一种 SPIHT 编码和运动补偿相结合的可扩展视频压缩算法

明 亮 谢桂海
(中国人民解放军军械工程学院控制工程系,石家庄 050003)
贺玉文
(清华大学计算机科学与技术系人机交互与媒体集成研究所,北京 100084)

摘 要 提出了一种 SPIHT(set partition in hierarchical trees)编码和分块运动补偿相结合的视频压缩算法,适用于视频监控、视频会议、可视电话等许多视频图像的运动范围较小的场合。该算法具有如下 3 个特点:一是采用基于最低码率的自适应算法,实现了波动带宽下编码端和解码端参考帧的一致性;二是采用对预测残差帧的小波压缩编码,进一步增加了残差图的零像素数量,从而获得了更高的压缩比;三是采用了 SPIHT 多级树集合分裂算法,实现了码率的可扩展性,能适应波动带宽和不同性能接受端的需要。运用本文提出的算法,能进一步提高视频编码的压缩比和解码复原质量。实验结果表明,本算法对于运动范围较小的视频序列 Akiyo,具有很高的压缩比和较好的复原质量,在 27kbps 带宽下,压缩比可达 355 : 1,相应的 PSNR(Y) = 36.33dB, PSNR(U) = 40.22dB, PSNR(V) = 42.52dB;对于运动范围较大的 Singer 序列,本算法与 MPEG-4 校验模型相比,性能相差不大。此外,在最大码率下,本算法的 PSNR(Y)比 MPEG-4 校验模型平均提高约 4.0dB。

关键词 视频压缩 可扩展编码 SPIHT 离散小波变换 运动补偿
中图法分类号: TN919.81 **文献标识码**: A **文章编号**: 1006-8961(2004)08-0908-08

An Algorithm for Rate Scalable Video Compression Based on SPIHT and Motion Compensation

MING Liang, XIE Gui-hai
(*Department of Control Engineering, Ordnance Engineering College, Shijiazhuang* 050003)
HE Yu-wen
(*Department of Computer Science and Technology, Tsinghua University, Beijing* 100084)

Abstract This paper presents a SPIHT(set partition in hierarchical trees) and Motion Compensation Based Rate Scalable Video Compression algorithm. We will refer to this new technique as the Scalable Adaptive Motion Compensated SPIHT (SAMCS) algorithm, which can be used in video surveillance, video meeting, etc. SAMCS has three characteristics: Firstly, it uses adaptive Intra/Inter block coding and the lowest rate based reference frame reconstruction to adapt to the various bandwidth; Secondly, it makes wavelet transform on prediction error frame (PEF) to create a lot of zero pixels and achieve much higher compression rate; finally, SPIHT is a fine scalable coding, so SAMCS can get scalable rate bitestream based on the lowest rate reference frame reconstruction, which is very appealing for network-oriented application. Experimental results show that, SAMCS has an excellent performance on compressing low-motion video. For example, the PSNRs of Akiyo's three components(Y, U, V) are 36.33dB, 40.22dB and 42.52dB by SAMCS at 27kbps, and the compression rate is 355 : 1. As to high-motion video, such as Singer, the performance of SAMCS is no better than MPEG-4 VM. What's more, compared to MPEG-4 VM, the PSNRs of Y components is improved 4.0dB on the average at the highest bit-rate.

Keywords video compression, scalable encoding, SPIHT, discrete wavelet transform, motion compensation

基金项目:国家自然科学基金(60072050)
收稿日期:2003-11-21;**改回日期**:2004-04-08

1　引　言

视频压缩编码系统中，“时域运动补偿＋DCT”一直是广泛使用的压缩技术。MPEG-1、MPEG-2、MPEG-4[1,2]、H. 261、H. 263，以及最近发布的H. 264等视频标准中都采用了这一技术。时域运动补偿，实质上是通过运动估计获取运动向量，再通过运动补偿获得预测帧。其充分利用了视频图像帧间的相关性，从而有效地消除了视频图像的时间冗余，尤其对运动变化较小的视频序列，压缩效果明显。DCT 是一种容易实现的简便变换编码方法，但在低码率环境下，会出现明显的块效应，严重影响视觉效果。此外，由于 DCT 是在块内进行变换，使用 Z 扫描，因此其无论是帧内（即 I 帧）编码还是帧间（即 P 帧）编码都会将图像中的零系数以块分开。另外，DCT 系数的位置和幅值与原图不具有相似性，P 帧残差中大量连续的零像素的位置信息不能得到充分利用，而相对数量较少的非零像素则作为高频信号仍然游离在 DCT 系数块的高频区，再加上噪声的影响，使起主要压缩作用的 P 帧残差的零系数更加分散，不利于简单 Z 扫描后的游程编码，尤其对运动变化较小或较缓的视频压缩，其劣势更明显。

