



# [12] 发明专利说明书

专利号 ZL 200310116090.2

[45] 授权公告日 2005 年 10 月 26 日

[11] 授权公告号 CN 1225128C

[22] 申请日 2003. 12. 31

[21] 申请号 200310116090.2

[71] 专利权人 中国科学院计算技术研究所

地址 100080 北京市海淀区中关村南大街 8 号中科院物理所南门研究生楼联合实验室

[72] 发明人 季向阳 高文 马思伟 赵德斌

吕岩

审查员 赵博华

[74] 专利代理机构 北京同立钧成知识产权代理有限公司

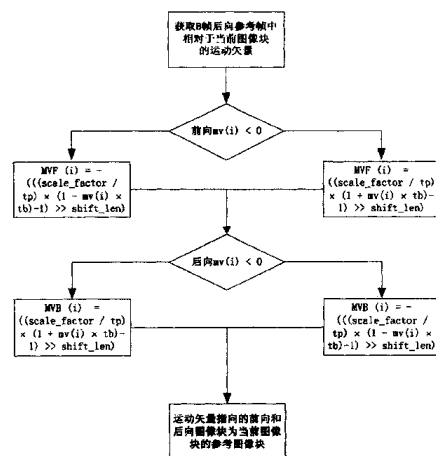
代理人 刘芳 刘薇

权利要求书 4 页 说明书 9 页 附图 1 页

[54] 发明名称 直接编码模式下确定参考图像块的方法

[57] 摘要

本发明提出了一种直接编码模式下确定参考图像块的方法，能够很好地解决在保持精确的运动矢量的同时，又可以通过无除法的方式来实现，从而提高运动矢量的计算的精度，可以更真实地体现视频中物体的运动，获得更准确的运动矢量预测，与前向预测编码，后向预测编码相结合，可用实现一种新的预测编码类型，既可以保证 direct mode 编码高效性又便于硬件的实现，取得与传统 B 帧编码相似的效果，可以用于正在制定的 AVS 标准。



ISSN 1008-4274

1、一种直接编码模式下确定参考图像块的方法，其特征在于：包括：

步骤一、获取 B 帧后向参考帧中相对于当前图像块的运动矢量；

步骤二、根据已经得到的后向参考帧中对应图像块的运动矢量，得出用

5 于 B 帧当前图像块直接编码的运动矢量向量  $MV(x,y)$ ，

当前块的前向运动矢量  $MV_f$  采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ， $td = tp - tb$ ；

如果  $MV(i) < 0$ ：

$MV_f(i) = -(((scale\_factor / tp) \times (1 - MV(i) \times tb) - 1) \gg shift\_len)$

10 否则，

$MV_f(i) = ((scale\_factor / tp) \times (1 + MV(i) \times tb) - 1) \gg shift\_len)$

当前块的后向运动矢量  $MV_b$  采用如下公式计算：

如果  $MV(i) < 0$ ：

$MV_b(i) = ((scale\_factor / tp) \times (1 - MV(i) \times td) - 1) \gg shift\_len$

15 否则，

$MV_b(i) = -(((scale\_factor / tp) \times (1 + MV(i) \times td) - 1) \gg shift\_len)$

其中  $scale\_factor$  值为小数放大因子， $shift\_len$  表示需要向右移位的次数， $MV_f$  与  $MV_b$  则是当前块相应的前向运动矢量与后向运动矢量， $tb$  是当前图片与前向参考图片在时域上的距离， $td$  是前向参考图片与后向参考图片在时域上的距离， $tp$  是当前图片与后向参考图片在时域上的距离。

$MV$  表示后向参考图片相应部分相对于前向参考帧的运动矢量， $MV(x,y) = (MV(x), MV(y))$  为二维矢量，其相应分量为  $MV(x), MV(y)$ ， $MV(i)$  代表  $MV(x)$  或  $MV(y)$ ； $MV(x) < 0$  代表参考块位于参考帧中当前块相对应位置的左侧； $MV(y) < 0$  代表代表参考块位于参考帧中当前块相对应位置的上方； $a/b$  表示为  $a$  与  $b$  的商向零方向取整。

