(19)中华人民共和国国家知识产权局

(12)发明专利

(10)授权公告号 CN 101917618 B
(45)授权公告日 2012.01.25

(21)申请号 201010262071.0

(22)申请日 2010.08.20

(73)专利权人 浙江大学
地址 310027 浙江省杭州市西湖区浙大路38号

(72)发明人 陈耀武 朱威 徐巍炜

(74)专利代理机构 杭州天勤知识产权代理有限公司 33224
代理人 胡红娟

(51)Int.Cl.
H04N 7/26(2006.01)
H04N 7/32(2006.01)
H04N 13/00(2006.01)

审查员 侯冠华

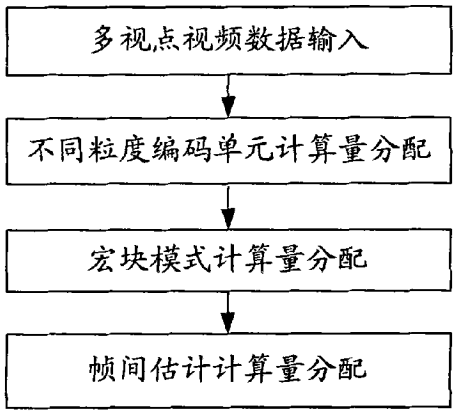
权利要求书 3 页 说明书 16 页 附图 4 页

(54)发明名称

多视点视频编码分层B帧预测结构的计算复杂度控制方法

(57)摘要

本发明公开了一种多视点视频编码分层B帧预测结构的计算复杂度控制方法，包括以下步骤：(1)输入多视点视频数据；(2)对不同粒度编码单元进行计算量分配；(3)对宏块帧间模式估计进行计算量分配；(4)对帧间估计进行计算量分配。本发明方法适用于多视点视频编码分层B帧预测结构的计算复杂度控制，可以准确控制多视点编码的计算整体计算量，减少计算量的波动，同时保持编码率失真性能。



CN 101917618 B

1. 一种多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，包括以下步骤：

(1) 输入多视点视频数据；

(2) 对不同粒度编码单元进行计算量分配；

(3) 对宏块帧间模式估计进行计算量分配；

(4) 对帧间估计进行计算量分配；

所述的不同粒度编码单元包括：GGOP、超帧、帧和宏块四个不同粒度的编码单元，所述的步骤 (2) 中对不同粒度编码单元进行计算量分配的过程如下：

(2.1) 对 GGOP 进行计算量分配：

在每个 GGOP 开始编码之前，计算 GGOP 的目标计算量 TC_{GGOP}，如式 (I) 所示：

$$TC_{GGOP}(r) = N_{SF}(r) \times TargetAvgC_{SF} + \min(VBC_{GGOP}(r), \alpha \times TargetAvgC_{SF}) \quad (I)$$

式 (I) 中，r 是当前 GGOP 编码索引，N_{SF} 是当前 GGOP 中的超帧个数；$TargetAvgC_{SF}$ 是超帧的目标平均计算量；VBC_{GGOP} 为前一 GGOP 实际计算量与其目标计算量之间的偏差；α 为 VBC_{GGOP} 的上限控制参数，$\alpha \times TargetAvgC_{SF}$ 为 VBC_{GGOP} 的上限值；

(2.2) 对超帧进行计算量分配：

根据超帧中各帧的帧间预测参考帧个数和超帧所处的时域层的计算复杂度权重因子计算每个超帧的计算复杂度权重值 W_{SF}，如式 (II′) 所示：

$$W_{SF}(m) = W_{Layer} \times RefN_{SF}(m) \quad (II')$$

式 (II′) 中 m 是超帧编码索引；W_{Layer} 代表超帧 m 所处时域层的计算复杂度权重因子；$RefN_{SF}$ 代表超帧 m 中各帧的帧间预测用到的参考帧个数；

再根据超帧计算复杂度权重值 W_{SF} 和当前 GGOP 剩余计算量，分配当前编码超帧目标计算量 $TC_{SF}(k)$，如式 (II) 所示：

$$TC_{SF}(k) = (TC_{GGOP} - C_{GGOP}) \times \frac{W_{SF}(k)}{\sum_{m \in \Phi_1} W_{SF}(m)} \quad (II)$$

其中，k 是当前超帧在 GGOP 中的编码索引，C_{GGOP} 是当前 GGOP 已消耗计算量，当前 GGOP 剩余计算量为当前目标计算量 TC_{GGOP} 与当前 GGOP 已消耗计算量 C_{GGOP} 的差值；Φ_1 是当前 GGOP 中待编码超帧的索引集合；

(2.3) 对帧进行计算量分配：

根据当前超帧剩余计算量和待编码帧的帧间预测参考帧个数分配当前编码帧目标计算量 $TC_F(i)$，如式 (III) 所示：

$$TC_F(i) = (TC_{SF} - C_{SF}) \times \frac{RefN_F(i)}{\sum_{j \in \Phi_2} RefN_F(j)} \quad (III)$$

其中，i 是当前帧在当前超帧中的编码索引，C_{SF} 为当前超帧已消耗计算量，当前超帧剩余计算量为当前目标计算量 TC_{SF} 与当前超帧已消耗计算量 C_{SF} 的差值；$RefN_F(i)$ 是当前帧的参考帧个数，Φ_2 是当前超帧中待编码帧的索引集合，j 是当前超帧中待编码帧的索引；

(2.4) 对宏块进行计算量分配：

(2.4.1) 根据当前宏块与参考帧在零矢量处的差值绝对值和以及当前宏块的纹理强度

来计算当前宏块的预测计算复杂度 $MbComplexity_{PRED}(n)$，如式 (IV) 所示：

$$MbComplexity_{PRED}(n) = SAD_{MV00}(n) \times (1 + \frac{DEV_{16\times16}(n)}{\delta}) \qquad (IV)$$

式 (IV) 中，n 是宏块索引，SAD_{MV00} 是当前宏块与前向第一个参考帧在零矢量处的像素差值绝对值和，DEV_{16x16} 是当前宏块的纹理强度，δ 是纹理强度的归一化系数；

(2.4.2) 在得到每个宏块的预测计算复杂度之后，利用当前帧所有宏块的预测计算复杂度的平均值 $AvgMbComplexity_{PRED}$ 对每个宏块的预测计算复杂度进行归一化，得到每个宏块的计算复杂度权重值 Mb Weight，如式 (V) 所示：

$$MbWeight(n) = \frac{MbComplexity_{PRED}(n)}{AvgMbComplexity_{PRED}} \qquad (V)$$

(2.4.3) 对每个宏块的计算复杂度权重值 Mb Weight 的上限进行限制，如式 (VI) 所示：

$$Mb\ Weight_{Clip}(n) = \min(Mb\ Weight(n), Mb\ Weight_{MAX}) \qquad (VI)$$

式 (VI) 中，$Mb\ Weight_{MAX}$ 为宏块计算复杂度权重上限阈值；

(2.4.4) 根据当前帧剩余计算量和剩余宏块的 $Mb\ Weight_{Clip}$ 分配当前宏块的目标计算量 $TC_{Mb}(q)$，如式 (VII) 所示：

$$TC_{Mb}(q) = (TC_{Frame} - C_{Frame}) \times \frac{MbWeight_{Clip}(q)}{\sum_{n=q}^{N-1} MbWeight_{Clip}(n)} \qquad (VII)$$

式 (VII) 中，q 是当前宏块的编码索引，从 0 开始计数；N 是当前帧宏块的个数；C_{Frame} 为当前帧已消耗的计算量，当前帧剩余计算量为当前目标计算量 TC_{Frame} 与当前帧已消耗计算量 C_{Frame} 的差值。

2. 如权利要求 1 所述的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，所述的步骤 (3) 中对宏块帧间模式估计进行计算量分配的过程如下：先对宏块帧间模式的估计顺序进行排列，然后按该顺序对各帧间模式逐一进行估计。

3. 如权利要求 2 所述的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，所述的对宏块帧间模式的估计顺序的排列，是根据各帧间模式被选为最佳宏块模式的比例统计、宏块的计算复杂度权重值和划分块纹理强度三要素相结合来进行的。

4. 如权利要求 2 或 3 所述的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，所述的对宏块帧间模式的估计顺序的排列为：

首先进行第一类 Skip 模式的估计；接着进行第二类 Inter16×16、Inter16×8 和 Inter8×16 模式的排列和估计；然后进行第三类模式中的 Inter8×8、Inter8×4 和 Inter4×8 模式的排列和估计；再进行第三类模式中的 Inter4×4 模式的估计；最后进行第三类模式中的 Inter8×8Frext 模式的估计。

5. 如权利要求 1 所述的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，所述的步骤 (4) 中对帧间估计进行计算量分配的过程为：先对每个划分块的各参考帧的帧间估计顺序进行排列，然后按顺序对各参考帧逐一进行估计；所述的参考帧的帧间估计顺序如下：

