



(12) 发明专利

(10) 授权公告号 CN 102231837 B

(45) 授权公告日 2013.06.12

(21) 申请号 201110170067.6

(22) 申请日 2011.06.23

(73) 专利权人 西交利物浦大学

地址 215123 江苏省苏州市工业园区独墅湖
高等教育区仁爱路 111 号

(72) 发明人 肖继民 罗天明

(74) 专利代理机构 苏州创元专利商标事务所有
限公司 32103

代理人 范晴

(51) Int. Cl.

H04N 7/64 (2006.01)

H04N 7/26 (2006.01)

审查员 龙玄耀

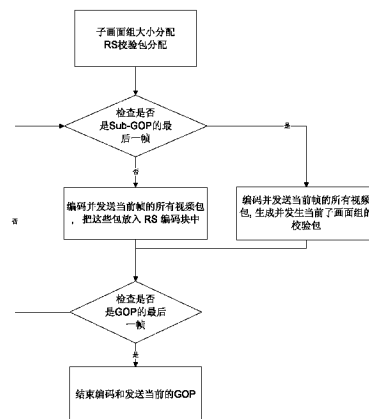
权利要求书1页 说明书6页 附图5页

(54) 发明名称

基于子画面群的前向纠错实时视频传输方法

(57) 摘要

本发明公开了一种基于子画面群的前向纠错实时视频传输方法,其特征在于所述方法中视频采用(N, K) Reed-Solomon 编码器进行编码,包括以下步骤:(1) 视频发送端将视频画面群中所有单幅影像画面的帧依次序划分成若干个子画面群;采用(N, K) Reed-Solomon 编码器以子画面群为编码单位将视频单幅影像画面的帧编码生成数据包立即放入网络传送,并在每个子画面群数据包的最后一帧基于整个子画面群产生校验包,将生成的校验包放入网络传输;(2) 视频接受端接收到数据包后立即解码,如果数据包未丢失,则无需处理;如果丢失的数据包小于N-K时,进行数据包恢复,重新解码。该方法不会造成任何时延,实现了实时通信。



1. 一种基于子画面群的前向纠错实时视频传输方法,其特征在于所述方法中视频采用(N, K) Reed-Solomon 编码器进行编码,包括以下步骤:

(1) 视频发送端将视频画面群中所有单幅影像画面的帧依次序划分成若干个子画面群;采用(N, K) Reed-Solomon 编码器以子画面群为编码单位将视频单幅影像画面的帧编码生成数据包立即放入网络传送,并在每个子画面群的最后一帧基于整个子画面群产生校验包,将生成的校验包放入网络传输;其中通过贪婪算法依次确定所有校验包中每个校验包在视频单幅影像画面的帧的位置,根据校验包的位置确定子画面群的分配情况;

(2) 视频接受端接收到数据包后立即解码,如果数据包未丢失,则无需处理;如果丢失的数据包小于N-K时,进行数据包恢复,重新解码。

2. 根据权利要求1所述的方法,其特征在于所述方法中当视频接受端尚未接受到子画面群的校验包时,采用错误掩盖技术处理丢失的数据包。

3. 根据权利要求1所述的方法,其特征在于所述方法中当视频接受端接受到子画面群的校验包时,使用RS解码器恢复该子画面群中丢失的所有数据包,并重新解码整个子画面群,并更新参考帧缓存。

4. 根据权利要求1所述的方法,其特征在于所述方法中所述贪婪算法是根据校验包放在单幅影像画面帧的不同的位置,获得不同子画面群分配造成的误差期望;当形成最小误差期望时,所放置的单幅影像画面的位置为该校验包的位置;然后继续寻找下一个校验包的位置。

5. 根据权利要求1所述的方法,其特征在于所述方法中校验包的数量等于系统RS(N, K) 编码中参数N和K的差值。

基于子画面群的前向纠错实时视频传输方法

技术领域

[0001] 本发明属于视频编码技术领域,具体涉及一种基于子画面群的前向纠错实时视频传输方法。

背景技术

[0002] 与 H. 263、MPEG2 视频编码标准相比, H. 264/AVC 视频编码标准提供了更高的编码效率和更强的网络适应性。但是,与以往的视频压缩标准相同的是, H. 264/AVC 是建立在混合编码方法基础上的融合运动补偿预测算法的变换编码。因此,经过 H. 264 编码的视频码留在丢包网络中传输时会受到误差扩散的影响,并由此产生严重的漂移现象。

[0003] 目前,各种抗误码技术已经被广泛应用,使得不可靠网络传送视频信号变得可行。这些技术包括帧内宏块刷新 (Intra Macroblock Refreshment), 自动重传请求 (ARQ), 基于反馈的参考帧选择 (RPS), 冗余图像编码 (使用相同或低于原图像质量的冗余) 和多描述编码 (MDC), 以及前向纠错 (FEC) 编码技术。在这些抗误码技术中,帧内宏块刷新,冗余图像编码和多描述编码不会产生额外延时,因此更适合实时视频通信。但是,就帧内宏块刷新技术而言,由于帧内编码效率远低于帧间的编码效率,导致整体编码效率显著降低。在冗余图像编码和多描述编码中,当冗余版本替代了最初版本或者当一些描述在传输过程中被丢失时,就会造成失配误差。而失配误差将会传播至整个画面群 (GOP)。

