

一种快速块运动估计的分级筛选方法¹

陈韩锋 戚飞虎

(上海交通大学计算机科学与工程系 上海 200030)

摘要: 该文提出了一种新的快速块匹配运动估计方法, 分级筛选法。该方法将搜索最佳匹配块的过程分为若干个筛选级别, 在初始的级别中用很少的运算代价通过简单的特征匹配先淘汰一部分候选块; 然后在上一级剩余的候选块中, 逐级用更加细致的特征继续筛选; 直至找到最佳匹配块。实验结果表明, 在估计精度非常相近的前提下, 该文方法的速度是全搜索方法的 12~14 倍, 而且该方法有很好的稳定性。

关键词: 运动估计, 块匹配, 视频编码, 分级筛选

中图分类号: TN941 **文献标识码:** A **文章编号:** 1009-5896(2004)07-1076-06

A Multi-level Elimination Algorithm for Fast Block Motion Estimation

Chen Han-feng Qi Fei-hu

(Dept of Computer Sci. & Eng., Shanghai Jiaotong Univ., Shanghai 200030, China)

Abstract A multi-level elimination algorithm for fast motion estimation is proposed in this paper. The most matched block of the current block is detected in the next frame with a multi-level elimination strategy in this algorithm. In the first level, some candidate blocks are eliminated by simple characters based block-matching, needing only very few computation costs. Then in further levels, more detailed characters are used for block-matching to eliminate more candidate blocks until the most matched block is selected in the last level. Experimental results show that the proposed algorithm can work steadily and get very close performance to the full-search algorithm while a computation reduction of about 12~14 times, with respect to the full-search algorithm.

Key words Motion estimation, Block-matching, Video coding, Multi-level elimination

1 引言

在全搜索块匹配运动估计方法中, 大小为 $M \times N$ 的当前帧图像 I_0 被划分成一些大小为 $N_1 \times N_2$ 的矩形块, 然后为 I_0 的每个块 B_0 在下一个相邻帧 I_1 中搜索最佳匹配块 B_1 , B_0 和 B_1 之间的位置偏差即为块 B_0 中所有像素的运动矢量。由于连续两帧图像之间物体的位移量有限, 搜索范围通常限制在一个 $(2M_1 + 1) \times (2M_2 + 1)$ 的窗口内。本文采用 SAD 作为块的最佳匹配准则。SAD 定义为

$$\text{SAD}_{i,j}(d_1, d_2) = \sum_{m=0}^{N_1-1} \sum_{n=0}^{N_2-1} |I_0(i+m, j+n) - I_1(i+m+d_1, j+n+d_2)| \quad (1)$$

其中 $\text{SAD}_{i,j}(d_1, d_2)$ 为块 $B_0(i, j)$ 和 $B_1(i+d_1, j+d_2)$ 之间的匹配误差, $\text{SAD}_{i,j}(d_1, d_2)$ 越小说明这两个块相似度越大; $B_0(i, j)$ 为 I_0 中左下角坐标为 (i, j) 的块, $B_1(i+d_1, j+d_2)$ 为 I_1 中左下角坐标为 $(i+d_1, j+d_2)$ 的块, (d_1, d_2) 为块 B_0 和 B_1 之间的位置偏差, $I_0(x, y)$ 和

¹ 2003-03-05 收到, 2003-08-04 改回

国家自然科学基金资助项目 (No.60072029)

$I_1(x, y)$ 分别为 I_0 和 I_1 中第 x 行第 y 列像素的灰度值。于是, 块 $B_0(i, j)$ 的运动矢量由下式给出:

$$V(i, j) = (\hat{d}_1, \hat{d}_2) = \arg \min_{(d_1, d_2)} \text{SAD}_{i,j}(d_1, d_2) \quad (2)$$

其中 $V(i, j) = (\hat{d}_1, \hat{d}_2)$ 为当前块的运动矢量。 $B_1(i + \hat{d}_1, j + \hat{d}_2)$ 即为 $B_0(i, j)$ 在 I_1 中的最佳匹配块。