在每个划分块的帧间估计过程中，先搜索前向队列的参考帧，再搜索后向队列的参考帧，最后根据前向搜索的结果和后向搜索的结果进行双向预测的搜索，其中，在对前向或后向参考队列中的参考帧进行帧间估计顺序排列的时候，始终将时域方向的参考帧排在视点间参考帧的前面进行估计。

6. 如权利要求 1 所述的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，其特征在于，所述的步骤 (2) 中对不同粒度编码单元进行计算量分配的过程还包括：

(2.5) 对各时域层计算复杂度权重因子的更新，其过程如下：

在当前 GGOP 编码结束后，利用各时域层的平均 SAD_{MV00} 值，自适应更新各时域层的计算复杂度权重因子，如式 (XV) 所示：

$$W_{\text{Layer}}(r+1,l)=\eta\times W_{\text{Layer}}(r,l)+(1-\eta)\times\frac{AvgSAD_{\text{MV00}}(r,l)}{AvgSAD_{\text{MV00}}(r,L_{\text{MAX}})} \quad (\text{XV})$$

式 (XV) 中，r 是当前 GGOP 的索引，l 是分层 B 帧预测结构时域层数索引，L_{MAX} 最大时域层数索引，η 是时域权重因子。

多视点视频编码分层 B 帧预测结构的计算复杂度控制方法

技术领域

[0001] 本发明涉及数字视频信号编码领域，具体涉及一种多视点视频编码分层 B 帧预测结构的计算复杂度控制方法。

背景技术

[0002] 随着视频采集和显示技术的飞速发展，已经有许多针对 3D 视频场景应用的设备出现，比如 3D 电视和自由视点电视。3D 视频由于可以为用户提供真实的视觉感受，正在逐渐取代传统的 2D 视频。多视点视频 (Multiview Video，MV) 是利用不同位置的摄像头对同一场景进行拍摄得到的视频数据，它包含了多个视角的视觉信息，是一种重要的 3D 视频数据。由于多视点视频的数据量随着视点个数的增多而成倍增加，因此为了解决其传输和存储的问题，多视点视频编码 (Multiview Video Coding，MVC) 对多视点视频数据进行高效的压缩。联合视频工作组 (Joint Video Team，JVT) 从 2006 年开始对多视点视频编码进行标准化工作，将其作为 H.264/AVC 的第四个增修案。为了提高编码压缩效率，多视点视频编码既采用了传统视频编码的时域预测来减少时间方向上的数据冗余，同时采用了视点间预测来减少视点方向上的数据冗余。JVT 发布的多视点视频编码校验模型中采用了 HHI 提出的多视点视频编码分层 B 帧预测结构 (Vetro A，Pandit P，Kimata H，Smolic A，Wang Y K.Joint multiview video model (JMVM)8.0.Doc.AA207，2008，Geneva，JVT 27th meeting)，同时采用时域预测和视点间预测，有效提高编码效率。

[0003] 视频编码器的实际应用中，编码器可获得计算资源通常是有限的，并且会随着整个系统资源的变化而有所调整，因此编码器需要具备计算复杂度可伸缩的能力，能够根据实际情况准确控制整体计算量。另外，在视频编码的实际应用中，整个应用系统除了视频编码器之外通常还会有其它相关任务在运行，如果编码器的计算量的波动过大，就可能会影响其它任务的正常运行。因此编码器还需要对计算量波动进行控制。综上所述，计算复杂度控制算法对视频编码的实际应用具有重要的意义。

[0004] 现有单视点视频编码的计算复杂度控制算法可以用于多视点视频编码中每个视点的单独控制。多视点视频编码要求不同视点帧的编码顺序排列是按时刻优先的原则进行，即同个时刻不同视点的帧要编码完之后才能开始其它时刻的帧进行编码，因此多视点的计算复杂度控制方法需要对各个视点计算量的联合控制。为了提高编码压缩效率，多视点的编码会选用比单视点编码更为复杂的编码预测结构，例如 HHI 提出的多视点视频编码分层 B 帧预测结构，因此其计算复杂度控制算法还需要对多视点复杂编码预测结构的支持。

[0005] 多视点视频编码的宏块模式估计采用同 H.264/AVC 一样的模式率失真优化技术，将具有最小模式率失真代价的宏块模式作为最佳宏块模式 (Sullivan G J，Wiegand T.Rate-distortion optimization for video compression [J].IEEE Signal Processing Magazine，1998，15(6)：74-90.)。多视点视频编码的帧间估计采用同 H.264/AVC 一样的帧间估计率失真优化技术，将具有最小帧间估计率失真代价的帧间匹配块作为划分块帧间估

计的最佳帧间匹配块(Wiegand T, Schwarz H, Joch A, et al..Rate-constrained coder control and comparison of video coding standards.IEEE Transactions on Circuits and Systems for Video Technology,2003,13(7):688-703.)。

发明内容

[0006] 本发明提供了一种多视点视频编码分层B帧预测结构的计算复杂度控制方法,可以准确控制编码的整体计算量,减少计算量的波动,并保持良好的编码率失真性能。

[0007] 一种多视点视频编码分层B帧预测结构的计算复杂度控制方法,包括以下步骤:

[0008] (1)输入多视点视频数据;

[0009] (2)对不同粒度编码单元进行计算量分配;

[0010] (3)对宏块帧间模式估计进行计算量分配;

[0011] (4)对帧间估计进行计算量分配;

[0012] 所述的不同粒度编码单元包括:GGOP、超帧(Super Frame,SF)、帧和宏块四个不同粒度的编码单元。其中,GGOP(Group of GOP)为GOP组,是指不同视点在同一时间段的所有图像组的组合,所述的图像组(GOP,Group of Picture)是指某一视点在某一时间段的图像的组合;所述的超帧是指同一个时刻不同视点的所有帧的组合。

[0013] 其中,所述的步骤(2)为:

[0014] (2.1)对GGOP进行计算量分配:

[0015] 在每个GGOP开始编码之前,计算GGOP的目标计算量TC_{GGOP},如式(I)所示:

[0016] $TC_{GGOP}(r) = N_{SF}(r) \times TargetAvgC_{SF} + \min(VBC_{GGOP}(r), \alpha \times TargetAvgC_{SF})$ (I)

[0017] 式(I)中,r是当前GGOP编码索引,N_{SF}是当前GGOP中的超帧个数;$TargetAvgC_{SF}$是超帧的目标平均计算量;VBC_{GGOP}是GGOP计算量虚拟缓冲区,为前一GGOP实际计算量与其目标计算量之间的偏差。VBC_{GGOP}的初始值为0,其在每个GGOP编码结束后根据TC_{GGOP}和GGOP实际消耗计算量来更新;α为VBC_{GGOP}的上限控制参数,通常根据经验来选取,设为0.1~4.0之间,本发明优选设为1.0;$\alpha \times TargetAvgC_{SF}$为$VBC_{GGOP}$的上限值。

[0018] (2.2)对超帧进行计算量分配:

[0019] 根据超帧中各帧的帧间预测参考帧个数和所处的时域层计算复杂度权重因子计算每个超帧的计算复杂度权重值W_{SF},如式(II′)所示:

[0020] $W_{SF}(m) = W_{Layer} \times RefN_{SF}(m)$ (II′)

[0021] 式(II′)中m是超帧编码索引;W_{Layer}代表超帧m所处时域层的计算复杂度权重因子,初始值根据经验来设定,所处时域层的层数越低,W_{Layer}初始值越大,W_{Layer}的更新如式(XV)所示;$RefN_{SF}$代表超帧m中各帧的帧间预测用到的参考帧个数;

[0022] 再根据超帧计算复杂度权重值W_{SF}和当前GGOP剩余计算量,分配当前编码超帧目标计算量$TC_{SF}(k)$,如式(II)所示:

[0023] $$TC_{SF}(k) = (TC_{GGOP} - C_{GGOP}) \times \frac{W_{SF}(k)}{\sum_{m \in \Phi_1} W_{SF}(m)} \quad (II)$$

[0024] 式(II)中,k是当前超帧在GGOP中的编码索引,C_{GGOP}是当前GGOP已消耗计算量,当前GGOP剩余计算量为当前目标计算量TC_{GGOP}与当前GGOP已消耗计算量C_{GGOP}的差值;Φ_1

是当前 GGOP 中待编码超帧的索引集合；

[0025] (2.3) 对帧进行计算量分配：

[0026] 根据当前超帧剩余计算量和待编码帧的帧间预测参考帧个数分配当前编码帧目标计算量 $TC_F(i)$，如式 (III) 所示：

[0027] $$TC_F(i) = (TC_{SF} - C_{SF}) \times \frac{RefN_F(i)}{\sum_{j \in \Phi_2} RefN_F(j)} \qquad (III)$$

