

一种基于n-bit截尾量化和块内二维预测的参考帧压缩方法

申请号：[201310144865.0](#)

申请日：2013-04-24

申请(专利权)人 [复旦大学](#)
地址 200433 上海市杨浦区邯郸路220号
发明(设计)人 [范益波](#) [尚青](#) [刘聪](#) [曾晓洋](#)
主分类号 [H04N7/26\(2006.01\)I](#)
分类号 [H04N7/26\(2006.01\)I](#) [H04N7/34\(2006.01\)I](#)
公开(公告)号 103248891A
公开(公告)日 2013-08-14
专利代理机构 [上海正旦专利代理有限公司](#) 31200
代理人 [陆飞](#) [盛志范](#)



(12) 发明专利

(10) 授权公告号 CN 103248891 B

(45) 授权公告日 2016. 04. 20

(21) 申请号 201310144865. 0

(22) 申请日 2013. 04. 24

(73) 专利权人 复旦大学

地址 200433 上海市杨浦区邯郸路 220 号

(72) 发明人 范益波 尚青 刘聪 曾晓洋

(74) 专利代理机构 上海正旦专利代理有限公司

31200

代理人 陆飞 盛志范

(51) Int. Cl.

H04N 19/176(2014. 01)

H04N 19/42(2014. 01)

(56) 对比文件

CN 102740068 A, 2012. 10. 17,

CN 102595127 A, 2012. 07. 18,

CN 102665078 A, 2012. 09. 12,

Vivienne Sze, Madhukar Budagavi. High

Throughput CABAC Entropy Coding in HEVC. 《IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY》. 2012, 第 22 卷 (第 12 期), 1778-1791.

审查员 刘冰洁

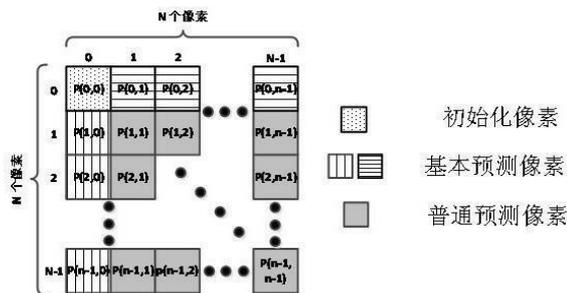
权利要求书1页 说明书5页 附图2页

(54) 发明名称

一种基于 n-bit 截尾量化和块内二维预测的参考帧压缩方法

(57) 摘要

本发明属于数字视频信号编解码技术领域, 具体为一种基于 n-bit 截尾量化和块内二维预测的参考帧压缩方法。该方法中, 参考帧图像首先经过 n-bit 截尾量化得到量化后图像和量化残差, 然后对量化后图像进行块内二维预测。本方法可根据使用要求提供不同质量的重构图像, 以便对解码开销和图像质量进行平衡; 适用于最新的视频编码标准, 具有极高的压缩效率, 可以无缝的嵌入到任何通用编解码器中。本方法符合参考帧压缩高速度的要求, 在无损模式先可以达到 50%~60% 的压缩率, 在有损模式下可以达到 60%~75% 的压缩率。



1. 一种基于n-bit截尾量化和块内二维预测的参考帧压缩方法,其特征在于,具体步骤如下:

(1)输入原始像素,并处理为 $M \times M$ 块为单位的存储模式,其中M为4、8、16或32;

(2)量化并存储量化残差:

将输入的原始的N-bit像素数据向右移n位进行截尾量化,移出的最后n位作为量化残差进行存储,所述N为任意bit数的像素数据的位数;

(3)量化后的(N-n)位作为块内预测的输入,块内预测后每个 $M \times M$ 块分为初始化像素 $P(0,0)$ 、基本参考像素 $P(0,y)$ 和 $P(x,0)$ 和普通预测像素 $P(x,y)$ 三种,其中 $x,y=1,2,3,\dots,M-1$;计算基本参考像素残差 $Residual(0,y)$ 和 $Residual(x,0)$,其计算公式如下:

$$Residual(0,x) = P(0,y) - P(0,y - 1);$$

$$Residual(x,0) = P(x,0) - P(x - 1,0);$$

其中 $P(0,y-1)$ 表示 $P(0,y)$ 上方像素点; $P(x-1,0)$ 表示 $P(x,0)$ 左面的像素点;

