



(12) 发明专利申请

(10) 申请公布号 CN 120051993 A

(43) 申请公布日 2025. 05. 27

(21) 申请号 202280100854.X

(22) 申请日 2022.10.13

(85) PCT国际申请进入国家阶段日
2025.04.07

(86) PCT国际申请的申请数据
PCT/CN2022/125219 2022.10.13

(87) PCT国际申请的公布数据
W02024/077572 ZH 2024.04.18

(71) 申请人 OPPO广东移动通信有限公司
地址 523860 广东省东莞市长安镇乌沙海
滨路18号

(72) 发明人 郝雪 霍俊彦 马彦卓 杨付正
李明

(74) 专利代理机构 北京派特恩知识产权代理有限公司 11270
专利代理师 马丽 徐川

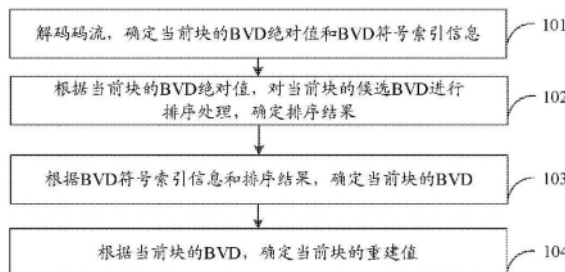
(51) Int.Cl.
H04N 19/52 (2006.01)
H04N 19/86 (2006.01)

(54) 发明名称

编解码方法、编码器、解码器以及存储介质

(57) 摘要

本申请实施例提供了一种编解码方法,在解码端,解码码流,确定当前块的BVD绝对值和BVD符号索引信息;根据当前块的BVD绝对值,对当前块的候选BVD进行排序处理,确定排序结果;根据BVD符号索引信息和排序结果,确定当前块的BVD;根据当前块的BVD,确定当前块的重建值。在编码端,根据当前块的BVD绝对值,对当前块的候选BVD进行排序处理,确定排序结果;根据当前块的BVD符号信息和排序结果,确定当前块的BVD符号索引信息;将BVD绝对值和BVD符号索引信息写入码流。



(12) 按照专利合作条约所公布的国际申请

(19) 世界知识产权组织
国际局

(43) 国际公布日
2024 年 4 月 18 日 (18.04.2024)



(10) 国际公布号
WO 2024/077572 A1

(51) 国际专利分类号:
H04N 19/52 (2014.01) **H04N 19/86** (2014.01)

(21) 国际申请号: PCT/CN2022/125219

(22) 国际申请日: 2022 年 10 月 13 日 (13.10.2022)

(25) 申请语言: 中文

(26) 公布语言: 中文

(71) 申请人: **OPPO 广东移动通信有限公司 (GUANGDONG OPPO MOBILE TELECOMMUNICATIONS CORP., LTD.)** [CN/CN]; 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。

(72) 发明人: 郝雪 (**HAO, Xue**); 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。
霍俊彦 (**HUO, Junyan**); 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。马

彦卓 (**MA, Yanzhuo**); 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。杨付正 (**YANG, Fuzheng**); 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。李明 (**LI, Ming**); 中国广东省东莞市长安镇乌沙海滨路 18 号, Guangdong 523860 (CN)。

(74) 代理人: 北京派特恩知识产权代理有限公司 (**CHINA PAT INTELLECTUAL PROPERTY OFFICE**); 中国北京市海淀区海淀南路 21 号中关村知识产权大厦 B 座 2 层, Beijing 100080 (CN)。

(81) 指定国(除另有指明, 要求每一种可提供的国家保护): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN,

(54) **Title:** ENCODING METHOD, DECODING METHOD, ENCODERS, DECODERS, AND STORAGE MEDIUM

(54) 发明名称: 编解码方法、编码器、解码器以及存储介质

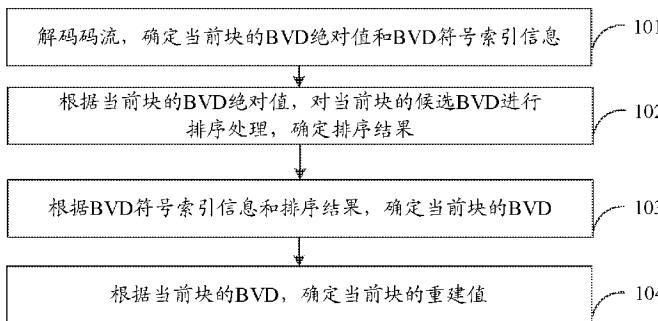


图 5

- 101 Decode a code stream to determine a BVD absolute value and BVD symbol index information of a current block
102 According to the BVD absolute value of the current block, sort candidate BVDs of the current block to determine a sorting result
103 According to the BVD symbol index information and the sorting result, determine a BVD of the current block
104 According to the BVD of the current block, determine a reconstruction value of the current block

(57) **Abstract:** Provided in the embodiments of the present application are an encoding method and a decoding method. The decoding method comprises: at a decoding end, decoding a code stream to determine a BVD absolute value and BVD symbol index information of a current block; according to the BVD absolute value of the current block, sorting candidate BVDs of the current block to determine a sorting result; according to the BVD symbol index information and the sorting result, determining a BVD of the current block; and, according to the BVD of the current block, determining a reconstruction value of the current block. The encoding method comprises: at an encoding end, sorting candidate BVDs of a current block according to a BVD absolute value of the current block to determine a sorting result; according to BVD symbol information and the sorting result of the current block, determining BVD symbol index information of the current block; and writing the BVD absolute value and the BVD symbol index information into a code stream.



MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW。

(84) 指定国(除另有指明, 要求每一种可提供的地区保护): ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), 欧亚 (AM, AZ, BY, KG, KZ, RU, TJ, TM), 欧洲 (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG)。

本国际公布:
— 包括国际检索报告(条约第21条(3))。

(57) 摘要: 本申请实施例提供了一种编解码方法, 在解码端, 解码码流, 确定当前块的BVD绝对值和BVD符号索引信息; 根据当前块的BVD绝对值, 对当前块的候选BVD进行排序处理, 确定排序结果; 根据BVD符号索引信息和排序结果, 确定当前块的BVD; 根据当前块的BVD, 确定当前块的重建值。在编码端, 根据当前块的BVD绝对值, 对当前块的候选BVD进行排序处理, 确定排序结果; 根据当前块的BVD符号信息和排序结果, 确定当前块的BVD符号索引信息; 将BVD绝对值和BVD符号索引信息写入码流。

编解码方法、编码器、解码器以及存储介质

技术领域

本申请实施例涉及图像处理技术领域，尤其涉及一种编解码方法、编码器、解码器以及存储介质。

背景技术

随着人们对视频显示质量要求的提高，高清和超高清视频等新视频应用形式应运而生。H.265/高效率视频编码 (High Efficiency Videocoding, HEVC) 已经无法满足视频应用迅速发展的需求，联合视频研究组 (Joint Video Exploration Team, JVET) 提出了下一代视频编码标准 H.266/多功能视频编码 (Versatile Videocoding, VVC)，其相应的测试模型为 VVC 的参考软件测试平台 (VVC Test Model, VTM)。

帧内块拷贝 (Intrablockcopy, IBC) 是 VVC 针对屏幕内容类型的视频序列的编码 (Screencontentcoding) 的扩展工具，它显著提高了屏幕内容序列的编码效率。然而，在 IBC 模式中，块向量差异 (Block Vector Difference, BVD) 的符号信息的传输方法往往比较复杂，在一定程度上增大了传输码率，进而降低 IBC 模式的编码性能。

发明内容

本申请实施例提供一种编解码方法、编码器、解码器以及存储介质，能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率，从而可以提高编码效率。

本申请实施例的技术方案可以如下实现：

第一方面，本申请实施例提供了一种解码方法，应用于解码器，所述方法包括：

解码码流，确定当前块的 BVD 绝对值和 BVD 符号索引信息；

根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；

根据所述 BVD 符号索引信息和所述排序结果，确定所述当前块的 BVD；

根据所述当前块的 BVD，确定所述当前块的重建值。

第二方面，本申请实施例提供了一种编码方法，应用于编码器，所述方法包括：

根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；

根据所述当前块的 BVD 符号信息和所述排序结果，确定所述当前块的 BVD 符号索引信息；

将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

第三方面，本申请实施例提供了一种编码器，所述编码器包括第一确定单元，编码单元；其中，

所述第一确定单元，配置为根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；

根据所述当前块的 BVD 符号信息和所述排序结果，确定所述当前块的 BVD 符号索引信息；

所述编码单元，配置为将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

第四方面，本申请实施例提供了一种编码器，所述编码器包括第一存储器和第一处理器；其中，

所述第一存储器，用于存储能够在所述第一处理器上运行的计算机程序；

所述第一处理器，用于在运行所述计算机程序时，执行如第二方面所述的方法。

第五方面，本申请实施例提供了一种解码器，所述解码器包括解码单元，第二确定单元；其中，

所述解码单元，配置为解码码流；

所述第二确定单元，配置为确定当前块的 BVD 绝对值和 BVD 符号索引信息；

根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；根据所述 BVD 符号索引信息和所述排序结果，确定所述当前块的 BVD；根据所述当前块的 BVD，确定所述当前块的重建值。

第六方面，本申请实施例提供了一种解码器，所述解码器包括第二存储器和第二处理器；其中，

所述第二存储器，用于存储能够在所述第二处理器上运行的计算机程序；

所述第二处理器，用于在运行所述计算机程序时，执行如第一方面所述的方法。

第七方面，本申请实施例提供了一种计算机可读存储介质，所述计算机可读存储介质存储有计算机程序，所述计算机程序被执行时实现如第一方面所述的方法、或者实现如第二方面所述的方法。

本申请实施例提供了一种编解码方法、编码器、解码器以及存储介质，在解码端，解码码流，确定当前块的 BVD 绝对值和 BVD 符号索引信息；根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果；根据 BVD 符号索引信息和排序结果，确定当前块的 BVD；根据当前块的 BVD，确定当前块的重建值。在编码端，根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果；根据当前块的 BVD 符号信息和排序结果，确定当前块的 BVD 符号索引信息；将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见，在本申请的实施例中，可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理，从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输，在解码时利用排序结果进行当前块的 BVD 符号信息的解析，同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理，能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率，从而可以提高编码效率。

附图说明

图 1 为 IBC 模式的预测示意图;
图 2 为相邻块的示意图;
图 3 为本申请实施例提供的一种视频编码系统的组成框图;
图 4 为本申请实施例提供的一种视频解码系统的组成框图;
图 5 为本申请实施例中的解码方法的示意图一;
图 6 为模板类型的示意图;
图 7 为当前编码块的 BVP 和 BVD 的示意图;
图 8 为模板运动补偿的示意图;
图 9 为 BVD 组合的示意图;
图 10 为本申请实施例中的解码方法的示意图二;;
图 11 为计算成本的示意图;
图 12 为本申请实施例中的编码方法的示意图一;
图 13 为本申请实施例中的编码方法的示意图二;
图 14 为编码器的组成结构示意图一;
图 15 为编码器的组成结构示意图二;
图 16 为解码器的组成结构示意图一;
图 17 为解码器的组成结构示意图二。

具体实施方式

为了能够更加详尽地了解本申请实施例的特点与技术内容,下面结合附图对本申请实施例的实现进行详细阐述,所附附图仅供参考说明之用,并非用来限定本申请实施例。

在以下的描述中,涉及到“一些实施例”,其描述了所有可能实施例的子集,但是可以理解,“一些实施例”可以是所有可能实施例的相同子集或不同子集,并且可以在不冲突的情况下相互结合。还需要指出,本申请实施例所涉及的术语“第一\第二\第三”仅是用于区别类似的对象,不代表针对对象的特定排序,可以理解地,“第一\第二\第三”在允许的情况下可以互换特定的顺序或先后次序,以使这里描述的本申请实施例能够以除了在这里图示或描述的以外的顺序实施。

在视频图像中,一般采用第一图像分量、第二图像分量和第三图像分量来表征编码块(Codingblock, CB);其中,这三个图像分量分别为一个亮度分量、一个蓝色色度分量和一个红色色度分量,具体地,亮度分量通常使用符号 Y 表示,蓝色色度分量通常使用符号 Cb 或者 U 表示,红色色度分量通常使用符号 Cr 或者 V 表示;这样,视频图像可以用 YCbCr 格式表示,也可以用 YUV 格式表示。

在本申请实施例中,第一图像分量可以为亮度分量,第二图像分量可以为蓝色色度分量,第三图像分量可以为红色色度分量,但是本申请实施例不作具体限定。

通用的视频编解码标准基于都采用基于块的混合编码框架。视频图像中的每一帧被分割成相同大小(比如 128×128, 64×64 等)的正方形的最大编码单元(Largestcoding Unit, LCU)或编码树单元(Coding Tree Unit, CTU),每个最大编码单元或编码树单元还可以根据规则划分成矩形的编码单元(Coding Unit, CU);而且编码单元可能还会划分成更小的预测单元(Prediction Unit, PU)、变换单元(Transform Unit, TU)等。

混合编码框架可以包括有预测(Prediction)、变换(Transform)、量化(Quantization)、熵编码(Entropyencoding)、环路滤波(Inloop Filter)等模块。其中,预测模块可以包括帧内预测(Intra Prediction)和帧间预测(Inter Prediction),帧间预测可以包括运动估计(Motion Estimation)和运动补偿(Motioncompensation)。由于视频图像的一个帧内相邻像素之间存在很强的相关性,在视频编解码技术中使用帧内预测方式能够消除相邻像素之间的空间冗余;但是由于视频图像中的相邻帧之间也存在着很强的相似性,在视频编解码技术中使用帧间预测方式消除相邻帧之间的时间冗余,从而能够提高编解码效率。

视频编解码器的基本流程如下:在编码端,将一帧图像划分成块,对当前块使用帧内预测或帧间预测产生当前块的预测块,当前块的原始块减去预测块得到残差块,对残差块进行变换、量化得到量化系数矩阵,对量化系数矩阵进行熵编码输出到码流中。在解码端,对当前块使用帧内预测或帧间预测产生当前块的预测块,另一方面解码码流得到量化系数矩阵,对量化系数矩阵进行反量化、反变换得到残差块,将预测块和残差块相加得到重建块。重建块组成重建图像,基于图像或基于块对重建图像进行环路滤波得到解码图像。编码端同样需要和解码端类似的操作获得解码图像。解码图像可以为后续的帧作为帧间预测的参考帧。编码端确定的块划分信息,预测、变换、量化、熵编码、环路滤波等模式信息或者参数信息如果有必要需要在输出到码流中。解码端通过解码及根据已有信息进行分析确定与编码端相同的块划分信息,预测、变换、量化、熵编码、环路滤波等模式信息或者参数信息,从而保证编码端获得的解码图像和解码端获得的解码图像相同。编码端获得的解码图像通常也叫做重建图像。在预测时可以将当前块划分成预测单元,在变换时可以将当前块划分成变换单元,预测单元和变换单元的划分可以不同。上述是基于块的混合编码框架下的视频编解码器的基本流程,随着技术的发展,该框架或流程的一些模块或步骤可能会被优化,本申请实施例适用于该基于块的混合编码框架下的视频编解码器的基本流程,但不限于该框架及流程。

当前块(Currentblock, CB)可以是当前编码单元(CU)或当前预测单元(PU)等。

帧内块拷贝(Intrablockcopy, IBC)是 VVC 针对屏幕内容类型的视频序列的编码(Screencontentcoding)的扩展工具,它显著提高了屏幕内容序列的编码效率。

IBC 是一种块级的编码模式, 和帧间技术类似, 编码端执行运动搜索, 即块匹配 (Block Matching, BM), 为每个 CU 找到其最佳的块向量 (Block Vector, BV), 也可以称为运动向量 (Motion Vector, MV), 块向量是从当前块指向参考块的向量。和帧间技术不同的地方在于, IBC 的最佳块向量是在当前 CU 所处帧 (即当前编码帧) 的重建区域搜索得到的, 而帧间的运动向量是通过当前编码帧在时间域上的相邻参考帧搜索得到的。

图 1 为 IBC 模式的预测示意图, 如图 1 所示, H.266/VVC 中 IBC 模式下获取当前编码块的重建像素的具体流程可以包括推导 BV、利用 BV 推导预测样本、推导残差样本和利用预测样本以及残差样本推导重建样本。

其中, 推导 BV 的具体流程如下:

(1) 亮度:

输入: 亮度位置 (xCb, yCb), 指定当前编码块的左上角样本相对于当前图片的左上角亮度样本, 一个变量 cbWidth, 指定亮度样本中当前编码块的宽度, 一个变量 cbHeight, 指定亮度样本中当前编码块的高度。

输出: 亮度的 bvL (Block Vector Luma)。

IBC 模式分为 IBCMERGE 和 IBCAMVP, 在推导 bvL 的时候均需要建立 IBC 块向量候选列表 bvCandList, 下面介绍 IBCMERGE 的列表的建立流程, IBCAMVP 的列表建立流程与 IBCMERGE 一致, 但两者的最大候选数量不一致。

步骤一: 当 IsGt4by4 等于 TRUE 时 (变量 IsGt4by4 为当亮度宽度乘以高度, 当 IsGt4by4 大于 16 时为 TRUE), 使用亮度编码块位置 (xCb, yCb)、亮度编码块宽度 cbWidth 和高度 cbHeight 调用解码规范所规定的来自相邻编码单元的空间块向量候选的推导过程作为输入, 输出是可用性标志 availableFlagA1、availableFlagB1 和块向量 bvA1 和 bvB1。其中, A1 和 B1 为相邻块。图 2 为相邻块的示意图, 如图 2 所示, A1 和 B1 所在的相邻块与当前编码块的相对位置分别为左下角和右上角。

步骤二: 当 IsGt4by4 等于 TRUE 时, 块向量候选列表 bvCandList 构造如下:

i=0

if (availableFlagA1)

 bvCandList[i++] = bvA1

if (availableFlagB1)

 bvCandList[i++] = bvB1

步骤三: 变量 numCurrCand 表示当前已经得到的候选数量, numCurrCand 推导过程如下:

若 IsGt4by4 等于 TRUE, numCurrCand 设置为等于 bvCandList 中的候选数; 否则 numCurrCand 设置为 0。

步骤四: 当 numCurrCand 小于 MaxNumIbcMergeCand (MERGE 模式下的最大候选数量) 且 NumHmvpIbcCand (表征 IBC 模式下的历史最优块向量 (Hmvp) 的最大候选数量) 大于 0 时, 以 bvCandList 和 numCurrCand 作为输入, 修改后的 bvCandList 和 numCurrCand 作为输出, 调用解码规范所规定的基于历史的 IBC 块向量候选的推导过程。

步骤五:

当 numCurrCand 小于 MaxNumIbcMergeCand 时, 以下适用直到 numCurrCand 等于 MaxNumIbcMergeCand:

 bvCandList[numCurrCand][0] (即 bv 的水平分量) 设置为等于 0。

 bvCandList[numCurrCand][1] (即 bv 的垂直分量) 设置为等于 0。

 numCurrCand 增加 1。

这样, 块向量候选列表 bvCandList 建立完成, 使用 general_merge_flag 表示是否为 IBCMERGE 模式, 候选的索引 bvIdx 采用如下方式推导:

 bvIdx = general_merge_flag[xCb][yCb] ? merge_idx[xCb][yCb] : mvp_l0_flag[xCb][yCb]

这样根据索引 bvIdx 以及块向量候选列表 bvCandList 就可以得到具体的 bvL:

 bvL[0] = bvCandList[bvIdx][0]

 bvL[1] = bvCandList[bvIdx][1]

对于 IBCAMVP 模式, 通过索引 bvIdx 以及块向量候选列表 bvCandList 就可以得到具体的 bvL 为预测 bvL, 真实 bvL 还需要加上块向量差异 (Block Vector Difference, BVD), 其具体流程如下:

步骤一: 得到 BVD 的水平和垂直分量, 其中 MvdL0 为前向的运动向量差异。

 bvd[0] = MvdL0[xCb][yCb][0]

 bvd[1] = MvdL0[xCb][yCb][1]

步骤二: 对上述获取的预测 bvL 进行舍入操作, 其中右移参数 AmvrShift 进行舍入, 左移参数 AmvrShift 用于提高分辨率。

 offset = (AmvrShift == 0) ? 0 : (((1 << (AmvrShift - 1)) - 1) bvL[0] = Sign(bvL[0]) × (((Abs(bvL[0]) + offset) >> AmvrShift) << AmvrShift)

 bvL[1] = Sign(bvL[1]) × (((Abs(bvL[1]) + offset) >> AmvrShift) << AmvrShift) 步骤三: 真实 bvL 的推导如下, 其范围需要控制在 -217 到 217 - 1 之间:

 u[0] = (bvL[0] + bvd[0] + 218) % 218

 bvL[0] = (u[0] >= 217) ? (u[0] - 218) : u[0]

 u[1] = (bvL[1] + bvd[1] + 218) % 218

 bvL[1] = (u[1] >= 217) ? (u[1] - 218) : u[1]

(2) 色度: 如果是双树划分, 色度不做 IBC; 如果是单树划分, 需要推导色度的 BV。

输入: 亮度的 bvL (1/16 像素精度)

输出: 色度的 bvC (Block Vectorchroma) (1/32 像素精度)

推导过程如下:

 bvC[0] = ((bvL[0] >> (3 + SubWidthC)) × 32)

 bvC[1] = ((bvL[1] >> (3 + SubHeightC)) × 32)

其中, 利用 BV 推导预测样本的具体流程如下:

输入: 亮度位置 (xCb, yCb), 指定当前编码块的左上角样本相对于当前图片的左上角亮度样本, 一个变量 cbWidth, 指定亮度样本中当前编码块的宽度, 一个变量 cbHeight, 指定亮度样本中当前编码块的高度, 块向量 bv, 一个变量 cIdx, 指定当前块的颜色分量索引。

输出: 预测样本的数组 predSamples。

推导过程如下:

当 cIdx 等于 0 即为亮度分量的时候, 对于 $x = xCb..xCb + cbWidth - 1$ 以及 $y = yCb..yCb + cbHeight - 1$:

$xVb = (x + (bv[0] >> 4)) \& (IbcBufWidthY - 1)$

$yVb = (y + (bv[1] >> 4)) \& (CtbSizeY - 1)$

$predSamples[x][y] = ibcVirBuf[0][xVb][yVb]$

IbcBufWidthY 为 IBC 存储的重建 buffer 的亮度像素的宽, CtbSizeY 为 CTU (Coding Tree Unit) 的大小, ibcVirBuf 为 IBC 存储的重建像素。

当 cIdx 不等于 0 即为色度分量的时候, 对于 $x = xCb / SubWidthC..xCb / SubWidthC + cbWidth / SubWidthC - 1$ 以及 $y = yCb / SubHeightC..yCb / SubHeightC + cbHeight / SubHeightC - 1$:

$xVb = (x + (bv[0] >> 5)) \& (IbcBufWidthC - 1)$

$yVb = (y + (bv[1] >> 5)) \& ((CtbSizeY / subHeightC) - 1)$

$predSamples[x][y] = ibcVirBuf[cIdx][xVb][yVb]$

变量 SubWidthC 和 SubHeightC 具体取决于通过 sps_chroma_format_idc 指定的色度格式采样结构, 具体对应关系如下表所示:

表 1

sps_chroma_format_idc	颜色采样格式	SubWidthC	SubHeightC
0	单色	1	1
1	4:2:0	2	2
2	4:2:2	2	1
3	4:4:4	1	1

其中, 在推导残差样本时, 可以调用解码规范所规定的帧间预测模式下编码块的残差信号的解码过程。在利用预测样本以及残差样本推导重建样本时, 可以调用解码规范所规定的指定颜色分量的图像重建过程。

进一步地, 在 IBC 的 BVD 的编码过程中, 对于 IBCAMVP 模式, 可以使用基础块向量预测 (Block Vector Prediction, BVP) 进行运动估计, 搜索最佳块运动向量, 搜索得到的最佳块运动向量和基础块向量的差值为块向量差异 BVD。

其中, BVD 的编码过程为: 对于 BVD 水平分量和垂直分量, 分别先编码 BVD 是否为零的标志: 如果其不为零则编码其绝对值, 然后再旁路编码 BVD 符号; 否则将不继续编码任何比特位。

进一步地, 对于 H.266/VVC 中 IBC 的 BVD 的解码过程, VVC 中 BVD 的解码方式同帧间的 MVD 的解码方式。MVD 具体的解码流程:

if (general_merge_flag[x0][y0])	
...	
else if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
mvd_coding (x0, y0, 0, 0)	
...	
}	

mvd_coding (x0, y0, refList, cpIdx) {	Descriptor
abs_mvd_greater0_flag[0]	ae (v)
abs_mvd_greater0_flag[1]	ae (v)
if (abs_mvd_greater0_flag[0])	
abs_mvd_greater1_flag[0]	ae (v)
if (abs_mvd_greater0_flag[1])	
abs_mvd_greater1_flag[1]	ae (v)
if (abs_mvd_greater0_flag[0]) {	
if (abs_mvd_greater1_flag[0])	
abs_mvd_minus2[0]	ae (v)
mvd_sign_flag[0]	ae (v)
}	
if (abs_mvd_greater0_flag[1]) {	

if (abs_mvd_greater1_flag[1])	
abs_mvd_minus2[1]	ae (v)
mvd_sign_flag[1]	ae (v)
}	
}	

其中, mvd_sign_flag[compIdx] 表示某个 (水平或垂直) 分量 compIdx 的运动矢量差异的符号: 如果 mvd_sign_flag[compIdx] 等于 0, 则对应的运动矢量分量差异具有正值; 否则 (mvd_sign_flag[compIdx] 等于 1), 对应的运动矢量分量差异为负值; 当 mvd_sign_flag[compIdx] 不存在时, 推断为等于 0。

compIdx=0..1 的运动矢量差异 IMvd[compIdx] 的推导如下, IMvd[compIdx] 的值应在 -217 到 217-1 的范围内:

$$\text{IMvd}[\text{compIdx}] = \text{abs_mvd_greater0_flag}[\text{compIdx}] \times (\text{abs_mvd_minus2}[\text{compIdx}] + 2) \times (1 - 2 \times \text{mvd_sign_flag}[\text{compIdx}])$$

IBC 模式下, refList 等于 0, 对于 compIdx=0..1 (0 表示水平分量, 1 表示垂直分量), MvdL0[x0][y0][compIdx] 设置为等于 IMvd[compIdx], 进而推导出 BVD 的水平 and 垂直分量:

bvd[0]=MvdL0[x0][y0][0]

bvd[1]=MvdL0[x0][y0][1]

表 2

Syntax element	binIdx					
	0	1	2	3	4	>= 5
mvd_sign_flag [0]	bypass	na	na	na	na	na
mvd_sign_flag [1]	bypass	na	na	na	na	na

进一步, Beyond VVC 中 IBC 的 BVD 的解码过程:

if (general_merge_flag[x0][y0])	
...	
else if (CuPredMode[chType][x0][y0] == MODE_IBC) {	
bvd_coding (x0, y0, 0, 0)	
...	
}	

bvd_coding (x0, y0, refList , cpIdx) {	Descriptor
abs_bvd_greater0_flag[0]	ae (v)
abs_bvd_greater0_flag[1]	ae (v)
if (abs_bvd_greater0_flag[0]) {	
abs_bvd_minus1[0]	ae (v)
bvd_sign_flag[0]	ae (v)
}	
if (abs_bvd_greater0_flag[1]) {	
abs_mvd_minus1[1]	ae (v)
bvd_sign_flag[1]	ae (v)
}	
}	

其中, bvd_sign_flag[compIdx] 表示某个 (水平或垂直) 分量 compIdx 的运动矢量差异的符号: 如果 bvd_sign_flag[compIdx] 等于 0, 则对应的运动矢量分量差异具有正值; 否则 (bvd_sign_flag[compIdx] 等于 1), 对应的运动矢量分量差异为负值; 当 bvd_sign_flag[compIdx] 不存在时, 推断为等于 0。

compIdx=0..1 的运动矢量差异 bvd[compIdx] 的推导如下:

$$\text{bvd}[\text{compIdx}] = \text{abs_bvd_greater0_flag}[\text{compIdx}] \times (\text{abs_bvd_minus1}[\text{compIdx}] + 1) \times (1 - 2 \times \text{bvd_sign_flag}[\text{compIdx}])$$

表 3

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvd_sign_flag[0]	bypass	na	na	na	na	na
bvd_sign_flag[1]	bypass	na	na	na	na	na

如上所示, 目前对 IBC 模式下的 BVD 的符号采用旁路编码, 而旁路编码是一种等概的编码方式, 无须对概率进行自适应更新, 而是采用 0 和 1 概率各占 1/2 的固定概率进行编码, 编码方式简单, 无法根据视频信息以及历史编码信息进行自适应更新, 1 个 BVD 符号信息对应码流的 1 个比特, 存在较大的码率冗余。

可见, 目前常见的 BVD 编码方案依然存在很大的改进空间。

参见图 3, 其示出了本申请实施例提供的一种视频编码系统的组成框图示例; 如图 3 所示, 该视频编码系统 10 包括变换与量化单元 101、帧内估计单元 102、帧内预测单元 103、运动补偿单元 104、运动估计单元 105、反变换与反量化单元 106、滤波器控制分析单元 107、滤波单元 108、编码单元 109 和解码图像缓存单元 110 等, 其中, 滤波单元 108 可

以实现去方块滤波及样本自适应缩进 (Sample Adaptive Offset, SAO) 滤波, 编码单元 109 可以实现头信息编码及基于上下文的自适应二进制算术编码 (Context-based Adaptive Binary Arithmetic Coding, CABAC)。针对输入的原始视频信号, 通过编码树块 (Coding Tree Unit, CTU) 的划分可以得到一个视频编码块, 然后对经过帧内或帧间预测后得到的残差像素信息通过变换与量化单元 101 对该视频编码块进行变换, 包括将残差信息从像素域变换到变换域, 并对所得的变换系数进行量化, 用以进一步减少比特率; 帧内估计单元 102 和帧内预测单元 103 是用于对该视频编码块进行帧内预测; 明确地说, 帧内估计单元 102 和帧内预测单元 103 用于确定待用以编码该视频编码块的帧内预测模式; 运动补偿单元 104 和运动估计单元 105 用于执行所接收的视频编码块相对于一或多个参考帧中的一或多个块的帧间预测编码以提供时间预测信息; 由运动估计单元 105 执行的运动估计为产生运动向量的过程, 所述运动向量可以估计该视频编码块的运动, 然后由运动补偿单元 104 基于由运动估计单元 105 所确定的运动向量执行运动补偿; 在确定帧内预测模式之后, 帧内预测单元 103 还用于将所选择的帧内预测数据提供到编码单元 109, 而且运动估计单元 105 将所计算确定的运动向量数据也发送到编码单元 109; 此外, 反变换与反量化单元 106 是用于该视频编码块的重构建, 在像素域中重构建残差块, 该重构建残差块通过滤波器控制分析单元 107 和滤波单元 108 去除方块效应伪影, 然后将该重构残差块添加到解码图像缓存单元 110 的帧中的一个预测性块, 用以产生经重构建的视频编码块; 编码单元 109 是用于编码各种编码参数及量化后的变换系数, 在基于 CABAC 的编码算法中, 上下文内容可基于相邻编码块, 可用于编码指示所确定的帧内预测模式的信息, 输出该视频信号的码流; 而解码图像缓存单元 110 是用于存放重构建的视频编码块, 用于预测参考。随着视频图像编码的进行, 会不断生成新的重构建的视频编码块, 这些重构建的视频编码块都会被存放在解码图像缓存单元 110 中。

参见图 4, 其示出了本申请实施例提供的一种视频解码系统的组成框图示例; 如图 4 所示, 该视频解码系统 20 包括解码单元 201、反变换与反量化单元 202、帧内预测单元 203、运动补偿单元 204、滤波单元 205 和解码图像缓存单元 206 等, 其中, 解码单元 201 可以实现头信息解码以及 CABAC 解码, 滤波单元 205 可以实现去方块滤波以及 SAO 滤波。输入的视频信号经过图 4 的编码处理之后, 输出该视频信号的码流; 该码流输入视频解码系统 20 中, 首先经过解码单元 201, 用于得到解码后的变换系数; 针对该变换系数通过反变换与反量化单元 202 进行处理, 以便在像素域中产生残差块; 帧内预测单元 203 可用于基于所确定的帧内预测模式和来自当前帧或图片的先前经解码块的数据而产生当前视频解码块的预测数据; 运动补偿单元 204 是通过剖析运动向量和其他关联语法元素来确定用于视频解码块的预测信息, 并使用该预测信息以产生正被解码的视频解码块的预测性块; 通过对来自反变换与反量化单元 202 的残差块与由帧内预测单元 203 或运动补偿单元 204 产生的对应预测性块进行求和, 而形成解码的视频块; 该解码的视频信号通过滤波单元 205 以便去除方块效应伪影, 可以改善视频质量; 然后将经解码的视频块存储于解码图像缓存单元 206 中, 解码图像缓存单元 206 存储用于后续帧内预测或运动补偿的参考图像, 同时也用于视频信号的输出, 即得到了所恢复的原始视频信号。

