同, 而且后续输入判决的条件概率分布也不完全相同. 对于给定的序列, 如果上下文不同, 对应的概率子空间也不相同, 编码输出的码字也不相同. 如果改变给定序列中的任何一个上下文或者判决, 就会导致概率子空间的不同, 并会对后续判决的条件概率分布产生影响.

自适应算术编码不仅能够有效提高编码效率, 同时也可以用来进行算术加密. 利用同一算术编码器实现数据压缩和数据加密, 这种方法称为联合压缩加密. 由于自适应算术编码器需要使用上下文和判决, 从理论上讲, 使用密钥对上下文或者判决进行修正都可以实现联合压缩加密. 本文对两种加密分别进行讨论.

基于判决修正的算术加密原理如下.

设 key 表示加密密钥, $D=(d_1,d_2,\ldots d_N)$ 表示编码产生的长度为 N 的二进制判决矢量, 定义一种运算

$$D_1 = f(D,key) \tag{5}$$

其中 $D=(d_{11},d_{12},\ldots d_{1N})$ 也是长度为 N 的矢量, 且其中每个元素仍然是二进制.

利用密钥对比特平面产生的二进制进行判决运算, 使得修正后的部分二进制判决与原来的二进制判决不同, 如果系统解码使用的密钥与编码使用密钥不同, 则会出现解码错误, 即图像出错. 从而实现数据加密.

设 key1 表示解密密钥, $\hat{D}=(\hat{d}_1,\hat{d}_2,\ldots\hat{d}_N)$ 表示解码后的判决矢量, 当 $key_1=key$ 时, 则 $\hat{D}=D$. 也就是说, 如果解密时密钥正确, 则应当正确的重建原始序列. 这也就说公式(5)的定义必须满足可逆:

$$\hat{D}=f^{-1}(D_1,key)=f^{-1}(f(D,key),key)=D \tag{6}$$

其中 f^{-1} 为公式(4)的逆运算.

基于上下文修正的加密原理如下.

设 key 表示加密密钥, CX 表示编码产生的原始上下文, 对应取值范围为 $(m,m+1,\ldots,m+L)$, 修正后上下文为 CX1, 对应取值范围为 $(m,m+1,\ldots,m+L')$, 上下文修正可以定义为

$$CX_1=g(CX,key,n) \tag{7}$$

其中 n 表示该类上下文出现的顺序, L, L' 分别表示原始上下文喝修正后上下文的种类. 如果系统加密、解密产生的原始上下文相同, 且加密、解密密钥相同, 使用相同的运算, 那么系统解密上下文 CX1 就与加密的上下文 CX1 相同. 反之, 如果解密使用的密钥与加密所使用的密钥不同, 就会造成解密使用的上下文与加密时不同, 从而造成数据分类错误, 解密出来的数据与原始数据出现差异, 出现解密错误;进一步, 一旦出现解密的数据错误, 一方面会导致后续数据产生原始上下文就有可能出错, 从而产生连锁反应, 出现连续解密错误;

自适应算术编码使用的各类上下文和对应判决共同确定了算术编码的概率跳转规律. 如果某类上下文的如果某类上下文的初始该类分布相同(大部分是如此), 输送到算术解码器的上下文与编码使用的上下文是同类上下文之间的一一映射, 则算术解码的结果是正确的, 即不能实现算术加密. 自适应算术编码器的这类特点决定了式(7)不能是一种 key 和 CX 的线性运算. 比如, 式(7)将所有编码产生的类别为 5 的上下文都映射为 3, 且 3,5 是同一类上下文, 编码初始概率分布相同, 则这类运算不能实现加密.

自适应算术编码使用的各类上下文范围一定, 比如某类上下文范围为 $(m,m+1,\ldots,m+L')$, 式(7)运算的结果应当在该范围内, 即满足

如果 $m\le CX\le m+L$, 则.