用全搜索法进行块匹配运动估计时, 运算量非常庞大。于是人们提出了一些快速算法来实现块匹配运动估计, 这些快速算法主要是通过 4 种途径来减少运算量。第 1 种途径是将最佳匹配块的搜索过程限制在一条匹配误差逐步递减的路径上, 最具代表性的是三步搜索法^[1]、新三步搜索法 N3SS^[2]、四步搜索法 4SS^[3] 等。这类方法^[1-3] 是基于这样一个假设 (本文简称为单调递减假设): 在一定的搜索范围内, 块的匹配误差呈现单调递减分布。当这个假设满足得不好时, 估计结果与全搜索方法的差距会比较大。第 2 种途径是通过对运动矢量场的采样来减少运算量^[4], 即先估计运动矢量场中部分块的运动矢量, 其他块的运动矢量则从其相邻块的矢量中挑选。第 3 种途径是通过将每个匹配块分成若干个子块来减少计算两个匹配块的匹配误差所需的运算量^[5]。第 4 种途径是从数学不等式关系入手, 在不影响估计精度的前提下提高估计速度^[6,7]。文献 [8] 中提出了一种结合第 3 和第 4 种方法的快速运动估计方法, 其基本依据是: 子块匹配误差之和总是小于或等于更小子块的匹配误差之和。它是一种无损提速方法, 因此提速空间有限。

本文提出了一种新的快速块匹配运动估计方法——分级筛选方法。该方法的基本依据是: 最佳匹配块的子块匹配误差之和虽然不一定比其他所有候选块的子块匹配误差之和都小, 但总是比较小的; 也就是说子块匹配误差之和比较大的候选块通常不会是最佳匹配块。

2 基于子块特征匹配的分级筛选方法

2.1 基于子块特征匹配的块匹配准则

在式 (1) 中, 如果将每个 $N_1 \times N_2$ 块都看作是一个元素集合, 则两个块之间的相似度是通过两个集合中的对应元素之间的绝对差值之和来衡量的。本文将这些元素称之为基元, 每个基元都具有一个特征值, 用来衡量两个基元之间的差异。这样, 式 (1) 可以改写为

$$\text{SAD}_{i,j}(d_1, d_2) = \sum_{k=0}^K |E_0(i, j, k) - E_1(i + d_1, j + d_2, k)| \quad (3)$$

其中 K 为每个匹配块中基元的个数, $E_0(i, j, k)$ 为块 $B_0(i, j)$ 的第 k 个基元的特征值, $E_1(i + d_1, j + d_2, k)$ 为块 $B_1(i + d_1, j + d_2)$ 的第 k 个基元的特征值。考虑到计算量的问题, 本文后面提到的基元特征值都指基元中所有像素的灰度之和。基于子块特征匹配的运动估计过程就是: 将每个匹配块分解成一些子块, 然后以子块作为基元, 并用式 (3) 计算每个候选匹配块与当前块的匹配误差 (即相似度), 匹配误差最小的块与当前块之间的位置偏移矢量即为当前块的运动矢量。从式 (3) 可以看出, 如果基元的数量越少, 计算块匹配误差的运算量就越少, 块匹配运动估计的速度也就越快。因此, 如果将每个 $N_1 \times N_2$ 匹配块分成几个子块, 再用一个特征量来描述子块, 并且用这些子块作为基元来计算两个匹配块的匹配误差, 则可以减少块匹配运动估计的运算量。

那么, 将不同大小的子块作为基元时, 运动估计的准确性会有什么不同呢? 对于一个 16×16 的块, 将 $16 \times 16, 8 \times 8, 4 \times 4, 2 \times 2, 1 \times 1$ 大小的子块分别定义为 1 级, 2 级, 3 级, 4 级, 5 级子块。子块越小, 则用式 (3) 和式 (2) 搜索最佳匹配块的准确性 (即符合度) 将越高。表 1 (搜索范围为 15×15) 给出了针对 10 个随机选取的视频的一组实验数据来说明这一点, 表 1 中的每一行分别表示将 $16 \times 16, 8 \times 8, 4 \times 4, 2 \times 2$ 大小的子块作为基元时 (每个匹配块内对应的基元