(4)依据HEVC编码器的帧内预测结果选择合适的块内预测模式,计算普通预测像素残差 $Residue(x,y)$;其中,所述块内预测模式如下所述:模式0选择残差计算 $C-R_1$,模式1选择残差计算 $C-R_1$,模式2选择残差计算 $C-0.5R_1-0.5R_2$,模式3选择残差计算 $C-0.5R_1-0.5R_2$,模式4选择残差计算 $C-0.6R_1-0.4R_2$,模式5选择残差计算 $C-0.5R_1-0.5R_2$,模式6选择残差计算 $C-0.35R_1-0.35R_2-0.3R_3$,模式7选择残差计算 $C-0.3R_1-0.3R_2-0.2R_3-0.2R_4$;

其中,模式0: R_1 位于C左侧,模式1: R_1 位于C上方,模式2: R_1 位于C左侧,且 R_2 位于C左上方,模式3: R_1 位于C上方,且 R_2 位于C右上方,模式4: R_1 位于C左侧,且 R_2 位于C右上方,模式5: R_1 位于C左侧,且 R_2 位于C上方,模式6: R_1 位于C左侧,且 R_2 位于C上方,且 R_3 位于C左上方,模式7: R_1 位于C左侧,且 R_2 位于C上方,且 R_3 位于C左上方,且 R_4 位于C右上方;

其中:C表示当前像素点; R_1 表示第一参考像素点; R_2 表示第二参考像素点; R_3 表示第三参考像素点; R_4 表示第四参考像素点;

所述HEVC编码器的帧内预测结果和块内预测模式的映射如下所述:帧内预测结果 **planar** 选择块内预测模式4、5、7,帧内预测结果DC选择块内预测模式4、5、7,帧内预测结果3-10选择块内预测模式0、4、5,帧内预测结果10-18选择块内预测模式0、2、6,帧内预测结果18-26选择块内预测模式1、5、6,帧内预测结果26-34选择块内预测模式1、3、4;

所述普通预测像素残差 $Residual(x,y)$ 的计算公式如下所示:

$$Residual(x,y) = P(x,y) - \sum \alpha_i \cdot Ref_i;$$

其中 $P(x,y)$ 表示处于 (x,y) 位置的当前像素点, Ref_i 表示当前像素的参考像素, α_i 是其权值,它的取值由块内二维预测模式决定;

(5)选择最佳的残差预测结果,再进行编码,实现参考帧压缩。

一种基于n-bit截尾量化和块内二维预测的参考帧压缩方法

技术领域

[0001] 本发明属于数字视频信号编解码技术领域,针对于HEVC视频标准,具体涉及参考帧压缩的算法。

背景技术

[0002] HEVC(High efficiency video coding)是由国际电信组织(ITU)和运动图像专家组(MPEG)联合制定而成的最新国际视频编码标准。相对于旧的H.264标准,HEVC具有更高的压缩效率,更适合超高分辨率视频的编码,但其计算量与复杂度也剧增,巨大的外存访问量严重降低了系统的速度,增加了系统功耗。对于编解码器来说,外存的访问主要来自参考帧与播放帧的读写,要降低系统外存访问带宽,就要从此处入手。

[0003] 参考帧压缩是解决上述问题的一种有力手段,它对存储在外存的参考帧进行快速高效的压缩,减少参考帧传输时的数据量,从而降低系统带宽。该方法可以无缝的嵌入到任何视频编解码器中,不会对原始视频编解码器结构产生太大影响。附图1中阐述了参考帧压缩模块系统中的位置:1-1为原始视频流的读入,1-2为压缩后的数据存向外存,1-3为编码器重构图像进入参考帧压缩器,1-4为参考帧解压器从外存读入压缩后的参考帧,1-5为将解压好的参考帧送向编码器。可见,该模块的加入不会影响原始编码器的结构。

[0004] 参考帧数据的形式一般为原始的YUV像素。参考帧在解码器中主要用于运动补偿(Motion Compensation),在编码器中用于运动估计(Motion estimation)和运动补偿。在进行运动估计、运动补偿时,编码器需要将参考帧中的原始像素从外存中载入到运动估计模块中进行运动估计。如果使用了参考帧压缩技术,存储在外存中的数据为已经压缩过的,这样载入参考帧时就能大大降低数据量,从而达到降低带宽的目的。从外存中载入的已压缩后数据在使用前需要先经过解压,所以参考帧压缩模块由压缩器和解压器构成。

[0005] 参考帧压缩算法有以下几点要求:

[0006] 1.较高的压缩效率;

