



(12) 发明专利

(10) 授权公告号 CN 103096090 B

(45) 授权公告日 2015. 12. 02

(21) 申请号 201310054884. 4

US 2013039414 A1, 2013. 02. 14,

(22) 申请日 2013. 02. 20

CN 102917225 A, 2013. 02. 06,

(73) 专利权人 广州柯维新数码科技有限公司
地址 510006 广东省广州市番禺区小谷围街
外环东路 232 号 13 栋 B301-B302

审查员 冀芊茜

(72) 发明人 梁凡 邱家文 陈焕滨

(74) 专利代理机构 广州科粤专利商标代理有限
公司 44001

代理人 黄培智

(51) Int. Cl.

H04N 19/61(2014. 01)

H04N 19/52(2014. 01)

H04N 19/573(2014. 01)

(56) 对比文件

CN 1585495 A, 2005. 02. 23,

CN 102740078 A, 2012. 10. 17,

CN 102752595 A, 2012. 10. 24,

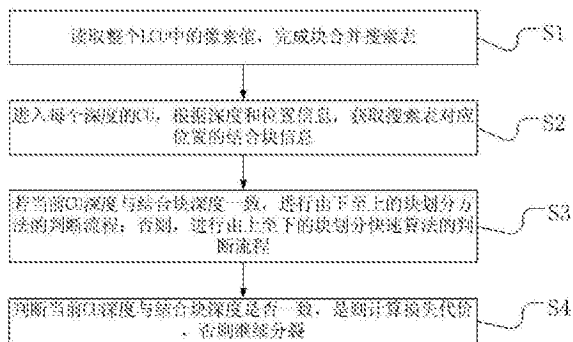
权利要求书2页 说明书7页 附图3页

(54) 发明名称

一种用于视频压缩中的编码块划分的方法

(57) 摘要

本发明公开了一种用于视频压缩中的编码块划分的方法,其特征在于,包括以下步骤:读取整个LCU中的像素值,完成块合并搜索表;进入每个深度的CU,根据深度和位置信息,获取搜索表对应位置的结合块的深度和位置信息;若当前CU深度与结合块深度一致,进行由下至上的块划分方法的判断流程;否则,进行由上至下的块划分快速算法的判断流程。采用本发明结合两个方面提出的块划分快速算法在保证 HEVC 编码器的视频质量和输出码率基本不变的前提下,大大加快了编码速度,提高编码效率。



1. 一种用于视频压缩中的编码块划分的方法,其特征在于,包括以下步骤:

步骤 S1:读取整个 LCU 中的像素值,从最小的 CU 开始,从下至上,根据梯度方向值判断小块能否合并为大块,完成块合并搜索表;所述梯度方向值是指主像素坐标方向梯度的绝对差平均值 Dire_Grad;

步骤 S2:进入每个深度的 CU,根据深度和位置信息,获取搜索表对应位置的结合块的深度和位置信息;

步骤 S3:若当前 CU 深度与结合块深度一致,进行由下至上的块划分方法的判断流程;否则,进行由上至下的块划分快速算法的判断流程;

步骤 S4:判断当前 CU 深度与结合块深度是否一致,是则计算损失代价,否则继续分裂;

若当前 CU 深度与结合块深度一致,进行由下至上的块划分方法的判断流程具体执行如下:

步骤 S31:判断当前 CU 的梯度方向值是否在 (1, 9) 内,是则进行 $N \times 2N$, $nU \times 2N$, $nD \times 2N$ 的块的预测编码,跳到步骤 S4;

步骤 S32:判断当前 CU 的梯度方向值是否在 (9, 16) 内,是则进行 $2N \times N$, $2N \times nU$, $2N \times nD$ 的块的预测编码,跳到步骤 S4;

上述步骤 S31 和步骤 S32 中的梯度方向值如下表所示:

梯度方向的坐标定义

梯度方向	坐标	梯度方向	坐标	梯度方向	坐标	梯度方向	坐标
1	{1, -1}	5	{1, 0}	9	{1, 1}	13	{0, 1}
2	{4, -3}	6	{4, 1}	10	{4, -3}	14	{1, -4}
3	{2, -1}	7	{2, 1}	11	{1, 2}	15	{1, -2}
4	{4, -1}	8	{4, 3}	12	{1, 4}	16	{3, -4}

若当前 CU 深度与结合块深度不一致,进行由上至下的块划分快速算法的判断流程具体执行如下:计算图像块内原像素小块均值与平面预测值之间绝对差的平均值,若该值小于设定阈值,则进行 class A 的预测,否则使用 class B 的预测,其中, class A 为: $A = \{PART_2N \times 2N, PART_N \times N\}$, class B 为: $B = \{PART_2N \times N, PART_N \times 2N, PART_2N \times nU, PART_2N \times nD, PART_nL \times 2N, PART_nR \times 2N\}$ 。

2. 根据权利要求 1 所述的用于视频压缩中的编码块划分的方法,其特征在于,计算图像块内原像素小块均值与平面预测值之间绝对差的方法为: $Diff(x, y) = \text{abs}[piOrg(x, y) - \text{predSample}(x, y)]$, 其中, $Diff(x, y)$ 是座标为 (x, y) 的 2×2 小块均值的预测值与原始值的绝对差, $\text{predSample}(x, y)$ 是坐标点 (x, y) 由上下左右四组参考小块双线性插值的预测值, $piOrg(x, y)$ 是上下左右四组参考数组在坐标点 (x, y) 位置上的参考值。