[0004] 由于网络往返时延,自动重传请求和参考帧选择 (RPS) 技术都会产生较长延时。因此,这两种技术都无法实践于实时视频通信中。而在前向纠错技术 (FEC) 中,时延的长短取决于 FEC 编码数据块的大小。在 E. Baccaglini, T. Tillo, and G. Olmo. Slice sorting for unequal loss protection of video streams. Signal Processing Letters, IEEE, 2008. 中, Reed-Solomon 编码块包含了整个画面群,并且产生了一个图片群的延时。在 X. Yang, Ce Zhu, Zheng Guo Li, Xiao Lin, and Nam Ling. An unequal packet loss resilience scheme for video over the internet. Multimedia, IEEE Transactions on, 7(4) :753-765, 2005. 中, RS 编码块包含了 1 个子画面群的帧,因此延迟取决于子画面群的长度。在 N. Thomos, S. Argyropoulos, N. V. Boulgouris, and M. G. Strintzis. Robust transmission of h. 264/avc video using adaptive slice grouping and unequal error protection. In Multimedia and Expo, 2006 IEEE International Conference on, pages 593-596, 2006. 中, RS 编码应用在每一帧中,因此没有产生前向纠错编码延时。然而,当 RS 编码块只包含一帧的数据包时,由于数据包数量太少,RS 保护的效率并不高。

[0005] RS 编码已被广泛应用在前向纠错编码中,以此来减少数据包在丢包网络中丢失所带来的影响。在 RS(N, K) 编码中,在 K 个源数据包中引入 N-K 个校验包来保护源数据包。只要接收端接收到 N 个数据包中的 K 个,所有的源数据包就可以被还原。就 RS 编码而言,对于相同的编码率 K/N,增加 K 的值会提高 RS 编码的性能。但是,高 K 值会造成大的 RS 编码块,并因此导致长时延。对于视频会议这样的实时应用,这样的时延是不可接受的。

发明内容

[0006] 本发明目的在于提供一种基于子画面群的前向纠错实时视频传输方法,解决了现有技术中,当视频编码基于每帧进行校验纠错导致纠错性能低,同时当以整个画面组进行校验纠错时,时延大的问题。

[0007] 为了解决现有技术中的这些问题,本发明提供的技术方案是:

[0008] 一种基于子画面群的前向纠错实时视频传输方法,其特征在于所述方法中视频采用 (N, K) Reed-Solomon 编码器进行编码,包括以下步骤:

[0009] (1) 视频发送端将视频画面群中所有单幅影像画面的帧依次序划分成若干个子画面群;采用 (N, K) Reed-Solomon 编码器以子画面群为编码单位将视频单幅影像画面的帧编码生成数据包立即放入网络传送,并在每个子画面群数据包的最后一帧基于整个子画面群产生校验包,将生成的校验包放入网络传输;

[0010] (2) 视频接受端接受到数据包后立即解码,如果数据包未丢失,则无需处理;如果丢失的数据包小于 $N-K$ 时,进行数据包恢复,重新解码。

[0011] 优选的,所述方法中当视频接受端尚未接受到子画面群的校验包时,采用错误掩盖技术处理丢失的数据包。

[0012] 优选的,所述方法中当丢失的数据包大于 $N-K$ 时,放弃解码。

[0013] 优选的,所述方法中当视频接受端尚接受到子画面群的校验包时,使用 RS 解码器恢复该子画面群中丢失的所有数据包,并重新解码整个子画面群,并更新参考帧缓存。

[0014] 优选的,所述方法步骤 (1) 中通过贪婪算法依次确定所有校验包中每个校验包在视频单幅影像画面的帧的位置,根据校验包的位置确定子画面群的分配情况。

[0015] 优选的,所述方法中所述贪婪算法是根据校验包放在单幅影像画面帧的不同的位置,获得不同子画面组分配造成的误差期望;当形成最小误差期望时,所放置的单幅影像画面的位置为该校验包的位置;然后继续寻找下一个校验包的位置。

[0016] 优选的,所述方法中校验包的数量等于系统 $RS(N, K)$ 编码中参数 N 和 K 的差值。

[0017] 具体的,视频发送端根据贪婪算法先进行子画面组的多少、大小分配以及 RS 校验包位置和多少的分配;子画面组确定后,开始进行第一个子画面组 (Sub-GOP) 的视频编码(每一帧图像进行编码),检查编码所在的帧是否是子画面组 (Sub-GOP) 的最后一帧。当编码所在的帧是子画面组 (Sub-GOP) 的最后一帧时,编码当前帧的所有视频数据包,并生成当前子画面组的校验包;当编码所在的帧不是 Sub-GOP 的最后一帧时,编码当前帧的所有视频包,把这些包直接放入 RS 编码块中。然后进行下一个子画面组 (Sub-GOP) 的视频编码,直到所有子画面组 (Sub-GOP) 编码完毕。当最后子画面组的最后一帧编码完毕时,结束视频发送端编码过程。

[0018] 具体的,视频接收端从网络上接收视频数据包和校验包;一旦接收完一个数据包,检查数据包是否是某一子画面组的最后一帧。当数据包不是某一子画面组的最后一帧时,解码当前的视频包,对于丢失的包使用错误掩盖技术恢复;当数据包是某一子画面组的最后一帧时,尝试使用接收到的 RS 校验包恢复所有的数据包,并使用恢复的数据包重新解码整个子画面组,更新参考帧缓存。当接收的数据包不是视频 (GOP) 的最后一帧时,继续进行下一个数据包的接收和解码。当接收的数据包是视频 (GOP) 的最后一帧时,结束所有视频的解码。