步骤三、步骤二得到的运动矢量指向的前向和后向图像块为当前图像块

的参考图像块。

2、根据权利要求1所述的直接编码模式下确定参考图像块的方法，其特征在于：所述的步骤二中，可以采用如下的方法得出用于B帧当前图像块直

5 接编码的运动矢量向量  $MV(x,y)$ ，

当前块的前向运动矢量  $MV_F$  采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ，

如果  $MV(i) < 0$ ：

10  $MV_F(i) = -(((scale\_factor / tp) - (tb \times scale\_factor / tp) \times MV(i) - 1)$   
 $>> shift\_len)$

否则，

$MV_F(i) = ((scale\_factor / tp) + (tb \times scale\_factor / tp) \times MV(i) - 1)$   
 $>> shift\_len$

当前块的后向运动矢量  $MV_B$  采用如下公式计算：

15 如果  $MV(i) < 0$ ：

$MV_B(i) = ((scale\_factor / tp) - (td \times scale\_factor / tp) \times MV(i) - 1)$   
 $>> shift\_len$

否则，

20  $MV_B(i) = -(((scale\_factor / tp) + (td \times scale\_factor / tp) \times MV(i) - 1)$   
 $>> shift\_len)$

其中  $scale\_factor$  值为小数放大因子， $shift\_len$  表示需要向右移位的次数， $MV_F$  与  $MV_B$  则是当前块相应的前向运动矢量与后向运动矢量， $tb$  是当前图片与前向参考图片在时域上的距离， $td$  是前向参考图片与后向参考图片在时域上的距离， $tp$  是当前图片与后向参考图片在时域上的距离。

25  $MV$  表示后向参考图片相应部分相对于前向参考帧的运动矢量， $MV(x,y) = (MV(x), MV(y))$  为二维矢量，其相应分量为  $MV(x), MV(y)$ ， $MV(i)$  代表  $MV(x)$

或  $MV(y)$ ;  $MV(x) < 0$  代表参考块位于参考帧中当前块相对应位置的左侧;  
 $MV(y) < 0$  代表代表参考块位于参考帧中当前块相对应位置的上方;  $a/b$  表示  
 为  $a$  与  $b$  的商向零方向取整。

- 5        3、根据权利要求 1 所述的直接编码模式下确定参考图像块的方法，其特征在于：所述的步骤二中，

当前块的前向运动矢量  $MV_F$  采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ,

如果  $MV(i) < 0$  :

$$10 \quad MV_F(i) = -(((scale\_factor / tp) - (tb \times scale\_factor / tp) \times MV(i)) \\ >> shift\_len)$$

否则，

$$MV_F(i) = ((scale\_factor / tp) + (tb \times scale\_factor / tp) \times MV(i)) \\ >> shift\_len$$

- 15        当前块的后向运动矢量  $MV_B$  采用如下公式计算：

如果  $MV(i) < 0$  :

$$MV_B(i) = ((scale\_factor / tp) - (td \times scale\_factor / tp) \times MV(i)) \\ >> shift\_len$$

否则，

$$20 \quad MV_B(i) = -(((scale\_factor / tp) + (td \times scale\_factor / tp) \times MV(i)) \\ >> shift\_len).$$

- 4、根据权利要求 2 所述的直接编码模式下确定参考图像块的方法，其特征在于：所述的步骤一之前，包括将步骤二中用到的  $scale\_factor / tp$  ,  
 25  $tb \times scale\_factor / tp$ ,  $td / tp \times scale\_factor$  参数，预先计算出来，  
 将计算结果存储在一个表中，用来在步骤二的计算中直接调用。

- 
- 5、根据权利要求1或2所述的直接编码模式下确定参考图像块的方法，其特征在于：所述的步骤二中的 shift\_len 为自然数。
- 5 6、根据权利要求1所述的直接编码模式下确定参考图像块的方法，其特征在于：所述的步骤一获取后向参考帧相应块的运动矢量，包括如下过程：  
选取后向参考P帧中与B帧中待编码宏块位置相同的宏块作为对应宏块，并获取P帧该宏块的运动矢量。

