交流图片
图像和视频编码课程
2014 年，第 361-410 页

第11章-沟通图片：跨网络交付

大卫·R· 布尔

https://doi.org/10.1016/B978-0-12-405906-1.00011-8

摘要

本章介绍如何利用各种错误检测、校正和缓解方法，通过有损渠道实现压缩视频内容的可靠交付。本章首先介绍了对有效抗错视频编码系统的要求，然后解释了错误是如何产生的，以及如何在空间和时间上传播。然后，它研究了一系列可用于减轻错误和错误传播影响的技术。最初，考虑了依赖网络参数的操纵或利用网络功能来实现这一点的方法。接下来是分析两种方法，EREC和PVQ，其中编解码器生成的比特流本质上对错误具有鲁棒性。最后，考虑了隐藏而不是纠正比特流错误的仅解码器方法。这些在不增加传输开销的情况下提高了主观质量。通篇提供了示例。

上 下一

关键词

弹性错误视频; 同步代码词; 可逆VLC; 错误跟踪; 跨层优化; 链接自适应; 错误弹性熵编码（EREC）; 金字塔矢量量化（PVQ）; 错误隐藏

视频压缩算法依靠时空预测和可变长度熵编码相结合来实现高压缩比，但因此，它们会产生一个编码的比特流，该比特流本质上对信道错误敏感。当视频信息通过不可靠的网络传输时，这成为

一个主要问题，因为传输过程中引入比特流的任何错误都会迅速传播到图像序列中的其他地区。

为了促进有损渠道的可靠交付，通常调用各种错误检测和纠正方法。本章首先介绍了对有效抗错视频编码系统的要求，然后解释了错误是如何产生的，以及如何在空间和时间上传播。然后，我们研究了一系列可用于减轻错误和错误传播影响的技术。我们最初考虑了依赖网络参数操作或利用网络功能来实现此目的的方法；然后我们继续考虑两种方法，EREC和PVQ，其中编解码器生成的比特流本质上对错误具有鲁棒性。最后，我们提出了仅解码器的方法，可以隐藏而不是纠正比特流错误。这些在不增加传输开销的情况下提高了主观质量。

11.1。运营环境

11.1.1.现代网络的特点

固定IP网络

*互联网*本质上是在同一协议下运行并通过路由器连接的独立分组交换网络的集合。每个数据包都有一个标头，用于标识其源IP地址和目标IP地址，其中数据包使用TCP/IP协议向目标传输。*互联网协议*（IP）为数据包交付提供了基础，*传输控制协议*（TCP）使用*自动重复重测*（ARQ）技术提供了最大努力的交付机制，但交付时间没有限制。因此，这是一种不可靠、最努力、无连接的数据包交付服务，无法保证数据包的实际交付。

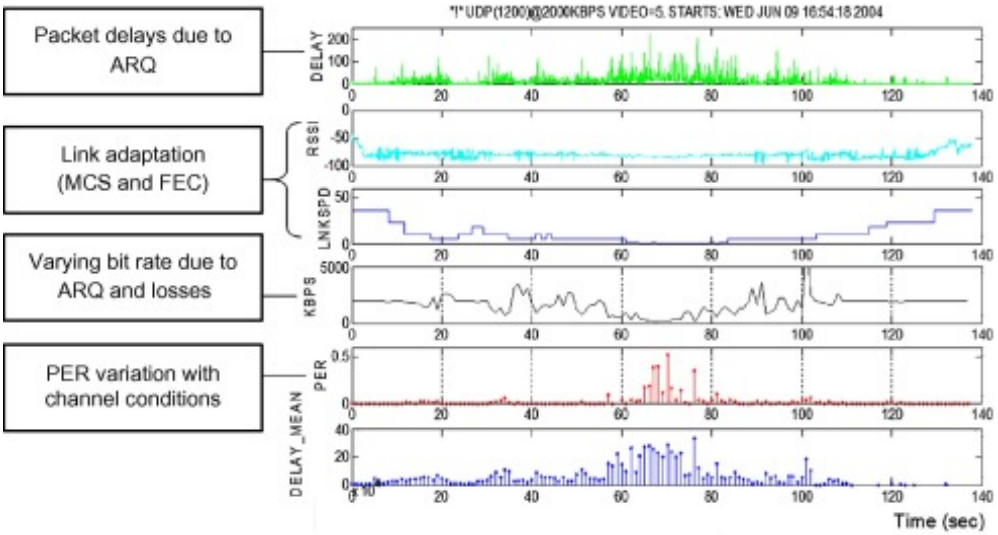
对于不太可靠的网络和实时延迟受限的应用程序，通常首选用户*数据报*协议（UDP）。*实时传输协议*（RTP）在UDP之上运行，包含必要的时间和序列信息。与RTP一起，*实时控制协议*（RTCP）允许接收方向发送方提供服务质量（QoS）反馈，使其能够调整其传输特性，以更好地适应当前的网络条件。这通常与*实时流协议*（RTSP）一起使用，用于控制流媒体服务器以及建立和控制媒体会话。

无线网络

大多数现代无线网络也在IP协议下运行，以支持与互联网主干网无缝集成。然而，无线环境的带宽限制和容易出错的性质需要额外的适应机制。这些通常基于一套结合不同错误控制和调制方案（MCS模式）的操作模式，可以根据流行的信道条件选择这些模式。它们在调制解调器*的物理层*（PHY）和*介质访问控制*（MAC）层实现。

无线网络主要分为两大类，它们在新的4G标准下正在迅速融合。*无线局域网*（WLAN）始终基于IEEE 802.11标准，并通过热点提供本地高速数据传输。数字*蜂窝网络*最初旨在支持语音和数据流量，但现在，在3G和4G标准下，提供视频下载和对话视频服务的主要手段。

我们可以在图11.1中看到一个典型的无线信道对系统性能的影响。这表明在布里斯托尔大学无线视频研究平台上记录的数据。数据日志跟踪显示了当用户离开接入点（时间：1-70 s）然后返回接入点（时间：70-140 s）时，802.11 WiFi 链接如何适应。这说明了无线电环境是如何变化的，以及这对数据包延迟、数据包延迟变异性、链路速度（调制和编码模式）以及应用程序层的比特率的影响。显然，我们的源编码方法需要适应这些可变带宽条件，同时需要添加信道编码或其他抗错手段来应对传输过程中引入的错误。



下载：下载全尺寸图像

图11.1。记录无线视频试用的无线电链路参数变化与信道条件。

11.1.2。传输类型

下载和流媒体

可以通过多种不同方式访问视频文件或流。在最简单的情况下，这可以采取文件传输的形式，其中数据存储在本地机器上，然后在适当的播放器中打开。*渐进式下载*也是文件传输，通常使用HTTP协议，文件也存储在本地，但它们与播放器更紧密地链接。播放器可以在下载完成之前开始播放，但通常不支持交互式。当文件从服务器下载时，它将被缓冲，当有足够的数据可以连续显示时，它将开始播放。

在流媒体的情况下，视频通常不存储在用户的设备上，而是直接从缓冲区播放。流媒体服务器将与用户的播放器交互，以启用视频的交互式导航（例如通过RTSP获得FFWD、RWD等）。

交互式通信

通过流媒体视频，当接收器缓冲区足够满以确保持续播放时，播放将开始。显然，如果条件规定网络无法支持编码的视频比特率，那么缓冲区将逐渐清空，最终视频将卡顿和冻结。在需要互动的情况下，对延迟的要求要严格得多；通常200-300毫秒被认为是对话服务的最大可容忍度，以避免不自然的犹豫。

单播传输

在单播传输的情况下，一个发送者连接到一个接收器。这对于会话服务来说是强制性的，但对于其他服务来说，效率可能很低，因为每个客户端都需要自己的带宽。对于单个发射机在Bbps到N接收器，所需带宽总为NBbps。单播传输的优点是发送方和接收方之间有密切联系，因此可以建立一个反馈通道来发出错误信号，从而支持重传和错误控制。YouTube和Skype等运营商使用单

播流媒体。

多播传输

在多播传输的情况下，许多接收器会加入一个流。这比单播案例效率高得多，因为我们的N接收器现在只消耗Bbps。然而，所有接收器将经历不同的信道条件，如果接收器数量众多，则反馈机制变得令人望而却步，必须调用其他错误控制机制。因此，与单播相比，多播提供的服务灵活性要小得多，但提供了更好的带宽灵活性。Ustream等运营商使用多播传输。

11.1.3。运营约束

如上所述，IP流、会议和娱乐视频一直是固定互联网和移动网络中多媒体通信服务增长的主要驱动力。与这些网络服务相关的信道可以由以下三个属性来描述：

1. **无线信道的带宽有限：**这需要使用复杂的源编码技术来匹配信号的比特率和信道的容量。
2. **无线网络的信道质量可能变化很大：**这是由于信号衰减和干扰导致传输错误，以及需要错误保护、错误校正和重传。
3. **数据包交换网络可能会发生拥塞：**这可能会导致数据包掉落和变量延迟。这些要求智能缓冲和速率控制策略，并对重传施加了严格的限制。

我们通过利用视频序列中的数据相关性——使用帧间预测（运动估计）、帧内预测和可变长度熵编码，如何解决属性（1）。然而，因此，这些方法会产生一个编码的比特流，该比特流本质上对信道错误敏感，这在属性（2）和（3）的背景下提出了一个特殊问题。

显然，我们希望我们的编码器表现出出色的速率失真（或速率-质量）性能，但用户真正感兴趣的是解码器的质量，因为这是他或她所经历的。事实上，如果所有数据包在传输过程中丢失，编码器的速率失真性能无关紧要。为了促进有损通道的可靠交付，通常调用各种额外的错误检测和纠正方法。总之，编码的视频比特流需要在速率失真意义上具有抗错性以及效率。

11.1.4。错误特征

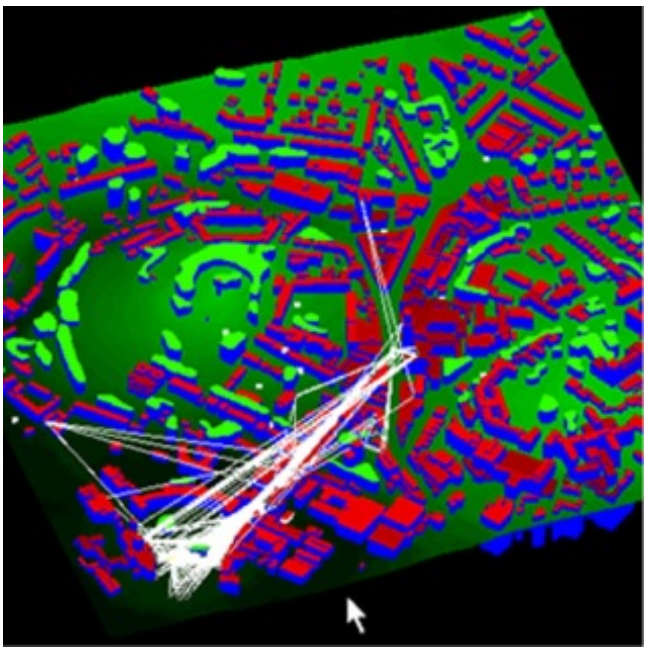
错误类型

传输过程中发生的错误大致可分为两类：*随机位错误*和*擦除错误*。前者通常使用统一的pdf建模，而后者则具有突发性。位误差通常以*位误差率*或BER为特征，可由热噪声引起。擦除错误会影响较长的数据段，可能由数据包网络的拥塞或无线网络中的衰落、阴影和退出引起。它们通常将超过任何信道编码系统的校正能力，并将要求重传或导致整个数据包的丢失（本身就是*应用程序层*的擦除）。

测试数据

为了建模错误对视频信号的影响，我们需要确保使用有代表性的测试数据来评估所使用的编码方法。这可以通过使用真实系统长时间记录数据（如图11.1所示）或使用无线传播建模或统计信道建模模拟系统性能来实现。我们这里不详细讨论图11.1，而是回到它所描述的一些事情。

在实践中，虽然使用更通用的模型来描述系统的性能是有用的，但为了现实的有用性，最好使用传播模型和适当的网络模拟工具在应用程序层生成现实的比特错误模式。参考文献中描述了这些方法。[1]。射线跟踪分析仪的示例输出为特定无线配置提供了特定环境中多路径效应的准确表示，如图11.2所示，参考文献中对此进行了描述。[2]。



下载：下载全尺寸图像

图11.2。使用光线跟踪的无线电波传播建模。

由A提供。尼克斯。

编码类型

编码模式将对错误在空间和时间上显示和传播的方式产生重大影响。例如，如果符号同步丢失，转换编码与可变长度编码相结合可能会导致块内的空间错误传播。如果EOB码字解码不正确，将发生跨多个块的传播；在这种情况下，转换块中的任何错误都将在空间上传播到相邻的像素区域。小波子波段的情况类似。

误差的空间或时间传播也可能由预测编码引起。任何用作解码器预测基础的像素或变换系数中的错误将导致不正确的重建，这些重建将围绕预测循环传播，导致后续像素或转换系数的错误。例如，用于运动估计的参考框架中的误差将通过运动补偿过程临时传播。

11.1.5。挑战和解决方案框架

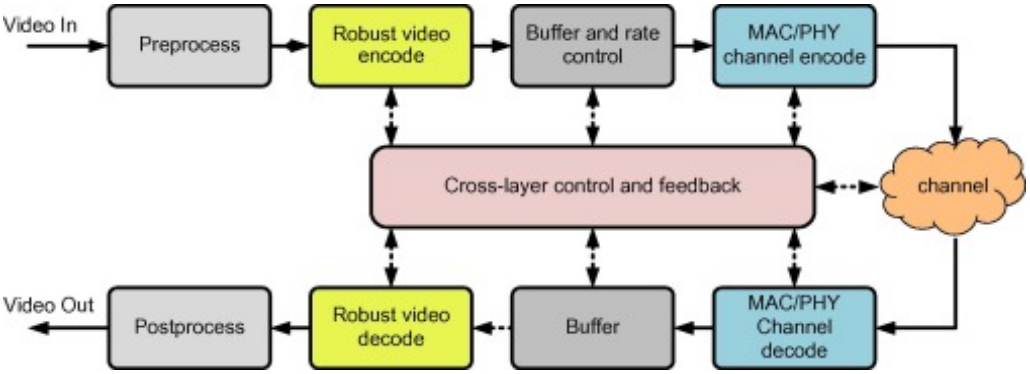
到目前为止，我们一直试图从压缩系统中实现最佳速率失真性能。我们在不影响质量的情况下，从视频信号中尽可能多地删除冗余。然而，我们现在面临着一个困境，即为了使压缩的比特流在出现错误时变得健壮，我们需要以信道编码的形式将冗余引入比特流，从而抵消压缩期间获得的部分或可能全部收益。显然，这不是一个好情况！

因此，抗错编码的挑战在于保持通过源压缩获得的编码增益，同时确保解码的比特流在传输过程

中保持完整性，从而可以忠实地解码。为了实现这一目标，我们需要：

1. 了解错误的原因及其影响（第11.2节）。
2. 从心理视觉角度描述这些，以了解丢失文物对人类视觉系统的影响（第2章和第11.3节）。
3. 利用适当的信道编码和重传解决方案，但在可能的情况下，以适合视频内容的方式使用这些解决方案（11.3节减轻比特流错误的影响，11.4传输层解决方案）。
4. 利用许多无线或互联网协议中存在的自适应机制，在应用程序、访问控制和物理层联合工作，以产生联合源-信道编码或跨层解决方案（11.4节传输层解决方案，11.6跨层解决方案）。
5. 利用视频编码过程的结构，使编码的比特流本质上对错误更健壮（第11.5节应用程序层解决方案，11.7本质上健壮的编码策略）。
6. 了解重建视频中错误发生的地方，并采取措施隐藏而不是纠正它们（第11.8节）。