[0019] 本发明技术方案采用 (N, K) RS 编码器进行编码,特别是在发送端采用了系统 RS 编码,源数据包并未改变,因此视频编码器生成的数据包将立即放入网络传送,而子画面群的最后一帧后产生基于整个子画面组的校验包在最后一帧的数据包后也放入网络传送;在接收端,对于子画面群中的非最后一帧数据包,接收到数据包后立即解码,对于丢失的数据包,使用错误掩盖技术处理;同时,当子画面群中的最后一帧的校验包到达时,尝试使用 RS 解码器恢复该子画面群中丢失的所有数据包,并重新解码整个子画面群,并更新参考帧缓存。

[0020] 本发明使用系统 RS 码 (systematic Reed-Solomon code) 来保护视频包。为了增加 RS 编码块的大小以提高 RS 码的性能,本发明采用了子画面群 (通常多于 1 帧) 作为 RS 编码块,同时,由于使用了系统 RS 码,所以不会造成任何编码时延。在编码端,数据包在进行系统 RS 编码时并未改变,而是在子画面群末尾增加了校验包,因此不会产生编码时延。另外,在解码端,视频解码器解码和显示一帧图片只需要与此帧相关的数据包。如果这一帧中的某些包在传输过程中丢失,误码掩盖技术将会被用来掩盖丢失包。这样,解码器就不需要等待所有属于这个子画面群的包。因此,解码端也不会产生时延。之后,当这个子画面群中所有的包都传输完毕,RS 解码器会尝试还原丢失包。如果这个子画面群中的丢失包少于 $(N-K)$,RS 解码器可以还原子画面群的所有数据包。视频解码器将会利用接受到的数据包和还原的数据包对这个子画面群进行重新解码并更新参考帧,所以误码掩盖失真不会传播到之后的帧。

[0021] 本发明使用基于子画面组的 RS 保护比基于每帧的 RS 保护要效果更好,但是如何划分子画面组并分配 RS 校验包还需确定。 K 是编码前数据包的数量, $N-K$ 是校验包的数量。一方面,如果子画面组包括了太少的帧,RS 编码中 K 的值不够大,RS 效率不高;另一方面,如果子画面组包括了太多的帧,解码端需到子画面组的最后才能使用 RS 包,这样使得子画面组中前面部分帧的质量比较差,所以需要合理的设置子画面组的大小。针对这个问题提出了基于贪婪算法的解决方案:假设在一个画面组中有 L 个帧,RS 校验包的数量是 R ,贪婪算法的基本思想是每次只分配一个 RS 校验包,并确保这次分配的结果是局部最优的。对于每个 RS 校验包,算法将尝试把它放在可能的 L 个位置,放在每个不同的位置将导致不同子画面组分配,造成的误差期望也将不同,贪婪算法将选择造成最小误差期望的位置放入该 RS 校验包,并针对每个 RS 校验包执行同样的操作。

附图说明

[0022] 下面结合附图及实施例对本发明作进一步描述:

[0023] 图 1 为本发明实施例子画面群的前向纠错实时视频传输方法原理图;

[0024] 图 2 为本发明实施例视频发送端的流程图;

[0025] 图 3 为本发明实施例视频接收端的流程图;

[0026] 图 4 为两个子画面组 (Sub-GOP) 和 RS 校验包分配的示意图,其中每个帧有 5 个分片;

[0027] 图 5 为两个子画面组 (Sub-GOP) 和 RS 校验包分配的示意图,其中每个帧有 10 个分片;

[0028] 图 6 为 Foreman 测试序列 PSNR- 码率曲线;

[0029] 图 7 为 Bus 测试序列 PSNR- 码率曲线。

具体实施方式

[0030] 以下结合具体实施例对上述方案做进一步说明。应理解,这些实施例是用于说明本发明而并不限于限制本发明的范围。实施例中采用的实施条件可以根据具体厂家的条件做进一步调整,未注明的实施条件通常为常规实验中的条件。

[0031] 实施例 子画面组的前向纠错实时视频传输实施例

[0032] 本实施例采用了 H. 264JM 软件来生成视频流,在生成视频包的时候采用了固定包长度的方式 (Fixed Slice Length),考虑到了无线网络的最大包长度 (MTU),将目标视频包大小设成了 400Byte,每个分片 (Slice) 放在一个网络包中传送。在 H. 264 预测编码过程中只使用一个编码参考帧。H. 264 画面组 (GOP) 的结构为 IPPP...,也就是说在一个画面组中除了第一帧是 I 帧外,其他的帧都为 P 帧。测试所用的视频序列为标准的 CIF 格式 Foreman 和 Bus 序列。

[0033] 如图 1 所示进行视频的子画面群的划分原理图。通过贪婪算法进行划分子画面群。贪婪算法的步骤如下所示:

[0034]

```
for i = 1 to L do
```

```
    R(i) ← 0
```

[0035]

```
end for
```

```
for j = 1 to R do
```

```
    index ← 0
```

```
    distortion ← ∞
```

```
    for i = 1 to L do
```

```
        R(i) ← R(i) + 1
```

```
        重新计算该 GOP 总的期望误差 (D_total)
```

```
        R(i) ← R(i) - 1
```

```
        if D_total ≤ distortion then
```

```
            index ← i
```

```
            distortion ← D_total
```

```
        end if
```

```
    end for
```

```
    R(index) ← R(index) + 1
```

```
end for
```

[0036] 其中 L 为一个画面组中 P 帧的数量, R(i) 为帧 i 中分配的校验包数量。

[0037] 如图 1 所示,划分得到三个子画面组,第一个子画面组包含了 P 帧 1、P 帧 2 和 P 帧 3 共 3 幅帧,后面二个画面组各包含了 2 幅帧,所有的 RS 包都放在子画面组的最后。

[0038] 如图 2 所示,视频发送端根据贪婪算法先进行子画面组的多少、大小分配以及 RS 校验包位置和多少的分配;子画面组确定后,开始进行第一个子画面组 (Sub-GOP) 的视频编码 (每一帧图像进行编码),检查编码所在的帧是否是子画面组 (Sub-GOP) 的最后一帧。当编码所在的帧是子画面组 (Sub-GOP) 的最后一帧时,编码当前帧的所有视频数据包,并生成的当前子画面组的校验包;当编码所在的帧不是 Sub-GOP 的最后一帧时,编码当前帧的所有视频包,把这些包直接放入 RS 编码块中。然后进行下一个子画面组 (Sub-GOP) 的视频编码,直到所有子画面组 (Sub-GOP) 编码完毕。当最后子画面组的最后一帧编码完毕时,结束视频发送端编码过程,发送当前的视频 (GOP) 的所有视频数据包和校验包。

[0039] 如图 3 所示,视频接收端从网络上接收视频数据包和校验包;一旦接收完一个数据包,检查数据包是否是某一子画面组的最后一帧。当数据包不是某一子画面组的最后一帧时,解码当前的视频包,对于丢失的包使用错误掩盖技术恢复;当数据包是某一子画面组的最后一帧时,尝试使用接收到的 RS 校验包恢复所有的数据包,并使用恢复的数据包重新解码整个子画面组,更新参考帧缓存。当接收的数据包不是视频 (GOP) 的最后一帧时,继续进行下一个数据包的接收和解码。当接收的数据包是视频 (GOP) 的最后一帧时,结束所有视频的解码。

[0040] 如图 1 中第一个画面组,如果在帧 1 和帧 2 中存在丢包,由于该系统是实时的,此时第一个画面组的 RS 校验包并未到达,所以不能尝试利用 RS 包复原所丢失的包,即只能使用错误掩盖技术。当第三帧和 RS 包到达接受端后,RS 解码器会尝试复原所有丢失的包。对于该子画面群,使用了三个 RS 校验包,因此可以复原三个网络丢包。如果 RS 校验包能够复原所有的丢包,视频解码器会利用接收到的数据包和复原的数据包重新解码该子画面组,这样 P 帧三中就不存在丢包误差,同时帧一和帧二的丢包误差也不扩散到后面的帧中。

[0041] 所述子画面群的数目根据贪婪算法形成的,把整个子画面组作为编码单元进行 RS 编码,例子中是以子画面组为单元进行 RS 编码的,如图 1 中,第五帧下面的校验包是用来保护第四和第五帧的。这里第四和第五帧就是一个子画面组。

[0042] 后面两个子画面组包含了两个帧,假如每个帧将生成 5 个数据包。RS 编码的冗余率是 20%,这也意味着对于这个子画面组的 10 个数据包,将使用 2 个 RS 校验包,也就是使用 (12, 10) RS 编码。在一个子画面组中,当视频编码器编码完第一个帧,实时通信系统马上发送生成的 5 个数据包。由于网络丢包,假设这 5 个包中的 2 个在传送过程中丢失。当这 3 个数据包成功到达接收端时,接收端解码并显示该帧,对于丢失的 2 个包,接收端将使用错误掩盖技术。这说明在接收端没有任何解码和显示时延。接下来,第二个帧也将产生 5 个数据包,同时由于第二帧是这个子画面组的最后一个帧,系统将基于这个子画面组的 10 个数据包生成 2 个 RS 校验包。假设这次 5 个数据包和 2 个 RS 校验包都顺利地到达接收端,在接收端 RS (12, 10) 解码器将能够复原第一帧丢失的 2 个数据包。然后视频解码器将接收到的和复原的数据包重新解码,并更新参考帧。在该情况下,这个子画面组的第二帧将没有任何误差,同时也没有任何误差将传播到后面的帧中。

[0043] 现有技术为了达到视频的实时传输,往往给每个帧都加 RS 校验包,对于同样的冗余率 20%,每个帧将使用 RS (6, 5) 编码,这意味着对于第一帧,RS 码就不能够复原丢失的

2 个数据包,最终掩盖误差将扩散到第二帧以及后面的所有帧中,严重影响整个画面组的视频质量。基于子画面组的 RS 保护和基于每帧的 RS 保护都没有造成任何时延,也就是实现了实时通信,但基于子画面组的 RS 保护能够有效的阻止误差扩散,效果明显。

[0044] 传输子画面群时,未接受校验包时,采用复制前面帧相同位置像素值的办法进行错误掩盖,这种方法是普遍使用的错误掩盖方法(Temporal Replacement);数据包的恢复过程就是 RS 的解码过程,也是个标准的过程。