本申请实施例中的编码方法, 可以应用在如图 3 所示的帧内预测单元 103 和能够实现 CABAC 编码算法的编码单元 109 部分。另外, 本申请实施例中的解码方法, 还可以应用在如图 4 所示的帧内预测单元 203 和能够实现 CABAC 解码算法的解码单元 201。也就是说, 本申请实施例中的编解码方法, 既可以应用于视频编码系统, 也可以应用于视频解码系统, 甚至还可以同时应用于视频编码系统和视频解码系统, 但是本申请实施例不作具体限定。还需要说明的是, 当该编解码方法应用于视频编码系统时, “当前块”具体是指帧内预测中的当前编码块; 当该编解码方法应用于视频解码系统时, “当前块”具体是指帧内预测中的当前解码块。

下面将结合本申请实施例中的附图, 对本申请实施例中的技术方案进行清楚、完整地描述。

本申请的实施例提出了一种解码方法, 该解码方法可以应用于解码器, 图 5 为本申请实施例中的解码方法的示意图一, 如图 5 所示, 解码器进行解码处理的方法可以包括以下步骤:

步骤 101、解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息。

在本申请的实施例中, 解码器通过解码码流, 可以确定出当前块的 BVD 绝对值和 BVD 符号索引信息。

需要说明的是, 在本申请的实施例中, 解码器可以直接从码流中解析出真实的当前块的 BVD 绝对值。即解码器解码码流, 可以确定当前块的 BVD 绝对值。

进一步地, 在本申请的实施例中, 解码器解码码流, 还可以确定出当前块的 BVD 符号索引信息, 即解析获得当前块的符号预测索引。

可以理解的是, 在本申请的实施例中, BVD 绝对值包括第一分量绝对值 absBvdX 和第二分量绝对值 absBvdY ; 即 BVD 绝对值包括水平分量绝对值和垂直分量绝对值。相应的, 当前块的 BVD 可以包括第一分量 candBvdX 和第二分量 candBvdY , 同时, 当前块的候选 BVD 也包括第一分量 candBvdX 和第二分量 candBvdY , 即 BVD 可以包括水平分量和垂直分量。

进一步地, 在本申请的实施例中, 如果 absBvdX 和 absBvdY 均不为 0, 那么可以确定 BVD 符号索引信息的比特位数为 2; 如果 absBvdX 或 absBvdY 为 0, 那么可以确定 BVD 符号索引信息的比特位数为 1。

需要说明的是, 在本申请的实施例中, 除了例如上述的定长编码方法以外, 还可以采用变长编码方法对 BVD 符号索引信息进行编码处理。例如采用截断一元码编码 BVD 符号索引信息, 既可以使用使用截断二进制码 (Truncated binarization) 进行二值化处理, 此时 BVD 符号索引信息的比特位数不一定为 2。

也就是说, 在本申请的实施例中, 如果采用截断一元码编码 BVD 符号索引信息 (这里, 该 BVD 符号索引信息可以为候选 BVD 列表中的索引值), 则不同候选 BVD 对应的符号索引信息的比特位数不完全相同, 即为变长编码。

需要说明的是, 在本申请的实施例中, 可以根据解码过程中的所解析出来的 BVD 绝对值来判断 BVD 符号索引信息的二进制数的比特位的数目。其中, 若 BVD 的水平分量和垂直分量的绝对值都不为 0 时, 则 BVD 符号索引信息的二进制数的比特位的数目为 2; 若 BVD 的水平分量为 0 并且垂直分量的绝对值不为 0 时, 则 BVD 符号索引信息的二进制数的比特位的数目为 1; 若 BVD 的水平分量不为 0 并且垂直分量的绝对值为 0 时, 则 BVD 符号索引信息的二进制数的比特位的数目为 1。

然后, 解码器可以使用 CABAC (上下文自适应的二进制算术编码) 或旁路对应的解码过程将 BVD 符号索引信息对应的二进制数可以从码流中解析出来。

针对 BVD 符号索引信息所使用的 CABAC 的上下文模型, 可以使用单个概率模型, 或者多个概率模型, 可以包括

但不限于根据 BVD 的水平分量或垂直分量的绝对值大小、BVD 的水平分量与垂直分量的和或差值等进行区分不同的概率模型。

例如，使用 BVD 的水平分量或垂直分量的绝对值大小对 BVD 的 BVD 符号索引信息的比特位使用不同的概率模型进行分类：

(1) 假设 BVD 的水平或垂直分量的绝对值分类的一个阈值设为 THR_0 ，共有两个概率模型，则当 BVD 符号索引信息为两个比特位时，此二进制数的第 0 个比特位根据 BVD 的水平分量的绝对值与 THR_0 的大小关系进行区分，即当 BVD 的水平分量的绝对值小于等于 THR_0 时，使用第一个概率模型，当 BVD 的水平分量的绝对值大于 THR_0 时，使用第二个概率模型；然后，上述二进制数的第 1 个比特位根据 BVD 的垂直分量的绝对值与 THR_0 的大小关系进行区分，即当 BVD 的垂直分量的绝对值小于等于 THR_0 时，使用第一个概率模型，当 BVD 的垂直分量的绝对值大于 THR_0 时，使用第二个概率模型。

(2) 假设 BVD 的水平分量的绝对值分类的一个阈值设为 THR_0 ，垂直分量的绝对值分类的一个阈值设为 THR_1 ，共有四个概率模型，则当 BVD 符号索引信息为两个比特位时，此二进制数的第 0 个比特位根据 BVD 的水平分量的绝对值与 THR_0 的大小关系进行区分，即当 BVD 的水平分量的绝对值小于等于 THR_0 时，使用第一个概率模型，当 BVD 的水平分量的绝对值大于 THR_0 时，使用第二个概率模型；然后，上述二进制数的第 1 个比特位根据 BVD 的垂直分量的绝对值与 THR_1 的大小关系进行区分，即当 BVD 的垂直分量的绝对值小于等于 THR_1 时，使用第三个概率模型，当 BVD 的垂直分量的绝对值大于 THR_1 时，使用第四个概率模型。

进一步地，在本申请的实施例中，在确定出当前块的 BVD 绝对值之后，可以利用 BVD 绝对值确定 BVD 的组合列表，其中，BVD 的组合列表包括当前块的任意数量的候选 BVD。即可以利用 BVD 绝对值确定当前块的候选 BVD，也可以理解为利用 BVD 绝对值确定当前块的候选 BVD 的初始集合。

需要说明的是，在本申请的实施例中，可以通过在 BVD 的可能符号和 BVD 绝对值之间创建 BVD 的组合列表（候选 BVD）。其中，可以包括仅水平 BVD 存在、仅垂直 BVD 存在以及二者都存在的情况。

进一步地，在本申请的实施例中，在创建 BVD 的组合列表（候选 BVD）时，主要包括对 BVD 的可能符号进行排列组合、对可能的水平符号和垂直符号分别乘以 BVD 的水平绝对值分量和垂直绝对值分量。

示例性的，在本申请的实施例中，如果仅垂直 BVD 存在，即如果 $absBvdX$ 等于 0，那么可以根据 $absBvdY$ 确定候选 BVD。

例如，BVD 分为水平分量和垂直分量，当水平分量为零的时候，只对垂直分量进行编码，所以只对垂直分量的符号构建组合列表，具体为以下两种情况：

第一种情况：水平分量都为正号，只是一个占位，垂直分量先为正号，后为负号，构建一个符号列表，然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值，进而得到 BVD 的组合列表。

```
{
    {+1, +1},
    {+1, -1},
};
{
    { 0, +BVDVER },
    { 0, -BVDVER },
}。
```

第二种情况：水平分量都为正号，只是一个占位，垂直分量先为负号，后为正号，构建一个符号列表，然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值，进而得到 BVD 的组合列表。

```
{
    {+1, -1},
    {+1, +1},
};
{
    { 0, -BVDVER },
    { 0, +BVDVER },
}。
```

示例性的，在本申请的实施例中，如果仅水平 BVD 存在，即如果 $absBvdY$ 等于 0，那么可以根据 $absBvdX$ 确定候选 BVD。

例如，BVD 分为水平分量和垂直分量，当垂直分量为零的时候，只对水平分量进行编码，所以只对水平分量的符号构建组合列表，具体为以下两种情况：

第一种情况：垂直分量都为正号，只是一个占位，水平分量先为正号，后为负号，构建一个符号列表，然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值，进而得到 BVD 的组合列表。

```
{
    {+1, +1},
    {-1, +1},
};
{
    {+BVDHOR, 0},
    {-BVDHOR, 0},
}。
```

};

第二种情况：垂直分量都为正号，只是一个占位，水平分量先为负号，后为正号，构建一个符号列表，然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值，进而得到 BVD 的组合列表。

{

{-1, +1 },

{+1, +1 },

};

{

{-BVD_{HOR}, 0 },

{+BVD_{HOR}, 0 },

};

示例性的，在本申请的实施例中，如果水平 BVD 和垂直 BVD 均存在，即如果 absBvdX 和 absBvdY 均不等于 0，那么可以根据 absBvdX 和 absBvdY 确定候选 BVD。

例如，BVD 分为水平分量和垂直分量，当两分量均不为零的时候，二者均需要进行编码，所以要对水平和垂直分量一起构建 BVD 列表，具体分为 24 种情况，即{+1, +1}、{+1, -1}、{-1, +1}和{-1, -1}这 4 种组合下的任意排列，下述举例说明：

列表第 0 位为水平和垂直分量均为正号，列表第 1 位为水平分量为正号垂直分量为负号，列表第 2 位为水平分量为负号垂直分量为正号，列表第 3 位为水平分量和垂直分量均为负号，构建一个符号列表，然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值，进而得到 BVD 的组合列表。

{

{+1, +1 },

{+1, -1 },

{-1, +1 },

{-1, -1 },

};

{

{+BVD_{HOR}, +BVD_{VER} },

{+BVD_{HOR}, -BVD_{VER} },

{-BVD_{HOR}, +BVD_{VER} },

{-BVD_{HOR}, -BVD_{VER} },

};

步骤 102、根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果。

在本申请的实施例中，在确定当前块的 BVD 绝对值和 BVD 符号索引信息之后，可以进一步根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果。

进一步地，在本申请的实施例中，在根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果时，可以先根据 BVD 绝对值，确定当前块的一个或多个候选 BVD 对应的第一代价值；然后可以根据第一代价值，对候选 BVD 进行排序，确定排序结果。

可以理解的是，在本申请的实施例中，一个或多个候选 BVD 可以包括 candBvdX 分别设置为等于-absBvdX 或 absBvdX、candBvdY 分别设置为等于-absBvdY 或 absBvdY 的 BVD。

需要说明的是，在本申请的实施例中，在根据当前块的 BVD 绝对值，确定当前块的一个或多个候选 BVD 对应的第一代价值时，可以先根据候选 BVD 确定的第一匹配模板；然后根据预设的误差准则，计算当前块的第一模板与第一匹配模板之间的匹配误差，以确定候选 BVD 对应的第一代价值。

示例性的，在本申请的实施例中，在进行成本计算时，即确定第一代价值时，对于计算模板区域的成本的成本函数有多种选择，即预设的误差准则可以有多种选择。例如，可以选择绝对误差和（SAD）、变换绝对误差和（SATD）、差值平方和（SSE）、平均绝对差值（MAD）、平均绝对误差（MAE）、平均平方误差（MSE）、率失真函数（RDO）等评价准则。下述内容所提到的某种评价准则均可从上述准则中选择其一，以 SAD 的评价准则为例，计算公式如下：

$$SAD = \sum_{i=0}^{predTempSizeW-1} \sum_{j=0}^{predTempSizeH-1} |predTemp[i][j] - recTempC[i][j]| \quad (1)$$

其中，predTempSizeW 为模板的宽度，predTempSizeH 为模板的高度，predTemp[i][j] 为 BV 处模板的像素点，recTempC[i][j] 为当前块模板的像素点。

也就是说，在本申请的实施例中，预设的误差准则可以包括绝对误差和 SAD、变换绝对误差和 SATD、差值平方和 SSE、平均绝对差值 MAD、平均绝对误差 MAE、平均平方误差 MSE、率失真函数 RDO 中的任意一种。

可以理解的是，在本申请的实施例中，第一模板包括当前块的相邻已解码区域中的一个或多个样值。

示例性的，在本申请的实施例中，在进行模板（当前块的第一模板）的选择时，可以根据当前块的邻近区域的像素可用性，判断模板位置的像素是否可用，包括重建亮度信息。图 6 为模板类型的示意图，如图 6 所示，根据模板与当前块的相对位置关系，可将模板分类为上模板、左模板、右上模板、左下模板、左上模板等模板类型。其中，不同编码块的不同类型的模板尺寸可以固定相同，也可以不同。

例如，模板尺寸对当前任意编码块（当前块）选择相同的模板尺寸，如下公式说明了一种模板尺寸的设置条件，其中，nTbW 和 nTbH 分别为当前亮度编码块（当前块）的宽度和高度，iTempW 和 iTempH 分别为采用的模板的宽度和高度：

$$\text{上模板: } \begin{cases} iTempW = nTbW \\ iTempH = 1 \end{cases} \quad (2)$$

$$\text{左模板: } \begin{cases} iTempW = 1 \\ iTempH = nTbH \end{cases} \quad (3)$$

例如, 可以根据当前编码块的不同尺寸选择不同的模板尺寸, 如下公式说明了一种模板的尺寸设置条件, $nTbW$ 和 $nTbH$ 分别为当前亮度编码块的宽度和高度, $iTempW$ 和 $iTempH$ 分别为采用的模板的宽度和高度:

$$\text{上模板: } \begin{cases} iTempW = nTbW, & \min(nTbW, nTbH) \leq 4 \\ iTempH = 2 \\ iTempW = nTbW, & \min(nTbW, nTbH) > 4 \\ iTempH = 4 \end{cases} \quad (4)$$

$$\text{左模板: } \begin{cases} iTempW = 2, & \min(nTbW, nTbH) \leq 4 \\ iTempH = nTbH \\ iTempW = 4, & \min(nTbW, nTbH) > 4 \\ iTempH = nTbH \end{cases} \quad (5)$$

$$\text{Min}(x, y) = \begin{cases} x, & x \leq y \\ y, & x > y \end{cases}$$

例如, 也可以根据当前亮度编码块内的像素数选择不同的模板尺寸, 如下公式说明了一种模板的尺寸设置条件, $nTbW$ 和 $nTbH$ 分别为当前亮度编码块的宽度和高度, $nTbW \times nTbH$ 分别为当前亮度编码块的像素数, $iTempW$ 和 $iTempH$ 分别为采用的模板的宽度和高度:

$$\text{上模板: } \begin{cases} iTempW = nTbW, & nTbW \times nTbH \leq 16 \\ iTempH = 2 \\ iTempW = nTbW, & nTbW \times nTbH > 16 \\ iTempH = 4 \end{cases} \quad (6)$$

$$\text{左模板: } \begin{cases} iTempW = 2, & nTbW \times nTbH \leq 16 \\ iTempH = nTbH \\ iTempW = 4, & nTbW \times nTbH > 16 \\ iTempH = nTbH \end{cases} \quad (7)$$

需要说明的是, 在本申请的实施例中, 在根据候选 BVD 确定的第一匹配模板时, 可以先确定当前块的块向量预测值 BVP; 然后根据 BVP 和候选 BVD 确定当前块的候选 BV; 其中, 候选 BV 用于指示第一匹配模板的位置; 接着, 可以根据候选 BV 确定第一匹配模板。

示例性的, 在本申请的实施例中, 假设 BVP 包括第三分量 currBvpX 和第四分量 currBvpY , 那么在根据 BVP 和候选 BVD 确定当前块的候选 BV 时, 可以选择将候选 BV 的二维矢量设置为 $(\text{currBvpX} + \text{candBvdX}, \text{currBvpY} + \text{candBvdY})$, 即根据候选 BVD 的第一分量 candBvdX 、第二分量 candBvdY , 以及 BVP 的第三分量 currBvpX 、第四分量 currBvpY 进行 BV 的二维矢量的确定。

进一步地, 在本申请的实施例中, 在根据候选 BV 确定第一匹配模板时, 可以根据当前点的位置和候选 BV 的二维矢量确定第一匹配模板。

示例性的, 在本申请的实施例中, 可以选择以当前点的位置为起点, 将候选 BV 的二维矢量所指示的、与第一模板具有相同形状和包含相同数量样值的区域, 确定为第一匹配模板。

在进行当前块的 BV 预测时, 可以利用当前块的 BVP 以及 BVD 的组合列表, 产生新的 BV。图 7 为当前编码块的 BVP 和 BVD 的示意图, 图 8 为模板运动补偿的示意图, 如图 7 和图 8 所示, 利用当前块判断这个 BV 是否可用 (可用的条件为该 BV 指向的参考块已经重建, 且未超出 IBC 所设定的搜索范围以及图像边界等条件), 如果当前 BV 是可用的, 然后就会利用当前块的模板和新的 BV 进行运动补偿, 得到对应 BV 处的模板。图 9 为 BVD 组合的示意图, 如图 9 所示, BVD 水平和垂直分量共产生 4 种组合的 BVD, 分别进行模板处的运动补偿。计算成本时的模板的使用情况存在以下几种情况:

第一种: 当前块的上模板和左模板均存在, 且对应 BV 处的上模板和左模板均存在, 此时上模板和左模板均可用。

第二种: 当前块的上模板和左模板均存在, 对应 BV 处仅上模板存在, 此时存在两种方法: 方法一: 对应 BV 处左模板不存在, 直接不用, 即仅用上模板计算。方法二: 对应 BV 处左模板不存在, 采用参考块内部最左侧的模板宽度列代替左模板, 即采用上模板和左模板计算。

第三种: 当前块的上模板和左模板均存在, 对应 BV 处仅左模板存在, 此时存在两种方法: 方法一: 对应 BV 处上模板不存在, 直接不用, 即仅用左模板计算。方法二: 对应 BV 处上模板不存在, 采用参考块内部最上侧的模板高度行代替上模板, 即采用上模板和左模板计算。

第四种: 当前块仅上模板存在, 此时存在三种方法: 方法一: 对应 BV 处上模板不存在, 直接不用。方法二: 对应 BV 处上模板不存在, 采用参考块内部最上侧的模板高度行代替上模板。方法三: 对应 BV 处上模板存在, 直接用。

第五种: 当前块仅左模板存在, 此时存在三种方法: 方法一: 对应 BV 处左模板不存在, 直接不用。方法二: 对应 BV 处左模板不存在, 采用参考块内部最左侧的模板宽度列代替左模板。方法三: 对应 BV 处左模板存在, 直接用。

第六种: 当前块上模板和左模板均不存在, 忽略此种情况的 BVD 组合情况。

以上六种情况下, 当 BV 指向的参考块不可用时, 均忽略此种情况下的 BVD 组合情况。当忽略某种情况下的 BVD 组合情况时, 将此种情况下的成本置为成本评价准则的最大值。

进一步地, 在本申请的实施例中, 如果上述两个模板均能获取到, 此时对这个模板进行处理, 包括但不限于以下方法: 将两者进行加权融合, 计算方式为:

$$\text{cost} = w_0 \times \text{above} + w_1 \times \text{left}$$

其中, 加权融合的标准可以选择固定加权系数或者根据各自模板的重建亮度与预测亮度的像素值之差的绝对值之和进行分配加权系数等, 后者的计算加权系数的公式如下, SAD_A 为上模板的平均绝对差值, SAD_L 为左模板的平均绝对差值:

$$\begin{cases} w_0 = \frac{SAD_L}{SAD_A + SAD_L} \\ w_1 = \frac{SAD_A}{SAD_A + SAD_L} \end{cases} \quad (8)$$

由此可见,在本申请的实施例中,可以根据当前块的 BVP 和 BVD 的组合列表(候选 BVD)进行模板预测,计算出 BVD 组合列表中的每个 BVD 组合(候选 BVD)的成本(第一代价值),即计算模板区域的成本,可以对 BVD 组合列表中的各种组合利用模板进行成本的计算,具体为利用当前块模板位置处的重建亮度像素和当前块的 BVP 及 BVD 的组合得出的 BV 进行模板预测得到的位置处的重建亮度像素计算成本。

需要说明的是,在本申请的实施例中,在计算成本(第一代价值)时,不仅可以采用亮度,还可以包含色度即 Cb 和 Cr 分量。本申请不进行具体限定。

进一步地,在本申请的实施例中,在计算获得每一个候选 BVD 的第一代价值之后,便可以利用第一代价值对候选 BVD 进行排序处理,从而可以获得排序结果。其中,排序结果可以为排序后的当前块的候选 BVD 列表,也可以为通过排序确定出第一代价值最小的候选 BVD 的候选 BVD 集合。本申请不进行具体限定。

需要说明的是,在本申请的实施例中,如果排序结果为当前块的候选 BVD 列表,那么相应的,执行排序的过程可以为根据第一代价值,对候选 BVD 进行排序,确定当前块的候选 BVD 列表,其中,候选 BVD 列表至少包括一个候选 BVD。

进一步地,在本申请的实施例中,在根据第一代价值,对候选 BVD 进行排序,确定当前块的候选 BVD 列表时,可以选择根据预设的误差准则,按第一代价值指示的匹配误差从小到大的顺序,对一个或多个候选 BVD 进行排序,确定候选 BVD 列表。

可以理解的是,在本申请的实施例中,排序结果可以为排列为升(降)序的新列表 OrderedList(当前块的候选 BVD 列表)。例如,采用冒泡排序、选择排序、插入排序、希尔排序、归并排序、快速排序、基数排序、堆排序、计数排序、桶排序等排序方法。

进一步地,在本申请的实施例中,排序结果还可以为特定顺序的新列表 OrderedList(当前块的候选 BVD 列表)。

示例性的,在本申请的实施例中,在根据第一代价值,对候选 BVD 进行排序,确定当前块的候选 BVD 列表时,如果 absBvdX 和 absBvdY 均不等于 0,将候选 BVD 分别存储至第一数组和第二数组;然后基于第一代价值,确定第一数组中的第一候选 BVD 和第二候选 BVD,以及第二数组中的第三候选 BVD 和第四候选 BVD;最后便可以基于第一代价值,对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理,确定候选 BVD 列表。

如果 BVD 水平分量和垂直分量均存在,即 absBvdX 和 absBvdY 均不等于 0,假设该组合列表(候选 BVD)为 $\{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, +BVD_{VER})\}$,其对应的成本列表为 $\{Cost(+BVD_{HOR}, +BVD_{VER}), Cost(+BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, +BVD_{VER})\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp, posiGrp 存储在同一分量(水平)上符号为正号的两个 BVD 组合, negaGrp 存储在同一分量(水平)上符号为负号的两个 BVD 组合。

假设 $posiGrp = \{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER})\}$, $negagr = \{(-BVD_{HOR}, +BVD_{VER}), (-BVD_{HOR}, -BVD_{VER})\}$ 。posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较:比较 $Cost(+BVD_{HOR}, +BVD_{VER})$ 和 $Cost(+BVD_{HOR}, -BVD_{VER})$ 的大小,将 posiGrp[0]置为两者较小的 BVD 组合,将 posiGrp[1]置为两者较大的 BVD 组合;

第二次比较:比较 $Cost(-BVD_{HOR}, +BVD_{VER})$ 和 $Cost(-BVD_{HOR}, -BVD_{VER})$ 的大小,将 negaGrp[0]置为两者较小的 BVD 组合,将 negaGrp[1]置为两者较大的 BVD 组合;

第三次比较:比较 posiGrp[0]和 negaGrp[0]对应的 Cost 的大小。

需要说明的是,在本申请的实施例中,在基于第一代价值,对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理,确定候选 BVD 列表时,一种可用的方式如下:如果第一候选 BVD 的第一代价值小于等于第三候选 BVD 的第一代价值,将第一候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD,将第三候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD,将第二候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD,将第四候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。如果第一候选 BVD 的第一代价值大于第三候选 BVD 的第一代价值,将第三候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD,将第一候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD,将第四候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD,将第二候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。

相应的,最后的排序过程可以为:若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negaGrp[0]$ 对应的 Cost,将 posiGrp[0]排序在候选 BVD 列表 OrderedList 的第 0 位,将 negaGrp[0] 排序在候选 BVD 列表 OrderedList 的第 1 位,将 posiGrp[1] 排序在候选 BVD 列表 OrderedList 的第 2 位,将 negaGrp[1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, $posiGrp[0]$ 对应的 $Cost > negaGrp[0]$ 对应的 Cost,将 negaGrp[0] 排序在候选 BVD 列表 OrderedList 的第 0 位,将 posiGrp[0] 排序在候选 BVD 列表 OrderedList 的第 1 位,将 negaGrp[1] 排序在候选 BVD 列表 OrderedList 的第 2 位,将 posiGrp[1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

其中,当前块的候选 BVD 列表的第 0 位的候选 BVD 即为候选 BVD 列表中的第一个候选 BVD,第 1 位的候选 BVD 即为候选 BVD 列表中的第二个候选 BVD,第 2 位的候选 BVD 即为候选 BVD 列表中的第三个候选 BVD,第 3 位的候选 BVD 即为候选 BVD 列表中的第四个候选 BVD。

需要说明的是,在本申请的实施例中,在基于第一代价值,对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理,确定候选 BVD 列表时,另一种可用的方式如下:如果第一候选 BVD 的第一代价值小于等于第三候选 BVD 的第一代价值,将第一候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD,将第二候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD,将第三候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD,将第四候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。如果第一候选 BVD 的第一代价值大于第三候选 BVD 的第一代价值,将第

三候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将第四候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD, 将第一候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD, 将第二候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。

相应的, 最后的排序过程可以为: 若在第三次比较中, $\text{posiGrp}[0]$ 对应的 $\text{Cost} \leq \text{negaGrp}[0]$ 对应的 Cost , 将 $\text{posiGrp}[0]$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $\text{negaGrp}[0]$ 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 $\text{posiGrp}[1]$ 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 $\text{negaGrp}[1]$ 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, $\text{posiGrp}[0]$ 对应的 $\text{Cost} > \text{negaGrp}[0]$ 对应的 Cost , 将 $\text{negaGrp}[0]$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $\text{posiGrp}[0]$ 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 $\text{negaGrp}[1]$ 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 $\text{posiGrp}[1]$ 排序在候选 BVD 列表 OrderedList 的第 3 位。

示例性的, 在本申请的实施例, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 列表时, 如果 absBvdX 等于 0 或者 absBvdY 等于 0, 将候选 BVD 中的第一代价值最小的一个候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将候选 BVD 中的另一个候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD。

如果仅 BVD 水平分量或者垂直分量存在, 即 absBvdX 或 absBvdY 等于 0, 那么可以直接按照第一代价值的大小对两个候选 BVD 进行排序处理, 以生成候选 BVD 列表。

对于仅 BVD 垂直分量存在的情况, 假设该组合列表为 $\{(0, +\text{BVD}_{\text{VER}}), (0, -\text{BVD}_{\text{VER}})\}$, 其对应的成本列表为 $\{\text{Cost}(0, +\text{BVD}_{\text{VER}}), \text{Cost}(0, -\text{BVD}_{\text{VER}})\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为: 比较 $\text{Cost}(0, +\text{BVD}_{\text{VER}})$ 和 $\text{Cost}(0, -\text{BVD}_{\text{VER}})$ 的大小:

若 $\text{Cost}(0, +\text{BVD}_{\text{VER}}) \leq \text{Cost}(0, -\text{BVD}_{\text{VER}})$, 则将 $(0, +\text{BVD}_{\text{VER}})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, -\text{BVD}_{\text{VER}})$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $\text{Cost}(0, +\text{BVD}_{\text{VER}}) > \text{Cost}(0, -\text{BVD}_{\text{VER}})$, 则将 $(0, -\text{BVD}_{\text{VER}})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, +\text{BVD}_{\text{VER}})$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

对于仅 BVD 水平分量存在的情况, 假设该组合列表为 $\{(+\text{BVD}_{\text{HOR}}, 0), (-\text{BVD}_{\text{HOR}}, 0)\}$, 其对应的成本列表为 $\{\text{Cost}(+\text{BVD}_{\text{HOR}}, 0), \text{Cost}(-\text{BVD}_{\text{HOR}}, 0)\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为: 比较 $\text{Cost}(+\text{BVD}_{\text{HOR}}, 0)$ 和 $\text{Cost}(-\text{BVD}_{\text{HOR}}, 0)$ 的大小:

若 $\text{Cost}(+\text{BVD}_{\text{HOR}}, 0) \leq \text{Cost}(-\text{BVD}_{\text{HOR}}, 0)$, 则将 $(+\text{BVD}_{\text{HOR}}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(-\text{BVD}_{\text{HOR}}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $\text{Cost}(+\text{BVD}_{\text{HOR}}, 0) > \text{Cost}(-\text{BVD}_{\text{HOR}}, 0)$, 则将 $(-\text{BVD}_{\text{HOR}}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(+\text{BVD}_{\text{HOR}}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

需要说明的是, 在本申请的实施例, 如果排序结果为当前块的候选 BVD 集合, 那么相应的, 执行排序的过程可以为根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合, 其中, 候选 BVD 集合至少包括一个候选 BVD。

进一步地, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合时, 如果 absBvdX 和 absBvdY 均不等于 0, 将候选 BVD 分别存储至第一数组和第二数组; 然后确定第一数组中的、第一代价值最小的第五候选 BVD, 以及第二数组中的、第一代价值最小的第六候选 BVD; 最后可以将第五候选 BVD 和第六候选 BVD 中的、第一代价值最小的候选 BVD, 确定为候选 BVD 集合中的代价最小 BVD。

如果 BVD 水平分量和垂直分量均存在, 即 absBvdX 和 absBvdY 均不等于 0, 假设该组合列表为 $\{(+, +), (+, -), (-, -), (-, +)\}$, 其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(+, -), \text{Cost}(-, -), \text{Cost}(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp , posiGrp 存储在同一分量 (水平) 上符号为正号的两个 BVD 的符号组合, negaGrp 存储在同一分量 (水平) 上符号为负号的两个 BVD 的符号组合。

假设 $\text{posiGrp} = \{(+, +), (+, -)\}$, $\text{negaGrp} = \{(-, +), (-, -)\}$ 。 posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较: 比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(+, -)$ 的大小, 将 $\text{posiGrp}[0]$ 置为两者较小的 BVD 的符号组合, 将 $\text{posiGrp}[1]$ 置为两者较大的 BVD 的符号组合;

第二次比较: 比较 $\text{Cost}(-, +)$ 和 $\text{Cost}(-, -)$ 的大小, 将 $\text{negaGrp}[0]$ 置为两者较小的 BVD 的符号组合, 将 $\text{negaGrp}[1]$ 置为两者较大的 BVD 的符号组合;

第三次比较: 比较 $\text{posiGrp}[0]$ 和 $\text{negaGrp}[0]$ 对应的 Cost 的大小。

进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

如果仅 BVD 水平分量或者垂直分量存在, 即 absBvdX 或 absBvdY 等于 0, 那么可以直接按照第一代价值的大小对两个候选 BVD 进行排序处理, 以确定出候选 BVD 集合中的代价最小 BVD。

示例性的, 在本申请的实施例, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合时, 如果 absBvdX 等于 0 或者 absBvdY 等于 0, 那么可以将候选 BVD 中的第一代价值最小的一个候选 BVD 确定为候选 BVD 集合中的代价最小 BVD。

对于仅 BVD 垂直分量存在的情况, 假设该组合列表为 $\{(+, +), (+, -)\}$, 其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(+, -)\}$ 。上述两个列表中成员的顺序可任意排列。比较过程为: 比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(+, -)$ 的大小, 进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

对于仅 BVD 水平分量存在的情况, 假设该组合列表为 $\{(+, +), (-, +)\}$, 其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。比较过程为: 比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(-, +)$ 的大小, 进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

步骤 103、根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD。

在本申请的实施例, 在根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果之后, 便

可以根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD。

需要说明的是, 在本申请的实施例中, 如果排序处理获得的当前块的候选 BVD 列表为升(降)序的列表, 那么可以选择采用转换列表间接得到真实 BVD 的符号, 也可以直接采用逻辑推理得到真实 BVD 符号。

其中, 在采用转换列表间接得到真实 BVD 的符号时, 可以先按照转换规则将 BVD 组合列表 OrderedList 转换为索引转换列表 TransformedList。

示例性的, 在本申请的实施例中, 可以对候选 BVD 列表进行转换处理, 确定索引转换列表 (TransformedList)。

进一步地, 在本申请的实施例中, 在根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD 时, 当 absBvdX 和 absBvdY 均不为 0 时, 若 BVD 符号索引信息的第一比特位取值为 0, 则确定当前块的 BVD 的 candBvdX 与索引转换列表中第一个候选 BVD 或第三个候选 BVD 的 candBvdX 相同; 然后可以按照第一个候选 BVD 或第三个候选 BVD 确定当前块的 BVD 的 candBvdY。

进一步地, 在本申请的实施例中, 在按照第一个候选 BVD 或第三个候选 BVD 确定当前块的 BVD 的 candBvdY 时, 若 BVD 符号索引信息的第二比特位取值为 0, 则确定当前块的 BVD 的 candBvdY 与第一个候选 BVD 的 candBvdY 相同; 若第二比特位取值为 1, 则确定当前块的 BVD 的 candBvdY 与第三个候选 BVD 的 candBvdY 相同。

