

基于方向判断的自适应运动估计快速算法

李扬曦 李涛 马义德

(兰州大学信息科学与工程学院电路与系统研究所, 兰州 730000)

摘要 运动估计是视频编码中举足轻重的环节,其性能的优劣很大程度上影响着编码质量。为此提出了一种基于方向判断的自适应运动估计快速搜索算法。该算法充分考虑序列自身的运动属性,利用相邻块运动矢量场的分布特性,根据相邻块运动矢量的相似程度自适应地设定阈值来判断静止块,并且对预测运动矢量进行方向的判断,使用方向性的模板搜索最优点。与以往算法相比,该算法考虑了图像序列本身的特性,具有更高的灵活性。仿真结果表明,在基本不损失性噪比的情况下,该算法对不同序列均有效的降低了运算量。

关键词 运动估计 自适应阈值 静止块判断 方向性模板

中图法分类号:TN919.81 文献标识码:A 文章编号:1006-8961(2008)08-1501-05

A Fast Adaptive Motion Estimation Algorithm Based on Direction Judgment

LI Yang-xi, LI Tao, MA Yi-de

(School of Information Science and Engineering, Lanzhou University, Lanzhou 730000)

Abstract Motion estimation (ME) plays an important role in video coding system, since it could significantly affect the output quality of an encoded sequence. This paper proposes a fast adaptive ME algorithm based on the judgment of the predictive motion vector (PMV). This proposed algorithm takes advantage of the correlation of neighboring blocks to adaptively decide the threshold for judging the stationary regions. And this algorithm sorts the PMV into several regions by its direction and uses different search patterns, respectively. Compared with the former algorithm, this algorithm considers the characteristic of different sequences and thus, it is more flexible than before. Experimental result shows that this algorithm reduces computation complexity variously for different sequence under the premise that the PSNR does not lose.

Keywords motion estimation, adaptive threshold, judgment of stationary region, directional search pattern

1 引言

运动估计是视频压缩中的关键技术,是视频编码中的主要开销,约占视频编码计算总量的60%~80%。快速高效的运动估计算法可以有效地改善编码质量。块匹配算法(BMA)是目前广泛使用的运动估计技术,它具有算法简单、便于硬件实现等优点。块匹配算法将图像序列帧划分为 16×16 或 8×8 像素的宏块(MB),每个宏块在参考帧的搜索窗口中按匹配准则进行搜索。衡量最佳匹配的准则有很多种,绝对差和(SAD)由于其运算简单被广泛

使用。全搜索算法是最简单可靠的块匹配算法,它遍历搜索窗口中的每个点以获得最佳匹配块。但是全搜索算法巨大的运算量使它无法实现实时的视频编码,因而提出了各种快速的运动估计算法。

各种快速算法的目标是在保持编码质量的条件下尽可能减少运算量。为了减少运算量,一类快速算法是按某种搜索策略只对搜索窗口中的部分参考点进行计算。经典的算法有三步搜索算法(TSS)^[1],对数搜索(2DLS)^[2],菱形模板搜索方法(DS)^[3],基于六边形模板的搜索法(HEXBS)^[4]等。这些算法都是基于以下两点假设:一是假定运动矢量场是中心偏移的,即运动矢量主要集中在搜索中

心附近很小的范围内;二是假定随着搜索点靠近最优点,匹配误差也是单调递减的。事实上很多序列并不符合以上假定,因此上述算法容易陷入局部最优,从而影响了编码质量。

预测快速搜索算法是近年来提出的基于运动矢量之间相关性的一类快速算法,性能优于前一类算法。这些算法利用空域或时域相邻块的运动矢量来预测当前块的搜索起始点,在预测起始点附近获得最优的运动矢量。这类算法包括矢量场自适应搜索算法(MVFAST)^[5-7]和带预测的矢量场自适应搜索算法(PMVFAST)^[8]等,其中MVFAST已经被MPEG-4标准第7部分所采用。本文提出了一种基于方向判断的自适应快速搜索算法,采用了阈值自适应,宏块活动性判断,方向性模板选择等技术,在获得与全搜索接近的编码效果的同时,有效地降低了运算量。

2 基于方向的自适应快速搜索算法

不同的图像序列有不同的运动特性。通常将运动图像分为前景和背景。例如报告的现场,背景静止不动,前景是人的头肩像,有缓慢的运动;而体育

比赛的图像序列前景与背景都有快速的变化;还有一类序列前景几乎不动而背景运动,例如摄像机跟踪观察一辆运动的小汽车,背景在快速变化,而前景始终是跟踪的汽车。物体运动的方式也各不相同。有物体水平方向的移动,也有摄像机旋转引起的转动。以往的算法并不考虑序列本身的这些特性,往往造成不必要的运算,或者编码质量受到影响。本文提出的算法有效的利用了序列的运动特性,并且从文献[9]可知,大部分序列的运动矢量都有很强的方向性,因而本文选用了方向性模板,进一步降低了运算量。