国内外相关研究成果有：文献[3]、[4]中使用了运动补偿且运用小波变换对预测残差帧（prediction error frame，PEF）进行压缩编码，但其使用的是固定码率的小波变换，视频压缩的码率不具有可扩展性；文献[5]提出了基于小波域内运动补偿的 SPIHT 分层视频编码算法，该算法具有可扩展性，但由于其中的运动补偿是在小波域中进行的，相位的偏移导致空间域的偏移非常明显，因此，运动补偿效果较差，尤其是在小波域的高频区子带里容易出现运动定位错误；文献[6]提出了一种可扩展的自适应运动补偿小波算法（SAMCoW），其利用了时域运动补偿，但采用的是传统的小波零树编码，且选用固定 I 帧，效率不是太高。为此，提出了块运动估计/补偿、SPIHT 编码、自适应码率控制和自适应算术编码相结合的压缩方法。实验结果表明，这种方法较好地利用了 P 帧残差的零像素，发挥了小波变换和 SPIHT 编码的优势，在对运动范围较小或在同等条件下，较缓的视频压缩时，可获得比 MPEG-4 更高的压缩比和更好的复原质量。

2　基于最低码率的自适应运动补偿

传统的运动补偿方法是在运动估计的基础上进行的。运动估计一般是将视频的帧分成 16×16 的宏块，每个宏块包括 4 个 8×8 的亮度分量 Y 块和 2 个色度分量 8×8 的 U、V 块，根据参考帧和当前帧，设定搜索范围和匹配规则（一般以最小 MSE 为匹配原则），按照一定的搜索算法，完成从参考帧到当前帧的宏块匹配，从而获得当前帧每一宏块的运动向量。运动补偿就是利用这些运动向量并结合参考帧完成对当前帧的有效预测，得到预测帧。运动估计一般仅对 Y 分量进行，由于 U、V 分量具有同 Y 分量极为相似的运动特性，因此，U、V 分量的运动补偿可以借用 Y 分量的运动向量完成。运动补偿的过程可表示为

$$\boldsymbol{P}_{\text{pred}} = M(\boldsymbol{P}_{\text{ref}}, \boldsymbol{m}) \tag{1}$$

其中，M 表示运动补偿函数，$\boldsymbol{P}_{\text{ref}}$表示参考帧，$\boldsymbol{m}$ 表示运动向量，$\boldsymbol{P}_{\text{pred}}$表示当前帧的预测帧。运动补偿的目的是通过已有的编码数据，有效地预测当前帧，通过只编码当前帧和预测帧的残差以及运动向量来代替编码整个当前帧，从而有效地减少编码数据量，消减视频帧时间上的冗余。预测残差可表示为

$$\boldsymbol{P}_{\text{diff}} = \boldsymbol{P} - \boldsymbol{P}_{\text{pred}} \tag{2}$$

其中，$\boldsymbol{P}$ 为当前帧，$\boldsymbol{P}_{\text{pred}}$为当前帧的预测帧，$\boldsymbol{P}_{\text{diff}}$为预测残差。

以上完成的是运动补偿的编码，编码结束后输出预测残差 $\boldsymbol{P}_{\text{diff}}$和运动向量 $\boldsymbol{m}$。

运动补偿的解码是在解码预测残差 $\hat{\boldsymbol{P}}_{\text{diff}}$、解码运动向量 $\hat{\boldsymbol{m}}$ 以及重建的参考帧 $\hat{\boldsymbol{P}}_{\text{ref}}$的基础上进行的，可表示为

$$\hat{\boldsymbol{P}}_{\text{pred}} = M(\hat{\boldsymbol{P}}_{\text{ref}}, \hat{\boldsymbol{m}}) \tag{3}$$

最终的解码帧 $\hat{\boldsymbol{P}}$ 由下式获得

$$\hat{\boldsymbol{P}} = \hat{\boldsymbol{P}}_{\text{pred}} + \hat{\boldsymbol{P}}_{\text{diff}} \tag{4}$$