3. 根据权利要求 2 所述的用于视频压缩中的编码块划分的方法,其特征在于,计算图像块内原像素小块均值与平面预测值之间绝对差的平均值的方法为:

$$Up_Diff = \frac{1}{[(length/2) - 2]^2} \sum_{x=1}^{x=[length/2]-1} \sum_{y=1}^{y=[length/2]-1} Diff(x, y) \quad \text{其中, Up_Diff 是}$$

图像块内原像素小块均值与平面预测值之间绝对差的平均值, Length 是 CU 块的宽度, Diff(x, y) 是坐标为 (x, y) 的 2×2 小块均值的预测值与原始值的绝对差。

一种用于视频压缩中的编码块划分的方法

技术领域

[0001] 本发明涉及视频处理领域,尤其涉及的是一种用于视频压缩中的编码块划分的方法。

背景技术

[0002] 新一代高效率视频编码(HEVC,High Efficiency Video Coding)是由国际标准化组织(ISO)与国际电信联盟远程通信标准化组织(ITU-T)成立的联合协作视频小组(Joint Video Team, JVT)从 2010 年 4 月开始制订的新一代视频编码标准。它采用了大尺寸的最大编码块(LCU)的划分。把 H. 264 中 16×16 尺寸的宏块,扩大到 64×64 尺寸的 LCU 块。HEVC 块结构的划分包括编码单元(CU)的划分、预测单元(PU)的划分以及变换单元(TU)的划分三部分。其中 CU 的划分基于 LCU,根据编码图像的特性以二叉树结构划分为不同大小的 CU,以灵活匹配图像内容,得到最佳编码效果。

[0003] 为适应人们对视频质量日益增长的需求,HEVC 在帧内帧间预测、熵编码、环路滤波和量化变换等各方面都应用了能提高编码效率的新技术,使得 HEVC 在保证视频质量不变的前提下,压缩效率比 H. 264 提高约 50%。高压压缩效率使得 HEVC 有很好的网络适应性,但同时也付出高计算复杂度的代价。因此,提出快速的编码算法来降低 HEVC 编码器的计算复杂度有着积极的意义。

[0004] 因此,现有技术还有待于改进和发展。

发明内容

[0005] 本发明的目的在于提供一种用于视频压缩中的编码块划分的方法,旨在解决现有的 HEVC 编码器计算复杂度高的问题。

[0006] 本发明的技术方案如下:一种用于视频压缩中的编码块划分的方法,其中,包括以下步骤:

[0007] 步骤 S1:读取整个 LCU 中的像素值,从最小的 CU 开始,从下至上,根据梯度方向值判断小块能否合并为大块,完成块合并搜索表;

[0008] 步骤 S2:进入每个深度的 CU,根据深度和位置信息,获取搜索表对应位置的结合块的深度和位置信息;

[0009] 步骤 S3:若当前 CU 深度与结合块深度一致,进行由下至上的块划分方法的判断流程;否则,进行由上至下的块划分快速算法的判断流程;

[0010] 步骤 S4:判断当前 CU 深度与结合块深度是否一致,是则计算损失代价,否则继续分裂。

[0011] 所述的用于视频压缩中的编码块划分的方法,其中,若当前 CU 深度与结合块深度一致,进行由下至上的块划分方法的判断流程具体执行如下:

[0012] 步骤 S31:判断当前 CU 的梯度方向值是否在 (1, 9) 内,是则进行 $N \times 2N$, $nU \times 2N$, $nD \times 2N$ 的块的预测编码,跳到步骤 S4;

[0013] 步骤 S32: 判断当前 CU 的梯度方向值是否在 (9, 16) 内, 是则进行 $2N \times N$, $2N \times nU$, $2N \times nD$ 的块的预测编码, 跳到步骤 S4。

[0014] 所述的用于视频压缩中的编码块划分的方法, 其中, 若当前 CU 深度与结合块深度不一致, 进行由上至下的块划分快速算法的判断流程具体执行如下: 计算图像块内原像素小块均值与平面预测值之间绝对差的平均值, 若该值小于设定阈值, 则进行 classA 的预测, 否则使用 classB 的预测, 其中, classA 为: $A = \{PART_2N \times 2N, PART_N \times N\}$, classB 为: $B = \{PART_2N \times N, PART_N \times 2N, PART_2N \times nU, PART_2N \times nD, PART_nL \times 2N, PART_nR \times 2N\}$ 。

[0015] 所述的用于视频压缩中的编码块划分的方法, 其中, 计算图像块内原像素小块均值与平面预测值之间绝对差的方法为: $Diff(x, y) = \text{abs}[\text{piOrg}(x, y) - \text{predSample}(x, y)]$, 其中, $Diff(x, y)$ 是座标为 (x, y) 的 2×2 小块均值的预测值与原始值的绝对差, $\text{predSample}(x, y)$ 是坐标点 (x, y) 由上下左右四组参考小块双线性插值的预测值, $\text{piOrg}(x, y)$ 是上下左右四组参考数组在坐标点 (x, y) 位置上的参考值。