个数 K 分别为 1, 4, 16, 64), 运动估计结果与全搜索方法估计结果的符合度。从表 1 中可以看出, 子块级别越高, 符合度也就越高。当子块级别为最高的 5 级时, 基于子块特征匹配的运动估计就退化为经典的全搜索法。

表 1 4 种不同大小子块下基于子块特征匹配的块匹配算法与全搜索方法的符合度

视频序列号	1	2	3	4	5	6	7	8	9	10
符合度 (%) (16×16)	9.6	3.5	2.6	1.6	5.8	11.1	12.2	9.5	6.4	2.7
符合度 (%) (8×8)	50.4	57.7	54.4	57.1	66.9	74.6	70.8	68.3	67.1	32.7
符合度 (%) (4×4)	79.0	81.1	68.0	75.4	81.0	86.6	83.7	82.9	83.6	68.3
符合度 (%) (2×2)	95.8	88.8	89.0	88.9	90.8	95.2	91.4	93.7	94.3	77.3

2.2 快速块运动估计的分级筛选方法

正如 2.1 节所分析的, 将子块作为基元, 在减少了块匹配的运算量的同时也降低了最佳匹配块的搜索准确性。但是, 如果放宽符合度的定义, 则基于子块特征匹配的方法和全搜索方法的符合度将会得到提高。放宽后的“符合”定义为: 用式 (3) 计算所有 $(2M_1 + 1) \times (2M_2 + 1)$ 个候选匹配块与当前块的匹配误差, 如果理论最佳匹配块为匹配误差最小的 S 个候选块之一, 则认为在该块上这种方法与全搜索方法是符合的。 S 称为放宽因子。表 2 给出了增加放宽因子后, 基于子块特征匹配的方法和全搜索方法的符合度的一组测试数据。表 2 中的 10 个视频与表 1 中的相同, 表 2 中的每一行分别表示将相应级别的子块作为基元时, 基于子块特征匹配的方法与全搜索方法估计结果的符合度。

表 2 增加放宽因子后基于子块特征匹配的块匹配算法与全搜索方法的符合度

	放宽因子 S	视频序列号									
		1	2	3	4	5	6	7	8	9	10
符合度 (%) (16×16)	150	98.5	99.8	98.1	100	98.8	100	99.7	100	100	98.9
符合度 (%) (8×8)	64	96.5	100	98.0	100	98.9	100	99.4	100	99.7	98.9
符合度 (%) (4×4)	16	96.9	99.8	98.9	100	98.5	100	99.0	98.4	100	98.5
符合度 (%) (2×2)	4	99.6	100	99.8	100	98.5	100	99.0	96.8	100	98.5

从表 2 中可以看出, 当放宽指数达到一定值以后, 将子块作为基元的方法能够和全搜索方法达到一个很高的符合度。而且对于越小的子块, 为确保很高的符合度所需的放宽因子也越小。另一方面, 在 2.1 节中已经提到, 子块越大, 用式 (3) 估计两个匹配块的运算量也就越小。由此可以想到, 如果先用较大的子块作为基元并设置足够大的放宽因子来排除一些不可能是最佳匹配块的候选块, 然后在剩下的候选块中再用更小的子块作为基元, 来进一步排除候选块, 直至找到最后的最佳匹配块, 这样既可以确保估计精度又可以提高估计速度。本文将这种通过逐步减小基元大小来分级排除候选块的方法称之为分级筛选方法。

在介绍分级筛选方法的详细步骤之前, 先说明一下基元的特征值的计算方法。设 $F_i(m, n)$ 为左下角坐标为 (m, n) 的第 i 级子块内所有像素的灰度之和, $i = 1, 2, 3, 4, 5$, 则

$$F_i(m, n) = F_{i+1}(m, n) + F_{i+1}(m + 2^{4-i}, n) + F_{i+1}(m, n + 2^{4-i}) + F_{i+1}(m + 2^{4-i}, n + 2^{4-i}),$$

$$i = 1, 2, 3, 4 \quad (4)$$

$$F_5(m, n) = \begin{cases} I_0(m, n), & I_0 \text{ 中的块} \\ I_1(m, n), & I_1 \text{ 中的块} \end{cases} \quad (5)$$