图11.3显示了一种通用架构，它抓住了这种方法的本质。在实践中，抗错解决方案以富有同情心的方式结合了传输层解决方案和应用层解决方案的各个方面。也就是说，在可能的情况下，应调整网络参数以适应视频需求，并管理视频参数以匹配网络约束。以下各节将更详细地讨论这些问题。



下载：下载全尺寸图像

图11.3。通用的抗错视频传输架构。

11.2。损失的影响

11.2.1.同步失败

正如我们在第7章中已经看到的，编码的视频比特流通常由一系列可变长度的码字组成。所有实用的图像和视频编码标准都采用霍夫曼口算编码等 VLC 方法。固定长度码字（FLC）在实践中很少使用，除非符号概率不能证明使用可变长度编码是合理的（例如，在某些标题信息中）。另一个例外是一类具有抗错性的方法，包括金字塔矢量量化（PVQ），我们将在第11.7.2节中研

究。

我们将在以下小节中看到，问题造成的程度将与错误发生的位置和使用的编码类型有关。

单位错误的影响

使用FLC，任何受单位错误影响的码字都将被错误解码，而所有其他码字都不会受到影响。这意味着不会丢失比特流或符号同步。与VLC相反，单位错误可能导致解码器解码更长或更短的码字，这意味着后续符号虽然正确传递，但由于解码器失去同步，极有可能被错误解码。这种同步的丢失可能会持续到下一个显式同步点，并且通常会导致解码错误的符号数量。这可能意味着重要的隐式同步符号，如EOB，将被遗漏。在实践中，大多数霍夫曼代码会重新同步自己，而算术代码很少同步。示例11.1中的两个场景说明了VLC中的同步丢失问题，展示了位流、符号和编码单元（例如转换块）级别的同步之间的微妙差异。

示例11.1由于单位错误导致VLC同步丢失

给定以下一组符号及其相应的霍夫曼码字，

字母表	**A**	**B**	**C**	**D**	**E**
霍夫曼代码	0	10	110	1110	1111

发送以下消息：

信息	**B**	**A**	**D**	**E**	**C**	**B**	**A**
编码比特流	10	0	1110	1111	110	10	0

（a） 如果第六位出现单位错误，请计算并注释解码后的符号序列。

（b） 对于在**10th**位位。

解决方案

（a） 解码位和符号表如下，错误位以粗体字体表示：

编码消息	**B**	**A**	**D**	**E**	**C**	**B**	**A**	**–**
接收比特流	10	0	1100	1111	110	10	0	
解析比特流	10	0	110	0	1111	110	10	0
解码消息	B	A	C	A	E	C	B	A

在这种情况下，符号D被错误地解码为C，后跟A，然后发生比特流重同步。然而，解码了一个额外的符号这一事实意味着后续符号被移位，因此符号同步丢失。

(b) 解码位和符号表如下，错误位以粗体字体表示：

编码消息	**B**	**A**	**D**	**E**	**C**	**B**	**A**
接收比特流	10	0	1110	1101	110	10	0
解析比特流	10	0	1110	110	1110	10	0
解码消息	B	A	D	C	D	B	A

在这种情况下，由于相邻符号的长度和编码，误差仅限于两个符号。之后，实现码流和符号重同步。表面上看，这比第一种情况好得多。然而，考虑符号表示变换系数的{运行/大小}编码的情况。第四、第五个符号出现错误，说明在第三个符号之后，由于符号4、5对应的零的运行不正确，整个块会被解码错误。然而，由于符号重新同步被恢复，任何后续的EOB符号都将被正确检测到，并保持块级同步。有关更多详细信息，请参阅示例11.2。

11.2.2.标题丢失

头信息的错误可能导致灾难性错误或错误传播。如果标头中的任何位都损坏了，那么结果通常是解码器完全失败，因为解码位流所需的基本参数不正确。因此，标题信息比其他数据受到更有力的保护是有道理的，因为没有它，其他信息就无关紧要了。

11.2.3。空间误差传播

由于以下原因，错误可以在解码过程中在空间传播：

1. **DPCM预测：**例如，用于JPEG中直流变换系数的编码。这通常会导致错误后解码的图像块的亮度或色度变化。
2. **预测内部：**现代压缩标准采用复杂的预测内部模式，可能导致误差的传播。
3. **块内VLC符号同步丢失：**例如，在基于DCT的系统中，这将导致出现与不正确的DCT基函数叠加相关的工件。
4. **块同步丢失：**如果EOB符号解码不正确，错误将在块之间传播。这通常会导致位置同步的丢失。

图11.4说明了单个错误对简单JPEG DCT/霍夫曼编码图像的影响。这显示了示例11.1中描述的效果，例如11.2。我们清楚地看到错误发生的块，因为损坏的DCT基函数是显而易见的。该图还说明了DPCM方法传播的腐败DC水平的影响。还显示了重建图像的空间移动的影响，因为错误块附近的块中遗漏了EOB符号。最终，符号同步被恢复，但空间位移的影响仍然存在。错误率越高的影响及其对主观图像质量的影响如图11.5所示。

示例11.2由于VLC错误而产生的空间误差传播

给定示例11.1中的符号集及其相应的霍夫曼码字，我们现在假设这些符号对应于图像中编码内部DCT块的{run/size}值。如果在这种情况下，符号B对应于EOB，则评论与以下传输序列相关的空间错误传播，当在图像第一个块的位位置12引入单个错误时：

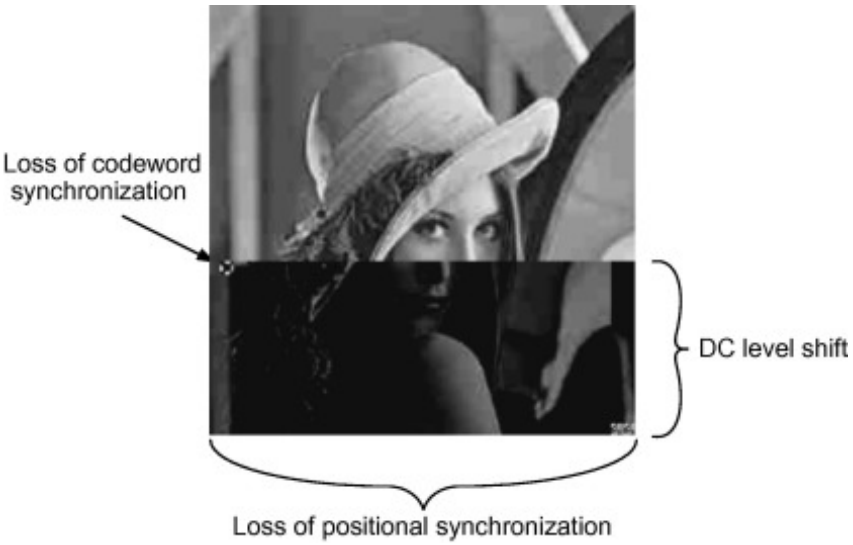
信息	**A**	**D**	**E**	**C**	**B**	**D**	**A**	**E**	**C**	**B**
编码比特流	0	1110	1111	110	10	1110	0	1111	110	10

解决方案

解码位和符号表如下，错误位以粗体字体表示：

信息	**A**	**D**	**E**	**C**	**EOB**	**D**	**A**	**E**	**C**	**EOB**
收到的位	0	1110	1111	111	10	1110	0	1111	110	10
解析位	0	1110	1111	1111	0	1110	0	1111	110	10
解码	A	D	E	E	A	D	A	E	C	EOB

在这种情况下，符号C和B被错误地解码为E和A，因此缺少第一个EOB符号。因此，尽管发生位流和符号重同步，但块同步丢失了。因此，所有随后解码的块都将被左侧的一个块位置所取代。因此，除非引入某种显式同步形式，否则第二个块之后的整个图像将被损坏。



下载：下载全尺寸图像

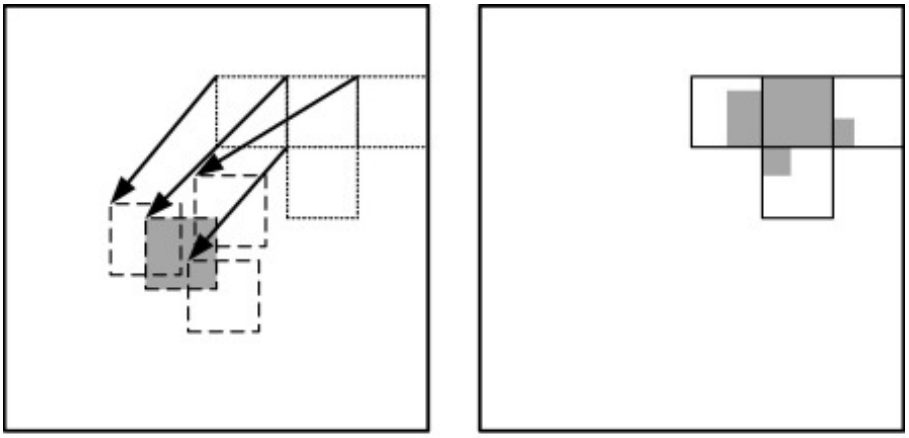
图11.4。单位错误对DCT/霍夫曼编码的Lena图像的影响。

下载：下载全尺寸图像

图11.5。错误率增加对DCT图像编解码器的影响。左：0.01% BER。中间：0.1% BER。右：1% BER。

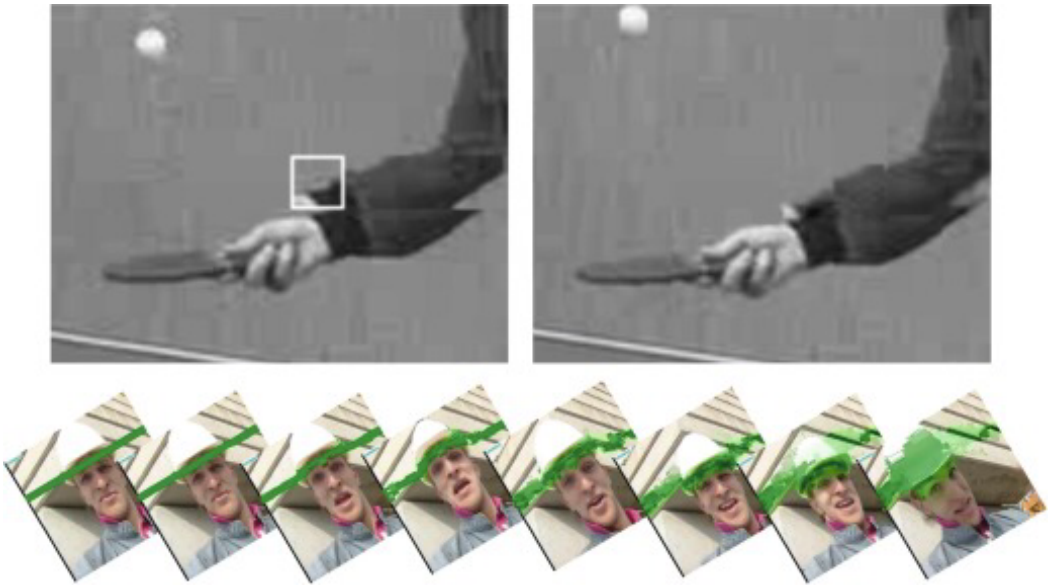
11.2.4。时间误差传播

当空间区域被腐蚀作为预测未来框架的基础时，就会发生时间误差传播。如图11.6所示，参考帧中的单个损坏块可以传播到预测当前帧中的多个（此处为4个）块。误差的空间传播取决于运动矢量的局部分布，而更活跃的区域往往比低运动区域更广地传播误差。图11.7显示了实践中两个时间传播的例子。在这里，由于空间数据丢失而在当地引入的错误仍然存在并蔓延，几帧后。



下载：下载全尺寸图像

图11.6。帧中单个损坏块的时间错误传播$k-1$帧中多个损坏的块k。



下载：下载全尺寸图像

图11.7。时间传播示例；顶部：横跨乒乓球序列的四帧。底部：横跨190帧工头。

应当指出，由于解码器的运动补偿不正确，损坏的运动矢量也可能提供错误源。

示例11.3由于预测而导致的误差的时间传播

考虑与示例11.2中给出的相同的符号编码和传输序列。假设这些符号表示与两个DFD块中的变换系数相关的{运行/大小}编码，并用作未来运动补偿预测的基础。评论位位置12中单位错误会导致的时间错误传播。

解决方案

与示例11.2一样，由于遗漏了EOB符号，因此块同步丢失，错误块后解码的所有块都将被替换。

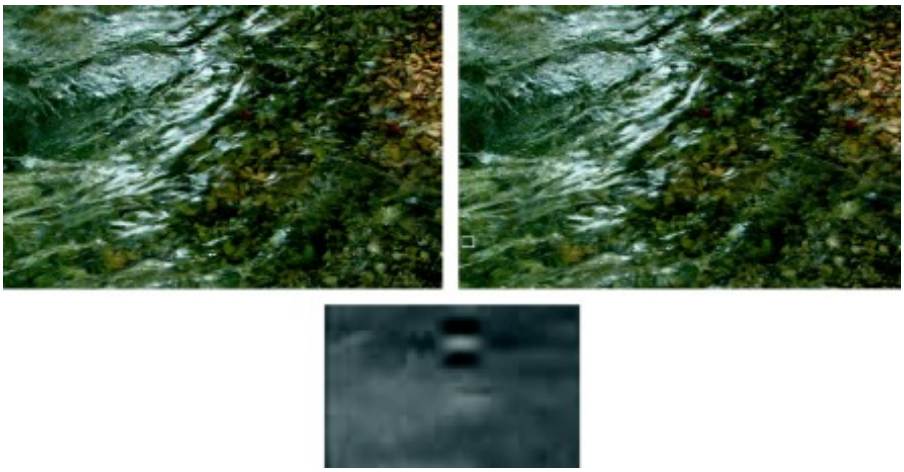
然而，在这种情况下，由于我们正在处理跨编码的DFD块，因此存在第二个问题。DFD块将用作解码器时间预测的基础，由于解码版本将包含空间错误，由于运动预测，这些块将暂时传播到后续帧。因为错误块之后的所有块都被移位，并且由于错误块发生在帧的早期，如果没有明确的重新同步点，那么未来预测帧中几乎所有的数据都可能被损坏。

11.3。减轻码流错误的影响

11.3.1.视频和数据不一样！

这看起来像一个奇怪的小节标题，但在某种程度上，这是真的。例如，如果我们考虑数字数据库的情况，那么为了保证其有效性，我们希望它能够完美传输。如果我们被告知它包含错误（特别是如果我们不知道它们在哪里），我们可能不会信任它，并避免使用它。然而，对于图像或视频序列，情况通常大不相同。

一方面，我们看到，由于错误传播，单个位错误可能具有高度破坏性。另一方面，我们知道人类视觉系统对某些类型的人工制品具有耐受性，如果处理正确，可以接受损失。事实上，压缩过程本身只是一个将噪声引入图像的托管过程，我们通常对原始内容的近似值非常满意。例如，考虑图11.8所示的视频帧。左边是原始压缩图像（使用金字塔矢量量化（PVQ）编码的30 Mb/s——见第11.7.2节），而右侧的图像在25%的*包错误率*（PER）的情况下重建（即所有数据包的1/4包含不校正的错误）。工件确实存在，如底部子图所示，但在这种情况下，它们屏蔽得很好，可以容忍。



