第32卷 第19期 计 算 机 工 程 2006年10月
Vol.32 № 19 Computer Engineering October 2006
·多媒体技术及应用· 文章编号：1000—3428(2006)19—0230—03 文献标识码：A 中图分类号：TP37

H.264中运动矢量特征分类的快速运动估计算法

张 鹏，郭宝龙
(西安电子科技大学机电工程学院，西安 710071)

摘 要：提出了一种应用在H.264中的快速运动估计算法(TAMV)。该算法根据运动矢量自身特性，将宏块分为3种运动类型：大运动快，中等运动块和小运动块。对每类宏块自适应地选择一种或者几种编码模式进行预测，有效地减少了由7种宏块(亚宏块)模式所引起的计算量；同时选用精确而且快速的方向性菱形模板，对不同的模式采用不同的搜索策略。实验分析表明，该算法搜索精度接近于全搜索算法，而搜索时间只为全搜索算法的5.4%～10.5%。

关键词：H.264；运动矢量特征分类；方向性菱形模板

Fast Motion Estimation of Texture Analysis for Motion Vector in H.264

ZHANG Peng, GUO Baolong
(School of Electromechanical Engineering, Xidian University, Xi'an 710071)

【Abstract】 This paper proposes a novel motion estimation algorithm based on texture analysis of motion vector in H.264 (TAMV). According to motion vectors distribution and correlation within modes, TAMV proposes an efficient inter mode decision approach to reduce the number of candidate modes of macro block. And it selects a directional diamond search pattern in the motion estimation while maintaining the coding efficiency. Experiments show that TAMV can obtain good PSNR performance and very fast speed as well as low bit rates.

【Key words】 H.264; Texture analysis for motion vector; Directional diamond search pattern

H.264是最新的国际视频编码标准，与以往的视频标准相比它能够达到更高的编码效率，其中一个关键因素是可变块运动估计和模式判定的运用。为了提高准确度，H.264在运动估计中提供7种不同大小的块，可以节省多达15%的比特率。但同时也使得处理时间线性上升，计算复杂度大大增加。在JVT参考软件JM9.0中，采用了由Zhibo Chen，Peng Zhou等人提出的快速整像素运动估计算法UMHexagonS[1,2]，该算法高效的起始点预测和搜索策略，使得它与全搜索算法相比，比特率和PSNR值与之不相上下，但是搜索速度能提高80%～90%。因为该方法对全部7种模式采取遍历搜索，所以在提高算法搜索速度方面还存在很大的上升空间。本文基于H.264中可变块运动估计和模式判定的原理，根据运动矢量自身的特性，提出了一种简单有效的快速模式判断算法(TAMV)，有效地减少了宏块的候选模式；又采用高效的搜索模板和搜索策略自适应地对各模式进行快速搜索，实验结果显示整个算法的性能优于现有算法，更利于实时应用。

1 基于运动矢量特征的模式判断算法TAMV

1.1 可变块运动估计和模式判定的基本原理

在一个宏块中，可能包含多个在不同方向运动的物体，或者该宏块处于运动物体的边界上时，整个宏块仅仅依靠一个运动矢量并不能充分反映出它的真实运动，这将导致严重的预测误差。使用7种可变块进行运动估计，能够更好地描述现实世界中具有复杂运动的视频图像。在H.264中，每个16×16的宏块以16×16、16×8、8×16和8×8块模式进行编码，当对8×8块模式进行编码时，该8×8宏块将再一次被独立地分成8×8、8×4、4×8和4×4的亚宏块，所以H.264中一共存在7种预测模式。这种把宏块分割为不同大小的部分或者亚宏块的方法，又可称之为树结构的运动估计。然后通过率失真优化准则RDO(Rate Distortion Optimized)(式(1))确定最优匹配模式，得到的结果即为该宏块最终的编码模式。

$$J(m,\lambda_{motion}) = SAD(s,c(m)) + \lambda_{motion} \bullet R(m-p) \quad (1)$$

其中 $m = (m_x, m_y)^T$ 代表运动矢量，$p = (p_x, p_y)^T$ 表示运动矢量的预测值，λ_{motion} 表示拉格朗日乘法器，R(m-p)代表用来对运动信息进行编码的比特率，在查找表中得到。SAD(绝对误差和)值通过式(2)得到：

$$SAD(s,c(m)) = \sum_{x=1,y=1}^{B,B} \left| s\mid x,y\mid -c\mid x-m_x, y-m_y \mid \right| \quad (2)$$