[0028] 式 (III) 中，i 是当前帧在当前超帧中的编码索引，C_{SF} 为当前超帧已消耗计算量，当前超帧剩余计算量为当前目标计算量 TC_{SF} 与当前超帧已消耗计算量 C_{SF} 的差值；$RefN_F(i)$ 是当前帧的参考帧个数，Φ_2 是当前超帧中待编码帧的索引集合，j 是当前超帧中待编码帧的索引；

[0029] (2.4) 对宏块进行计算量分配：

[0030] (2.4.1) 根据当前宏块与参考帧在零矢量处的差值绝对值和以及当前宏块的纹理强度来计算当前宏块的预测计算复杂度 $MbComplexity_{PRED}(n)$，如式 (IV) 所示：

[0031] $$MbComplexity_{PRED}(n) = SAD_{MV00}(n) \times (1 + \frac{DEV_{16\times16}(n)}{\delta}) \qquad (IV)$$

[0032] 式 (IV) 中，n 是宏块索引，SAD_{MV00} 是当前宏块与前向第一个参考帧在零矢量处的像素差值绝对值和，DEV_{16x16} 是当前宏块的纹理强度，δ 是纹理强度的归一化系数，通常根据经验来选取，设为 4000 ～ 32000 之间，本发明优选设为 16000；

[0033] (2.4.2) 在得到每个宏块的预测计算复杂度之后，利用当前帧所有宏块的预测计算复杂度的平均值 $AvgMbComplexity_{PRED}$ 对每个宏块的预测计算复杂度进行归一化，得到每个宏块的计算复杂度权重值 MbWeight，如式 (V) 所示：

[0034] $$MbWeight(n) = \frac{MbComplexity_{PRED}(n)}{AvgMbComplexity_{PRED}} \qquad (V)$$

[0035] (2.4.3) 由式 (V) 计算得到的计算复杂度权重值，并不能十分精确的反映每个宏块将会消耗的计算量，因此为了兼顾宏块间计算量分配的均匀性，提高分配算法的鲁棒性，进一步对每个宏块的计算复杂度权重值 MbWeight 的上限进行限制，如式 (VI) 所示：

[0036] $$MbWeight_{Clip}(n) = \min(MbWeight(n), MbWeight_{MAX}) \qquad (VI)$$

[0037] 式 (VI) 中，$MbWeight_{MAX}$ 为宏块计算复杂度权重上限阈值，通常根据经验来选取，设为 1.0 ～ 5.0 之间，本发明优选设为 2.0；

[0038] (2.4.4) 根据当前帧剩余计算量和剩余宏块的 $MbWeight_{Clip}$ 分配当前宏块的目标计算量 $TC_{Mb}(q)$，如式 (VII) 所示：

[0039] $$TC_{Mb}(q) = (TC_{Frame} - C_{Frame}) \times \frac{MbWeight_{Clip}(q)}{\sum_{n=q}^{N-1} MbWeight_{Clip}(n)} \qquad (VII)$$

[0040] 式 (VII) 中，q 是当前宏块的编码索引，从 0 开始计数；N 是当前帧宏块的个数；C_{Frame} 为当前帧已消耗的计算量，当前帧剩余计算量为当前目标计算量 TC_{Frame} 与当前帧已消耗计算量 C_{Frame} 的差值。

[0041] 其中，所述的步骤 (3) 为：先对宏块帧间模式的估计顺序进行排列，然后按该顺序

对各帧间模式逐一进行估计。

[0042] 由于在帧间模式估计之前进行了模式估计顺序的排列，使最有可能被选为最佳模式的帧间模式排在优先位置，并且模式估计是按该顺序依次进行，排在优选位置上先进行估计的模式可独占使用当前宏块剩余的计算量，因此，在有限的计算量下仍可获得良好的宏块模式率失真性能。这种情况下，每个待估计模式的可获得计算量 AC_{Mode} 按式 (VIII) 计算：

[0043] $$AC_{Mode} = TC_{Mb} - C_{Mb} \qquad (VIII)$$

[0044] 其中，C_{Mb} 代表当前宏块已消耗的计算量，在每个模式估计完成之后进行更新。在结束对上一顺序的帧间模式估计之后，比较当前宏块的目标计算量 TC_{Mb} 与当前宏块已消耗的计算量 C_{Mb} 的差值，如差值小于或等于零，则结束估计；否则，继续进行下一顺序的帧间模式估计。

[0045] 所述的对宏块中各帧间模式的估计顺序进行排列，是根据各帧间模式被选为最佳宏块模式的比例统计、宏块的计算复杂度权重值和划分块纹理强度三要素相结合来进行。

[0046] 将宏块帧间模式 (Inter 模式) 分成三类：第一类只包括 Skip 模式，第二类包括 Inter16×16、Inter16×8 和 Inter8×16 模式，第三类包括 Inter8×8，Inter8×4，Inter4×8，Inter4×4 和 Inter8×8Frext 模式。根据实验统计，第一类模式在所有最佳模式所占的比例最多，且不需要进行帧间估计，计算复杂度可以忽略；第二类模式占最佳模式的比例较多，其帧间估计复杂度较大；第三类模式占最佳模式的比例很少，其帧间估计复杂度很大。因此，将这三类模式采用固定顺序的排列方式，其顺序是：第一类模式总是先进行估计，然后是第二类模式，最后是第三类模式的估计。

[0047] 在第二类模式的估计过程中，采用动态模式排序的方法。根据宏块的计算复杂度权重值和划分块纹理强度 (整体纹理强度、水平划分纹理强度和竖直划分纹理强度) 来排列 Inter16×16、Inter16×8 和 Inter8×16 的估计顺序。如果当前宏块的计算复杂度权重值 MbWeight 大于等于宏块计算复杂度权重上限阈值 $MbWeight_{MAX}$，则认为当前宏块处于高计算复杂度区域，否则就认为当前宏块处于低计算复杂度区域。在低计算复杂度区域，静止的物体较多，宏块更容易选择较大划分的模式，因此将 Inter16×16 较其它两模式先进行估计；而在高计算复杂度区域，运动的物体较多，本发明方法根据划分块纹理强度对 Inter16×16 与 Inter16×8 的估计顺序、以及 Inter16×16 与 Inter8×16 的估计顺序各自进行排列。16×16 块整体纹理强度为 $DEV_{16\times16}$，16×16 块水平划分的纹理强度 ($Blk16\times16DEV_{16\times8}$) 的计算如式 (IX) 所示，16×16 块竖直划分的纹理强度 ($Blk16\times16DEV_{16\times8}$) 的计算如式 (X) 所示：

[0048] $$Blk16\times16DEV_{16\times8} = \sum_{b=1}^{2} DEV_{16\times8}(b) \qquad (IX)$$

[0049] $$Blk16\times16DEV_{8\times16} = \sum_{b=1}^{2} DEV_{8\times16}(b) \qquad (X)$$

[0050] 如果 $Blk16\times16DEV_{16\times8}$ 比 $DEV_{16\times16}$ 小很多，则 Inter16×8 较 Inter16×16 先估计，否则 Inter16×16 较 Inter16×8 先估计。同理对 Inter16×16 和 Inter8×16 的估计顺序进行排列。另外，Inter16×8 与 Inter8×16 之间的估计顺序直接由 $Blk16\times16DEV_{16\times8}$ 和 $Blk16\times16DEV_{8\times16}$ 的大小来决定：$Blk16\times16DEV_{16\times8}$ 较 $Blk16\times16DEV_{8\times16}$ 小，则 Inter16×8

要比 Inter8×16 先进行估计，否则 Inter8×16 要比 Inter16×8 先进行估计。

[0051] 根据实验统计，在第三类模式的估计过程中，由于 Inter8×8Frext 较 Inter8×8 及其子模式被选为最佳模式的次数要少，因此将其估计放到 Inter8×8 及其子模式之后。在 Inter8×8 及其子模式中，Inter4×4 选为最佳模式的比例最少，并且其估计消耗的计算量最多，因此本发明方法将其放到 Inter8×8、Inter8×4 和 Inter4×8 之后进行估计。Inter8×8、Inter8×4 和 Inter4×8 排序同 Inter16×16、Inter16×8、Inter8×16 之间的模式估计顺序排列类似，是根据 8×8 块的整体纹理强度 ($DEV_{8\times8}$)，8×8 块的水平划分纹理强度 ($Blk8\times8DEV_{8\times4}$) 和 8×8 块的竖直划分纹理强度 ($Blk8\times8DEV_{4\times8}$) 来排序。$Blk8\times8DEV_{8\times4}$ 和 $Blk8\times8DEV_{4\times8}$ 的计算分别如式 (XI) 和 (XII) 所示：