## 直接编码模式下确定参考图像块的方法

### 技术领域

- 5 本发明涉及一种用于视频编码的确定参考图像块的方法，尤其是一种直接（Direct）编码模式下确定参考图像块的方法；属于图像视频编/解码技术领域。

### 背景技术

- 10 高效的视频编解码技术是实现高质量、低成本多媒体数据存储与传输关键。现在比较流行的图像编码国际标准都是基于这种编码理论，采用的是基于块匹配的运动补偿、离散余弦变换和量化相结合的编码方法。典型的有国际标准化组织/国际电工技术委员会第一联合技术组（ISO/IEC JTC1）推出的 MPEG-1，MPEG-2 和 MPEG-4 等国际标准，以及国际电信联盟（ITU-T）  
15 提出的 H.26x 系列推荐。这些视频编码标准在工业界得到了广泛应用。

- 这些视频编码标准都采用了混合视频编码（Hybrid Video Coding）策略，通常包括：预测、变换、量化和信息熵编码等四个主要模块。预测模块的主要功能是利用已经编码并重建的图像对当前要编码的图像进行预测，即帧间预测，或者利用图像中已经编码并重建的图像块或宏块对当前要编码的  
20 图像块或宏块进行预测，即帧内预测。

- 基于运动补偿的预测的主要功能是消除视频序列在时间上的冗余。视频编码的大部分编码效率来自于预测模块。视频编码过程就是对视频序列的每一帧图像进行编码的过程。常用的视频编码系统对每一帧图像的编码是以宏块为基本单元进行的。对当前宏块进行编码时，从当前宏块与参考块之间  
25 通过运动矢量连接。在编码每一帧图像时，又可以分为内编码 I 帧、预测 P 帧编码和双向预测 B 帧编码等情况。一般来说，编码时，I 帧、P 帧和 B 帧编码是穿插进行的，比如按照 IBBPBBP 的顺序。

由于 B 帧技术需要同时进行前向与后向的运动估计，因此需要较高的运算复杂度，同时为了区分前后向运动矢量要引入额外的标识信息。而采用 Direct 编码模式，其前后向运动矢量由有后向参考图片的运动矢量推导得到，可以不对运动矢量信息进行编码，因此可以减少编码运动矢量信息的比特，有效地提高编码效率。

现有技术中直接编码模式 (direct mode) 运动矢量的推导以如下的方式实现，：

$$MV_F = td / tp \times MV$$

$$MV_B = (tb - tp) / tp \times MV$$

$MV_F$  与  $MV_B$  则是当前块相应的前向运动矢量与后向运动矢量。这里  $tb$  是当前图片与前向参考图片在时域上的距离， $td$  是前向参考图片与后向参考图片在时域上的距离， $MV$  表示后向参考图片相应部分相对于前向参考帧的运动矢量。

从以上公式可以看出  $MV_F$ 、 $MV_B$  的值可能不是整数，如果带运动矢量有小数部分，由于运动矢量是以像素为单位的，则不可能在参考图片中找到运动矢量所指的匹配块，因此，需要对  $MV_F$ 、 $MV_B$  的计算值进行处理，确定为整数的运动矢量。

例如：在 H.264/AVC 中，direct mode 运动向量  $MV_F$ 、 $MV_B$  基于如下公式计算得到：

$$Z = (tb \times 256) / tp$$

$$W = Z - 256$$

$$MV_F = (Z \times MV + 128) \gg 8$$

$$MV_B = (W \times MV + 128) \gg 8$$

其中 “ $\gg 8$ ” 表示右移 8 位。

但是，这种舍入方法仍不能精确地得到当前块的预测块，使得根据获得的运动矢量指向的图像块出现偏差，严重的影响 direct mode 运动矢量推

导的精确性，而 direct mode 在 B 帧所用的模式中经常会占 70-80%，从而使得 B 帧的压缩效率大大降低。因此，如何更好地确定 direct mode 的运动矢量，成为视频压缩领域的一个难题。