[0045] 在图 4 和图 5 给出了两个子画面组(Sub-GOP)和 RS 校验包分配的例子,在实例中假设丢包率 p 为 5%,一个 GOP 中有 30 个 P 帧,RS 包冗余率为 20%,图 4 中假设每个帧有 5 个分片(slice),图 3 中假设每个帧有 10 个分片。可以看出,使用该算法分配具有以下特点,在画面组的前面分配较多的 RS 校验包,而在画面组的后面部分分配较少的 RS 校验包。这是因为误差扩散时候,位于画面组前面部分的误差会扩散到后面的帧中。同时,可以发现画面组前面部分分配的子画面组比较大,后面分子画面组比较小,这同样也是由于前面部分的帧误差扩展路径较长,需要使用较大的子画面群来更加有效的切断误差传播。

[0046] 在图 6 和图 7 中,把本实施例和平均分配 RS 包 FEC 方案和 RS-MDC[T. Tillo, M. Grangetto, and G. Olmo. Redundant slice optimal allocation for h.264 multiple description coding. Circuits and Systems for Video Technology, IEEE Transactions on, 18(1):59-70, 2008.]进行了 PSNR 性能对比,其中 Y 轴显示的是 PSNR, X 轴是数据码率。为了公平的比较,选用的平均分配 RS 包 FEC 方案和 RS-MDC 都具有实时的效果。可以看出在相同码率的情况下,本实施例基于子画面组的 FEC 方案可以提供更高的 PSNR,也就是更好的视频质量。图 6 为 Foreman 测试序列 PSNR-码率曲线,网络丢包率为 5%,校验包率 20%。图 7 为 Bus 测试序列 PSNR-码率曲线,网络丢包率为 5%,校验包率 20%。

[0047] 上述实例只为说明本发明的技术构思及特点,其目的在于让熟悉此项技术的人是能够了解本发明的内容并据以实施,并不能以此限制本发明的保护范围。凡根据本发明精神实质所做的等效变换或修饰,都应涵盖在本发明的保护范围之内。