[0016] 所述的用于视频压缩中的编码块划分的方法, 其中, 计算图像块内原像素小块均值与平面预测值之间绝对差的平均值的方法为:

$$[0017] \quad Up_Diff = \frac{1}{[(length/2) - 2]^2} \sum_{x=1}^{x=[length/2]-1} \sum_{y=1}^{y=[length/2]-1} Diff(x, y) \quad \text{其中, } Up_Diff$$

是图像块内原像素小块均值与平面预测值之间绝对差的平均值, Length 是 CU 块的宽度, $Diff(x, y)$ 是座标为 (x, y) 的 2×2 小块均值的预测值与原始值的绝对差。

[0018] 本发明的有益效果: 本发明通过对 HEVC 编码器中多层次的 CU 划分和多种类型 PU 划分模式进行深入研究, 提出了 HEVC 块划分的快速算法。一方面, 根据当前 CU 内图像内容的复杂度, 本算法对 CU 判断是否跳过当前 CU 或 PU 的预测编码过程。另一方面, 本算法基于图像纹理方向特征, 从小尺寸 CU 到大尺寸 CU 进行块结合判断, 从而达到块划分模式快速终止的目的。实验结果表明, 本发明结合两个方面提出的块划分快速算法在保证 HEVC 编码器的视频质量和输出码率基本不变的前提下, 大大加快了编码速度, 提高编码效率。

附图说明

[0019] 图 1 是本发明提出的块划分快速算法的流程图。

[0020] 图 2 是由上至下的基于双线性插值的平面预测图。

[0021] 图 3 是由下至上的块划分快速算法示意图。

[0022] 图 4 是 (3, 4) 坐标方向上的梯度值计算过程示意图。

[0023] 图 5 是块结合搜索表示意图。

具体实施方式

[0024] 为使本发明的目的、技术方案及优点更加清楚、明确, 以下参照附图并举实施例对本发明进一步详细说明。

[0025] 本发明提出一种双向夹逼的块划分方法, 如图 1 所示, 双向夹逼块划分方法流程包括以下步骤:

[0026] 步骤 S1: 读取整个 LCU 中的像素值, 从最小的 CU 开始, 从下至上, 根据梯度方向值

判断小块能否合并为大块,完成块合并搜索表。

[0027] 步骤 S2:进入每个深度的 CU,根据深度和位置信息,获取搜索表对应位置的结合块的深度和位置信息。

[0028] 步骤 S3:若当前 CU 深度与结合块深度一致,进行由下至上的块划分方法的判断流程;否则,进行由上至下的块划分快速算法的判断流程。

[0029] 其中,若当前 CU 深度与结合块深度一致,执行如下:

[0030] 步骤 S31:判断当前 CU 的梯度方向值是否在 (1,9) 内,是则进行 $N \times 2N$, $nU \times 2N$, $nD \times 2N$ 的块的预测编码,跳到步骤 S4;

[0031] 步骤 S32:判断当前 CU 的梯度方向值是否在 (9,16) 内,是则进行 $2N \times N$, $2N \times nU$, $2N \times nD$ 的块的预测编码,跳到步骤 S4;

[0032] 其中,若当前 CU 深度与结合块深度不一致,执行如下:

[0033] 计算图像块内原像素小块均值与平面预测值之间绝对差的平均值,若该值小于设定阈值,则进行 classA 的预测,否则使用 classB 的预测。其中,两个类别如下:

[0034] $A = \{PART_2N \times 2N, PART_N \times N\}$,

[0035] $B = \{PART_2N \times N, PART_N \times 2N, PART_2N \times nU,$

[0036] $PART_2N \times nD, PART_nL \times 2N, PART_nR \times 2N\}$

[0037] 步骤 S4:判断当前 CU 深度与结合块深度是否一致,是则计算损失代价,否则继续分裂。

[0038] 其中,所述由上至下的块划分方法是一种基于双线性插值的平面预测快速算法。其具体方法为:

[0039] 在帧内的块划分过程中,大部分的平坦图像内没有明显边界的 LCU 块会划分为 64×64 或 32×32 尺寸的 CU 块。而对于图像内容复杂,细节较多的 LCU 块则会进一步划分出 16×16 或 8×8 。但不管 LCU 块内的图像内容复杂度如何,HEVC 编码器在预测编码过程中,都要经历从 64×64 至 8×8 各个深度 CU 的预测编码过程。而在帧间的块划分过程中,图像内容平坦的 CU 块偏向选择 $2N \times 2N$, $N \times N$ 的 PU 块模式。

[0040] 因此,如果能用低复杂度的快速算法区分图像内容复杂的 LCU 块和图像内容平坦的 LCU 块,并对图像内容复杂的 LCU 块跳过 64×64 、 32×32 这种大尺寸 CU 块的编码过程,则能在基本维持原有的视频压缩效率和质量的同时有效地提高编码速度。

[0041] 如图 2 所示,对于 64×64 、 32×32 尺寸的 CU 块,为减少视频图像中的一些噪声颗粒像素的影响把 CU 块划分为一连串 2×2 尺寸的小块,计算每个小块的均值,以 CU 块上下左右四组小块作参考用双线性插值的方法预测生成其余小块值。而对于 16×16 、 8×8 尺寸的 CU,以 CU 块上下左右四组原像素作参考用双线性插值的方法生成其余像素。