## 5 发明内容

本发明所要解决的技术问题在于：提出一种 Direct 模式下，视频编码中确定运动矢量的方法，可以更准确的进行运动矢量预测，提高压缩效率。

本发明的技术方案，包括：

步骤一、获取 B 帧后向参考帧中相对于当前图像块的运动矢量；

10 步骤二、根据已经得到的后向参考帧中对应图像块的运动矢量，得出用于 B 帧当前图像块直接编码的运动矢量向量  $MV(x,y)$ ，

当前块的前向运动矢量  $MV_f$  采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ， $td = tp - tb$ ；

如果  $mv(i) < 0$ ：

15  $MV_f(i) = -(((scale\_factor / tp) \times (1 - mv(i) \times tb) - 1) \gg shift\_len)$

否则，

$MV_f(i) = ((scale\_factor / tp) \times (1 + mv(i) \times tb) - 1) \gg shift\_len)$

当前块的后向运动矢量  $MV_b$  采用如下公式计算：

如果  $mv(i) < 0$ ：

20  $MV_b(i) = ((scale\_factor / tp) \times (1 - mv(i) \times td) - 1) \gg shift\_len$

否则，

$MV_b(i) = -(((scale\_factor / tp) \times (1 + mv(i) \times td) - 1) \gg shift\_len)$

b)

其中  $scale\_factor$  值为小数放大因子， $shift\_len$  表示需要向右移位的  
25 次数， $MV_f$  与  $MV_b$  则是当前块相应的前向运动矢量与后向运动矢量， $tb$  是当前图片与前向参考图片在时域上的距离， $td$  是前向参考图片与后向参考图



片在时域上的距离；

$MV$ 表示后向参考图片相应部分相对于前向参考帧的运动矢量， $MV(x,y) = (MV(x), MV(y))$ 为二维矢量，其相应分量为 $MV(x), MV(y)$ ， $MV(i)$ 代表 $MV(x)$ 或 $MV(y)$ ； $a/b$ 表示为 $a$ 与 $b$ 的商向零方向取整；

- 5        步骤三、步骤二得到的运动矢量指向的前向和后向图像块为当前图像块的参考图像块。

也可以将以上的公式中，“ $1 \pm mv(i) \times tb) - 1$ ”改为“ $1 \pm mv(i) \times tb)$ ”结果相差不多，但是一般情况下，采用“ $1 \pm mv(i) \times tb) - 1$ ”的编码效果更好。

- 10        本发明通过本发明提出了一种能够“零方向”舍入的方法能够很好地解决在保持精确的运动矢量的同时，又可以通过无除法的方式来实现，从而提高运动矢量的计算的精度，可以更真实地体现视频中物体的运动，获得更准确的运动矢量预测，与前向预测编码，后向预测编码相结合，可用实现一种新的预测编码类型，既可以保证 direct mode 编码高效性又便于硬件的
- 15        实现，取得与传统 B 帧编码相似的效果，可以用于正在制定的 AVS 标准。

### 附图说明

图 1 是本发明 Direct 模式下应用的示意图；

图 2 是本发明技术方案的流程图。

20

### 具体实施方式

下面结合附图和实施例对本发明的技术方案进一步详细说明：

- 图 1 是本发明技术方案应用方式示意图，图中表示的是 direct mode 运动矢量的推导过程，这里  $tb$  是当前图片与前向参考图片在时域上的距离，
- 25         $td$  是前向参考图片与后向参考图片在时域上的距离，图中 1 表示当前编码块，2 表示 P 帧相同位置的对应块。

首先，给出一些描述用到的定义：

定义  $a/b$  表示为  $a$  与  $b$  的商向零方向取整，如： $15/7=2$ ， $-15/7=-2$ ；再定义  $MV(x,y) = (MV(x), MV(y))$  为二维矢量，其相应分量为  $MV(x), MV(y)$ ， $MV(i)$  代表  $MV(x)$  或  $MV(y)$ 。

参考图 2，在编码端 direct 模式下，B 前后向参考块的确定，包括如下步骤：

步骤一、获取 B 帧后向参考帧中相对于当前图像块的运动矢量；

步骤二、根据已经得到的后向参考帧中对应图像块的运动矢量，得出用于 B 帧当前图像块 direct mode 运动矢量推导的向量  $MV(x,y)$ ，

当前块的前向运动矢量  $MV_f$  采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ， $td = tp - tb$ ；