通常，运动向量是无损编码，而且在编码端完成变换、量化、熵编码，输出比特流的同时，也进行熵解码、反量化、反变换，并以解码出的参考帧作为编码参考帧，从而保证了编码器和解码器使用的是相同的参考帧，即 $\boldsymbol{P}_{\text{pred}} = \hat{\boldsymbol{P}}_{\text{pred}}$。这样，$\hat{\boldsymbol{P}}_{\text{diff}}$在编解码过程中的损失就成了解码帧 $\hat{\boldsymbol{P}}$ 仅有的失真源，从而有效地保证了解码图像质量。由此可见，确保编码端和解码端使用相同的参考帧至关重要，传统的固定码率的运动补偿就是采用上述方法，对输出码流解码来达

到与参考帧的一致性，其缺点是对变码率的适应性较差。

为了适应网络资源波动对可扩展编解码的需要，提出基于最低码率的自适应运动补偿算法。该算法分为自适应 I、P 宏块编码和基于最低码率的运动估计两部分。

2.1 自适应 I、P 宏块编码

自适应 I、P 宏块编码，其类似传统的运动补偿，将视频的每一帧都分成 16×16 的宏块，第一帧使用帧内(I 帧)编码，其他帧都作为帧间(P 帧)编码。P 帧的宏块可以自适应地选择帧内(I 块)或帧间(P 块)编码，最终预测残差帧可以是既包括帧间残差宏块也包括帧内编码的宏块。决定用帧间或帧内编码模式的步骤如下：

(1)按下式对每一个 16×16 的宏块计算 $SAD_N(x,y)$

$$SAD_N(x,y)=\sum_{i,j=1}^{N}|p_{orig}-p_{prev}| \quad (5)$$

$$x,y\in[-64,63],N=16\text{ 或 }8$$

其中，P_{orig}是当前帧宏块像素，P_{prev}是前一重建帧(参考帧)的对应宏块像素，初值为

$$SAD_{16}(0,0)=\sum_{i,j=1}^{16}|P_{orig}-P_{prev}|-(N/2+1) \quad (6)$$

其中，$N=256$，表示一个宏块的像素个数。这样可以使零向量的 $SAD_{16}(0,0)$相对较小。

(2)根据下式计算 $SAD_{16}(X,Y)$和 $SAD_8(X,Y)$，并将(X,Y)作为当前块的运动向量。

$$SAD_N(X,Y)=\min(SAD_N(x,y)) \quad (7)$$

$$x,y\in[-64,63],N=16\text{ 或 }8$$

(3)根据下式计算 SAD_{inter}

$$SAD_{inter}=\min(SAD_{16}(X,Y),SAD_{k\times 8}) \quad (8)$$

其中，$SAD_{k\times 8}=\sum_{i=1}^{k}SAD_8(X,Y),k=4$。

(4)根据下式计算 A

$$A=\sum_{i,j=1}^{16}|p_{orig}-E| \quad (9)$$

其中，$E=(\sum_{i,j=1}^{16}p_{orig})/N,N=256$。

(5)按下式中 A 值的大小，确定编码模式

$$A<(SAD_{inter}-2N) \quad (10)$$

当式(10)成立时，则该宏块采用帧内编码模式；反之，采用帧间编码模式。

由于采用了自适应的帧内/帧间块编码模式，因此，可以对第 1 帧以后的所有帧都作为 P 帧处理。这样就有效避免了因固定的 I 帧而增加的不必要的码流。按上述步骤，对 Pigeon 序列图进行编码模式选择，所获残差如图版 I 图 1 所示，图中部分区域运动变化较大，编码器自适应地采用了帧内块编码模式。

2.2 基于最低码率的运动估计

设可适应的码率是 $R_L\leqslant R\leqslant R_H$，其中，$R_L$ 是能进行编解码最低码率，R_H 是编解码所需的最高码率，R 是实际可用码率。对所生成的预测残差帧和运动向量，在确保运动向量无损解码的前提下，无论是编码端还是解码端都在最低码率(R_L)下解码，并将解码重建帧作为参考帧。然后，由这样得到的参考帧进行运动估计，得到下一帧的运动向量。这样，只要满足最低码率下的应用需要，无论网络带宽等可用资源怎样波动，都能确保编、解码器参考帧的一致性和运动向量的有效性。

3 SPIHT 可扩展编码

SPIHT(多级树集合分裂算法)是一种改进的零树小波算法[7,8]，它基于小波变换，可有效提高视频压缩编码效率。使用 SPIHT 压缩图像具有以下优点：(1)解码图像质量好，尤其适用于彩色图像，可进行无损编码；(2)可生成完全嵌入式比特流文件，有利于渐进式图像传输；(3)可针对确切的比特流或失真率进行编码；(4)量化算法简单，编解码快速且几乎对称，具有较强的自适应性，应用广泛；(5)可进行有效的错误保护[9]。

