

基于帧拆分的 MPEG 视频的可靠传输

Robust MPEG Video Transmission Based on Frame Partition

李国东 杨海荣 王东奎 张德富

(南京大学计算机软件新技术国家重点实验室 南京210093)

Abstract This paper provides a robust MPEG video transmission scheme based on frame partition: according to the playback rate, frame partitioner divides the MPEG video flow into two sub-flows with different priorities, the important data are stored in the high-priority sub-flow. After being Reed-Soloman coding, the high-priority sub-flow is sent together with low-priority sub-flow parallelly. This scheme is particularly designed for MPEG video transmission, and the MPEG video frame's characteristic is fully considered while designing the frame partitioner, RS coding and sending way. Test results show that this scheme guarantees video transmission's QoS, optimizes sender's bandwidth utilization and improves the network performance.

Keywords QoS, FEC, Reed-Soloman coding MPEG, Internet video

1. 前言

近年来在 Internet 上传送 MPEG 视频的需求和应用越来越多。多媒体交互式应用要求短时延,一般采用 UDP 协议进行传送,但 UDP 是不可靠传输协议,传送过程中的包丢失将导致视频质量下降。MPEG 使用的帧间预测技术使得一旦 I-帧或 P-帧在传送中丢失,依赖它们的许多帧将受影响,即使很小的包丢失率也会严重影响接受到的 MPEG 视频的质量。

ATM 网络支持优先权传输;IPV6 包头部的流标志中可设置优先权信息,对 IPV4 包头部的服务标记的设计,在传统 Internet 上也可设置优先权,在路由器中运行优先权处理算法后,就能在 Internet 上进行接优先权传输。将 MPEG 流拆分为不同优先权的子流,高优先权子流包含重要数据(如 I-帧和 P-帧),低优先权子流包含相对不重要信息(如 B-帧),再将拆分后的流进行优先权传输,能有效地保证 MPEG 流传输的 QoS(服务质量)。在不支持优先权传输的网络上,为重要数据加入冗余信息以在数据丢失时进行恢复,同样能保证 MPEG 流的传输质量。故此,本文提出一个 MPEG 可靠传输的方案:用帧分离器将源 MPEG 视频流拆分为不同优先权的子流,拆分标准由任意定制的回放速率决定,重要信息保存在高优先权子流中。拆分后,若网络支持优先权传输,则直接在网络上传送,否则对高

优先权子流进行前向纠错编码(FEC)^[1]后再发送数据,本文的 FEC 采用牛顿插值 Reed-Soloman 编码^[2]。拆分算法还可被 VoD 系统用来提供 VBR 服务。

MPEG-2 使用分层视频编码技术^[3]来提高传输的 QoS,编码器产生一个基层和一个或多个增强层,接收端接收基层后根据带宽能力再接收一个或多个增强层,这种技术通常要求网络支持优先权传输^[4]。MPEG-4 中提供了错误预防和恢复机制^[5],如 RVLC 和数据划分技术,RVLC 适用于随机字节错误的情况,但对于整个包丢失的情况无能为力,而数据划分技术通过将宏块动作和纹理信息分开来拆分数据。与 MPEG-2 和 MPEG-4 在编码器中进行拆分^[6]不同,本文的分离器以帧为单位拆分,且不需要在数据编码过程中进行,适用于对已生成的视频流进行拆分,特别适合 VoD 系统使用,因为 VoD 系统存放的都是已生成的视频数据。现在大多数 VoD 系统只能提供定制速率为源数据率倍数的 VBR 服务,本文的帧分离器通过不发送低优先权流,能为 VoD 系统提供任意速率的 VBR 服务。

对多媒体传输进行错误预防和恢复的技术可分为反馈式和非反馈式两类。反馈式通过发送者和接受者的应答来进行丢失重传,如 TCP 协议,由于实时视频传送要求低时延,反馈式不适用;非反馈式的研究有很多^[4,5,7],其中文^[4]利用分层编码技术产生基层和增

李国东 硕士生,主要研究领域为分布对象技术,网络通信,杨海荣 硕士生,主要研究领域为并行/分布处理,网络计算,王东奎 硕士生,主要研究领域为并行/分布处理,网络平台,张德富 教授,博导,主要研究领域为并行处理技术和分布式系统。