进一步地, 在本申请的实施例中, 在根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD 时, 当 absBvdX 和 absBvdY 均不为 0 时, 若 BVD 符号索引信息的第一比特位取值为 1, 则确定第一分量符号与索引转换列表中第二个候选 BVD 或第四个候选 BVD 的第一分量符号相同; 然后可以按照第二个候选 BVD 或第四个候选 BVD 确定当前块的 BVD 的 candBvdY。

进一步地, 在本申请的实施例中, 在按照第二个候选 BVD 或第四个候选 BVD 确定当前块的 BVD 的 candBvdY 时, 若 BVD 符号索引信息的第二比特位取值为 0, 则确定当前块的 BVD 的 candBvdY 与第二个候选 BVD 的 candBvdY 相同; 若第二比特位取值为 1, 则确定当前块的 BVD 的 candBvdY 与第四个候选 BVD 的 candBvdY 相同。

也就是说, 在本申请的实施例中, 对于 BVD 水平分量和垂直分量均存在的情况, BVD 的符号预测索引为两个比特位。

假设排序后的 BVD 组合列表 OrderedList 为:

表 4. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

利用以下转换规则将排序后的 BVD 组合列表 OrderedList 转换为 TransformedList (索引转换列表):

将排序后最小 cost 的 BVD 组合放在 TransformedList 列表的第 0 位, 即将(a1, a2)放在 TransformedList 列表的第 0 位。将与 a1 同号的 BVD 组合放在 TransformedList 列表的第 2 位, 即假设 b1 与其同号, 则将(b1, b2)放在 TransformedList 列表的第 2 位。将与 a1 异号的较小 cost 的 BVD 组合放在 TransformedList 列表的第 1 位, 即将(c1, c2)放在 TransformedList 列表的第 1 位。将与 a1 异号的较大 cost 的 BVD 组合放在 TransformedList 列表的第 3 位, 即将(d1, d2)放在 TransformedList 列表的第 3 位。

则索引转换列表 TransformedList 为:

表 5. 水平垂直 BVD 均存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	a2
1	c1(-a1)	c2
2	b1(a1)	b2
3	d1(-a1)	d2

对于索引转换列表, 按照以下规则得到真实 BVD 的符号, 此时解析的符号预测索引等于表中最左列的索引:

解析符号预测索引的第 0 个比特位:

第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1, 检查其垂直分量, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 a2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -a2。

第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 检查索引 1 的候选 BVD 的垂直分量, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 c2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -c2。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 6. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	-BVD _{HOR}	+BVD _{VER}
1	-BVD _{HOR}	-BVD _{VER}
2	+BVD _{HOR}	+BVD _{VER}
3	+BVD _{HOR}	-BVD _{VER}

则按照转换规则将排序后的 BVD 组合列表 OrderedList 转换为 TransformedList:

表 34. 水平垂直 BVD 均存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	-BVD _{HOR}	+BVD _{VER}
1	+BVD _{HOR}	+BVD _{VER}
2	-BVD _{HOR}	-BVD _{VER}

3	$+BVD_{HOR}$	$-BVD_{VER}$
---	--------------	--------------

(1) 假设解析出的

$$bvslIdx = 0$$

则真实 BVD 组合为水平分量为 $-BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ 。

(2) 假设解析出的

$$bvslIdx = 2$$

则真实 BVD 组合为水平分量为 $-BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ 。

(3) 假设解析出的

$$bvslIdx = 1$$

则真实 BVD 组合为水平分量为 $+BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ 。

(4) 假设解析出的

$$bvslIdx = 3$$

则真实 BVD 组合为水平分量为 $+BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ 。

进一步地，在本申请的实施例中，在根据 BVD 符号索引信息和排序结果，确定当前块的 BVD 时，当 $absBvdX$ 为 0 时，若 BVD 符号索引信息的取值为 0，则确定当前块的 BVD 的 $candBvdY$ 与索引转换列表中的第一个候选 BVD 的 $candBvdY$ 相同；若 BVD 符号索引信息的取值为 1，则确定当前块的 BVD 的 $candBvdY$ 与索引转换列表中的第二个候选 BVD 的 $candBvdY$ 相同。

也就是说，在本申请的实施例中，对于 BVD 垂直分量存在的情况，BVD 的符号预测索引为一个比特位。

假设排序后的 BVD 组合列表 $OrderedList$ 为：

表 7. 仅垂直 BVD 存在的排序后的 BVD 组合列表 $OrderedList$

索引	水平分量	垂直分量
0	0	$a2$
1	0	$b2(-a2)$

则索引转换列表 $TransformedList$ 与排序后的 BVD 组合列表 $OrderedList$ 相同。

索引转换列表 $TransformedList$ 为：

表 8. 仅垂直 BVD 存在的索引转换列表 $TransformedList$

索引	水平分量	垂直分量
0	0	$a2$
1	0	$b2(-a2)$

对于索引转换列表，按照以下规则得到真实 BVD 的符号，此时解析的符号预测索引等于表中最左列的索引：

如果第 0 个比特位为 0，则真实的 BVD 垂直分量为 $a2$ ，否则，真实的 BVD 垂直分量为 $-a2$ 。

例如：假设排序后的 BVD 组合列表 $OrderedList$ 为：

表 9. 仅垂直 BVD 存在的排序后的 BVD 组合列表 $OrderedList$ 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

索引转换列表 $TransformedList$ 为：

表 10. 仅垂直 BVD 存在的索引转换列表 $TransformedList$ 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

(1) 假设符号预测索引为：

$$bvslIdx = 0$$

则真实的 BVD 的水平分量为 0，垂直分量为 $+BVD_{VER}$ 。

(2) 假设符号预测索引为：

$$bvslIdx = 1$$

则真实的 BVD 的水平分量为 0，垂直分量为 $-BVD_{VER}$ 。

进一步地，在本申请的实施例中，在根据 BVD 符号索引信息和排序结果，确定当前块的 BVD 时，当 $absBvdY$ 为 0 时，若 BVD 符号索引信息的取值为 0，则确定当前块的 BVD 的 $candBvdX$ 与索引转换列表中的第一个候选 BVD 的 $candBvdX$ 相同；若 BVD 符号索引信息的取值为 1，则确定当前块的 BVD 的 $candBvdX$ 与索引转换列表中的第二个候选 BVD 的 $candBvdX$ 相同。

也就是说，在本申请的实施例中，对于 BVD 水平分量存在的情况，BVD 的符号预测索引为一个比特位。

假设排序后的 BVD 组合列表 $OrderedList$ 为：

表 11. 仅水平 BVD 存在的排序后的 BVD 组合列表 $OrderedList$

索引	水平分量	垂直分量
0	$a1$	0
1	$b1(-a1)$	0

则索引转换列表 $TransformedList$ 与排序后的 BVD 组合列表 $OrderedList$ 相同。

索引转换列表 $TransformedList$ 为：

表 12. 仅水平 BVD 存在的索引转换列表 $TransformedList$

索引	水平分量	垂直分量
----	------	------

0	a1	0
1	b1(-a1)	0

对于索引转换列表, 按照以下规则得到真实 BVD 的符号, 此时解析的符号预测索引等于表中最左列的索引: 如果第 0 个比特位为 0, 则真实的 BVD 水平分量为 a1, 否则, 真实的 BVD 水平分量为 -a1。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 13. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

索引转换列表 TransformedList 为:

表 14. 仅水平 BVD 存在的转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

(1) 假设符号预测索引为:

$$bvsdIdx = 0$$

则真实的 BVD 的水平分量为 $+BVD_{HOR}$, 垂直分量为 0。

(2) 假设符号预测索引为:

$$bvsdIdx = 1$$

则真实的 BVD 的水平分量为 $-BVD_{HOR}$, 垂直分量为 0。

进一步地, 在本申请的实施例中, 如果选择采用逻辑推理得到真实 BVD 符号, 在根据 BVD 符号索引信息和候选 BVD 列表, 确定当前块的 BVD 时, 当 $absBvdX$ 和 $absBvdY$ 均不为 0 时, 若 BVD 符号索引信息的第一比特位取值为 0, 则确定当前块的 BVD 的 $candBvdX$ 与候选 BVD 列表中的第一个候选 BVD 的 $candBvdX$ 相同, 否则不同; 然后遍历候选 BVD 列表, 确定与当前块的 BVD 具有相同 $candBvdX$ 的首个候选 BVD; 若 BVD 符号索引信息的第二比特位取值为 0, 则确定当前块的 BVD 的 $candBvdY$ 与首个候选 BVD 的 $candBvdY$ 相同, 否则不同。

第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 15. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1; 若第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 即 a1 的相反数, 此时判断真实 BVD 的水平分量为 e1。然后从索引 0 开始遍历排序后的列表 (当前块的候选 BVD 列表), 当列表中首次出现与上述判断出的真实 BVD 水平分量相同的 BVD 时, 检查其垂直分量, 假设其垂直分量为 f2, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 f2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -f2, 即 f2 的相反数, 此时判断真实 BVD 的垂直分量为 e2。这样, 就得到了真实 BVD, 即水平分量为 e1, 垂直分量为 e2 的 BVD。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 16. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$-BVD_{HOR}$	$-BVD_{VER}$
2	$+BVD_{HOR}$	$+BVD_{VER}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

(1) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 0, 第 1 个比特位为 0。

则真实 BVD 的水平分量为 $-BVD_{HOR}$, 垂直分量为 $+BVD_{VER}$ 。

(2) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 1, 第 1 个比特位为 0。

则真实 BVD 的水平分量为 $+BVD_{HOR}$, 垂直分量为 $+BVD_{VER}$ 。

(3) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 0, 第 1 个比特位为 1。

则真实 BVD 的水平分量为 $-BVD_{HOR}$, 垂直分量为 $-BVD_{VER}$ 。

(4) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 1, 第 1 个比特位为 1。

则真实 BVD 的水平分量为 $+BVD_{HOR}$, 垂直分量为 $-BVD_{VER}$ 。

进一步地, 在本申请的实施例中, 如果选择采用逻辑推理得到真实 BVD 符号, 在根据 BVD 符号索引信息和候选 BVD 列表, 确定当前块的 BVD 时, 当 $absBvdX$ 为 0 时, 若 BVD 符号索引信息的取值为 0, 则确定当前块的 BVD 的 $candBvdY$ 与候选 BVD 列表中的第一个候选 BVD 的 $candBvdY$ 相同; 若 BVD 符号索引信息的取值为 1, 则确定当前块的 BVD 的 $candBvdY$ 与候选 BVD 列表中的第二个候选 BVD 的 $candBvdY$ 相同。

第二种情况仅 BVD 的垂直分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 17. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 垂直分量等于 a2; 若第 0 个比特位为 1 时, 则真实 BVD 垂直分量等于 -a2, 即 a2 的相反数, 此时判断真实 BVD 的垂直分量为 e2, 水平分量为 0。这样, 就得到了真实 BVD, 即水平分量为 0, 垂直分量为 e2 的 BVD。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 18. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	0	+BVD _{VER}
1	0	-BVD _{VER}

(1) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 0。

则真实 BVD 的水平分量为 0, 垂直分量为 +BVD_{VER}。

(2) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 1。

则真实 BVD 的水平分量为 0, 垂直分量为 -BVD_{VER}。

进一步地, 在本申请的实施例中, 如果选择采用逻辑推理得到真实 BVD 符号, 在根据 BVD 符号索引信息和候选 BVD 列表, 确定当前块的 BVD 时, 当 absBvdY 为 0 时, 若 BVD 符号索引信息的取值为 0, 则确定当前块的 BVD 的 candBvdX 与候选 BVD 列表中的第一个候选 BVD 的 candBvdX 相同; 若 BVD 符号索引信息的取值为 1, 则确定当前块的 BVD 的 candBvdX 与候选 BVD 列表中的第二个候选 BVD 的 candBvdX 相同。

第三种情况仅 BVD 的水平分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 19. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1; 若第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 即 a1 的相反数, 此时判断真实 BVD 的水平分量为 e1, 垂直分量为 0。这样, 就得到了真实 BVD, 即水平分量为 e1, 垂直分量为 0 的 BVD。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 20. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	+BVD _{HOR}	0
1	-BVD _{HOR}	0

(1) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 0。

则真实 BVD 的水平分量为 +BVD_{HOR}, 垂直分量为 0。

(2) 假设解析出的 BVD 的符号预测索引的第 0 个比特位为 1。

则真实 BVD 的水平分量为 -BVD_{HOR}, 垂直分量为 0。

需要说明的是, 在本申请的实施例中, 如果排序处理获得的当前块的候选 BVD 列表为特定顺序的列表, 那么在确定当前块的 BVD 时, 可以直接将候选 BVD 列表中的、BVD 符号索引信息指示的候选 BVD 确定为当前块的 BVD。其中, 可以将排序后的 BVD 组合列表 OrderedList, 直接赋值给 TransformedList, 即候选 BVD 列表 OrderedList 和索引转换列表 TransformedList 是相同的。因此在 TransformedList (或 OrderedList) 中, BVD 的符号预测索引对应的 BVD 组合, 即为真实的 BVD。

可以理解的是, 在本申请的实施例中, 如果排序处理获得的当前块的候选 BVD 列表为特定顺序的列表, 那么在编码端, 可以直接对 BVD 对应的两个分量同时编码, 对应的, 在解码端, BVD 对应的两个分量可以直接通过 BVD 符号索引信息一起确定出来。

也就是说, 在本申请的实施例中, 可以直接通过解码获得的 BVD 符号索引信息, 从候选 BVD 列表中确定出 BVD 符号索引信息所指示的候选 BVD。

相应的, 在本申请的实施例中, 对于解码端解析组合索引 (BVD 符号索引信息) 后同时得出水平和垂直分量的方案, 在编码端, 编码器可以采用遍历的方法决定所传索引 (BVD 符号索引信息)。

可以理解的是, 在本申请的实施例中, 对于按照第一代价值指示的匹配误差从小到大的顺序排序所确定的候选 BVD 列表, 即在当前块的候选 BVD 列表为升 (降) 序的列表的情况下, 也可以直接将候选 BVD 列表中的、BVD 符号索引信息指示的候选 BVD 确定为当前块的 BVD。

第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 21. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2
2	c1(a1)	c2
3	d1(-a1)	d2

则索引转换列表 TransformedList 为:

表 22. 水平垂直 BVD 均存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2
2	c1(a1)	c2
3	d1(-a1)	d2

对于索引转换列表，按照以下规则得到真实 BVD 的符号，此时解析的符号预测索引等于表中最左列的索引：
解析符号预测索引的第 0 个比特位：

第 0 个比特位为 0 时，则真实 BVD 水平分量等于 a1，检查其垂直分量，若第 1 个比特位为 0，则真实 BVD 垂直分量等于 a2；若第 1 个比特位为 1 时，则真实 BVD 垂直分量等于 -a2。

第 0 个比特位为 1 时，则真实 BVD 水平分量等于 -a1，检查索引 1 的候选 BVD 的垂直分量，若第 1 个比特位为 0，则真实 BVD 垂直分量等于 c2；若第 1 个比特位为 1 时，则真实 BVD 垂直分量等于 -c2。

例如：假设排序后的 BVD 组合列表 OrderedList 为：

表 23. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$+BVD_{HOR}$	$+BVD_{VER}$
2	$-BVD_{HOR}$	$-BVD_{HOR}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

则 TransformedList 为：

表 24. 水平垂直 BVD 均存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$+BVD_{HOR}$	$+BVD_{VER}$
2	$-BVD_{HOR}$	$-BVD_{HOR}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

(1) 假设解析出的

$$bvsdIdx = 0$$

则真实 BVD 组合为水平分量为 $-BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ 。

(2) 假设解析出的

$$bvsdIdx = 2$$

则真实 BVD 组合为水平分量为 $-BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ 。

(3) 假设解析出的

$$bvsdIdx = 1$$

则真实 BVD 组合为水平分量为 $+BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ 。

(4) 假设解析出的

$$bvsdIdx = 3$$

则真实 BVD 组合为水平分量为 $+BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ 。

第二种情况仅 BVD 垂直分量存在，此时 BVD 的符号预测索引为一个比特位：

假设排序后的 BVD 组合列表 OrderedList 为：

表 25. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

则索引转换列表 TransformedList 为：

表 54. 仅垂直 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

对于索引转换列表，按照以下规则得到真实 BVD 的符号，此时解析的符号预测索引等于表中最左列的索引：
第 0 个比特位为 0 时，则真实 BVD 的垂直分量等于 a2，否则，真实 BVD 的垂直分量等于 -a2。

例如：假设排序后的 BVD 组合列表 OrderedList 为：

表 26. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

TransformedList 为：

表 56. 仅垂直 BVD 存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

(1) 假设解析出的符号预测索引为：

$$bv\text{sd}l\text{dx} = 0$$

则真实的 BVD 的水平分量为 0，垂直分量为 $+BVD_{VER}$ 。

(2) 假设解析出符号预测索引:

$$bv\text{sd}l\text{dx} = 1$$

则真实的 BVD 的水平分量为 0，垂直分量为 $-BVD_{VER}$ 。

第三种情况仅 BVD 水平分量存在，此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 27. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

则索引转换列表 TransformedList 为:

表 28. 仅水平 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

对于索引转换列表，按照以下规则得到真实 BVD 的符号，此时解析的符号预测索引等于表中最左列的索引:

第 0 个比特位为 0 时，则真实 BVD 的水平分量等于 a1，否则，真实 BVD 的水平分量等于 -a1。

例如：假设排序后的 BVD 组合列表 OrderedList 为:

表 29. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

TransformedList 为:

表 60. 仅水平 BVD 存在的转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

(1) 假设解析出的符号预测索引为:

$$bv\text{sd}l\text{dx} = 0$$

则真实的 BVD 的水平分量为 $+BVD_{HOR}$ ，垂直分量为 0。

(2) 假设解析出的符号预测索引为:

$$bv\text{sd}l\text{dx} = 1$$

则真实的 BVD 的水平分量为 $-BVD_{HOR}$ ，垂直分量为 0。

进一步地，在本申请的实施例中，如果排序处理后的排序结果为当前块的候选 BVD 集合，那么排序后即确定了模
板匹配代价最小的候选 BVD，即确定了候选 BVD 集合中的代价最小 BVD。此时，如果 $bv\text{sd}l\text{dx}$ 的数值是 $\text{bit}0 + \text{bit}1 \ll 1$ ，
即 $\text{bit}0$ 代表水平分量符号， $\text{bit}1$ 代表垂直分量符号，那么，最小的 BVD 符号组合（代价最小 BVD），确定了 $\text{bit}0$ 和 $\text{bit}1$
在取值等于 0 时的符号，取值等于 1 时的符号与此相反。

需要说明的是，在本申请的实施例中，在根据 BVD 符号索引信息和排序结果，确定当前块的 BVD 时，当 absBvdX
和 absBvdY 均不为 0 时，若 BVD 符号索引信息的第一比特位取值为 0，则确定当前块的 BVD 的 candBvdX 与候选 BVD
集合中的代价最小 BVD 的 candBvdX 相同；那么可以继续按照候选 BVD 集合中的代价最小 BVD 确定当前块的 BVD 的
 candBvdY 。

需要说明的是，在本申请的实施例中，在根据 BVD 符号索引信息和排序结果，确定当前块的 BVD 时，当 absBvdX
和 absBvdY 均不为 0 时，若 BVD 符号索引信息的第一比特位取值为 1，则确定当前块的 BVD 的 candBvdX 与候选 BVD
集合中的代价最小 BVD 的 candBvdX 符号相反；那么可以继续按照候选 BVD 集合中的代价最小 BVD 所在的数组以外的、
另一个数组中的代价最小 BVD 确定当前块的 BVD 的 candBvdY 。

进一步地，在本申请的实施例中，在按照候选 BVD 集合中的代价最小 BVD 确定当前块的 BVD 的 candBvdY 时，
若 BVD 符号索引信息的第二比特位取值为 0，则确定当前块的 BVD 的 candBvdY 与候选 BVD 集合中的代价最小 BVD
的 candBvdY 相同；若第二比特位取值为 1，则确定当前块的 BVD 的 candBvdY 与候选 BVD 集合中的代价最小 BVD 的
 candBvdY 符号相反。

进一步地，在本申请的实施例中，在按照候选 BVD 集合中的代价最小 BVD 所在的数组以外的、另一个数组中的代
价最小 BVD 确定当前块的 BVD 的 candBvdY 时，若 BVD 符号索引信息的第二比特位取值为 0，则确定当前块的 BVD
的 candBvdY 与另一个数组中的代价最小 BVD 的 candBvdY 相同；若第二比特位取值为 1，则确定当前块的 BVD 的
 candBvdY 与另一个数组中的代价最小 BVD 的 candBvdY 符号相反。

需要说明的是，在本申请的实施例中，假设真实 BVD 的水平分量和垂直分量的符号为 $\text{bvdSign}[0]$ 和 $\text{bvdSign}[1]$ 。

其中，取值为 0 则为正，取值为 1 则为负。

示例性的，在本申请的实施例中，如果 BVD 水平和垂直分量均存在，对于排序后确定的候选 BVD 集合中的代价最
小 BVD 的两种可能情况如下:

(1) 若在第三次比较中， $\text{posiGrp}[0]$ 对应的 $\text{Cost} \leq \text{negaGrp}[0]$ 对应的 Cost :

如果第 0 个比特位为 0，则 $\text{bvd_sign}[0] = 0$ 。如果第 1 个比特位为 0，则 $\text{bvd_sign}[1] = \text{posiGrp}[0][1]$ ；如果第 1
个比特位为 1，则 $\text{bvd_sign}[1] = \text{posiGrp}[1][1]$ 。

如果第 0 个比特位为 1, 则 $\text{bvd_sign}[0] = 1$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[1] = \text{negaGrp}[0][1]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[1] = \text{negaGrp}[1][1]$ 。

(2) 若在第三次比较中, $\text{posiGrp}[0]$ 对应的 $\text{Cost} > \text{negaGrp}[0]$ 对应的 Cost :

如果第 0 个比特位为 1, 则 $\text{bvd_sign}[0] = 0$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[1] = \text{posiGrp}[0][1]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[1] = \text{posiGrp}[1][1]$ 。

如果第 0 个比特位为 0, 则 $\text{bvd_sign}[0] = 1$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[1] = \text{negaGrp}[0][1]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[1] = \text{negaGrp}[1][1]$ 。

需要说明的是, 在本申请的实施例中, 在根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD 时, 当 absBvdX 为 0 时, 若 BVD 符号索引信息的取值为 0, 则确定当前块的 BVD 的 candBvdY 与候选 BVD 集合中的代价最小 BVD 的 candBvdY 相同; 若 BVD 符号索引信息的取值为 1, 则确定当前块的 BVD 的 candBvdY 与候选 BVD 集合中的代价最小 BVD 的 candBvdY 符号相反。

示例性的, 在本申请的实施例中, 如果 BVD 仅垂直分量存在, 对于排序后确定的候选 BVD 集合中的代价最小 BVD 的两种可能情况如下:

(1) 若 $\text{Cost}(+, +) \leq \text{Cost}(+, -)$:

如果第 0 个比特位为 0, 那么 $\text{bvd_sign}[1] = 0$ 。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[1] = 1$ 。

(2) 若 $\text{Cost}(+, +) > \text{Cost}(+, -)$:

如果第 0 个比特位为 0, 那么 $\text{bvd_sign}[1] = 1$ 。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[1] = 0$ 。

需要说明的是, 在本申请的实施例中, 在根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD 时, 当 absBvdY 为 0 时, 若 BVD 符号索引信息的取值为 0, 则确定当前块的 BVD 的 absBvdX 与候选 BVD 集合中的代价最小 BVD 的 absBvdX 相同; 若 BVD 符号索引信息的取值为 1, 则确定当前块的 BVD 的 absBvdX 与候选 BVD 集合中的代价最小 BVD 的 absBvdX 符号相反。

示例性的, 在本申请的实施例中, 如果 BVD 仅水平分量存在, 对于排序后确定的候选 BVD 集合中的代价最小 BVD 的两种可能情况如下:

(1) 若 $\text{Cost}(+, +) \leq \text{Cost}(-, +)$:

如果第 0 个比特位为 0, 那么 $\text{bvd_sign}[0] = 0$ 。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[0] = 1$ 。

(2) 若 $\text{Cost}(+, +) > \text{Cost}(-, +)$:

如果第 0 个比特位为 0, 那么 $\text{bvd_sign}[0] = 1$ 。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[0] = 0$ 。

可见, 上述基于候选 BVD 集合确定当前块的 BVD 的方法, 可以省去建立列表和得到符号预测索引的步骤, 对于 BVD 的水平和垂直分量均存在的情况, 解码端直接通过三次比较便可以得到真实的 BVD 符号, 复杂度更低。

步骤 104、根据当前块的 BVD, 确定当前块的重建值。

在本申请的实施例中, 在根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD 之后, 解码器便可以利用当前块的 BVD 进行运动补偿, 从而确定当前块的重建值。

示例性的, 在本申请的实施例中, 解码端整体流程如下:

存在以下几种方式:

(1) 解析 BVD 符号在解析 BVD 绝对值的过程之后进行:

$\text{if}(\text{general_merge_flag}[x0][y0])$	
...	
$\text{else if}(\text{CuPredMode}[\text{chType}][x0][y0] == \text{MODE_IBC}) \{$	
$\text{bvd_coding}(x0, y0, 0, 0)$	
$\text{bvds_data}(x0, y0)$	
...	
$\}$	

$\text{bvd_coding}(x0, y0, \text{refList}, \text{cplIdx}) \{$	Descriptor
$\text{abs_bvd_greater0_flag}[0]$	$\text{ae}(v)$
$\text{abs_bvd_greater0_flag}[1]$	$\text{ae}(v)$
$\text{if}(\text{abs_bvd_greater0_flag}[0]) \{$	
$\text{abs_bvd_minus1}[0]$	$\text{ae}(v)$
$\}$	
$\text{if}(\text{abs_bvd_greater0_flag}[1]) \{$	
$\text{abs_bvd_minus1}[1]$	$\text{ae}(v)$
$\}$	
$\}$	

bvsd_data(x0, y0) {	Descriptor
if(abs_bvd_greater0_flag[0] && abs_bvd_greater0_flag[1])	
bvsd_idx	ae(v)
else if(abs_bvd_greater0_flag[0] abs_bvd_greater0_flag[1])	
bvsd_idx_single	ae(v)
}	

符号预测索引（即 BVD 符号索引信息）bvdIdx 的推导如下：
若 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]都为 TRUE 时，则 bvsdIdx = bvsd_idx；
否则，若 abs_bvd_greater0_flag[0]或 abs_bvd_greater0_flag[1]一者为 TRUE 时，则 bvsdIdx = bvsd_idx_single。
compIdx = 0..1 的运动矢量差异的绝对值 abs_bvd[compIdx]的推导如下：
abs_bvd[compIdx] = abs_bvd_greater0_flag[compIdx] * (abs_bvd_minus1[compIdx]+ 1)
bvsd_idx 既可以使用不同的两个概率模型：

表 30

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	abs_bvd[0] <= Thres ? 0:1	abs_bvd[1] <= Thres ? 0:1	na	na	na	na

或使用不同的四个概率模型：

表 31

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	abs_bvd[0] <= Thres ? 0:1	abs_bvd[1] <= Thres ? 2:3	na	na	na	na

也可以是一个使用概率模型，一个使用旁路编码：

表 32

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	abs_bvd[0] <= Thres ? 0:1	bypass	na	na	na	na

或

表 33

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	abs_bvd[1] <= Thres ? 0:1	na	na	na	na

bvsd_idx_single 既可以根据绝对值大小使用不同的概率模型：

表 34

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx_single	abs_bvd[compIdx] <= Thres ? 0:1	na	na	na	na	na

compIdx=0..1，表示水平分量或者垂直分量。
也可以使用一个单独的概率模型：

表 35

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx_single	0	na	na	na	na	na

也可以使用旁路编码：

表 36

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx_single	bypass	na	na	na	na	na

(2) 解析 BVD 符号的过程在解析 BVD 绝对值的函数 bvd_coding() 中间进行, 即在解析 BVD 的水平和垂直分量是否为零的标识 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]之后进行:

if(general_merge_flag[x0][y0])	
...	
else if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
bvd_coding(x0, y0, 0, 0)	
...	
}	

bvd_coding(x0, y0, refList, cpIdx) {	Descriptor
abs_bvd_greater0_flag[0]	ae(v)
abs_bvd_greater0_flag[1]	ae(v)
if(abs_bvd_greater0_flag[0] && abs_bvd_greater0_flag[1])	
bvsd_idx	ae(v)
else if(abs_bvd_greater0_flag[0] abs_bvd_greater0_flag[1])	
bvsd_idx_single	ae(v)
if(abs_bvd_greater0_flag[0]) {	
abs_bvd_minus1[0]	ae(v)
}	
if(abs_bvd_greater0_flag[1]) {	
abs_bvd_minus1[1]	ae(v)
}	
}	

- 5
- 符号预测索引 bvdIdx 的推导如下:
 若 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]都为 TRUE 时, 则 bvsdIdx = bvsd_idx;
 否则, 若 abs_bvd_greater0_flag[0]或 abs_bvd_greater0_flag[1]一者为 TRUE 时, 则 bvsdIdx = bvsd_idx_single。
bvsd_idx 既可以使用不同的两个概率模型:

表 37

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	na	na	na	na

- 10
- 也可以是一个使用概率模型, 一个使用旁路编码:

表 38

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	bypass	na	na	na	na

或

表 39

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	0	na	na	na	na

bvsd_idx_single 既可以使用单独的概率模型:

- 15
- 表 40

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx_single	0	na	na	na	na	na

也可以使用旁路编码:

表 41

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx_single	bypass	na	na	na	na	na

(3) 解析 BVD 符号的过程在解析 BVD 绝对值的函数 bvd_coding() 中间进行, 即在解析 BVD 的水平和垂直分量是否为零的标识 abs_bvd_greater0_flag[0] 和 abs_bvd_greater0_flag[1] 之后进行, 但与 (2) 不同的是仅引入一个 bvsd_idx 变量, 该变量的二值化移至 CABAC 中处理 (该过程涉及的二值化与 CABAC 的处理流程在 (1) 中也可实现, 即解析 BVD 符号的过程在解析 BVD 绝对值的函数 bvd_coding() 之后进行也可应用类似于下述的二值化流程):

if(general_merge_flag[x0][y0])	
...	
else if(CuPredMode[chType][x0][y0] == MODE_IBC) {	
bvd_coding(x0, y0, 0, 0)	
...	
}	

bvd_coding(x0, y0, refList, cpIdx) {	Descriptor
abs_bvd_greater0_flag[0]	ae(v)
abs_bvd_greater0_flag[1]	ae(v)
if(abs_bvd_greater0_flag[0] abs_bvd_greater0_flag[1])	
bvsd_idx	ae(v)
if(abs_bvd_greater0_flag[0]){	
abs_bvd_minus1[0]	ae(v)
}	
if(abs_bvd_greater0_flag[1]){	
abs_bvd_minus1[1]	ae(v)
}	
}	

符号预测索引 bvdIdx 的推导如下:

$$\text{bvsdIdx} = \text{bvsd_idx}$$

下面描述了 bvsd_idx 的二值化过程及对应的具体形式, 包括但不限于以下几种:

1) 使用定长码 (Fixed-length binarization) 对 bvsd_idx 进行二值化:

此时, 存在两种形式:

第一种形式为:

若 abs_bvd_greater0_flag[0] 和 abs_bvd_greater0_flag[1] 都为 TRUE 时:

表 42

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	FL	cMax = 3

否则, 若 abs_bvd_greater0_flag[0] 或 abs_bvd_greater0_flag[1] 一者为 TRUE 时:

表 43

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	FL	cMax = 1

对上述过程进行简述:

若 abs_bvd_greater0_flag[0] 和 abs_bvd_greater0_flag[1] 都为 TRUE 时, 当解析到的 bvsd_idx 的二值化后二进制串为 00B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 01B 时, bvsd_idx 为 1; 当解析到的 bvsd_idx 的二值化后二进制串为 10B 时, bvsd_idx 为 2; 当解析到的 bvsd_idx 的二值化后二进制串为 11B 时, bvsd_idx 为 3。

否则, 若 abs_bvd_greater0_flag[0] 或 abs_bvd_greater0_flag[1] 一者为 TRUE 时, 当解析到的 bvsd_idx 的二值化后二进制串为 0B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 1B 时, bvsd_idx 为 1。

此时, 关于 bvsd_idx 语法元素的上下文概率模型与 binIdx 的关系为:

若 abs_bvd_greater0_flag[0] 和 abs_bvd_greater0_flag[1] 都为 TRUE 时:

既可以是不同的两个概率模型:

表 44

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	na	na	na	na

也可以是一个使用概率模型, 一个使用旁路编码:

表 45

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	bypass	na	na	na	na

或

表 46

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	0	na	na	na	na

否则, 若 abs_bvd_greater0_flag[0]或 abs_bvd_greater0_flag[1]一者为 TRUE 时:

既可以使用独立的概率模型:

表 47

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	na	na	na	na	na

也可以使用旁路编码:

表 48

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	na	na	na	na	na

第二种形式为:

若 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]都为 TRUE 时:

表 49

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	FL	cMax = 3

即若 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]都为 TRUE 时, 当解析到的 bvsd_idx 的二值化后二进制串为 00B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 01B 时, bvsd_idx 为 1; 当解析到的 bvsd_idx 的二值化后二进制串为 10B 时, bvsd_idx 为 2; 当解析到的 bvsd_idx 的二值化后二进制串为 11B 时, bvsd_idx 为 3。