一幅图像经过小波分解后形成了金字塔结构，图 1 为 3 级小波变换分解图。根据其数据结构的特点，就可以对小波系数进行零树编码，实现高压缩比[10]。

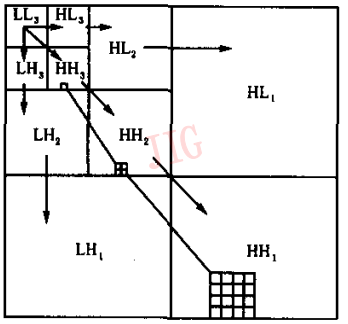


图 1 3 级小波分解图

零树编码的理论依据是在面变换和欧式范数度量空间等理论下得出的[10]，即，对原始图像小波变换后的所有子图像来说，像素的值越大，它对于重建图像质量的影响越大，越重要。依照这一假设，若给定一传输比特率，为得到尽可能好的重建图像，应以所有子图像中像素的大小为序进行输出，直至达到给定的比特率为止。零树编码有许多可能的方式，与小波变换相结合的 SPIHT 算法是一种十分有效的算法。它主要是通过改进的小波系数数据结构描述，实现快速有效的零树检测；通过层次树集合划分对重要系数信息进行优先编码，使得每一帧的重要信息都集中在码流的前面，每编码一比特都能增加图像的清晰度。

4　SPIHT 编码和运动补偿相结合的可扩展视频压缩算法

结合自适应运动补偿和 SPIHT 算法的精细可扩展优势，实现对视频图像的可扩展压缩，可适应于不同网络、硬件性能条件下的传输和解压缩。其方框图如图 2 所示。

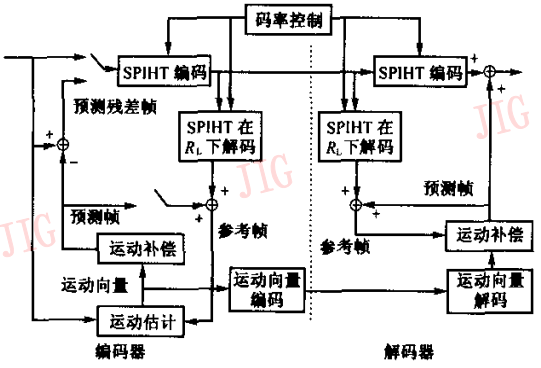


图 2　本文算法方框图

4.1　编码器

针对视频监控等实际应用的特点，将视频的第 1 帧作为 I 帧，以后每帧作为 P 帧，P 帧采用上述基于最低码率的自适应运动补偿，得到每帧的运动向量和预测残差图，运动向量单独无损编码输出，预测残差图先进行离散小波变换，再利用 SPIHT 算法进行小波系数的量化和分层编码，得到渐进的码率，最后使用自适应算术编码，输出码流。与此同时，解码反馈回最低码率下的重构帧，作为下一帧解码的参考帧。

传统的小波编码算法都是对静态纹理图像进行处理，由于正常图像数据都在[0,255]中，因此，为提高变换效率，都是采用无符号字符型(UChar)图像数据，作为离散小波变换的输入；解码时，离散小波变换的输出也是字符型。但在本算法中，由于残差数据不可避免地落在[−255,255]中，即有负值，虽然经过运动补偿后的残差数据的绝对值都很小，但如果简单将其限制到[0,255]中，经过小波变换和反变换后，重构数据会出现明显的失真，即使在运动非常小的连续视频中，几帧的差错积累就会使重构图像的质量陡然下降。为此，采用了短整型(UShort)离散小波变换(其在 UChar 型离散小波变换的基础上改进实现)，UShort 型变量占 2 个字节，完全能处理[−255,255]的数据，且占用内存空间不多。由于小波系数用 SPIHT 编码，需要对系数位平面从最高层起，逐层输出比特信息和定位信息。因为小波系数和原始图像具有相似性，为避免原始残差的负值转换为 UShort 型后变成 65535 附近的伪大系数，在输入残差图数据之前，先将每个像素的残差都加上 255，从而确保输入残差都为正值，有效避免了伪大系数的干扰。在小波解码后，再将重构的残差都减去 255，恢复原值。实验证明，这种 UShort 型的小波变换适合对残差进行小波变换，效果很好。图版 I 图 2 是视频序列 brea_cif 使用 UChar 型小波变换和 UShort 型小波变换的前 3 帧重构图像的比较，其中，上行是使用 UChar 型，下行是使用 UShort 型。