如果  $MV(i) < 0$ ：

$$MV_f(i) = -(((scale\_factor / tp) \times (1 - MV(i) \times tb) - 1) \gg shift\_len)$$

否则，

$$MV_f(i) = ((scale\_factor / tp) \times (1 + MV(i) \times tb) - 1) \gg shift\_len$$

当前块的后向运动矢量  $MV_b$  采用如下公式计算：

如果  $MV(i) < 0$ ：

$$MV_b(i) = ((scale\_factor / tp) \times (1 - MV(i) \times td) - 1) \gg shift\_len$$

否则，

$$MV_b(i) = -(((scale\_factor / tp) \times (1 + MV(i) \times td) - 1) \gg shift\_len)$$

其中  $scale\_factor$  值为小数放大因子， $scale\_factor = 2^{shift\_len}$ ， $shift\_len$  为自然数，表示需要向右移位的次数。 $scale\_factor$  值越大，越能精确地保持公式中的运算结果；

步骤三、步骤二得到的运动矢量指向的前向和后向图像块为当前图像块的参考图像块。

所述的步骤二中，可以采用如下的方法得出用于 B 帧当前图像块直接编

码的运动矢量向量  $MV(x,y)$ ,

当前块的前向运动矢量  $MV_F$  采用如下公式计算:

设  $scale\_factor = 2^{shift\_len}$ ,

如果  $MV(i) < 0$  :

$$MV_F(i) = -(((scale\_factor / tp) - (tb \times scale\_factor / tp) \times MV(i) - 1)$$

>> shift\_len )

否则,

$$MV_F(i) = ((scale\_factor / tp) + (tb \times scale\_factor / tp) \times MV(i) - 1)$$

>> shift\_len

当前块的后向运动矢量  $MV_B$  采用如下公式计算:

如果  $MV(i) < 0$  :

$$MV_B(i) = ((scale\_factor / tp) - (td \times scale\_factor / tp) \times MV(i) - 1)$$

>> shift\_len

否则,

$$MV_B(i) = -(((scale\_factor / tp) + (td \times scale\_factor / tp) \times MV(i) - 1)$$

>> shift\_len)

其中  $scale\_factor$  值为小数放大因子,  $shift\_len$  表示需要向右移位的次数,  $MV_F$  与  $MV_B$  则是当前块相应的前向运动矢量与后向运动矢量,  $tb$  是当前图片与前向参考图片在时域上的距离,  $td$  是前向参考图片与后向参考图片在时域上的距离,  $tp$  是当前图片与后向参考图片在时域上的距离。

$MV$  表示后向参考图片相应部分相对于前向参考帧的运动矢量,  $MV(x,y) = (MV(x), MV(y))$  为二维矢量, 其相应分量为  $MV(x), MV(y)$ ,  $MV(i)$  代表  $MV(x)$  或  $MV(y)$ ;  $MV(x) < 0$  代表参考块位于参考帧中当前块相对应位置的左侧;  $MV(y) < 0$  代表代表参考块位于参考帧中当前块相对应位置的上方;  $a/b$  表示为  $a$  与  $b$  的商向零方向取整。

所述的步骤二中，

当前块的前向运动矢量  $MV_F$  还可以采用如下公式计算：

设  $scale\_factor = 2^{shift\_len}$ ，

如果  $MV(i) < 0$ ：

$$MV_F(i) = -(((scale\_factor / tp) - (tb \times scale\_factor / tp) \times MV(i))$$

>> shift\_len )

否则，

$$MV_F(i) = ((scale\_factor / tp) + (tb \times scale\_factor / tp) \times MV(i))$$

>> shift\_len

当前块的后向运动矢量  $MV_B$  采用如下公式计算：

如果  $MV(i) < 0$ ：

$$MV_B(i) = ((scale\_factor / tp) - (td \times scale\_factor / tp) \times MV(i))$$

>> shift\_len

否则，

$$MV_B(i) = -(((scale\_factor / tp) + (td \times scale\_factor / tp) \times MV(i))$$

