

## 一种新型的无损视频压缩算法

夏杰 侯朝焕

(中国科学院声学研究所 北京 100080)

**摘要** 多次使用有损压缩技术压缩数字视频,会导致视频的质量大幅下降,从而降低了数字视频的可再利用性,为此设计了一种新型的无损视频压缩算法。该算法的特点在于:(1)提出了一种改进的基于上下文树的算术编码来压缩运动补偿后的误差帧。(2)针对改进的算术编码,优化基于宏块的运动估计与补偿算法,以提高无损视频压缩算法的压缩率。对于压缩运动补偿后误差帧的算法,与静态图像无损压缩算法 JPEG-LS 和 CALIC 相比表明,该文设计的无损视频压缩算法的压缩率超过 JPEG-LS 算法最高为 23.3%,超过 CALIC 算法最高为 19.3%。

**关键词** 无损视频压缩, 算术编码, 运动估计与补偿

中图分类号: TN919.8

文献标识码: A

文章编号: 1009-5896(2006)03-0385-05

## Lossless Video Compression System Designation

Xia Jie Hou Chao-huan

(Institute of Acoustics, Chinese Academy of Science, Beijing 100080, China)

**Abstract** It will create the problem of great degradation in video quality, when the digital video is compressed by using lossy video compression algorithm more than once. That will reduce the rate of reusing the digital video. Thus a new lossless video compression algorithm has been proposed. This algorithm features in: (1) an improved context-tree based arithmetic coding is proposed to compress the residual frame after motion compensation, and(2) macro-block based motion estimation and compensation method is optimized to improve the compression ratio of lossless video compression algorithm, with consideration to the improved arithmetic coding. When using JPEG-LS and CALIC, the existing state-of-the-art in lossless compression of still images, to encode those motion-compensated error-frames or residue frames to formulate benchmarks, the experiments illustrate that the proposed algorithm outperforms JPEG-LS by up to 23.3%, and CALIC by up to 19.3%.

**Key words** Lossless video compression, Arithmetic coding, Motion estimation and compensation

### 1 引言

由于有限的网络带宽和人类视觉系统的不敏感性,有损压缩技术成为了数字视频压缩研究领域的主流,其中的代表是MPEG系列<sup>[1,2]</sup>和H.26x系列<sup>[3]</sup>视频压缩标准。有损压缩技术利用舍弃一些人类视觉不敏感的视频图像信息,来获取较高的压缩比率。但是当我们多次使用有损压缩技术压缩数字视频时,可能丢失非常大的信息量,从而导致压缩后的视频质量非常低,这就大大降低了数字视频的可再利用性。无损视频压缩技术则可以在压缩过程中保持完整的视频信息,从而避免了压缩失真。

无损视频压缩技术之所以没有得到应有的关注,其中的原因包括:(1)无损视频压缩可以达到的压缩率远远赶不上实际应用的需要;(2)大部分的数字视频应用产品并不需要

解压的视频效果达到无损的级别。然而随着网络带宽的增加和人们对高质量数字视频需求的提高,高质量数字视频的无损压缩可能成为学术界和工业领域未来发展的方向。在e-cinema中,高质量数字电影采用逐帧渐进式的播放模式,而且每一帧都采用无压缩的三原色图像文件。在这样情况下,无损视频压缩技术将会对高质量数字电影的存档,发行和编辑等应用起到很大的帮助。

本文在MPEG2的基础上设计了一个新型的无损视频压缩算法,提出了一种改进的基于上下文树的算术编码算法作为无损压缩算法中的熵编码算法部分,同时针对改进的熵编码算法,优化了MPEG-2中基于宏块的运动估计与补偿算法,以提高无损视频压缩算法的压缩率。