4.2　解码器

接收到渐进的码流后，进行反算术编码，然后分直流和交流子带分别解码。对于直流子带，进行反 DPCM、反量化；对于交流子带，则采用 SPIHT 解码，这样得到完整的小波系数后，若是 I 帧则直接小波反变换，得到重构图像，并作为下一帧的参考帧；若是 P 帧，首先提取出运动向量和最低码率下的残差帧比特流，以前一重构帧作为参考帧，结合无损解码出的运动向量，进行运动补偿，得到当前帧的预测帧，然后，加上最低码率下的 SPIHT 解码重构残差帧，作为下一帧解码的参考帧，同时继续利用该帧余下的码流进一步细化该帧纹理并显示出来。

4.3　自适应码率控制

为了适应带宽波动的情况，降低不必要的编码精度，提高编码效率和解码视频的连续性，采用自适应码率控制机制对每一帧图像的输出比特数进行控制。其实现方法如下：

(1)初始化

第 1 帧(I 帧)不进行码率控制。第 1 帧编码完后，做如下设置：设可用带宽为 R_s，令编码器的输出缓冲区

容量为 $B_s=R_s/2$；当前缓冲区水平为 $B_c=R_s/4$；前一帧头信息和运动向量占用比特数 $H_p=500$，当前帧的头信息和运动向量占用比特数 $H_c=500$。

用下式计算除去 I 帧后本段视频编码可用的比特总数

$$R_r = T_s R_s - R_f \tag{12}$$

其中，R_f 是第 1 帧(I 帧)编码用去的比特数，T_s 是本段视频的播放时间，例如：一段 300f 的视频序列，以 30fps 播放，则其 T_s 为 10s。

用下式计算在 R_s 带宽下，缓冲区在一帧时间平均能输出的比特数

$$R_p = R_r/N_r \tag{13}$$

其中，N_r 为视频序列帧数减 1(第 1 帧)。

(2)分配当前帧编码可用比特数 Z

从第 2 帧起，在每帧小波变换之后，SPIHT 编码之前都用下式计算当前帧编码可用比特数。

$$Z'=\mathrm{Max}(R_s/30, R_p\times 0.95+S\times 0.05) \tag{14}$$

其中，S 是前一帧编码用去的比特数。式(14)确保了每帧最小编码比特数为 $R_s/30$，同时结合前一帧和以后每帧的可用比特数来预测当前帧的可用比特数。然后，根据当前缓冲区状态再对可用比特数进行调节，保证缓冲区被充分利用，既不“溢出”也不“过剩”。

$$Z''=Z'(B_c+2(B_s-B_c))/(2B_c+(B_s-B_c)) \tag{15}$$

$$Z'''=\begin{cases}\mathrm{Max}(R_s/30, 0.9B_s-B_c), & B_c+Z''>0.9B_s\\ R_p-B_c+0.1B_s, & B_c+Z''<R_p+0.1B_s\end{cases} \tag{16}$$

$$Z=\mathrm{Max}(R_p/3+H_p, Z''') \tag{17}$$

(3)更新参数

每编码完一帧后，都要更新相关的码率控制参数，过程如下：

```
Bc=Bc+Rc-Rp;//Rc 为当前帧编码用去的比特数
Rr=Rr-Rc;
S=Rc;
Hp=Hc;
Nr=Nr-1;
While(Bc>0.8*Bs){ //表示在当前缓冲区水平为缓冲区的 80%时，就跳过下一帧
  Skip_next_frame;//表示跳过下一帧
  Nr=Nr-1;
  Bc=Bc-Rp;
}
```

转到第 2 步，进行下一帧的可用比特数分配。

在编码器端，通过自适应码率控制，较为合理地为每一帧分配可用的编码比特数，SPIHT 算法就依据这个比特数，输出相应的帧编码比特流。这样，编码器能充分利用现有带宽资源，并有效避免了缓冲区“溢出”。为了得到相同的参考帧，无论是编码器还是解码器，都必须在初始设定的最低带宽下进行解码。这样，编码器端需要在最低带宽和当前带宽这两种带宽下进行码率控制，从而得到两种精细程度的比特流：较粗糙的比特流用于重建参考帧，供下一帧编码使用；较精细的比特流用于编码器输出。相应地，解码器端需要在最低带宽和当前可用带宽这两种带宽下进行解码，得到两种精细程度的比特流：较粗糙的比特流用于重建参考帧，供下一帧解码使用；较精细的比特流用于解码输出重建图像。由于 SPIHT 是可扩展的，因此，实际编码时，上述两种比特流只传输较精细的比特流，而较粗糙的比特流是截取了较精细比特流的前面部分。图 3 是 Akiyo_cif 视频序列的在 30fps，带宽为 60kbps 和 120kbps 下，通过码率控制计算的每帧可用比特数的曲线图，可以看出，带宽变窄时，比特流会自适应下降。