强层,赋予增强层较高的优先权后进行传输;文[5]使用 MPEG-4的数据划分技术拆分流后进行传输差错恢复;文[7]也采用了分层编码,并对基层和增强层都使用了 FEC 编码,但目前的研究都是在编码过程中拆分信息,且没有针对 MPEG 视频帧类型的特点进行设计。本文的帧拆分器设计、FEC 编码和发送方法都充分考虑了 MPEG 视频的特点,以帧为单位进行计算和控制,保证了 MPEG 视频传输的 QoS,并优化了发送端的带宽利用,提高了网络效率。

2. MPEG 视频帧的拆分

2.1 MPEG 视频帧的类型和拆分原则

用户由于机器能力和网络情况的不同需要的回放率也不同,服务器也可根据本身情况和网络环境调整 MPEG 流的输出速率。本文根据回放率为视频流引入优先权机制,设计了一个 MPEG 帧优先权拆分算法,在保证最低信息传输的基础上追求更高的服务质量。也可如传统 VoD 系统一样直接控制回放率,此时低优先权数据不被发送。MPEG 视频帧有三种类型:

I-帧:自身保持内容,不指引其它类型的帧,不需要其它帧的信息。在三种帧中压缩率最低,I-帧是 MPEG 流中随机访问的点,是最基本和最重要的帧。

P-帧:需要之前 I-帧和/或所有之前 P-帧的信息被编码和解码。

B-帧:需要之前和之后的 I-帧和/或 P-帧的信息被编码和解码,压缩率最高,不能用作随机访问,是独立性最差的帧。

帧优先权的拆分原则体现为高优先权子流的选取原则:根据 MPEG 视频帧的特点,先选择 I-帧,然后第一个 P-帧,第二个 P-帧……直至最后的 P-帧,最后选择 B-帧;帧的选择在不违背依赖关系的情况下应尽量分布平均。

2.2 定义

MPEG 视频流 F 由三元组 (P, S, N) 标识,其中 P 是帧的位置,从 0 开始编号, S 是帧的大小, N 是流中有效帧(长度为 0)的个数, F_j 是流 F 的第 j 个帧,用 (P_j, S_j) 来标识。

通常同一视频文件中 GoP 模式不变,令视频 GoP 模式为 $IB \dots BP_1B \dots BP_2B \dots \dots P_nB \dots B$ 。令 $\text{sum}(\text{GoP})$ 为一个 GoP 中帧的总数, $I(a), P_1(a), P_2(a), \dots, P_n(a), B(a)$ 分别为 a 个帧中 I-, P_1 -, P_2 -, \dots , P_n -, B-帧的数目。令 T_p 为两个周期请求间的帧数和 GoP 模式中帧数的最小公倍数,即

$$T_p = \text{LCM}(\text{sum}(\text{GoP}), R)$$

其中 $\text{LCM}(x, y)$ 返回 x, y 的最小公倍数,则在 T_p 个帧中 I-, P_1 -, P_2 -, \dots , P_n -, B-帧的数目分别为

$$I(T_p) = P_1(T_p) = \dots = P_n(T_p) = T_p / \text{sum}(\text{GoP})$$

$$B(T_p) = T_p - (n+1) \times I(T_p)$$

若要求的回放率为 r 帧/秒,设 R 为 MPEG 视频流的标准播放速率(30 帧/秒,即 $R=30$),则在 R 帧中取 r 帧相当于在 T_p 个帧中取 $r' = r \times T_p / R$ 个帧。由于 T_p 是流的最小周期模式帧数,对整条流的拆分体现为对 T_p 个帧的拆分:源 MPEG 流 $F = (P, S, T_p)$ 拆分为高优先权流 $F^H = (P^H, S^H, r')$ 和低优先权流 $F^L = (P^L, S^L, T_p - r')$ 。对于源流中的帧 $F_j = (j, S_j)$,相应 F^H 和 F^L 的帧为 (j, S_j^H) 和 (j, S_j^L) ,其中 $S_j^H \oplus S_j^L = 0$,即 F_j 的有效信息只保留在一条子流中;流的 P 值用于两条子流的发送同步和优化, $P_j^H = P_j^L = j$ 。

2.3 算法

该算法设置 F^H 和 F^L 的 S 值,由于 S^H 和 S^L 互为补集,对于 $0 \leq j < N$,有:

$$S_j^L = \begin{cases} S_j & S_j^H = 0 \\ 0 & S_j^H \neq 0 \end{cases}$$

求出 S_j^H 后, S_j^L 的值便可相应得到,故以下只进行 S^H 的求值。

根据 I-, P-, 和 B-相互之间的重要性和所要求播放率 r' 的不同,各帧优先权的拆分算法区分三种情况:

情况 1:如果 $r' \leq I(T_p)$ 则应在 $I(T_p)$ 个 I-帧中均匀地取 r' 个帧构造 F^H ,其他的帧用来构造 F^L 。为此,使用一个辅助数组 IArray,其大小是 $I(T_p)$,将 I-帧的位置值依次存入 IArray 数组,计算 IArray 的符合以下条件的 j' 值

$$j' = \lfloor k \times I(T_p) / r' \rfloor \quad k=0, 1, 2, \dots, r'-1 \quad (1)$$

则

$$S_j^H = \begin{cases} S_j & j = \text{IArray}(j') \\ 0 & \text{其他 } j \end{cases} \quad j' = 0, 1, \dots, r'-1 \quad (2)$$

情况 2:如果 $(a+1) \times I(T_p) < r' \leq (a+2) \times I(T_p)$,这里 $a=0, 1, \dots, n-1$,则所有的 I-帧和 T_p 之内的 P_1 -, P_2 -, \dots , P_n -帧用来构造 F^H ,其中 I-帧的位置值为

$$j = \lfloor k \times I(T_p) / r' \rfloor \quad k=0, 1, 2, \dots, I(T_p)-1; \quad (3)$$

对于 P-帧,令 $\text{pos}(k)$ 为 T_p 中第一个 P_k 出现的位置,符合要求的 P_1 到 P_n 的 P-帧的位置是

$$j = \lfloor k \times \frac{P_{x+1}(T_p)}{r' - (x+1) \times P_{x+1}(T_p)} \rfloor \times \text{sum}(\text{GoP}) + \text{pos}(x+1) \quad (4)$$

$$x=0, 1, 2, \dots, a-1$$

$$k=0, 1, 2, \dots, P_k(T_p)-1$$

若 I-, P-帧间和 P-帧相互间的间隔相同,设 m 为间隔间 B-帧的数目,则 $\text{pos}(k) = (m+1) \times k$, $\text{pos}(x+1) = (x+2) \times k$, $I(T_p)$ 个 P_{x+1} -帧中前 $m^2 - (a+1) \times I(T_p)$ 个帧的优先权也用来构造 F^H ,它们的位置值 j 满足

$$j = \lfloor k \cdot \frac{P_{a+1}(T_p)}{r' - (a-1) \cdot P_{a-1}(T_p)} \rfloor + \text{sum}(GoP) + \text{pos}(a+1) \quad (5)$$

$$k = 0, 1, 2, \dots, r' - (a+1) \times P_{a-1}(T_p) - 1$$

对于满足(4)式和(5)式的 j , $S_j^H = S_j$, 而对于其他 j , $S_j^H = 0$.

情况3: 如果 $r' > (n+1) \times I(T_p)$, 所有 I 和 P 帧的用来构造 F^H , 其中另外 I 帧的位置值见公式(2), P 帧的位置值见公式(3), 不过此时 $x = 0, 1, 2, \dots, r'-1$; B 帧中有 $r' - (n+1) \times I(T_p)$ 个帧也用来构造 F^H . 因为所有 B 帧的重要性是相同的, 故 HP 优先权的 B 帧应平均分布. B 帧的选取方法与情况1中 I 帧选取方法类似(参照公式(1)(2)), 其他的帧用来构造 F^L .

3. 错误恢复技术

在传输前使用前向错误纠正(FEC)^[1]在多媒体流中增加冗余信息, 使得接受端不需要和发送端联系就能修复包丢失, 这在网络组播中尤为重要.

信息 k	1	2	3	...	m
	1	2	3	...	m
冗余 k	1	2	3	...	m
	1	2	3	...	m

图1 RS(n,k)编码

对于 FEC 已有许多著名的理论和技术, 由于 Reed-Solomon 编码(RS 编码)是最大距离独立编码, 特别适用于突发帧丢失的错误保护, 本文采用 RS 编码技术. 当一个 RS(n,k)分量从字节数据中构造时, h 个奇偶校验字节从 k 个信息字节中产生, 共有 $n = k + h$ 个字节被传送. 如图1, 一个包为一行, h 个冗余包从 k 个信息包中产生, 所有包的长度必须相同, k 个长度为 m 的信息包用 m 个 RS 分量进行编码, 对于每个 RS 分量, k 个信息字节分别从 k 个不同的包中取出, h 个构造好的奇偶校验字节放在 h 个独立的奇偶校验包中, 编码后 $n = k + h$ 个包被发送. RS 编码能够纠正任意 $h/2$ 个字节的错误或任意 h 个已知位置的字节错误, 能够应付 n 个连续错误的情况. 如果所有 k 个信息包被接收, 则接受端不需要进行计算以重构信息; 如果在传输过程中 h(或更少)个包丢失, k 个信息包能被恢复; 若多于 h 个包丢失, 则对于无法恢复的包, 直接拷贝前一个包的信息.