### 2 无损视频压缩算法的结构

Motion-JPEG<sup>[4]</sup>视频压缩标准中提供了无损压缩的模式,

2004-09-16收到,2005-01-07改回  
国家973计划(G1999032900)和国家自然科学基金(60028102)资助课题

在整体上定义整个视频码流的数据格式,而每幅单帧的编码则分别由无损图像压缩算法进行处理,标准中不使用任何视频帧间编码算法。Motion-JPEG的框架虽然在硬件实现的成本上有一定的优势,但是无法利用视频中包含的时间域上的信息冗余也是相当大的损失。3D JPEG-LS<sup>[5,6]</sup>线性组合一系列基于参考帧的时间预测算子和JPEG-LS<sup>[7-9]</sup>中的空间预测算子,形成新的预测算子去估计需要编码的像素,但是预测效果依赖于线性系数的选择,同时基于图像压缩的框架不利于与现有发展成熟的有损视频压缩算法进行互补,基于这个原因,我们研究无损视频压缩算法的策略是,在目前最好的有损视频压缩算法的基础上,通过移除有损视频压缩算法中的有损子算法部分,使用新的无损子算法取代而设计新的无损视频压缩算法。这样设计的另一个好处在于不同的子算法,比如运动估计和熵编码,可以在平等的基础上进行性能比较。由于MPEG-4和MPEG-2在基本压缩算法上,比如运动估计,DCT和熵编码上采用类似的设计,所以我们采用MPEG-2作为基础设计新的无损视频压缩算法。

给定一个数字视频序列,我们根据不同的场景,将视频序列顺次分成若干个GOP(图像组),每个GOP中视频帧的场景基本一致,并且GOP之间的场景有明显的变化。由于场景发生了变化,所以对于每个GOP中的第一个帧,算法直接使用JPEG-LS进行帧内编码,而不使用运动估计与补偿,类似MPEG-2中的I帧。对于GOP中的其余帧,算法则通过由前一帧的运动预测和补偿,去除时间域上的信息冗余,然后使用熵编码算法编码运动补偿的误差和运动向量,类似MPEG-2中的P帧。图1给出了新的无损视频压缩算法的整体结构。

无损视频压缩算法需要消除两种信息冗余:像素原始值之间的信息冗余和预测后误差值之间的信息冗余。因此要优化无损视频压缩算法,重点应该放在两方面:(1)优化运动估计和补偿子算法部分,更好地去除像素原始值之间的信息冗余<sup>[5,6]</sup>;(2)优化熵编码子算法部分,更好地去除预测后误差值之间的信息冗余。本文的重点集中于优化熵编码子算法部分,提出了一种改进的基于上下文树的算术编码,作为无损视频压缩算法中的熵编码子算法部分,来提高无损视频压缩算法的压缩率。同时针对MPEG-2框架中的基于宏块的运动

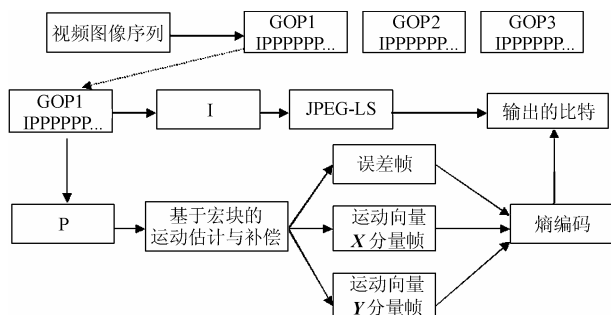


图1 无损视频压缩算法的整体结构

估计和补偿算法进行了优化,使之可以配合熵编码子算法部分,从而发挥更好的效果。

根据我们在有损视频压缩研究上的经验,在视频图像序列采用YUV格式的情况下,运动估计和补偿可以取得更佳的效果。所以我们依然在YUV格式上执行运动估计和补偿算法,这一点和MPEG-2一样,不同的是在我们的算法中,颜色向量 $U$ 和 $V$ 的图像尺寸和 $Y$ 向量的图像尺寸保持一致,以避免信息的丢失。预测后的运动向量分别储存于 $X$ 分量帧和 $Y$ 分量帧,这样可以更好地去除各运动分量帧内的空间信息冗余。