下载：下载全尺寸图像

图11.8。主观上出现损失。PVQ编码河床序列为30 Mb/s。左：编码为30 Mb/s（内部）。右：每满25%后重建。底部：右上角图像所示的盒子放大的神器。

管理比特流错误的过程并不简单，因为它们不可预测。然而，我们可以模拟丢失对比特流的影响，并采取措施避免、纠正或隐藏它，这些方法将在以下章节中讨论。

11.3.2.抗错解决方案

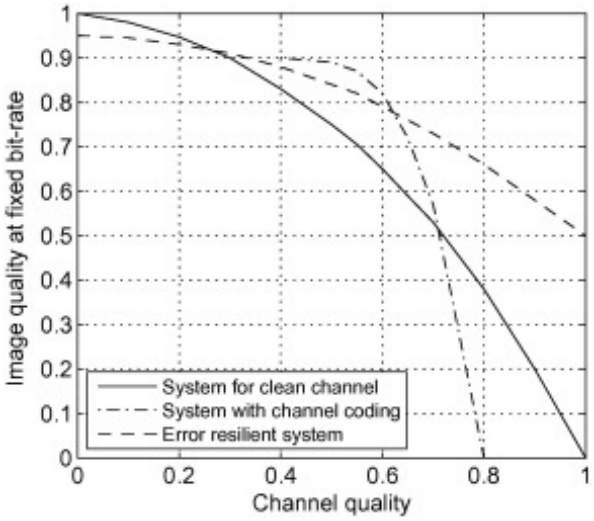
我们已经看到，我们从源头删除的冗余越多，我们生成的比特流就越脆弱。因此，我们需要在干净的渠道中交换一些速率失真性能，以提高存在错误时的误差恢复力。

让我们首先考虑通过不同损失程度的噪声通道传输编码图像的情况。表11.1比较了三种情况：（一）具有固定长度编码的仅DCT变换，（二）具有直流预测的基于DCT变换的系统，以及（iii）具有直流预测和基于霍夫曼的VLC的基于DCT的系统。可以看到，基于VLC的系统在干净的信道中提供了出色的编码性能，但随着信道条件的恶化，它严重下降。另一方面，FLC方案在干净的通道中性能不佳，但在有损通道中性能明显优于VLC情况。

表11.1。适用于各种信道条件的FLC与VLC编码。$\mathbf{256 \times 256}$ 莉娜图片为2bpp。

PSNR（分B）	仅限 DCT	DCT + DC pred。	DCT + DC pred。+ VLC
无错误	33.7	33.9	38.5
0.1% BER	24.6	21.0	6.4

清洁通道性能和抗错性之间的这种权衡是我们在设计系统时面临的挑战之一。理想情况下，我们希望在不影响干净通道质量的情况下获得最佳的错误恢复力。然而，在实践中，有必要做出妥协，牺牲一些干净的渠道性能，以提高在存在损失时的错误恢复力。这些特征如图11.9所示，以下各节介绍了以优雅退化实现这种误差恢复力的方法。



下载：下载全尺寸图像

图11.9。理想的性能比较。

11.4。传输层解决方案

在本节中，我们使用术语传输层（有点松散）来描述视频传输过程中与应用程序层（即视频编解码器）没有密切关联的所有方面。在许多情况下，误差控制机制将在ISO 7层模型的传输层实现。然而，它们也可以在MAC和PHY层中实现。在这个经过充分研究的主题上，有许多优秀的参考文献，例如参考文献。[3]，[4]，[5]，[6]。

11.4.1。自动重复请求（ARQ）

ARQ是一种交互式技术，广泛应用于许多通信系统。它依靠解码器的反馈来确定消息（数据包）是否正确接收。错误检测通常通过添加某种形式的*循环冗余检查*（CRC）或通过*前向错误更正*（FEC）编码来完成。当检测到错误时，解码器返回一个NACK（负ACKnowledge）信号，指示编码器重新传输数据。这种方法有很多明显的优点和问题：

优势

- ARQ简单高效。通过重传引入的冗余是有效的，因为它专门针对丢失的数据。
- 它可以有效地应对数据包丢失或大突发错误。

问题

- 当解码器等待数据重新传输时，会引入延迟，在实时视频应用程序中，这可能意味着必须删除帧（视频帧必须在特定时间间隔内呈现）。
- ARQ需要一个并不总是可用的反馈通道，例如在多播或广播时。

延迟受限的再传输

在实践中，ARQ是一种简单而有效的技术，可以确保数据可靠传输，但在实时视频的情况下，它具有有用的寿命，受编码和解码之间的最大可容忍延迟的限制。为了解决这个问题，通常使用*延迟约束的重传*，这可以在接收器和发射机上实现。在前一种情况下，解码器只会请求重新传输数据包n如果满足以下条件：

$$T_c + T_{RT} + T_x < T_d(n) \tag{11.1}$$

在哪里T_c是现在，T_{RT}是网络的往返延迟，$T_d(n)$分组的播放时间是n，以及T_x是一个变量，它解释了其他因素，如延迟估计的容忍度和其他不确定性。同样，如果满足以下条件，可以在编码器上做出此决定：

$$T_c + T_{RT}/2 + T_x < T'_d(n) \tag{11.2}$$

在哪里$T'_d(n)$在这种情况下，是数据包的估计播放时间。

11.4.2。FEC信道编码

在FEC方法中，额外的奇偶校验数据附加到压缩信号中，这使得解码器可以校正一定数量的错误。FEC可以与分层编码（或数据分区）相结合，以提供不等的错误保护，其中压缩比特流的不同部分根据重要性受到不同强度代码的保护。

使用FEC提高了传输所需的总数据速率，并明确抵消了从源压缩中获得的一些好处——对于1/2速率代码来说，冗余可以达到100%。此外，一般来说，选择FEC时必须考虑到特定最坏情况的渠道情况。对于质量变化很大的信道，这种最坏的情况可能意味着需要一个非常强大的FEC代码，它可以严重降低压缩性能。此外，每当超过这个最坏情况的阈值时，这样一个系统就会灾难性地失败。一种经常用于帮助减轻突发错误影响的方法是使用交织。然而，这是次优的，可能会造成巨大的延迟。

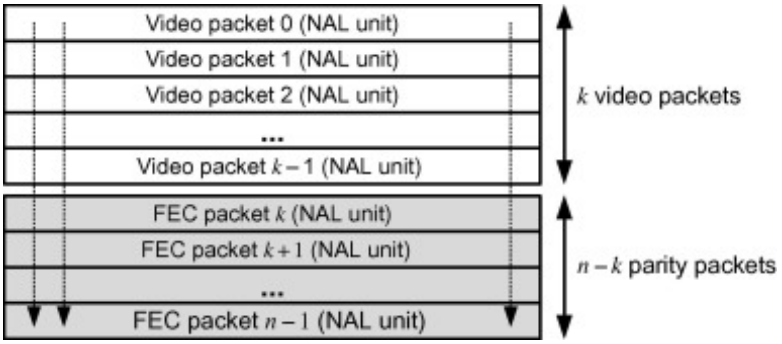
擦除代码

FEC在数据包删除的情况下非常有效，因为通常已知丢失数据包的位置。高效的擦除代码包括里德·所罗门擦除（RSE）校正代码。使用RSE（n,k）代码，k数据包用于构造r奇偶校验数据包，其中$r = n - k$，导致总共n待传输的数据包。此k 源数据包可以从任何k出自n传输数据包。这为最多提供了无错误的解码r丢失的包裹n。选择值的主要考虑因素r和k是：

- **编码/解码延迟：**在数据包丢失的情况下，解码器必须至少等待k在解码之前，已经收到了数据包。因此，为了尽量减少解码延迟，k不能太大。
- **鲁棒到爆亏：**一个更高值的k这意味着，对于相同数量的冗余，FEC将能够更正更多连续丢失的数据包或突发。例如，RSE(4,2)代码和RSE(6,4)代码都可以防止突发错误长度为2，但RSE(6,4)只有50%的开销，而RSE(4,2)的开销为100%。

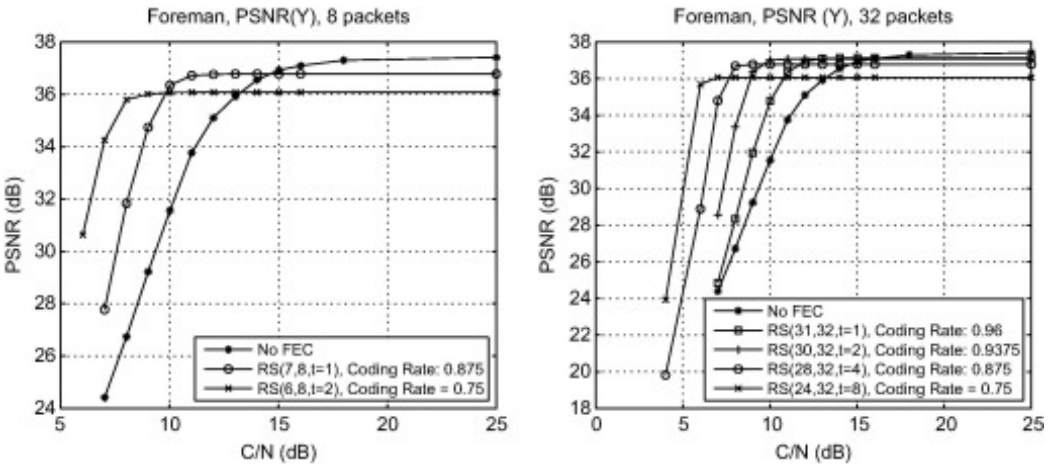
交叉分组FEC

如果FEC应用于数据包或附加到单个数据包中，如果数据包不仅错误，而且在传输过程中完全丢失（例如通过拥堵的互联网连接），则代码的校正能力将丢失。相反，如图11.10所示，将FEC应用于多个数据包是有益的。使用FEC的一个问题是所有k数据包需要相同长度，如果不允许GOB碎片化，可能会有问题。不同编码深度（8和32）情况下交叉分组FEC的性能如图11.11所示。这清楚地表明了清洁信道性能和不同编码速率的错误恢复力之间的妥协。



下载：下载全尺寸图像

图11.10。交叉数据包FEC。



下载：下载全尺寸图像

图11.11。不同编码速率的编码深度为8（左）和32（右）的交叉分组FEC性能。

不等式错误保护和数据分区

我们在第11.2节中看到，并非所有类型的视频数据都同样重要。例如，标题信息通常最为重要，因为没有它，其他信息就不可能。运动信息扮演着特别重要的角色，特别是因为它占总比特率的相对较小（约10%）。编码序列的忠实重建在很大程度上取决于运动矢量的准确性。为了反映这些优先级，可以将视频数据划分为单独的数据段，每个数据段都分配了与其重要性相适应的保护级别。

数据优先级也可以与第11.9节所述的分层编码一起实现。

无税率代码

Shokrollahi[7]提出的猛禽代码作为多媒体信息信道编码的有效手段，已获得显著的牵引力。猛禽代码是一种喷泉代码（最初由Luby[8]作为LT代码引入），它结合了LDPC和LT代码，并提供线性属性，$O(k)$，时间编码和解码。喷泉代码编码k符号被编码符号的（潜在）无限序列，消息可以被恢复的概率随着收到的符号数量而增加。比如猛禽码，一次解码成功的几率是99%k已经收到了符号。

猛禽代码是第三代合作伙伴项目（3GPP）的一部分，用于移动蜂窝多播。它们为处理可变无线信道条件提供了一个灵活和适应性的机制。

11.4.3。混合ARQ（HARQ）

HARQ[9]将FEC编码与ARQ相结合。它受到ARQ的一些基本限制，但提供了一些效率收益。在HARQ中，数据使用适当的FEC代码编码，但奇偶校验位不会与数据一起自动发送。只有当解码器检测到错误时，才会传输这些额外的奇偶校验位。如果FEC代码的强度足以纠正错误，则不再采取进一步行动。然而，如果情况并非如此，则系统将恢复到完整的ARQ重传。通常，系统会在重传期间在数据和FEC数据包之间交替。HARQ是常规ARQ和常规FEC之间的妥协。它在干净的通道中与ARQ一样运行，在有损通道中与FEC一样好。

在实践中，通过同一数据的多次重传，接收器将存储所有传输，并组合使用这些多个副本。这通常被称为*软组合*。这种方法已应用于UMTS 3G标准、WIMAX（宽带无线接入-IEEE 802.16）和4G LTE的高速数据上下行链路。

11.4.4。打包策略

分组策略[10]在存在损失时会对性能产生影响。基本原则可以概括为：

- 较长的数据包可以提高清洁通道的吞吐量（较低的开销）。
- 较短的数据包有助于更好的错误恢复力，因为它们的丢失对视频序列的影响较小，错误隐藏将更容易。
- 为了支持错误恢复，应避免视频数据包碎片化。因此，每个NAL单元（见第11.5.1节）应在一个且只有一个UDP帧中传输。

为了说明第三个要点，如果包含编码的VLC符号的视频数据包被分割到两个UDP数据包中，那么其中一个数据包的丢失将使另一个由于VLC依赖性而变得毫无用处。

11.5。应用层解决方案

11.5.1。网络抽象

自H.264/AVC以来的标准采用了网络抽象层（NAL）的原则，该原理提供了适用于广泛网络的通用语法。高级参数集还用于将与多个切片相关的信息从媒体比特流解耦。例如，H.264/AVC既描述了序列参数集（SPS）和图像参数集（PPS）。SPS适用于整个序列，PPS适用于整个帧。这些描述了帧大小、编码模式和切片结构等参数。关于H.264等标准的结构的更多详细信息，请参阅第12章。

11.5.2。框架类型的影响

I型框架、P型框架和B型框架

所用框架类型直接影响误差的时间传播方式。I型机架、P型机架、B型机的主要特点如下：

- **I-帧**：I-帧不参考任何其他帧进行预测。因此，它们为时间误差的传播提供了障碍。然而，在H.264/AVC和HEVC中，它们确实利用了内部预测，因此可以在空间上传播错误。I-帧也是预测P-和B-帧的基础，因此I-帧中发生的任何空间错误都可以临时传播到后续帧。
- **P型帧**：P型帧从其他P型帧或I型帧中预测。因此，它们既会传播其参考框架中包含的错误，又引入新的空间错误，这些错误可以在空间和时间上传播。
- **B帧**：通常不使用B帧作为预测其他帧的基础。因此，它们不会暂时传播误差，B帧中引入的任何空间误差都受该帧的限制。

刷新内

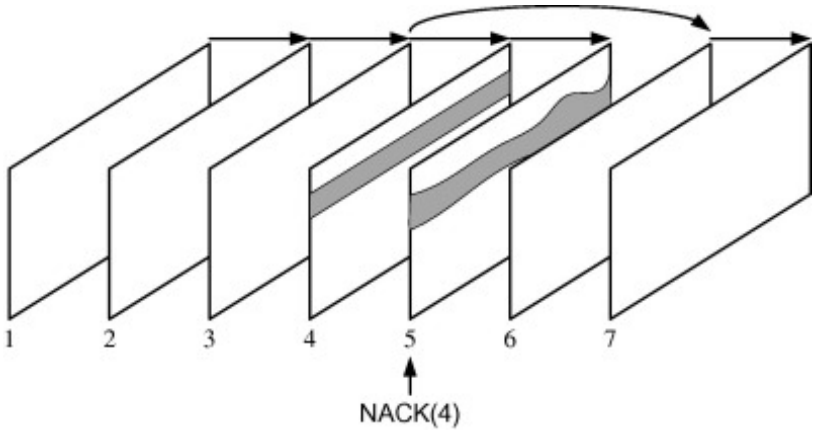
我们看到，视频序列的图片组（GOP）结构定义了I-、P-和B-帧之间的关系。所有编码序列都将至少有一个I帧，大多数将定期插入它们，例如每12或15帧。许多编解码器支持P帧中宏块的内部编码。这为速率-失真优化提供了更灵活的方法，并有助于限制时间误差的传播。以这种方式内部编码的宏块可以根据速率-失真标准，或根据其隐藏困难或随机选择。