当B=16、8或者4时，s是原始的视频信号，c是已编码的视频信号。

1.2 基于运动矢量特征分析的早期模式选择算法TAMV

可变块运动估计的主要观念是使用大宏块对平滑区域进行编码，而较小的宏块则用来对包含复杂运动的区域进行编码，这样对残差和运动矢量的熵编码结果可以最小化。虽然它现在已很成功地运用在H.264中，但是对每个宏块穷尽检测所有7种不同的编码模式并不总是必要的，例如一个宏块属于平滑区域，那它通过RDO被再次分割为更小块的可能性会很小。在可变块运动估计中，肯定存在对某宏块编码作用非常小的模式，称之为冗余模式，如果能够准确地选择一种或者几种编码模式，或者去除一些冗余的编码模式，将减少大量不必要的计算。

作者简介：张 鹏(1980－)，女，硕士生，主研方向：图像压缩编码，视频通信；郭宝龙，教授、博导
收稿日期：2005-12-03 **E-mail：**yqyte@yahoo.com.cn

如某个宏块是静止的或者只含有很小的运动，那么在多模式搜索时直接选用较大宏块进行搜索，排除较小宏块的运动模式搜索，可以达到很好的搜索精度，同时减少不必要的计算量；同样，如果某个宏块含有很丰富的运动细节，或者物体运动剧烈，那么直接采用较小的宏块进行搜索，得到的最优点必然也能满足编码要求。同时根据运动矢量场的特性可知，如果宏块中运动方向趋向于水平，那么有可能选择 16×8 或 8×4 编码模式，而没有必要再对其执行 8×16 或 4×8 的垂直模板搜索，反之亦然。本文正是基于上述分析，提出了一种通过运动矢量特征分析来进行早期模式选择的方法。

本文提出的 TAMV 算法先从宏块的 16×16 块模式开始搜索，由此得到的运动矢量来进行宏块类型判定，将宏块分成小运动块、中等运动块和大运动块 3 种类型，对应不同的分类结果，自适应地选取一种或者几种模式进行搜索，以达到减少计算量、提高搜索速度的目的。第 1 步所得到的运动矢量记为 MV，其水平和垂直分量分别写为 MV_x、MV_y。定义一个函数 L,并且 $L=|MV_{\_x}|+|MV_{\_y}|$，具体的分类规则如下：若 L< THL，那么此宏块属于小运动块，直接选用 16×8 和 8×16 进行编码；若 L>THH，那么此宏块属于大运动块，直接采用 8×8 模式进行编码，不再进行较大宏块的搜索。若 THL <= L <= THH，那么采用全搜索模式进行编码，即对 7 种可变块模式进行遍历搜索。

同时，再定义一个函数，$l=|MV_{\_x}|-|MV_{\_y}|$，其中 MV_x、MV_y 分别为 16×16 和 8×8 块的运动矢量水平和垂直分量。若 l >= thh ，那么此宏块含有水平运动矢量，直接排除掉 8×16 模式。只采用 16×8(8×4)模式。若 l < thh ，那么此宏块含有垂直运动矢量，直接采用 8×16(4×8)搜索模式。特别当 l = 0 时，那么采用 16×8(或 8×4)和 8×16(或 4×8)模式进行编码。

1.3 TAMV 算法的搜索模板

在 JM9.0 中的 UMHexagonS 中采用非对称的十字搜索和不均匀的多六边形栅格搜索相结合的思想，对宏块或亚宏块进行固定模板检测，这样虽然可以保证较高的搜索精度，但是会使得搜索点数过高，比如对 16×16 的宏块而言，在这两步一共需要搜索 8+4+16×4=76 个点，搜索点数占到全部点数的 35.2%，同时它的搜索模板不能自适应地结束搜索，导致搜索效率较低。所以在 TMAC 算法中，选用的是文献[3]中的方向性菱形模板。该模板的搜索性能较好，能用较少的点自适应地对宏块或者亚宏块进行搜索，且图像的 PSNR 值接近全搜索算法。

具体的搜索分类如下：

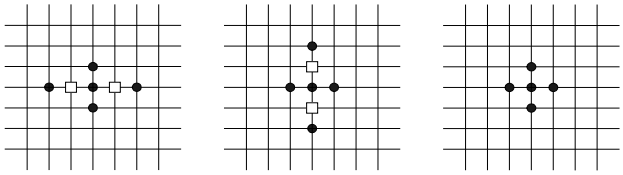
(1)在 16×16，8×8，搜索时采用方向性菱形模板(HDDSP 和 VDDSP)。

