

VVC ME 电路设计规范

1. VVC ME 简介

多功能视频编码（Versatile Video Coding，简称 VVC），也称为 H.266、MPEG-I 第 3 部分或未来视频编码（FVC），是由联合视频专家组（JVET），于 2020 年 7 月 6 日最终确定的视频压缩标准。它是高效视频编码（HEVC，也称为 ITU-T H.265 和 MPEG-H 第 2 部分）的后继标准。

H.266/VVC 标准广泛采用了帧间预测技术来提高编码效率。帧间预测技术的基本思想是，根据组成视频的连续图像之间的时域相关性，通过将已经编码的图像作为当前编码图像的参考图像，使用运动矢量表示当前编码块在参考图像中运动的相对位置，并记录参考图像的索引，达到除去连续图像的时域相关性，并且提高编码效率的目的。

运动估计（Motion Estimation，简称 ME）是帧间预测技术中最重要的环节之一。大多数视频序列中，相邻图像内容非常相似，仅是一些运动物体的位置发生了变化。因此，对于运动物体，可以通过仅对其运动信息编码来减少编码量。ME 就是指提取当前图像运动信息的过程。基于块的运动表示法（将图像分成大小不同的像素块，只要块大小选择合适，则各个块的运动形式可以看成是统一的）兼顾了运动估计精度和复杂度，是历代视频编码国际标准的核心技术。

2. 课程设计目标

面向 H.266/VVC 标准，设计支持 4K 视频的 ME 电路，完成电路的架构设计、Verilog 代码设计、逻辑仿真、性能分析、逻辑综合、时序分析与验证和物理设计，同时进行结果分析比较。

3. 课程设计指标及应用要求

- 1) 每秒 60 帧 4K 视频（3840×2160@60fps）的实时处理能力
- 2) 采用全搜索 ME 算法、支持 8×8 块大小的 SAD 计算、搜索区间为[-7,8]
- 3) 芯片设计工艺：华力 55nm 工艺
- 4) 评价指标：电路的实时处理能力、芯片的 PPA（Performance or frequency, Power, Area）、输入/输出数据的带宽及其利用效率

4. ME 算法介绍

由于时间上的相关性，视频每一帧图像与相邻的若干帧图像之间存在一定的信息冗余量，运动估计的目的便是找出冗余的信息，从而将冗余的信息消除，这样就能够使得传输与存储的信息大幅度的降低，提升压缩率。

4.1 基于块匹配的运动估计算法

基于块匹配的运动估计算法思路如图 1 所示。当前时刻的视频帧称为当前帧，上一时刻的视频帧称为先前帧。在当前帧中的某个参考块，可以用先前帧中的某个候选块以及二者之间的坐标差（运动向量）来表示。

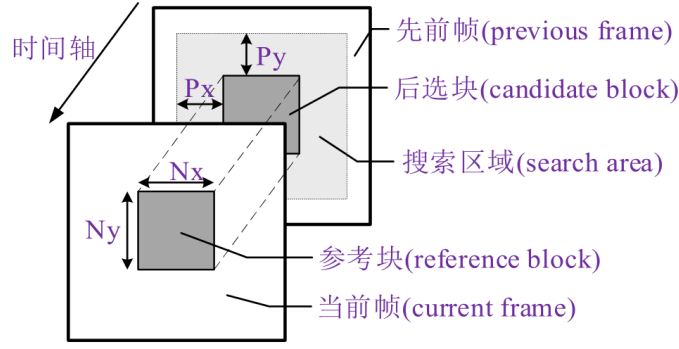


图 1 运动估计块匹配示意图

在进行运动估计时，首先在先前帧中，找到和参考块具有相同坐标的一个候选块。

其次，在该候选块附近确定搜索区域。以该候选块左上角为坐标原点，假定水平方向搜索范围为 $[-Px, Px]$ ，垂直方向搜索范围为 $[-Py, Py]$ ，则左上角横坐标在 $-Px \sim Px$ 范围内，纵坐标在 $-Py \sim Py$ 范围内的块称作候选块。由 $-Px \sim Px$ 以及 $-Py \sim Py$ 构成的区间为搜索区间。

之后，需要在搜索区间内寻找和参考块匹配程度最高的块，并以该块右上角坐标为运动向量(Motion Vector, 简称 MV)。

4.2 匹配准则

在搜索区域内进行匹配时，匹配准则为参考块和候选块之间的像素亮度值的累计绝对误差。假定每个块的大小为 $N \times N$ ，搜索区域为 $[-p, p]$ （水平、垂直方向均为 $[-p, p]$ ）。令 $x(i, j)$ 表示当前帧在坐标 (i, j) 处的像素亮度值， $y(i+m, j+n)$ 表示先前帧在坐标 $(i+m, j+n)$ 处的像素亮度值，则二者之间的向量为 (m, n) ，且二者之间的像素亮度值的累计绝对误差 (SAD) 可表示为：