否则, 若 abs_bvd_greater0_flag[0]或 abs_bvd_greater0_flag[1]一者为 TRUE 时:

表 50

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	FL	cMax = 3

即当解析到的 bvsd_idx 的二值化后二进制串为 00B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 01B 时, bvsd_idx 为 1。

此时, 关于 bvsd_idx 语法元素的上下文概率模型与 binIdx 的关系为:

若 abs_bvd_greater0_flag[0]和 abs_bvd_greater0_flag[1]都为 TRUE 时, 与第一种形式相同;

否则, 若 abs_bvd_greater0_flag[0]或 abs_bvd_greater0_flag[1]一者为 TRUE 时:

既可以使用一个或两个概率模型：

表 51

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	0	na	na	na	na

或

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	na	na	na	na

也可以独立的概率模型和旁路编码共用：

5

表 52

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	0	na	na	na	na

也可以均使用旁路编码：

表 53

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	bypass	bypass	na	na	na	na

10 2) 使用截断二进制码 (Truncated binarization) 对 bvsd_idx 进行二值化：

表 54

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	TB	cMax = 5

当解析到的 bvsd_idx 的二值化后二进制串为 00B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 01B 时, bvsd_idx 为 1; 当解析到的 bvsd_idx 的二值化后二进制串为 100B 时, bvsd_idx 为 2; 当解析到的 bvsd_idx 的二值化后二进制串为 101B 时, bvsd_idx 为 3。

15 此时, 关于 bvsd_idx 语法元素的上下文概率模型与 binIdx 的关系为：

既可以使用三个概率模型, 或者两个, 或者一个：

表 55

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	2	na	na	na

或

表 56

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	1	na	na	na

20

或

表 57

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	0	0	na	na	na

也可以将上述的任意 binIdx 使用旁路编码, 即可任意组合, 此处仅举一例：

表 58

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	bypass	na	na	na

3) 使用零阶指数哥伦布码 (Exp-Golomb binarization) 对 bvsd_idx 进行二值化:

表 59

Syntax element	Binarization	
	Process	Input parameters
bvsd_idx	EG0	

当解析到的 bvsd_idx 的二值化后二进制串为 1B 时, bvsd_idx 为 0; 当解析到的 bvsd_idx 的二值化后二进制串为 010B 时, bvsd_idx 为 1; 当解析到的 bvsd_idx 的二值化后二进制串为 011B 时, bvsd_idx 为 2; 当解析到的 bvsd_idx 的二值化后二进制串为 00100B 时, bvsd_idx 为 3。

此时, 关于 bvsd_idx 语法元素的上下文概率模型与 binIdx 的关系为:

可以使用概率模型与旁路编码的任意组合, 下述仅举两例:

表 60

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	1	2	3	4	na

或

表 61

Syntax element	binIdx					
	0	1	2	3	4	>= 5
bvsd_idx	0	0	bypass	1	1	na

综上所述, 通过上述步骤 101 至步骤 104 所提出的解码方法, 通过在 BVD 的可能符号和 BVD 的绝对值之间创建 BVD 的组合列表, 并对组合列表利用模板进行排序, 利用真实的 BVD 符号和排序后的列表得到符号预测索引进行 CABAC 的编码, 相较于比旁路编码具有更高的编码效率, 从而节省了 IBC 中的 BVD 传输所需的码率。

需要说明的是, 本申请实施例提出的解码方法, 一方面, 可以充分利用模板与当前编码块的相似性, 将 BVD 的符号组合列表进行有效地排序; 另一方面, 可以使用 CABAC 进行编码, 比旁路编码具有更高的编码效率。

本申请实施例提供了一种编解码方法, 在解码端, 解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息; 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD; 根据当前块的 BVD, 确定当前块的重建值。在编码端, 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息; 将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见, 在本申请的实施例中, 可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理, 从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输, 在解码时利用排序结果进行当前块的 BVD 符号信息的解析, 同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理, 能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率, 从而可以提高编码效率。

基于上述实施例, 本申请的再一实施例提出了一种解码方法, 其中, 对解码器而言, 在进行解码处理时, BVD 解码输入可以为 BVD 绝对值和符号 (BVD 符号索引信息) 的码流信息。而 BVD 解码输出可以为 BVD 绝对值和符号的真实信息。

其中, 图 10 为本申请实施例中的解码方法的示意图二, 如图 10 所示, BVD 解码过程包含解析 BVD 绝对值 (步骤 201)、解析 BVD 的符号预测索引, 即解析 BVD 符号索引信息 (步骤 202)、通过在 BVD 的可能符号和 BVD 绝对值之间创建 BVD 的组合列表, 即确定当前块的候选 BVD (步骤 203)、根据当前块的 BVP 和 BVD 的组合列表进行模板预测, 计算出 BVD 组合列表中的每个 BVD 组合的成本, 即对组合列表计算模板区域的成本 (第一代价值) (步骤 204), 对列表排序并根据符号预测索引和排序列表推导真实 BVD, 并进行运动补偿 (步骤 205)。

对于步骤 201 所提出的解析 BVD 绝对值, 解码器可以直接从码流中解析出真实的 BVD 绝对值。即解码器解码码流, 可以确定当前块的 BVD 绝对值。

对于步骤 202 所提出的符号预测索引 (BVD 符号索引信息) 的确定, 首先, 可以根据解码过程中的、步骤 201 所解析出来的 BVD 绝对值来判断符号预测索引的二进制数的比特位的数目。其中, 若 BVD 的水平分量和垂直分量的绝对值都不为 0 时, 则符号预测索引的二进制数的比特位的数目为 2; 若 BVD 的水平分量为 0 并且垂直分量的绝对值不为 0 时, 则符号预测索引的二进制数的比特位的数目为 1; 若 BVD 的水平分量不为 0 并且垂直分量的绝对值为 0 时, 则符号预测索引的二进制数的比特位的数目为 1。

然后, 解码器可以使用 CABAC (上下文自适应的二进制算术编码) 或旁路对应的解码过程将符号预测索引对应的二进制数可以从码流中解析出来。

针对符号预测索引所使用的 CABAC 的上下文模型, 可以使用单个概率模型, 或者多个概率模型, 可以包括但不限于

于根据 BVD 的水平分量或垂直分量的绝对值大小、BVD 的水平分量与垂直分量的和或差值等进行区分不同的概率模型。

对于步骤 203 所提出的确定当前块的候选 BVD 过程,即通过在 BVD 的可能符号和 BVD 绝对值之间创建 BVD 的组合列表(候选 BVD 或初始集合),可以通过不同的情况创建 BVD 的组合列表,比如仅水平 BVD 存在、仅垂直 BVD 存在以及二者都存在的情况。

5 BVD 组合列表的确定包括对 BVD 的可能符号进行排列组合、对可能的水平符号和垂直符号分别乘以 BVD 的水平绝对值分量和垂直绝对值分量。

为了更好的说明如何构建 BVD 组合列表(候选 BVD 或初始集合),以三个简单分类为例说明:

分类例子 1: BVD 分为水平分量和垂直分量,当水平分量为零的时候,只对垂直分量进行编码,所以只对垂直分量的符号构建组合列表,具体为以下两种情况:

10 第一种情况:水平分量都为正号,只是一个占位,垂直分量先为正号,后为负号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值,进而得到 BVD 的组合列表。

第二种情况:水平分量都为正号,只是一个占位,垂直分量先为负号,后为正号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值,进而得到 BVD 的组合列表。

15 分类例子 2: BVD 分为水平分量和垂直分量,当垂直分量为零的时候,只对水平分量进行编码,所以只对水平分量的符号构建组合列表,具体为以下两种情况:

第一种情况:垂直分量都为正号,只是一个占位,水平分量先为正号,后为负号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值,进而得到 BVD 的组合列表。

第二种情况:垂直分量都为正号,只是一个占位,水平分量先为负号,后为正号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值,进而得到 BVD 的组合列表。

20 分类例子 3: BVD 分为水平分量和垂直分量,当两分量均不为零的时候,二者均需要进行编码,所以要同时对水平和垂直分量一起构建 BVD 列表,具体分为 24 种情况,即 $\{+1, +1\}$ 、 $\{+1, -1\}$ 、 $\{-1, +1\}$ 和 $\{-1, -1\}$ 这 4 种组合下的任意排列,下述举例说明:

列表第 0 位为水平和垂直分量均为正号,列表第 1 位为水平分量为正号垂直分量为负号,列表第 2 位为水平分量为负号垂直分量为正号,列表第 3 位为水平分量和垂直分量均为负号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值,进而得到 BVD 的组合列表。

25 对于步骤 204 所提出的根据当前块的 BVP 和 BVD 的组合列表(候选 BVD)进行模板预测,计算出 BVD 组合列表中的每个 BVD 组合(候选 BVD)的成本(第一代价值),即计算模板区域的成本,可以对 BVD 组合列表中的各种组合利用模板进行成本的计算,具体为利用当前块模板位置处的重建亮度像素和利用当前块的 BVP 及 BVD 的组合得出的 BV 进行模板预测得到的位置处的重建亮度像素计算成本。

30 需要说明的是,在本申请的实施例中,在计算成本(第一代价值)时,不仅可以采用亮度,还可以包含色度即 Cb 和 Cr 分量。本申请不进行具体限定。

在进行模板(当前块的第一模板)的选择时,可以根据当前块的邻近区域的像素可用性,判断模板位置的像素是否可用,包括重建亮度信息。如图 6 所示,根据模板与当前块的相对位置关系,可将模板分类为上模板、左模板、右上模板、左下模板、左上模板等模板类型。其中,不同编码块的不同类型的模板尺寸可以固定相同,也可以不同。

35 例如,模板尺寸对当前任意编码块(当前块)选择相同的模板尺寸,也可以根据当前编码块的不同尺寸选择不同的模板尺寸,还可以根据当前亮度编码块内的像素数选择不同的模板尺寸。

在进行当前块的 BV 预测时,可以利用当前块的 BVP 以及 BVD 的组合列表,产生新的 BV。如图 8 所示,利用当前块判断这个 BV 是否可用(可用的条件为该 BV 指向的参考块已经重建,且未超出 IBC 所设定的搜索范围以及图像边界等条件),如果当前 BV 是可用的,然后就会利用当前块的模板和新的 BV 进行运动补偿,得到对应 BV 处的模板。如图 9 所示, BVD 水平和垂直分量共产生 4 种组合的 BVD,分别进行模板处的运动补偿。计算成本时的模板的使用情况存在以下几种情况:

第一种:当前块的上模板和左模板均存在,且对应 BV 处的上模板和左模板均存在,此时上模板和左模板均可用。

45 第二种:当前块的上模板和左模板均存在,对应 BV 处仅上模板存在,此时存在两种方法:方法一:对应 BV 处左模板不存在,直接不用,即仅用上模板计算。方法二:对应 BV 处左模板不存在,采用参考块内部最左侧的模板宽度列代替左模板,即采用上模板和左模板计算。

第三种:当前块的上模板和左模板均存在,对应 BV 处仅左模板存在,此时存在两种方法:方法一:对应 BV 处上模板不存在,直接不用,即仅用左模板计算。方法二:对应 BV 处上模板不存在,采用参考块内部最上侧的模板高度行代替上模板,即采用上模板和左模板计算。

50 第四种:当前块仅上模板存在,此时存在三种方法:方法一:对应 BV 处上模板不存在,直接不用。方法二:对应 BV 处上模板不存在,采用参考块内部最上侧的模板高度行代替上模板。方法三:对应 BV 处上模板存在,直接用。

第五种:当前块仅左模板存在,此时存在三种方法:方法一:对应 BV 处左模板不存在,直接不用。方法二:对应 BV 处左模板不存在,采用参考块内部最左侧的模板宽度列代替左模板。方法三:对应 BV 处左模板存在,直接用。

第六种:当前块上模板和左模板均不存在,忽略此种情况的 BVD 组合情况。

55 以上六种情况下,当 BV 指向的参考块不可用时,均忽略此种情况下的 BVD 组合情况。其中,当忽略某种情况下的 BVD 组合情况时,将此种情况下的成本置为成本评价准则的最大值。

在进行成本计算时,即确定第一代价值时,对于计算模板区域的成本的成本函数有多种选择,即预设的误差准则可以有多种选择。例如,可以选择绝对误差和(SAD)、变换绝对误差和(SATD)、差值平方和(SSE)、平均绝对差值(MAD)、平均绝对误差(MAE)、平均平方误差(MSE)、率失真函数(RDO)等评价准则。

60 在通过上述步骤就 204 计算模板区域的成本(第一代价值)之后,便可以执行步骤 205,即按照模板区域得到的成本对组合列表进行排序,即可以利用第一代价值对候选 BVD 进行排序处理,并利用 BVD 的符号预测索引和排序后的列表(当前块的候选 BVD 列表),得到真实的 BVD,然后再进行运动补偿。

在根据模板区域得到的成本对组合列表进行排序,即按照第一代价值对候选 BVD 进行排序处理时,既可以采用排列为升(降)序的新列表 OrderedList (当前块的候选 BVD 列表),也可以排列为特定顺序的新列表 OrderedList。

(1) 排列为升(降)序的新列表 OrderedList。

例如,采用冒泡排序、选择排序、插入排序、希尔排序、归并排序、快速排序、基数排序、堆排序、计数排序、桶排序等排序方法。

(2) 排列为特定顺序的新列表 OrderedList。

包括但不限于以下形式:这里列举一种特定顺序的排列方法:

第一种情况:假设该组合列表(候选 BVD)为 $\{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, +BVD_{VER})\}$,其对应的成本列表为 $\{Cost(+BVD_{HOR}, +BVD_{VER}), Cost(+BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, +BVD_{VER})\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp, posiGrp 存储在同一分量(水平)上符号为正号的两个 BVD 组合, negaGrp 存储在同一分量(水平)上符号为负号的两个 BVD 组合。

假设 $posiGrp = \{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER})\}$, $negagr = \{(-BVD_{HOR}, +BVD_{VER}), (-BVD_{HOR}, -BVD_{VER})\}$ 。posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较:比较 $Cost(+BVD_{HOR}, +BVD_{VER})$ 和 $Cost(+BVD_{HOR}, -BVD_{VER})$ 的大小,将 posiGrp[0]置为两者较小的 BVD 组合,将 posiGrp[1]置为两者较大的 BVD 组合;

第二次比较:比较 $Cost(-BVD_{HOR}, +BVD_{VER})$ 和 $Cost(-BVD_{HOR}, -BVD_{VER})$ 的大小,将 negaGrp[0]置为两者较小的 BVD 组合,将 negaGrp[1]置为两者较大的 BVD 组合;

第三次比较:比较 posiGrp[0]和 negaGrp[0]对应的 Cost 的大小;

最后的排序过程:

若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negaGrp[0]$ 对应的 Cost, 将 posiGrp[0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 negaGrp[0] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 posiGrp[1] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 negaGrp[1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, $posiGrp[0]$ 对应的 $Cost > negaGrp[0]$ 对应的 Cost, 将 negaGrp [0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 posiGrp [0] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 negaGrp [1] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 posiGrp [1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

其中,当前块的候选 BVD 列表的第 0 位的候选 BVD 即为候选 BVD 列表中的第一个候选 BVD, 第 1 位的候选 BVD 即为候选 BVD 列表中的第二个候选 BVD, 第 2 位的候选 BVD 即为候选 BVD 列表中的第三个候选 BVD, 第 3 位的候选 BVD 即为候选 BVD 列表中的第四个候选 BVD。

还有一种排序的实施方式:

若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negaGrp[0]$ 对应的 Cost, 将 posiGrp[0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 negaGrp[0] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 posiGrp[1] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 negaGrp[1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, $posiGrp[0]$ 对应的 $Cost > negaGrp[0]$ 对应的 Cost, 将 negaGrp [0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 posiGrp [0] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 negaGrp [1] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 posiGrp [1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

第二种情况:假设该组合列表为 $\{(0, +BVD_{VER}), (0, -BVD_{VER})\}$,其对应的成本列表为 $\{Cost(0, +BVD_{VER}), Cost(0, -BVD_{VER})\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为:比较 $Cost(0, +BVD_{VER})$ 和 $Cost(0, -BVD_{VER})$ 的大小:

若 $Cost(0, +BVD_{VER}) \leq Cost(0, -BVD_{VER})$, 则将 $(0, +BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, -BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $Cost(0, +BVD_{VER}) > Cost(0, -BVD_{VER})$, 则将 $(0, -BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, +BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

第三种情况:假设该组合列表为 $\{(+BVD_{HOR}, 0), (-BVD_{HOR}, 0)\}$,其对应的成本列表为 $\{Cost(+BVD_{HOR}, 0), Cost(-BVD_{HOR}, 0)\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为:比较 $Cost(+BVD_{HOR}, 0)$ 和 $Cost(-BVD_{HOR}, 0)$ 的大小:

若 $Cost(+BVD_{HOR}, 0) \leq Cost(-BVD_{HOR}, 0)$, 则将 $(+BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(-BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $Cost(+BVD_{HOR}, 0) > Cost(-BVD_{HOR}, 0)$, 则将 $(-BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(+BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

在根据模板区域得到的成本对组合列表进行排序,即按照第一代价值对候选 BVD 进行排序处理之后,获得的排序结果也可以为当前块的候选 BVD 集合。其中,可以为根据第一代价值,对候选 BVD 进行排序,确定当前块的候选 BVD 集合,其中,候选 BVD 集合至少包括一个候选 BVD。

如果 BVD 水平分量和垂直分量均存在,即 $absBvdX$ 和 $absBvdY$ 均不等于 0, 假设该组合列表为 $\{(+, +), (+, -), (-, -), (-, +)\}$,其对应的成本列表为 $\{Cost(+, +), Cost(+, -), Cost(-, -), Cost(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp, posiGrp 存储在同一分量(水平)上符号为正号的两个 BVD 的符号组合, negaGrp 存储在同一分量(水平)上符号为负号的两个 BVD 的符号组合。

假设 $posiGrp = \{(+, +), (+, -)\}$, $negagr = \{(-, +), (-, -)\}$ 。posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较:比较 $Cost(+, +)$ 和 $Cost(+, -)$ 的大小,将 posiGrp[0]置为两者较小的 BVD 的符号组合,将 posiGrp[1]

置为两者较大的 BVD 的符号组合;

第二次比较: 比较 Cost(-, +)和 Cost(-, -)的大小, 将 negaGrp[0]置为两者较小的 BVD 的符号组合, 将 negaGrp[1]置为两者较大的 BVD 的符号组合;

第三次比较: 比较 posiGrp[0]和 negaGrp[0]对应的 Cost 的大小。

5 进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

对于仅 BVD 垂直分量存在的情况, 假设该组合列表为{(+, +), (+, -)}, 其对应的成本列表为{Cost(+, +), Cost(+, -)}。上述两个列表中成员的顺序可任意排列。比较过程为: 比较 Cost(+, +)和 Cost(+, -)的大小, 进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

10 对于仅 BVD 水平分量存在的情况, 假设该组合列表为{(+, +), (-, +)}, 其对应的成本列表为{Cost(+, +), Cost(-, +)}。上述两个列表中成员的顺序可任意排列。比较过程为: 比较 Cost(+, +)和 Cost(-, +)的大小, 进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

在利用 BVD 的符号预测索引 (BVD 符号索引信息) 和排序后的列表 (当前块的候选 BVD 列表), 确定真实的 BVD 时, 因为排序后的列表 (当前块的候选 BVD 列表) 分为三种情况, 即 BVD 仅存在水平分量、BVD 仅存在垂直分量和 BVD 既存在水平分量又存在垂直分量, 所以这里也分三种情况介绍。

15 其中, 对于排列为升 (降) 序的新列表 OrderedList, 可以存在两种方案, 一种是采用转换列表间接得到真实 BVD 的符号, 另一种是直接采用逻辑推理得到真实 BVD 符号。

在采用转换列表间接得到真实 BVD 的符号, 解码器需要先按照转换规则将 BVD 组合列表 OrderedList 转换为索引转换列表 TransformedList。

第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

20 假设排序后的 BVD 组合列表 OrderedList 为:

表 62. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

利用以下转换规则将排序后的 BVD 组合列表 OrderedList 转换为 TransformedList (索引转换列表):

将排序后最小 cost 的 BVD 组合放在 TransformedList 列表的第 0 位, 即将(a1, a2)放在 TransformedList 列表的第 0 位。将与 a1 同号的 BVD 组合放在 TransformedList 列表的第 2 位, 即假设 b1 与其同号, 则将(b1, b2)放在 TransformedList 列表的第 2 位。将与 a1 异号的较小 cost 的 BVD 组合放在 TransformedList 列表的第 1 位, 即将(c1, c2)放在 TransformedList 列表的第 1 位。将与 a1 异号的较大 cost 的 BVD 组合放在 TransformedList 列表的第 3 位, 即将(d1, d2)放在 TransformedList 列表的第 3 位。

25 则索引转换列表 TransformedList 为:

表 63. 水平垂直 BVD 均存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	a2
1	c1(-a1)	c2
2	b1(a1)	b2
3	d1(-a1)	d2

30 对于索引转换列表, 按照以下规则得到真实 BVD 的符号, 此时解析的符号预测索引等于表中最左列的索引: 解析符号预测索引的第 0 个比特位:

第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1, 检查其垂直分量, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 a2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -a2。

35 第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 检查索引 1 的候选 BVD 的垂直分量, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 c2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -c2。

第二种情况仅 BVD 垂直分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 64. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

则索引转换列表 TransformedList 与排序后的 BVD 组合列表 OrderedList 相同。

40 索引转换列表 TransformedList 为:

表 65. 仅垂直 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

对于索引转换列表, 按照以下规则得到真实 BVD 的符号, 此时解析的符号预测索引等于表中最左列的索引: 如果第 0 个比特位为 0, 则真实的 BVD 垂直分量为 a2, 否则, 真实的 BVD 垂直分量为 -a2。

第三种情况仅 BVD 水平分量存在, 此时 BVD 的符号预测索引为一个比特位:

45 假设排序后的 BVD 组合列表 OrderedList 为:

表 66. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

则索引转换列表 TransformedList 与排序后的 BVD 组合列表 OrderedList 相同。

索引转换列表 TransformedList 为:

表 67. 仅水平 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

对于索引转换列表, 按照以下规则得到真实 BVD 的符号, 此时解析的符号预测索引等于表中最左列的索引:

如果第 0 个比特位为 0, 则真实的 BVD 水平分量为 a1, 否则, 真实的 BVD 水平分量为 -a1。

在直接采用逻辑推理得到真实 BVD 符号时, 第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 68. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1; 若第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 即 a1 的相反数, 此时判断真实 BVD 的水平分量为 e1。然后从索引 0 开始遍历排序后的列表 (当前块的候选 BVD 列表), 当列表中首次出现与上述判断出的真实 BVD 水平分量相同的 BVD 时, 检查其垂直分量, 假设其垂直分量为 f2, 若第 1 个比特位为 0, 则真实 BVD 垂直分量等于 f2; 若第 1 个比特位为 1 时, 则真实 BVD 垂直分量等于 -f2, 即 f2 的相反数, 此时判断真实 BVD 的垂直分量为 e2。这样, 就得到了真实 BVD, 即水平分量为 e1, 垂直分量为 e2 的 BVD。

第二种情况仅 BVD 的垂直分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 69. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 垂直分量等于 a2; 若第 0 个比特位为 1 时, 则真实 BVD 垂直分量等于 -a2, 即 a2 的相反数, 此时判断真实 BVD 的垂直分量为 e2, 水平分量为 0。这样, 就得到了真实 BVD, 即水平分量为 0, 垂直分量为 e2 的 BVD。

第三种情况仅 BVD 的水平分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 70. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

若解析出的 BVD 的符号预测索引的第 0 个比特位为 0 时, 则真实 BVD 水平分量等于 a1; 若第 0 个比特位为 1 时, 则真实 BVD 水平分量等于 -a1, 即 a1 的相反数, 此时判断真实 BVD 的水平分量为 e1, 垂直分量为 0。这样, 就得到了真实 BVD, 即水平分量为 e1, 垂直分量为 0 的 BVD。

进一步地, 对于排列为特定顺序的新列表 OrderedList, 可以将排序后的 BVD 组合列表 OrderedList, 直接赋值给 TransformedList。

在 TransformedList 中, BVD 的符号预测索引对应的 BVD 组合, 即为真实的 BVD。

第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 71. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2
2	c1(a1)	c2
3	d1(-a1)	d2

则索引转换列表 TransformedList 为:

表 72. 水平垂直 BVD 均存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2

2	c1(a1)	c2
3	d1(-a1)	d2

对于索引转换列表,按照以下规则得到真实 BVD 的符号,此时解析的符号预测索引等于表中最左列的索引:

解析符号预测索引的第 0 个比特位:

第 0 个比特位为 0 时,则真实 BVD 水平分量等于 a1,检查其垂直分量,若第 1 个比特位为 0,则真实 BVD 垂直分量等于 a2;若第 1 个比特位为 1 时,则真实 BVD 垂直分量等于 -a2。

第 0 个比特位为 1 时,则真实 BVD 水平分量等于 -a1,检查索引 1 的候选 BVD 的垂直分量,若第 1 个比特位为 0,则真实 BVD 垂直分量等于 c2;若第 1 个比特位为 1 时,则真实 BVD 垂直分量等于 -c2。

第二种情况仅 BVD 垂直分量存在,此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 73. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

则索引转换列表 TransformedList 为:

表 74. 仅垂直 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

对于索引转换列表,按照以下规则得到真实 BVD 的符号,此时解析的符号预测索引等于表中最左列的索引:

第 0 个比特位为 0 时,则真实 BVD 的垂直分量等于 a2,否则,真实 BVD 的垂直分量等于 -a2。

第三种情况仅 BVD 水平分量存在,此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 75. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

则索引转换列表 TransformedList 为:

表 76. 仅水平 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

对于索引转换列表,按照以下规则得到真实 BVD 的符号,此时解析的符号预测索引等于表中最左列的索引:

第 0 个比特位为 0 时,则真实 BVD 的水平分量等于 a1,否则,真实 BVD 的水平分量等于 -a1。

进一步地,在本申请的实施例中,如果排序处理后的排序结果为当前块的候选 BVD 集合,那么排序后即确定了模板匹配代价最小的候选 BVD,即确定了候选 BVD 集合中的代价最小 BVD。此时,如果 bvsdIdx 的数值是 bit0+bit1<<1,即 bit0 代表水平分量符号,bit1 代表垂直分量符号,那么,最小的 BVD 符号组合(代价最小 BVD),确定了 bit0 和 bit1 在取值等于 0 时的符号,取值等于 1 时的符号与此相反。

需要说明的是,在本申请的实施例中,假设真实 BVD 的水平分量和垂直分量的符号为 bvdSign[0] 和 bvdSign[1]。其中,取值为 0 则为正,取值为 1 则为负。

第一种情况: BVD 水平和垂直分量均存在。

(1) 若在第三次比较中, posiGrp[0] 对应的 Cost ≤ negaGrp[0] 对应的 Cost:

如果第 0 个比特位为 0,则 bvd_sign[0] = 0。如果第 1 个比特位为 0,则 bvd_sign[1] = posiGrp[0][1]; 如果第 1 个比特位为 1,则 bvd_sign[1] = posiGrp[1][1]。

如果第 0 个比特位为 1,则 bvd_sign[0] = 1。如果第 1 个比特位为 0,则 bvd_sign[1] = negaGrp[0][1]; 如果第 1 个比特位为 1,则 bvd_sign[1] = negaGrp[1][1]。

(2) 若在第三次比较中, posiGrp[0] 对应的 Cost > negaGrp[0] 对应的 Cost:

如果第 0 个比特位为 1,则 bvd_sign[0] = 0。如果第 1 个比特位为 0,则 bvd_sign[1] = posiGrp[0][1]; 如果第 1 个比特位为 1,则 bvd_sign[1] = posiGrp[1][1]。

如果第 0 个比特位为 0,则 bvd_sign[0] = 1。如果第 1 个比特位为 0,则 bvd_sign[1] = negaGrp[0][1]; 如果第 1 个比特位为 1,则 bvd_sign[1] = negaGrp[1][1]。

第二种情况: BVD 仅垂直分量存在。

(1) 若 Cost(+, +) ≤ Cost(+, -):

如果第 0 个比特位为 0,那么 bvd_sign[1] = 0。

如果第 0 个比特位为 1,那么 bvd_sign[1] = 1。

(2) 若 Cost(+, +) > Cost(+, -):

如果第 0 个比特位为 0,那么 bvd_sign[1] = 1。

如果第 0 个比特位为 1,那么 bvd_sign[1] = 0。

第三种情况: BVD 仅水平分量存在。

(1) 若 Cost(+, +) ≤ Cost(-, +):

如果第 0 个比特位为 0,那么 bvd_sign[0] = 0。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[0] = 1$ 。

(2) 若 $\text{Cost}(+, +) > \text{Cost}(-, +)$:

如果第 0 个比特位为 0, 那么 $\text{bvd_sign}[0] = 1$ 。

如果第 0 个比特位为 1, 那么 $\text{bvd_sign}[0] = 0$ 。

5 可见, 上述基于候选 BVD 集合确定当前块的 BVD 的方法, 可以省去建立列表和得到符号预测索引的步骤, 解码端直接通过三次比较便可以得到真实的 BVD 符号。

示例性的, 在本申请的实施例中, 以 CABAC 模型中有两个概率模型的情况为例, 解析位置如主技术方案的解码端整体流程中的 (1) 所述的位于 $\text{bvd_coding}()$ 之后为例, 解码端的整体流程如下:

$\text{if}(\text{general_merge_flag}[x0][y0])$	
...	
$\text{else if}(\text{CuPredMode}[\text{chType}][x0][y0] == \text{MODE_IBC}) \{$	
$\text{bvd_coding}(x0, y0, 0, 0)$	
$\text{bvsd_data}(x0, y0, 0)$	
$\text{bvsd_data}(x0, y0, 1)$	
...	
$\}$	

$\text{bvd_coding}(x0, y0, \text{refList}, \text{cplDx}) \{$	Descriptor
$\text{abs_bvd_greater0_flag}[0]$	$\text{ae}(v)$
$\text{abs_bvd_greater0_flag}[1]$	$\text{ae}(v)$
$\text{if}(\text{abs_bvd_greater0_flag}[0]) \{$	
$\text{abs_bvd_minus1}[0]$	$\text{ae}(v)$
$\}$	
$\text{if}(\text{abs_bvd_greater0_flag}[1]) \{$	
$\text{abs_mvd_minus1}[1]$	$\text{ae}(v)$
$\}$	
$\}$	

$\text{bvsd_data}(x0, y0, \text{compIdx}) \{$	Descriptor
$\text{if}(\text{abs_bvd_greater0_flag}[\text{compIdx}])$	
$\text{bvd_sign_bin}[\text{compIdx}]$	$\text{ae}(v)$
$\}$	

若 $\text{abs_bvd_greater0_flag}[0]$ 和 $\text{abs_bvd_greater0_flag}[1]$ 都为 TRUE 时, 则 $\text{bvd_sign_bin}[0]$ 为上述的第 0 个比特位, $\text{bvd_sign_bin}[1]$ 为第 1 个比特位;

若仅 $\text{abs_bvd_greater0_flag}[0]$ 为 TRUE 时, 则 $\text{bvd_sign_bin}[0]$ 为上述的第 0 个比特位;

若仅 $\text{abs_bvd_greater0_flag}[1]$ 为 TRUE 时, 则 $\text{bvd_sign_bin}[1]$ 为上述的第 0 个比特位。

15 $\text{compIdx} = 0..1$ 的运动矢量差异的绝对值 $\text{abs_bvd}[\text{compIdx}]$ 的推导如下:

$$\text{abs_bvd}[\text{compIdx}] = \text{abs_bvd_greater0_flag}[\text{compIdx}] * (\text{abs_bvd_minus1}[\text{compIdx}] + 1)$$

表 77

Syntax element	binIdx					
	0	1	2	3	4	≥ 5
$\text{bvd_sign_bin}[\text{compIdx}]$	$\text{abs_bvd}[\text{compIdx}] \leq \text{Thres} ? 0:1$	na	na	na	na	na

$\text{compIdx} = 0..1$, 表示水平分量或者垂直分量。

真实 BVD 的推导过程如下:

$$\text{bvd}[\text{compIdx}] = \text{abs_bvd_greater0_flag}[\text{compIdx}] * (\text{abs_bvd_minus1}[\text{compIdx}] + 1) * (1 - 2 * \text{bvd_sign}[\text{compIdx}])$$

进一步地, 在本申请的实施例中, 对于上述实施例在采用特定顺序的候选 BVD 列表 OrderedList 和确定候选 BVD 集合中所提出的使用两个数组分别存储候选 BVD 的方案, 其中, BVD 水平垂直分量均存在的情况下, 水平分量和垂直分量的先后顺序可进行互换。

25 示例性的, 在本申请的实施例中, 如果 BVD 水平和垂直分量均存在, 假设该组合列表为 $\{(+, +), (+, -), (-, -), (-, +)\}$, 其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(+, -), \text{Cost}(-, -), \text{Cost}(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp , posiGrp 存储在同一分量 (垂直) 上符号为正号的两个 BVD 的符号组合, negaGrp 存储在同一分量 (垂直) 上符号为负号的两个 BVD 的符号组合。