[0042] 由左右两参考数组生成的预测值如式(3):

$$[0043] \text{Left_pred}(x, y) = [(\text{length}/2) - 0.5 \times (y+1)] \times \text{piOrg}(0, y) \quad (1)$$

$$[0044] \text{Right_pred}(x, y) = 0.5 \times (y+1) \times \text{piOrg}(\text{length}-1, y) \quad (2)$$

$$[0045] \text{Hor_pred}(x, y) = \text{Left_pred}(x, y) + \text{Right_pred}(x, y) \quad (3)$$

[0046] 由上下两参考数组生成的预测值如式(6):

$$[0047] \text{Up_pred}(x, y) = [(\text{length}/2) - 0.5 \times (x+1)] \times \text{piOrg}(x, 0) \quad (4)$$

$$[0048] \text{Down_pred}(x, y) = 0.5 \times (x+1) \times \text{piOrg}(x, \text{length}-1) \quad (5)$$

[0049] $Ver_pred(x, y) = Up_pred(x, y) + Down_pred(x, y)$ (6)

[0050] 本发明采用的双线性插值如式(7)：

[0051] $predSample(x, y) = [Ver_pred(x, y) + Hor_pred(x, y)] / [length \times 2]$ (7)

[0052] $piOrg(x, 0)$ 、 $piOrg(x, length-1)$ 分别为上下两参考数组在 x 点位置上的参考值。 $piOrg(0, y)$ 、 $piOrg(length-1, y)$ 分别为左右两参考数组在 y 点位置上的参考值。 $length$ 是 CU 块的宽度, 由于在该算法中以 2×2 块为单位, 故其宽度为 $length/2$ 。由于在帧内预测中, 预测值更依赖于上边和左边两组参考值, 故式中 0.5 是对下边和右边两组参考值的加权值。而 $(x+1)$ 则是预测点与参考点的距离加权值。 $predSample(x, y)$ 是坐标点 (x, y) 由上下左右四组参考小块双线性插值的预测值。

[0053] 本发明以双线性插值获取的预测与原始值之间的绝对差作为反映图像块内复杂度的特征值。计算方式如式(9)所示：

[0054] $Diff(x, y) = abs[piOrg(x, y) - predSample(x, y)]$ (8)

[0055]

$$Up_Diff = \frac{1}{[(length/2) - 2]^2} \sum_{x=1}^{x=\{length/2\}-1} \sum_{y=1}^{y=\{length/2\}-1} Diff(x, y) \quad (9)$$

[0056] $Diff(x, y)$ 是座标为 (x, y) 的 2×2 小块均值的预测值与原始值的绝对差, Up_Diff 是图像块内原像素小块均值与平面预测值之间绝对差的平均值。 Up_Diff 用于判断 LCU 块图像内容平坦度, Up_Diff 的值越大, 图像内容复杂度越高; Up_Diff 的值越小, 图像内容越简单。

[0057] 使用公式(10)来判决不同深度的 CU 复杂度。

[0058] $Up_Diff < Up_Th[depth]$ (10)

[0059] 通过实验数据统计, 根据其分析结果, 本文总结出阈值 Up_Th 在不同编码块深度和不同量化参数 QP 下的变化规律。如下面式子所示。

[0060] $Up_Th[depth+1] = Up_Th[depth] - 5$ (11)

[0061] $Up_Th[0, QP] = 3 + QP$ (12)

[0062] 其中: $QP = 22, 27, 32, 37$ 。

[0063] 步骤 S3 中所述的由下至上的块划分快速算法是一种基于图像纹理的块划分快速算法。具体细节如图 3 所示: 本算法在 LCU 进入预测编码前, 从下至上, 以计算梯度方向值来获取图像纹理信息, 通过图像纹理信息先判断 4×4 块是否能合成 8×8 块, 再在已判断能合成 8×8 块中通过图像纹理信息判断能否合成 16×16 块, 继而在已判断能合成 16×16 块中通过图像纹理信息判断能否合成 32×32 块, 以此类推。LCU 在预测编码过程中, 若 CU 分裂的深度已达到快速算法判断能合成块的深度, 则停止继续分裂。该算法能快速停止 CU 分裂以提高编码速度。

[0064] 本算法使用计算梯度方向值的方法来获取图像纹理信息, 梯度方向的坐标定义如表 1 所示。

[0065]

梯度方向	坐标	梯度方向	坐标	梯度方向	坐标	梯度方向	坐标
------	----	------	----	------	----	------	----

1	{1, -1}	5	{1, 0}	9	{1, 1}	13	{0, 1}
2	{4, -3}	6	{4, 1}	10	{4, -3}	14	{1, -4}
3	{2, -1}	7	{2, 1}	11	{1, 2}	15	{1, -2}
4	{4, -1}	8	{4, 3}	12	{1, 4}	16	{3, -4}