(2)在 16×8，8×4，搜索时采用方向性菱形模板中的水平模板(HDDSP)。

(3)在 8×16，4×8，搜索时采用方向性菱形模板中的垂直模板(VDDSP)。

(4)在 4×4，采用 SDSP 模板进行搜索(SDSP)。

各搜索模板如图 1 所示。对 1 而言，先对水平模板的 5 个点进行匹配，如果 MBD(最小点)在中心位置，搜索结束；若在近点，则用水平模板进行再搜索，如在远点，改用垂直模板进行搜索。当 MBD 点是中心点时，此时加上方格点，计算 3 个点的 SAD 值，最小值即为搜索点。对 2、3 而言，不换模板，直到得到最后的最小值点。对 4，SDSP 简单搜索，直到找到最佳匹配点。



(a) HDDSP　(b) VDDSP　(C)SDSP

图 1 TAMV 算法的搜索模板

1.4 TAMV 算法的搜索策略

整个算法的具体搜索步骤如下：

Step1 首先方向性菱形模板的水平模板和垂直模板交替使用，对当前宏块进行 16×16 的块模式检测，根据所得到的运动矢量判断下一步所进行的搜索模式。

(1)若 L < THL，那么此宏块属于小运动块，直接采用 16×8 和 8×16 进行编码。此时再进一步判断，如果

1)l >= thh ，那么此宏块含有水平运动矢量，直接排除掉其他模式，只选用 16×8 模式，采用 HDDSP 进行搜索，得到的最优点为全局最优点，结束其它模式的搜索，转到 Step3；

2)l < thh ，那么此宏块含有垂直运动矢量，直接选用 8×16 搜索模式，使用 VDDSP 进行搜索，得到的最优点为全局最优点，结束其它模式的搜索，转到 Step3；

3)l = 0 ，那么选用 16×8 和 8×16 模式进行编码，分别采用各自模式所对应的搜索模板进行搜索，得到各自对应的运动矢量，结束其它模式的搜索，转到 Step3。

(2)若 L >THH，那么此宏块属于大运动块，直接采用 8×8 模式进行编码，此时选用 HDDSP 和 VDDSP 交替搜索得到 8×8 块的运动矢量，然后如(1)所述再次进行运动矢量方向性的判断，特别当 l = 0 时，选用 4×4 模式进行编码，使用 SDSP 进行搜索。

(3)若 THL <= L<= THH ，那么此宏块属于中等运动块，采用全搜索模式进行编码,转到 Step2；

Step2 对该宏块余下的 6 种模式进行遍历搜索，得到各自最优运动矢量，转到 Step3。

Step3 对当前宏块存在的编码模式进行 RDO 比较，选择最优模式为最终编码模式。

实验结果显示，当 THL=0.9、THH=1.9，并且 thh=thv=0 时，可以得到最好的实验结果。

2 实验结果分析

为了验证算法的有效性，选取 JM9.0 中的全搜索算法(Full)和快速算法(UMHexagonS)在相同条件下与 TAMV 算法进行对比实验。实验中选取 4 个 QCIF(176×144)图像：claire(250 帧)，salesman(250 帧)，foreman (350 帧)，carphone(350 帧)和 CIF 图像 mobile(352×288, 298 帧)序列。图像序列以 30fps 的速度进行编码，采用 Hadamard 变换，CABAC 作为熵编码，5 个参考帧，最大的搜索范围为 16，QP 值分别选取为 28、30、32。对这 5 种运动程度不一的图像分别从图像亮度分量的平均峰值信噪比(即 JM9.0 中的 SNRY 值)、每帧图像帧间编码的平均比特率和运动估计耗时 3 个方面进行比较。结果为表 1、表 2 和表 3 所示，ΔA 表示该算法与 A 算法的结果差值，Δ%(A)表示该算法与 A 算法的增量百分比，(A/B)%代表该算法结果与 B 算法结果的百分比。