### 3 改进的基于上下文树的算术编码

虽然在无损视频压缩方面的研究工作不多,但是对于熵编码的研究却非常广泛,尤其在文本压缩和静态图像压缩方面。在已经发表的研究算法中,我们发觉这一领域最好的技术可以总结为:(1)基于树结构统计模型的算术编码;(2)基于简化统计模型的Golomb编码。前者代表了理论上最好的技术,通过为误差建立复杂的统计模型,来获取更好的压缩效率,但是需要花费更多的计算代价,CALIC<sup>[10]</sup>是使用这一技术的代表。后者代表了实际运用上的最好技术,通过为误差建立简化的统计模型,可以在花费较低计算代价的情况下取得相对不差的压缩效率,从而更加适合于在硬件和软件上的实现,JPEG-LS是使用这一技术的代表。由于无损视频压缩算法的压缩效果很大程度上依赖于熵编码,所以我们希望设计一种熵编码算法既能够保留基于树结构统计模型的算术编码的压缩效率,同时又可以通过简化结构和缩小计算复杂度来提高编码的速度,使之可以适应于实际应用的需要。基于这个目的,我们根据以下3个原则改进基本算术编码<sup>[11]</sup>和基于树结构的统计模型<sup>[12]</sup>。

(1)对于误差帧的每个误差像素“ $x$ ”,使用其附近的4个误差像素来建立上下文序列,具体结构如图2(a)所显示。对于边界上的误差像素,上下文序列的长度可以根据其所在的位置进行调整,原则如下:(a)第一行的第一个误差像素没有上下文序列;(b)其他所有第一行的误差像素的上下文序列中只有“1”;(c)其他所有第一列的误差像素的上下文序列只有“2”和“4”;(4)其他所有最后一列的误差像素的上下文序列只有“1”,“2”和“3”。与CALIC算法中使用的误差统计模型相比较,我们仅仅选择了和“ $x$ ”的4个相邻位置的误差像素(CALIC算法中为7个)来建立上下文序列。虽然我们使用了较少的上下文,但是这4个误差像素与“ $x$ ”的关联最大,并且没有像在CALIC算法中那样,被量化后才用于产生上下文,所以我们实际使用的上下文的数目超过了CALIC算法(576×8),这样可以更加准确地估计误差的统计模型。

(2)通过计算“ $x$ ”,“ $1x$ ”,“ $12x$ ”,“ $123x$ ”,和“ $1234x$ ”

在误差帧中出现的次数，来建立误差值“x”基于不同上下文的条件概率统计模型。

$$\left. \begin{aligned} P(x|1) &= \frac{n_1(x)}{\sum_{y \in \bar{E}} n_1(y)}, & \sum_{x \in \bar{E}} P(x|1) &= 1 \\ P(x|12) &= \frac{n_{12}(x)}{\sum_{y \in \bar{E}} n_{12}(y)}, & \sum_{x \in \bar{E}} P(x|12) &= 1 \\ P(x|123) &= \frac{n_{123}(x)}{\sum_{y \in \bar{E}} n_{123}(y)}, & \sum_{x \in \bar{E}} P(x|123) &= 1 \\ P(x|1234) &= \frac{n_{1234}(x)}{\sum_{y \in \bar{E}} n_{1234}(y)}, & \sum_{x \in \bar{E}} P(x|1234) &= 1 \end{aligned} \right\} \quad (1)$$