>> shift\_len)。

因为  $scale\_factor$  是一个放大因子，为了避免除法，可以通过先把被除数放大，然后进行右移来减少精度的损失，这是本发明的主要思想。

由于  $tp$ 、 $tb$  的取值有一定的范围，如一般而言， $tp$  和  $tb$  都不会大于 10；所以可以  $scale\_factor / tp$ ， $tb \times scale\_factor / tp$ ， $td / tp \times scale\_factor$  都可以预先计算出来，形成一个列表。计算需要时，直接从列表中读出所需值，以减少计算的复杂度，提高计算的速度。

也可以将以上的公式中，“ $tb \times scale\_factor / tp) \times MV(i) - 1$ ”改为“ $tb \times scale\_factor / tp) \times MV(i)$ ”结果相差不多，但是一般情况下，采用“ $tb \times scale\_factor / tp) \times MV(i) - 1$ ”的编码效果更好。

同时由于  $td = tp - tb$ ；需要的话可以进一步简化后向运动矢量的推导，

结果如下:

当前块的前向运动矢量  $MV_F$  采用如下公式计算:

设  $scale\_factor = 2^{shift\_len}$ ,

如果  $MV(i) < 0$  :

$MV_F(i) = -(((scale\_factor / tp) - (tb \times scale\_factor / tp) \times MV(i)) \gg shift\_len)$

否则,

$MV_F(i) = ((scale\_factor / tp) + (tb \times scale\_factor / tp) \times MV(i)) \gg shift\_len$

当前块的后向运动矢量  $MV_B$  采用如下公式计算:

如果  $MV(i) < 0$  :

$MV_B(i) = ((scale\_factor / tp) - (td \times scale\_factor / tp) \times MV(i)) \gg shift\_len$

否则,

$MV_B(i) = -(((scale\_factor / tp) + (td \times scale\_factor / tp) \times MV(i)) \gg shift\_len)$

也可以将以上的公式中, “ $1 \pm MV(i) \times tb) - 1$ ” 改为 “ $1 \pm MV(i) \times tb$ ” 结果相差不多, 但是一般情况下, 采用 “ $1 \pm MV(i) \times tb) - 1$ ” 的编码效果更好。

下面通过一个具体实施进一步说明本发明的技术方案:

当  $tb=1, td=2, MV = (5, -7)$  时 则  $tp = 3$

在 H. 264/AVC 中:

$$Z = (tb \times 256) / tp = 85$$

$$W = Z - 256 = -171$$

$$MV_F(x) = (Z \times MV(x) + 128) \gg 8 = (Z \times 5 + 128) \gg 8 = 2;$$

$$MV_B(x) = (W \times MV(x) + 128) \gg 8 = (W \times 5 + 128) \gg 8 = -3;$$

$$MV_F(y) = (Z \times MV(y) + 128) \gg 8 = (Z \times -7 + 128) \gg 8 = 2;$$

$$MV_B(y) = (W \times MV(y) + 128) \gg 8 = (W \times -7 + 128) \gg 8 = 5;$$

用本发明的方法

这里  $scale\_factor = 256$ , 则  $shift\_len = 8$ ;

对于前向运动矢量:

因为  $MV(x) = 5 > 0$

$$MV_F(x) = ((256 / 3) + (1 \times 256 / 3) \times 5) \gg 8 = 1$$

因为  $MV(y) = -7 < 0$

$$MV_F(y) = -(((256 / 3) - (1 \times 256 / 3) \times -7) \gg 8) = -2$$

对于后向运动矢量:

因为  $MV(x) = 5 > 0$

$$MV_B(x) = -(((256 / 3) + (2 \times 256 / 3) \times 5) \gg 8) = -3$$

因为  $MV(y) = -7 < 0$

$$MV_B(y) = ((256 / 3) - (2 \times 256 / 3) \times -7) \gg 8 = 4.$$

由于视频序列时域上相近的图片之间存在着较大的时域相关性, 同时由于相邻图片间将物体的运动大部分保持匀速和较小偏移的平移运动, 因此本发明采用如上方式, 尽量减少了舍入误差造成运动矢量推导的影响, 这样可以得到更好的预测参考块。