需要说明的是, 这里的 F 跟式 (3) 中的 E 实际上是同一个概念, 只不过描述方式有所不同。

块运动估计的分级筛选方法的主要过程为:

(1) 用式 (4), 式 (5) 计算 I_0 和 I_1 中左下角坐标为 (m, n) 的基元的特征值 $F_i(m, n)$;

(2) 将基元设为第 1 级子块 (16×16), 然后根据式 (3) 计算当前块 $B_0(i, j)$ 和 I_1 中搜索范围内所有候选块之间的匹配误差, 计算这些匹配误差的平均值 a_1 , 淘汰那些匹配误差大于 a_1 的候选块, 即认为它们不可能是最佳匹配块;

(3) 将基元设为第 2 级子块 (8×8), 然后重新根据式 (3) 计算当前块 $B_0(i, j)$ 和所有第 (2) 步中没有被淘汰的候选块之间的匹配误差, 计算这些匹配误差的平均值 a_2 , 再淘汰那些匹配误差大于 a_2 的候选块, 即认为它们也不可能是最佳匹配块;

(4) 将基元设为第 3 级子块 (4×4), 然后重新根据式 (3) 计算当前块 $B_0(i, j)$ 和所有第 (3) 步中没有被淘汰的候选块之间的匹配误差, 然后保留匹配误差最小的 S_1 个候选块 (通过实验表明 (表 2), $S_1 = 20$ 已经足够), 淘汰其余的候选块;

(5) 将基元设为第 4 级子块 (2×2), 重新根据式 (3) 计算当前块 $B_0(i, j)$ 和所有第 (4) 步中仍然没有被淘汰的候选块之间的匹配误差, 然后保留匹配误差最小的 S_2 个候选块 (通过实验表明 (表 2), $S_2 = 4$ 已经足够), 淘汰其余的候选块;

(6) 将基元设为第 5 级子块 (1×1), 根据式 (3) 计算当前块 $B_0(i, j)$ 和第 (5) 步中保留的 S_2 个候选块之间的匹配误差, 匹配误差最小的一个块即为 $B_0(i, j)$ 在 I_1 中的最佳匹配块, 这两个块之间的位置偏移量即为 $B_0(i, j)$ 处的运动矢量;

(7) 对 I_0 中不同的块 $B_0(i, j)$ 重复步骤 (2)–(6), 分别获得它们的运动矢量。

在步骤 (2) 和 (3) 中, 使用了匹配误差的均值来筛选候选块, 而不是像表 2 中那样用一个固定的放宽因子 S 来筛选, 这主要是考虑到用固定的放宽因子 S 来筛选时, 需要较大的排序运算开销。后面的实验结果将表明, 用来平均匹配误差筛选候选块可以获得和放宽因子类似好的筛选效果。表 2 已经说明, 当放宽因子达到一定值以后, 将子块作为基元的方法能够和全搜索方法达到一个很高的符合度。也就是说, 尽管分级筛选过程中的每一步都淘汰了很多候选匹配块, 但是最佳匹配块被淘汰的概率很小, 这就保证了最后得到的运动矢量的高准确性。

2.3 分级筛选方法的运算量

分级筛选方法的运算量主要集中在 3 个部分。一部分是计算每一帧中所有 $F_i(m, n)$ 所需的运算量 T_1 , 另一部分是计算基于基元的块匹配误差所需的运算量 T_2 , 还有一部分就是计算 a_1 、 a_2 所需的运算量。下面分别用 ‘+’, ‘-’, ‘|·|’ 来表示一次加法、一次减法、一次求绝对值运算所需的时间。

从式 (4), 式 (5) 不难看出, 对于大小为 $M \times N$ 的图像帧和 16×16 大小的匹配块

$$T_1 = 4 \times 3 \times M \times N \times \text{‘+’} \quad (6)$$

第 i 级子块作为基元时, 计算一个块匹配误差所需的运算量分别为 (依据式 (3))

$$T_{2i} = (4^{i-1} \times \text{‘-’} + 4^{i-1} \times \text{‘|·|’} + (4^{i-1} - 1) \times \text{‘+’})$$