[0052] $$Blk8\times8DEV_{8\times4}=\sum_{b=1}^{2}DEV_{8\times4}(b) \qquad (XI)$$

[0053] $$Blk8\times8DEV_{4\times8}=\sum_{b=1}^{2}DEV_{4\times8}(b) \qquad (XII)$$

[0054] 即，对宏块帧间模式的估计顺序的排列为：

[0055] 首先进行第一类 Skip 模式的估计；接着进行第二类 Inter16×16、Inter16×8 和 Inter8×16 模式的排列和估计；然后进行第三类模式中的 Inter8×8、Inter8×4 和 Inter4×8 模式的排列和估计；再进行第三类模式中的 Inter4×4 模式的估计；最后进行第三类模式中的 Inter8×8Frext 模式的估计。

[0056] 每个帧间模式估计过程中，帧间模式的不同划分块的计算量分配采用剩余计算量均分的方法，即将当前模式剩余计算量均分给待估计的划分块，每个待估计划分块的可获得计算量 AC_{Block} 按式 (XIII) 计算：

[0057] $$AC_{Block}=\frac{AC_{Mode}-C_{Mode}}{N_{Block}} \qquad (XIII)$$

[0058] 其中 C_{Mode} 是当前模式估计已消耗的计算量，N_{Block} 是剩余待估计的划分块的个数。

[0059] 其中，所述的步骤 (4) 为：

[0060] 按每个划分块的各参考帧的帧间估计顺序对各参考帧逐一进行估计，所述的参考帧的帧间估计顺序如下：

[0061] 在每个划分块的帧间估计过程中，先进行前向队列的参考帧的帧间搜索，再进行后向队列的参考帧的帧间搜索，最后进行双向预测的帧间搜索。考虑到划分块在选择时域方向的预测比视点方向的预测要多，因此在对前向或后向参考队列中的参考帧进行帧间估计顺序排列的时候，始终将时域方向的参考帧排在视点方向参考帧的前面进行估计。这种情况下，排在前面位置上先进行估计的参考帧可独占使用当前划分块剩余的计算量，每个待估计参考帧的可获得计算量 AC_{Search} 按式 (XIV) 计算：

[0062] $$AC_{Search}=AC_{Block}-C_{Block} \qquad (XIV)$$

[0063] 其中 C_{Block} 是当前划分块的帧间估计已消耗的计算量。在每个参考帧的帧间估计之前，确定该次帧间估计的最大帧间搜索次数，如果帧间搜索次数达到了最大帧间搜索次数，就中止该次帧间估计。所述的帧间估计的最大帧间搜索次数由待估计参考帧的可获得计算量 AC_{Search} 除以当前划分块单次帧间搜索的计算量得到。

[0064] 在上述步骤(1)～(4)的每个处理过程结束之后，都要对已消耗计算量进行统计，并对相关控制参数进行更新，具体如下：

[0065] 在划分块中每次参考帧的帧间估计结束之后，需要根据该次帧间估计的帧间搜索次数和单次帧间搜索计算量来计算划分块在该次参考帧的帧间估计的计算量，然后用于更新当前划分块的帧间估计已消耗计算量 C_{Block}；在模式中每个划分块的帧间估计完成之后，需要更新当前模式已消耗的计算量 C_{Mode}；在宏块中每个模式估计完成之后，需要更新当前宏块已消耗计算量 C_{Mb}；在一帧中每个宏块编码完成之后，需要更新当前帧已消耗计算量 C_{Frame}；当超帧中的一个帧编码完成，需要更新当前超帧已消耗计算量 C_{SF}；当 GGOP 中的一个超帧编码完成，需要更新当前 GGOP 中已消耗计算量 C_{GGOP}。

[0066] 由于运动场景的变化，导致分层 B 帧预测结构中各时域层之间的计算复杂度差异也在变化。当图像静止区域较多的时候，各层之间计算复杂度相互接近；而当图像的运动区域较多的时候，各层之间的计算复杂度差异就变大。因此为了提高编码效率，需要对各时域层的计算复杂度权重因子 W_{Layer} 进行动态调整。在当前 GGOP 编码结束后，本发明方法利用 GGOP 中各时域层的平均 SAD_{MV00}($AvgSAD_{MV00}$) 来自适应更新计算复杂度权重因子，每层的计算复杂度权重因子更新如式 (XV) 所示：

[0067] $$W_{Layer}(r+1,l)=\eta\times W_{Layer}(r,l)+(1-\eta)\times\frac{AvgSAD_{MV00}(r,l)}{AvgSAD_{MV00}(r,L_{MAX})} \quad (XV)$$

[0068] 其中 r 是当前 GGOP 的索引，1 是分层 B 帧预测结构时域层数索引，L_{MAX} 是最大层数索引，η 是时域权重因子，通常根据经验来选取，设为 0.1～0.9，本发明中优选设为 0.5。

[0069] 另外，在当前 GGOP 编码结束后，根据当前 GGOP 目标计算量和实际编码消耗计算量来更新 GGOP 计算量虚拟缓冲区，如式 (XVI) 所示：

[0070] $VBC_{GGOP}(r+1) = TC_{GGOP}(r)-C_{GGOP}(r)$ (XVI)。

[0071] 本发明的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，以实现在实际应用中多视点视频编码计算复杂度的精确控制。首先对 GGOP、超帧、帧和宏块不同粒度的编码单元的计算量进行分配，从而控制 GGOP、超帧、帧和宏块不同粒度的编码单元的计算复杂度；然后对宏块帧间模式估计进行计算量分配，从而控制宏块帧间模式估计的计算复杂度；接着对每个划分块的各参考帧的帧间估计的计算量进行分配，从而控制对每个划分块的帧间估计的计算复杂度。

[0072] 本发明中，主要对帧间预测的计算复杂度进行控制。由于多视点视频编码在进行帧间预测的时候采用了同 H.264/AVC 一样的可变块帧间预测技术，每个宏块的帧间预测划分为 16×16，16×8，8×16，8×8，8×8，8×4，4×8，4×4 等 7 种不同粒度的模式进行帧间估计，因此帧间预测由于需要对多个模式进行帧间估计，是整个编码过程中计算量最为集中部分。而根据对多视点视频编码参考代码中各编码模块的运行时间统计，也发现帧间预测占据了绝大部分的编码时间。所以，对帧间预测的计算复杂度进行控制，就可以控制多视点视频编码的计算复杂度。

[0073] 本发明中，对超帧计算量进行分配时，既考虑超帧所在的时域层数的帧间预测计算复杂度，又考虑到超帧中各帧的帧间预测参考帧个数。这是因为在多视点视频编码的分层 B 帧预测结构中，处在不同时域层的帧与其参考帧之间时域间隔不同，造成不同时域层的帧之间在帧间预测计算复杂度上的差异。如果超帧处于较小的时域层，由于超帧中的各

帧与其参考帧之间的时域间隔较大，各帧的帧间预测计算复杂度较大，因此需要给超帧分配更多的计算量。同时，超帧的帧间预测计算复杂度与其中各帧的参考帧个数有直接关系，参考帧个数越多，计算复杂度也越大。所以，本发明可以在控制超帧计算复杂度的同时保持图像质量。

[0074] 本发明中，对宏块计算量进行分配时，在每帧编码开始前先对所有宏块的计算复杂度进行预测，建立宏块计算复杂度权重表。主要是考虑到在一帧图像中，宏块之间由于运动状态和纹理特征的差异，它们的帧间预测计算复杂度存在较大差异：(1) 对于帧间静止块，其最佳匹配块都集中在零矢量附近，帧间估计快速算法可以较快的选定最佳匹配块，估计过程中消耗的计算量比较少；而对于帧间运动块，其运动轨迹并不确定，帧间估计快速算法需要通过增加帧间搜索次数来选取最佳匹配块，估计过程中消耗的计算量比较多；(2) 纹理复杂的块更容易选用较小的划分，其模式估计的计算复杂度更高，且纹理复杂的块较纹理简单块更难获得准确的匹配块，其帧间估计的复杂度也更高。因此，本发明方法根据宏块的运动状态和纹理特征建立的宏块计算复杂度权重表，能够准确的给每个宏块分配计算量。

[0075] 本发明中，对帧间模式估计计算量进行分配时，先对各帧间模式的估计顺序进行排列，然后逐一进行估计。由于宏块的最佳模式只有一个，因此在帧间模式估计顺序排列的时候，根据各帧间模式被选为最佳宏块模式的比例统计、宏块的计算复杂度权重值和划分块纹理强度等三要素相结合来将被选为最佳模式可能性大的帧间模式排列在前。

[0076] 与现有技术相比，本发明具有以下的有益效果：

[0077] 本发明的多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，对不同粒度的编码单元、模式估计和帧间估计进行计算量的多层次自适应分配和控制。该方法可以准确控制多视点视频编码的整体计算量，并减少计算量的波动，同时保持编码率失真性能，适用于多视点视频编码计算复杂度的控制。