其中  $n_s(x)$  代表了  $x$  与上下文序列  $s$  一起出现的次数， $\bar{E}$  代表所有可能的误差值。所有的条件概率统计模型按照相应的上下文组织在一个类似文献[12]的树形数据结构内，我们称之为上下文树。由于上下文序列最多只能由 4 个误差像素构成，所以上下文树中最多只包含 4 层子节点，每个子节点代表了 1 个基于在这个节点之前所出现的所有上下文的误差值的条件概率统计模型。图 2(b)展示了一个典型的上下文树的例子。为了减少计算的代价和算法的复杂度，我们没有像文献[12]一样，为每个误差值“x”建立全部的四个上下文节点：“1x”，“12x”，“123x”，和“1234x”，因为在误差帧中，唯一路径上的上下文节点的频率计数是完全一样的，所以我们可以合并这些节点成为一个节点，并建立一个分支连接这个节点到上下文树相应的位置，因此每个分支代表了这些频率计数相同的上下文。图 2(b)中的树可以简化成图 2(c)中的结构。

(3)在上下文树中的每个节点上我们只对在该节点出现的误差值进行频率计数和保存，而不是对所有可能的误差值进行频率计数，因此每个节点的统计模型中都含有一个“US” (Unkown-Symbol)符号来编码在统计模型中新出现的误差值。

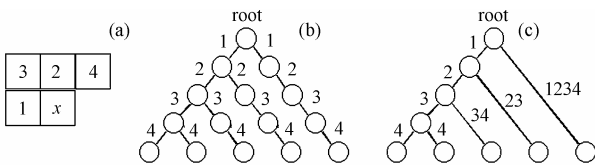


图 2

(a) 上下文结构 (b)上下文树的典型例子 (c)简化的上下文树

基于以上 3 个原则，对于视频序列中的每一个  $P$  帧，我们都建立一个上下文树来编码运动补偿后误差帧中的误差像素。首先对于一个需要编码的误差像素“x”，从上下文树的根节点起开始搜索，依次匹配“x”的上下文序列“1234”，直到出现第一个无法匹配的上下文。这时，在上下文树中的搜索停止位置可以分为 4 种情况：(a)搜索停止在 root 节点，上下文序列中没有上下文得到匹配；(b)搜索停止在一个已经存在的非 root 节点，上下文序列全部得到匹配；(c)搜索

停止

在一个已经存在的非 root 节点，上下文序列部分得到匹配；

(d)搜索停止在上下文树中的一个分支上。

对于情况(a)，根据 root 节点的统计模型编码误差像素“x”，如果 root 节点下没有误差像素“x”的频率计数，则编码一个“US”符号，同时将误差像素值“x”输入到一个文件 us.file 中保存。初始化一个“1234x”的节点，同时将整个上下文序列“1234”作为一个分支连接 root 节点和新产生的“1234x”节点，然后 root 节点统计模型中误差像素值“x”的频率计数加 1。

在使用误差值“x”初始化一个节点  $P$  的统计模型时，只保存“x”和“US”两个值的频率计数，且分别记为 1。

对于情况(b)，假设搜索停止在一个节点  $P$  上，如果在节点  $P$  的统计模型中，各个误差值的频率计数的总和大于一个预定的阈值 Threshold(表明该节点已经发展成熟)，则根据节点  $P$  的统计模型编码误差像素“x”。否则上移到节点  $P$  的父节点  $p'$ ，如果在节点  $p'$  的统计模型中，各个误差值的频率计数的总和大于 Threshold，则根据节点  $p'$  的统计模型编码误差像素“x”。否则依次类推，直到达到 root 节点。在编码过程中，如果需要编码的节点下没有误差像素“x”的频率计数，则编码一个“US”符号，同时将误差像素值“x”输入到一个文件 us.file 中保存。从节点  $P$  到 root 节点路径上的所有节点统计模型中误差像素值“x”的频率计数加 1。

对于情况(c)，假设搜索停止在一个节点  $P$  上，匹配的上下文序列为“12”，未匹配的序列为“34”，使用情况(b)的算法在节点  $P$  上编码误差像素“x”，然后初始化一个“1234x”的节点，同时将未匹配的上下文序列“34”作为一个分支连接节点  $P$  和新产生的“1234x”节点。