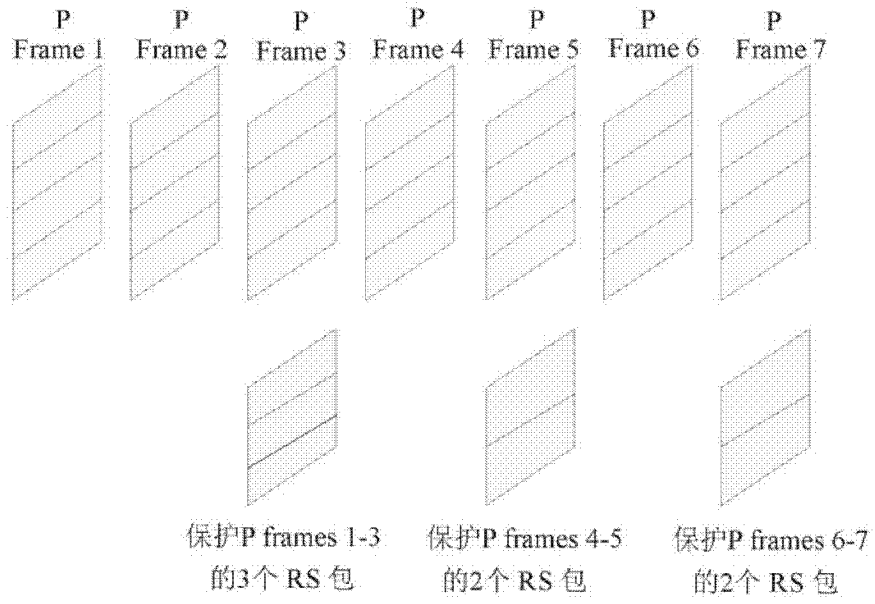


图 1

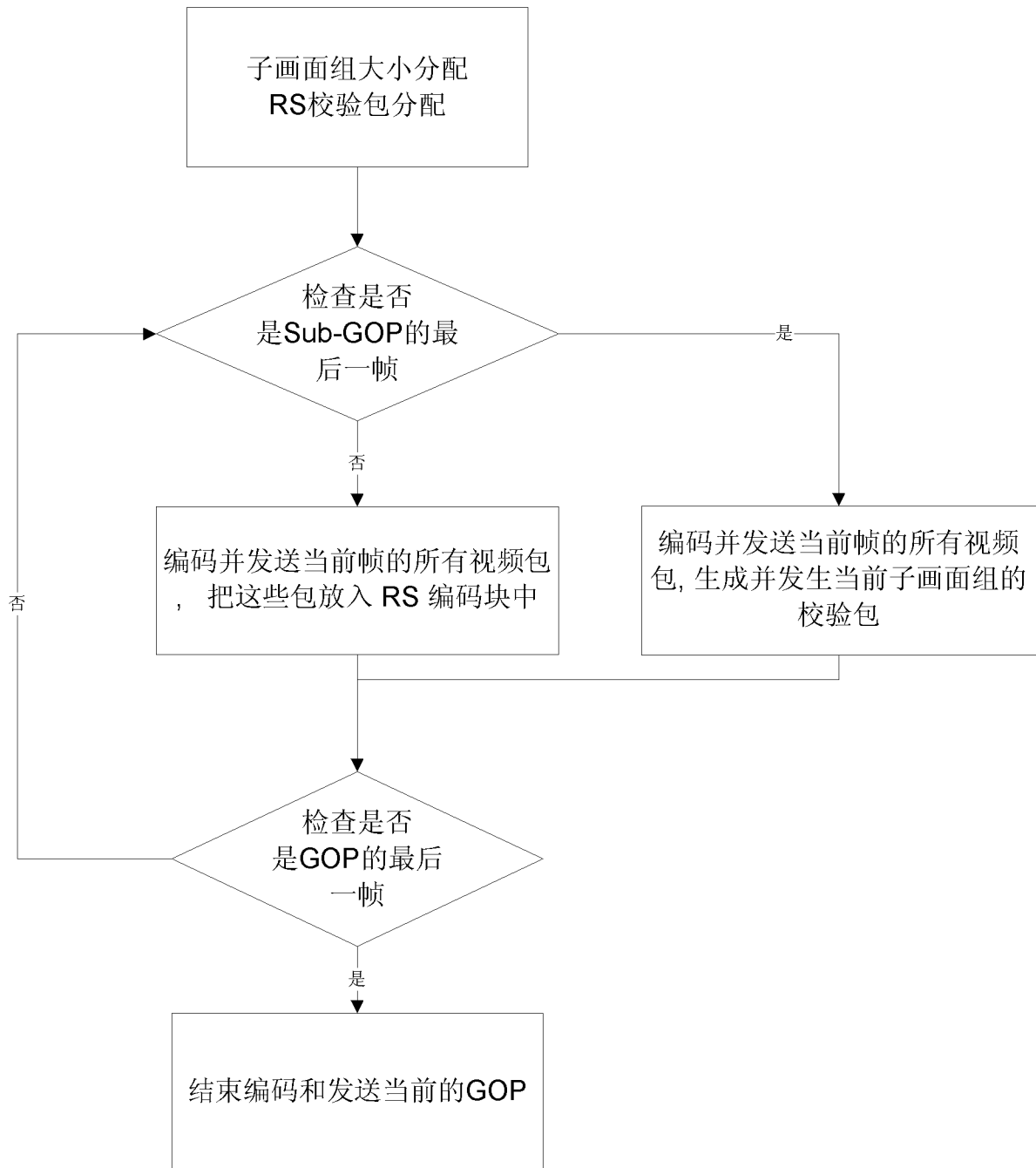


图 2