设分级筛选方法的 (2)–(5) 步骤中, 候选块的保留率分别为 R_1, R_2, R_3, R_4 (所谓保留率是指, 该步骤过后未被淘汰的候选块数目与该步骤之前的候选块数目之间的比率), 于是

$$T_2 = \frac{M \times N}{16 \times 16} \times (2M_1 + 1) \times (2M_2 + 1) \times [T_{21} + R_1(T_{22} + R_2(T_{23} + R_3(T_{24} + R_4 \times T_{25})))]) \quad (7)$$

相比于 T_1 和 T_2 , T_3 非常小, 完全可以忽略不计。因此, 用分级筛选方法估计一帧中所有 16×16 块的运动矢量平均所需的运算量大约为

$$T_s \approx T_1 + T_2 = \frac{M \times N}{16 \times 16} \times (2M_1 + 1) \times (2M_2 + 1) \times T'_s \quad (8)$$

其中

$$T'_s = [T_{21} + R_1(T_{22} + R_2(T_{23} + R_3(T_{24} + R_4 \times T_{25})))] + \frac{12 \times 16 \times 16}{(2M_1 + 1) \times (2M_2 + 1)} \times \text{'+'} \quad (9)$$

以 $R_1 = R_2 = 0.5$, $R_3 = 0.36$ (此时, $S_1 = 20$), $R_4 = 0.2$ (此时, $S_4 = 4$), $M_1 = M_2 = 7$ 为例,

$$T'_s \approx 17.4 \times \text{'-'} + 17.4 \times \text{'|·|'} + 29.2 \times \text{'+'} \quad (10)$$

而全搜索方法的运算量为(根据式(1))

$$T_f = \frac{M \times N}{16 \times 16} \times (2M_1 + 1) \times (2M_2 + 1) T'_f \quad (11)$$

其中

$$T'_f = 256 \times \text{'-'} + 256 \times \text{'|·|'} + 255 \times \text{'+'} \quad (12)$$

比较式(10)和式(12)可以看出,在 $R_1 = R_2 = 0.5$, $R_3 = 0.36$, $R_4 = 0.2$, $M_1 = M_2 = 7$ 时,分级筛选方法的速度是全搜索方法的14~15倍。

3 实验结果

为了比较全搜索方法、三步搜索方法和本文提出的分级筛选法在运行速度和准确性方面的特性,本文分别对10组视频序列(Akiyo, Calender, Coastguard, Garden, Tennis, Foreman, Hall Monitor, Mother & Daughter, PuDong, Taxi)进行了块运动估计实验,这10组视频依次用序号1至10表示。其中, Akiyo, Calender, Coastguard, Garden, Tennis的帧大小为 352×240 , Foreman, Hall Monitor, Mother & Daughter的帧大小为 176×144 , PuDong的帧大小为 368×288 , Taxi的帧大小为 256×192 。实验的硬件环境为:赛扬633+256M内存;软件环境为:Windows2000+VC;程序中的计时函数采用VC中的clock()函数。此外,在本节的实验中, $N_1 = N_2 = 16$, $M_1 = M_2 = 7$, $S_1 = 20$, $S_2 = 4$ 。

表3给出了用三步搜索法和分级筛选法估计得到的运动矢量场与全搜索法估计得到的运动矢量场的符合度。需要说明的是,表3中符合度的含义是指2.1节中所定义的符合度(即严格符合)而非2.2节中放宽后的符合度。表4则给出了三种方法的平均匹配误差,平均匹配误差越小在一定程度上说明运动估计越准确,视频编码中通过运动补偿减少空间冗余的效率也就越高。从表3和表4中可以看出,在有些视频中,三步搜索法可以得到跟全搜索法相近的结果。但是在有些视频中(如Foreman, Garden, Mother & Daughter, Tennis),三步搜索法的效果就不理想,甚至很差,这主要是由于其基于匹配误差单调递减假设的缺陷。而本文提出的分级筛选法的估计结果不仅非常接近于全搜索方法,而且很稳定。表5中是三种方法的运算速度比较,以全搜索法为参照(速度为1),三步搜索法提高到了将近9倍,而分级搜索法能提高到12~14倍。