对于情况(d)，假设搜索停止在一个分支  $B$ “234”上，未匹配的上下文序列为“4”，使用情况(b)的算法在分支  $B$  的父节点“1x”上编码误差像素“x”。由于分支  $B$  产生了新的分支，所以分支  $B$  将分裂为两个分支：“23”和“4”。同时产生一个节点“123x”作为分支“23”和“4”的连接点，其统计模型直接从父节点“1x”继承。然后初始化一个“1234'x”的节点，同时将未匹配的上下文序列“4”作为一个分支连接节点“123x”和“1234'x”。

最后完成单幅误差帧的编码后，剩余的us.file使用基本算术编码<sup>[11]</sup>直接编码，同时X和Y运动分量帧分别采用与编码误差帧相同的方法进行压缩编码。

在该算法中，我们仅使用发展成熟的节点来编码误差值，这样可以解决由于上下文数目过多而产生的上下文稀释问题，同时在更新上下文树的时候，没有通过量化误差值来削减上下文的数目，可以使建立起来的上下文树更加符合误差分布的实际统计规律，从而取得更高的压缩率。

#### 4 优化的基于宏块的运动估计和补偿算法

MPEG-2 框架中  $P$  帧的运动估计和补偿算法是基于  $16 \times 16$  的宏块实现的, 对于当前帧的每个宏块, 从前一参考帧中寻找最相似的宏块, 进行求差运算, 从而得到误差值, 宏块的偏移量则是该宏块的运动向量。对于以  $(x, y)$  为起点的宏块, 运动向量  $(V_x, V_y)$  是通过下列公式获得的:

$$\text{SAD}(P_x, P_y) = \sum_{i=0}^{15} \sum_{j=0}^{15} |F_{t-1}(P_x+i, P_y+j) - F_t(x+i, y+j)| \quad (2)$$

$(x', y') = (P_x, P_y)$  满足:

$$\left. \begin{array}{l} |P_x - x| < 11, \\ |P_y - y| < 11, V_x = x' - x, V_y = y' - y \\ \text{SAD}(P_x, P_y) \text{ 为最小值} \end{array} \right\} \quad (3)$$

其中  $F_t$  和  $F_{t-1}$  分别是第  $t$  帧和第  $t-1$  帧。

该宏块中的每一个像素  $(i, j)$  运动补偿后的误差值  $E(i, j)$

$$E(i, j) = F_t(i, j) - F_{t-1}(i+V_x, j+V_y) \quad (4)$$

改进的基于上下文树的算术编码算法侧重于提取误差帧中空间信息冗余, 误差值之间关联性越大, 越容易获取更大的压缩率。根据式(3), MPEG-2 框架中运动估计和补偿算法使用的宏块预测标准是最小化误差绝对值之和(SAD), 但是在满足最小化 SAD 标准的预测结果中, 有些宏块的预测误差值之间的波动比较大, 反映出相邻像素误差值之间的关联性被减弱了, 在这种情况下会降低使用改进的基于上下文树算术编码算法的性能。因此我们在 SAD 的计算公式中引入一个反映宏块预测误差值之间关联性的参考变量 COR, 使得预测误差值可以更好地被压缩。新的 SAD(NewSAD)的计算公式如下:

$$\text{NewSAD} = \text{SAD} + \omega \cdot \text{COR},$$

$$\text{COR} = \sum_{i=0}^{15} \sum_{j=0}^{15} |E(i, j) - \bar{E}|, \quad \bar{E} = \frac{1}{256} \sum_{i=0}^{15} \sum_{j=0}^{15} E(i, j) \quad (5)$$

其中  $E(i, j)$  是宏块中预测后的误差值,  $\omega$  是相应的权值。

优化后的宏块预测标准就变为最小化 NewSAD。从式(5)中可以发现, 在运动估计的过程中, 我们依然需要将预测误差值尽可能地保持在一定的范围内, 这样不仅可以使宏块之