图2 流的拆分、编码和发送

在本文中, RS 编码只用于高优先权流, 如图2源流经帧拆分器拆分后, 高优先权流经 RS 编码器编码, 然后封装为包进行发送. 为了减少延迟, 只对单独帧进行编码, 一个帧被分成 k 个信息包, 构造好的 h 个奇偶校验包附在信息包之后发送. RS 编码和解码实际上是插值问题: 从 k 个给定的样本中计算 h 个未知的样本, 鉴于牛顿插值编码^[2]是一个快速灵活的算法, 能任意选取 n 值和 k 值, 本文采用文[2, 7]中所示的牛顿插值 RS 编码. 基本原理是迭代产生各次多项式 $C_0, C_0 + C_1(x - Z_0), C_0 + C_1(x - Z_0) + C_2(x - Z_0)(x - Z_1), \dots, C_0 + \sum_{j=1}^{k-1} \prod_{i=0}^{j-1} (x - Z_i)$, 其中 Z_i 是样本的位置, C_i 是牛顿系数, 由样本的差分计算所得.

编码后对于长度不为零的高优先权帧, $S_j^H = S_j^H + D_j^H$ (D_j^H 代表帧 j 的冗余信息), 由于不对低优先权流采用 FEC, 其 S_j^L 值不变.

4. 发送方法和带宽分析

令 B_j 为源 MPEG 流中发送第 j 个帧所需带宽, 设源 MPEG 流帧的发送时间间隔为 t, 则源 MPEG 流中 $B_j = (S_j/t)$, 整条流最大需求带宽为

$$B_{\max} = \max_{i=0}^{N-1} \left(\frac{S_i}{t} \right) = \frac{S_{\max}}{t}$$

将源流拆分后, 流 F^H 和 F^L 并行发送, 用 P 值来保持同步, 用 F^{H+L} 来表示 F^H 和 F^L 的合成, 如图3. 在发送一个非空高优先权包 (P_j^H, S_j^H) 的时间间隔中, 一个或多个低优先权的包也被发送, 令 A 为这些包的位置的集合

$$A = \{i | P_i^L \in (P_j^H, P_{j+x}^H) \& x = \min\{y | S_{j+y}^H \neq 0\}\}$$

S^H	H_1	H_4	H_1	
S^L	L_2, L_3	L_5, L_6	L_1, L_4	

图3 子流的并行发送

令 $m = \|A\|$, 即 m 为 A 中元素的个数, 则在 mt 的时间间隔内发送 1 个高优先权包和 m 个低优先权包. 所需带宽为

$$B_j^{H+L} = \frac{S_j^H + \sum_{i \in A} S_i^L}{mt}$$

所需最大带宽为

$$B_{\max}^{H+L} = \max_{k=0}^{N-1} (B_k^{H+L})$$

由于 I 帧和 P 帧的长度远大于 B 帧的长度, 在大多数情形下, 尤其在帧拆分算法的情形1和情形2中, 不等式 $B_{\max}^{H+L} < B_{\max}$ 成立, 发送端能同时发送更多的流, 网络效率得到了提高.

对 F^H 使用 RS(n,k)编码后, $S_j^H = S_j^H + D_j^H$, 所需

带宽为

$$B_{F^H+F^L} = \frac{S^I + S^P + \sum_{i=1}^n S^B_i}{mt} = \frac{\frac{n}{k} S^I + \sum_{i=1}^n S^B_i}{mt}$$

接受端根据奇偶校验包进行纠错,与不采用 RS 编码的情形一样,网络效率比串行发送的高。

5. 实验结果

实验以电视广告、球赛和 MTV 三种类型的 MPEG 视频流为代表,它们的 I-、P-、B-帧长度比各不相同,三种视频的 GoP 都为 IBBPBBPBBPBB,标准播放速率 $R=30$,则 $\text{sum}(\text{GoP})=12, T_p=\text{LCM}(12, 30)$