2.1 自适应阈值判断静止块

在图像序列中存在大量的静止块,即宏块在当前帧中的位置相对参考帧并没有变化。当判定当前块为静止块时,不必进行更多的计算就可以认为当前块的运动矢量是(0,0)。利用静止块的判断有效地减少了运算量。在以往的算法中使用固定的阈值判断静止块,即当绝对差和 SAD 小于某一确定值即判定该宏块为静止块。显然对不同运动特性的序列使用同样的阈值有可能造成附加的运算或图像质量的下降。表1是在不同阈值下不同图像序列的编码质量(用峰值信噪比 PSNR 衡量)的对比。

表 1 多个阈值下不同图像序列编码质量对比
Tab. 1 comparison of PSNR under different thresholds

	阈 值								
	0	128	256	384	512	640	768	896	1024
Claire	43.49	43.48	43.43	43.45	43.38	43.38	43.40	43.36	43.29
Container	37.39	37.39	37.39	37.38	37.38	37.40	37.41	37.42	37.38
Foreman	40.60	40.60	40.62	40.59	40.57	40.51	40.37	40.18	40.06
Carphone	43.36	43.36	43.35	43.33	43.29	43.23	43.17	43.11	43.06

从表1可以看出,对于活动性小,空间细节少的 Claire 序列,当阈值大于 896 时,图像的质量才有了较为明显的下降,阈值从 0 增加到 1 024 时图像的质量仅有 0.2dB 的下降;对于空间细节适中,运动缓慢的 Container 序列,阈值的变化并没有显著影响图像的质量;而对中等运动量的 Foreman 序列,当阈值增大到 768 以后,图像质量有了显著的下降,在阈值从 0 变化到 1 024 的过程中图像质量下降了 0.54dB;对细节较多的 Carphone 序列,阈值大于 640 以后图像质量有明

显下降。这是因为 Claire 和 Container 两个序列运动非常缓慢,大部分宏块运动量非常小,阈值的改变并没有显著影响对静止块的判断,因而图像的质量无明显变化;而 Foreman 和 Carphone 两个序列由于细节较多,运动量较大,随着阈值的增加,很多运动量不大的宏块被判定为静止块,从而影响了图像的质量。可见阈值的恰当选择可以显著改善图像编码质量。

采用一种自适应的阈值设置方法^[10]。如图 1 所示,设 $MB(i, j)$ 为当前块, i 和 j 分别为当前块的

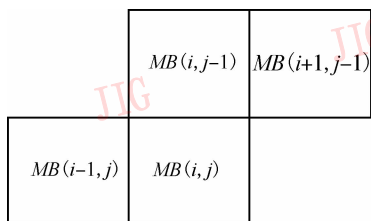


图 1 构成参考列表的相邻块

Fig. 1 Neighboring blocks of current block

横坐标和纵坐标。 $MB(i-1, j)$, $MB(i, j-1)$, $MB(i+1, j-1)$ 分别为当前块的左、上和右上块, 在编码的过程中它们都先于当前块编码, 因此可以选择上述 3 块为当前块编码的参考列表 CL (candidate list), 利用 CL 中的块与当前块运动特性之间的相关性来确定阈值, 具体方法如下:

当 CL 中所有块的运动矢量都是 $(0, 0)$ 时, 则取 3 个块中最大的残差 SAD 为预测阈值 T_p 。此时参考块都属于静止区域, 当前块属于静止块的概率较大, 选择较大的阈值。

当 CL 中有任一块的运动矢量不为 $(0, 0)$, 则取最小的 SAD 为阈值 T_p 。这是因为此时有参考块不属于静止区域, 那么当前块属于静止块的可能性大大降低, 故选择较小的阈值。

为了在编码效果与速度之间取得较好的折中, 应该为阈值 T_p 设定上下限 T_u 和 T_l , 因此阈值 T 定义如下:

$$T = \max(T_l, \min(T_u, T_p)) \quad (1)$$

取 $T_u = 1024$, $T_l = 512$ 。

2.2 宏块属于相同运动对象的判断

序列中同一运动对象往往是包括了若干个宏块的区域。因此当 CL 中参考块与当前块属于同一个运动物体时, 例如同属于静止的背景, 或移动的汽车, 它们的运动矢量之间的相关性更强。充分利用这种相关性可以进一步降低运算量。在文献[11]中提出了局部活动性 LMA (local motion activity) 的概念来描述宏块的运动程度, 对此概念进行改进, 定义运动矢量偏差标志 MVDF (motion vector difference flag) 如下:

$$\begin{aligned} l_i &= |x_i - \bar{x}| + |y_i - \bar{y}| \\ L &= \max(l_i) \\ \text{MVDF} &= \begin{cases} \text{low} & L \leq L_0 \\ \text{high} & L > L_0 \end{cases} \end{aligned} \quad (2)$$