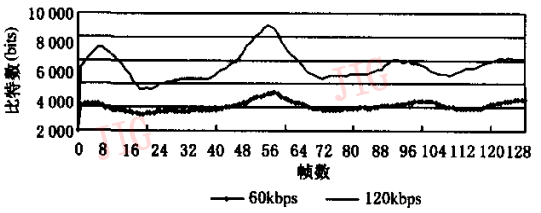


图 3 在不同带宽下自适应码率控制计算的每帧可用比特数

5 实验结果及讨论

测试序列为 CIF 格式，Akiyo 序列，300 帧，帧频为 10Hz；Paris 序列，120 帧，帧频为 10Hz；Singer 序列，250 帧，帧频为 30Hz。为便于比较，对 MPEG-4 校验模型关闭形状编码，第 1 帧为 I 帧，以后每帧都是 P 帧，没有 B 帧，表 1～表 3 给出了本文算法与 MPEG-4 校验模型的测试比较结果，其中列出了 Y，U，V 三分量的 PSNR 值，δ 为使用本文算法较 MPEG-4 校验模型各分量的提高值，表 4 为自适应性能测试比较结果，图 4 是 3 个序列的 Y 分量的率失真(Rate-Distortion)图，图版 I 图 3 是 Akiyo 序列在 30kbps 带宽下和 Paris 序列在 300kbps 带宽下复原的对比图，图版 I 图 4 是 Singer 序列在 500kbps 和 2000kbps 带宽下的复原对比图，其中(a)、(b)是在 500kbps 带宽下的结果；(c)、(d)是在 2000kbps 带宽下的结果。

表 1　Akiyo_cif 序列测试结果

带宽(kbps)	方法	Akiyo_cif/10fps/300f						
		Y(dB)	δ	U(dB)	δ	V(dB)	δ	编码帧数
20	MPEG-4	31.99	4.33	36.74	3.31	39.60	2.84	202
	本文算法	36.32		40.05		42.44		261
30	MPEG-4	32.81	3.60	37.29	2.98	39.96	2.44	300
	本文算法	36.41		40.27		42.10		272
40	MPEG-4	34.21	2.98	38.26	2.23	40.46	2.17	298
	本文算法	37.19		40.49		42.63		299
60	MPEG-4	35.80	2.80	39.53	1.95	41.24	2.34	299
	本文算法	38.60		41.48		43.58		297
80	MPEG-4	36.86	2.65	40.33	1.66	41.85	1.66	298
	本文算法	39.51		41.99		43.51		299

表 2　Paris_cif 序列测试结果

带宽(kbps)	方法	Paris_cif/10fps/120f						
		Y(dB)	δ	U(dB)	δ	V(dB)	δ	编码帧数
100	MPEG-4	26.33	5.17	32.36	3.45	32.81	3.35	110
	本文算法	31.51		35.81		36.16		70
300	MPEG-4	29.26	5.31	33.96	3.61	34.49	3.47	120
	本文算法	34.57		37.57		37.96		120
500	MPEG-4	31.47	6.06	35.67	3.94	36.13	3.84	120
	本文算法	37.53		39.61		39.97		120
800	MPEG-4	34.15	5.89	37.89	3.62	38.25	3.55	120
	本文算法	40.04		41.51		41.80		120
1 000	MPEG-4	35.45	6.09	38.88	3.93	39.19	3.85	120
	本文算法	41.54		42.81		43.04		120

表 3　Singer_cif 序列测试结果

带宽(kbps)	方法	Singer_cif/30fps/250f						
		Y(dB)	δ	U(dB)	δ	V(dB)	δ	编码帧数
500	MPEG-4	37.90	−1.0	39.11	−1.2	39.11	−1.8	247
	本文算法	36.86		37.84		37.30		250
800	MPEG-4	40.37	−1.2	41.32	−1.7	41.29	−2.1	248
	本文算法	39.14		39.62		39.19		250
1 000	MPEG-4	41.03	−0.8	41.73	−1.1	41.86	−1.6	250
	本文算法	40.20		40.55		40.26		250
1 500	MPEG-4	41.13	1.36	41.81	0.69	41.96	0.39	250
	本文算法	42.49		42.50		42.35		250
3 000	MPEG-4	41.13	4.19	41.81	3.49	41.96	3.38	250
	本文算法	45.32		45.30		45.34		250