一些编解码器，如X.264，更进一步，支持*定期刷新*而不是关键帧。这有效地将I-frame扩展到多个帧之间，使用一列内部编码块，这些块逐步跨过视频帧。在这种情况下，运动矢量以与切片结构相同的方式受到限制，因此刷新列一侧的块不引用另一边的块。这种方法的另一个好处是峰值位元比率受到限制（与孤立使用大型I帧的情况不同），提供更好的流和更低的延迟。

参考图片选择

在简单的运动补偿混合编解码器中，最近以前的编码帧被用作预测当前帧的参考。更近期的标准（从H.263+开始）支持使用多个参考帧和参考图片选择（RPS）模式。这些允许在一些约束下选择任何以前编码的帧作为引用。这有利于限制错误多发环境中的错误传播。

在参考图片选择（RPS）中，编码器和解码器都存储多个参考帧。如果通过解码器的反馈，指示编码器错误地接收了最新的参考帧，那么它将切换到另一个（无错误）参考图片，以进行未来的预测。RPS选择模式最初在H.263+中作为附件N采用。如图11.12所示，其中NACK（4）信号从解码器反馈，并在编码第5帧时接收。对于第6帧，编码器切换到使用第3帧，该帧接收无错误，作为新的参考帧。

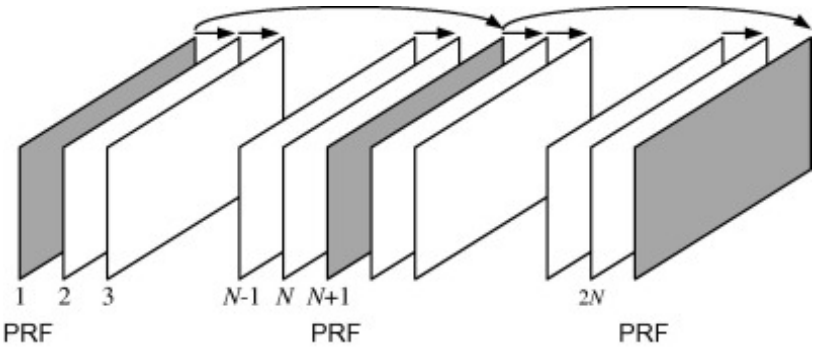


下载：下载全尺寸图像

图11.12。基于参考图片选择的错误恢复。

周期性参考框架

就编码效率而言，通常比使用帧内编码更有效地从帧中预测几帧移出的图像。Periodic RPS[11]等技术通过编码每个Nth序列帧使用Nth前一帧作为参考。所有其他帧都像往常一样编码。该方案如图11.13所示。其优点是，如果在下一个PR帧引用之前，可以通过使用ARQ或FEC来纠正PR帧中的任何错误，那么这将有效地将最大时间误差传播限制在PR帧之间的帧数。



下载：下载全尺寸图像

图11.13。定期参考框架。

由于PR帧编码所需的比特远远小于相同质量的帧内，因此可以使用额外的比特来保护PR帧，从

而在数据包丢失下比帧内编码提供更好的性能。由于该技术不依赖于反馈，因此适用于广播和组播场景。这种方法现在在编码标准中很常见，例如在H.264/AVC中，称为*带子序列的交叉预测*。

11.5.3。同步码字

通过插入显式同步码字，可以实现比特流及其关联符号和数据结构的周期性重新同步。例如，这些可能会插入到每行块、切片或帧的末尾。重同步码字是唯一可识别的，与所有其他码或代码串联不同。它们具有以下优点和缺点：

优势

如果遇到错误，最大损失受同步标记之间的距离限制。一旦遇到新的同步标记，解码可以正确地再次进行。

缺点

重新同步码字会产生相当大的开销，特别是如果经常使用。对于JPEG编码的图像，每行末尾的重新同步标记通常会产生1-2%的开销，而如果它们位于每个块之后，它们通常会产生超过30%的开销。

11.5.4。可逆VLC

当常规VLC中遇到错误时，误差极有可能在空间传播，直到下一个重同步点。在*可逆VLC*（RVLC）的情况下，可以进行向后解码和正向解码。因此，常规VLC丢失的一些数据可以用RVLC忠实解码。RVLC方法最初在H.263+和MPEG-4中受到青睐，可以与同步标记有效结合使用，以增强错误恢复力。它们由高岛等人介绍[12]，他们展示了如何生成对称和非对称代码。

RVLC代码[12]，[13]应该同时显示前缀和后缀属性，以便码字可以双向解码。可以提供这种所谓的双前缀属性的代码类型是有限的，很少是最优的。Golomb-Rice代码或Exp-Golomb代码等代码可以映射到同样有效的双前缀代码中，但在大多数情况下，产生的双前缀代码效率较低。Girod[14] 提出了一种基于前缀码词的修改，生成双向可解码 VLC 比特流的替代方法。这种优雅的方法产生了相当于霍夫曼编码的编码效率。

众所周知，由DCT系数或微分运动矢量组成的VLC数据表现出*广义高斯分布*。如第7章所述，这些可以使用Exp-Golomb代码进行有效的熵编码。这些无需查找表即可实现，可逆版本易于设计。Exp-Golomb代码是H.264和HEVC标准的重要组成部分。

示例11.4提供了RVLC的简单示例。
示例11.4可逆VLC

考虑符号-码字映射：

$$a \leftrightarrow 0; \quad b \leftrightarrow 11; \quad c \leftrightarrow 101 = \mathrm{EOB}$$

序列$\{a,b,c,b,a,a,b,c,b,a\}$传输后跟一个SYNC码字。如果$9\mathrm{th}$和$11\mathrm{th}$位被损坏，演示此编码如何通过双向解码将错误最小化。

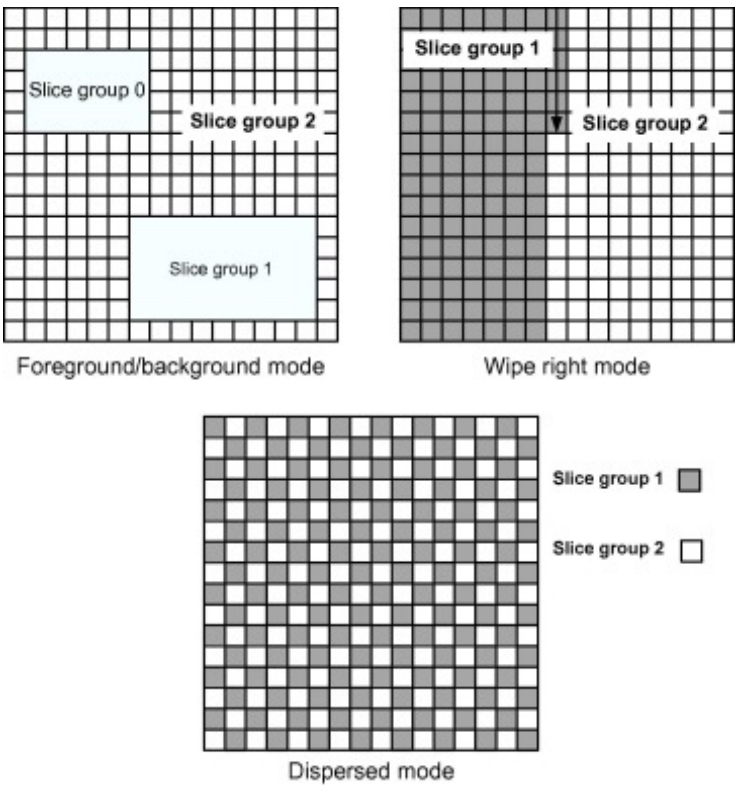
解决方案

信息	***a***	***b***	***c***	***b***	***a***	***a***	***b***	***c***	***b***	***a***	**SYNC**
Rx. bitstream	0	11	101	11	**1**	0	**01**	101	11	0	SYNC
Fwd.解码	a	b	c	b	–	–	X	X	X	X	SYNC
修订解码	X	X	X	X	X	X	–	c	b	a	SYNC

请注意，码字是对称的，这给了我们一个线索，即它们应该是双向可解码的。该表显示了正向和反向解码操作。通过双向解码，我们成功地恢复了大多数符号，只丢失了发生错误的三个符号。如果我们假设那个码字***c***表示一个EOB，然后我们可以看到RVLC方法恢复传输的三块数据中的两个。

11.5.5。切片结构

最新标准支持在编码前将帧分解为切片的想法。切片由切片头和整数个宏块或编码单元组成。由于所有形式的预测都限制在切片边界内，切片结构有效地限制了误差传播。因此，它为抗错编码提供了简单的基础。一些典型的通用切片结构如图11.14所示。



下载：下载全尺寸图像

图11.14。切片结构示例。

灵活的宏块排序（FMO）

灵活的*宏块*排序（FMO）是一种切片结构方法，对误差恢复力和支持隐藏特别有吸引力。它是H.264/AVC标准的一部分。视频帧分为几个可独立解码的切片组（图11.14），这些组可以假设棋盘、分散和交错模式，如图所示。例如，在棋盘模式中，每个帧可以表示为两个切片组，宏块相互交错，如图所示。很容易看出，如果一组中的一片丢失，那么剩下的一片可以用于为缺失切片的插值提供依据。如果两个片中只丢失一个，那么FMO就会将错误分布在帧的更大区域。

切片结构的一个缺点是编码增益减少。在FMO中尤其如此，因为相邻的宏块不能用作预测的基础。据报道，QP = 16时效率下降或比特率增长约为5%，QP = 28[15]时效率下降或比特率增长高达20%。

使用分散模式切片组的例子如图11.15所示。在这里，每个切片对应两行宏块，图显示了切片组之一丢失切片2和3的情况。可以清楚地看到，来自剩余切片组的数据的分布方式是为了方便对丢失的数据进行插值。这是第11.8.4节中提出的错误隐藏方法的基础。

下载：下载全尺寸图像

图11.15。使用分散模式切片组作为隐藏错误的基础。

多余的切片

冗余切片（RS）是H.264/AVC等标准的特征，该功能支持为防错性目的包含冗余信息。冗余切片可以是原始切片的简单副本，也可以是保真度较低的副本。如果丢失了，此副本可以代替原件。问题是——哪些切片应该作为冗余包含？一些作者已经谈到了这个问题。例如，Ferre等人[16]提出了一种方法，根据端到端失真模型选择宏块进行包含，该模型旨在最大限度地减少每个冗余位的失真。这比交叉数据包FEC和损失自适应速率-失真优化等替代方法更具优势。

11.5.6。错误跟踪

该技术还依赖于反馈通道的存在，不仅用于指示发生了错误，还用于提供有关错误位置的信息（例如哪些切片或块已损坏）。如果编码器知道哪些块在错误中接收，它可以预测错误将如何传播到当前编码的帧；然后它可以采取以下两个操作之一：

1. 在模式内编码相应的块。

2. 仅根据正确接收的块的预测对相应块进行编码。

这种所谓的*选择性恢复*技术是由和田提出的[17]。Steinbach等人在参考文献中提出了一种有效的错误跟踪算法。[18]。

11.6。跨层解决方案

我们看到，压缩视频（在人类视觉系统中）可以容忍一定程度的错误，前提是相关工件得到有效管理。然而，几乎所有网络的目的都是提供完美的数据；任何其他内容都被视为无用，通常被较低的网络层丢弃，而不发送给应用程序。传统上在OSI堆栈（物理（PHY）、中型访问控制（MAC）和传输）的下层实现的资源管理和保护机制故意与应用程序层隔离。然而，我们已经看到了这些较低层的错误特性如何与它们在应用程序层的表现紧密耦合。

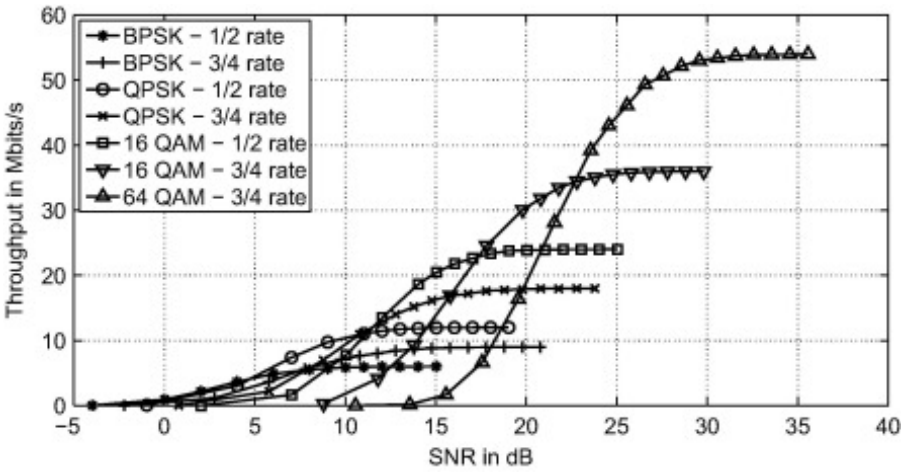
因此，最近人们对跨通信层进行优化非常感兴趣，以便应用程序的需求可以反映在网络层的参数化中，反之亦然。这被称为*跨层优化*。跨层优化不同于*联合源通道编码*，后者的目的是根据通道的速率约束优化通道编码。然而，前者采取了更全面的方法，同时考虑到分组策略、重传策略、延迟约束和正在编码数据的损失容忍度。

范德沙尔等人对交叉层优化的各个方面进行了出色的概述。[4]，[19]。

11.6.1。链接适应

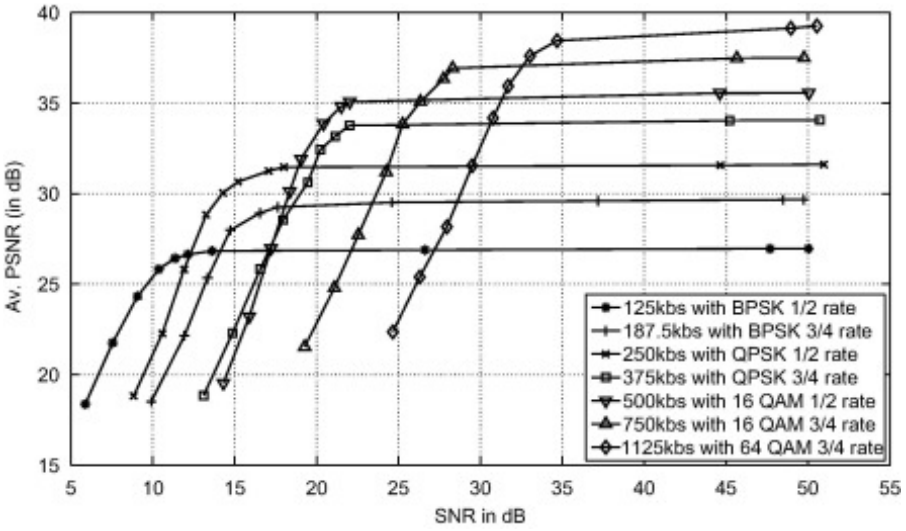
一种在实践中可以实现的跨层优化形式是基于端到端失真度量和建模的链接自适应。IEEE 802.11a/g/n等无线局域网（WLAN）支持许多调制和编码方案（MCS），每个方案都提供不同的吞吐量和可靠性水平。传统的MCS适应算法试图基于射频信号测量，如剩余信号强度指示（RSSI）和比特或分组错误率（BER或PER）来最大化无错误吞吐量。它们不考虑数据流的内容，严重依赖ARQ和FEC来纠正错误的传输。相比之下，Ferre等人[20]等作者提出了通过尽量减少接收序列的视频失真来实现这一目标的方法。