表 1 图像亮度分量的平均峰值信噪比(单位：dB)

	claire			salesman			foreman			carphone			mobile		
QP	28	30	32	28	30	32	28	30	32	28	30	32	28	30	32
Full	40.04	38.48	37.10	35.79	34.24	32.74	35.86	34.51	33.23	36.86	35.35	33.97	33.91	32.21	30.64
UMHS	39.99	38.46	37.04	35.77	34.25	32.70	35.83	34.47	33.14	36.81	35.33	33.88	33.90	32.20	30.62
Δ(F)	-0.05	-0.02	-0.06	-0.02	+0.01	-0.04	-0.03	-0.04	-0.09	-0.05	-0.02	-0.09	-0.01	-0.01	-0.02
AC	40.04	38.45	36.99	35.75	34.23	32.70	35.81	34.46	33.13	36.83	35.31	33.89	33.90	32.19	30.62
Δ(F)	0.00	-0.03	-0.11	-0.04	-0.01	-0.02	-0.05	-0.05	-0.10	-0.03	-0.04	-0.08	-0.01	-0.02	-0.02

从表 1 看出，全搜索法的平均 SNRY 值最高，说明它的搜索精度是最高的。TMAC 算法的亮度分量的峰值信噪比平均值略差于全搜索算法(小于 0.11 个 dB)，与 UMHexagonS 不相上下。并且该算法在运动剧烈的 mobile 序列中性能比全搜索算法只相差 0.02 个 dB 左右，与 UMHexagonS 相当，搜索准确度高。图 2 直观地显示了不同算法在 QP=28 时的 mobile 序列图像的 SNRY 值，可以看出 TAMV 十分接近于最优的全搜索算法。

图 2 算法的预测精度比较

表 2 列出了各算法在不同图像条件时帧间编码的平均比特率。从该表可以看出，UMHexagonS 算法的传输比特率是最优的。在 5 种测试图像中，TAMV 算法的比特率最多超出全搜索算法的 1.74%，高出 UMHexagonS 算法 1.93%。这主要由于 foreman 序列存在的剧烈运动、画面人物和镜头的运动以及场景切换造成的。在运动较为缓和或者微小的 claire 序列中，TAMV 算法比特率比 UMHexagonS 有轻微的增加，而在大运动的 mobile 序列中，比特率增加更少，不到 0.3%。

表 2 图像帧间编码的平均比特率

	QP	Full		UMHexagonS		AMSA		
		BR(kb/s)	**Δ(F)%**	BR(kb/s)	**Δ(F)%**	**BR(kb/s)**	**Δ(F)%**	**Δ(U)%**
claire	28	1 358.56	0	1 345.97	-0.94	1348.34	**-0.75**	**+0.18**
	30	1 016.94	0	1 008.43	-0.84	1016.75	**-0.02**	**+0.83**
	32	747.77	0	746.27	-0.20	746.20	**-0.21**	**-0.01**
sales-man	28	2 327.33	0	2 336.15	+0.38	2362.06	**+1.49**	**+1.11**
	30	1 765.52	0	1 764.35	-0.07	1777.73	**+0.69**	**+0.76**
	32	1 328.60	0	1 325.02	-0.27	1332.64	**+0.30**	**+0.58**
foreman	28	6 165.28	0	6 162.62	-0.04	6242.08	**+1.25**	**+1.29**
	30	4 640.26	0	4 631.45	-0.19	4720.94	**+1.74**	**+1.93**
	32	3 504.11	0	3 504.14	+0.000 9	3555.82	**+1.48**	**+1.47**
carph-one	28	6 258.33	0	6 256.44	-0.03	6341.82	**+1.33**	**+1.36**
	30	4 751.78	0	4 761.71	+0.21	4830.54	**+1.66**	**+1.45**
	32	3 596.61	0	3 602.89	+0.17	3643.41	**+1.30**	**+1.12**
mobile	28	54 597.97	0	54 480.45	-0.22	54639.50	**+0.08**	**+0.29**
	30	40 074.59	0	40 091.00	+0.04	40136.30	**+0.15**	**+0.11**
	32	28 699.38	0	28 697.43	-0.93	28757.18	**+0.20**	**+0.21**