附图说明

[0078] 图 1 为本发明方法的基本流程图；

[0079] 图 2 为 Inter16×16、Inter16×8 和 Inter8×16 模式的估计顺序排列图；

[0080] 图 3 为 Inter8×8、Inter8×4 和 Inter4×8 模式的估计顺序排列图；

[0081] 图 4 为宏块帧间模式估计流程图；

[0082] 图 5 为序列③在不同目标计算量下的 GGOP 计算量曲线图。

具体实施方式

[0083] 下面结合实施例和附图来详细说明本发明，但本发明并不仅限于此。

[0084] 如图 1 所示，一种多视点视频编码分层 B 帧预测结构的计算复杂度控制方法，包括以下步骤：

[0085] (1) 输入多视点视频数据；

[0086] (2) 对不同粒度编码单元进行计算量分配；

[0087] (3) 对宏块帧间模式估计进行计算量分配；

[0088] (4) 对帧间估计进行计算量分配；

[0089] 所述的不同粒度编码单元包括：GGOP、超帧(Super Frame，SF)、帧和宏块四个不同粒度的编码单元。其中，GGOP(Group of GOP)为GOP组，是指不同视点在同一时间段的所有图像组的组合，所述的图像组(GOP，Group of Picture)是指某一视点在某一时间段的图像的组合；所述的超帧是指同一个时刻不同视点的所有帧的组合。

[0090] 步骤(2)具体为：

[0091] (2.1)对GGOP进行计算量分配：

[0092] 在每个GGOP开始编码之前，计算GGOP的目标计算量TC_{GGOP}，如式(I)所示：

[0093] $TC_{GGOP}(r) = N_{SF}(r) \times TargetAvgC_{SF} + \min(VBC_{GGOP}(r), \alpha \times TargetAvgC_{SF})$ (I)

[0094] 式(I)中，r是当前GGOP编码索引，N_{SF}是当前GGOP中的超帧个数；$TargetAvgC_{SF}$是超帧的目标平均计算量；VBC_{GGOP}是GGOP计算量虚拟缓冲区，为前一GGOP实际计算量与其目标计算量之间的偏差。VBC_{GGOP}的初始值为0，其在每个GGOP编码结束后根据TC_{GGOP}和GGOP实际消耗计算量来更新。α为VBC_{GGOP}的上限控制参数，通常根据经验来选取，设为0.1～4.0之间，此处设为1.0；$\sigma \times TargetAvgC_{SF}$为$VBC_{GGOP}$的上限值。

[0095] 由于分层B帧预测结构将第一个超帧单独算作一个GGOP，为第一个GGOP，并作为其后的GGOP的参考基础，因此其图像质量对其后的GGOP有重要影响。

[0096] 为了使得第一个GGOP获得较高的图像质量，在对GGOP进行计算量分配时，对第一个GGOP不进行计算量分配，而是将第一个GGOP与第二个GGOP合并作为一个GGOP来进行计算量分配，该合并的GGOP中的超帧包括第一个超帧(第一个GGOP)和第二个的GGOP中的超帧，N_{SF}值为两个GGOP中的超帧个数之和，即第二个的GGOP中的超帧个数加1。这样，可以有效提高后续的GGOP的编码效果。

[0097] (2.2)对超帧进行计算量分配：

[0098] 根据超帧中各帧的帧间预测参考帧个数和所处的时域层计算复杂度权重因子计算每个超帧的计算复杂度权重值W_{SF}，如式(II′)所示：

[0099] $W_{SF}(m) = W_{Layer} \times RefN_{SF}(m)$ (II′)

[0100] 式(II′)中，m是超帧编码索引；W_{Layer}代表超帧m所处时域层的计算复杂度权重因子，初始值根据经验来设定，所处时域层的层数越低，W_{Layer}初始值越大，W_{Layer}的更新如式(XV)所示；$RefN_{SF}$代表超帧m中各帧的帧间预测用到的参考帧个数；

[0101] 再根据超帧计算复杂度权重值W_{SF}和当前GGOP剩余计算量，分配当前编码超帧目标计算量$TC_{SF}(k)$，如式(II)所示：

[0102] $$TC_{SF}(k) = (TC_{GGOP} - C_{GGOP}) \times \frac{W_{SF}(k)}{\sum_{m \in \Phi_1} W_{SF}(m)} \quad (II)$$

[0103] 式(II)中，k是当前超帧在当前GGOP中的编码索引，C_{GGOP}是当前GGOP已消耗计算量，当前GGOP剩余计算量为当前GGOP的目标计算量TC_{GGOP}与当前GGOP已消耗计算量C_{GGOP}的差值；Φ_1是当前GGOP中待编码超帧的索引集合；

[0104] (2.3)对帧进行计算量分配：

[0105] 根据当前超帧剩余计算量和待编码帧的帧间预测参考帧个数分配当前编码帧目标计算量$TC_F(i)$，如式(III)所示：

[0106] $$TC_F(i)=(TC_{SF}-C_{SF})\times\frac{RefN_F(i)}{\sum_{j\in\Phi_2}RefN_F(j)} \quad (III)$$

[0107] 式 (III) 中，i 是当前帧在当前超帧中的编码索引，C_{SF} 为当前超帧已消耗计算量，当前超帧剩余计算量为当前目标计算量 TC_{SF} 与当前超帧已消耗计算量 C_{SF} 的差值；$RefN_F(i)$ 是当前帧的参考帧个数，Φ_2 是当前超帧中待编码帧的索引集合，j 是当前超帧中待编码帧的索引；

[0108] (2.4) 对宏块进行计算量分配：

[0109] (2.4.1) 根据当前宏块与参考帧在零矢量处的差值绝对值和以及当前宏块的纹理强度来计算当前宏块的预测计算复杂度 $MbComplexity_{PRED}(n)$，如式 (IV) 所示：

[0110] $$MbComplexity_{PRED}(n)=SAD_{MV00}(n)\times(1+\frac{DEV_{16\times16}(n)}{\delta}) \quad (IV)$$

[0111] 式 (IV) 中，n 是宏块索引，SAD_{MV00} 是当前宏块与前向第一个参考帧在零矢量处的像素差值绝对值和，DEV_{16x16} 是当前宏块的纹理强度，δ 是纹理强度的归一化系数，通常根据经验来选取，设为 4000 ～ 32000 之间，此处设为 16000。

[0112] 当前宏块的纹理强度 DEV_{16x16} 的计算可参考 W×H 块的纹理强度 DEV_{WxH} 计算，其中，W 和 H 均取值 16。

[0113] W×H 块的纹理强度 DEV_{WxH} 的计算如式 (IV′) 所示：

[0114] $$DEV_{W\times H}=\sum_{h=1}^{H}\sum_{w=1}^{W}|Pixel(w,h)-AVG_{W\times H}| \quad (IV')$$

[0115] 式 (IV′) 中 Pixel(w, h) 代表 W×H 块中水平索引为 w，竖直索引为 h 的像素值，$AVG_{W\times H}$ 代表 W×H 块中的平均像素值。$DEV_{W\times H}$ 值较小，说明 W×H 块中各像素之间差异较小，纹理特征比较简单；$DEV_{W\times H}$ 值较大，说明 W×H 块中各像素之间的差异较大，其纹理特征较为复杂。

[0116] (2.4.2) 在得到每个宏块的预测计算复杂度之后，利用当前帧所有宏块的预测计算复杂度 $MbComplexity_{PRED}$ 的平均值 $AvgMbComplexity_{PRED}$ 对每个宏块的预测计算复杂度进行归一化，得到每个宏块的计算复杂度权重值 MbWeight，如式 (V) 所示：

[0117] $$MbWeight(n)=\frac{MbComplexity_{PRED}(n)}{AvgMbComplexity_{PRED}} \quad (V)$$

[0118] (2.4.3) 由式 (V) 计算得到的计算复杂度权重值，并不能十分精确的反映每个宏块将会消耗的计算量，因此为了兼顾宏块间计算量分配的均匀性，提高分配算法的鲁棒性，进一步对每个宏块的计算复杂度权重值 MbWeight 的上限进行限制，如式 (VI) 所示：

[0119] $$Mb\ Weight_{Clip}(n)=\min(MbWeight(n),Mb\ Weight_{MAX}) \quad (VI)$$

[0120] 式 (VI) 中，$MbWeight_{MAx}$ 为宏块计算复杂度权重上限阈值，通常根据经验来选取，设为 1.0 ～ 5.0 之间，此处设为 2.0。

[0121] (2.4.4) 根据当前帧剩余计算量和剩余宏块的 $MbWeight_{Clip}$ 分配当前宏块的目标计算量 $TC_{Mb}(q)$，如式 (VII) 所示：

[0122] $$TC_{Mb}(q)=(TC_{Frame}-C_{Frame})\times\frac{MbWeight_{Clip}(q)}{\sum_{n=q}^{N-1}MbWeight_{Clip}(n)} \quad (VII)$$