表3 分级筛选方法和三步搜索法与全搜索方法的符合度的比较

视频序列号	1	2	3	4	5	6	7	8	9	10
符合度(%) (三步搜索法)	99.4	96.8	99.6	77.8	85.9	99.8	96.0	94.0	96.4	51.6
符合度(%) (分级筛选法)	96.8	99.8	97.8	99.5	97.8	100	98.7	97.3	99.7	96.8

表4 分级筛选方法、三步搜索法、全搜索方法的平均匹配误差的比较

视频序列号	1	2	3	4	5	6	7	8	9	10
平均匹配误差(全搜索法)	1.120	9.450	6.343	2.159	4.506	2.469	1.980	2.934	2.002	5.649
平均匹配误差(三步搜索法)	1.126	9.581	6.356	2.505	4.905	2.469	1.990	2.996	2.010	7.742
平均匹配误差(分级筛选法)	1.148	9.457	6.361	2.159	4.527	2.469	1.981	2.970	2.006	5.696

表5 分级筛选方法、三步搜索法、全搜索方法的运算速度的比较

视频序列号	1	2	3	4	5	6	7	8	9	10
加速倍数 (全搜索法)	1	1	1	1	1	1	1	1	1	1
加速倍数 (三步搜索法)	8.78	8.89	8.86	8.74	8.92	8.75	8.94	8.73	8.77	8.85
加速倍数 (分级筛选法)	14.07	13.98	14.04	12.46	13.98	12.21	14.31	12.46	13.81	14.00

4 结论

本文提出了一种新的快速块匹配运动估计方法——分级筛选方法。该方法通过分级策略大大提高了运算速度,而且由于第一级筛选中考虑了所有候选块,因此不容易漏掉最佳匹配块,确保了方法的精确度。实验结果也表明,分级筛选方法能够在确保运动矢量估计精度与全搜索方法基本相近的情况下,使估计速度有很大的提高,而且该方法具有很好的稳定性和可靠性。本文的实验中所给出的数据是只采用了分级筛选方法后得到的结果。事实上,该方法还可以和引言中所介绍的第2和第4类方法结合使用,以进一步提高估计速度。此外,本文是以最常用的 16×16 大小的匹配块为例,来介绍分级筛选方法。对于 8×8 的匹配块,该方法同样适用,不同的仅仅是少一个级别的子块(16×16 子块),也就是省去2.2节中的步骤(2)。

参 考 文 献

- [1] Koga T, Linuma K, Hirano A, Lijima Y, Lshiguro T. Motion-compensated interframe coding for video conferencing. Proc. of NTC81, New Orleans, LA. 1981: 961-965.
- [2] Li Reoxiang, Zeng Bing, Liou M L. A new three-step search algorithm for block motion estimation. *IEEE Trans. on Circuits and Systems for Video Technology*, 1994, 4(4): 438-442.
- [3] Po Lai-Man, Ma Wing-Chung. A novel four-step search algorithm for fast block motion estimation. *IEEE Trans. on Circuits and Systems for Video Technology*, 1996, 6(3): 313-317.
- [4] Liu B, Zaccarin A. New fast algorithms for the estimation of block motion vectors. *IEEE Trans. on Circuits and Systems for Video Technology*, 1993, 3(2): 148-157.
- [5] Kim J S, Park R H. A fast feature-based block matching algorithm using integral projections. *IEEE Journal on Selected Areas in Communications*, 1992, 10(5): 968-971.
- [6] Li W, Salari E. Successive elimination algorithm for motion estimation. *IEEE Trans. on Image Processing*, 1995, IP-4(1): 105-107.
- [7] Hong W G, Oh T M, Ko S J. Fast motion-estimation algorithm based on progressive partial norm matching. *Electronics Letters*, 2001, 37(14): 890-892.
- [8] Gao X Q, Duanmu C S, Zou C R. A multilevel successive elimination algorithm for block matching motion estimation. *IEEE Trans. on Image Processing*, 2000, IP-9(3): 501-504.

陈韩锋: 男, 1976年生, 博士生, 研究方向为数字图像处理、视频分割、视频运动分析。

戚飞虎: 男, 1938年生, 教授, 博士生导师, 研究方向为图像处理、模式识别、视频监控。