最后所应说明的是: 以上实施例仅用以说明而非限制本发明的技术方案, 尽管参照上述实施例对本发明进行了详细说明, 本领域的普通技术人员应当理解: 依然可以对本发明进行修改或者等同替换, 而不脱离本发明的精神和范围的任何修改或局部替换, 其均应涵盖在本发明的权利要求范围当中。

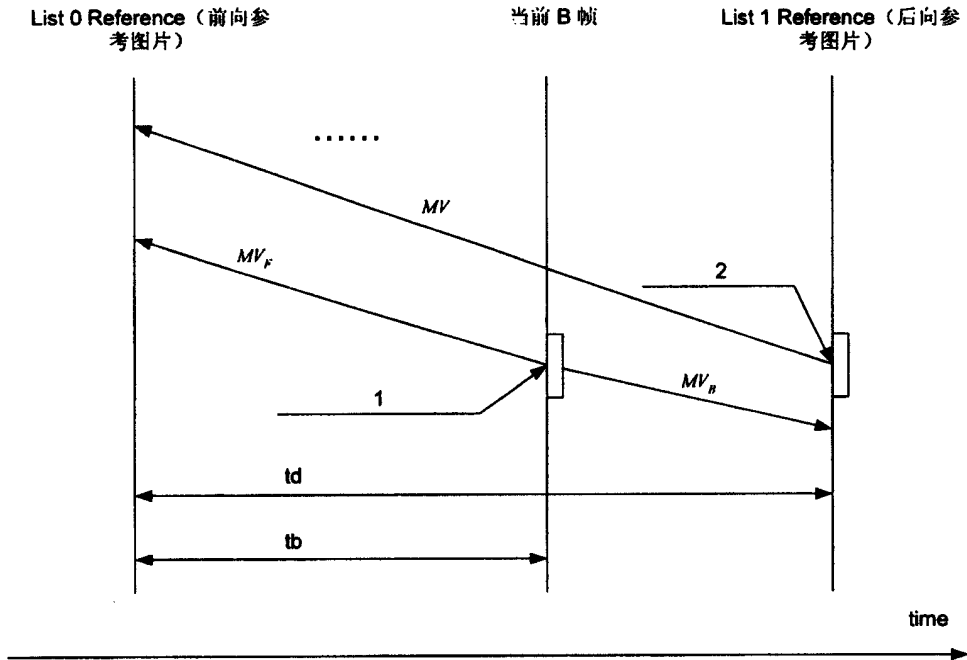


图 1

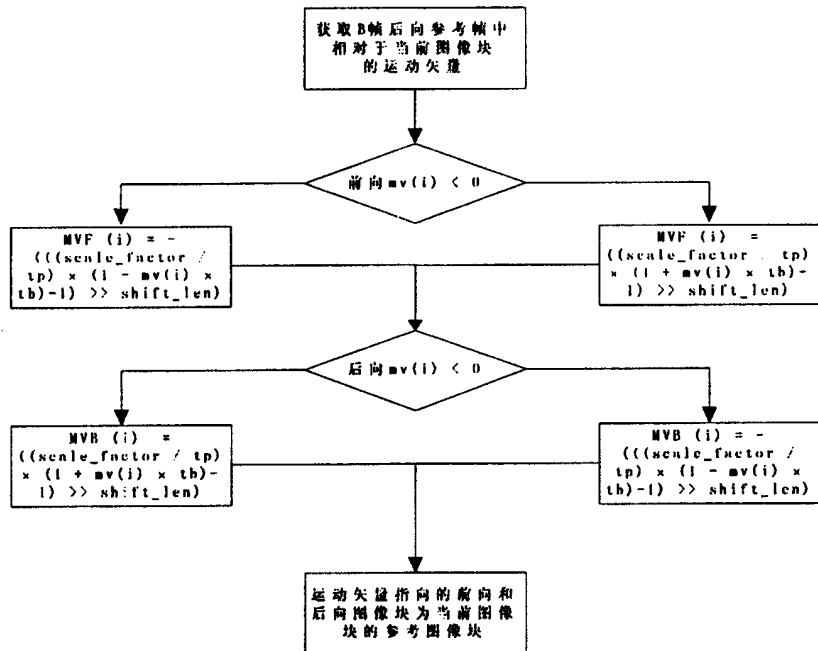


图 2