表 4　自适应性能测试结果

视频序列	方法	可用带宽范围(kbps)	PSNR(Y)范围(dB)	最大可用带宽提高(%)	最大 PSNR(Y)提高(dB)
Akiyo	MPEG-4	20～400	31.993～42.636	87.5	2.592
	本文算法	20～750	36.322～45.228		
Paris	MPEG-4	100～2000	26.325～39.669	0	5.288
	本文算法	100～2000	31.514～44.957		
Singer	MPEG-4	500～1500	37.896～41.127	300	4.229
	本文算法	500～6000	36.857～45.356		

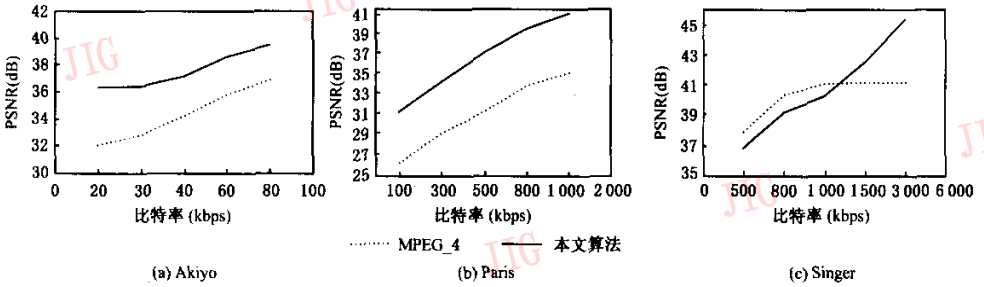


图 4 3 个序列的 Y 分量 Rate_Distortion 图

由表 1 可知，对于 Akiyo 序列，在不同带宽下，本文的算法比 MPEG-4 校验模型 PSNR(Y)平均提高了约 3.3dB。由表 2 可知，对于 Paris 序列，在不同带宽下，本文的算法比 MPEG-4 校验模型 PSNR(Y)平均提高了约 5.6dB。同时，本文的算法具有较高的压缩比，以 Akiyo 序列为例，在 27kbps 带宽下，压缩比可达 355∶1，相应的 PSNR(Y)＝36.33 dB，PSNR(U)＝40.22dB，PSNR(V)＝42.52dB。此外，由表 4 可知，本文算法较 MPEG-4 校验模型有更强的带宽自适应能力，在最大可用带宽下，其 PSNR(Y)比 MPEG-4 校验模型平均提高约4.0dB。但是，由表 3 数据可以看到，本文的算法，对运动范围较大的 Singer 序列效果不是非常理想，在码率较低时，比 MPEG-4 校验模型的 PSNR(Y)平均还低 1dB 左右，直到码率高于 1500kbps 时，PSNR(Y)才有明显提升，最终高于 MPEG-4 校验模型约 4dB。其主要原因是当运动较大时，预测残差帧中边缘纹理较多，且几乎都为高频信号，相应的小波系数也较复杂，这使编码数据量和计算量明显增大。因此，当带宽较小时，在自适应码率控制下，SPIHT 编码的输出被截断较多，使得 PSNR 较低；而带宽较大时，SPIHT 编码的大部分码流都输出了，PSNR 就会迅速提高。

6 结 论

SPIHT 编码和运动补偿相结合的可扩展视频压缩算法，适用于视频监控、视频会议、可视电话等许多视频图像的运动范围是较小的场合。实验结果表明：在 27kbps 带宽下，压缩比可达 355∶1，相应地，PSNR(Y)＝36.33dB，PSNR(U)＝40.22dB，PSNR(V)＝42.52dB。在最大码率下，本文算法的 PSNR(Y)比 MPEG-4 校验模型平均提高约 4.0dB；在低码率下，本文算法优势不明显，但当运动范围较小时，本算法的 PSNR(Y)、PSNR(U)、PSNR(V)比 MPEG-4 校验模型仍有较大的提高。为了和 MPEG-4 进行比较，本文采用的是同 MPEG-4 类似的固定大小(16×16)块运动估计算法。下一步将研究运动估计改进算法，以实现在运动范围较大时，仍能获得较小的预测残差帧，从而使本算法适用于更广泛的视频压缩应用中。

参 考 文 献

1 ISO/IEC JTC1/SC29/WG11 N3908, MPEG-4 Video Verification Model Version 18.0[S]. The Pisa meeting of 2001.

2 钟玉琢，王琪，贺玉文. 基于对象的多媒体数据压缩编码国际标准——MPEG-4 及其校验模型[M]. 北京：科学出版社，2000：460～468.