[0123] 式 (VII) 中，q 是当前宏块的编码索引，从 0 开始计数；N 是当前帧宏块的个数；C_{Frame} 为当前帧已消耗的计算量，当前帧剩余计算量为当前目标计算量 TC_{Frame} 与当前帧已消耗计算量 C_{Frame} 的差值。

[0124] 步骤 (3) 具体为：先对宏块帧间模式的估计顺序进行排列，然后按该顺序对各帧间模式逐一进行估计。

[0125] 由于在帧间模式估计之前进行了模式估计顺序的排列，使最有可能被选为最佳模式的帧间模式排在优先位置，并且模式估计是按该顺序依次进行，排在优选位置上先进行估计的模式可独占使用当前宏块剩余的计算量，因此，在有限的计算量下仍可获得良好的宏块模式率失真性能。这种情况下，每个待估计模式的可获得计算量 AC_{Mode} 按式 (VIII) 计算：

[0126] $$AC_{Mode} = TC_{Mb}-C_{Mb} \quad (VIII)$$

[0127] 其中，C_{Mb} 代表当前宏块已消耗的计算量，在每个模式估计完成之后进行更新。在结束对上一顺序的帧间模式估计之后，比较当前宏块的目标计算量 TC_{Mb} 与当前宏块已消耗的计算量 C_{Mb} 的差值，如差值小于或等于零，则结束估计；否则，继续进行下一顺序的帧间模式估计。

[0128] 所述的对宏块中各帧间模式的估计顺序进行排列，是根据各帧间模式被选为最佳宏块模式的比例统计、宏块的计算复杂度权重值和划分块纹理强度等三要素相结合来进行的。

[0129] 将宏块帧间模式（Inter 模式）分成三类：第一类只包括 Skip 模式，第二类包括 Inter16×16、Inter16×8 和 Inter8×16 模式，第三类包括 Inter8×8，Inter8×4，Inter4×8，Inter4×4 和 Inter8×8Frext 模式。根据实验统计，第一类模式在所有最佳模式所占的比例最多，且不需要进行帧间估计，计算复杂度可以忽略；第二类模式占最佳模式的比例较多，其帧间估计复杂度较大；第三类模式占最佳模式的比例很少，其帧间估计复杂度很大。因此，将这三类模式采用固定顺序的排列方式，其顺序是：第一类模式总是先进行估计，然后是第二类模式，最后是第三类模式的估计。

[0130] 在第二类模式的估计过程中，采用动态模式排序的方法。根据宏块的计算复杂度权重值和划分块纹理强度（整体纹理强度、水平划分纹理强度和竖直划分纹理强度）来排列 Inter16×16、Inter16×8 和 Inter8×16 的估计顺序。如果当前宏块的计算复杂度权重值 MbWeight 大于等于宏块计算复杂度权重上限阈值 $MbWeight_{MAX}$，则认为当前宏块处于高计算复杂度区域，否则就认为当前宏块处于低计算复杂度区域。在低计算复杂度区域，静止的物体较多，宏块更容易选择较大划分的模式，因此将 Inter16×16 较其它两模式先进行估计；而在高计算复杂度区域，运动的物体较多，本发明方法根据划分块纹理强度对 Inter16×16 与 Inter16×8 的估计顺序、以及 Inter16×16 与 Inter8×16 的估计顺序各自进行排列。另外，Inter16×8 与 Inter8×16 的估计顺序始终根据水平划分纹理强度和竖直划分纹理强度的大小进行排列。

[0131] 每个宏块只有处在高计算复杂度区域，同时其 16×16 块整体纹理强度

($DEV_{16\times16}$) 与水平划分纹理强度或竖直划分纹理强度存在显著差异的情况下，则其较小划分模式 (Inter16×8 或 Inter8×16) 先进行预测。16×16 块水平划分的纹理强度 ($Blk16\times16DEV_{16\times8}$) 的计算如式 (IX) 所示，16×16 块竖直划分的纹理强度 ($Blk16\times16DEV_{16\times8}$) 的计算如式 (X) 所示：

[0132] $$Blk16\times16DEV_{16\times8}=\sum_{b=1}^{2}DEV_{16\times8}(b) \qquad (IX)$$

[0133] $$Blk16\times16DEV_{8\times16}=\sum_{b=1}^{2}DEV_{8\times16}(b) \qquad (X)$$

[0134] 如果 $DEV_{16\times16}$、$Blk16\times16DEV_{16\times8}$ 和 $Blk16\times16DEV_{8\times16}$ 三者值相近，则 16×16 块中平均像素值与 16×8 块或 8×16 块中的平均像素值相近，它们的纹理强度一致。在这种情况下，认为整个 16×16 块属于同质物体，其运动特性一致，因此 Inter16×16 较 Inter16×8 和 Inter8×16 优先进行估计；如果 $Blk16\times16DEV_{16\times8}$ 比 $DEV_{16\times16}$ 小很多，则说明 16×16 块在竖直方向上有明显差异，使 $DEV_{16\times16}$ 较大，而水平方向基本同质，Blk16×16DEV16×8 较小，因此在这情况下，Inter16×8 较 Inter16×16 先估计；同理，如果 $Blk16\times16DEV_{8\times16}$ 比 $DEV_{16\times16}$ 小很多，则 Inter8×16 较 Inter16×16 先估计。

[0135] Inter16×8 与 Inter8×16 之间的估计顺序直接由 $Blk16\times16DEV_{16\times8}$ 和 $Blk16\times16DEV_{8\times16}$ 的大小来决定：$Blk16\times16DEV_{16\times8}$ 较小，则说明 16×8 划分（水平划分）比 8×16 划分（竖直划分）更适合于纹理特征，Inter16×8 要比 Inter8×16 先进行估计；$Blk16\times16DEV_{8\times16}$ 较小，则说明 8×16 划分比 16×8 划分更适合于纹理特征，Inter8×16 要比 Inter16×8 先进行估计。Inter16×16、Inter16×8 和 Inter8×16 之间的估计顺序排列如图 2(a)、(b) 和 (c)，其中 μ 值通常根据经验来选取，设为 0.2 ～ 0.8，此处设为 0.4。

[0136] 根据实验统计，在第三类模式的估计过程中，由于 Inter8×8Frext 较 Inter8×8 及其子模式被选为最佳模式的次数要少，因此将其估计放到 Inter8×8 及其子模式之后。在 Inter8×8 及其子模式中，Inter4×4 选为最佳模式的比例最少，并且其估计消耗的计算量最多，因此本发明方法将其放到 Inter8×8、Inter8×4 和 Inter4×8 之后进行估计。8×8 块水平划分和竖直划分的纹理强度计算分别如式 (XI) 和 (XII) 所示：

[0137] $$Blk8\times8DEV_{8\times4}=\sum_{b=1}^{2}DEV_{8\times4}(b) \qquad (XI)$$

[0138] $$Blk8\times8DEV_{4\times8}=\sum_{b=1}^{2}DEV_{4\times8}(b) \qquad (XII)$$

[0139] 同 Inter16×16、Inter16×8、Inter8×16 之间的模式估计顺序排列类似，Inter8×8、Inter8×4 和 Inter4×8 也是根据 $DEV_{8\times8}$、$Blk8\times8DEV_{8\times4}$、$Blk8\times8DEV_{4\times8}$ 这三者关系进行排序，具体如图 3(a)、(b) 和 (c)。

[0140] 综上所述，所有宏块帧间模式的估计流程如图 4 所示：首先进行第一类 Skip 模式的估计；接着进行第二类 Inter16×16、Inter16×8 和 Inter8×16 模式的排列和估计；然后进行第三类模式中的 Inter8×8、Inter8×4 和 Inter4×8 模式的排列和估计；再进行第三类模式中的 Inter4×4 模式的估计；最后进行第三类模式中的 Inter8×8Frext 模式的估计。

[0141] 每个帧间模式估计过程中，帧间模式的不同划分块的计算量分配采用剩余计算量均分的方法，即将当前模式剩余计算量均分给待估计的划分块，每个待估计划分块的可获得计算量 AC_{Block} 按式 (XIII) 计算：

[0142] $$AC_{Block} = \frac{AC_{Mode} - C_{Mode}}{N_{Block}} \qquad (XIII)$$

[0143] 其中 C_{Mode} 是当前模式估计已消耗的计算量，N_{Block} 是剩余待估计的划分块的个数。

[0144] 步骤 (4) 具体为：

[0145] 按每个划分块的各参考帧的帧间估计顺序对各参考帧逐一进行估计，所述的参考帧的帧间估计顺序如下：

[0146] 在每个划分块的帧间估计过程中，先进行前向队列的参考帧的帧间搜索，再进行后向队列的参考帧的帧间搜索，最后进行双向预测的帧间搜索。考虑到划分块在选择时域方向的预测比视点方向的预测要多，因此在对前向或后向参考队列中的参考帧进行帧间估计顺序排列的时候，始终将时域方向的参考帧排在视点方向参考帧的前面进行估计。