[0066] 表 1 梯度方向对应的坐标方向

[0067] 如图 4 所示,以 16×16 块为例,该算法先设 (0, 0) 位置的像素为主像素,然后以主像素位置为基准以坐标方向 (3, 4) 为跳跃距离寻找同一直线方向上的其它像素作为次像素。计算并记录主像素与每个次像素的绝对差的平均值。当寻找的位置越过二维数组的边界,以 (0, 1) 位置的像素为主像素,再重复上述步骤。如此类推,遍历块内的所有像素作为主像素,寻找对应的次像素,求出该坐标方向梯度的绝对差平均值 Dire_Grad。

[0068] 计算公式如式 (14) :

$$[0069] \quad Diff(d, x, y) = \sum_{n=1}^{x+nX_d < S, y+nY_d < S} abs(P[x, y] - P[x+nX_d, y+nY_d]) \quad (13)$$

$$[0070] \quad Dire_Grad(d) = \frac{1}{Count_Num} \sum_{x=0}^{x < S} \sum_{y=0}^{y < S} Diff(d, x, y) \quad (14)$$

[0071] Diff(d, x, y) 是梯度方向号为 d, 主像素为 (x, y) 时的方向梯度值。式中 P[x, y] 为主像素, P[x+nXd, y+nYd] 为次像素 ;d 为梯度方向代号 ;S 是图像块的尺寸 ;Xd, Yd 为 d 梯度方向在表 4-4 上对应的坐标方向在 X 轴和 Y 轴上的偏移值 ;Count_Num 为绝对差运算的计算次数。

[0072] 本文通过实验数据统计,设立阈值 Down_Th 对梯度值 Dire_Grad 进行判断来确定图像块内是否有明显的纹理方向,如式 (15~17) 所示。

$$[0073] \quad Dire_Grad(\min) < Down_Th(\text{depth}) \quad (15)$$

$$[0074] \quad Down_Th[\text{depth}+1] = 0.5 \times Down_Th[\text{depth}+1] \quad (16)$$

$$[0075] \quad Down_Th[0, QP] = 2.5 \times [(QP-17)/5] + 10 \quad (17)$$

[0076] 其中 QP=22, 27, 32, 37。

[0077] 通过上述的基于纹理方向的梯度值计算算法与阈值 Down_Value 判断是否合并,更新结构体内各元素的值。从而形成一个块结合搜索表如图 5 所示,棕色的小块表示经过结合判断后包含块合并信息的结构体。该图表反映的是在预测编码过程中 CU 可分裂的最小尺寸。

[0078] 若当前预测块的深度与导出的块合并深度相等 (Curr_Depth=Down_Depth) 且梯度方向值 Dire_Num 在 (1, 9) 内,可跳过 2N×N 和 2N×nU、2N×nD 尺寸的水平划分 PU 块的预测编码过程。若当前预测块的深度与导出的块合并深度相等 (Curr_Depth==Down_Depth) 且梯度方向值 Dire_Num 在 (9, 16) 内, CU 块的图像纹理偏向水平方向,因此可跳过 N×2N 和 nR×2N、nD×2N 尺寸的垂直划分 PU 块的预测编码过程。完成 PU 的预测编码过程后,若当前预测块的深度与导出的块合并深度相等 (Curr_Depth==Down_Depth) 当前块不再继续

分裂。否则,进入分裂判断流程。

[0079] 本发明提出的算法在 HM 上实现,测试时采用随机接入配置参数 (Random-access configuration) :

[0080] (1) 编码结构 :IbB₂bB₄bB₆bB₈; (2) I 帧周期 :32

[0081] (3) 图像组尺寸 :8 ; (4) 编码帧数 :96 帧 ;

[0082] 为了评估算法的性能,算法性能分析用到以下计算公式及性能参数 :

$$[0083] \quad \Delta Bit(\%) = \frac{BitRate_{prop} - BitRate_{HM}}{BitRate_{HM}} \times 100\% \quad (18)$$

$$[0084] \quad \Delta Time(\%) = \frac{EncTime_{prop} - EncTime_{HM}}{EncTime_{HM}} \times 100\% \quad (19)$$