间相邻像素的误差值变化不会太大, 从而降低了对这些像素误差值的编码代价, 而且从整幅误差帧的角度来看, 误差值的全局关联性得到了一定的保证, 从而降低了上下文树中节点的数目, 节约了运算时间。权值  $\omega$  的引入就是为了取得全局关联性和局部关联性的平衡点, 通过实验, 我们发现权值  $\omega$  取值为 0.4 适用于很多情况。

#### 5 实验结果与分析

我们使用 5 个标准的 MPEG-2 测试视频: Mobile, Flower-Garden, Susie, Mom, Ttable-Tennis 和 1 个使用摄像机拍摄的视频: OSU-2, 进行了两组对比实验, 来验证设计的无损视频压缩算法的优越性。在第 1 组实验中, 比较在 Motion-JPEG 的框架下, 分别使用 JPEG-LS 和 CALIC 两种目前最好的无损图像压缩算法产生视频每一帧内容的码流的无损视频压缩算法, 与使用我们设计的基于 MPEG-2 框架的无损视频压缩算法 1(熵编码器部分采用基本算术编码<sup>[11]</sup>以及保留 MPEG-2 中原有的运动估计与补偿算法)分别压缩测试视频的结果, 从中我们发现 MPEG-2 框架下使用帧间基于宏块的运动估计与预测算法的压缩效果明显好于 Motion-JPEG 框架下 JPEG-LS 和 CALIC 中仅使用的帧内空间预测的算法, 表 1 列出了第 1 组实验的详细结果。

在第 2 组实验中, 我们将无损视频压缩算法 1 的熵编码器部分替换为提出的基于上下文树的算术编码, 同时使用优化的运动估计与补偿算法形成无损视频压缩算法 2。而参照算法变为: 参照算法 1 使用 JPEG-LS 来压缩运动补偿后的误差帧; 参照算法 2 使用和 CALIC 来压缩运动补偿后的误差帧。通过对比实验, 从中我们发现提出的基于上下文树的算术编码可以更加准确的估计预测误差的统计模型, 从而压缩率超过使用 JPEG-LS 的参照算法最多为 20.1%, 超过使用 CALIC 的参照算法最多为 18%, 表 2 列出了第 2 组实验的详细结果。

在算法实现的复杂度方面, 相比于基本算术编码<sup>[11]</sup>, 计算量主要增加在庞大的上下文树中搜索, 找到合适的节点。通过在每个节点的对象中加入一个长度为 256, 初始值为空的指针数组, 将子节点的指针放入数组中以子节点包含的上下文序列第一个数值的位置中, 这样匹配字符时就可以直接

表 1 无损视频压缩算法对比实验的结果(压缩比)

视频名称	视频尺寸	Motion-JPEG+JPEG-LS	Motion-JPEG+CALIC	无损视频压缩算法 1
Mobile	720×576	1.69:1	1.71:1	2.23:1
Flower-Garden	352×240	1.30:1	1.33:1	1.78:1
Susie	720×480	2.15:1	2.18:1	3.01:1
Mom	360×240	2.24:1	2.24:1	2.53:1
Table-tennis	352×240	1.33:1	1.35:1	2.29:1
OSU-2	320×240	1.84:1	1.85:1	2.03:1

表2 使用基于上下文树的算术编码后无损视频压缩算法对比实验的结果

视频名称	视频尺寸	参照算法 1	参照算法 2	无损视频压缩 算法 2	对比参照算法 1 的提高(%)	对比参照算法 2 的提高(%)
Mobile	720×576	2.18:1	2.28:1	2.48:1	13.8	8.8
Flower-Garden	352×240	1.81:1	1.91:1	1.99:1	9.9	4.2
Susie	720×480	2.83:1	2.98:1	3.49:1	23.3	17.1
Mom	360×240	2.92:1	2.96:1	3.53:1	20.1	19.3
Table-tennis	352×240	2.58:1	2.63:1	2.87:1	11.2	9.1
OSU-2	320×*240	2.11:1	2.19:1	2.32:1	9.9	5.9