假设 $\text{posiGrp} = \{(+, +), (-, +)\}$, $\text{negaGrp} = \{(+, -), (-, -)\}$ 。 posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较: 比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(-, +)$ 的大小, 将 $\text{posiGrp}[0]$ 置为两者较小的 BVD 组合, 将 $\text{posiGrp}[1]$ 置为两者较大的 BVD 组合;

第二次比较: 比较 $\text{Cost}(+, -)$ 和 $\text{Cost}(-, -)$ 的大小, 将 $\text{negaGrp}[0]$ 置为两者较小的 BVD 组合, 将 $\text{negaGrp}[1]$ 置为两者较大的 BVD 组合;

第三次比较: 比较 $\text{posiGrp}[0]$ 和 $\text{negaGrp}[0]$ 对应的 Cost 的大小;

(1) 若在第三次比较中, $\text{posiGrp}[0]$ 对应的 $\text{Cost} \leq \text{negaGrp}[0]$ 对应的 Cost:

如果第 0 个比特位为 0, 则 $\text{bvd_sign}[1] = 0$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[0] = \text{posiGrp}[0][0]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[0] = \text{posiGrp}[1][0]$ 。

如果第 0 个比特位为 1, 则 $\text{bvd_sign}[1] = 1$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[0] = \text{negaGrp}[0][0]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[0] = \text{negaGrp}[1][0]$ 。

(2) 若在第三次比较中, $\text{posiGrp}[0]$ 对应的 $\text{Cost} > \text{negaGrp}[0]$ 对应的 Cost:

如果第 0 个比特位为 1, 则 $\text{bvd_sign}[1] = 0$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[0] = \text{posiGrp}[0][0]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[0] = \text{posiGrp}[1][0]$ 。

如果第 0 个比特位为 0, 则 $\text{bvd_sign}[1] = 1$ 。如果第 1 个比特位为 0, 则 $\text{bvd_sign}[0] = \text{negaGrp}[0][0]$; 如果第 1 个比特位为 1, 则 $\text{bvd_sign}[0] = \text{negaGrp}[1][0]$ 。

进一步地, 在本申请的实施例, 在 BVD 水平垂直分量均存在的情况下, 水平分量和垂直分量的先后顺序可进行互换的操作流程, 可以通过某种准则来决策水平分量和垂直分量的先后顺序。其中包括但不限于以下的方式, 通过 BVD 的水平分量和垂直分量的绝对值大小等准则来决策水平分量和垂直分量的先后顺序。

进一步地, 在本申请的实施例, 针对排序后的 BVD 组合列表 (候选 BVD 列表), 可以存储它们各自的 cost, 然后基于这些存储的 cost 进行一个重新排序, 具体过程为: 如果列表中一个候选与前一个候选之间的 cost 差异低于 λ 值, 例如 $|D1 - D2| < \lambda$, 则认为该候选是冗余的, 其中 D1 和 D2 是在第一次排序后获得的 cost, λ 是编码器端 RD 准则中使用的拉格朗日参数。

示例性的, 图 11 为计算成本的示意图, 如图 11 所示, 计算成本的算法定义如下:

(1) 确定列表所有候选中候选与其之前候选之间的最小成本差异

如果最小成本差异大于等于 λ , 则认为列表足够多样化并且停止重排。

如果这个最小成本差异小于 λ , 则候选被认为是多余的, 并将其移动到另外一个位置。这个位置是和其之前的候选相比足够多样化的一个位置。

(2) 该算法在有限次数的迭代后停止。

由此可见, 本申请实施例提出的编解码方法可以有效地节省 IBC 模式中的 BVD 的符号传输所需的码率。具体地, 可以通过在 BVD 的可能符号和 BVD 的绝对值之间创建 BVD 的组合列表, 并对组合列表利用模板进行排序, 利用真实的 BVD 符号和排序后的列表得到符号预测索引进行 CABAC 的编码。所提方案比旁路编码具有更高的编码效率, 从而节省了 IBC 中的 BVD 传输所需的码率。

也就是说, 本申请实施例提出了一种新的 IBC 中的 BVD 的符号传输方法, 可以充分利用模板与当前编码块的相似性, 将 BVD 的符号组合列表进行有效地排序; 同时使用 CABAC 进行编码, 比旁路编码具有更高的编码效率。

本申请实施例提供了一种编解码方法, 在解码端, 解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息; 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD; 根据当前块的 BVD, 确定当前块的重建值。在编码端, 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息; 将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见, 在本申请的实施例, 可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理, 从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输, 在解码时利用排序结果进行当前块的 BVD 符号信息的解析, 同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理, 能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率, 从而可以提高编码效率。

本申请的实施例提出了一种编码方法, 该编码方法可以应用于编码器, 图 12 为本申请实施例中的编码方法的示意图一, 如图 12 所示, 解码器进行解码处理的方法可以包括以下步骤:

步骤 301、根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果。

在本申请的实施例, 编码器可以先确定当前块的 BVD 绝对值, 然后根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 从而可以确定排序结果。

可以理解的是, 在本申请的实施例, BVD 绝对值包括第一分量绝对值 absBvdX 和第二分量绝对值 absBvdY ; 即 BVD 绝对值包括水平分量绝对值和垂直分量绝对值。相应的, 当前块的 BVD 可以包括第一分量 candBvdX 和第二分量 candBvdY , 同时, 当前块的候选 BVD 也包括第一分量 candBvdX 和第二分量 candBvdY , 即 BVD 可以包括水平分量和垂直分量。

进一步地, 在本申请的实施例, 如果 absBvdX 和 absBvdY 均不为 0, 那么可以确定 BVD 符号索引信息的比特位数为 2; 如果 absBvdX 或 absBvdY 为 0, 那么可以确定 BVD 符号索引信息的比特位数为 1。

需要说明的是, 在本申请的实施例, 除了例如上述的定长编码方法以外, 还可以采用变长编码方法对 BVD 符号索引信息进行编码处理。例如采用截断一元码编码 BVD 符号索引信息, 既可以使用使用截断二进制码 (Truncated binarization) 进行二值化处理, 此时 BVD 符号索引信息的比特位数不一定为 2。

也就是说, 在本申请的实施例, 如果采用截断一元码编码 BVD 符号索引信息 (这里, 该 BVD 符号索引信息可以为候选 BVD 列表中的索引值), 则不同候选 BVD 对应的符号索引信息的比特位数不完全相同, 即为变长编码。

进一步地, 在本申请的实施例, 在确定出当前块的 BVD 绝对值之后, 可以利用 BVD 绝对值确定 BVD 的组合列

表, 其中, BVD 的组合列表中包括当前块的任意数量的候选 BVD。即可以利用 BVD 绝对值确定当前块的候选 BVD, 也可以理解为利用 BVD 绝对值确定当前块的候选 BVD 的初始集合。

需要说明的是, 在本申请的实施例中, 可以通过在 BVD 的可能符号和 BVD 绝对值之间创建 BVD 的组合列表 (候选 BVD)。其中, 可以包括仅水平 BVD 存在、仅垂直 BVD 存在以及二者都存在的情况。

进一步地, 在本申请的实施例中, 在创建 BVD 的组合列表 (候选 BVD) 时, 主要包括对 BVD 的可能符号进行排列组合、对可能的水平符号和垂直符号分别乘以 BVD 的水平绝对值分量和垂直绝对值分量。

示例性的, 在本申请的实施例中, 如果仅垂直 BVD 存在, 即如果 absBvdX 等于 0, 那么可以根据 absBvdY 确定候选 BVD。

例如, BVD 分为水平分量和垂直分量, 当水平分量为零的时候, 只对垂直分量进行编码, 所以只对垂直分量的符号构建组合列表, 具体为以下两种情况:

第一种情况: 水平分量都为正号, 只是一个占位, 垂直分量先为正号, 后为负号, 构建一个符号列表, 然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值, 进而得到 BVD 的组合列表。

```
{
    {+1, +1 },
    {+1, -1 },
};
{
    { 0, +BVDVER },
    { 0, -BVDVER },
}
```

第二种情况: 水平分量都为正号, 只是一个占位, 垂直分量先为负号, 后为正号, 构建一个符号列表, 然后将构建的符号列表乘以 BVD 的水平和垂直分量的绝对值, 进而得到 BVD 的组合列表。

```
{
    {+1, -1 },
    {+1, +1 },
};
{
    { 0, -BVDVER },
    { 0, +BVDVER },
}
```

示例性的, 在本申请的实施例中, 如果仅水平 BVD 存在, 即如果 absBvdY 等于 0, 那么可以根据 absBvdX 确定候选 BVD。

例如, BVD 分为水平分量和垂直分量, 当垂直分量为零的时候, 只对水平分量进行编码, 所以只对水平分量的符号构建组合列表, 具体为以下两种情况:

第一种情况: 垂直分量都为正号, 只是一个占位, 水平分量先为正号, 后为负号, 构建一个符号列表, 然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值, 进而得到 BVD 的组合列表。

```
{
    {+1, +1 },
    {-1, +1 },
};
{
    {+BVDHOR, 0 },
    {-BVDHOR, 0 },
}
```

第二种情况: 垂直分量都为正号, 只是一个占位, 水平分量先为负号, 后为正号, 构建一个符号列表, 然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值, 进而得到 BVD 的组合列表。

```
{
    {-1, +1 },
    {+1, +1 },
};
{
    {-BVDHOR, 0 },
    {+BVDHOR, 0 },
}
```

示例性的, 在本申请的实施例中, 如果水平 BVD 和垂直 BVD 均存在, 即如果 absBvdX 和 absBvdY 均不等于 0, 那么可以根据 absBvdX 和 absBvdY 确定候选 BVD。

例如, BVD 分为水平分量和垂直分量, 当两分量均不为零的时候, 二者均需要进行编码, 所以要对水平和垂直分量一起构建 BVD 列表, 具体分为 24 种情况, 即 $\{+1, +1\}$ 、 $\{+1, -1\}$ 、 $\{-1, +1\}$ 和 $\{-1, -1\}$ 这 4 种组合下的任意排列, 下述举例说明:

列表第 0 位为水平和垂直分量均为正号, 列表第 1 位为水平分量为正号垂直分量为负号, 列表第 2 位为水平分量为

负号垂直分量为正号,列表第3位为水平分量和垂直分量均为负号,构建一个符号列表,然后将构建的符号列表乘以 BVD 的垂直和水平分量的绝对值,进而得到 BVD 的组合列表。

```
{
    {+1, +1 },
    {+1, -1 },
    {-1, +1 },
    {-1, -1 },
};
{
    {+BVDHOR, +BVDVER },
    {+BVDHOR, -BVDVER },
    {-BVDHOR, +BVDVER },
    {-BVDHOR, -BVDVER },
}
```

进一步地,在本申请的实施例中,在确定当前块的 BVD 绝对值之后,可以进一步根据 BVD 绝对值,对当前块的候选 BVD 进行排序处理,确定排序结果。

进一步地,在本申请的实施例中,在根据 BVD 绝对值,对当前块的候选 BVD 进行排序处理,确定排序结果时,可以先根据 BVD 绝对值,确定当前块的一个或多个候选 BVD 对应的第一代价值;然后可以根据第一代价值,对候选 BVD 进行排序,确定排序结果。

可以理解的是,在本申请的实施例中,一个或多个候选 BVD 可以包括 candBvdX 分别设置为等于 $-\text{absBvdX}$ 或 absBvdX 、candBvdY 分别设置为等于 $-\text{absBvdY}$ 或 absBvdY 的 BVD。

需要说明的是,在本申请的实施例中,在根据当前块的 BVD 绝对值,确定当前块的一个或多个候选 BVD 对应的第一代价值时,可以先根据候选 BVD 确定的第一匹配模板;然后根据预设的误差准则,计算当前块的第一模板与第一匹配模板之间的匹配误差,以确定候选 BVD 对应的第一代价值。

示例性的,在本申请的实施例中,在进行成本计算时,即确定第一代价值时,对于计算模板区域的成本的成本函数有多种选择,即预设的误差准则可以有多种选择。例如,可以选择绝对误差和 (SAD)、变换绝对误差和 (SATD)、差值平方和 (SSE)、平均绝对差值 (MAD)、平均绝对误差 (MAE)、平均平方误差 (MSE)、率失真函数 (RDO) 等评价准则。下述内容所提到的某种评价准则均可从上述准则中选择其一,以 SAD 的评价准则为例,计算公式如下:

$$SAD = \sum_{i=0}^{\text{predTempSizeW}-1} \sum_{j=0}^{\text{predTempSizeH}-1} |\text{predTemp}[i][j] - \text{recTempC}[i][j]| \quad (1)$$

其中, predTempSizeW 为模板的宽度, predTempSizeH 为模板的高度, $\text{predTemp}[i][j]$ 为 BV 处模板的像素点, $\text{recTempC}[i][j]$ 为当前块模板的像素点。

也就是说,在本申请的实施例中,预设的误差准则可以包括绝对误差和 SAD、变换绝对误差和 SATD、差值平方和 SSE、平均绝对差值 MAD、平均绝对误差 MAE、平均平方误差 MSE、率失真函数 RDO 中的任意一种。

可以理解的是,在本申请的实施例中,第一模板包括当前块的相邻已解码区域中的一个或多个样值。

示例性的,在本申请的实施例中,在进行模板(当前块的第一模板)的选择时,可以根据当前块的邻近区域的像素可用性,判断模板位置的像素是否可用,包括重建亮度信息。如图 6 所示,根据模板与当前块的相对位置关系,可将模板分类为上模板、左模板、右上模板、左下模板、左上模板等模板类型。其中,不同编码块的不同类型的模板尺寸可以固定相同,也可以不同。

例如,模板尺寸对当前任意编码块(当前块)选择相同的模板尺寸,也可以根据当前编码块的不同尺寸选择不同的模板尺寸,还可以根据当前亮度编码块内的像素数选择不同的模板尺寸。

需要说明的是,在本申请的实施例中,在根据候选 BVD 确定的第一匹配模板时,可以先确定当前块的块向量预测值 BVP;然后根据 BVP 和候选 BVD 确定当前块的候选 BV;其中,候选 BV 用于指示第一匹配模板的位置;接着,可以根据候选 BV 确定第一匹配模板。

示例性的,在本申请的实施例中,假设 BVP 包括第三分量 currBvpX 和第四分量 currBvpY ,那么在根据 BVP 和候选 BVD 确定当前块的候选 BV 时,可以选择将候选 BV 的二维矢量设置为 $(\text{currBvpX}+\text{candBvdX}, \text{currBvpY}+\text{candBvdY})$,即根据候选 BVD 的第一分量 candBvdX、第二分量 candBvdY,以及 BVP 的第三分量 currBvpX 、第四分量 currBvpY 进行 BV 的二维矢量的确定。

进一步地,在本申请的实施例中,在根据候选 BV 确定第一匹配模板时,可以根据当前点的位置和候选 BV 的二维矢量确定第一匹配模板。

示例性的,在本申请的实施例中,可以选择以当前点的位置为起点,将候选 BV 的二维矢量所指示的、与第一模板具有相同形状和包含相同数量样值的区域,确定为第一匹配模板。

在进行当前块的 BV 预测时,可以利用当前块的 BVP 以及 BVD 的组合列表,产生新的 BV。如图 8 所示,利用当前块判断这个 BV 是否可用(可用的条件为该 BV 指向的参考块已经重建,且未超出 IBC 所设定的搜索范围以及图像边界等条件),如果当前 BV 是可用的,然后就会利用当前块的模板和新的 BV 进行运动补偿,得到对应 BV 处的模板。如图 9 所示, BVD 水平和垂直分量共产生 4 种组合的 BVD,分别进行模板处的运动补偿。计算成本时的模板的使用情况存在以下几种情况:

第一种:当前块的上模板和左模板均存在,且对应 BV 处的上模板和左模板均存在,此时上模板和左模板均可用。

第二种:当前块的上模板和左模板均存在,对应 BV 处仅上模板存在,此时存在两种方法:方法一:对应 BV 处左模板不存在,直接不用,即仅用上模板计算。方法二:对应 BV 处左模板不存在,采用参考块内部最左侧的模板宽度列代替左模板,即采用上模板和左模板计算。

第三种:当前块的上模板和左模板均存在,对应 BV 处仅左模板存在,此时存在两种方法:方法一:对应 BV 处上

模板不存在, 直接不用, 即仅用左模板计算。方法二: 对应 BV 处上模板不存在, 采用参考块内部最上侧的模板高度行代替上模板, 即采用上模板和左模板计算。

第四种: 当前块仅上模板存在, 此时存在三种方法: 方法一: 对应 BV 处上模板不存在, 直接不用。方法二: 对应 BV 处上模板不存在, 采用参考块内部最上侧的模板高度行代替上模板。方法三: 对应 BV 处上模板存在, 直接用。

第五种: 当前块仅左模板存在, 此时存在三种方法: 方法一: 对应 BV 处左模板不存在, 直接不用。方法二: 对应 BV 处左模板不存在, 采用参考块内部最左侧的模板宽度列代替左模板。方法三: 对应 BV 处左模板存在, 直接用。

第六种: 当前块上模板和左模板均不存在, 忽略此种情况的 BVD 组合情况。

以上六种情况下, 当 BV 指向的参考块不可用时, 均忽略此种情况下的 BVD 组合情况。当忽略某种情况下的 BVD 组合情况时, 将此种情况下的成本置为成本评价准则的最大值。

进一步地, 在本申请的实施例, 如果上述两个模板均能获取到, 此时对这个模板进行处理, 包括但不限于以下方法: 将两者进行加权融合。其中, 加权融合的标准可以选择固定加权系数或者根据各自模板的重建亮度与预测亮度的像素值之差的绝对值之和进行分配加权系数等。

由此可见, 在本申请的实施例, 可以根据当前块的 BVP 和 BVD 的组合列表 (候选 BVD) 进行模板预测, 计算出 BVD 组合列表中的每个 BVD 组合 (候选 BVD) 的成本 (第一代价值), 即计算模板区域的成本, 可以对 BVD 组合列表中的各种组合利用模板进行成本的计算, 具体为利用当前块模板位置处的重建亮度像素和利用当前块的 BVP 及 BVD 的组合得出的 BV 进行模板预测得到的位置处的重建亮度像素计算成本。

需要说明的是, 在本申请的实施例, 在计算成本 (第一代价值) 时, 不仅可以采用亮度, 还可以包含色度即 Cb 和 Cr 分量。本申请不进行具体限定。

进一步地, 在本申请的实施例, 在计算获得每一个候选 BVD 的第一代价值之后, 便可以利用第一代价值对候选 BVD 进行排序处理, 从而可以获得排序结果。其中, 排序结果可以为排序后的当前块的候选 BVD 列表, 也可以为通过排序确定出第一代价值最小的候选 BVD 的候选 BVD 集合。本申请不进行具体限定。

需要说明的是, 在本申请的实施例, 如果排序结果为当前块的候选 BVD 列表, 那么相应的, 执行排序的过程可以为根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 列表, 其中, 候选 BVD 列表至少包括一个候选 BVD。

进一步地, 在本申请的实施例, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 列表时, 可以选择根据预设的误差准则, 按第一代价值指示的匹配误差从小到大的顺序, 对一个或多个候选 BVD 进行排序, 确定候选 BVD 列表。

可以理解的是, 在本申请的实施例, 排序结果可以为排列为升 (降) 序的新列表 OrderedList (当前块的候选 BVD 列表)。例如, 例如, 采用冒泡排序、选择排序、插入排序、希尔排序、归并排序、快速排序、基数排序、堆排序、计数排序、桶排序等排序方法。

进一步地, 在本申请的实施例, 排序结果还可以为特定顺序的新列表 OrderedList (当前块的候选 BVD 列表)。

示例性的, 在本申请的实施例, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 列表时, 如果 $absBvdX$ 和 $absBvdY$ 均不等于 0, 将候选 BVD 分别存储至第一数组和第二数组; 然后基于第一代价值, 确定第一数组中的第一候选 BVD 和第二候选 BVD, 以及第二数组中的第三候选 BVD 和第四候选 BVD; 最后便可以基于第一代价值, 对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理, 确定候选 BVD 列表。

如果 BVD 水平分量和垂直分量均存在, 即 $absBvdX$ 和 $absBvdY$ 均不等于 0, 假设该组合列表 (候选 BVD) 为 $\{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, -BVD_{VER}), (-BVD_{HOR}, +BVD_{VER})\}$, 其对应的成本列表为 $\{Cost(+BVD_{HOR}, +BVD_{VER}), Cost(+BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, -BVD_{VER}), Cost(-BVD_{HOR}, +BVD_{VER})\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 $posiGrp$ 和 $negaGrp$, $posiGrp$ 存储在同一分量 (水平) 上符号为正号的两个 BVD 组合, $negaGrp$ 存储在同一分量 (水平) 上符号为负号的两个 BVD 组合。

假设 $posiGrp = \{(+BVD_{HOR}, +BVD_{VER}), (+BVD_{HOR}, -BVD_{VER})\}$, $negaGrp = \{(-BVD_{HOR}, +BVD_{VER}), (-BVD_{HOR}, -BVD_{VER})\}$ 。 $posiGrp$ 和 $negaGrp$ 两个列表中成员的顺序可任意排列。

第一次比较: 比较 $Cost(+BVD_{HOR}, +BVD_{VER})$ 和 $Cost(+BVD_{HOR}, -BVD_{VER})$ 的大小, 将 $posiGrp[0]$ 置为两者较小的 BVD 组合, 将 $posiGrp[1]$ 置为两者较大的 BVD 组合;

第二次比较: 比较 $Cost(-BVD_{HOR}, +BVD_{VER})$ 和 $Cost(-BVD_{HOR}, -BVD_{VER})$ 的大小, 将 $negaGrp[0]$ 置为两者较小的 BVD 组合, 将 $negaGrp[1]$ 置为两者较大的 BVD 组合;

第三次比较: 比较 $posiGrp[0]$ 和 $negaGrp[0]$ 对应的 Cost 的大小。

需要说明的是, 在本申请的实施例, 在基于第一代价值, 对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理, 确定候选 BVD 列表时, 一种可用的方式如下: 如果第一候选 BVD 的第一代价值小于等于第三候选 BVD 的第一代价值, 将第一候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将第三候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD, 将第二候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD, 将第四候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。如果第一候选 BVD 的第一代价值大于第三候选 BVD 的第一代价值, 将第三候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将第一候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD, 将第四候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD, 将第二候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。

相应的, 最后的排序过程可以为: 若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negaGrp[0]$ 对应的 Cost, 将 $posiGrp[0]$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $negaGrp[0]$ 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 $posiGrp[1]$ 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 $negaGrp[1]$ 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, $posiGrp[0]$ 对应的 Cost > $negaGrp[0]$ 对应的 Cost, 将 $negaGrp[0]$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $posiGrp[0]$ 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 $negaGrp[1]$ 排序在候选 BVD 列表

表 OrderedList 的第 2 位, 将 posiGrp [1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

其中, 当前块的候选 BVD 列表的第 0 位的候选 BVD 即为候选 BVD 列表中的第一个候选 BVD, 第 1 位的候选 BVD 即为候选 BVD 列表中的第二个候选 BVD, 第 2 位的候选 BVD 即为候选 BVD 列表中的第三个候选 BVD, 第 3 位的候选 BVD 即为候选 BVD 列表中的第四个候选 BVD。

需要说明的是, 在本申请的实施例中, 在基于第一代价值, 对第一候选 BVD、第二候选 BVD、第三候选 BVD、第四候选 BVD 进行排序处理, 确定候选 BVD 列表时, 另一种可用的方式如下: 如果第一候选 BVD 的第一代价值小于等于第三候选 BVD 的第一代价值, 将第一候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将第二候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD, 将第三候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD, 将第四候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。如果第一候选 BVD 的第一代价值大于第三候选 BVD 的第一代价值, 将第三候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将第四候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD, 将第一候选 BVD 设置为候选 BVD 列表中的第三个候选 BVD, 将第二候选 BVD 设置为候选 BVD 列表中的第四个候选 BVD。

相应的, 最后的排序过程可以为: 若在第三次比较中, posiGrp[0] 对应的 Cost $\leq$ negaGrp[0] 对应的 Cost, 将 posiGrp[0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 negaGrp[0] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 posiGrp[1] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 negaGrp[1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

若在第三次比较中, posiGrp[0] 对应的 Cost $>$ negaGrp[0] 对应的 Cost, 将 negaGrp [0] 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 posiGrp [0] 排序在候选 BVD 列表 OrderedList 的第 2 位, 将 negaGrp [1] 排序在候选 BVD 列表 OrderedList 的第 1 位, 将 posiGrp [1] 排序在候选 BVD 列表 OrderedList 的第 3 位。

示例性的, 在本申请的实施例中, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 列表时, 如果 absBvdX 等于 0 或者 absBvdY 等于 0, 将候选 BVD 中的第一代价值最小的一个候选 BVD 设置为候选 BVD 列表中的第一个候选 BVD, 将候选 BVD 中的另一个候选 BVD 设置为候选 BVD 列表中的第二个候选 BVD。

如果仅 BVD 水平分量或者垂直分量存在, 即 absBvdX 或 absBvdY 等于 0, 那么可以直接按照第一代价值的大小对两个候选 BVD 进行排序处理, 以生成候选 BVD 列表。

对于仅 BVD 垂直分量存在的情况, 假设该组合列表为 $\{(0, +BVD_{VER}), (0, -BVD_{VER})\}$, 其对应的成本列表为 $\{Cost(0, +BVD_{VER}), Cost(0, -BVD_{VER})\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为: 比较 $Cost(0, +BVD_{VER})$ 和 $Cost(0, -BVD_{VER})$ 的大小:

若 $Cost(0, +BVD_{VER}) \leq Cost(0, -BVD_{VER})$, 则将 $(0, +BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, -BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $Cost(0, +BVD_{VER}) > Cost(0, -BVD_{VER})$, 则将 $(0, -BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(0, +BVD_{VER})$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

对于仅 BVD 水平分量存在的情况, 假设该组合列表为 $\{(+BVD_{HOR}, 0), (-BVD_{HOR}, 0)\}$, 其对应的成本列表为 $\{Cost(+BVD_{HOR}, 0), Cost(-BVD_{HOR}, 0)\}$ 。上述两个列表中成员的顺序可任意排列。

排序过程为: 比较 $Cost(+BVD_{HOR}, 0)$ 和 $Cost(-BVD_{HOR}, 0)$ 的大小:

若 $Cost(+BVD_{HOR}, 0) \leq Cost(-BVD_{HOR}, 0)$, 则将 $(+BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(-BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位;

若 $Cost(+BVD_{HOR}, 0) > Cost(-BVD_{HOR}, 0)$, 则将 $(-BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 0 位, 将 $(+BVD_{HOR}, 0)$ 排序在候选 BVD 列表 OrderedList 的第 1 位。

需要说明的是, 在本申请的实施例中, 如果排序结果为当前块的候选 BVD 集合, 那么相应的, 执行排序的过程可以为根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合, 其中, 候选 BVD 集合至少包括一个候选 BVD。

进一步地, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合时, 如果 absBvdX 和 absBvdY 均不等于 0, 将候选 BVD 分别存储至第一数组和第二数组; 然后确定第一数组中的、第一代价值最小的第五候选 BVD, 以及第二数组中的、第一代价值最小的第六候选 BVD; 最后可以将第五候选 BVD 和第六候选 BVD 中的、第一代价值最小的候选 BVD, 确定为候选 BVD 集合中的代价最小 BVD。

如果 BVD 水平分量和垂直分量均存在, 即 absBvdX 和 absBvdY 均不等于 0, 假设该组合列表为 $\{(+, +), (+, -), (-, -), (-, +)\}$, 其对应的成本列表为 $\{Cost(+, +), Cost(+, -), Cost(-, -), Cost(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp, posiGrp 存储在同一分量 (水平) 上符号为正号的两个 BVD 的符号组合, negaGrp 存储在同一分量 (水平) 上符号为负号的两个 BVD 的符号组合。

假设 $posiGrp = \{(+, +), (+, -)\}$, $negagr = \{(-, +), (-, -)\}$ 。posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较: 比较 $Cost(+, +)$ 和 $Cost(+, -)$ 的大小, 将 posiGrp[0] 置为两者较小的 BVD 的符号组合, 将 posiGrp[1] 置为两者较大的 BVD 的符号组合;

第二次比较: 比较 $Cost(-, +)$ 和 $Cost(-, -)$ 的大小, 将 negaGrp[0] 置为两者较小的 BVD 的符号组合, 将 negaGrp[1] 置为两者较大的 BVD 的符号组合;

第三次比较: 比较 posiGrp[0] 和 negaGrp[0] 对应的 Cost 的大小。

进而可以确定出 Cost 最小的一个 BVD 的符号组合, 即确定出候选 BVD 集合中的代价最小 BVD。

如果仅 BVD 水平分量或者垂直分量存在, 即 absBvdX 或 absBvdY 等于 0, 那么可以直接按照第一代价值的大小对两个候选 BVD 进行排序处理, 以确定出候选 BVD 集合中的代价最小 BVD。

示例性的, 在本申请的实施例中, 在根据第一代价值, 对候选 BVD 进行排序, 确定当前块的候选 BVD 集合时, 如果 absBvdX 等于 0 或者 absBvdY 等于 0, 那么可以将候选 BVD 中的第一代价值最小的一个候选 BVD 确定为候选 BVD

集合中的代价最小 BVD。

对于仅 BVD 垂直分量存在的情况，假设该组合列表为 $\{(+, +), (+, -)\}$ ，其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(+, -)\}$ 。上述两个列表中成员的顺序可任意排列。比较过程为：比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(+, -)$ 的大小，进而可以确定出 Cost 最小的一个 BVD 的符号组合，即确定出候选 BVD 集合中的代价最小 BVD。

对于仅 BVD 水平分量存在的情况，假设该组合列表为 $\{(+, +), (-, +)\}$ ，其对应的成本列表为 $\{\text{Cost}(+, +), \text{Cost}(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。比较过程为：比较 $\text{Cost}(+, +)$ 和 $\text{Cost}(-, +)$ 的大小，进而可以确定出 Cost 最小的一个 BVD 的符号组合，即确定出候选 BVD 集合中的代价最小 BVD。

步骤 302、根据当前块的 BVD 符号信息和排序结果，确定当前块的 BVD 符号索引信息。

在本申请的实施例中，在根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果之后，可以进一步根据当前块的 BVD 符号信息和排序结果，从而可以确定当前块的 BVD 符号索引信息。

需要说明的是，在本申请的实施例中，如果排序处理获得的当前块的候选 BVD 列表为升（降）序的列表，那么可以选择采用转换列表间接得到 BVD 符号索引信息（符号预测索引），也可以直接采用逻辑推理得到 BVD 符号索引信息。

其中，在采用转换列表间接得到 BVD 符号索引信息时，可以先按照转换规则将 BVD 组合列表 *OrderedList* 转换为索引转换列表 *TransformedList*。

示例性的，在本申请的实施例中，可以对候选 BVD 列表进行转换处理，确定索引转换列表（*TransformedList*）。

进一步地，在本申请的实施例中，在根据当前块的 BVD 符号信息和排序结果，确定当前块的 BVD 符号索引信息时，当 absBvdX 和 absBvdY 均不为 0 时，若当前块的 BVD 符号信息的第一分量符号与索引转换列表中的第一个候选 BVD 的第一分量符号相同，则将 BVD 符号索引信息的第一比特位确定为 0；否则将第一比特位确定为 1。

进一步地，在本申请的实施例中，当第一比特位确定为 0 时，若当前块的 BVD 符号信息的第二分量符号与索引转换列表中的第一个候选 BVD 的第二分量符号相同，则将 BVD 符号索引信息的第二比特位确定为 0；否则将第二比特位确定为 1。

进一步地，在本申请的实施例中，当第一比特位确定为 1 时，若当前块的 BVD 符号信息的第二分量符号与索引转换列表中、索引为 1 的候选 BVD 的第二分量符号相同，则将 BVD 符号索引信息的第二比特位确定为 0；否则将第二比特位确定为 1。

也就是说，在本申请的实施例中，对于 BVD 水平分量和垂直分量均存在的情况，BVD 的符号预测索引为两个比特位。

假设排序后的 BVD 组合列表 *OrderedList* 为：

表 78. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 *OrderedList*

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

利用以下转换规则将排序后的 BVD 组合列表 *OrderedList* 转换为 *TransformedList*（索引转换列表）：

将排序后最小 cost 的 BVD 组合放在 *TransformedList* 列表的第 0 位，即将 (a1, a2) 放在 *TransformedList* 列表的第 0 位。将与 a1 同号的 BVD 组合放在 *TransformedList* 列表的第 2 位，即假设 b1 与其同号，则将 (b1, b2) 放在 *TransformedList* 列表的第 2 位。将与 a1 异号的较小 cost 的 BVD 组合放在 *TransformedList* 列表的第 1 位，即将 (c1, c2) 放在 *TransformedList* 列表的第 1 位。将与 a1 异号的较大 cost 的 BVD 组合放在 *TransformedList* 列表的第 3 位，即将 (d1, d2) 放在 *TransformedList* 列表的第 3 位。