如果超出自适应算术编码使用的某类上下文范围, 则进入其他类别的上下文范围, 不同类别的上下文所对应的初始概率分布不同, 条件概率的跳转规律也不相同, 进行联合压缩加密时, 可能会导致重建图像质量下降.

基于上下文、判决修正的联合压缩加密原理如图

6 所示. 原始图像数据经过变换, 将系数的相关冗余映射为系数的统计冗余;变换后的系数进行比特平面编码, 产生原始上下文 CX 和原始判决 D, 密钥 key 对原始上下文和原始判决进行修正, 产生修正上下文 CX1 和修正判决 D1, 并送往算术编码器.

本论文中, 判决修正采用简单的异或, 对密钥 key 进行循环移位得到密钥 key1, 即首先利用 key1 的最低位与原始判决进行异或运算, 然后密钥循环移位一次, 供下一次判决修正使用.

上下文修正算法是, 对密钥 key 进行循环移位, 取移位后的最低若干位二进制数据 dk 与原始上下文进行运算, 对于范围为 $(m,m+1,\ldots,m+L')$ 原始上下文的计算

$$CX_1 = m + (d_k + CX) \bmod (L_m) \tag{8}$$

其中 mod() 表示模运算. 修正后的上下文范围为 $(m,m+1,...,m+L_m)$. 这种上下文修正方法可以满足上文提出的上下文修正规则.

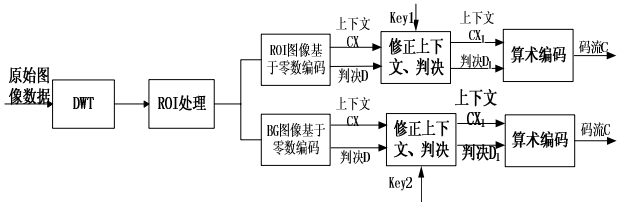

图 6 联合压缩加密原理框图

4 实验结果与分析

本算法采用 C 语言进行编程, 使用标准 lena 图像(大小为 512×512, 每个像素为 8bit)进行测试. 在编码之前, 图像经过三级小波分解. 图 7 为当加权因子 α 取 0.75 且输出码率分别为 0.3, 0.5, 1.0 时得到的图像. 图 8 为当码率相同加权因子取值不同时的重建图像, 码率为 0.5, 加权因子分别为 0.75, 0.8, 0.9.

图 7 在相同加权因子不同码率下的重建图像

图 8 在相同码率不同加权因子下的重建图像

以上图片均没有打开联合压缩加密算法，只是基于感兴趣区域而进行的编解码. 可以看出在加权因子 α 相同时，码率越高，整体图像重建质量都有提高. 而在码率相同时，加权因子 α 越大，ROI 区域重构质量越高，相对的BG区域重建质量降低. 可以看出本文提出的新算法(可以先不考虑联合加密，将在下文会进行具体详述)，通过加权因子 α 分配编码量，使 ROI 区域和 BG 区域的相对质量得到灵活的调整.

下面再接着讨论本文所提及的联合压缩加密. 表 1 为联合压缩加密重建质量(分别为仅打开上下文，仅打开判决修正，上下文判决同时打开)与不加密仅对感兴趣区域独立编码时算法重建质量(PSNR)的比较.

表 1　感兴趣区域编码质量与联合压缩加密重建图像质量(单位 dB)比较

码率	1.0	1.5	1.75	2.0
上下文修正	11.21	11.00	10.87	10.82
判决修正	26.02	25.03	24.38	24.15
上下文、判决修正	11.72	11.42	11.22	11.06

可以看出，只要参数选择合理，联合压缩加密与不加密算法具有同样的效果. 甚至有些加密可以有更好的重建效果.

当使用联合压缩加密算法密钥出错时，解码处理方法是使用错误密钥进行解码. 重建质量随码率和加密条件的变化如表 2 所示. 其中加权因子 α 为 0.8.