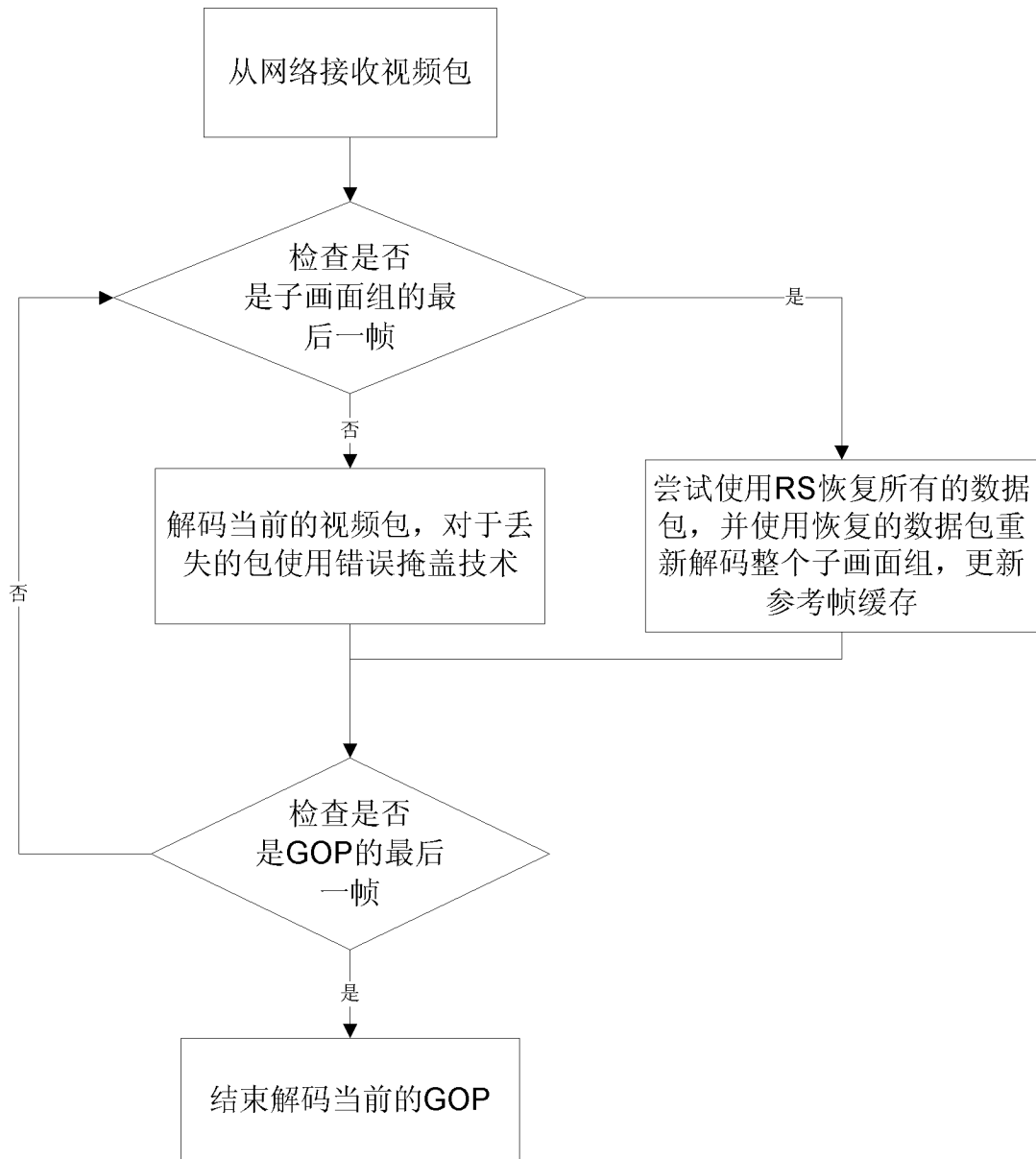


图 3

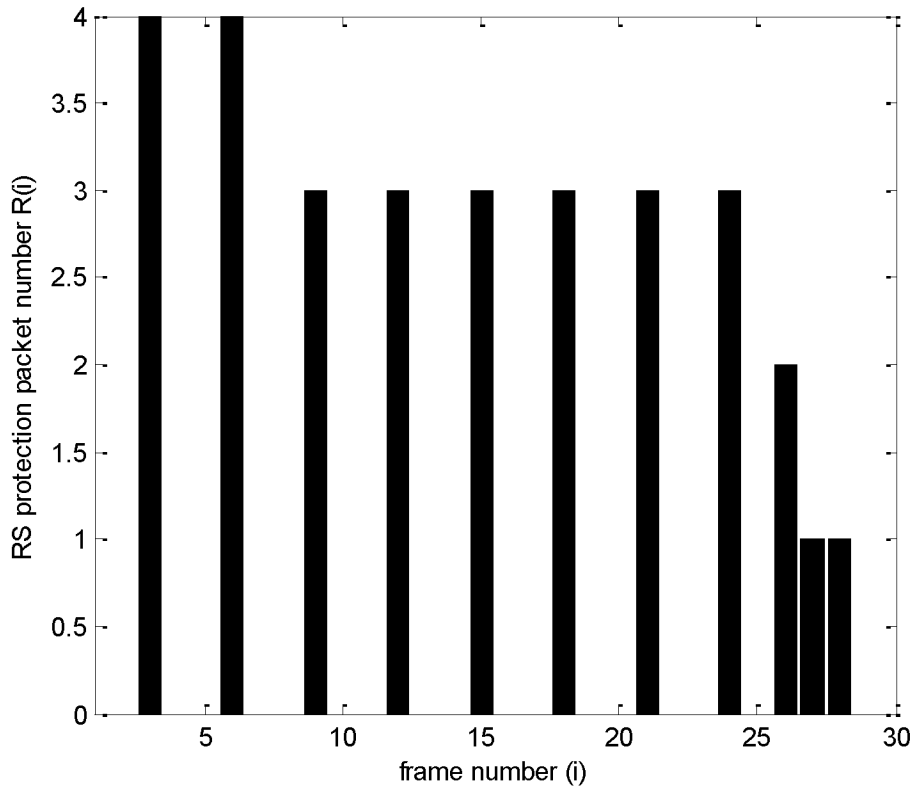


图 4

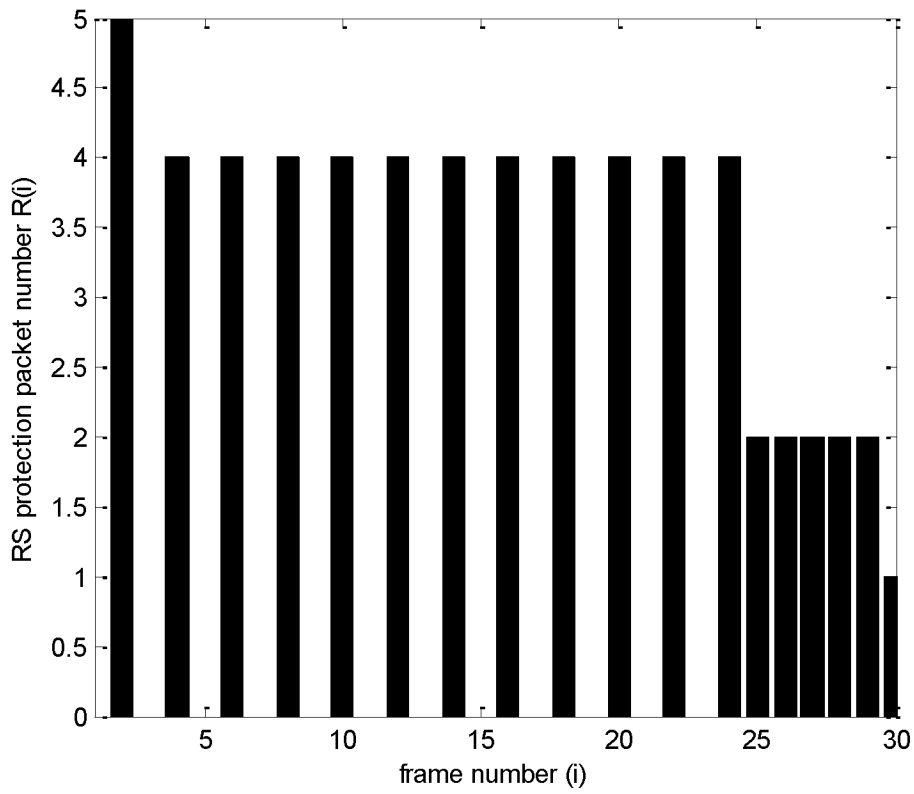