则索引转换列表 *TransformedList* 为：

表 79. 水平垂直 BVD 均存在的索引转换列表 *TransformedList*

索引	水平分量	垂直分量
0	a1	a2
1	c1(-a1)	c2
2	b1(a1)	b2
3	d1(-a1)	d2

对于索引转换列表，按照以下规则得到符号预测索引，此时符号预测索引等于表中最左列的索引：

假设真实的 BVD 的水平分量为 e1，垂直分量为 e2。

针对索引 0 的候选 BVD：

如果满足水平分量 $a1 == e1$ ，则第 0 个比特位为 0，然后检查其垂直分量，如果满足垂直分量 $a2 == e2$ ，则第 1 个比特位为 0，否则第 1 个比特位为 1。

如果不满足水平分量 $a1 == e1$ ，则第 0 个比特位为 1，然后检查索引 1 的候选 BVD 的垂直分量，如果满足垂直分量 $c2 == e2$ ，则第 1 个比特位为 0，否则第 1 个比特位为 1。

符号预测索引表示如下：

$$bvsIdx = bit0 + bit1 \ll 1$$

例如：假设排序后的 BVD 组合列表 *OrderedList* 为：

表 80. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 *OrderedList* 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$-BVD_{HOR}$	$-BVD_{VER}$

2	$+BVD_{HOR}$	$+BVD_{VER}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

则按照上述转换规则将排序后的 BVD 组合列表 OrderedList 转换为 TransformedList:

表 81. 水平垂直 BVD 均存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$+BVD_{HOR}$	$+BVD_{VER}$
2	$-BVD_{HOR}$	$-BVD_{VER}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

(1) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$, 垂直分量为 $+BVD_{VER}$, 则:

$$bvslidx = 0$$

(2) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$, 垂直分量为 $-BVD_{VER}$, 则:

$$bvslidx = 2$$

(3) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$, 垂直分量为 $+BVD_{VER}$, 则:

$$bvslidx = 1$$

(4) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$, 垂直分量为 $-BVD_{VER}$, 则:

$$bvslidx = 3$$

进一步地, 在本申请的实施例中, 在根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息时, 当 $absBvdX$ 为 0 时, 若当前块的 BVD 符号信息的第二分量符号与索引转换列表中的第一个候选 BVD 的第二分量符号相同, 则将 BVD 符号索引信息确定为 0; 否则将 BVD 符号索引信息确定为 1。

也就是说, 如果仅 BVD 垂直分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 82. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

则索引转换列表 TransformedList 与排序后的 BVD 组合列表 OrderedList 相同。

索引转换列表 TransformedList 为:

表 6. 仅垂直 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

对于索引转换列表, 按照以下规则得到符号预测索引, 此时符号预测索引等于表中最左列的索引:

假设真实的 BVD 的水平分量为 0, 垂直分量为 $e2$ 。

如果 $a2 == e2$, 则第 0 个比特位为 0, 否则, 第 0 个比特位为 1。

符号预测索引表示如下:

$$bvslidx = bit0$$

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 83. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

索引转换列表 TransformedList 为:

表 8. 仅垂直 BVD 存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

(1) 假设真实的 BVD 的水平分量为 0, 垂直分量为 $+BVD_{VER}$, 则:

$$bvslidx = 0$$

(2) 假设真实的 BVD 的水平分量为 0, 垂直分量为 $-BVD_{VER}$, 则:

$$bvslidx = 1$$

进一步地, 在本申请的实施例中, 在根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息时, 当 $absBvdY$ 为 0 时, 若当前块的 BVD 符号信息的第一分量符号与索引转换列表中的第一个候选 BVD 的第一分量符号相同, 则将 BVD 符号索引信息确定为 0; 否则将 BVD 符号索引信息确定为 1。

也就是说, 如果仅 BVD 水平分量存在, 此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 84. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

则索引转换列表 TransformedList 与排序后的 BVD 组合列表 OrderedList 相同。

索引转换列表 TransformedList 为:

表 85. 仅水平 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

对于索引转换列表, 按照以下转换规则得到符号预测索引, 此时符号预测索引等于表中最左列的索引:

假设真实的 BVD 的水平分量为 e1, 垂直分量为 0。

如果 a1==e1, 则第 0 个比特位为 0, 否则, 第 0 个比特位为 1。

符号预测索引表示如下:

$$bvsdIdx = bit0$$

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 86. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	+BVD _{HOR}	0
1	-BVD _{HOR}	0

索引转换列表 TransformedList 为:

表 87. 仅水平 BVD 存在的转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	+BVD _{HOR}	0
1	-BVD _{HOR}	0

(1) 假设真实的 BVD 的水平分量为 +BVD_{HOR}, 垂直分量为 0, 则:

$$bvsdIdx = 0$$

(2) 假设真实的 BVD 的水平分量为 -BVD_{HOR}, 垂直分量为 0, 则:

$$bvsdIdx = 1$$

进一步地, 在直接采用逻辑推理得到 BVD 符号索引信息时, 如果 BVD 水平分量和垂直分量均存在, 那么在根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息时, 当 absBvdX 和 absBvdY 均不为 0 时, 若当前块的 BVD 符号信息的第一分量符号与与候选 BVD 列表中的第一个候选 BVD 的第一分量符号相同, 则将 BVD 符号索引信息的第一比特位确定为 0; 否则将第一比特位确定为 1; 遍历候选 BVD 列表, 确定与当前块的 BVD 符号信息具有相同第一分量符号的首个候选 BVD; 若当前块的 BVD 符号信息的第二分量符号与首个候选 BVD 的第二分量符号相同, 则将 BVD 符号索引信息的第二比特位确定为 0; 否则将第二比特位确定为 1。

由此可见, 第一种情况 BVD 水平分量和垂直分量均存在, 此时 BVD 的符号预测索引为两个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 88. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1	b2
2	c1	c2
3	d1	d2

假设真实的 BVD 的水平分量为 e1, 垂直分量为 e2。

针对索引 0 的候选 BVD, 如果满足水平分量 a1==e1, 则第 0 个比特位为 0, 否则, 第 0 个比特位为 1。然后从索引 0 开始遍历排序后的列表, 当列表中首次出现与真实 BVD 水平分量相同的 BVD 时, 检查其垂直分量。假设其垂直分量为 f2, 若满足 f2==e2, 则第 1 个比特位为 0, 否则第 1 个比特位为 1。符号预测索引表示如下:

$$bvsdIdx = bit0 + bit1 \ll 1$$

bit0 为第 0 个比特位, bit1 为第 1 个比特位。

例如: 假设排序后的 BVD 组合列表 OrderedList 为:

表 89. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	-BVD _{HOR}	+BVD _{VER}
1	-BVD _{HOR}	-BVD _{VER}
2	+BVD _{HOR}	+BVD _{VER}
3	+BVD _{HOR}	-BVD _{VER}

(1) 假设真实的 BVD 的水平分量为 -BVD_{HOR}, 垂直分量为 +BVD_{VER}。

则第 0 个比特位为 0, 第 1 个比特位为 0。符号预测索引表示如下:

$$bvsdIdx = bit0 + bit1 \ll 1 = 0$$

(2) 假设真实的 BVD 的水平分量为 -BVD_{HOR}, 垂直分量为 -BVD_{VER}。

则第 0 个比特位为 0, 第 1 个比特位为 1。符号预测索引表示如下:

$$bvsdIdx = bit0 + bit1 \ll 1 = 2$$

(3) 假设真实的 BVD 的水平分量为 +BVD_{HOR}, 垂直分量为 +BVD_{VER}。

则第 0 个比特位为 1, 第 1 个比特位为 0。符号预测索引表示如下:

$$bvsdIdx = bit0 + bit1 \ll 1 = 1$$

(4) 假设真实的 BVD 的水平分量为 +BVD_{HOR}, 垂直分量为 -BVD_{VER}。

则第 0 个比特位为 1, 第 1 个比特位为 1。符号预测索引表示如下:

$$bvsdIdx = bit0 + bit1 \ll 1 = 3$$

进一步地,在直接采用逻辑推理得到 BVD 符号索引信息时,如果仅 BVD 垂直分量存在,那么在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 absBvdX 为 0 时,若当前块的 BVD 符号信息的第二分量符号与候选 BVD 列表中的第一个候选 BVD 的第二分量符号相同,则将 BVD 符号索引信息确定为 0,否则将 BVD 符号索引信息确定为 1。

5 由此可见,第二种情况仅 BVD 的垂直分量存在,此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 90 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

假设真实的 BVD 的水平分量为 0,垂直分量为 e2。

如果 $a2==e2$,则第 0 个比特位为 0,否则,第 0 个比特位为 1。

10 符号预测索引表示如下:

$$bvsdIdx = bit0$$

例如:假设排序后的 BVD 组合列表 OrderedList 为:

表 91. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

(1) 假设真实的 BVD 的水平分量为 0,垂直分量为 $+BVD_{VER}$ 。

第 0 个比特位为 0。符号预测索引表示如下:

$$bvsdIdx = bit0 = 0$$

15 (2) 假设真实的 BVD 的水平分量为 0,垂直分量为 $-BVD_{VER}$ 。

第 0 个比特位为 1。符号预测索引表示如下:

$$bvsdIdx = bit0 = 1$$

进一步地,在直接采用逻辑推理得到 BVD 符号索引信息时,如果仅 BVD 水平分量存在,那么在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 absBvdY 为 0 时,若当前块的 BVD 符号信息的第一分量符号与候选 BVD 列表中的第一个候选 BVD 的第一分量符号相同,则将 BVD 符号索引信息确定为 0,否则将 BVD 符号索引信息确定为 1。

20 由此可见,第三种情况仅 BVD 的水平分量存在,此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 92. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	0
1	b1(-a1)	0

假设真实的 BVD 的水平分量为 e1,垂直分量为 0。

25 如果 $a1==e1$,则第 0 个比特位为 0,否则,第 0 个比特位为 1。

符号预测索引表示如下:

$$bvsdIdx = bit0$$

例如:

假设排序后的 BVD 组合列表 OrderedList 为:

表 93. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

30 (1) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$,垂直分量为 0。

第 0 个比特位为 0。符号预测索引表示如下:

$$bvsdIdx = bit0 = 0$$

(2) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$,垂直分量为 0。

第 0 个比特位为 1。符号预测索引表示如下:

$$bvsdIdx = bit0 = 1$$

需要说明的是,在本申请的实施例中,如果排序处理获得的当前块的候选 BVD 列表为特定顺序的列表,那么可以直接将与真实的 BVD 的水平和垂直分量的符号对应相等的候选 BVD 的列表索引,确定为 BVD 的 BVD 符号索引信息(符号预测索引)。其中,可以将排序后的 BVD 组合列表 OrderedList,直接赋值给 TransformedList,即候选 BVD 列表 OrderedList 和索引转换列表 TransformedList 是相同的。

可以理解的是,在本申请的实施例中,如果排序处理获得的当前块的候选 BVD 列表为特定顺序的列表,那么在编码端,可以直接对 BVD 对应的两个分量同时编码,对应的,在解码端,BVD 对应的两个分量可以直接通过 BVD 符号索引信息一起确定出来。

40 也就是说,在本申请的实施例中,可以直接通过解码获得的 BVD 符号索引信息,从候选 BVD 列表中确定出 BVD 符号索引信息所指示的候选 BVD。

相应的,在本申请的实施例中,对于解码端解析组合索引(BVD 符号索引信息)后同时得出水平和垂直分量的方案,

在编码端，编码器可以采用遍历的方法决定所传索引（BVD 符号索引信息）。

可以理解的是，在本申请的实施例中，对于按照第一代价值指示的匹配误差从小到大的顺序排序所确定的候选 BVD 列表，即在当前块的候选 BVD 列表为升（降）序的列表的情况下，也可以直接按照与当前块的 BVD 符号信息相同的候选 BVD 在列表中的索引值设置 BVD 符号索引信息。

- 5 示例性的，在本申请的实施例中，在根据当前块的 BVD 符号信息和排序结果，确定当前块的 BVD 符号索引信息时，可以在候选 BVD 列表中确定与当前块的 BVD 符号信息相同的候选 BVD；然后按照候选 BVD 的索引值设置 BVD 符号索引信息。

第一种情况 BVD 水平分量和垂直分量均存在，此时 BVD 的符号预测索引为两个比特位：

假设排序后的 BVD 组合列表 OrderedList 为：

10 表 94. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2
2	c1(a1)	c2
3	d1(-a1)	d2

则索引转换列表 TransformedList 为：

表 95. 水平垂直 BVD 均存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a1	a2
1	b1(-a1)	b2
2	c1(a1)	c2
3	d1(-a1)	d2

对于索引转换列表，按照以下规则得到符号预测索引，此时符号预测索引等于表中最左列的索引：

假设真实的 BVD 的水平分量为 e1，垂直分量为 e2。

- 15 针对索引 0 的候选 BVD：

如果满足水平分量 $a1 == e1$ ，则第 0 个比特位为 0，然后检查其垂直分量，如果满足垂直分量 $a2 == e2$ ，则第 1 个比特位为 0，否则第 1 个比特位为 1。

如果不满足水平分量 $a1 == e1$ ，则第 0 个比特位为 1，然后检查索引 1 的候选 BVD 的垂直分量，如果满足垂直分量 $c2 == e2$ ，则第 1 个比特位为 0，否则第 1 个比特位为 1。

- 20 符号预测索引表示如下：

$$bvIdx = bit0 + bit1 \ll 1$$

例如：假设排序后的 BVD 组合列表 OrderedList 为：

表 96. 水平垂直 BVD 均存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$+BVD_{HOR}$	$+BVD_{VER}$
2	$-BVD_{HOR}$	$-BVD_{HOR}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

则 TransformedList 为：

表 97. 水平垂直 BVD 均存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$-BVD_{HOR}$	$+BVD_{VER}$
1	$+BVD_{HOR}$	$+BVD_{VER}$
2	$-BVD_{HOR}$	$-BVD_{HOR}$
3	$+BVD_{HOR}$	$-BVD_{VER}$

- 25 (1) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ ，则：

$$bvIdx = 0$$

- (2) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ ，则：

$$bvIdx = 2$$

- (3) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$ ，垂直分量为 $+BVD_{VER}$ ，则：

$$bvIdx = 1$$

- (4) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$ ，垂直分量为 $-BVD_{VER}$ ，则：

$$bvIdx = 3$$

第二种情况仅 BVD 垂直分量存在，此时 BVD 的符号预测索引为一个比特位：

- 30 假设排序后的 BVD 组合列表 OrderedList 为：

表 98. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	0	a2
1	0	b2(-a2)

则索引转换列表 TransformedList 为：

表 99. 仅垂直 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
----	------	------

0	0	a2
1	0	b2(-a2)

对于索引转换列表,按照以下规则得到符号预测索引,此时符号预测索引等于表中最左列的索引:

假设真实的 BVD 的水平分量为 0,垂直分量为 e_2 。

如果 $a_2 == e_2$,则第 0 个比特位为 0,否则,第 0 个比特位为 1。

符号预测索引表示如下:

$$bvslidx = bit0$$

5 例如:假设排序后的 BVD 组合列表 OrderedList 为:

表 100. 仅垂直 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

TransformedList 为:

表 101. 仅垂直 BVD 存在的索引转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	0	$+BVD_{VER}$
1	0	$-BVD_{VER}$

(1) 假设真实的 BVD 的水平分量为 0,垂直分量为 $+BVD_{VER}$,则:

$$bvslidx = 0$$

10 (2) 假设真实的 BVD 的水平分量为 0,垂直分量为 $-BVD_{VER}$,则:

$$bvslidx = 1$$

第三种情况仅 BVD 水平分量存在,此时 BVD 的符号预测索引为一个比特位:

假设排序后的 BVD 组合列表 OrderedList 为:

表 102. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList

索引	水平分量	垂直分量
0	a_1	0
1	$b_1(-a_1)$	0

则索引转换列表 TransformedList 为:

表 103. 仅水平 BVD 存在的索引转换列表 TransformedList

索引	水平分量	垂直分量
0	a_1	0
1	$b_1(-a_1)$	0

对于索引转换列表,按照以下规则得到符号预测索引,此时符号预测索引等于表中最左列的索引:

假设真实的 BVD 的水平分量为 e_1 ,垂直分量为 0。

如果 $a_1 == e_1$,则第 0 个比特位为 0,否则,第 0 个比特位为 1。

符号预测索引表示如下:

$$bvslidx = bit0$$

20 例如:假设排序后的 BVD 组合列表 OrderedList 为:

表 104. 仅水平 BVD 存在的排序后的 BVD 组合列表 OrderedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

TransformedList 为:

表 105. 仅水平 BVD 存在的转换列表 TransformedList 实例

索引	水平分量	垂直分量
0	$+BVD_{HOR}$	0
1	$-BVD_{HOR}$	0

(1) 假设真实的 BVD 的水平分量为 $+BVD_{HOR}$,垂直分量为 0,则:

$$bvslidx = 0$$

25 (2) 假设真实的 BVD 的水平分量为 $-BVD_{HOR}$,垂直分量为 0,则:

$$bvslidx = 1$$

需要说明的是,在本申请的实施例中,如果排序处理后的排序结果为当前块的候选 BVD 集合,那么排序后即确定了模板匹配代价最小的候选 BVD,即确定了候选 BVD 集合中的代价最小 BVD。此时,如果 $bvslidx$ 的数值是 $bit0+bit1 \ll 1$,即 $bit0$ 代表水平分量符号, $bit1$ 代表垂直分量符号,那么,最小的 BVD 符号组合(代价最小 BVD),确定了 $bit0$ 和 $bit1$ 在取值等于 0 时的符号,取值等于 1 时的符号与此相反。

30 进一步地,在本申请的实施例中,在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 $absBvdX$ 和 $absBvdY$ 均不为 0 时,若当前块的 BVD 符号信息的第一分量符号与候选 BVD 集合中的代价最小 BVD 的第一分量符号相同,则确定 BVD 符号索引信息的第一比特位取值为 0;接着可以按照候选 BVD 集合中的代价最小 BVD 确定当前块的 BVD 符号索引信息的第二比特位。

35 进一步地,在本申请的实施例中,在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 $absBvdX$ 和 $absBvdY$ 均不为 0 时,若当前块的 BVD 符号信息的第一分量符号与候选 BVD 集合中的代价最小 BVD 的第一分量符号不同,则确定 BVD 符号索引信息的第一比特位取值为 1;接着可以按照候选 BVD 集合中的代价最小 BVD

所在的数组以外的、另一个数组中的代价最小 BVD 确定当前块的 BVD 符号索引信息的第二比特位。

需要说明的是,在本申请的实施例中,在按照候选 BVD 集合中的代价最小 BVD 确定当前块的 BVD 符号索引信息的第二比特位时,若当前块的 BVD 符号信息的第二分量符号与候选 BVD 集合中的代价最小 BVD 的第二分量符号相同,则确定 BVD 符号索引信息的第二比特位取值为 0;若当前块的 BVD 符号信息的第二分量符号与候选 BVD 集合中的代

5 价最小 BVD 的第二分量符号不同,则确定 BVD 符号索引信息的第二比特位取值为 1。
需要说明的是,在本申请的实施例中,在按照候选 BVD 集合中的代价最小 BVD 所在的数组以外的、另一个数组中的代价最小 BVD 确定当前块的 BVD 符号索引信息的第二比特位时,若当前块的 BVD 符号信息的第二分量符号与另一个数组中的代价最小 BVD 的第二分量符号相同,则确定 BVD 符号索引信息的第二比特位取值为 0;若当前块的 BVD 符号信息的第二分量符号与另一个数组中的代价最小 BVD 的第二分量符号不同,则确定 BVD 符号索引信息的第二比特

10 位取值为 1。
需要说明的是,在本申请的实施例中,假设真实 BVD 的水平分量和垂直分量的符号为 $bvdSign[0]$ 和 $bvdSign[1]$ 。其中,取值为 0 则为正,取值为 1 则为负。

示例性的,在本申请的实施例中,如果 BVD 水平和垂直分量均存在,对于排序后确定的候选 BVD 集合中的代价最小 BVD 的两种可能情况如下:

15 (1) 若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negaGrp[0]$ 对应的 $Cost$:

如果 $bvd_sign[0] = 0$, 那么第 0 个比特位为 0。然后,判断 $bvd_sign[1]$ 是否等于 $posiGrp[0][1]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

如果 $bvd_sign[0] = 1$, 那么第 0 个比特位为 1。然后,判断 $bvd_sign[1]$ 是否等于 $negagr[0][1]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

20 (2) 若在第三次比较中, $posiGrp[0]$ 对应的 $Cost > negaGrp[0]$ 对应的 $Cost$:

如果 $bvd_sign[0] = 0$, 那么第 0 个比特位为 1。然后,判断 $bvd_sign[1]$ 是否等于 $posiGrp[0][1]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

如果 $bvd_sign[0] = 1$, 那么第 0 个比特位为 0。然后,判断 $bvd_sign[1]$ 是否等于 $negagr[0][1]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

25 进一步地,在本申请的实施例中,在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 $absBvdX$ 为 0 时,若当前块的 BVD 符号信息的第二分量符号与候选 BVD 集合中的代价最小 BVD 的第二分量符号相同,则确定 BVD 符号索引信息的取值为 0; 否则确定 BVD 符号索引信息的取值为 1。

示例性的,在本申请的实施例中,如果 BVD 仅垂直分量存在,对于排序后确定的候选 BVD 集合中的代价最小 BVD 的两种可能情况如下:

30 (1) 若 $Cost(+, +) \leq Cost(+, -)$:

如果 $bvd_sign[1] = 0$, 那么第 0 个比特位为 0。

如果 $bvd_sign[1] = 1$, 那么第 0 个比特位为 1。

(2) 若 $Cost(+, +) > Cost(+, -)$:

如果 $bvd_sign[1] = 0$, 那么第 0 个比特位为 1。

35 如果 $bvd_sign[1] = 1$, 那么第 0 个比特位为 0。

进一步地,在本申请的实施例中,在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息时,当 $absBvdY$ 为 0 时,若当前块的 BVD 符号信息的第一分量符号与候选 BVD 集合中的代价最小 BVD 的第一分量符号相同,则确定 BVD 符号索引信息的取值为 0; 否则确定 BVD 符号索引信息的取值为 1。

示例性的,在本申请的实施例中,如果 BVD 仅水平分量存在,对于排序后确定的候选 BVD 集合中的代价最小 BVD 的两种可能情况如下:

40 (1) 若 $Cost(+, +) \leq Cost(-, +)$:

如果 $bvd_sign[0] = 0$, 那么第 0 个比特位为 0。

如果 $bvd_sign[0] = 1$, 那么第 0 个比特位为 1。

(2) 若 $Cost(+, +) > Cost(-, +)$:

45 如果 $bvd_sign[0] = 0$, 那么第 0 个比特位为 1。

如果 $bvd_sign[0] = 1$, 那么第 0 个比特位为 0。

可见,上述基于候选 BVD 集合确定当前块的 BVD 符号索引信息的方法,可以省去建立列表和得到符号预测索引的步骤,对于 BVD 的水平和垂直分量均存在的情况,编码端直接通过三次比较得到要编码的 bit0 和 bit1,复杂度更低。

步骤 303、将 BVD 绝对值和 BVD 符号索引信息写入码流。

50 在本申请的实施例中,在根据当前块的 BVD 符号信息和排序结果,确定当前块的 BVD 符号索引信息之后,便可以将 BVD 绝对值和 BVD 符号索引信息写入码流。

进一步地,在本申请的实施例中,可以选择使用上下文自适应的二进制算术编码 CABAC 编码 BVD 绝对值和 BVD 符号索引信息。也可以使用旁路模式编码 BVD 绝对值和 BVD 符号索引信息。

需要说明的是,在本申请的实施例中,需要将 BVD 绝对值和 BVD 符号索引信息(符号预测索引)写入码流。其中,55 可以先将 BVD 符号索引信息(符号预测索引)二值化。

示例性的,在本申请的实施例中,符号预测索引为两个比特位时,其为 0 时,编码为 00B; 其为 1 时,编码为 01B; 其为 2 时,编码为 10B; 其为 3 时,编码为 11B。符号预测索引为一个比特位时,其为 0 时,编码为 0B; 其为 1 时,编码为 1B。

然后,可以使用 CABAC(上下文自适应的二进制算术编码)或旁路对上述转换的二进制数进行每个比特位的编码,60 即将 BVD 的符号预测索引写入码流中。

针对符号预测索引所使用的 CABAC 的上下文模型,可以使用单个概率模型,或者多个概率模型:包括但不限于根

据 BVD 的水平分量或垂直分量的绝对值大小、BVD 的水平分量与垂直分量的和或差值等进行区分不同的概率模型。

例如, 根据 BVD 的水平分量或垂直分量的绝对值大小对 BVD 的符号预测索引的比特位使用不同的概率模型:

(1) 假设 BVD 的水平或垂直分量的绝对值分类的一个阈值设为 THR_0 , 共有两个概率模型, 则当符号预测索引为两个比特位时, 此二进制数的第 0 个比特位根据 BVD 的水平分量的绝对值与 THR_0 的大小关系进行区分, 即当 BVD 的水平分量的绝对值小于等于 THR_0 时, 使用第一个概率模型, 当 BVD 的水平分量的绝对值大于 THR_0 时, 使用第二个概率模型; 然后, 上述二进制数的第 1 个比特位也根据 BVD 的垂直分量的绝对值与 THR_0 的大小关系进行区分, 即当 BVD 的垂直分量的绝对值小于等于 THR_0 时, 使用第一个概率模型, 当 BVD 的垂直分量的绝对值大于 THR_0 时, 使用第二个概率模型。

(2) 假设 BVD 的水平分量的绝对值分类的一个阈值设为 THR_0 , 垂直分量的绝对值分类的一个阈值设为 THR_1 , 共有四个概率模型, 则当符号预测索引为两个比特位时, 此二进制数的第 0 个比特位根据 BVD 的水平分量的绝对值与 THR_0 的大小关系进行区分, 即当 BVD 的水平分量的绝对值小于等于 THR_0 时, 使用第一个概率模型, 当 BVD 的水平分量的绝对值大于 THR_0 时, 使用第二个概率模型; 然后, 上述二进制数的第 1 个比特位也根据 BVD 的垂直分量的绝对值与 THR_1 的大小关系进行区分, 即当 BVD 的垂直分量的绝对值小于等于 THR_1 时, 使用第三个概率模型, 当 BVD 的垂直分量的绝对值大于 THR_1 时, 使用第四个概率模型。

综上所述, 通过上述步骤 301 至步骤 303 所提出的编码方法, 通过在 BVD 的可能符号和 BVD 的绝对值之间创建 BVD 的组合列表, 并对组合列表利用模板进行排序, 利用真实的 BVD 符号和排序后的列表得到符号预测索引进行 CABAC 的编码, 相较于比旁路编码具有更高的编码效率, 从而节省了 IBC 中的 BVD 传输所需的码率。

需要说明的是, 本申请实施例提出的编码方法, 一方面, 可以充分利用模板与当前编码块的相似性, 将 BVD 的符号组合列表进行有效地排序; 另一方面, 可以使用 CABAC 进行编码, 比旁路编码具有更高的编码效率。

进一步地, 本申请的实施例提出的编码方法, 对编码器而言, 在进行编码处理时, BVD 编码输入可以为 BVD 绝对值和符号的真实信息。而 BVD 编码输出可以为 BVD 绝对值和符号 (BVD 符号索引信息) 的码流信息。

其中, 图 13 为本申请实施例中的编码方法的示意图二, 如图 13 所示, BVD 编码过程包含编码 BVD 的绝对值 (步骤 401)、通过在 BVD 的可能符号和 BVD 的绝对值之间创建 BVD 的组合列表, 即确定当前块的候选 BVD (步骤 402)、根据当前块的 BVP 和 BVD 的组合列表进行模板预测, 计算出 BVD 组合列表中的每个 BVD 组合的成本, 即对组合列表计算模板区域的成本 (第一代价值) (步骤 403)、对列表排序并利用真实的 BVD 符号和排序后的列表得到符号预测索引 (BVD 符号索引信息) (步骤 404) 以及最后将 BVD 的符号预测索引 (BVD 符号索引信息) 进行编码 (步骤 405)。

需要说明的是, 在本申请的实施例中, 在计算成本 (第一代价值) 时, 不仅可以采用亮度, 还可以包含色度即 Cb 和 Cr 分量。本申请不进行具体限定。

进一步地, 在本申请的实施例中, 在计算成本时, 如果两个模板均能获取到, 此时对这个模板进行处理, 包括但不限于以下方法: 将两者进行加权融合, 其中, 加权融合的标准可以选择固定加权系数或者根据各自模板的重建亮度与预测亮度的像素值之差的绝对值之和进行分配加权系数等。

进一步地, 在本申请的实施例中, 针对排序后的 BVD 组合列表 (候选 BVD 列表), 可以存储它们各自的 cost, 然后基于这些存储的 cost 进行一个重新排序, 具体过程为: 如果列表中一个候选与前一个候选之间的 cost 差异低于 λ 值, 例如 $|D1-D2| < \lambda$, 则认为该候选是冗余的, 其中 D1 和 D2 是在第一次排序后获得的 cost, λ 是编码器端 RD 准则中使用的拉格朗日参数。

示例性的, 计算成本的算法定义如下:

(1) 确定列表所有候选中候选与其之前候选之间的最小成本差异

如果最小成本差异大于等于 λ , 则认为列表足够多样化并且停止重排。

如果这个最小成本差异小于 λ , 则候选被认为是多余的, 并将其移动到另外一个位置。这个位置是和其之前的候选相比足够多样化的一个位置。

(2) 该算法在有限次数的迭代后停止。

进一步地, 在本申请的实施例中, 对于上述实施例在采用特定顺序的候选 BVD 列表 OrderedList 和确定候选 BVD 集合中所提出的使用两个数组分别存储候选 BVD 的方案, 其中, BVD 水平垂直分量均存在的情况下, 水平分量和垂直分量的先后顺序可进行互换。

示例性的, 在本申请的实施例中, 假设该组合列表为 $\{(+, +), (+, -), (-, -), (-, +)\}$, 其对应的成本列表为 $\{Cost(+, +), Cost(+, -), Cost(-, -), Cost(-, +)\}$ 。上述两个列表中成员的顺序可任意排列。

定义两个数组 posiGrp 和 negaGrp, posiGrp 存储在同一分量 (垂直) 上符号为正号的两个 BVD 的符号组合, negaGrp 存储在同一分量 (垂直) 上符号为负号的两个 BVD 的符号组合。

假设 $posiGrp = \{(+, +), (-, +)\}$, $negagr = \{(+, -), (-, -)\}$ 。posiGrp 和 negaGrp 两个列表中成员的顺序可任意排列。

第一次比较: 比较 $Cost(+, +)$ 和 $Cost(-, +)$ 的大小, 将 $posiGrp[0]$ 置为两者较小的 BVD 组合, 将 $posiGrp[1]$ 置为两者较大的 BVD 组合;

第二次比较: 比较 $Cost(+, -)$ 和 $Cost(-, -)$ 的大小, 将 $negagr[0]$ 置为两者较小的 BVD 组合, 将 $negagr[1]$ 置为两者较大的 BVD 组合;

第三次比较: 比较 $posiGrp[0]$ 和 $negagr[0]$ 对应的 Cost 的大小;

(1) 若在第三次比较中, $posiGrp[0]$ 对应的 $Cost \leq negagr[0]$ 对应的 Cost:

如果 $bvd_sign[1] = 0$, 那么第 0 个比特位为 0。然后, 判断 $bvd_sign[0]$ 是否等于 $posiGrp[0][0]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

如果 $bvd_sign[1] = 1$, 那么第 0 个比特位为 1。然后, 判断 $bvd_sign[0]$ 是否等于 $negagr[0][0]$, 如果相等则第 1 个比特位为 0, 否则第 1 个比特位为 1。

(2) 若在第三次比较中, $posiGrp[0]$ 对应的 $Cost > negagr[0]$ 对应的 Cost:

如果 $bvd_sign[1] = 0$ ，那么第 0 个比特位为 1。然后，判断 $bvd_sign[0]$ 是否等于 $posiGrp[0][0]$ ，如果相等则第 1 个比特位为 0，否则第 1 个比特位为 1。

如果 $bvd_sign[1] = 1$ ，那么第 0 个比特位为 0。然后，判断 $bvd_sign[0]$ 是否等于 $negGrp[0][0]$ ，如果相等则第 1 个比特位为 0，否则第 1 个比特位为 1。

进一步地，在本申请的实施例中，在 BVD 水平垂直分量均存在的情况下，水平分量和垂直分量的先后顺序可进行互换的操作流程，可以通过某种准则来决策水平分量和垂直分量的先后顺序。其中包括但不限于以下的方式，通过 BVD 的水平分量和垂直分量的绝对值大小等准则来决策水平分量和垂直分量的先后顺序。

本申请实施例提供了一种编解码方法，在解码端，解码码流，确定当前块的 BVD 绝对值和 BVD 符号索引信息；根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果；根据 BVD 符号索引信息和排序结果，确定当前块的 BVD；根据当前块的 BVD，确定当前块的重建值。在编码端，根据当前块的 BVD 绝对值，对当前块的候选 BVD 进行排序处理，确定排序结果；根据当前块的 BVD 符号信息和排序结果，确定当前块的 BVD 符号索引信息；将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见，在本申请的实施例中，可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理，从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输，在解码时利用排序结果进行当前块的 BVD 符号信息的解析，同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理，能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率，从而可以提高编码效率。