$$[0085] \quad \Delta Ypsnr(\text{dB}) = Ypsnr_{prop} - Ypsnr_{HM} \quad (20)$$

[0086] ΔBit 、 $\Delta Time$ 、 $\Delta Ypsnr$ 分别为码率增长率、编码时间下降百分率以及峰值信噪比下降值。

[0087] 实验结果如表 2、3 所示。

序列	QP=22			QP=27		
	ΔBit (%)	$\Delta Ypsnr$ (dB)	$\Delta Time$ (%)	ΔBit (%)	$\Delta Ypsnr$ (dB)	$\Delta Time$ (%)
PeopleOnStreet	1.6096	-0.0371	-26.064	1.7412	-0.0652	-30.827
Traffic	0.5346	-0.0321	-37.515	0.7480	-0.0489	-43.256
ParkScene	0.8303	-0.0386	-45.688	0.7846	-0.0627	-50.792
Kimono1	0.6525	-0.0151	-52.612	1.1475	-0.0239	-55.138
Cactus	-0.2918	-0.0242	-40.958	0.8415	-0.0283	-46.882
BQTerrace	-0.1522	-0.0191	-32.870	-0.0452	-0.0200	-38.813
[0088] BasketballDrive	0.3470	-0.0228	-49.663	0.8814	-0.0356	-53.929
BasketballDrill	1.7903	-0.0529	-39.884	1.6483	-0.0700	-45.177
BQMall	1.2878	-0.0446	-31.129	1.7033	-0.0757	-38.195
PartyScene	0.6475	-0.0376	-30.138	0.8748	-0.0591	-32.747
BasketballPass	1.4646	-0.0749	-38.734	1.7778	-0.0675	-46.450
BlowingBubbles	0.9151	-0.0436	-27.329	1.0361	-0.0644	-39.368
BQSquare	0.2125	-0.0344	-21.145	0.0554	-0.0479	-19.363
RaceHorses	1.3496	-0.0776	-35.967	2.0012	-0.0883	-38.905
vidyo1	0.0669	-0.0341	-47.289	0.7973	-0.0396	-51.604
vidyo3	0.1432	-0.0271	-45.855	0.9214	-0.0519	-51.169
[0089] vidyo4	0.3787	-0.0289	-47.722	1.1886	-0.0481	-52.298
均值	0.6933	-0.03792	-38.2685	1.0649	-0.05277	-43.2302

[0090] 表 2QP=22, 27 的块划分快速算法实验结果

序列	QP=32			QP=37		
	Δ Bit (%)	Δ Ypsnr (dB)	Δ Time (%)	Δ Bit (%)	Δ Ypsnr (dB)	Δ Time (%)
PeopleOnStreet	1.4709	-0.0826	-36.205	0.9931	-0.1082	-42.088
Traffic	0.6377	-0.0656	-48.698	0.4725	-0.0715	-52.376
ParkScene	0.5330	-0.0705	-53.486	-0.0724	-0.0634	-55.422
KimonoI	0.8636	-0.0414	-57.126	0.6733	-0.0502	-58.203
Cactus	0.9083	-0.0539	-51.329	0.7355	-0.0663	-53.343
BQTerrace	-0.0367	-0.0269	-45.018	-0.3047	-0.0306	-48.600
BasketballDrive	0.9555	-0.051	-56.763	0.2603	-0.0642	-58.687
[0091] BasketballDrill	0.9874	-0.079	-50.719	0.8841	-0.0768	-54.481
BQMall	1.6394	-0.1114	-44.632	1.1026	-0.1288	-48.801
PartyScene	0.7962	-0.0676	-40.315	-0.0395	-0.0858	-45.871
BasketballPass	1.5588	-0.0703	-51.619	0.2547	-0.1300	-54.362
BlowingBubbles	1.1135	-0.0762	-47.364	0.6124	-0.0873	-50.032
BQSquare	0.1899	-0.0097	-39.424	0.3276	-0.0435	-27.886
RaceHorses	1.7622	-0.1183	-42.657	1.8383	-0.1114	-49.195
vidyo1	0.8211	-0.0657	-54.769	0.5047	-0.0730	-57.036
vidyo3	1.0437	-0.1064	-56.909	1.1296	-0.1136	-59.956
vidyo4	1.3156	-0.0535	-55.435	0.7994	-0.0650	-57.648
均值	0.9741	-0.06764	-48.9687	0.5983	-0.080565	-51.4111

[0092] 表 3QP=32, 37 的块划分快速算法实验结果

[0093] 由表 2, 3 的实验数据可以看出, 本发明提出的算法在 QP=22, 27, 32, 37 时, 相对于现有算法, 分别减少了 38%, 43%, 48%, 51% 的编码时间, 而只是增加了小于 1% 的码率。本发明提出的快速算法对帧内编码的压缩效率和视频质量基本没有影响。

[0094] 本发明通过对 HEVC 编码器中多层次的 CU 划分和多种类型 PU 划分模式进行深入研究, 提出了 HEVC 块划分的快速算法。一方面, 根据当前 CU 内图像内容的复杂度, 本算法对 CU 判断是否跳过当前 CU 或 PU 的预测编码过程。另一方面, 本算法基于图像纹理方向特征, 从小尺寸 CU 到大尺寸 CU 进行块结合判断, 从而达到块划分模式快速终止的目的。实验结果表明, 本发明结合两个方面提出的块划分快速算法在保证 HEVC 编码器的视频质量和输出码率基本不变的前提下, 大大加快了编码速度, 提高编码效率。

[0095] 应当理解的是, 本发明的应用不限于上述的举例, 对本领域普通技术人员来说, 可以根据上述说明加以改进或变换, 所有这些改进和变换都应属于本发明所附权利要求的保护范围。