[0007] 2.简单的压缩和解压算法,确保不会影响编解码器的工作速度;

[0008] 3.支持数据随机访问,满足编解码器的要求;

[0009] 4. 重构的参考帧无损或低损,保证编解码图像的质量;

[0010] 压缩率是压缩算法的一个重要衡量依据,它定义为:

$$[0011] \quad \text{压缩率} = \frac{\text{压缩后参考帧存大小}}{\text{原始参考帧存储大小}}$$

发明内容

[0012] 本发明的目的在于提出一种快速高效的参考帧压缩方法,适用于HEVC视频规范所规定编解码过程,可以实现多质量层次的压缩和解码。该压缩算法主要包括两个过程:n-bit截尾量化和块内二维预测。

[0013] n-bit截尾量化是指将输入的原始N-bit像素最后n位进行截尾,即将原始的N-bit数据向右移n位(相当于对原始像素进行了量化系数为 2^n 的量化)。移出的最后n位作为量化残差进行存储。解码时,通过使用不同位数的量化残差,可以得到不同品质的重构图像。

[0014] 块内二维预测是指通过结合HEVC的帧内预测结果,在 $M \times M$ 的像素块内对像素进行预测并计算残差, M为4、8、16或32,以此来减小相邻像素间的空间冗余。在进行块内二维预测时,算法需结合HEVC的帧内预测结果,通过合适映射方法将该结果映射到块内预测模式上,以达到好的预测效果,减小残差的大小,增加压缩效率。

[0015] 本发明提供一种基于n-bit截尾量化和块内二维预测的参考帧压缩方法,具体步骤如下:

[0016] (1)输入原始像素数据,并处理为 $M \times M$ 块为单位的存储模式, M为4、8、16或32;

[0017] (2)量化并存储量化残差:

[0018] 将输入的原始的N-bit像素数据向右移n位进行截尾量化,移出的最后n位作为量化残差进行存储;所述N为任意bit数的像素数据的位数;

[0019] (3)上述量化后的(N-n)位作为块内预测的输入,块内预测后每个 $M \times M$ 块分为初始化像素 $P(0,0)$ 、基本参考像素 $P(0,y)$ 和 $P(x,0)$ 和普通预测像素 $P(x,y)$ 三种,其中 $x,y=1,2,3,\dots,M-1$,计算基本参考像素残差 $Residual(0,y)$ 和 $Residual(x,0)$,其计算公式如式(1)、式(2)所示:

$$[0020] \quad Residual(0,x) = P(0,y) - P(0,y - 1); \quad (1)$$

$$[0021] \quad Residual(x,0) = P(x,0) - P(x - 1,0); \quad (2)$$

[0022] 其中: $P(0,y-1)$ 表示 $P(0,y)$ 上方像素点; $P(x-1,0)$ 表示 $P(x,0)$ 左面的像素点;

[0023] (4)依据HEVC编码器的帧内预测结果选择合适的块内预测模式,计算普通预测像素残差 $Residue(x,y)$,其中:

[0024] 所述块内预测模式如表1所示:

[0025] 表1:块内预测模式

模式	残差计算	模式	残差计算
0	$C - R_1$	4	$C - 0.6R_1 - 0.4R_2$
1	$C - R_1$	5	$C - 0.5R_1 - 0.5R_2$
2	$C - 0.5R_1 - 0.5R_2$	6	$C - 0.35R_1 - 0.35R_2 - 0.3R_3$
3	$C - 0.5R_1 - 0.5R_2$	7	$C - 0.3R_1 - 0.3R_2 - 0.2R_3 - 0.2R_4$

[0027] 其中,模式0: R_1 位于C左侧,模式1: R_1 位于C上方,模式2: R_1 位于C左侧,且 R_2 位于C左上方,模式3: R_1 位于C上方,且 R_2 位于C右上方,模式4: R_1 位于C左侧,且 R_2 位于C右上方,模式5: R_1 位于C左侧,且 R_2 位于C上方,模式6: R_1 位于C左侧,且 R_2 位于C上方,且 R_3 位于C左上方,模

式7:R₁位于C左侧,且R₂位于C上方,且R₃位于C左上方,且R₄位于C右上方;

[0028] 其中C表示当前像素点; R₁表示第一参考像素点; R₂表示第二参考像素点; R₃表示第三参考像素点; R₄表示第四参考像素点;

[0029] 所述HEVC编码器的帧内预测结果和块内预测模式的映射如表2所示:

[0030] 表2:块内二维预测模式映射

帧内预测结果	块内预测模式
planar	4, 5, 7
DC	4, 5, 7
3-10	0, 4, 5
10-18	0, 2, 6
18-26	1, 5, 6
26-34	1, 3, 4

[0032] 所述普通预测像素残差Residual(x,y)的计算公式如式(3)所示:

[0033]
$$\text{Residual}(x,y) = P(x,y) - \sum \alpha_i \cdot \text{Ref}_i; \quad (3)$$

[0034] 其中 P(x,y)表示处于(x,y)位置的当前像素,Ref_i表示当前像素的参考像素, α_i 是其权值,它的取值由块内二维预测模式决定;

[0035] (5)选择最佳的残差预测结果,再进行编码,实现参考帧压缩。

[0036] 基于n-bit截尾量化和块内二维预测的参考帧压缩过程如图2所示。

[0037] 通过以上的压缩过程,可以将原始的64个像素用一个原始像素、63个预测残差以及(n*64)位量化残差表示。

[0038] 本发明的有益效果在于:本发明提出的方法适用于最新的视频编码标准,具有极高的压缩效率,可以无缝的嵌入到任何通用编解码器中。算法的复杂度很低,整个压缩过程只涉及到移位和加减运算,符合参考帧压缩高速度的要求。经过该算法编码的数据能实现随即访问,并可根据要求重构出不同质量的图像,方便进行质量和速度的平衡。对于无损压缩模式,该方法可以提供50%~60%的压缩率,而在有损压缩模式下,其压缩率可以达到60%~75%。

附图说明

[0039] 图1:编码器中的参考帧压缩模块。

[0040] 图2:基于n-bit截尾量化和块内二维预测的参考帧压缩流程图。

[0041] 图3:8×8块块内预测结果。

[0042] 图4:块内预测模式图示;其中,a为模式0,b为模式1,c为模式2,d为模式3,e为模式4,f为模式5,g为模式6,h为模式7。

具体实施方式

[0043] 下面结合附图对本发明做进一步的描述。

[0044] 本发明所述的参考帧压缩方法,具体实施方式如下:

[0045] (1)原始像素输入;

[0046] 编码器外部输入的N-bit原始YUV数据首先要处理为 8×8 块为单位的存储模式以便后续处理。

[0047] (2)量化并存储残差;

[0048] 量化过程通过对原始像素进行向右移位来完成,当向右移n位时,其量化系数 $Q=2^n$ 。而每个像素移出的n位将作为其量化残差被存储。而量化后的 $(N-n)$ 将作为块内预测的输入。在重构图像时,如果将量化残差完整加入,则可以得到无损参考帧,如果只是用n为中的前m位,则会重构出有损的图像,但其需要载入的数据量将减少。使用的位数越多,重构图像质量越好,数据量越大;反之,图像质量降低,传输的数据量也随之减少。

[0049] (3)计算基本参考像素残差;

[0050] 如图3所示, $P(0,0)$ 为初始化参考像素, $P(0,y)$ 和 $P(x,0)$ ($x,y=1,2,3,\dots,M-1$)为基本参考像素,其余为普通预测像素。初始化像素以原始的N-bit模式存储,基本参考像素由初始化像素和其本身产生,普通预测像素由初始化像素,基本参考像素以及其本身产生。

[0051] 基本参考像素残差通过式(1)、式(2)计算得到:

$$[0052] \quad \text{Residual}(0,x) = P(0,y) - P(0,y-1); \quad (1)$$

$$[0053] \quad \text{Residual}(x,0) = P(x,0) - P(x-1,0); \quad (2)。$$

[0054] (4)选择合适的预测方式,根据式(3)计算普通预测像素残差;

$$[0055] \quad \text{Residual}(x,y) = P(x,y) - \sum \alpha_i \cdot \text{Ref}_i; \quad (3)$$

[0056] 要得到普通预测像素,就要选择合适的预测方向,其选择过程依据HEVC编码器的帧内预测结果。由于帧内预测的结果包含了图像的方向信息,利用该结果可以很好的消除图像的空间冗余,达到压缩图像的目的,由HEVC帧内预测结果到本方法中块内预测模式的映射如表2所示。

[0057] (5)选择最佳的残差预测结果,再进行编码,实现参考帧压缩。

[0058] 经过上述五个步骤,即可很好地消除图像的空间冗余,得到的 8×8 图像块中每个像素($P(0,0)$ 除外)都由取值很小的残差表示,通过合适方法编码这种残差将大大减小数据量,起到压缩的效果。