Ferre等人在编码器上使用简单的本地速率失真度量和端到端失真模型来估计当前传输速率以及相邻的低链路和高链路速度（MCS模式）接收的视频失真。这允许系统选择失真最低的模式，适应频道条件，以提供最佳视频质量。基于H.264/AVC over IEEE 802.11g的仿真结果表明，该系统紧随最优理论解。示例结果见图11.16，图11.17，图11.18，图11.19。前三个数字说明了基于吞吐量的度量和基于视频质量的度量之间的切换特性存在显著差异。图11.19显示了根据信道载波与噪声比绘制的*工头*序列的度量的结果。可以清楚地看到，开关特性密切跟踪MCS模式之间的最佳轨迹。



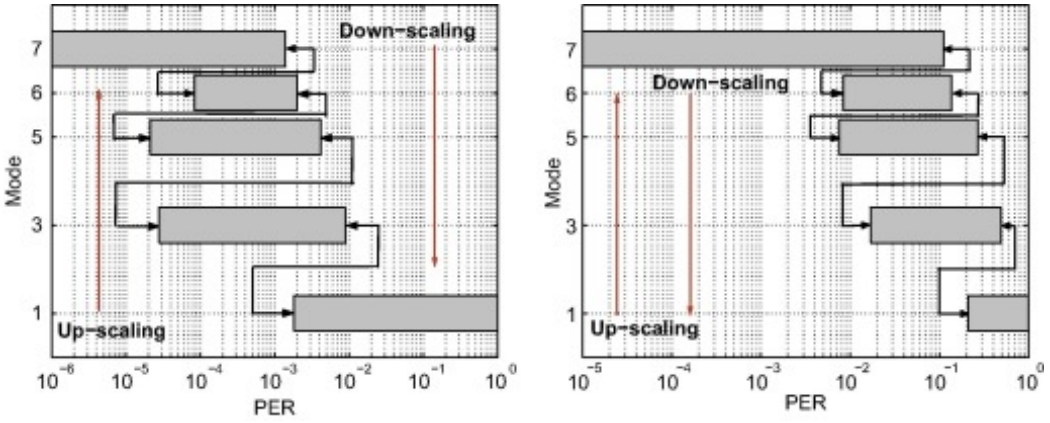
下载：下载全尺寸图像

图11.16。基于吞吐量的802.11g单片机切换特性。（经参考文献许可转载。[20]）。



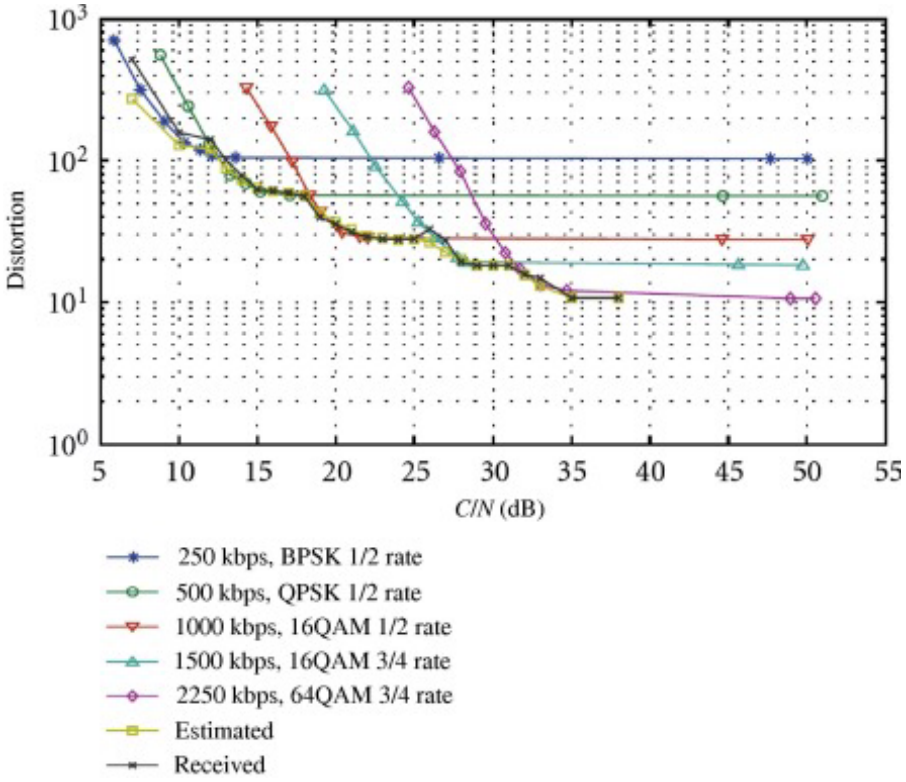
下载：下载全尺寸图像

图11.17。基于视频失真的802.11g单片机切换特性（经参考文献许可转载。[20]）。



下载：下载全尺寸图像

图11.18。MCS切换特性比较：基于视频质量（左）和基于吞吐量（右）。（经参考文献许可转载。[20]）。



下载：下载全尺寸图像

图11.19。MCS链接适应结果。（经参考文献许可转载。[20]）。

11.7。固有的健壮编码策略

我们在上面看到，压缩视频中错误传播的原因之一是可变长度熵编码。在本节中，我们考虑了两

种方法，它们提供了固定长度编码（FLC）的好处，同时保留了VLC提供的压缩性能。第一种是*抗错熵编码*（EREC），是一种伪固定长度方案，通过码流重组引入固有同步。第二种，金字塔*矢量量化*（PVQ），是一个真正的固定长度方案，它利用子带统计来创建一个支持FLC的模型。

11.7.1。抗错熵编码（EREC）

操作原则

弹性误差熵代码（EREC）[21]，[22]是一种简单而优雅的数据块编码方法。它由雷德米尔和金斯伯里[21]引入。基于一种形式的垃圾箱打包，使用固定长度的开槽结构，它允许引入隐式同步点，而无需显式同步码字的开销。该算法将可变长度编码的数据块（例如，基于块的转换和熵编码产生的数据块）重新排列成一个具有预定义的维度的结构，以确保它有足够的容量。

假设有N可变长度块，每个长度$b_i : i = 1 \cdots N$我们希望将这些打包进去M长度的插槽$s_j : j = 1 \cdots M$。因此，开槽的EREC数据结构应满足以下条件：

$$\begin{cases} N \geq M \\ T = \sum_{j=1}^{M} s_i \geq \sum_{i=1}^{N} b_i \end{cases} \tag{11.3}$$

在哪里T是插槽结构的总数据大小。第一种条件确保有足够的插槽，而第二种条件确保EREC结构足够大，可以对所有可变长度的数据块进行编码。在大多数情况下，对于图像或视频编码，设置是合适的$N = M$从现在开始，我们将承担这个。因此，我们可以计算方程（11.4）中给出的结构的槽长：

$$s_i = \left\lceil \frac{1}{N} \sum_{i=1}^{N} b_i \right\rceil \tag{11.4}$$

安N然后使用阶段算法将块打包到槽状结构中。为了实现可以在解码器上反转的系统编码过程，使用偏移序列。这定义了算法中每个阶段使用的偏移量。在舞台上n，算法将搜索插槽$k = i + \phi_n (\mathrm{mod}(N))$，在哪里$\phi$是一个伪随机偏移序列。在我们的示例中，我们将简单地使用序列：$\phi = \{0,1,2,3,4,\ldots,N\}$。在过程的每个阶段，EREC算法在适当的偏移量处搜索插槽，看看是否有空间打包部分或全部多余的

算法11.1 EREC编码

1. Define offset sequence ϕ_n;
2. INPUT N blocks of data;
3. Compute slot length, s_i, using equation (11.4);
4. Assign block b_i to slot s_i;
5. FOR $n = 1 \cdots N$
   6. Compute residual data for each slot: $r_i = b_i - s_i$;
   7. FOR $i = 1 \cdots N$
      8. Compute offset: $k = i + \phi_n (\text{mod}(N))$;
      9. IF $r_i > 0$ AND $r_k < 0$ THEN pack slot k with residual from block i;
         IF $|r_k| \geq |r_i|$ THEN $r_k \leftarrow r_k + r_i$; $r_i \leftarrow 0$;
         IF $|r_k| < |r_i|$ THEN $r_i \leftarrow r_k + r_i$; $r_k \leftarrow 0$;
   10. END FOR;
   11. IF $\{r_i = 0 : i = 1 \cdots N\}$ THEN END;
12. END FOR.

下载：下载全尺寸图像

来自超长块的数据——即在哪里$b_i > s_i$。这个过程重复，直到来自超长块的所有多余数据被打包到结构中。EREC编码过程在算法11.1中定义。

解码过程只是逆操作，逐步重建，超过N解码阶段，每个原始块，直到块被完全解码（例如，检测到块末代码）。因此，在没有信道错误的情况下，解码器将正确解码所有数据。

图11.20显示了六个数据块的简单EREC编码示例。参考图11.20：在0阶段，第3、4和6块完全编码在插槽3、4和6中，剩余空间。然而，第1、2和第5块只有部分编码，数据还剩下放在插槽3、4和6的剩余空间中。在第一阶段，对于$\phi = 1$，来自第2块的剩余数据在插槽3中编码，来自第5块的一些数据在插槽6中编码。这个过程一直持续到第六阶段结束时，所有数据都编码了。示例11.5，示例11.6给出了进一步的工作示例，说明了特定块集的编码和解码过程。

示例11.5 EREC编码

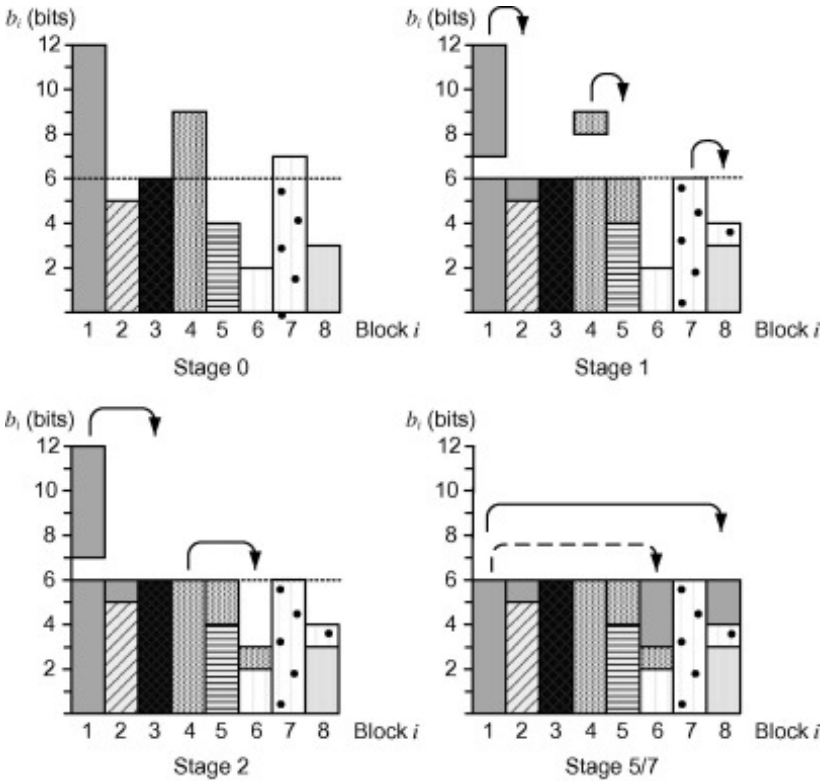
基于DCT/VLC的编码器生成了一个由以下长度的八个块组成的数据片段：

$Block\ i$	1	2	3	4	5	6	7	8
长度b_i	12	5	6	9	4	2	7	3

假设偏移序列$\{0,1,2,3,4,5,6,7\}$，显示EREC算法将如何对这段数据进行编码。

解决方案

切片的总长度为48位，因此插槽长度为6位。编码过程如下：



下载：下载全尺寸图像

编码过程的第一阶段和第二阶段如上所示。然后，对于第3和第4阶段，没有多余的空间来填补第4和第5个空缺，因此没有采取任何行动。在第5阶段，来自1块的3位被打包到插槽6中，在第7阶段，来自该块的其余2位被打包到插槽8中。为了方便起见，上述两个操作合并显示。

示例11.6 EREC解码

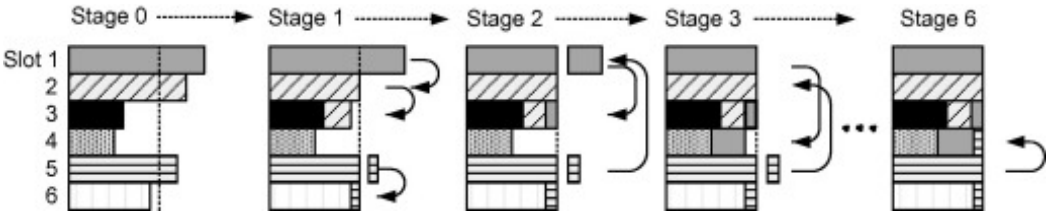
对上面示例11.5中EREC编码的图像数据块执行EREC解码。假设每个块都以EOB符号结束。

解决方案

为了方便起见，我们在这里使用更紧凑的表格结构来说明解码过程的每个阶段，其中**X**表示不完整的块，**C**表示完全解码的块。

舞台	偏移	bk1	bk2	bk3	bk4	bk5	bk6	bk7	bk8
0	0	X	C	C	X	C	C	X	C
1	1	X			X			C	
2	2	X			C				
3	3	X							
4	4	X							
5	5	X							
6	6	C							

在第 0 阶段，我们从第一个块的底部开始解码，直到到达插槽的末尾或解码 EOB 符号。在第 1 块的情况下，我们到达插槽的末端，没有检测到 EOB。我们为所有其他插槽做同样的事情。因此，第2、3、5、6和8块在第0阶段都完全解码了。在第一阶段，我们搜索（所有不完整的块）以偏移量为1的其他数据。例如，在块1的情况下，我们搜索插槽2：我们检测到块2的EOB以外的其他数据，但仍然没有检测到1块的EOB，因此它仍然不完整。这一直持续到第6阶段，在这种情况下，所有块都完全解码了。



下载：下载全尺寸图像

图11.20。EREC操作。

EREC 性能

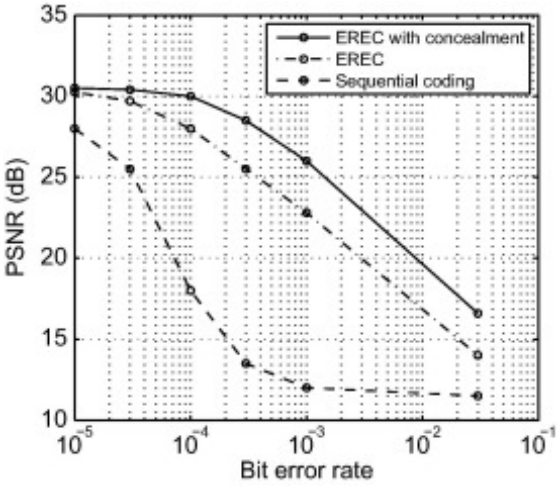
在干净的通道中，EREC解码器将正确解码所有数据，开销微不足道（通常仅T needs to be sent as side information). When channel errors occur, their effect depends on the particular distribution of codewords in the block and how rapidly symbol resynchronization is achieved. If symbol resynchronization is not regained before the end of the block, then this can cause the EOB symbols to be missed or falsely detected. This means that the decoder will incorrectly decode following information which was packed at later stages of the algorithm. This, in turn, implies that data placed in later stages (that toward the ends of long blocks) are more susceptible to error propagation than data placed in earlier stages. For typical compression methods, these propagation effects occur in high frequency data from active regions of the image. The important property of EREC is that, because the decoding of each block commences from a known location, block synchronization is preserved.