访问包含下一个上下文的节点,直到遇到空指针。对于压缩一幅像素总数为 $N$ 的误差帧来说,比较基本算术编码<sup>[11]</sup>,计算复杂度的增加为 $O(N)$ 。当然添加指针数组使每个节点的对象大小增加了大概 512 个byte,考虑上节点对象包含的统计概率数组的大小,在不做任何优化的情况下,每个节点在内存中大约占据 1.5kbyte。这样在最坏的情况下(编码每个节点时都产生两个新的节点),对于 352×240(CIF)大小的误差帧进行编码时,整个上下文树将有 $2 \times 352 \times 240$ 个节点,大约需要占据 250M 的内存空间。目前的个人PC已经可以忍受这样的空间代价。

## 5 结束语

本文在 MPEG-2 的基础上设计了无损视频压缩算法,提出了一种改进的基于上下文树的算术编码来压缩运动补偿后的误差帧,同时针对改进的算术编码,优化基于宏块的运动估计与补偿算法,以提高无损视频压缩算法的压缩率。从实验结果发现,相比于 JPEG-LS 和 CALIC 中使用的帧内预测模型,基于宏块的帧间运动估计可以取得更好的预测结果,使得补偿后的误差帧更加易于压缩。同时提出的基于上下文树的算术编码可以更好估计补偿误差的概率分布,配合优化后的运动估计与补偿算法,从而进一步提高无损视频压缩算法的性能。进一步的工作可以把重点放在开发更好的运动估计与补偿算法,比如类似 H.264 中的基于可变尺寸块的运动估计算法,或者基于对象的运动估计算法,来提高无损视频压缩算法的性能。

## 参考文献

- [1] Mitchell J L, Pennebaker W B, Fogg C E, LeGall D J. MPEG Video: Compression Standard. New York: Chapman & Hall, 1996: 171 – 333.
- [2] ISO/IEC JTC1/SC29/WG11 N2725, MPEG-4 Overview, 1999.
- [3] TU-T Draft, H263, Video Coding for Low Bit Rate Communication, 1998.
- [4] Bhaskaran V, Konstantinides K. Image and Video Compression Standards: Algorithms and Architectures. Norwell: Kluwer Academic Publishers, 1997: 15 – 195.
- [5] Carotti E S G, De Martin J C, Meo A R. Low-complexity lossless video coding via adaptive spatio-temporal prediction. Proceedings. 2003 International Conference on Image Processing, Barcelona, 2003: 197 – 200.
- [6] Brunello D, Calvagno G, Mian G A, Rinaldo R. Lossless compression of video using temporal information. *IEEE Trans. on Image Processing*, 2003, 12(2): 575 – 586.
- [7] Weinberger M J, Seroussi G, Sapiro G. The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS. *IEEE Trans. on Image Processing*, 2000, 9(8): 1309 – 1324.
- [8] ISO/IEC 14495-1, ITU Recommendation T.87, Information technology—Lossless and near-lossless compression of continuous-tone still images, 1999.
- [9] ISO/IEC FCD 14495-2, JPEG-LS Part 2, 2000.
- [10] Wu X, Memon N. Context-based adaptive lossless image coding. *IEEE Trans. on Communications*, 1997, 45(4): 437 – 444.
- [11] Witten I H, Neal R M, Cleary J G. Arithmetic coding for data compression. *Communications of the ACM*, 1987, 30(6): 520 – 540.
- [12] Weinberger M J, Rissanen J J, Arps R B. Applications of universal context to lossless compression of grey-level images. *IEEE Trans. on Image Processing*, 1996, 5(4): 575 – 586.

夏杰: 男, 1979年生, 博士生, 研究方向为图像与视频处理。  
侯朝焕: 男, 1936年生, 中国科学院院士, 研究方向为信息和信号处理、声学等。