基于上述实施例，在本申请的再一实施例中，基于前述实施例相同的发明构思，图 14 为编码器的组成结构示意图一，如图 14 所示，编码器 110 可以包括：第一确定单元 111，编码单元 112，其中，

所述第一确定单元 111，配置为根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；根据所述当前块的 BVD 符号信息和所述排序结果，确定所述当前块的 BVD 符号索引信息；

所述编码单元 112，配置为将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

可以理解地，在本实施例中，“单元”可以是部分电路、部分处理器、部分程序或软件等等，当然也可以是模块，还可以是非模块化的。而且在本实施例中的各组成部分可以集成在一个处理单元中，也可以是各个单元单独物理存在，也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现，也可以采用软件功能模块的形式实现。

所述集成的单元如果以软件功能模块的形式实现并非作为独立的产品进行销售或使用，可以存储在一个计算机可读存储介质中，基于这样的理解，本实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来，该计算机软件产品存储在一个存储介质中，包括若干指令用以使得一台计算机设备（可以是个人计算机，服务器，或者网络设备等）或 processor（处理器）执行本实施例所述方法的全部或部分步骤。而前述的存储介质包括：U 盘、移动硬盘、只读存储器（Read Only Memory, ROM）、随机存取存储器（Random Access Memory, RAM）、磁碟或者光盘等各种可以存储程序代码的介质。

因此，本申请实施例提供了一种计算机可读存储介质，应用于编码器 110，该计算机可读存储介质存储有计算机程序，所述计算机程序被第一处理器执行时实现前述实施例中任一项所述的方法。

基于上述编码器 110 的组成以及计算机可读存储介质，图 15 为编码器的组成结构示意图二，如图 15 所示，编码器 110 可以包括：第一存储器 113 和第一处理器 114，第一通信接口 115 和第一总线系统 116。第一存储器 113、第一处理器 114、第一通信接口 115 通过第一总线系统 116 耦合在一起。可理解，第一总线系统 116 用于实现这些组件之间的连接通信。第一总线系统 116 除包括数据总线之外，还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见，在图 10 中将各种总线都标为第一总线系统 116。其中，

第一通信接口 115，用于在与其他外部网元之间进行收发信息过程中，信号的接收和发送；

所述第一存储器 113，用于存储能够在所述第一处理器上运行的计算机程序；

所述第一处理器 114，用于在运行所述计算机程序时，根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；根据所述当前块的 BVD 符号信息和所述排序结果，确定所述当前块的 BVD 符号索引信息；将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

可以理解，本申请实施例中的第一存储器 113 可以是易失性存储器或非易失性存储器，或可包括易失性和非易失性存储器两者。其中，非易失性存储器可以是只读存储器（Read-Only Memory, ROM）、可编程只读存储器（Programmable ROM, PROM）、可擦除可编程只读存储器（Erasable PROM, EPROM）、电可擦除可编程只读存储器（Electrically EPROM, EEPROM）或闪存。易失性存储器可以是随机存取存储器（Random Access Memory, RAM），其用作外部高速缓存。通过示例性但不是限制性说明，许多形式的 RAM 可用，例如静态随机存取存储器（Static RAM, SRAM）、动态随机存取存储器（Dynamic RAM, DRAM）、同步动态随机存取存储器（Synchronous DRAM, SDRAM）、双倍数据速率同步动态随机存取存储器（Double Data Rate SDRAM, DDRSDRAM）、增强型同步动态随机存取存储器（Enhanced SDRAM, ESDRAM）、同步连接动态随机存取存储器（Synchlink DRAM, SLDRAM）和直接内存总线随机存取存储器（Direct Rambus RAM, DRRAM）。本申请描述的系统和方法的第一存储器 113 旨在包括但不限于这些和任意其它适合类型的存储器。

而第一处理器 114 可能是一种集成电路芯片，具有信号的处理能力。在实现过程中，上述方法的各步骤可以通过第一处理器 114 中的硬件的集成逻辑电路或者软件形式的指令完成。上述的第一处理器 114 可以是通用处理器、数字信号处理器（Digital Signal Processor, DSP）、专用集成电路（Application Specific Integrated Circuit, ASIC）、现成可编程门阵列（Field Programmable Gate Array, FPGA）或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成，或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器，闪存、只读存储器，可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于第一存储器 113，第一处理器 114 读取第一存储器 113 中的信息，结合其硬件完成上述方法的步骤。

可以理解的是，本申请描述的这些实施例可以用硬件、软件、固件、中间件、微码或其组合来实现。对于硬件实现，

处理单元可以实现在一个或多个专用集成电路 (Application Specific Integrated Circuits, ASIC)、数字信号处理器 (Digital Signal Processing, DSP)、数字信号处理设备 (DSP Device, DSPD)、可编程逻辑设备 (Programmable Logic Device, PLD)、现场可编程门阵列 (Field-Programmable Gate Array, FPGA)、通用处理器、控制器、微控制器、微处理器、用于执行本申请所述功能的其它电子单元或其组合中。对于软件实现, 可通过执行本申请所述功能的模块 (例如过程、函数等) 来实现本申请所述的技术。软件代码可存储在存储器中并通过处理器执行。存储器可以在处理器中或在处理器外部实现。

可选地, 作为另一个实施例, 第一处理器 114 还配置为在运行所述计算机程序时, 执行前述实施例中任一项所述的方法。

图 16 为解码器的组成结构示意图一, 如图 16 所示, 解码器 120 可以包括: 解码单元 121 和第二确定单元 122; 其中,

所述解码单元 121, 配置为解码码流;

所述第二确定单元 122, 配置为确定当前块的 BVD 绝对值和 BVD 符号索引信息;

根据当前块的 BVD 绝对值, 对所述当前块的候选 BVD 进行排序处理, 确定排序结果; 根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的 BVD; 根据所述当前块的 BVD, 确定所述当前块的重建值。

可以理解地, 在本实施例中, “单元”可以是部分电路、部分处理器、部分程序或软件等等, 当然也可以是模块, 还可以是非模块化的。而且在本实施例中的各组成部分可以集成在一个处理单元中, 也可以是各个单元单独物理存在, 也可以两个或两个以上单元集成在一个单元中。上述集成的单元既可以采用硬件的形式实现, 也可以采用软件功能模块的形式实现。

所述集成的单元如果以软件功能模块的形式实现并非作为独立的产品进行销售或使用时, 可以存储在一个计算机可读存储介质中, 基于这样的理解, 本实施例的技术方案本质上或者说对现有技术做出贡献的部分或者该技术方案的全部或部分可以以软件产品的形式体现出来, 该计算机软件产品存储在一个存储介质中, 包括若干指令用以使得一台计算机设备 (可以是个人计算机, 服务器, 或者网络设备等) 或 processor (处理器) 执行本实施例所述方法的全部或部分步骤。而前述的存储介质包括: U 盘、移动硬盘、只读存储器 (Read Only Memory, ROM)、随机存取存储器 (Random Access Memory, RAM)、磁碟或者光盘等各种可以存储程序代码的介质。

因此, 本申请实施例提供了一种计算机可读存储介质, 应用于解码器 120, 该计算机可读存储介质存储有计算机程序, 所述计算机程序被第一处理器执行时实现前述实施例中任一项所述的方法。

基于上述解码器 120 的组成以及计算机可读存储介质, 图 17 为解码器的组成结构示意图二, 如图 17 所示, 解码器 120 可以包括: 第二存储器 123 和第二处理器 124, 第二通信接口 125 和第二总线系统 126。第二存储器 123 和第二处理器 124, 第二通信接口 125 通过第二总线系统 126 耦合在一起。可理解, 第二总线系统 126 用于实现这些组件之间的连接通信。第二总线系统 126 除包括数据总线之外, 还包括电源总线、控制总线和状态信号总线。但是为了清楚说明起见, 在图 12 中将各种总线都标为第二总线系统 126。其中,

第二通信接口 125, 用于在与其他外部网元之间进行收发信息过程中, 信号的接收和发送;

所述第二存储器 123, 用于存储能够在所述第二处理器上运行的计算机程序;

所述第二处理器 124, 用于在运行所述计算机程序时, 解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息; 根据当前块的 BVD 绝对值, 对所述当前块的候选 BVD 进行排序处理, 确定排序结果; 根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的 BVD; 根据所述当前块的 BVD, 确定所述当前块的重建值。

可以理解, 本申请实施例中的第二存储器 123 可以是易失性存储器或非易失性存储器, 或可包括易失性和非易失性存储器两者。其中, 非易失性存储器可以是只读存储器 (Read-Only Memory, ROM)、可编程只读存储器 (Programmable ROM, PROM)、可擦除可编程只读存储器 (Erasable PROM, EPROM)、电可擦除可编程只读存储器 (Electrically EPROM, EEPROM) 或闪存。易失性存储器可以是随机存取存储器 (Random Access Memory, RAM), 其用作外部高速缓存。通过示例性但不是限制性说明, 许多形式的 RAM 可用, 例如静态随机存取存储器 (Static RAM, SRAM)、动态随机存取存储器 (Dynamic RAM, DRAM)、同步动态随机存取存储器 (Synchronous DRAM, SDRAM)、双倍数据速率同步动态随机存取存储器 (Double Data Rate SDRAM, DDRSDRAM)、增强型同步动态随机存取存储器 (Enhanced SDRAM, ESDRAM)、同步连接动态随机存取存储器 (Synchlink DRAM, SLDRAM) 和直接内存总线随机存取存储器 (Direct Rambus RAM, DRRAM)。本申请描述的系统和方法的第二存储器 123 旨在包括但不限于这些和任意其它适合类型的存储器。

而第二处理器 124 可能是一种集成电路芯片, 具有信号的处理能力。在实现过程中, 上述方法的各步骤可以通过第二处理器 124 中的硬件的集成逻辑电路或者软件形式的指令完成。上述的第二处理器 124 可以是通用处理器、数字信号处理器 (Digital Signal Processor, DSP)、专用集成电路 (Application Specific Integrated Circuit, ASIC)、现成可编程门阵列 (Field Programmable Gate Array, FPGA) 或者其他可编程逻辑器件、分立门或者晶体管逻辑器件、分立硬件组件。可以实现或者执行本申请实施例中的公开的各方法、步骤及逻辑框图。通用处理器可以是微处理器或者该处理器也可以是任何常规的处理器等。结合本申请实施例所公开的方法的步骤可以直接体现为硬件译码处理器执行完成, 或者用译码处理器中的硬件及软件模块组合执行完成。软件模块可以位于随机存储器, 闪存、只读存储器, 可编程只读存储器或者电可擦写可编程存储器、寄存器等本领域成熟的存储介质中。该存储介质位于第二存储器 123, 第二处理器 124 读取第二存储器 123 中的信息, 结合其硬件完成上述方法的步骤。

可以理解的是, 本申请描述的这些实施例可以用硬件、软件、固件、中间件、微码或其组合来实现。对于硬件实现, 处理单元可以实现在一个或多个专用集成电路 (Application Specific Integrated Circuits, ASIC)、数字信号处理器 (Digital Signal Processing, DSP)、数字信号处理设备 (DSP Device, DSPD)、可编程逻辑设备 (Programmable Logic Device, PLD)、现场可编程门阵列 (Field-Programmable Gate Array, FPGA)、通用处理器、控制器、微控制器、微处理器、用于执行本申请所述功能的其它电子单元或其组合中。对于软件实现, 可通过执行本申请所述功能的模块 (例如过程、函数等) 来实现本申请所述的技术。软件代码可存储在存储器中并通过处理器执行。存储器可以在处理器中或在处理器外部实现。

本申请实施例提供了一种编码器和解码器, 在解码端, 解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息; 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据 BVD 符号索引信息和排序结果,

确定当前块的 BVD; 根据当前块的 BVD, 确定当前块的重建值。在编码端, 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息; 将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见, 在本申请的实施例中, 可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理, 从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输, 在解码时利用排序结果进行当前块的 BVD 符号信息的解析, 同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理, 能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率, 从而可以提高编码效率。

需要说明的是, 在本申请的实施例中, 术语“包括”、“包含”或者其任何其他变体意在涵盖非排他性的包含, 从而使得包括一系列要素的过程、方法、物品或者装置不仅包括那些要素, 而且还包括没有明确列出的其他要素, 或者是还包括为这种过程、方法、物品或者装置所固有的要素。在没有更多限制的情况下, 由语句“包括一个……”限定的要素, 并不排除在包括该要素的过程、方法、物品或者装置中还存在另外的相同要素。

上述本申请实施例序号仅仅为了描述, 不代表实施例的优劣。

本申请所提供的几个方法实施例中所揭露的方法, 在不冲突的情况下可以任意组合, 得到新的方法实施例。

本申请所提供的几个产品实施例中所揭露的特征, 在不冲突的情况下可以任意组合, 得到新的产品实施例。

本申请所提供的几个方法或设备实施例中所揭露的特征, 在不冲突的情况下可以任意组合, 得到新的方法实施例或设备实施例。

以上所述, 仅为本申请的具体实施方式, 但本申请的保护范围并不局限于此, 任何熟悉本技术领域的技术人员在本申请揭露的技术范围内, 可轻易想到变化或替换, 都应涵盖在本申请的保护范围之内。因此, 本申请的保护范围应以所述权利要求的保护范围为准。

工业实用性

本申请实施例提供了一种编解码方法、编码器、解码器以及存储介质, 在解码端, 解码码流, 确定当前块的 BVD 绝对值和 BVD 符号索引信息; 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据 BVD 符号索引信息和排序结果, 确定当前块的 BVD; 根据当前块的 BVD, 确定当前块的重建值。在编码端, 根据当前块的 BVD 绝对值, 对当前块的候选 BVD 进行排序处理, 确定排序结果; 根据当前块的 BVD 符号信息和排序结果, 确定当前块的 BVD 符号索引信息; 将 BVD 绝对值和 BVD 符号索引信息写入码流。由此可见, 在本申请的实施例中, 可以根据当前块的 BVD 绝对值对当前块的候选 BVD 进行有效排序处理, 从而可以在编码时利用排序结果进行 BVD 符号索引信息的确定和传输, 在解码时利用排序结果进行当前块的 BVD 符号信息的解析, 同时可以使用 CABAC 编解码技术对 BVD 符号信息进行编解码处理, 能够有效地节省 IBC 模式中的 BVD 的符号传输所需的码率, 从而可以提高编码效率。

权利要求书

1、一种解码方法，应用于解码器，所述方法包括：

解码码流，确定当前块的块向量差 BVD 绝对值和 BVD 符号索引信息；

根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；

根据所述 BVD 符号索引信息和所述排序结果，确定所述当前块的 BVD；

根据所述当前块的 BVD，确定所述当前块的重建值。

2、根据权利要求 1 所述的方法，其中，

所述 BVD 绝对值包括第一分量绝对值 absBvdX 和第二分量绝对值 absBvdY ；

所述候选 BVD 包括第一分量 candBvdX 和第二分量 candBvdY 。

3、根据权利要求 2 所述的方法，其中，所述方法还包括：

若所述 absBvdX 等于 0，根据所述 absBvdY 确定所述候选 BVD。

4、根据权利要求 2 所述的方法，其中，所述方法还包括：

若所述 absBvdY 等于 0，根据所述 absBvdX 确定所述候选 BVD。

5、根据权利要求 2 所述的方法，其中，所述方法还包括：

若所述 absBvdX 和所述 absBvdY 均不等于 0，根据所述 absBvdX 和所述 absBvdY 确定所述候选 BVD。

6、根据权利要求 2 所述的方法，其中，所述根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果，包括：

根据所述 BVD 绝对值，确定所述当前块的一个或多个候选 BVD 对应的第一代价值；

根据所述第一代价值，对所述候选 BVD 进行排序，确定所述排序结果。

7、根据权利要求 6 所述的方法，其中，所述方法还包括：

所述一个或多个候选 BVD 包括所述 candBvdX 分别设置为等于 $-\text{absBvdX}$ 或 absBvdX 、所述 candBvdY 分别设置为等于 $-\text{absBvdY}$ 或 absBvdY 的 BVD。

8、根据权利要求 7 所述的方法，其中，所述根据所述当前块的 BVD 绝对值，确定所述当前块的一个或多个候选 BVD 对应的第一代价值，包括：

根据所述候选 BVD 确定的第一匹配模板；

根据预设的误差准则，计算所述当前块的第一模板与所述第一匹配模板之间的匹配误差，以确定所述候选 BVD 对应的第一代价值。

9、根据权利要求 8 所述的方法，其中，所述方法还包括：

所述第一模板包括所述当前块的相邻已解码区域中的一个或多个样值。

10、根据权利要求 8 所述的方法，其中，所述根据所述候选 BVD 确定的第一匹配模板，包括：

确定所述当前块的块向量预测值 BVP；

根据所述 BVP 和所述候选 BVD 确定所述当前块的候选 BV；其中，所述候选 BV 用于指示所述第一匹配模板的位置；

根据所述候选 BV 确定所述第一匹配模板。

11、根据权利要求 10 所述的方法，其中，所述 BVP 包括第三分量 currBvpX 和第四分量 currBvpY ，所述根据所述 BVP 和所述候选 BVD 确定所述当前块的候选 BV，包括：

将所述候选 BV 的二维矢量设置为 $(\text{currBvpX} + \text{candBvdX}, \text{currBvpY} + \text{candBvdY})$ 。

12、根据权利要求 11 所述的方法，其中，所述根据所述候选 BV 确定所述第一匹配模板，包括：

根据所述当前点的位置和所述候选 BV 的二维矢量确定所述第一匹配模板。

13、根据权利要求 12 所述的方法，其中，所述方法还包括：

以所述当前点的位置为起点，将所述候选 BV 的二维矢量所指示的、与所述第一模板具有相同形状和包含相同数量样值的区域，确定为所述第一匹配模板。

14 根据权利要求 8 所述的方法，其中，

所述预设的误差准则包括绝对误差和 SAD、变换绝对误差和 SATD、差值平方和 SSE、平均绝对差值 MAD、平均绝对误差 MAE、平均平方误差 MSE、率失真函数 RDO 中的任意一种。

15、根据权利要求 8 所述的方法，其中，所述排序结果为所述当前块的候选 BVD 列表，所述方法还包括：

根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，其中，所述候选 BVD 列表至少包括一个候选 BVD。

16、根据权利要求 15 所述的方法，其中，所述根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，包括：

根据所述预设的误差准则，按所述第一代价值指示的匹配误差从小到大的顺序，对所述一个或多个候选 BVD 进行排序，确定所述候选 BVD 列表。

17、根据权利要求 15 所述的方法，其中，所述根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，包括：

若所述 absBvdX 和所述 absBvdY 均不等于 0，将所述候选 BVD 分别存储至第一数组和第二数组；

基于所述第一代价值，确定所述第一数组中的第一候选 BVD 和第二候选 BVD，以及所述第二数组中的第三候选 BVD 和第四候选 BVD；

基于所述第一代价值，对所述第一候选 BVD、所述第二候选 BVD、所述第三候选 BVD、所述第四候选 BVD 进行

排序处理, 确定所述候选 BVD 列表。

18、根据权利要求 17 所述的方法, 其中, 所述方法还包括:

若所述第一候选 BVD 的第一代价值小于等于所述第三候选 BVD 的第一代价值, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述
5 第二候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的
第四个候选 BVD;

否则, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第
10 二候选 BVD 设置为所述候选 BVD 列表中的第四个候选 BVD。

19、根据权利要求 17 所述的方法, 其中, 所述方法还包括:

若所述第一候选 BVD 的第一代价值小于等于所述第三候选 BVD 的第一代价值, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第二候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述
15 第三候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的
第四个候选 BVD;

否则, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第
20 二候选 BVD 设置为所述候选 BVD 列表中的第四个候选 BVD。

20、根据权利要求 15 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的
候选 BVD 列表, 包括:

若所述 absBvdX 等于 0 或者所述 absBvdY 等于 0, 将所述候选 BVD 中的所述第一代价值最小的一个候选 BVD 设置
25 为所述候选 BVD 列表中的第一个候选 BVD, 将所述候选 BVD 中的另一个候选 BVD 设置为所述候选 BVD 列表中的第
二个候选 BVD。

21、根据权利要求 8 所述的方法, 其中, 所述排序结果为所述当前块的候选 BVD 集合, 所述方法还包括:

根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的候选 BVD 集合, 其中, 所述候选 BVD 集合
25 至少包括一个候选 BVD。

22、根据权利要求 21 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的
候选 BVD 集合, 包括:

若所述 absBvdX 和所述 absBvdY 均不等于 0, 将所述候选 BVD 分别存储至第一数组和第二数组;

确定所述第一数组中的、所述第一代价值最小的第五候选 BVD, 以及所述第二数组中的、所述第一代价值最小的第
30 六候选 BVD;

将所述第五候选 BVD 和所述第六候选 BVD 中的、所述第一代价值最小的候选 BVD, 确定为所述候选 BVD 集合中的
代价最小 BVD。

23、根据权利要求 21 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的
候选 BVD 集合, 包括:

若所述 absBvdX 等于 0 或者所述 absBvdY 等于 0, 将所述候选 BVD 中的所述第一代价值最小的一个候选 BVD 确定
35 为所述候选 BVD 集合中的代价最小 BVD。

24、根据权利要求 16-20 任一项所述的方法, 其中, 所述根据所述 BVD 符号索引信息和所述排序结果, 确定所述当
前块的 BVD 包括:

将所述候选 BVD 列表中的、所述 BVD 符号索引信息指示的候选 BVD 确定为所述当前块的 BVD。

25、根据权利要求 16 所述的方法, 其中, 所述方法还包括:

对所述候选 BVD 列表进行转换处理, 确定索引转换列表。

26、根据权利要求 25 所述的方法, 其中, 所述根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的
BVD, 包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述 BVD 符号索引信息的第一比特位取值为 0, 则确定所述当前
45 块的 BVD 的 candBvdX 与所述索引转换列表中第一个候选 BVD 或第三个候选 BVD 的 candBvdX 相同;

按照所述第一个候选 BVD 或第三个候选 BVD 确定所述当前块的 BVD 的 candBvdY 。

27、根据权利要求 26 所述的方法, 其中, 所述方法还包括:

若所述 BVD 符号索引信息的第二比特位取值为 0, 则确定所述当前块的 BVD 的 candBvdY 与所述第一个候选 BVD
50 的 candBvdY 相同;

若所述第二比特位取值为 1, 则确定所述当前块的 BVD 的 candBvdY 与所述第三个候选 BVD 的 candBvdY 相同。

28、根据权利要求 25 所述的方法, 其中, 所述根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的
BVD, 包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述 BVD 符号索引信息的第一比特位取值为 1, 则确定所述第一
分量符号与所述索引转换列表中第二个候选 BVD 或第四个候选 BVD 的第一分量符号相同;

按照所述第二个候选 BVD 或第四个候选 BVD 确定所述当前块的 BVD 的 candBvdY 。

29、根据权利要求 28 所述的方法, 其中, 所述方法还包括:

若所述 BVD 符号索引信息的第二比特位取值为 0, 则确定所述当前块的 BVD 的 candBvdY 与所述第二个候选 BVD
55 的 candBvdY 相同;

若所述第二比特位取值为 1, 则确定所述当前块的 BVD 的 candBvdY 与所述第四个候选 BVD 的 candBvdY 相同。

30、根据权利要求 25 所述的方法, 其中, 所述根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的
BVD, 包括:

当所述 absBvdX 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述索引转换列表中的第一个候选 BVD 的 candBvdY 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 candBvdY 与所述索引转换列表中的第二个候选 BVD 的 candBvdY 相同。

31、根据权利要求 25 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述排序结果,确定所述当前块的 BVD,包括:

当所述 absBvdY 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 candBvdX 与所述索引转换列表中的第一个候选 BVD 的 candBvdX 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 candBvdX 与所述索引转换列表中的第二个候选 BVD 的 candBvdX 相同。

32、根据权利要求 16 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述候选 BVD 列表,确定所述当前块的 BVD,包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时,若所述 BVD 符号索引信息的第一比特位取值为 0,则确定所述当前块的 BVD 的 candBvdX 与所述候选 BVD 列表中的第一个候选 BVD 的 candBvdX 相同,否则不同;

遍历所述候选 BVD 列表,确定与所述当前块的 BVD 具有相同 candBvdX 的首个候选 BVD;

若所述 BVD 符号索引信息的第二比特位取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述首个候选 BVD 的 candBvdY 相同,否则不同。

33、根据权利要求 16 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述候选 BVD 列表,确定所述当前块的 BVD,包括:

当所述 absBvdX 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 列表中的第一个候选 BVD 的 candBvdY 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 列表中的第二个候选 BVD 的 candBvdY 相同。

34、根据权利要求 16 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述候选 BVD 列表,确定所述当前块的 BVD,包括:

当所述 absBvdY 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 candBvdX 与所述候选 BVD 列表中的第一个候选 BVD 的 candBvdX 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 candBvdX 与所述候选 BVD 列表中的第二个候选 BVD 的 candBvdX 相同。

35、根据权利要求 21 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述排序结果,确定所述当前块的 BVD,包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时,若所述 BVD 符号索引信息的第一比特位取值为 0,则确定所述当前块的 BVD 的 candBvdX 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdX 相同;

按照所述候选 BVD 集合中的代价最小 BVD 确定所述当前块的 BVD 的 candBvdY。

36、根据权利要求 21 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述排序结果,确定所述当前块的 BVD,包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时,若所述 BVD 符号索引信息的第一比特位取值为 1,则确定所述当前块的 BVD 的 candBvdX 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdX 符号相反;

按照所述候选 BVD 集合中的代价最小 BVD 所在的数组以外的、另一个数组中的代价最小 BVD 确定所述当前块的 BVD 的 candBvdY。

37、根据权利要求 35 所述的方法,其中,所述方法还包括:

若所述 BVD 符号索引信息的第二比特位取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdY 相同;

若所述第二比特位取值为 1,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdY 符号相反。

38、根据权利要求 36 所述的方法,其中,所述方法还包括:

若所述 BVD 符号索引信息的第二比特位取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述另一个数组中的代价最小 BVD 的 candBvdY 相同;

若所述第二比特位取值为 1,则确定所述当前块的 BVD 的 candBvdY 与所述另一个数组中的代价最小 BVD 的 candBvdY 符号相反。

39、根据权利要求 21 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述排序结果,确定所述当前块的 BVD,包括:

当所述 absBvdX 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdY 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 candBvdY 与所述候选 BVD 集合中的代价最小 BVD 的 candBvdY 符号相反。

40、根据权利要求 21 所述的方法,其中,所述根据所述 BVD 符号索引信息和所述排序结果,确定所述当前块的 BVD,包括:

当所述 absBvdY 为 0 时,若所述 BVD 符号索引信息的取值为 0,则确定所述当前块的 BVD 的 absBvdX 与所述候选 BVD 集合中的代价最小 BVD 的 absBvdX 相同;若所述 BVD 符号索引信息的取值为 1,则确定所述当前块的 BVD 的 absBvdX 与所述候选 BVD 集合中的代价最小 BVD 的 absBvdX 符号相反。

41、根据权利要求 2 所述的方法,其中,所述方法还包括:

若所述 absBvdX 和所述 absBvdY 均不为 0,确定所述 BVD 符号索引信息的比特位数为 2;若所述 absBvdX 或所述 absBvdY 为 0,确定所述 BVD 符号索引信息的比特位数为 1。

42、根据权利要求 2 所述的方法,其中,所述方法还包括:

采用截断一元码编码所述 BVD 符号索引信息,则不同候选 BVD 对应的所述符号索引信息的比特位数不完全相同。

43、一种编码方法，应用于编码器，所述方法包括：

根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果；

根据所述当前块的 BVD 符号信息和所述排序结果，确定所述当前块的 BVD 符号索引信息；

将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

5 44、根据权利要求 43 所述的方法，其中，

所述 BVD 绝对值包括第一分量绝对值 absBvdX 和第二分量绝对值 absBvdY ；

所述候选 BVD 包括第一分量 candBvdX 和第二分量 candBvdY 。

45、根据权利要求 44 所述的方法，其中，所述方法还包括：

若所述 absBvdX 等于 0，根据所述 absBvdY 确定所述候选 BVD。

10 46、根据权利要求 44 所述的方法，其中，所述方法还包括：

若所述 absBvdY 等于 0，根据所述 absBvdX 确定所述候选 BVD。

47、根据权利要求 44 所述的方法，其中，所述方法还包括：

若所述 absBvdX 和所述 absBvdY 均不等于 0，根据所述 absBvdX 和所述 absBvdY 确定所述候选 BVD。

15 48、根据权利要求 44 所述的方法，其中，所述根据当前块的 BVD 绝对值，对所述当前块的候选 BVD 进行排序处理，确定排序结果，包括：

根据所述 BVD 绝对值，确定所述当前块的一个或多个候选 BVD 对应的第一代价值；

根据所述第一代价值，对所述候选 BVD 进行排序，确定所述排序结果。

49、根据权利要求 48 所述的方法，其中，所述方法还包括：

20 所述一个或多个候选 BVD 包括所述 candBvdX 分别设置为等于 $-\text{absBvdX}$ 或 absBvdX 、所述 candBvdY 分别设置为等于 $-\text{absBvdY}$ 或 absBvdY 的 BVD。

50、根据权利要求 49 所述的方法，其中，所述根据所述当前块的 BVD 绝对值，确定所述当前块的一个或多个候选 BVD 对应的第一代价值，包括：

根据所述候选 BVD 确定的第一匹配模板；

25 根据预设的误差准则，计算所述当前块的第一模板与所述第一匹配模板之间的匹配误差，以确定所述候选 BVD 对应的第一代价值。

51、根据权利要求 50 所述的方法，其中，所述方法还包括：

所述第一模板包括所述当前块的相邻已解码区域中的一个或多个样值。

52、根据权利要求 50 所述的方法，其中，所述根据所述候选 BVD 确定的第一匹配模板，包括：

确定所述当前块的块向量预测值 BVP；

30 根据所述 BVP 和所述候选 BVD 确定所述当前块的候选 BV；其中，所述候选 BV 用于指示所述第一匹配模板的位置；

根据所述候选 BV 确定所述第一匹配模板。

53、根据权利要求 52 所述的方法，其中，所述 BVP 包括第三分量 currBvpX 和第四分量 currBvpY ，所述根据所述 BVP 和所述候选 BVD 确定所述当前块的候选 BV，包括：

35 将所述候选 BV 的二维矢量设置为 $(\text{currBvpX} + \text{candBvdX}, \text{currBvpY} + \text{candBvdY})$ 。

54、根据权利要求 53 所述的方法，其中，所述根据所述候选 BV 确定所述第一匹配模板，包括：

根据所述当前点的位置和所述候选 BV 的二维矢量确定所述第一匹配模板。

55、根据权利要求 54 所述的方法，其中，所述方法还包括：

40 以所述当前点的位置为起点，将所述候选 BV 的二维矢量所指示的、与所述第一模板具有相同形状和包含相同数量样值的区域，确定为所述第一匹配模板。

56、根据权利要求 50 所述的方法，其中，

所述预设的误差准则包括绝对误差和 SAD、变换绝对误差和 SATD、差值平方和 SSE、平均绝对差值 MAD、平均绝对误差 MAE、平均平方误差 MSE、率失真函数 RDO 中的任意一种。

57、根据权利要求 50 所述的方法，其中，所述排序结果为所述当前块的候选 BVD 列表，所述方法还包括：

45 根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，其中，所述候选 BVD 列表至少包括一个候选 BVD。

58、根据权利要求 57 所述的方法，其中，所述根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，包括：

50 根据所述预设的误差准则，按所述第一代价值指示的匹配误差从小到大的顺序，对所述一个或多个候选 BVD 进行排序，确定所述候选 BVD 列表。

59、根据权利要求 57 所述的方法，其中，所述根据所述第一代价值，对所述候选 BVD 进行排序，确定所述当前块的候选 BVD 列表，包括：

若所述 absBvdX 和所述 absBvdY 均不等于 0，将所述候选 BVD 分别存储至第一数组和第二数组；

55 基于所述第一代价值，确定所述第一数组中的第一候选 BVD 和第二候选 BVD，以及所述第二数组中的第三候选 BVD 和第四候选 BVD；

基于所述第一代价值，对所述第一候选 BVD、所述第二候选 BVD、所述第三候选 BVD、所述第四候选 BVD 进行排序处理，确定所述候选 BVD 列表。

60、根据权利要求 59 所述的方法，其中，所述方法还包括：

60 若所述第一候选 BVD 的第一代价值小于等于所述第三候选 BVD 的第一代价值，将所述第一候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD，将所述第三候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD，将所述第二候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD，将所述第四候选 BVD 设置为所述候选 BVD 列表中的

第四个候选 BVD;

否则, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第二候选 BVD 设置为所述候选 BVD 列表中的第四个候选 BVD。

5 61、根据权利要求 59 所述的方法, 其中, 所述方法还包括:

若所述第一候选 BVD 的第一代价值小于等于所述第三候选 BVD 的第一代价值, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第二候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的第四个候选 BVD;

10 否则, 将所述第三候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述第四候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD, 将所述第一候选 BVD 设置为所述候选 BVD 列表中的第三个候选 BVD, 将所述第二候选 BVD 设置为所述候选 BVD 列表中的第四个候选 BVD。