图11.21显示了0.1% BER和1% BER的EREC编码后重建图像的例子。将这些与图11.5中的等价物进行比较，可以清楚地看到EREC编码的鲁棒性。如果进行隐藏，主观质量可以进一步提高（见第11.8节）。图11.22显示了EREC与常规顺序编码的相对性能示例，显示了在噪声信道中高达10分贝的改进。

下载：下载全尺寸图像

图11.21。错误传输后重建DCT/霍夫曼编码图像的EREC性能示例。左：0.1% BER。右：1% BER。

由D提供。雷德米尔。



下载：下载全尺寸图像

图11.22。与常规DCT/VLC编码相比，EREC性能的增加BER。

由D提供。雷德米尔。

Redmill和Bull[22]已经表明，EREC在算术编码系统和传统的VLC方法中可以很好地运行。有趣的是，论文证明，如果没有EREC，霍夫曼编码比算术编码对错误更健壮。然而，对于EREC，这两种方法都有显著的改进，其性能相似，PSNR的改进高达15分贝。

EREC已被考虑在MPEG-4等标准中采用，但从未被纳入最终草案。可能的原因是，它需要处理一些块才能打包，尽管这些延迟可以在切片结构环境中管理。更可能的原因是，这种方法不适合常规加工管道。此外，就视频而言，使用预测编码来消除帧间冗余意味着必须采用另一种方法来编码运动矢量。斯旺和金斯伯里在参考文献[23]中展示了EREC对视频编码的益处，比MPEG-2中的常规方法有显著好处。

11.7.2。金字塔矢量量化（PVQ）

操作原则

EREC等技术通过重组比特流来增强弹性，从而实现伪固定长度编码。相比之下，金字塔矢量量化（PVQ）通过使用实际固定长度的码字来防止错误传播。菲舍尔[24]引入的PVQ利用变换系数的拉普拉斯概率分布，以提供压缩系数的有效方法。

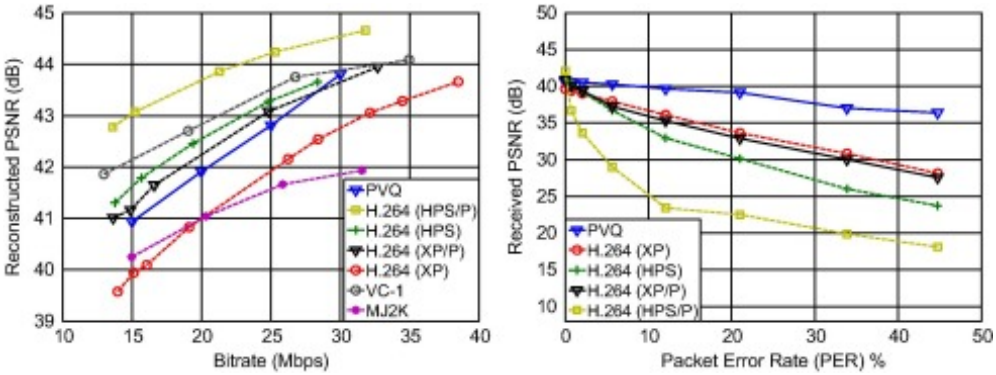
PVQ是针对i.i.d优化的一种矢量量化形式。拉普拉斯随机变量。向量$\mathbf{x}$通过分组形成L此类随机变量（例如变换系数）。基于大数定律和渐近均分性质，可以证明几乎所有向量都位于等高线或恒定概率区域上。这个区域是超金字塔L- 维度空间与$2L$顶点和$2L$面孔。因此，对于足够大的矢量维度，如果$\mathbf{x}$近似于其在超金字塔上的最近点，则失真可以忽略不计。这是PVQ编码算法的基础。

PVQ编码算法由两个步骤组成。第一个找到最近的向量$\hat{\mathbf{x}}$到输入向量$\mathbf{x}$。第二步生成位代码$\hat{\mathbf{x}}$。由于超金字塔上的所有点都同样可能，FLC是分配位码的最佳方法。因此，PVQ编码的比特流包括一系列代码簿矢量索引。

Chung-How和Bull[25]表明，PVQ比JPEG或H.263更能抵御比特错误，并且基于小波的PVQ在压缩性能方面优于基于DCT的PVQ。在参考文献中。[26] 博哈里等人表明，模内PVQ是室内无线传输高清视频的有效编解码器。由于帧间编码可能导致时间错误传播，因此使用了内部编码。虽然由于使用内部模式，压缩性能会下降，但错误恢复力的提高证明了这一点。

表演

Bokhari等人[26]为PVQ引入了一种高效的速率失真（RD）算法，该算法具有令人印象深刻的压缩性能，可与H.264/AVC和Motion JPEG 2000相当。他们还使用现实的有损无线信道评估了编解码器在高清视频背景下的错误恢复力。在有损条件下，与H.264/AVC相比，PSNR有高达11分贝的改善。他们方法的基本压缩性能如图11.23所示，用于网球序列。可以看出，PVQ内部提供压缩性能和视频质量（43 dB PSNR @25 Mbps）与其他编解码器在模式内相当。重要的是，它还避免了对复杂工具的需求，如内部预测、可变大小变换、解块过滤器或熵编码（CABAC）。



下载：下载全尺寸图像

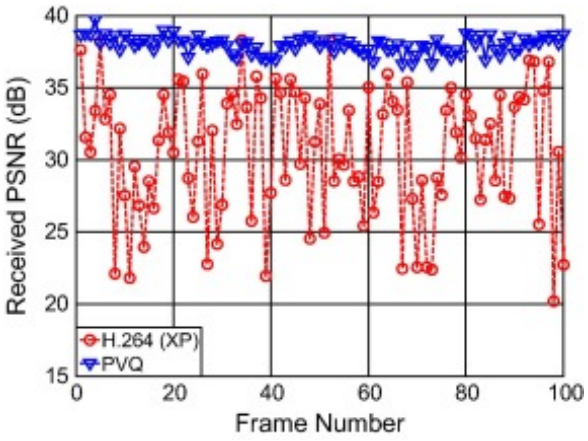
图11.23。RD优化了网球的PVQ性能。左：在干净的通道中进行RD性能。右：在有损802.11n通道中具有相关错误的性能（使用权限转载）。[26]）。

然而，PVQ的关键方面是其错误恢复力。这被描述并与参考文献[26]中的其他标准化编解码器进行比较，结果也见图11.23的右图。在误差恢复力方面，PVQ比性能最好的H.264/AVC对应方提供高达13.3分贝的PSNR性能增益。将PVQ编码帧示例与图11.24中的H.264/AVC（高配置）进行比较。

下载：下载全尺寸图像

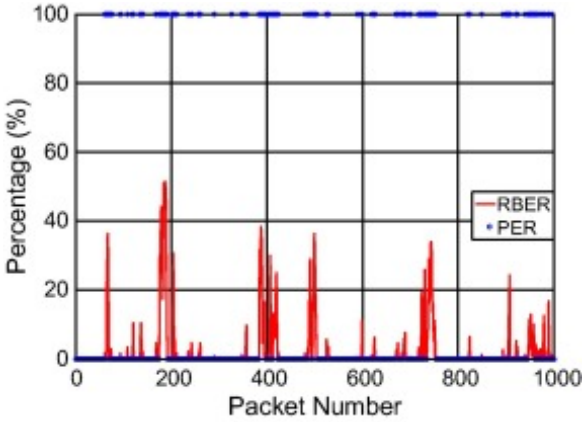
图11.24。网球的帧数为25 Mb/s和25% BER。顶部：H.264/AVC（HPS）。底部：RD优化PVQ（经参考文献许可转载）。[26]）。

与H.264/AVC相比，PVQ的一个主要属性是损坏帧之间的失真低方差。图11.25说明了这一点。影响PVQ性能的因素之一是它处理损坏数据的能力。在大多数无线场景中，整个数据包没有丢失，而是没有足够的完整性从FEC中受益。在这种情况下，数据包通常在发送到应用程序层之前被丢弃。在PVQ和其他健壮格式中，编解码器受益于使用损坏的数据。对于基于VLC的方案，限制因子是PER，而对于基于FLC的方案，它是（在接收的数据包中）的实际比特错误数。这被定义为剩余比特错误率（RBER）。图11.26显示了这一点，该图表明，尽管PER可能很高，但相应的RBER可以低得多。图为样本包错误跟踪及其对应的RBER跟踪。在突发或相关通道中，瞬时RBER 差异很大。这会导致暂时局部的心理视觉影响。在不相关的通道中，瞬时RBER变化要小得多，在整个序列中质量损失更一致。



下载：下载全尺寸图像

图11.25。PVQ和H.264/AVC的帧失真变化（经参考文献许可转载）。[26]）。



下载：下载全尺寸图像

图11.26。802.11n通道的分组错误率与剩余比特错误率（经参考文献许可转载）。[26]）。

11.8。错误隐藏

错误隐藏是一种面向解码器的后处理技术，试图通过提供主观上可接受的原始数据的近似值来隐藏错误的影响。这利用了视频序列中的时空相关性来估计以前收到的数据中丢失的信息。使用空间或时间插值重建丢失的数据，以减少对时空误差传播的感知。错误隐藏方法可以受益于免费的抗错编码，以便错误尽可能本地化，并具有本地良好数据，可以在此基础上进行估计。

隐藏方法可以在其他技术，特别是需要反馈通道的交互式方法无法提供解决方案的地方——例如，在广播或多播系统中。隐藏是一项后处理操作，任何编码标准都没有强制要求。存在许多算法，其中一些算法在下面被描述，分为空间、时间或混合方法。读者还参考了许多关于该主题的优秀论文和评论，包括参考文献。[3]，[5]，[6]，[27]。参考文献[28]还包含基于运动场插值和多假

设运动估计的具体方法的性能细节。表现出一些潜力的替代方法是基于纹理合成或绘画的方法。这些在这里没有进一步介绍，但读者会被参考。[30]。

11.8.1.检测缺失的信息

首先，在应用隐藏方法之前，我们需要了解错误的确切位置。这可以通过多种方式完成，并在一定程度上取决于实现细节。一些例子是：

- **使用标头信息：**例如，数据包头中的序列号可用于检测数据包丢失。
- 使用前向纠错（FEC）：FEC可以在应用程序层和链接层使用。
- **检测语法和语义违规：**例如非法解码的代码词、无效的解码单元数、超出范围的解码参数等。
- **检测违反视频信号一般特征的行为：**例如，具有高度饱和颜色（粉红色或绿色）的块，大多数解码像素需要剪切的块，块边界处的强烈不连续等。

这些方法都不能保证发现所有错误，有些方法可能会被错误检测。因此，在实践中，经常使用组合。现在让我们考虑在检测到错误后隐藏错误的方法。

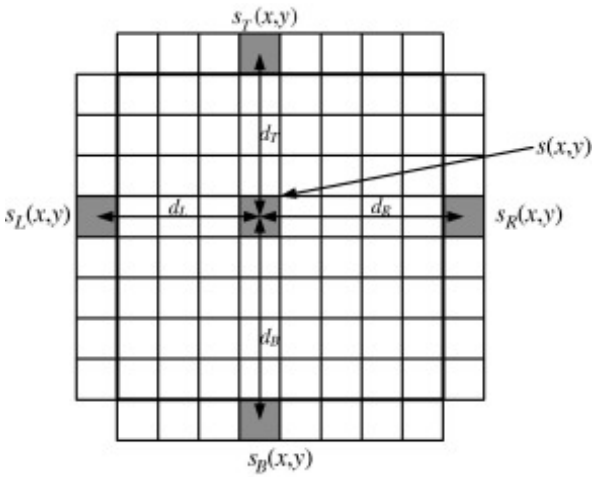
11.8.2。空间错误隐藏（SEC）

空间误差隐藏方法利用了大多数视频信号中存在的空间相关性。它们在空间连续性和平滑的假设下运作。最常见的空间插值方法是基于参考文献中提出的加权平均技术。[31]。隐藏最好基于正确接收的数据；然而，在损失范围更广的地区，如果以前隐藏的块或像素用作估计基础，可以使用渐进方法。

基于加权平均的空间插值如图11.27所示。在检测到丢失的块（以较厚的衬里边界表示）后，每个像素，$s(x,y)$，使用以下表达式进行插值：

$$s(x,y)=\frac{d_L s_R(x,y)+d_R s_L(x,y)+d_T s_B(x,y)+d_B s_T(x,y)}{d_L+d_R+d_T+d_B} \tag{11.5}$$

在哪里$s_R(x,y)$等是位于缺失块的右侧、左侧、顶部和底部的空间相邻宏块的边框像素。适用于*工头*序列的空间误差隐藏示例如图11.28所示。在这种情况下，与原始无错误帧的MSE为46.6。



下载：下载全尺寸图像

图11.27。使用最近的邻像素隐藏空间错误。

下载：下载全尺寸图像

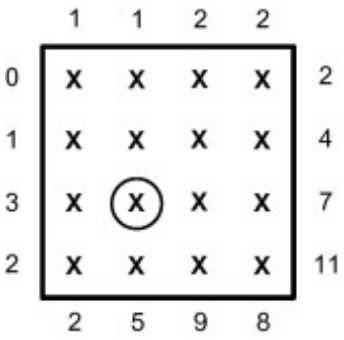
图11.28。空间误差隐藏适用于工头序列中丢失的一排宏块。左：原件，显示错误。中间：空间误差隐藏的结果。右：放大差分信号。MSE = 46.6。

由D提供。Agrafiotis。

SEC基本方法有一些扩展可以提供额外的好处，但通常以额外的复杂性为代价。例如，可以强制执行*边缘连续性*。这意味着，如果在相邻的块中检测到一条或多条边，那么在隐藏期间，应该保留这条边。Agrafiotis等人[32]使用一种边缘保存方法，这种方法不仅可以保存现有边缘，还可以避免引入新的强边。该方法根据相邻边缘的定向熵，在定向插值和常规双线性插值之间切换。拟议战略在不影响这两种方法性能的情况下，利用了这两种方法的优势，在某些情况下提供了超过1分贝的性能改进。马等人也提出了一种类似的方法。[33]。

示例11.7空间误差隐藏

考虑下面显示的缺失数据块，并指示正确接收的边界像素。通过在此块上执行SEC，计算图表中显示的缺失像素的值。



下载：下载全尺寸图像

解决方案

使用方程（11.5）：

$$s(x,y) = \frac{2\times7+3\times3+3\times5+2\times1}{10} = 4$$

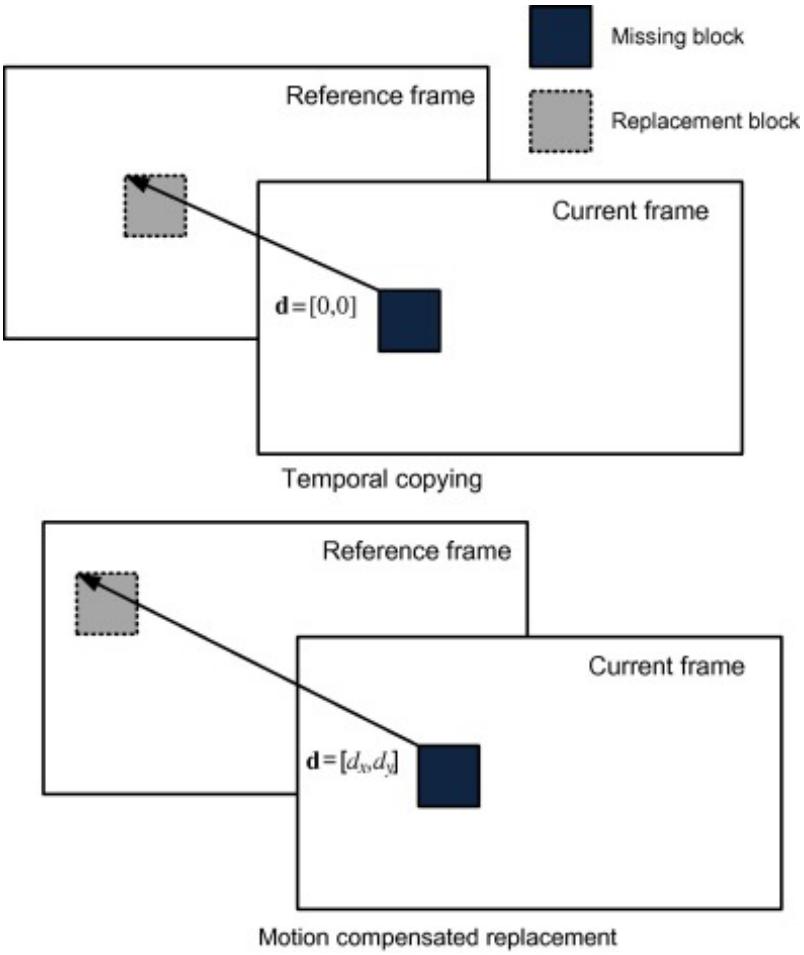
块中所有其他缺失的像素值都可以以类似的方式插值。

11.8.3。时间错误隐藏

时间误差隐藏方法在运动场的时间连续性和平滑性假设下运行[28]、[29]。他们利用了大多数自然序列的高度相关性，并根据周围地区的预测隐藏了损坏或丢失的像素。它们在编码块之间工作得很好，但在编码块内通常不太成功。下面考虑两种基本技术。