表 2　密钥错误时的重建质量

码率	原始压缩算法	上下文修正	判决修正	上下文、判决修正
0.5	31.90	32.07	31.53	31.58
0.75	34.32	34.39	34.05	34.12
1.0	35.44	35.55	35.06	35.10
1.5	37.51	37.54	37.37	37.38
1.75	38.00	38.04	37.86	37.86
2.0	38.65	38.69	38.42	38.43

从表中可以看出一些特点，当无密钥出错时，重建图像质量随码率的增加而增加;一旦密钥出错，重建图像质量会随码率的增加而降低. 这是因为密钥出错对应子带以及更低子带重建系数产生错误更加严重，经过小波变换，错误系数引起的图像数据错误更加严重，从而导致重建图像质量随着码率的增加而有所降低，加密算法更安全. 图 9 为密钥出错时与密钥正确时 ROI 重建图像的比较.

图 9　密钥出错与正确时的重建图像

以上数据我们可以得出，该算法与原始算法在秘钥正确时所得出的 ROI 图像重建质量无论在主观与客观上并无太大差别，在某些特定码率时，甚至高于原始算法的重建质量，从而验证了本文所提出算法的可行性. 而当秘钥错误时，所得重建质量会因码率的增加而降低，其安全性更高.

5　结论和展望

文中采用了改进零树编码算法来处理图像中感兴趣区域. 可以减少传输时间，解决其数据量大、耗时长的瓶颈问题. 在数据量大的数字图像的传输和存储中，在有效利用带宽或节约存储空间的意义上，有损压缩是更期望被采用的. 同时，使用联合压缩加密可以使感兴趣图像单独进行加密. 通过适当参数选择，可以保证联合压缩加密的重建图像质量相对新压缩算法的质量基本不变. 且由于采用自适应算术编码，相对区间分裂算术加密，其概率分布的复杂度更高，因此密码安全性更高.

但本文仍有不足的地方需要进一步的研究. 例如在划分感兴趣区域时，采用的只是简单的手动划分. 在下一步的研究中也可以将在 ROI 编码中融入性能更好的 ROI 区域自动分割方法. 同时论文采用的改进零树进行编码，进行小波变换后，对不同级子带进行编码，下一步加密时可以对每一级子带进行单独加密，即不同分辨率可使用不同密钥，这样可以给不同权限用户提供不同的重建图像质量. 此外，本文在加密过

程中的算术加密所采用的只是简单的异或算法，以后可以加以更复杂的加密算法，使得密文安全性更好.

参考文献

1 李登实.数字图像压缩与加密技术研究[学位论文].武汉:华中师范大学,2007.

2 Katti RS, Srinivasan SK, Vosoughi A. On the Security of Randomized Arithmetic Codes Against Ciphertext-only Attacks. IEEE Trans. on Information Forensics and Security, 2011, 6(1): 19−27.

3 Kim H, Wen JT, Villasenor JD. Secure arithmetic coding. IEEE Trans. Signal Process., 2007, 55(5): 2263–2272.

4 Grangetto M, Magli E, Olmo G. Multimedia selective encryption by means of randomized arithmetic coding. IEEE Trans. Multimedia, 2006, 8: 905–917.

5 聂玉明,邹雪妹.一种基于像素域的改进的 SP IHT 算法.计算机仿真,2009,26(1):209−211.

6 邓家先,吴成柯,陈军.基于率失真斜率提升的干涉多光谱图像压缩.光学学报,2004,24(3):299−303.

7 Christopoulos C, AskelÖf J, Larsson M. Efficient methods for encoding regions of interest in the upcoming JPEG2000 still image coding standard. IEEE Signal Processing Letters, 2000, 7(9): 247−249.

8 Taubman D. High performance calable image compression with EBCOT. IEEE Trans. on Image Processing, 2000, 9(7): 1151−1170.

9 Taubman D, Ordentlich E, Weinberger M, Seroussi G. Embedded block coding in JPEG2000. Signal Processing-Image Communication, 2002, 17(1): 49−72.