[0147] 采取所述的参考帧的帧间估计顺序，使最有可能被选为最佳参考帧的候选参考帧排在优先位置，并且帧间估计是按该顺序依次进行，排在优选位置上先进行估计的参考帧可独占使用当前划分块剩余的计算量，因此，在有限的计算量下仍可获得良好的帧间估计率失真性能。每个待估计参考帧的可获得计算量 AC_{Search} 按式 (XIV) 计算：

[0148] $$AC_{Search} = AC_{Block} - C_{Block} \qquad (XIV)$$

[0149] 其中 C_{Block} 是当前划分块的帧间估计已消耗的计算量。在每个参考帧的帧间估计之前，确定该次帧间估计的最大帧间搜索次数，如果帧间搜索次数达到了最大帧间搜索次数，就中止该次帧间估计。所述的帧间估计的最大帧间搜索次数由待估计参考帧的可获得计算量 AC_{Search} 除以当前划分块单次帧间搜索的计算量得到。

[0150] 在上述步骤 (1) ～ (4) 的每个处理过程结束之后，都要对已消耗计算量进行统计，并对相关控制参数进行更新，具体如下：

[0151] 在划分块中每次参考帧的帧间估计结束之后，需要根据该次帧间估计的帧间搜索次数和单次帧间搜索计算量来计算划分块在该次参考帧的帧间估计的计算量，然后用于更新当前划分块的帧间估计已消耗计算量 C_{Block}；在模式中每个划分块的帧间估计完成之后，需要更新当前模式已消耗的计算量 C_{Mode}；在宏块中每个模式估计完成之后，需要更新当前宏块已消耗计算量 C_{Mb}；在一帧中每个宏块编码完成之后，需要更新当前帧已消耗计算量 C_{Frame} 当超帧中的一个帧编码完成，需要更新当前超帧已消耗计算量 C_{SF}；当 GGOP 中的一个超帧编码完成，需要更新当前 GGOP 中已消耗计算量 C_{GGOP}。

[0152] 由于运动场景的变化，导致分层 B 帧预测结构中各时域层之间的计算复杂度差异也在变化。当图像静止区域较多的时候，各层之间计算复杂度相互接近；而当图像的运动区域较多的时候，各层之间的计算复杂度差异就变大。因此为了提高编码效率，需要对各时域层的计算复杂度权重因子 W_{Layer} 进行动态调整。在当前 GGOP 编码结束后，本发明方法利用 GGOP 中各时域层的平均 SAD_{MV00} ($AvgSAD_{MV00}$) 来自适应更新计算复杂度权重因子，每层的计算复杂度权重因子更新如式 (XV) 所示：

[0153] $$W_{Layer}(r+1,l) = \eta \times W_{Layer}(r,l) + (1-\eta) \times \frac{AvgSAD_{MV00}(r,l)}{AvgSAD_{MV00}(r,L_{MAX})} \qquad (XV)$$

[0154] 其中，r 是当前 GGOP 的索引，l 是分层 B 帧预测结构时域层数索引，L_{MAX} 是最大层数索引，η 是时域权重因子，通常根据经验来选取，设为 0.1 ～ 0.9，此处设为 0.5。

[0155] 另外，在当前 GGOP 编码结束后，根据当前 GGOP 目标计算量和实际编码消耗计算量来更新 GGOP 计算量虚拟缓冲区，如式 (XVI) 所示：

[0156] $$VBC_{GGOP}(r+1) = TC_{GGOP}(r) - C_{GGOP}(r) \qquad (XVI)$$

[0157] 性能评估实验：

[0158] 实验在多视点视频编码参考代码 JMVC4.0 上进行，整体测试配置以多视点视频编码通用测试条件为基础 (Su Y P, Vetro A, Smolic A. Common test conditions for multiview video coding. Doc. U211, JVT 21st meeting, Hangzhou, 2006)。JMVC 的搜索模式选用了其快速搜索算法，搜索范围设置为 48，基础量化参数 QP 选用 22、27、32 和 37。实验选用了四个典型的多视点视频测试序列：序列① (MERL 的 Exit 序列)、序列② (MERL 的 Ballroom 序列)，序列③ (KDDI 的 Race1 序列) 和序列④ (Tanimoto 实验室的 Rena 序列)。实验选取这些序列的前三个视点，采用了 HHI 提出的多视点视频编码分层 B 帧预测结构，其时域前向参考帧个数、时域后向参考帧个数、视点前向参考帧个数和视点后向参考帧个数为 1。式 (XV) 中，分层 B 帧预测结构的时域第 0 层到第 4 层的计算复杂度权重因子初始值分别设为：4.0、1.5、1.3、1.1 和 1.0。

[0159] 实验采用不同粒度划分块帧间搜索的权重计算量来衡量编码计算复杂度，实现编码计算复杂度的客观量化。不同粒度划分块单次帧间搜索的权重计算量如表 1 所示。在表 1 中，实验将 16×16 划分块进行单次帧间搜索的权重计算量设为 100，其它划分块的权重计算量是通过其与 16×16 划分块在单次帧间搜索的处理时间上的比例来计算。

[0160] 表 4.1 不同粒度划分块单次帧间搜索的权重计算量

[0161] | 划分块类型 | 权重计算量 |
|---|---|

[0162] | 16×16 | 100 |

[0163] | 16×8 | 51 |

[0164] | 8×16 | 56 |

[0165] | 8×8 | 29 |

[0166] | 8×4 | 16 |

[0167] | 4×8 | 18 |

[0168] | 4×4 | 10 |

[0169] 实验开始的时候，先在不做计算复杂度控制的情况下，统计各序列在不同 QP 设置下的计算量，并将这些计算量作为相应的初始计算量；然后启动计算复杂度控制算法，选用这些初始计算量的 10％、30％、50％、70％和 90％作为目标计算量来对计算复杂度控制算法进行性能测试。

[0170] 图 5 所示为基础量化参数 QP 为 32 设置下，序列③在不同的目标计算量设置下 GGOP 计算量曲线以及不做计算复杂度控制的 GGOP 初始计算量曲线，曲线 1 ～ 6 分别对应不做计算复杂度控制的 GGOP 初始计算量曲线、以初始计算量的 10％、30％、50％、70％和 90％作为目标计算量的 GGOP 计算量曲线。从图 5 中可以看出，在使用了本发明方法之后，GGOP 计算量曲线在不同的目标计算量下的都表现平稳，不会随着图像运动特征变化而波动。从图 5 中还可以看到，GGOP 初始计算量曲线有较大波动，尤其是前面 10 个 GGOP 的初始计算

量波动较大，而后面几个 GGOP 开始恢复平稳。这是由于编码帧间预测的计算量随着视频运动特征的变化而有所起伏：序列③的前半段由于摄像头的快速转动和短暂停止而让运动特征存在较大变化，而其后半段的摄像头都处于静止，图像内容变化较小，因此前后 GGOP 的初始计算量变化较大。从上面的比较可以看出，在做过计算复杂度控制算法之后，GGOP 计算量的波动得到了有效的减小。

[0171] 为了在不同的基础量化参数 QP 设置条件下，对本发明方法在不同目标计算量下的率失真性能进行评估，实验将其以初始计算量下的率失真性能为参考，统计信噪比变化 (BDPSNR，即 Bjontegaard delta PSNR) 和码率变化 (BDBR，即 Bjontegaard delta bit rate)。其中 BDPSNR 为负数或 BDBR 为正数代表算法率失真性能的降低。另外，为了评估本发明方法对整体计算量的控制准确度，实验统计实际消耗计算量 (RC，即 Real Complexity)，并用与初始计算量的百分比来度量。

[0172] 与初始计算量下的编码性能相比，本发明方法在不同目标计算量下的编码性能如表 2 ～表 6 所示。从这些表中可以看出，在 10％的目标计算量下，本发明方法的率失真性能降低最大，但平均也只有 0.19dB 的 BDPSNR 下降，以及 5.3％的 BDBR 增加；而在 30％、50％、70％和 90％的目标计算量设置下，本发明方法的率失真性能降低很小，其中在 30％目标计算量下平均有 0.03dB 的 BDPSNR 下降和 0.8％的 BDBR 的增加，在 50％目标计算量下平均有 0.01dB 的 BDPSNR 下降和 0.2％的 BDBR 的增加，在 70％和 90％目标运算下的率失真性能几乎保持不变。另外，在整体计算量控制方面，不同条件下的实际计算量都小于目标计算量，并且两者非常的接近，在 10％、30％、50％、70％目标计算量设置下，得到的实际计算量同目标计算量相同。在 90％目标计算量设置下，得到的实际计算量比目标计算量要稍小，但差异也仅在 2％以内。