3 Ohta M, Nogaki S. Hybrid picture coding with wavelet transform and overlapped motion-compensated interframe prediction coding[J]. IEEE Transactions on Signal Processing, 1993, **41**(12): 3416～3424.

4 Martucci S A, Sodagar I, Chiang T, *et al*. A zerotree wavelet video coder[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1997, **7**(2): 109～118.

5 Lee Jae-Yong, Oh Hyo-sub, Ko Sung-Jea. Motion-compensated layered video coding for playback scalability [J]. IEEE Transactions on Circuits and Systems for Video Technology, 2001, **11**(5): 619～628.

6 Shen Ke, Delp Edward J. Wavelet based rate scalable video compression[J]. IEEE Transactions on Circuits and Systems for Video Technology, 1999, **9**(2): 109～122.

7 Said A, Pearlman W A. A new fast and efficient image codec based on set partitioning hierarchical trees [J]. IEEE Transactions on Circuits and Systems for Video Technology, 1996, **6**(3): 243～250.

8 Khan E, Ghanbari M. Efficent SPIHT-based embedded colour image coding techniques[J]. Electronics Letters, 2001, **37**(15): 951.

9 Said A. SPIHT image Compression[OL], http://www.cipr.rpi.edu/research/SPIHT/spiht1.html. 2002-11-11.

10 Shapiro J. Embedded image coding using zerotrees of wavelet coefficients[J]. IEEE Transactions on Signal Processing, 1993, 41(12): 3445～3462.

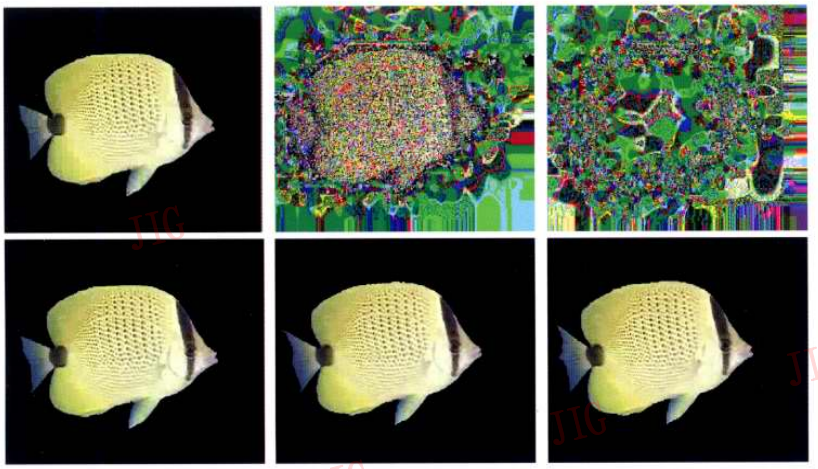
明　亮　1979 年生。2001 年于中国人民解放军军械工程学院控制工程系获工学学士学位，2004 年于中国人民解放军军械工程学院计算机应用技术专业获工学硕士学位。现在北京系统工程研究所工作。主要研究兴趣为视频编码、网络多媒体信号处理等。已发表论文 9 篇。

谢桂海　1941 年生。教授、博士生导师。1965 年毕业于清华大学自动控制系。目前的主要研究方向为小波理论及应用、多媒体信号处理、模式识别等。已发表论文 60 余篇，获国家发明专利 2 项，军队科技进步奖 6 项，出版专著 5 部。
E-mail：xghpro@163.net

贺玉文　1974 年生。讲师。2002 年于清华大学计算机科学与技术系获博士学位。主要研究领域为视频图像处理、多媒体技术等。

(a) 第279帧残差图　(b) 第279帧原图　(c) 第280帧残差图　(d) 第280帧原图

图1　帧内帧间混合残差图和对应的原图



(a) 第1帧　(b) 第2帧　(b) 第3帧

图2　UChar型和UShort型小波变换的前3帧重建图像

(a) 本文算法 PSNR=36.324dB　(b) MPEG-4 PSNR=31.985dB　(c) 本文算法 PSNR=34.476dB　(d) MPEG-4 PSNR=29.133dB

图3　Akiyo序列的第153帧和Paris序列的第73帧复原比较

(a) 本文算法 PSNR=36.605dB　(b) MPEG-4 PSNR=37.488dB　(c) 本文算法 PSNR=42.570dB　(d) MPEG-4 PSNR=41.319dB

图4　在500kbps和2000kbps带宽下Singer序列的第83帧复原比较