=60。

表1是对三种不同类型的视频流进行帧拆分的情况:电视广告的拆分标准为 $r'=24$ 帧/秒,属于拆分算法中的第三种情况, F^H 流包含所有 I-、P-帧和部分 B-帧;球赛的拆分标准是 $r'=6$ 帧/秒,属于第二种情况, F^H 流包含所有 I-帧和部分 P-帧;MTV 的回放率为 $r'=2$ 帧/秒,属于第一种情况, F^H 流只包含部分 I-帧。表中给出了拆分后 F^H 和 F^L 中 I-、P-、B-帧的比例情况,显示了拆分算法的有效性,三种视频流的 I-、P-、B-帧大小不同,拆分后的数据率由 I-、P-、B-的组成和长度决定,表中给出了 F^H 和 F^L 的数据率, F^H 流将用来进行 RS 编码。

表1 三种视频流帧拆分情况

视频流	视频类型	帧长度比 I:P:B	源数据率(b/s)	回放率(帧/秒)	F^H 中帧个数比 I:P:B	F^L 中帧个数比 I:P:B	F^H 数据率	F^L 数据率	F^H 占 F 的比例
1	电视广告	4:3:1	1M	24	5:15:28	0:0:12	886	114	88.5%
2	球赛	6:5:1	1M	6	5:7:0	0:8:40	448	552	44.8%
3	MTV	9:5:1	900k	2	4:0:0	1:15:40	202	798	22.5%

表2是对各流的 F^H 子流进行 RS 冗余编码的情况。球赛和 MTV 的 F^H 流都用10%的冗余率进行编码,而电视广告的 F^H 用四种不同的冗余率进行编码,其中0%代表不进行 RS 编码。尽管增加冗余率能提高网络低负载时的接收质量,当网络负载较大时,会因冗余信息占用带宽而引起视频质量的下降,故采用的最高冗余率为30%。表中给出了各流所用的 RS 编码参数,尽管 RS(n,k)的 n 值越大,编码效率越高,n 值过大将导致包丢失率大大增加,经过实验取 $n=100$ 较佳。

表2 各流的 F^H 子流的 RS 编码情况

视频流	F^H 数据率(kbps)	冗余率	RS 编码参数	编码后数据率(kbps)
1	886	0%	/	886
		10%	(100,91)	974
		20%	(100,83)	1063
		30%	(100,77)	1151
2	448	10%	(100,91)	492
3	202	10%	(100,91)	222

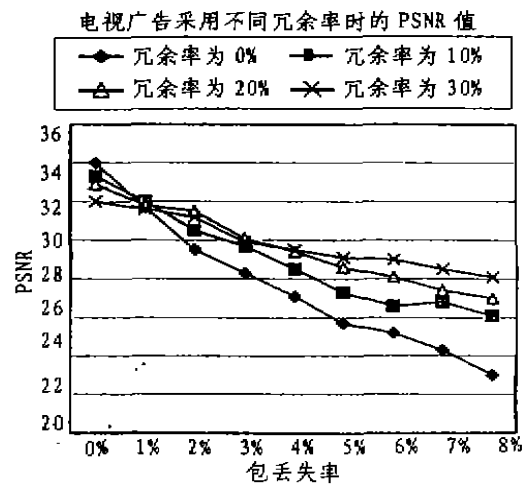
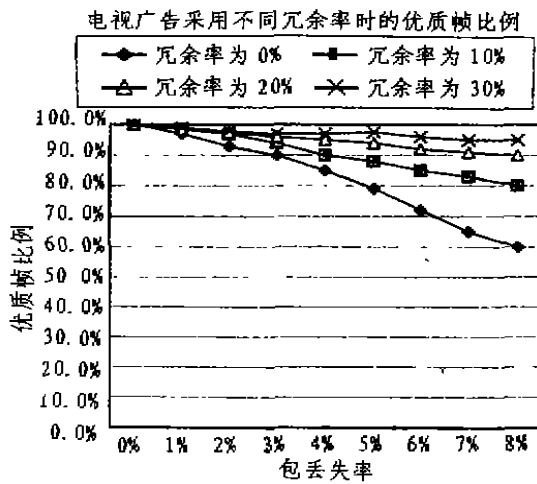