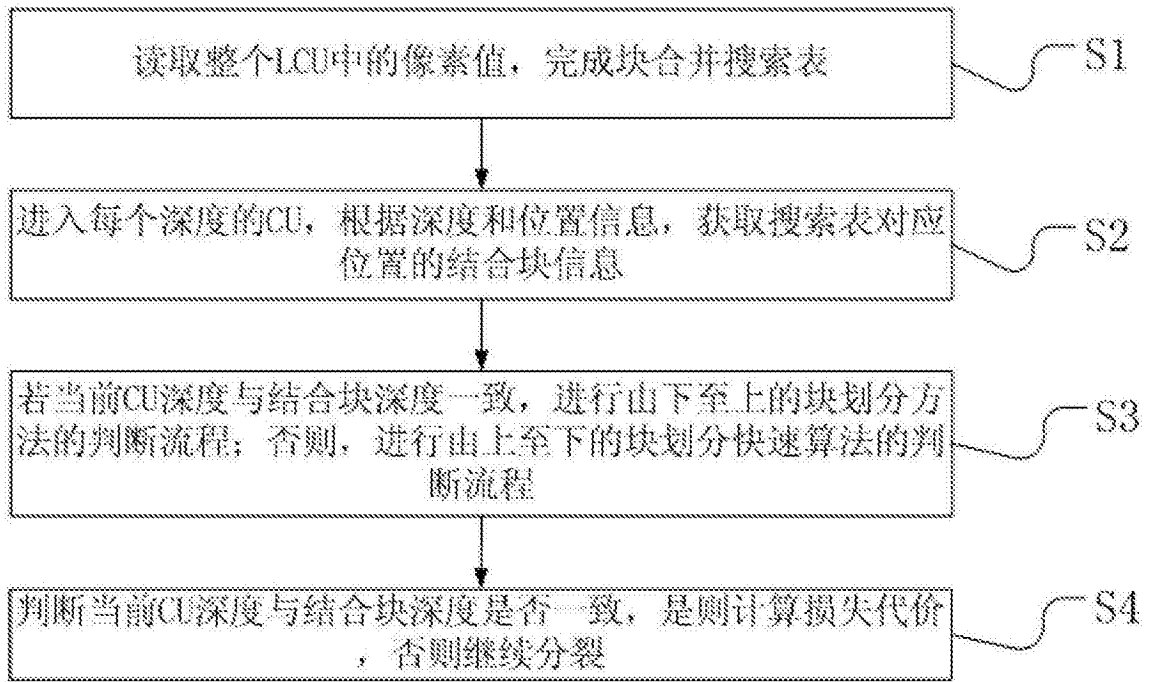


图 1

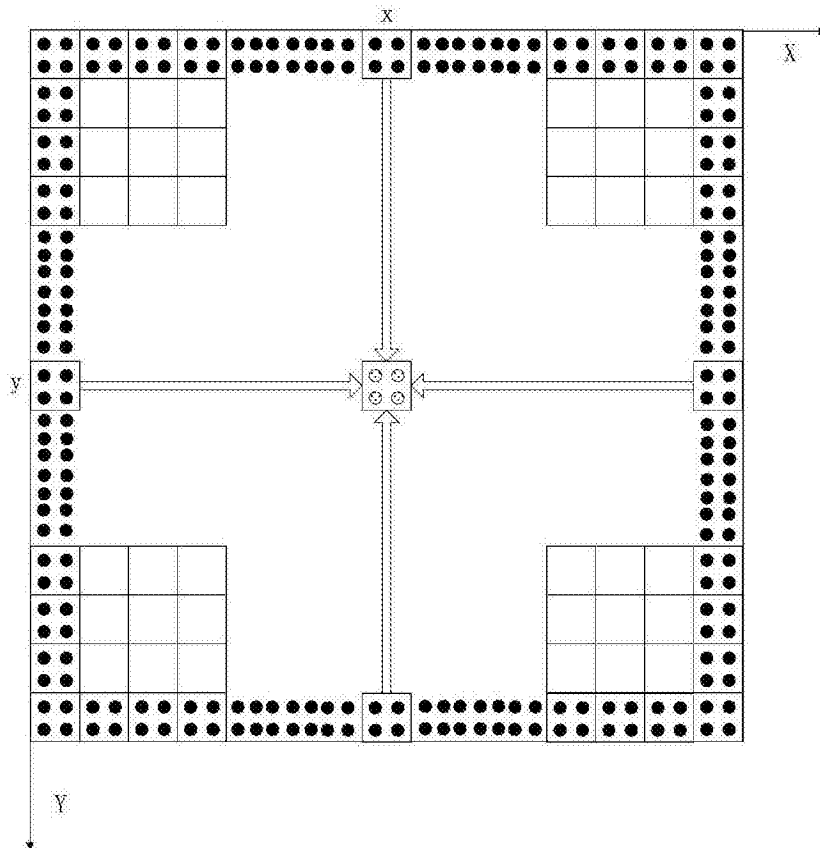


图 2

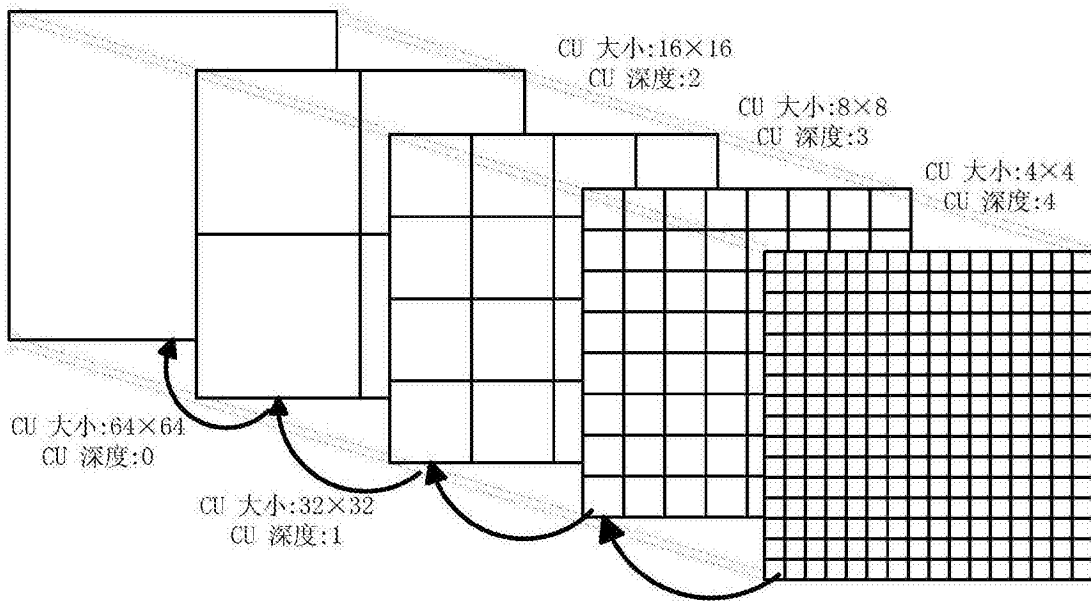