图 5

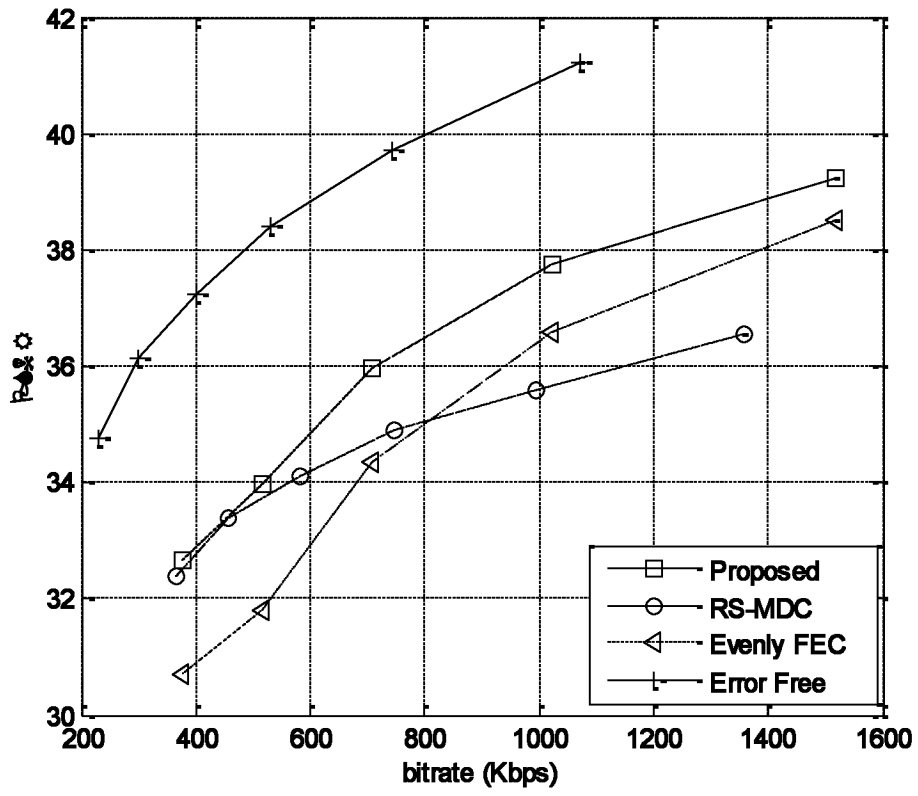


图 6

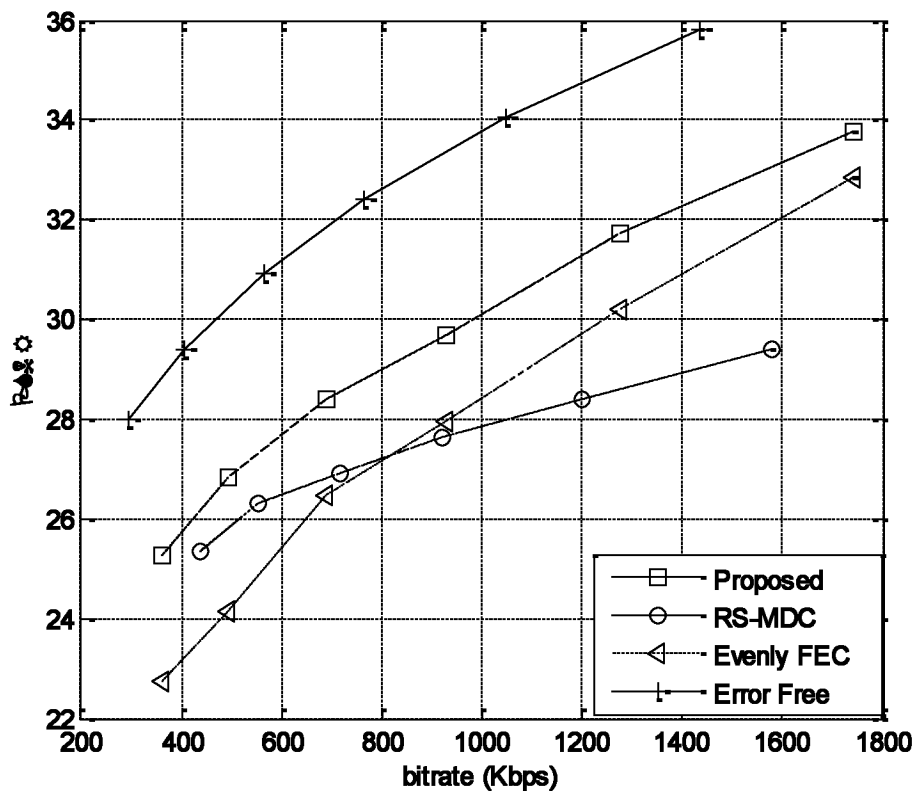


图 7