表 3 图像平均每帧的运动估计耗时

	QP	Full		UMHexagonS		AMSA		
		time(ms)	**(F/F)%**	time(ms)	**(U/F)%**	time(ms)	(A/F)%	(A/U)%
claire	28	1 901.49	100	125.07	6.58	101.75	**5.35**	**81.35**
	30	1 911.83	100	123.20	6.44	105.19	**5.50**	**85.38**
	32	1 917.82	100	123.64	6.45	99.76	**5.20**	**80.69**
sales-man	28	1 827.00	100	155.04	8.49	120.35	**6.59**	**77.63**
	30	1 830.73	100	163.36	8.92	118.45	**6.47**	**72.51**
	32	1 844.12	100	173.16	9.39	120.20	**6.52**	**69.42**
foreman	28	1 887.79	100	351.85	18.64	196.77	**10.42**	**55.92**
	30	1 898.73	100	349.18	18.39	192.19	**10.12**	**55.04**
	32	1 917.55	100	354.92	18.51	195.09	**10.17**	**54.97**
carph-one	28	1 885.80	100	269.73	14.30	165.68	**8.79**	**61.42**
	30	1 892.38	100	272.86	14.42	160.79	**850**	**58.93**
	32	1 906.86	100	266.56	13.98	158.12	**8.29**	**59.32**
mobile	28	7 188.19	100	1 489.32	20.72	745.99	**10.38**	**50.09**
	30	7 169.29	100	1 518.51	21.18	743.85	**10.38**	**48.99**
	32	7 225.91	100	1 524.73	21.10	745.91	**10.32**	**48.92**

表 3 是算法搜索速度的比较结果。从表 3 中可以看出，TAMV 算法在速度上占有绝对优势。在这 5 种测试图像中，TAMV 的运动估计搜索时间只占到全搜索算法的 5.4%～10.5%，节省了约 90%以上的时间。同时它比 UMHexagonS 算法也节省了将近 15%～50%的时间，特别对运动较为剧烈的序列，它可以节省更多的搜索时间。

图 3 显示的是各算法在 QP=28 时针对 mobile 图像的平均耗时比较，很明显，3 种算法种 TAMV 算法在速度上有绝对性的优势，更加利于实时应用。

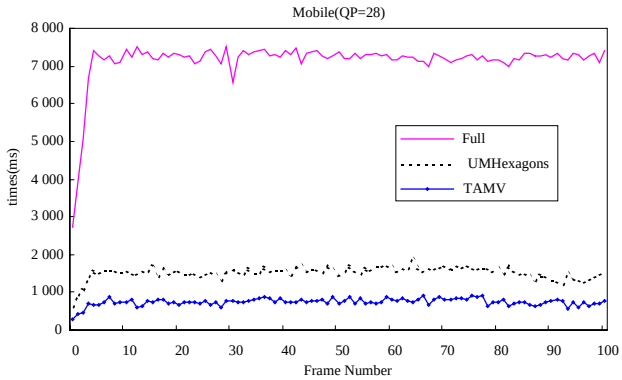


图 3 算法的预测速度比较

综合可知，TAMV 算法只需要牺牲很少的比特率就可以达到搜索质量与全搜索算法接近、搜索时间优于 UMHexagonS 的性能。特别是针对运动情况剧烈的大运动序列图像而言，该算法在搜索精度、传输比特率和搜索速度上都具有很大的优势。

3 结束语

本文提出了一种基于 H.264 的简单有效的快速运动估计算法，它先对每个宏块以 16×16 模式搜索得到的运动矢量进行合理的阈值判定，确定该宏块的所属类型，然后根据判定结果，选择一种或者几种搜索模式进行下一步检测；在具体搜索时，又选用精确而且快速的方向性菱形模板，对不同的模式采用不同的搜索策略，在提高搜索速度的同时又确保了编码质量。实验结果显示该算法只需要牺牲很少的比特率就可以达到搜索质量与全搜索算法接近、搜索时间优于 UMHexagonS 的性能，更利于实时应用，是一种运用在 H.264 中性能非常优秀的快速算法。

参考文献

1 Chen Zhibo, Zhou Peng, He Yun. Fast Interger Pel and Fractional Pel Motion Estimation for JVT[Z]. fttp://standards. polycom.com/imtc_jvtexperts/2002_12_Awaji/JVT-F017.zip.

2 Chen Zhibo, Zhou Peng, He Yun. Fast Motion Estimation for JVT[Z]. fttp://standards.polycom.com/imtc_jvtex perts/2003_03_Pattaya/JVT-G016.zip.

3 Jia Hongjun, Zhang Li. Directional Diamond Search Pattern for Fast Block Motion Estimation[J]. IEEE Electronics Letters, 2003, 39 (22) :1581-1583.

4 Zhou Zhi, Sun Mingting, Hsu Yuhfeng. Fast Variable Block-size Motion Estimation Algorithms Based on Merge and Split Procedures for H.264/MPEG-4 AVC[C]. Proc. of IEEE International Symposium on Circuits and Systems, Vancouver, Canada, 2004: 725-728.