图4 电视广告编码后的视频质量

图4显示了对电视广告的 F^H 分别用四种不同冗余率进行 RS 编码产生四条子流,然后在不同网络包

丢失率的情况下的视频质量情况。左图用优质帧的比例来衡量视频质量,优质帧是指传送过程中没有丢失

或丢失信息能够从冗余信息中恢复的帧,其比例直接影响着回放效果;右图用通用的 PSNR(顶峰信噪比)值来衡量视频质量。当网络包丢失率较低时,视频质量都较高,冗余率高的子流由于冗余信息占用更多的带宽而 PSNR 值相对稍低;随着网络包丢失率的增加,冗余率低的子流视频质量迅速下降,而冗余率高的子流能保持较好的视频质量。

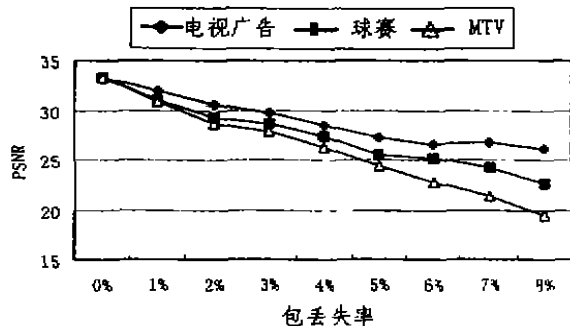


图5 三种视频编码后的视频质量

图5显示了对三种视频流的 F^H 子流用10%的冗余率编码后传输的情况。电视广告由于 F^H 占 F 的比例最高,大部分信息被冗余编码,传输最可靠;而 MTV 由于 F^H 占 F 的比例很低,只有部分 I-帧被冗余编码,包丢失时大部分信息无法恢复,传输最不可靠。

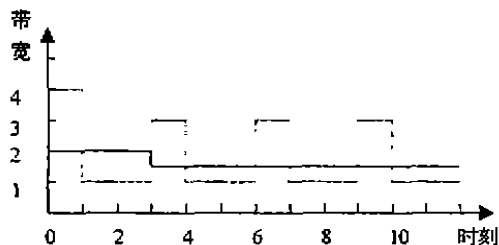


图6 串并行发送比较

图6是对电视广告流进行 RS 编码后两子流并行发送时的带宽需求情况(实线),虚线是采用传统串行发送方式时的带宽需求,最大值为4个带宽单位(为方

便比较纵坐标采用带宽单位为量度,每个带宽单位为 572kb/s);而并行方式的最大带宽需求为2个带宽单位,相对串行方式同一时刻可发送更多的视频流,提高了网络效率。

结论 本文提出一个基于 MPEG 视频帧拆分的提高 MPEG 传输 QoS 的方案,该方案以帧为单位进行拆分,编码和发送,区别于其他的拆分和传输方案,特别适合于已生成流(如在 VoD 系统中)的使用。针对 MPEG 使用帧间预测技术的特点,拆分器尽量将重要的 I-P-帧信息保留在高优先级子流中,有效减少了因重要帧丢失而引起的错误扩散现象。实验结果表明,高优先级子流的数据比例越高,传输越可靠;流的数据冗余率越高,对网络包丢失的抵抗能力越强,视频质量越好。本文还对拆分后子流的发送方法进行了优化,在保证可靠传输的同时提高了网络效率。

参考文献

- 1 Wang Y, Zhu O F. Error control and concealment for video communication-a review. In: Proc. IEEE 86(5) 1998. 974~997
- 2 Sorger U K A new Reed-Soloman decoding algorithm based on Newton's interpolation. IEEE Trans. Inform. Theory, 1993. 39(March): 358~365
- 3 Horn H, Girod B. Scalable video transmission for the internet. Comput. Network ISDN systems, 1997, 29(15): 1833~1842
- 4 Aravind R, Civanlar M, Reibman A. Packet loss resilient of MPEG-2 scalable video coding algorithms. IEEE Trans. Circuits Systems Video Technol, 1996, 6(5): 426~435
- 5 Talluri R. Error-resilient video coding in the ISO MPEG-4 standard. IEEE Comm. Mag., 1998, 36(6): 112~119
- 6 ITU-T, Rec. H. 263, Video Coding for Low Bitrate Communication, Version 2, January 1998
- 7 Horn U, et al. Robust Internet video transmission based on scalable coding and unequal error protection. Image Communication, 1999, 15: 77~94

科学技术贵以奉献与共享
《计算机科学》夙愿做益友

欢迎阅读/订阅《计算机科学》