[0173] 从上面的实验数据可以看出，本发明方法能够准确控制整体计算量，减少计算量的波动，同时保持了良好的编码率失真性能。

[0174] 表 2 本发明方法在 10％目标计算量设置下的编码性能

[0175]

序列名称	与初始计算量下的性能比较		
	BDPSNR (dB)	BDBR (%)	RC (%)
序列①	-0.13	5.4	10.0
序列②	-0.20	5.3	10.0
序列③	-0.24	6.2	10.0
序列④	-0.19	4.1	10.0
平均	-0.19	5.3	10.0

[0176] 表 3 本发明方法在 30％目标计算量设置下的编码性能

[0177]

序列名称	与初始计算量下的性能比较		
	BDPSNR (dB)	BDBR (%)	RC (%)
序列①	-0.02	0.5	30.0
序列②	-0.03	0.9	30.0
序列③	-0.04	1.1	30.0
序列④	-0.03	0.6	30.0
平均	-0.03	0.8	30.0

[0178] 表 4 本发明方法在 50% 目标计算量设置下的编码性能

[0179]

序列名称	与初始计算量下的性能比较		
	BDPSNR (dB)	BDBR (%)	RC (%)
序列①	0.00	0.1	50.0
序列②	-0.01	0.2	50.0
序列③	-0.01	0.3	50.0
序列④	-0.01	0.1	50.0
平均	-0.01	0.2	50.0

[0180] 表 5 本发明方法在 70% 目标计算量设置下的编码性能

[0181]

序列名称	与初始计算量下的性能比较		
	BDPSNR (dB)	BDBR (%)	RC (%)
序列①	0.00	0.0	70.0
序列②	0.00	0.0	70.0
序列③	0.00	0.1	70.0
序列④	0.00	0.0	70.0
平均	0.00	0.0	70.0

[0182] 表 6 本发明方法在 90% 目标计算量设置下的编码性能

[0183]

序列名称	与初始计算量下的性能比较		
	BDPSNR (dB)	BDBR (%)	RC (%)
序列①	0.00	0.0	89.7
序列②	0.00	0.0	89.6
序列③	0.00	0.1	89.2
序列④	0.00	0.0	88.2
平均	0.00	0.0	89.3

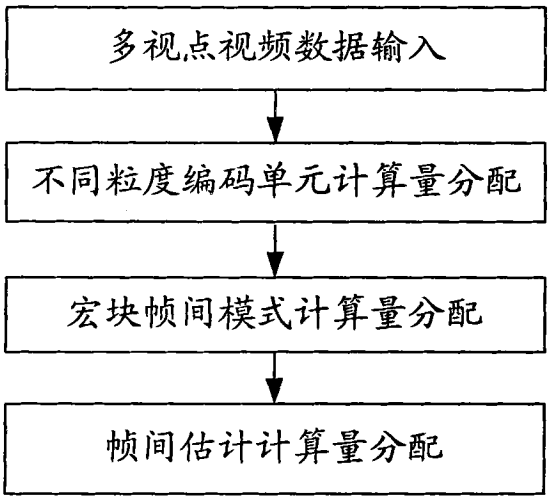


图 1

```
If ( MbWeight >= MbWeight_MAX
  && B1k16 × 16DEV_16×8 < μ × DEV_16×16)
{
    Inter16×8较Inter16×16先估计
}
else
{
    Inter16×16较Inter16×8先估计
}
```

(a)

```
iF ( MbWeight >= MbWeight_MAX
  && B1k16 × 16DEV_8×16 < μ × DEV_16×16)
{
    Inter8×16较Inter16×16先估计
}
else
{
    Inter16×16较Inter8×16先估计
}
```

(b)

```
if (B1k16 × 16DEV_16×8 < B1k16 × 16DEV_8×16)
{
    Inter16×8较Inter8×16先估计
}
else
{
    Inter8×16较Inter16×8先估计
}
```

(c)

图 2

```
If( MbWeight >= MbWeight_MAX
 && B1k8× 8DEV_8×4 < μ × DEV_8×8)
{
    Inter8×4较Inter8×8先估计
}
else
{
    Inter8×8较Inter8×4先估计
}
```

(a)

```
iF( MbWeight >= MbWeight_MAX
 && B1k8× 8DEV_4×8 < μ × DEV_8×8)
{
    Inter4×8较Inter8×8先估计
}
else
{
    Inter8×8较Inter4×8先估计
}
```

(b)

```
if( B1k8× 8DEV_8×4 < B1k8× 8DEV_4×8)
{
    Inter8×4较Inter4×8先估计
}
else
{
    Inter4×8较Inter8×4先估计
}
```

(c)

图 3

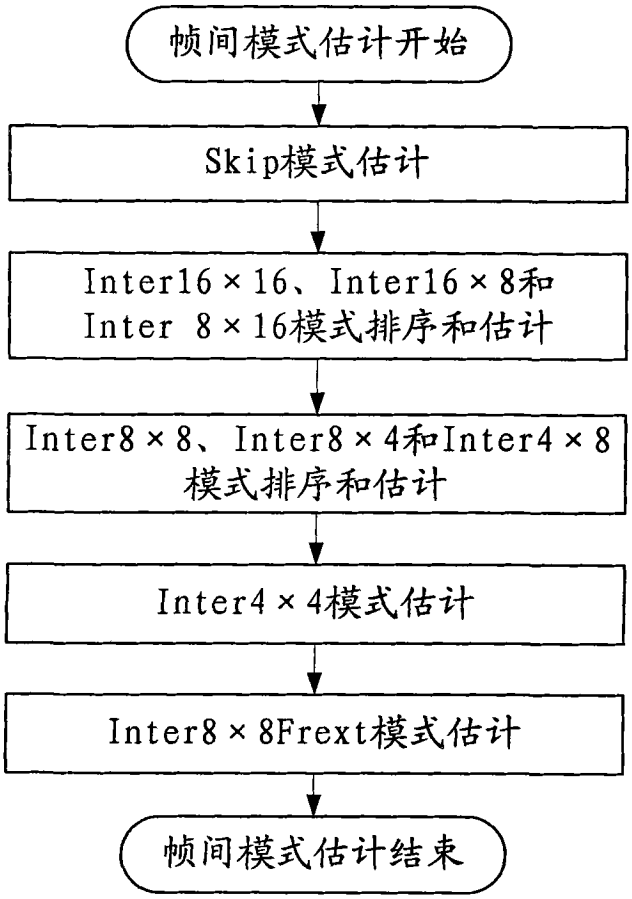

帧间模式估计开始
Skip模式估计
Inter16×16、Inter16×8和
Inter 8×16模式排序和估计
Inter8×8、Inter8×4和Inter4×8
模式排序和估计
Inter4×4模式估计
Inter8×8Frext模式估计
帧间模式估计结束

图 4

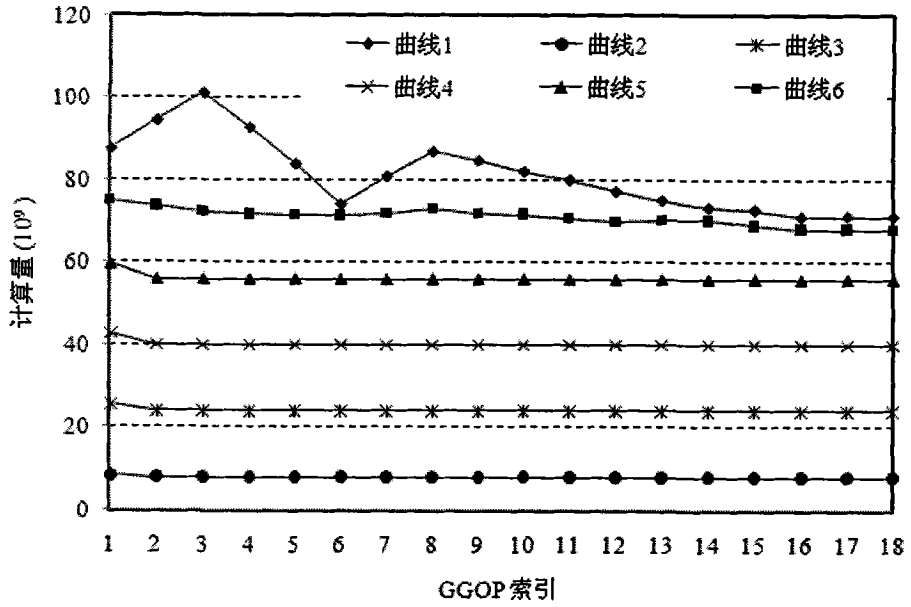

曲线1
曲线2
曲线3
曲线4
曲线5
曲线6
计算量(10^9)
120
100
80
60
40
20
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18
GGOP索引

图 5