临时复制（TEC_TC）

时间错误隐藏的最简单形式是*时间复制*。在这种情况下，如图11.29所示，底层模型是零运动模型之一。这种方法非常容易实现，但实际中很少有足够的运动模型。包含工头序列一片的单个丢失数据包的结果如图11.30所示。虽然这个结果看起来合理，但由于人的头部移动，误差实际上相当大。如果你检查他脸部左缘的区域，这种位移尤为明显。



下载：下载全尺寸图像

图11.29。基于时间复制（顶部）和运动补偿预测（底部）的时间误差隐藏。

下载：下载全尺寸图像

图11.30。基于时间复制的工头时间误差隐藏结果。MSE = 19.86。

由D提供。Agrafiotis。

运动补偿时间替换（TEC_MCTR）

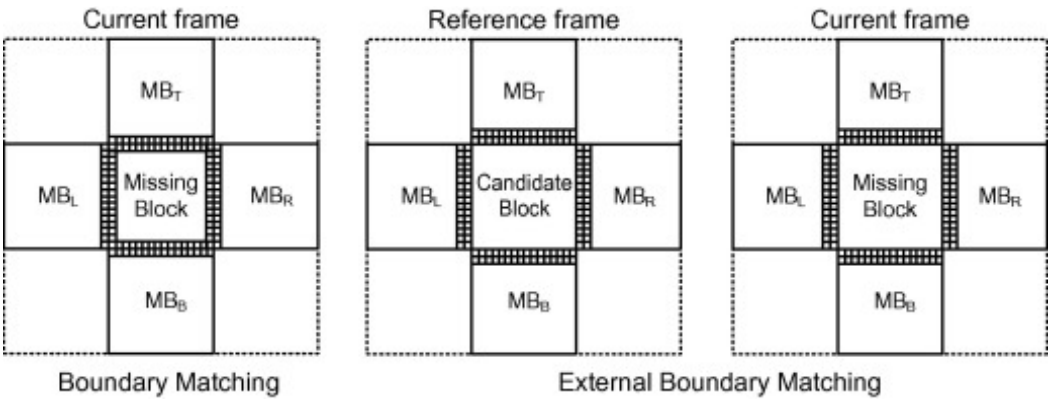
如图11.29（底部）所示，此方法将缺失的块替换为参考帧中的最佳候选区域。这比之前的方法复杂得多，因为它需要生成一个候选运动矢量列表，以选择最佳替换块。该方法采用两个阶段的过程：

1. 隐蔽位移的估计。
2. 位移评估和更换选择。

第一阶段提出了一个问题：我们如何为TEC_MCTR选择候选运动矢量？假设缺失块的运动矢量也丢失了，那么有几个可能替代的候选者。这些包括：

- 来自空间相邻块的运动矢量。
- 来自相邻块的运动矢量的秩统计或平均数。
- 来自暂时相邻块的运动矢量。

然后，我们需要评估这些候选运动矢量，并选择产生最佳匹配的矢量。因此，第二个问题是——我们使用什么匹配度量来决定哪个候选向量最好？这通常使用*边界匹配错误*（BME）的形式完成。该指标的选项如图11.31所示。显示的第一种方法简称BME，通过计算匹配块的边界像素和周围块的相邻像素之间的SAD值来评估所有候选替换块。第二种方法称为*外部边界匹配错误*（EBME），再次计算一个SAD值，但这次是在当前帧中围绕缺失块的像素行和引用帧中候选块周围的像素行之间。EBME是一个优越的指标，因为它符合时间隐藏是保持时间光滑的假设。另一方面，BME与模型不一致，因为它使用空间连续性指标来评估时间平滑度。



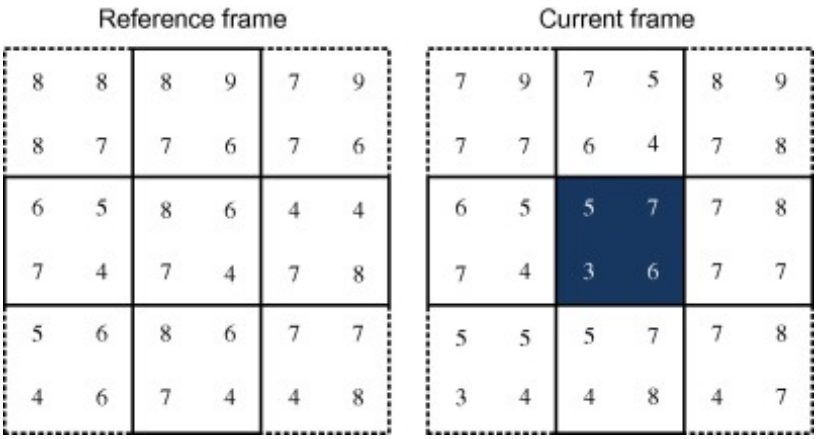
下载：下载全尺寸图像

图11.31。边界匹配错误措施。左：边界匹配错误（BME）。右：外部边界匹配错误（EBME）。

基于运动补偿替换的工头序列的时间误差隐藏结果如图11.32所示。由于底层模型是平滑平移运动模型，因此该技术通常比时间复制性能好得多。从图11.32可以看出，残余误差低于图11.30，工头侧面的人工制品显著减少。

示例11.8时间错误隐藏

考虑当前帧的区域和相邻参考帧的同地区域，如下所示。假设2×2在传输过程中，灰色显示的块已丢失，但显示的所有其他数据已正确接收，请使用BME指标计算TEC_MCTR块。



下载：下载全尺寸图像

当前帧中相邻块的运动矢量值为：

$$\mathbf{d}_L = [0,0]$$
$$\mathbf{d}_R = [-1,1]$$
$$\mathbf{d}_T = [-2,2]$$
$$\mathbf{d}_B = [2,0]$$

解决方案

我们可以使用这四个运动矢量来选择要替换的候选块。在每种情况下，我们计算BME值如下：

$d = [0,0] : BME = 19$
$d = [-1,1] : BME = 11$
$d = [-2,2] : BME = 7$
$d = [2,0] : BME = 13$

根据BME指标，最好的候选块是与$[-2,2]$运动矢量。因此，在这种情况下，替换像素是：

$$\begin{bmatrix} 5 & 6 \\ 4 & 6 \end{bmatrix}$$

下载：下载全尺寸图像

图11.32。基于运动补偿替换的*工头*时间误差隐藏结果。MSE = 14.61。

由D提供。Agrafiotis。

11.8.4.混合方式与模式选择

选择使用空间或时间隐藏并不总是简单的。在大多数情况下，时间误差隐藏提供了最佳效果，许多简单的系统仅使用时间复制方法，如果没有运动，这种方法效果很好！在实践中，有一种方法比另一种方法更受欢迎。例如，在镜头切割或褪色的情况下，空间隐藏通常更优越。在静态场景中，时间块复制将产生良好效果，在相机或物体运动的情况下，运动补偿的时间替换最有可能产生最佳结果。然而，如果场景运动与用于查找候选矢量的运动模型不匹配，那么恢复空间隐藏可能更好。

我们在第 11.5.5 节中看到，切片结构模式在错误恢复方面是有益的。特别是，柔性*宏块*排序（FMO）等技术可以帮助确保，如果一个切片丢失，一些好数据保留在相邻的切片中，从而可以插入丢失的数据。这种类型的抗错编码与隐藏相结合可以带来显著的好处。在实现混合系统时出现的问题是：我们如何选择操作模式——时间模式还是空间模式？这可以使用模式选择算法完成。

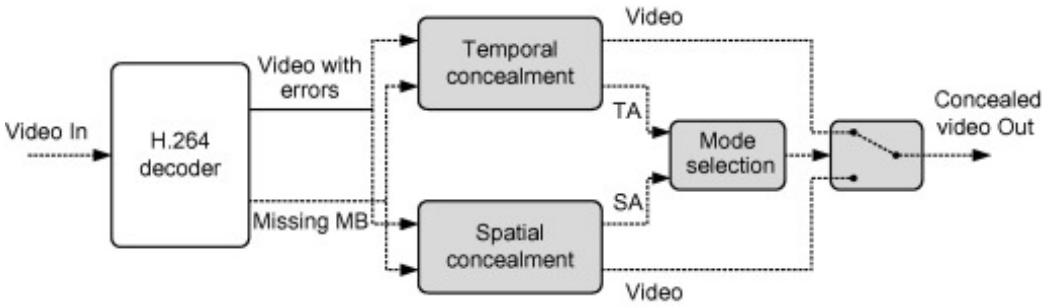
例如，图11.33显示了Agrafiotis等人提出的EECMS算法的架构。[27]。这在算法11.2中更正式地定义。EECMS根据对运动补偿的比较执行模式选择

算法11.2 EECMS

1. INPUT video with lost block;
2. REPEAT for each corrupted block:
   3. Perform TEC: Use EBME for matching;
   4. Compute spatial activity: $SA = \mathrm{E}\left\{(s_c - \mu)^2\right\}$;
   5. Compute temporal activity: $TA = \mathrm{E}\left\{(s_c - s_r)^2\right\}$;
   6. IF ($TA < SA$) OR ($TA <$ Threshold) THEN (use TEC, GOTO (8));
   7. Perform SEC; use directional interpolation;
8. UNTIL all missing blocks concealed;
9. END.

下载：下载全尺寸图像

丢失块附近的时间活动和*空间活动*。空间活动（SA）计算为当前帧中周围MB的方差，而时间活动（TA）根据像素之间的平均平方误差来衡量运动均匀性(s_c)在当前帧中缺失块周围的MB中(s_r)围绕参考框架中的替换MB。只有当SA低于SA且SA高于指定阈值时，才会调用SEC。表11.2总结了EECMS方法的选定结果，并与H.264/AVC联合模型（JM）中使用的隐藏算法进行了比较。在参考文献[27]中评估的所有序列中，性能都报告了高达9分贝的PSNR。



下载：下载全尺寸图像

图11.33。使用模式选择增强错误隐藏[27] 。

表11.2。EECMS的选定性能结果（来自Ref.[27]）。

PSNR（分B）	0% PER		1%/		每20%
		JM	EECMS	JM	EECMS
工头	40.14	38.11	39.44	27.33	31.10
公交车	37.20	35.08	36.33	23.79	26.99
足球	31.27	30.14	30.56	22.05	23.22

11.9。拥堵管理和可扩展的视频编码

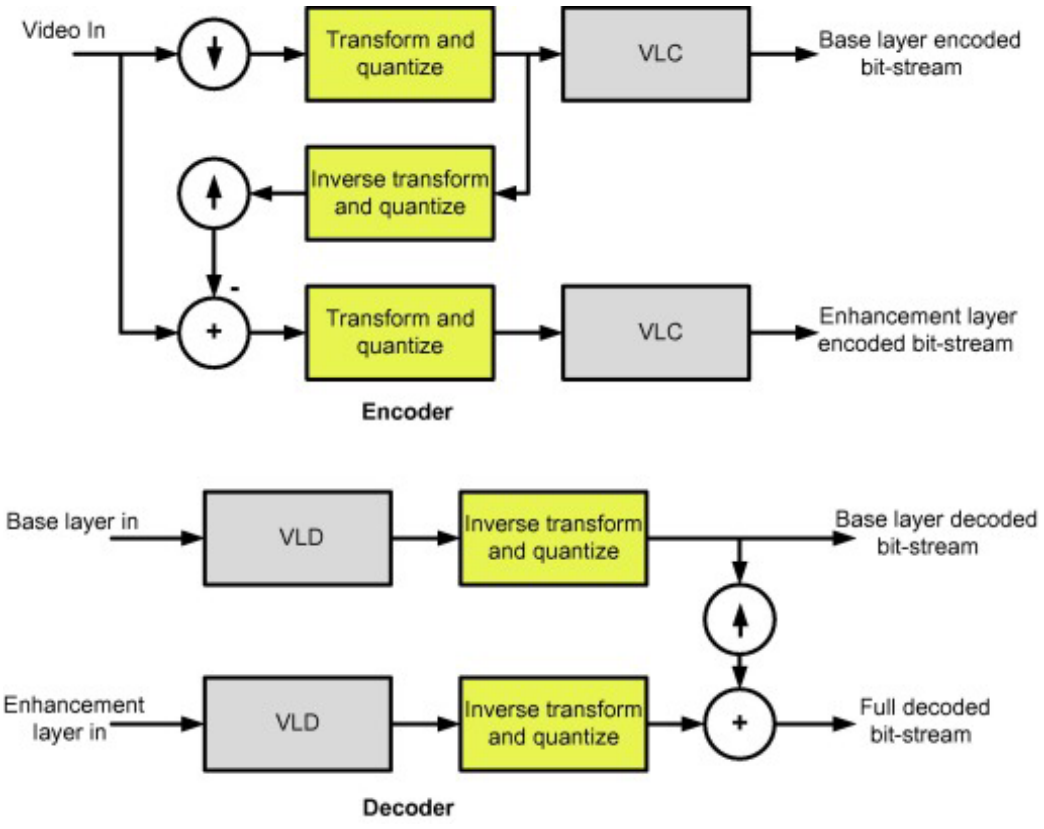
在流在线编码的情况下，可以控制视频速率以匹配频道速率。然而，在许多情况下，信道条件可能非常多变，并且可能没有详细的知识（特别是对于多播应用程序）。此外，对于预编码流，除非对同一内容的多个版本进行编码，否则无法调整速率。解决这个问题的方法是在多个分层层或多个独立描述中表示比特流。这些方法将在下面简要描述。

可伸缩视频编码

在视频传输中，信号在不明确了解下游网络条件的情况下进行编码和传输是常见的。如果网络阻塞，如互联网等数据包交换网络存在网络拥塞，路由器将丢弃由于拥塞而无法转发的数据包。在某些情况下，流中的数据包可以优先排序或嵌入，以便在更重要的数据之前丢弃最不重要的信息。这种机制适合可伸缩或分层编码，其中视频信号由空间分辨率、时间分辨率或信噪比的分层组成。这使设备能够传输和接收多层视频流，通过使用可选的额外层来提高分辨率、帧率和/或质量，从而提高基本质量水平。