图 3

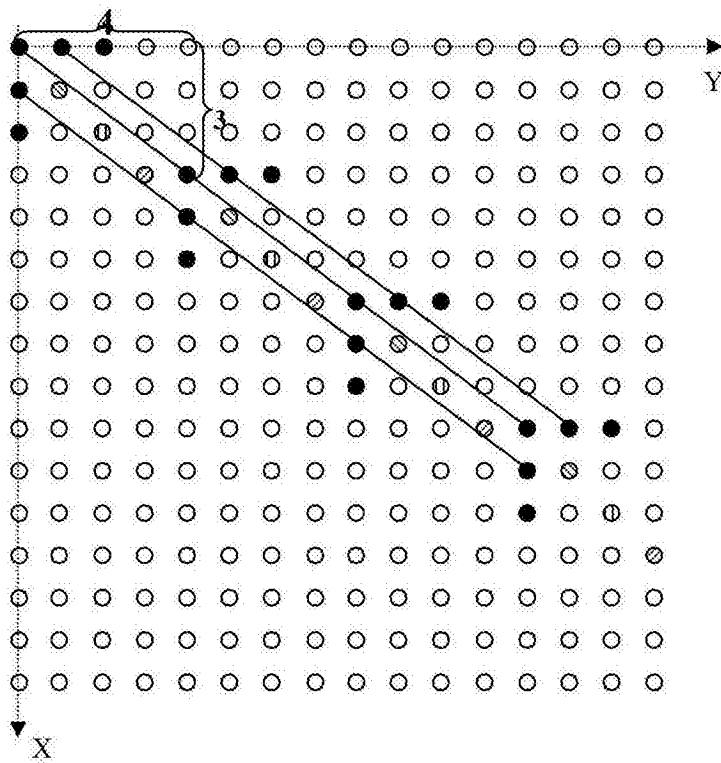


图 4

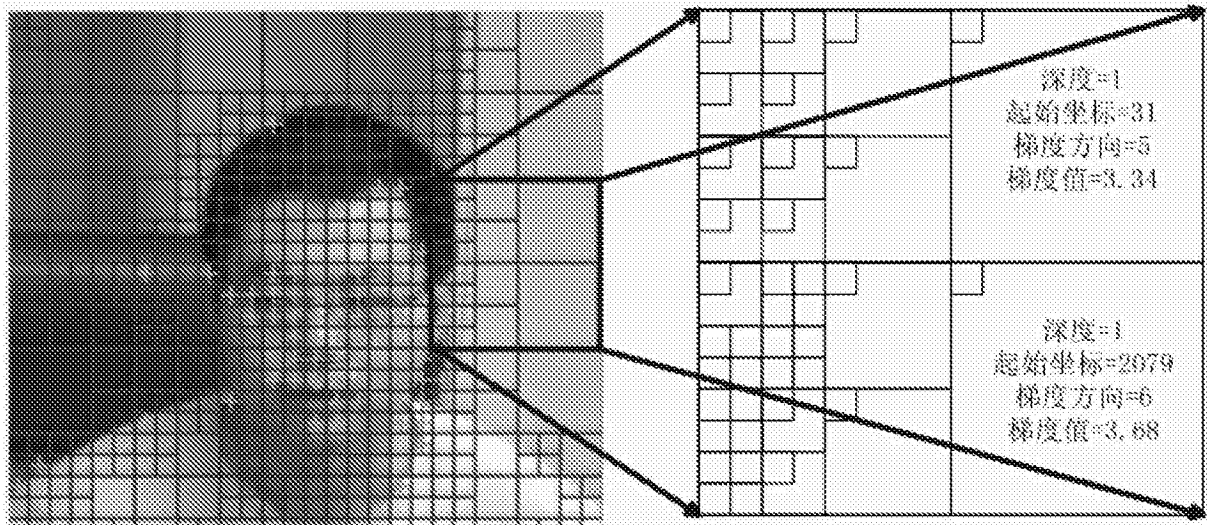


图 5