62、根据权利要求 57 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的候选 BVD 列表, 包括:

15 若所述 absBvdX 等于 0 或者所述 absBvdY 等于 0, 将所述候选 BVD 中的所述第一代价值最小的一个候选 BVD 设置为所述候选 BVD 列表中的第一个候选 BVD, 将所述候选 BVD 中的另一个候选 BVD 设置为所述候选 BVD 列表中的第二个候选 BVD。

63、根据权利要求 50 所述的方法, 其中, 所述排序结果为所述当前块的候选 BVD 集合, 所述方法还包括:

20 根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的候选 BVD 集合, 其中, 所述候选 BVD 集合至少包括一个候选 BVD。

64、根据权利要求 62 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的候选 BVD 集合, 包括:

若所述 absBvdX 和所述 absBvdY 均不等于 0, 将所述候选 BVD 分别存储至第一数组和第二数组;

25 确定所述第一数组中的、所述第一代价值最小的第五候选 BVD, 以及所述第二数组中的、所述第一代价值最小的第六候选 BVD;

将所述第五候选 BVD 和所述第六候选 BVD 中的、所述第一代价值最小的候选 BVD, 确定为所述候选 BVD 集合中的代价最小 BVD。

65、根据权利要求 63 所述的方法, 其中, 所述根据所述第一代价值, 对所述候选 BVD 进行排序, 确定所述当前块的候选 BVD 集合, 包括:

30 若所述 absBvdX 等于 0 或者所述 absBvdY 等于 0, 将所述候选 BVD 中的所述第一代价值最小的一个候选 BVD 确定为所述候选 BVD 集合中的代价最小 BVD。

66、根据权利要求 56-62 任一项所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

在所述候选 BVD 列表中确定与所述当前块的 BVD 符号信息相同的所述候选 BVD;

35 按照所述候选 BVD 的索引值设置所述 BVD 符号索引信息。

67、根据权利要求 58 所述的方法, 其中, 所述方法还包括:

对所述候选 BVD 列表进行转换处理, 确定索引转换列表。

68、根据权利要求 67 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

40 当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述索引转换列表中的第一个候选 BVD 的第一分量符号相同, 则将所述 BVD 符号索引信息的第一比特位确定为 0; 否则将所述第一比特位确定为 1。

69、根据权利要求 67 所述的方法, 其中, 所述方法还包括:

45 当所述第一比特位确定为 0 时, 若所述当前块的 BVD 符号信息的第二分量符号与所述索引转换列表中的第一个候选 BVD 的第二分量符号相同, 则将所述 BVD 符号索引信息的第二比特位确定为 0; 否则将所述第二比特位确定为 1;

当所述第一比特位确定为 1 时, 若所述当前块的 BVD 符号信息的第二分量符号与所述索引转换列表中、索引为 1 的候选 BVD 的第二分量符号相同, 则将所述 BVD 符号索引信息的第二比特位确定为 0; 否则将所述第二比特位确定为 1。

50 70、根据权利要求 67 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdX 为 0 时, 若所述当前块的 BVD 符号信息的第二分量符号与所述索引转换列表中的第一个候选 BVD 的第二分量符号相同, 则将所述 BVD 符号索引信息确定为 0; 否则将所述 BVD 符号索引信息确定为 1。

71、根据权利要求 67 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

55 当所述 absBvdY 为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述索引转换列表中的第一个候选 BVD 的第一分量符号相同, 则将所述 BVD 符号索引信息确定为 0; 否则将所述 BVD 符号索引信息确定为 1。

72、根据权利要求 58 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

60 当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与与所述候选 BVD 列表中的第一个候选 BVD 的第一分量符号相同, 则将所述 BVD 符号索引信息的第一比特位确定为 0; 否则将所述第一比特位确定为 1;

遍历所述候选 BVD 列表, 确定与所述当前块的 BVD 符号信息具有相同第一分量符号的首个候选 BVD;

若所述当前块的 BVD 符号信息的第二分量符号与所述首个候选 BVD 的第二分量符号相同, 则将所述 BVD 符号索引信息的第二比特位确定为 0; 否则将所述第二比特位确定为 1。

73、根据权利要求 58 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdX 为 0 时, 若所述当前块的 BVD 符号信息的第二分量符号与所述候选 BVD 列表中的第一个候选 BVD 的第二分量符号相同, 则将所述 BVD 符号索引信息确定为 0, 否则将所述 BVD 符号索引信息确定为 1。

74、根据权利要求 58 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdY 为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述候选 BVD 列表中的第一个候选 BVD 的第一分量符号相同, 则将所述 BVD 符号索引信息确定为 0, 否则将所述 BVD 符号索引信息确定为 1。

75、根据权利要求 63 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述候选 BVD 集合中的代价最小 BVD 的第一分量符号相同, 则确定所述 BVD 符号索引信息的第一比特位取值为 0;

按照所述候选 BVD 集合中的代价最小 BVD 确定所述当前块的 BVD 符号索引信息的第二比特位。

76、根据权利要求 63 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdX 和所述 absBvdY 均不为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述候选 BVD 集合中的代价最小 BVD 的第一分量符号不同, 则确定所述 BVD 符号索引信息的第一比特位取值为 1;

按照所述候选 BVD 集合中的代价最小 BVD 所在的数组以外的、另一个数组中的代价最小 BVD 确定所述当前块的 BVD 符号索引信息的第二比特位。

77、根据权利要求 75 所述的方法, 其中, 所述方法还包括:

若所述当前块的 BVD 符号信息的第二分量符号与所述候选 BVD 集合中的代价最小 BVD 的第二分量符号相同, 则确定所述 BVD 符号索引信息的第二比特位取值为 0;

若所述当前块的 BVD 符号信息的第二分量符号与所述候选 BVD 集合中的代价最小 BVD 的第二分量符号不同, 则确定所述 BVD 符号索引信息的第二比特位取值为 1。

78、根据权利要求 76 所述的方法, 其中, 所述方法还包括:

若所述当前块的 BVD 符号信息的第二分量符号与所述另一个数组中的代价最小 BVD 的第二分量符号相同, 则确定所述 BVD 符号索引信息的第二比特位取值为 0;

若所述当前块的 BVD 符号信息的第二分量符号与所述另一个数组中的代价最小 BVD 的第二分量符号不同, 则确定所述 BVD 符号索引信息的第二比特位取值为 1。

79、根据权利要求 63 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdX 为 0 时, 若所述当前块的 BVD 符号信息的第二分量符号与所述候选 BVD 集合中的代价最小 BVD 的第二分量符号相同, 则确定所述 BVD 符号索引信息的取值为 0; 否则确定所述 BVD 符号索引信息的取值为 1。

80、根据权利要求 63 所述的方法, 其中, 所述根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息, 包括:

当所述 absBvdY 为 0 时, 若所述当前块的 BVD 符号信息的第一分量符号与所述候选 BVD 集合中的代价最小 BVD 的第一分量符号相同, 则确定所述 BVD 符号索引信息的取值为 0; 否则确定所述 BVD 符号索引信息的取值为 1。

81、根据权利要求 43 所述的方法, 其中, 所述方法还包括:

使用上下文自适应的二进制算术编码 CABAC 编码所述 BVD 绝对值和所述 BVD 符号索引信息。

82、根据权利要求 43 所述的方法, 其中, 所述方法还包括:

使用旁路模式编码所述 BVD 绝对值和所述 BVD 符号索引信息。

83、一种编码器, 所述编码器包括第一确定单元, 编码单元; 其中,

所述第一确定单元, 配置为根据当前块的 BVD 绝对值, 对所述当前块的候选 BVD 进行排序处理, 确定排序结果; 根据所述当前块的 BVD 符号信息和所述排序结果, 确定所述当前块的 BVD 符号索引信息;

所述编码单元, 配置为将所述 BVD 绝对值和所述 BVD 符号索引信息写入码流。

84、一种编码器, 所述编码器包括第一存储器和第一处理器; 其中,

所述第一存储器, 用于存储能够在所述第一处理器上运行的计算机程序;

所述第一处理器, 用于在运行所述计算机程序时, 执行如权利要求 41 至 79 任一项所述的方法。

85、一种解码器, 所述解码器包括解码单元, 第二确定单元; 其中,

所述解码单元, 配置为解码码流;

所述第二确定单元, 配置为确定当前块的 BVD 绝对值和 BVD 符号索引信息;

根据当前块的 BVD 绝对值, 对所述当前块的候选 BVD 进行排序处理, 确定排序结果; 根据所述 BVD 符号索引信息和所述排序结果, 确定所述当前块的 BVD; 根据所述当前块的 BVD, 确定所述当前块的重建值。

86、一种解码器, 所述解码器包括第二存储器和第二处理器; 其中,

所述第二存储器, 用于存储能够在所述第二处理器上运行的计算机程序;

所述第二处理器, 用于在运行所述计算机程序时, 执行如权利要求 1 至 40 任一项所述的方法。

87、一种计算机可读存储介质, 所述计算机可读存储介质存储有计算机程序, 所述计算机程序被执行时实现如权利要求 1 至 42 任一项所述的方法、或者实现如权利要求 43 至 82 任一项所述的方法。

5

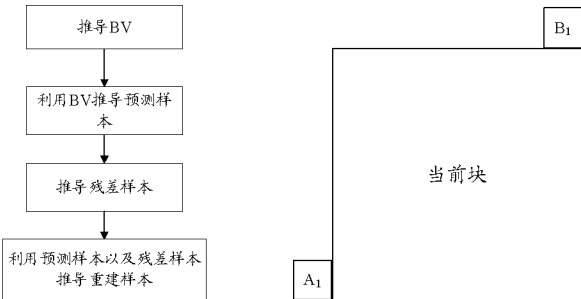


图 1图 2

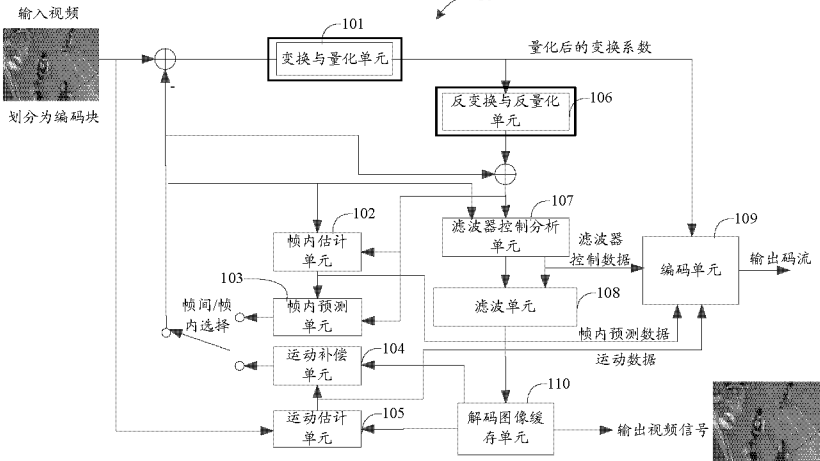


图 3

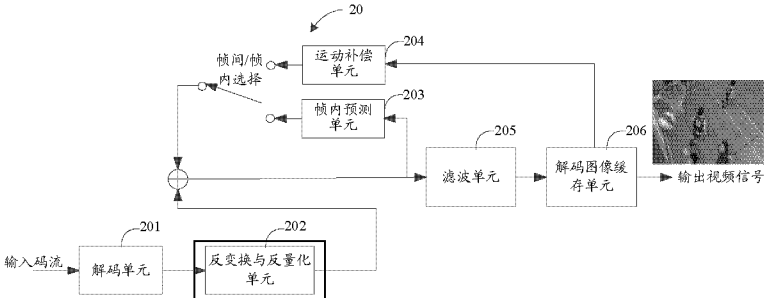


图 4

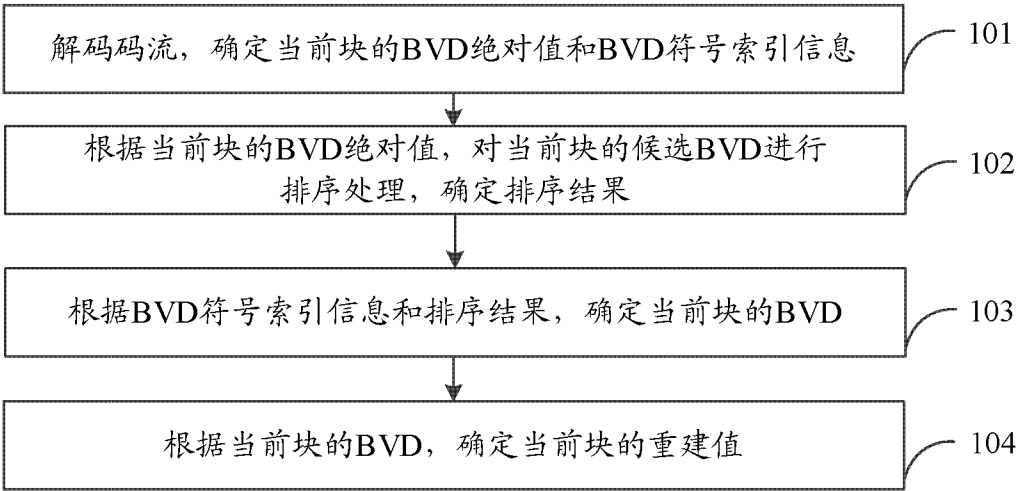


图 5

10

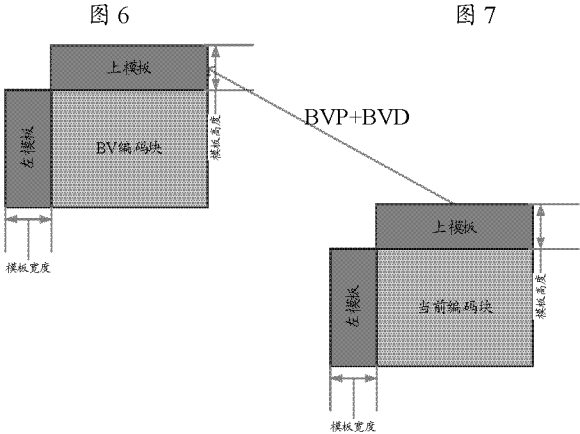
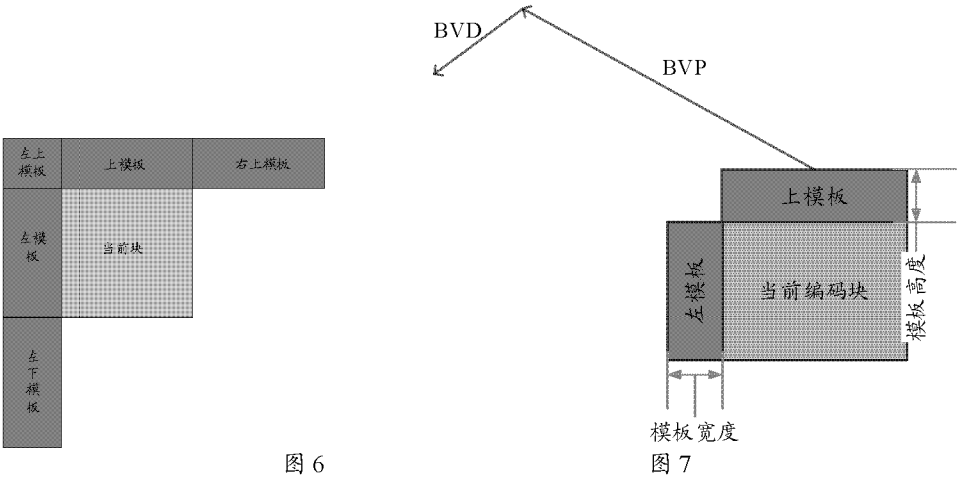


图 8

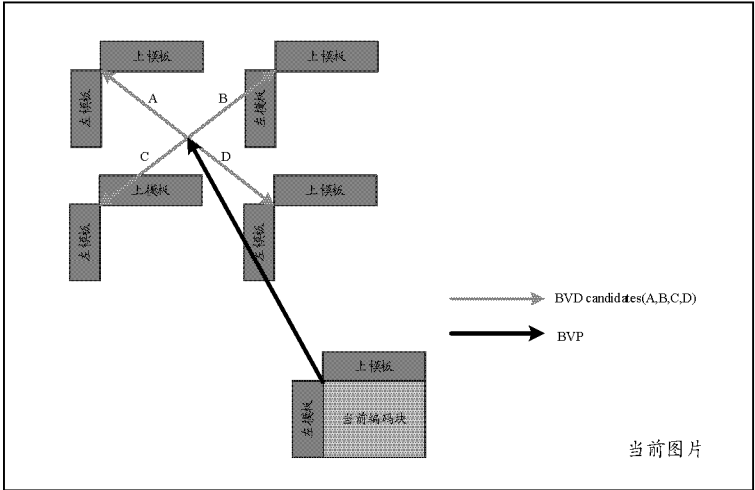


图 9

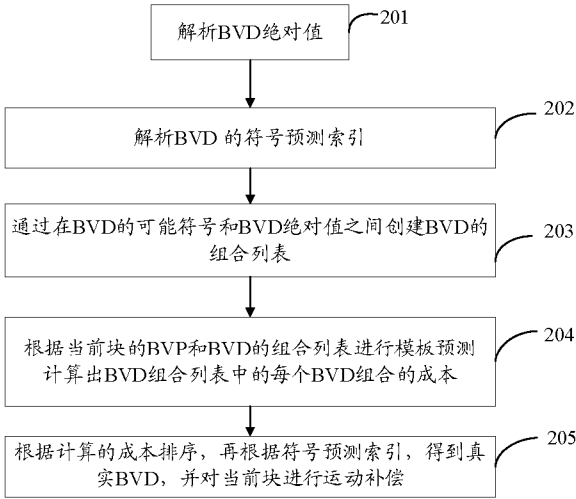


图 10

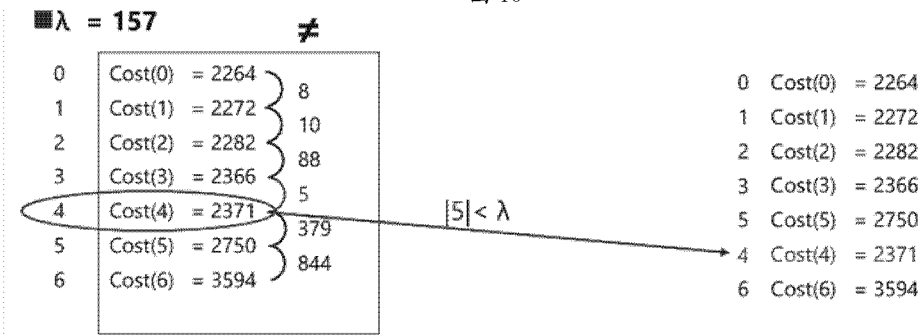


图 11

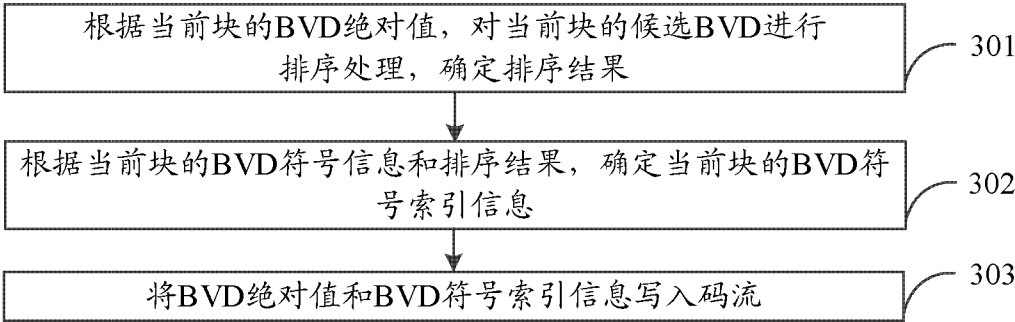


图 12

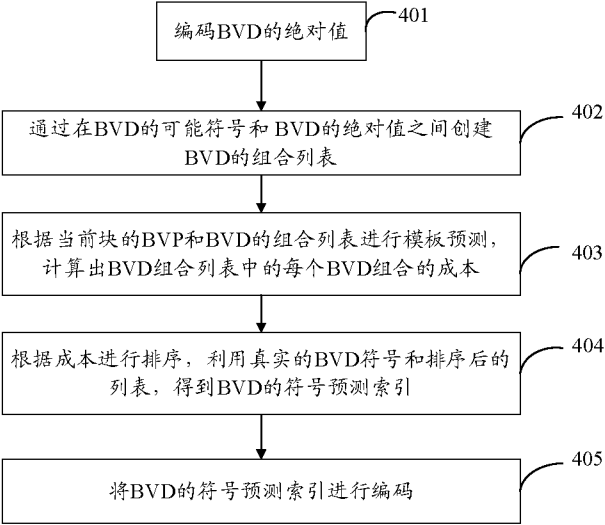


图 13

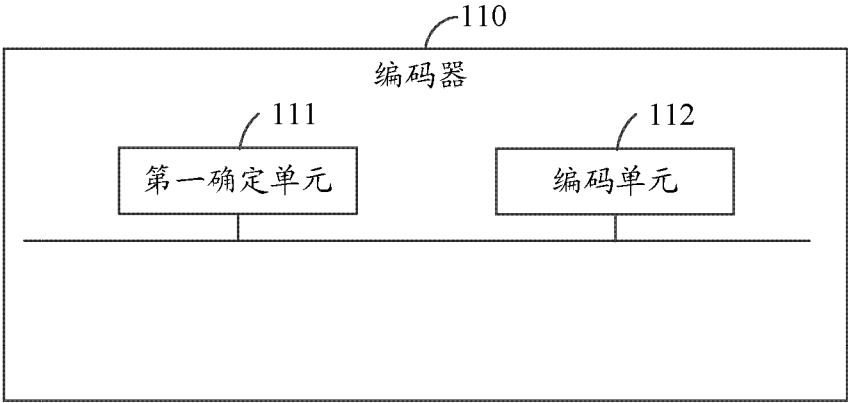


图 14

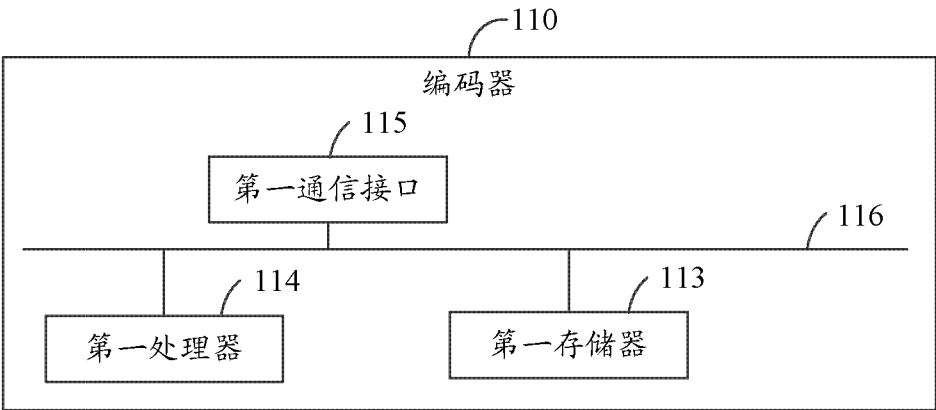


图 15

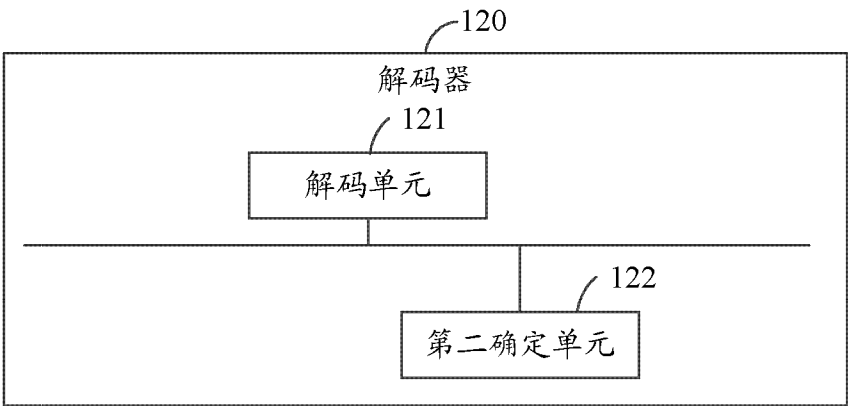


图 16

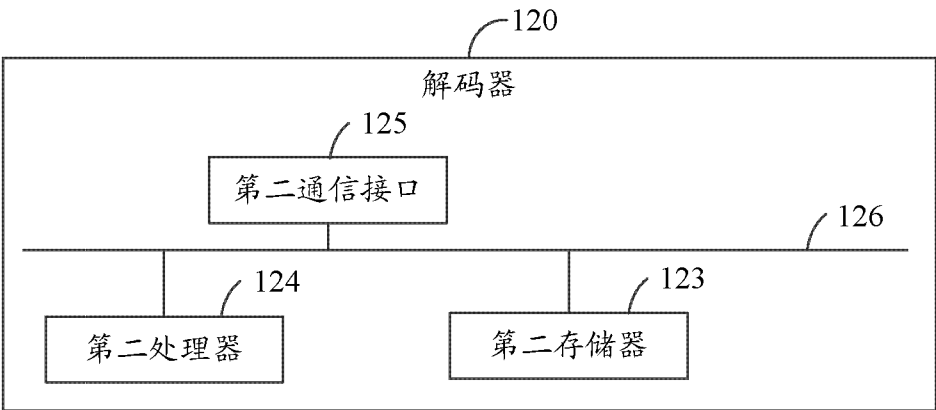


图 17

5

10

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/125219

A. CLASSIFICATION OF SUBJECT MATTER

H04N19/52(2014.01)i; H04N19/86(2014.01)i

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

CNABS, CNTXT, ENTXT, DWPI, JVET: 块, 复制, 拷贝, 向量, 矢量, 差, 绝对值, 符号, 索引, 排序, 顺序, 成本, 代价, 失真, IBC, BVD, intra, block, copy, vector, difference, signal, index, syntax, +order+, cost, distort+

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	CN 106464904 A (QUALCOMM INC.) 22 February 2017 (2017-02-22) description, paragraphs [0006]-[0011]	1-87
A	CN 111866515 A (HANGZHOU HIKVISION DIGITAL TECHNOLOGY CO., LTD.) 30 October 2020 (2020-10-30) entire document	1-87
A	WO 2020245391 A1 (FRAUNHOFER-GESELLSCHAFT ZURFORDERUNG DER ANGEWANDTEN FORSCHUNG E.V.) 10 December 2020 (2020-12-10) entire document	1-87
A	US 2017332097 A1 (KT CORP.) 16 November 2017 (2017-11-16) entire document	1-87
A	SALEHIFAR, Mehdi et al. "CE8-Related : On MVD Coding (JVET-N0289)" <i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19-27 March 2019, 27 March 2019 (2019-03-27),</i> entire document	1-87



Further documents are listed in the continuation of Box C.



See patent family annex.

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"D" document cited by the applicant in the international application

"E" earlier application or patent but published on or after the international filing date

"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)

"O" document referring to an oral disclosure, use, exhibition or other means

"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention

"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone

"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art

"&" document member of the same patent family

Date of the actual completion of the international search

18 May 2023

Date of mailing of the international search report

25 May 2023

Name and mailing address of the ISA/CN

China National Intellectual Property Administration (ISA/
CN)
China No. 6, Xitucheng Road, Jimenqiao, Haidian District,
Beijing 100088

Authorized officer

Telephone No.

INTERNATIONAL SEARCH REPORT

International application No.

PCT/CN2022/125219

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	SUN, Yucheng et al. "CE8-related: Simple BVD coding scheme without adding more context (JVET-O0325)" <i>Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3–12 July 2019, 12 June 2019 (2019-06-12), entire document</i>	1-87

INTERNATIONAL SEARCH REPORT
Information on patent family members

International application No.

PCT/CN2022/125219

Patent document cited in search report			Publication date (day/month/year)	Patent family member(s)			Publication date (day/month/year)
CN	106464904	A	22 February 2017	WO	2015196087	A1	23 December 2015
				EP	3158750	A1	26 April 2017
				US	2015373370	A1	24 December 2015
				JP	2017519447	A	13 July 2017
				IN	201647039848	A	23 December 2016
CN	111866515	A	30 October 2020	WO	2020221317	A1	05 November 2020
WO	2020245391	A1	10 December 2020	EP	3981147	A1	13 April 2022
				US	2022094925	A1	24 March 2022
				CN	114026853	A	08 February 2022
US	2017332097	A1	16 November 2017	KR	20160063996	A	07 June 2016
				KR	20200132829	A	25 November 2020
				WO	2016085231	A1	02 June 2016
				CN	107005696	A	01 August 2017
				VN	53858	A	25 September 2017
				CN	111031323	A	17 April 2020

国际检索报告

国际申请号

PCT/CN2022/125219

A. 主题的分类

H04N19/52 (2014.01) i; H04N19/86 (2014.01) i

按照国际专利分类 (IPC) 或者同时按照国家分类和 IPC 两种分类

B. 检索领域

检索的最低限度文献 (标明分类系统和分类号)

IPC: H04N

包含在检索领域中的除最低限度文献以外的检索文献

在国际检索时查阅的电子数据库 (数据库的名称, 和使用的检索词 (如使用))

CNABS, CNTXT, ENTXT, DWPI, JNET: 块, 复制, 拷贝, 向量, 矢量, 差, 绝对值, 符号, 索引, 排序, 顺序, 成本, 代价, 失真, IBC, BVD, intra, block, copy, vector, difference, signal, index, syntax, +order+, cost, distort+

C. 相关文件

类 型*	引用文件, 必要时, 指明相关段落	相关的权利要求
A	CN 106464904 A (高通股份有限公司) 2017年2月22日 (2017 - 02 - 22) 说明书第0006-0011段	1-87
A	CN 111866515 A (杭州海康威视数字技术股份有限公司) 2020年10月30日 (2020 - 10 - 30) 全文	1-87
A	WO 2020245391 A1 (FRAUNHOFER-GESELLSCHAFT ZURFORDERUNG DER ANGEWANDTEN FORSCHUNG E. V.) 2020年12月10日 (2020 - 12 - 10) 全文	1-87
A	US 2017332097 A1 (KT CORPORATION) 2017年11月16日 (2017 - 11 - 16) 全文	1-87
A	SALEHIFAR, Mehdi等. "CE8-Related : On MVD Coding (JVET-N0289)" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 14th Meeting: Geneva, CH, 19 - 27 March 2019, 2019年3月27日 (2019 - 03 - 27), 全文	1-87

☒ 其余文件在C栏的续页中列出。☒ 见同族专利附件。

* 引用文件的具体类型:

“A” 认为不特别相关的表示了现有技术一般状态的文件

“D” 申请人在国际申请中引证的文件

“E” 在国际申请日的当天或之后公布的在先申请或专利

“L” 可能对优先权要求构成怀疑的文件, 或为确定另一篇引用文件的公布日而引用的或者因其他特殊理由而引用的文件 (如具体说明的)

“O” 涉及口头公开、使用、展览或其他方式公开的文件

“P” 公布日先于国际申请日但迟于所要求的优先权日的文件

“T” 在申请日或优先权日之后公布, 与申请不相抵触, 但为了理解发明之理论或原理的在后文件

“X” 特别相关的文件, 单独考虑该文件, 认定要求保护的发明不是新颖的或不具有创造性

“Y” 特别相关的文件, 当该文件与另一篇或者多篇该类文件结合并且这种结合对于本领域技术人员为显而易见时, 要求保护的发明不具有创造性

“&” 同族专利的文件

国际检索实际完成的日期

2023年5月18日

国际检索报告邮寄日期

2023年5月25日

ISA/CN的名称和邮寄地址

中国国家知识产权局

中国北京市海淀区蓟门桥西土城路6号 100088

授权官员

谢佳妮

电话号码 (+86) 010-53961703

C. 相关文件		
类 型*	引用文件，必要时，指明相关段落	相关的权利要求
A	SUN, Yucheng等. "CE8-related: Simple BVD coding scheme without adding more context (JVET-00325)" Joint Video Experts Team (JVET) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11 15th Meeting: Gothenburg, SE, 3 - 12 July 2019, 2019年6月12日 (2019 - 06 - 12), 全文	1-87

国际检索报告
关于同族专利的信息

国际申请号

PCT/CN2022/125219

检索报告引用的专利文件			公布日 (年/月/日)	同族专利			公布日 (年/月/日)
CN	106464904	A	2017年2月22日	WO	2015196087	A1	2015年12月23日
				EP	3158750	A1	2017年4月26日
				US	2015373370	A1	2015年12月24日
				JP	2017519447	A	2017年7月13日
				IN	201647039848	A	2016年12月23日
CN	111866515	A	2020年10月30日	WO	2020221317	A1	2020年11月5日
WO	2020245391	A1	2020年12月10日	EP	3981147	A1	2022年4月13日
				US	2022094925	A1	2022年3月24日
				CN	114026853	A	2022年2月8日
US	2017332097	A1	2017年11月16日	KR	20160063996	A	2016年6月7日
				KR	20200132829	A	2020年11月25日
				WO	2016085231	A1	2016年6月2日
				CN	107005696	A	2017年8月1日
				VN	53858	A	2017年9月25日
				CN	111031323	A	2020年4月17日