[0059] (6)具体实施举例:

[0060] 本例中输入原始像素为8bit的YUV数据,压缩块的大小为 8×8 ,量化位数为 $n=0,1$ 或2。这里采用的7个1080p的视频流进行测试,并得到其压缩率,如下表所示:

[0061] 表3:不同量化系数下的压缩率

[0062]

		向日葵	蓝天	车站 2	河床	拖拉机	人行道	高峰时期
n=0	有损
	无损	0.381616	0.427008	0.484855	0.534561	0.495971	0.416645	0.440156
n=1	有损	0.250115	0.301489	0.344788	0.399781	0.374643	0.294862	0.302265
	无损	0.376041	0.427415	0.439788	0.524781	0.499643	0.419862	0.427265
n=2	有损	0.173653	0.216552	0.222972	0.265164	0.262573	0.188235	0.19134
	无损	0.423653	0.466552	0.472972	0.515164	0.512573	0.438235	0.44134

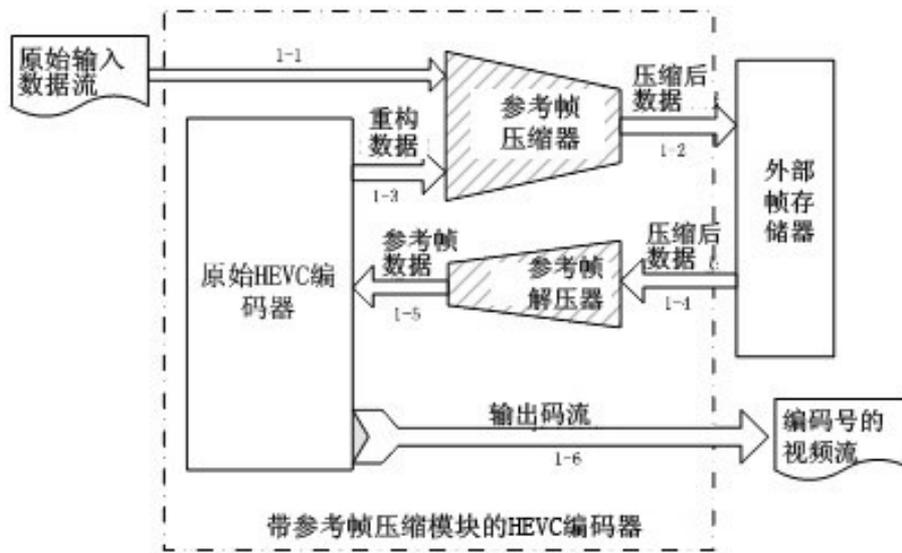


图 1

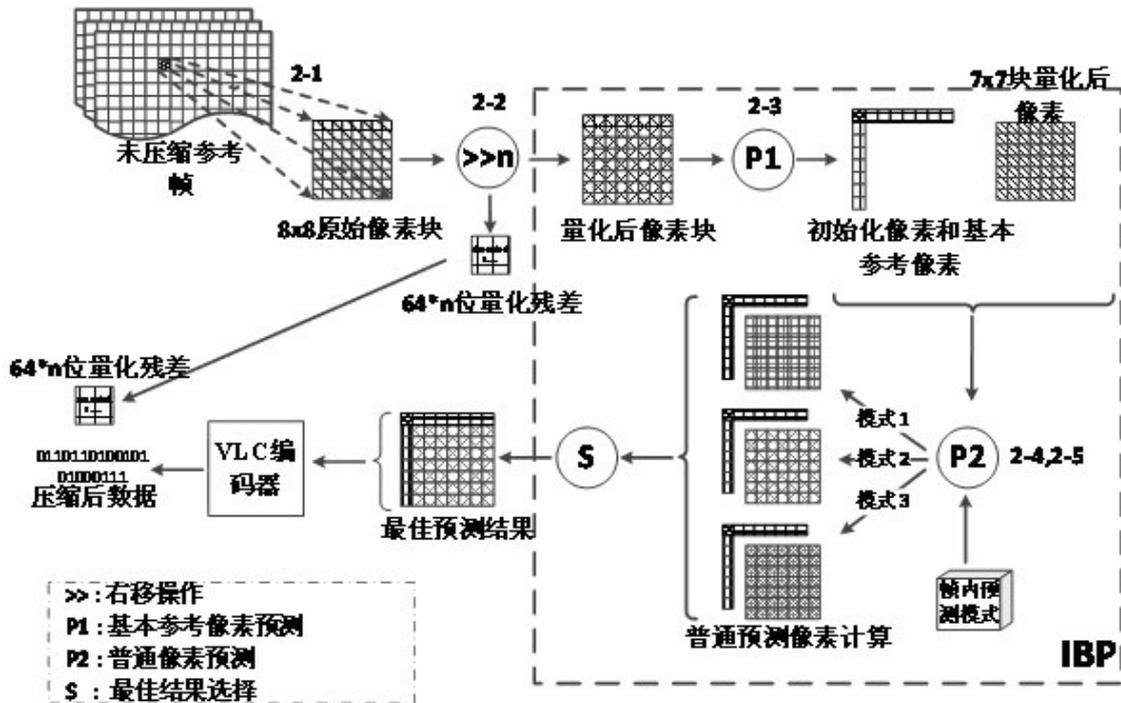


图 2

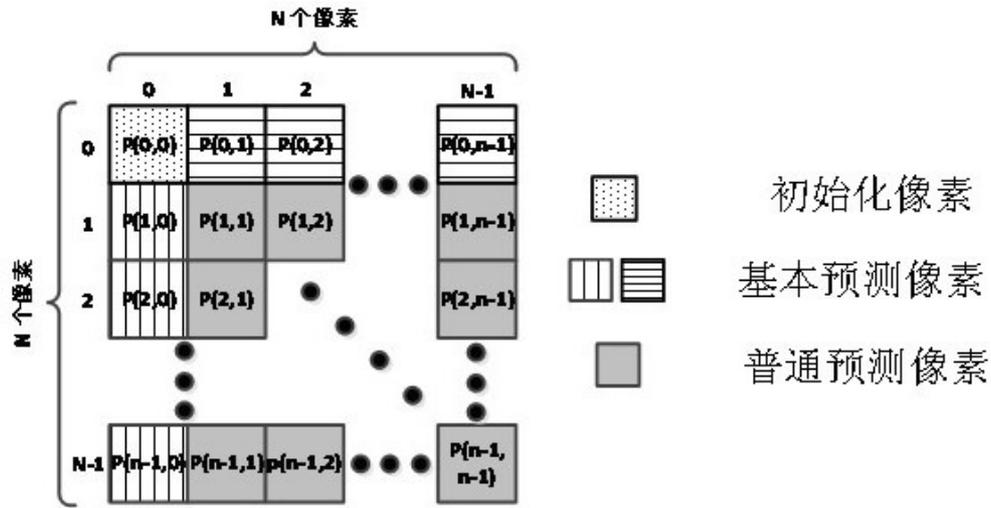
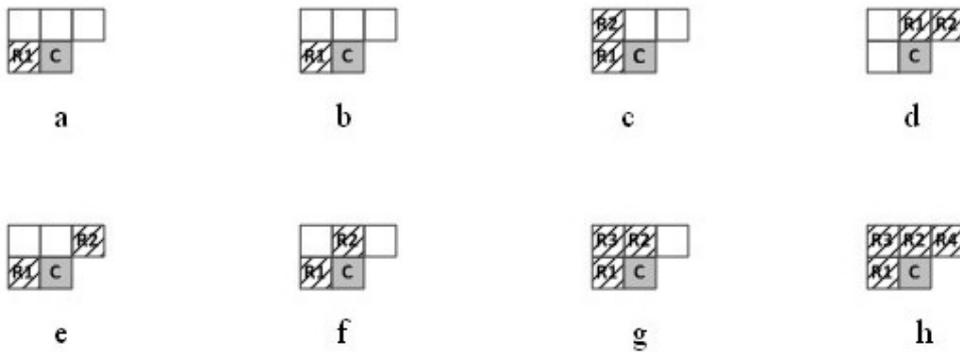


图 3



模式	残差计算	模式	残差计算
0	$C - R_1$	4	$C - 0.6R_1 - 0.4R_2$
1	$C - R_1$	5	$C - 0.5R_1 - 0.5R_2$
2	$C - 0.5R_1 - 0.5R_2$	6	$C - 0.35R_1 - 0.35R_2 - 0.3R_3$
3	$C - 0.5R_1 - 0.5R_2$	7	$C - 0.3R_1 - 0.3R_2 - 0.2R_3 - 0.2R_4$

图 4