第一个包含可选可伸缩模式的标准化编解码器是MPEG-2。然而，这些在实践中很少使用。最近，H.264/AVC在其可扩展视频编码（SVC）附件G扩展H.264/MPEG-4 AVC中引入了一套全面的可伸缩模式。SVC具有许多吸引人的功能，尤其是它与传统的H.264/AVC比特流兼容。正如我们在第六章中已经看到的，可伸缩方法在JPEG2000等静态图像编码标准中也受到了青睐。一些监控和视频会议应用程序目前正在采用可扩展编码方法。

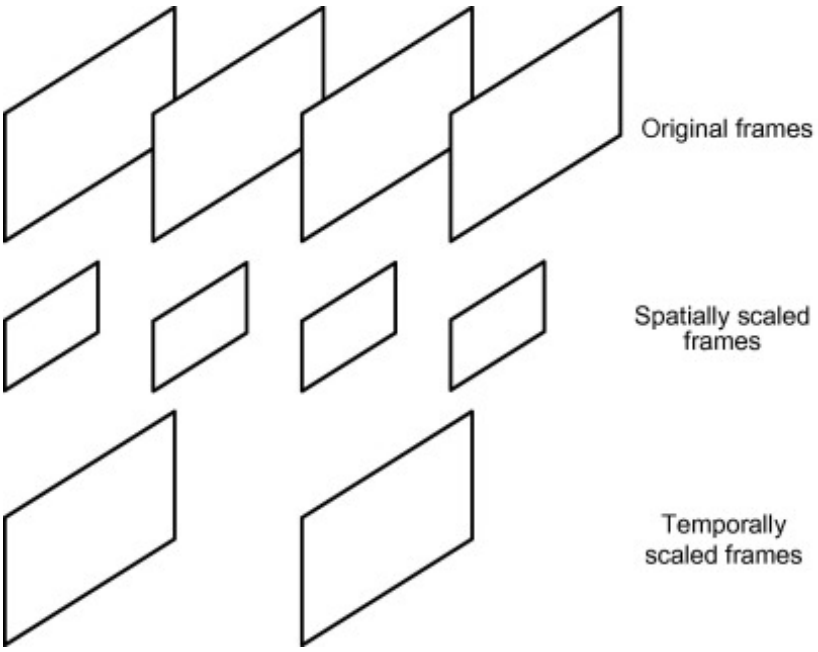
在可伸缩编解码器中，目的是创建一个分层的比特流，即使一些信息在传输过程中丢失，或者终端无法解码，也可以在解码器上重建视频的有意义的表示形式。可伸缩视频编码器和解码器对的基本架构如图11.34所示。该图显示了空间或时间可伸缩性的情况。另一种情况——SNR可伸缩性——是省略上下采样运算符的子集。[2]



下载：下载全尺寸图像

图11.34。可伸缩（空间或时间）视频编解码器架构。

考虑空间可伸缩的情况，如图11.34所示，图11.35。原始视频输入首先被采样到基层的分辨率，然后像往常一样进行转换和量化。该层被编码并作为基层传输。然后，基层在采样到原始分辨率之前进行逆变换和逆量化。然后从原始视频中减去重建的基层，形成一个剩余信号，并将其编码为增强层。显然，基层必须比其增强层具有更高的优先级，因此有必要使用不平等的错误保护。



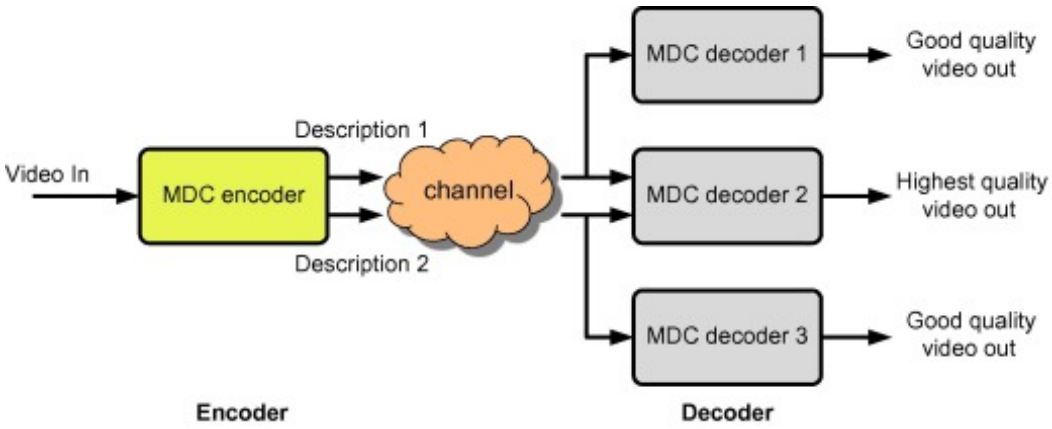
下载：下载全尺寸图像

图11.35。可伸缩编码的帧类型。

在解码器上，基层像往常一样解码，但如果增强层可用，则可以向上采样基层，添加到解码后的增强层，形成全质量的增强输出。这个过程类似于时间和信噪比的可伸缩性。有关更多信息，请参阅参考文献。[4]，[34]，[35]。

多重描述编码（MDC）

多种描述编码以不同于可扩展编码的方式解决了拥塞管理问题。MDC没有创建依赖信息的层次结构，其中每个层的解码取决于下一个最粗糙层的存在，而是创建许多独立的编码（描述）。这些可以独立传输，并且可能从源头到目的地走不同的路径。利用这种路径多样性，MDC确保，如果正确接收到任何描述组合，则可以在解码器中实现有意义的重建。如果收到所有描述，则可以进行全面质量重建。MDC正在直播和点对点分发等方法中找到应用。这种方法如图11.36所示。



下载：下载全尺寸图像

图11.36。多描述编解码器架构。

形成高效的独立视频描述，其中编码增益取决于预测编码的使用，这并不简单。已经提出了许多方法，Aposolopoulos等人[36]和Wang等人对这些方法及其表现进行了审查。[37]。例如，如第11.5.5节所述，FMO切片结构方法的使用已被王等人所利用。[38]。这种方法修改了H.264/AVC，使用三个运动补偿回路和两个切片组来形成两个独立的描述。中央编码器与单个描述H.264编码器相同，而侧面编码器对每个切片组进行处理。

Tesanovic等人[39]将路径多样性的概念扩展到基于空间多路复用的多输入多输出（MIMO）无线技术。他们的论文建议使用多描述视频编码（MDC）作为模拟传统空间多路复用（SM）系统中缺乏的空间多样性的手段。奇异值分解（SVD）用于创建正交子信道，为将视频内容映射到无线信道提供了有效的方法。结果表明，与标准单描述视频传输相比，解码测试序列的平均PSNR约为5-7分贝。在信噪比值低的信道中，通过使用不平等的功率分配，额外的2-3分贝进一步增强了这一点。

11.10。摘要

在本章中，我们研究了一些可用于提高视频传输在有损通道上的可靠性的方法。我们考虑了为什么误差在可变长度和预测编码机制的影响下在空间和时间上传播，并看到了一些例子，这些例子清楚地表明，本书前面考虑的基本编码方法需要以保持速率-失真性能的方式进行调整，同时提高比特流对错误的恢复力。我们看到了如何利用FEC和ARQ等网络技术来提供更好的性能，以及如何修改源代码本身，以更低的开销提供更好的错误恢复力。显然，PVQ等方法对损坏的数据提供了更大的容忍度，同时避免了对复杂编码工具的需求。最后，我们看到错误隐藏，可能与其他错误恢复方法相结合，如何在传输开销很少或没有的情况下显著提高主观质量。

参考文献

[1] C.钟，C.-M.谭，D。劳伦森，S。麦克劳克林，M.Beach，A.尼克斯
一种新的5GHz频段无线局域网系统的统计宽带时空信道模型
IEEE 通信选定领域期刊，21(2) (2003)，p.139-150
在Scopus中查看记录 谷歌学术

[2] G.阿塔纳西亚杜，A。尼克斯，J。麦吉汉
微细胞射线跟踪传播模型及其窄带和宽带预测的评价
IEEE 通信选定领域期刊，18(3) (2000)，p.322-335
在Scopus中查看记录 谷歌学术

[3] T。斯托克哈默，M.汉努克塞拉，T。维冈德
无线环境中的H.264/AVC
IEEE 视频技术电路和系统交易，13（7）（2003），p.657-673
在Scopus中查看记录 谷歌学术

[4] 范德沙尔先生，P.周（编辑），IP和无线网络上的多媒体：压缩、网络和系统，学术出版社（2011年）
谷歌学术

[5] Y。王，S。温格，J。温，A。卡察格洛斯
实时视频通信的抗误差编码技术综述
IEEE信号处理杂志，17（4）（2000年），p.61-82
在Scopus中查看记录 谷歌学术

[6] Y。王，Q。朱
视频通信的错误控制和隐藏：回顾
IEEE会议记录（1998年），p.974-997
（多媒体信号处理专题）
在Scopus中查看记录 谷歌学术

[7] A.肖克罗拉希
猛禽代码
IEEE Transactions on Information Theory，52 (6) (2006)，pp.2551-2567
CrossRef 在Scopus中查看记录 谷歌学术

[8] M.卢比
LT代码
第43届IEEE计算机科学基础研讨会记录（FOCS）（2002年），第271-280页
CrossRef 在Scopus中查看记录 谷歌学术

[9] 美国。林，P。于
用于卫星信道误差控制的奇偶校验重传混合ARQ方案
IEEE通信交易，30（I）（1982），第1701-1719页
在Scopus中查看记录 谷歌学术

[10] P.费雷，A。杜费西，J。钟豪，A。尼克斯

无线局域网的强大视频传输
IEEE 车辆技术交易，57 (4) (2008)，p.2596-2602
CrossRef 在Scopus中查看记录 谷歌学术

[11] J。钟豪，D。公牛
失去弹性H.263网络视频
信号处理：图像通信，16（2001年），第891-908页
文章 下载PDF 在Scopus中查看记录 谷歌学术

[12] Y。马里兰州高岛市和田，H。村上
可逆可变长度代码
IEEE通信交易，43（1995年），第158-162页
谷歌学术

[13] A.弗兰克尔，S。克莱因
双向霍夫曼编码
计算机期刊，33（4）（1990年），第297-307
谷歌学术

[14] B.吉罗德
前缀码字双向可解码流
IEEE Communications Letters，3（8）页（1999年），245-247
在Scopus中查看记录 谷歌学术

[15] JVT-B027.doc，散射切片：H.26L的新错误恢复工具，ISO/IEC MPEG和ITU-T VCEG的联合视频团队，2002年2月。
谷歌学术

[16] P.费雷，D。Agrafiotis，D。公牛
一种视频误差恢复冗余切片算法及其相对于其他固定冗余方案的性能
信号处理：图像通信，25（2010），第163-178页
文章 下载PDF 在Scopus中查看记录 谷歌学术

[17] M.和田
使用错误隐藏选择性恢复视频数据包丢失
IEEE 通信选定领域期刊，7（5）（1989年），第807-814
在Scopus中查看记录 谷歌学术

[18] E。斯坦巴赫，N.法伯，B。吉罗德
H.263的标准兼容扩展，用于移动环境中的健壮视频传输
IEEE 视频技术电路和系统交易，7 (6) (1997)，pp.872-881
在Scopus中查看记录 谷歌学术

[19] 范德沙尔先生，S.Krishnamachari，S。崔，X。徐
自适应跨层保护策略，用于强大的可扩展多媒体视频传输
IEEE 通信选题期刊，21(10) (2003)，p.1752-1763

谷歌学术

[20] P.费雷，J。钟豪，A。尼克斯，D。公牛
基于失真无线视频传输的链接自适应
EURASIP 信号处理进展期刊（2008 年），p.17
（第253706条）
谷歌学术

[21] D.雷德米尔，N。金斯伯里
EREC：一种编码可变长度数据块的错误弹性技术
IEEE图像处理交易，5（4）（1996年），第565-574页
在Scopus中查看记录 谷歌学术

[22] D.Redmill，D。公牛
静止图像的弹性误差算法编码
IEEE国际图像处理会议（1996年），p.109-112
在Scopus中查看记录 谷歌学术

[23] R.斯旺，N。金斯伯里
MPEG-2的转码，以提高对传输错误的恢复力
IEEE国际图像处理会议（1996年），p.813-816
在Scopus中查看记录 谷歌学术

[24] T。费舍尔
金字塔矢量量化器
IEEE信息理论交易，32（4）（1986年），p.568-583
在Scopus中查看记录 谷歌学术

[25] J。钟豪，D。公牛
强大的图像和视频编码，带有金字塔矢量
IEEE电路和系统国际研讨会，第4卷（1999年），第4卷。332-335
在Scopus中查看记录 谷歌学术

[26] 美国。博哈里，A。尼克斯，D。公牛
使用金字塔矢量量化进行速率-失真优化的视频传输
IEEE图像处理交易，21（8）（2012年），第3560-3572页
CrossRef 在Scopus中查看记录 谷歌学术

[27] D.Agrafiotis，D。公牛，C。卡纳加拉贾
使用模式选择增强错误隐藏
IEEE Transactions on Circuits and Systems for Video Technology，16 (8) (2006)，p.960-973
CrossRef 在Scopus中查看记录 谷歌学术

[28] M.Al-Mualla，C.卡纳加拉贾，D。公牛
移动通信视频编码

学术出版社（2002）
谷歌学术

[29] M.Al-Mualla，C.卡纳加拉贾，D。公牛
使用运动场插值隐藏时间误差
电子信件，35（3）（1999年），第215-217
在Scopus中查看记录 谷歌学术

[30] H。拉克什曼，M。科佩尔，P.Ndjiki-Nya，T。维冈德
使用基于稀疏重建的纹理细化进行图像恢复
IEEE国际声学语音和信号处理会议（ICASSP）（2010年），p.786-789
在Scopus中查看记录 谷歌学术

[31] P.萨拉马，E。什罗夫，E。科伊尔，E。德尔普
编码视频流的错误隐藏技巧
IEEE国际图像处理会议，第1卷（1995年），第1页。9-12
CrossRef 谷歌学术

[32] D.Agrafiotis，D。公牛，C。卡纳加拉贾
空间误差隐藏与边缘相关的感知考虑
信号处理：图像通信，21（2006年），第130-142页
文章 下载PDF 在Scopus中查看记录 谷歌学术

[33] M.马，O。Au，G。陈，M.-T.周日
边缘定向错误隐藏
IEEE 视频技术电路和系统交易，20 (3) (2010)，p.382-395
在Scopus中查看记录 谷歌学术

[34] Y。王，J。Ostermann，Y.-Q。张
视频处理和通信
普伦蒂斯大厅（2002）
谷歌学术

[35] H。施瓦茨，D。马普，T。维冈德
H.264/AVC标准的可扩展视频编码扩展概述
IEEE Transactions on Circuits and Systems for Video Technology，17 (9) (2007)，p.1103-1120
CrossRef 在Scopus中查看记录 谷歌学术

[36] J。阿波索洛普洛斯，M。特罗特，W.-T.谭
多媒体流的路径多样性
范德沙尔先生，P.周（编辑），IP和无线网络上的多媒体：压缩、网络和系统，学术出版社（2011年）
谷歌学术

[38] D.王，N。卡纳加拉贾，D。公牛
基于切片组的多个描述视频编码，具有三个运动补偿回路
IEEE电路和系统国际研讨会，第2卷（2005年），第2页。960-963
在Scopus中查看记录 谷歌学术

[39] M.特萨诺维奇，D。公牛，A。杜费西，A。尼克斯
使用多描述编码增强MIMO无线视频通信
信号处理：图像通信，23（2008年），第325-336页
文章 下载PDF 在Scopus中查看记录 谷歌学术

⋆ 有关图11.8的颜色和更高质量的版本，请参阅电子版本或网站。

1 架空会根据压缩比和内容而变化。

2 对于信噪比可伸缩性，修改图11.34中的编码器是正常的，以便向上缩放路径中只包含量化和逆量化操作。