$$SAD(m, n) = \sum_{i=0}^{N-1} \sum_{j=0}^{N-1} |x(i, j) - y(i+m, j+n)|$$

$$-p \leq m, n \leq p$$

在搜索区域 $[-p, p]$ 内找到所有候选块的对应 SAD 值，其中对应最小 SAD 值的 (m, n) 即为该参考块的运动向量，即

$$MV = \arg \{ \min SAD(m, n) \} \quad -p \leq m, n \leq p$$

4.3 搜索方式

在搜索方式中，全搜索块匹配算法(Full- Search Block Matching Algorithm, FSBM)是一种常见的方式。它通过对搜索区域进行穷尽式匹配比较来找到最佳的搜索匹配块。虽然 FSBM 算法计算量巨大，但由于其匹配效果好而在视频点播、数字电视等对视频图象质量要求很高的领域得到广泛的应用。

图 2 展示了一个 FSBM 的例子。假定每个块大小为 2×2 ，搜索区域为 $[-1, 1]$ 。对于左上角坐标为 $(2, 2)$ 的参考块，一共可以找到 9 个不同的候选块。对于每一个候选块，计算其 SAD 值，并找出最小 SAD (m, n) ，该参考块运动向量为 (m, n) 。

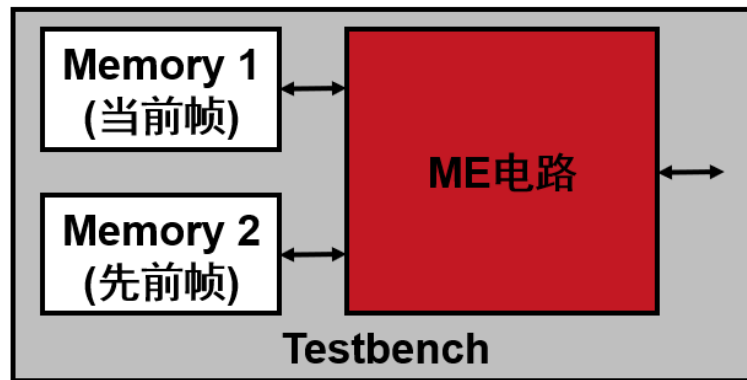


图 4 Testbench 框图

为了方便各位验证，助教将会提供当前帧和先前帧像素值以及对应的 ME 电路参考运算结果。

(1) 输入数据

- a) 当前帧数据: `cur_img.txt`;
- b) 先前帧数据: `pre_img.txt`
- c) 数据格式
 - i. 数据按逐行的方式保存，例如 4k 目录下，每一帧为 3840*2160，则相应文件中前 3840 行数据对应图像第一行像素点的值
 - ii. 输入数据读取顺序和读取速度可根据自己的设计需要自行决定（可用脚本预处理）
 - iii. Testbench 中可用 `$readmemh`, `$readmemb`, `$fopen` 等方式读取文件

(2) 参考输出结果

- a) `motion_vector_x.txt` 和 `motion_vector_y.txt` 分别为运动向量的横向和纵向分量。同一个搜索区域内，可能多个候选块具有相同的、最小的 SAD 值，因此可能得到不同的运动向量。参考结果中，这种情况下总是选取最靠近左上的候选块用于计算运动向量。大家可根据自己电路的实现方式，选择自己合适的取舍方式（在报告中注明）。
- b) `sad.txt` 为每个参考块对应的 SAD 值。同学们最终得到的 SAD 值必须和该文件完全一致。
- c) 数据格式: 逐行方式保存，例如前 3840/8=480 行对应图像第一行 8×8 块的 SAD 值
- d) Testbench 中可用 `$fwrite` 等方式写文件

(3) 补充说明

- a) 对于边界处的参考块，搜索区域内可能没有像素值，以 0 值填补
- b) 4K 视频数据仿真验证可能速度较慢，前期可先用 `hall_qcif_176x144` 数据进行验证

6. 评价指标说明

电路的处理性能指电路每秒钟能够处理的视频帧数。例如，假定电路的时钟周期为 10ns，每周期能得到一个 4×4 参考块的运动向量，则对于 1920×1080 的视频，完成当前帧中所有参考块的运动向量计算需要 480×270 个时钟周期。那么，每秒钟能处理的视频帧数为： $1 \text{ 帧} / (480 \times 270 \times 10\text{ns}) = 772 \text{ 帧/s}$ 。同学们可根据指标和上述计算方法，合理确定电路每周期的数据处理量。

电路的频率、面积和功耗以完成物理设计之后的值为准。

电路的带宽指的是单位时间里，电路的输入或者输出的数据量。例如电路时钟周期为 10ns，电路每周期需要读取的数据量为 30bit，那么该电路的输入带宽为 $30\text{bit}/10\text{ns}=3\text{Gb/s}$

带宽的利用效率指单位时间里实际利用数据量除以带宽。例如，电路时钟周期为 10ns，每 10 个周期读取一次数据，每次读取数据量为 30bit，那么电路仍然需要 3Gb/s 的峰值带宽，然而实际利用率为 10%。