式中, x_i 和 y_i 是参考列表中第 i 个宏块的运动矢量,

$\bar{x}$ 和 $\bar{y}$ 是参考列表中各宏块运动矢量的平均值。 l_i 度量了第 i 个宏块运动矢量偏离平均值的程度。 L 是 l_i 中最大的一个, 表示了参考列表中各宏块运动矢量偏离平均值的最大程度。当 L 小于一定值时, 表明参考列表中的各宏块的运动程度比较接近, 即它们属于同一运动物体的概率增加, 那么可以认为当前块也是属于同一运动对象的, 在搜索时可以只在预测中心附近很小的范围搜索; 而当 L 较大时, 表明参考列表中各块运动矢量之间的相似程度下降, 属于同一运动对象的概率减小, 它们的运动矢量与当前块的相关性也随之降低, 在搜索时应当进行更多的运算。本文算法根据 MVDF 的不同使用不同的搜索模板, 取 $L_0 = 2$ 。

2.3 基于角度的方向性模板的选择

由文献[9]可知, 通过对 18 个常见序列运动矢量的分析, 有 22.69% 的运动矢量在水平方向, 有 7.42% 的运动矢量在垂直方向, 有 58% 的运动矢量为 $(0, 0)$ 。显然其他方向上的运动矢量只占很少的一部分。那么如果在水平和垂直方向上使用有方向特性的简单模板可以显著地减少搜索点数, 从而降低运算量。本文算法使用的模板如图 2 所示。

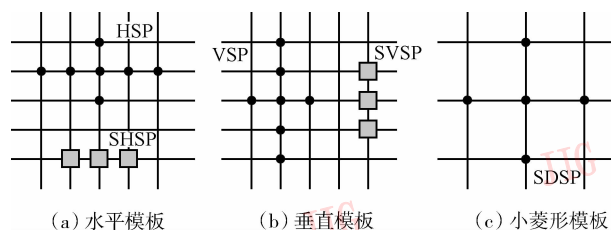


图 2 搜索模板

Fig. 2 Search patterns

取参考列表 CL 中 SAD 值最小的块的运动矢量作为当前块的预测运动矢量 (PMV)。将 PMV 绝对值的分布区域分为 3 块, 如图 3 所示。与水平方向

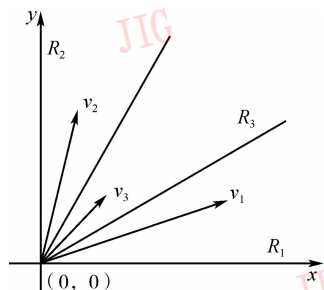


图 3 PMV 分布区域分类图

Fig. 3 Classification of PMV

夹角小于 30° 的区域称为 R_1 , 大于 60° 小于 90° 的区域称为 R_2 , 30° 与 60° 之间称为 R_3 。对于落入 R_1 的 PMV, 例如图中的 v_1 , 由于其接近水平方向, 可以认为当前块沿水平方向运动的概率较大, 所以采用图 2(a) 中的水平模板 HSP 或 SHSP 搜索, 以 SAD 为匹配准则得到全局最优点。同理对 R_2 中的 PMV 使用图 2(b) 中的垂直模板 VSP 或 SVSP 搜索。而对 R_3 中的 PMV, 由于无法将其归类到水平或垂直方向, 则使用图 2(c) 中小菱形模板 SDSP 反复搜索, 直到 SAD 最小点落在菱形中心为止。

2.4 算法描述

- 本文算法流程如下:
- (1) 使用式(1)确定阈值;
 - (2) 判断当前块是否为静止块, 是则转至步骤 5, 否则至步骤 3;
 - (3) 选参考列表中 SAD 值最小的宏块的运动矢量为当前块的预测运动矢量 (PMV), 从 PMV 指向的点处开始搜索;
 - (4) 判断 PMV 的方向性:
 - 若属于 R_1 , 利用式(2)判断 MVDF, 若为 low, 则使用 SHSP 搜索, 否则使用 HSP 搜索;
 - 若属于 R_2 , 利用式(2)判断 MVDF, 若为 low, 则使用 SVSP 搜索, 否则使用 VSP 搜索;
 - 若属于 R_3 , 则使用 SDSP 模板反复搜索, 直到最小 SAD 点落在菱形中心;
 - (5) 算法结束。

3 算法仿真及分析

以 H. 264 的验证模型 JM10.2 为仿真平台。通过在 JM10.2 中实现本文算法以及全搜索算法、菱形搜索算法、矢量场自适应搜索算法 (MVFAST) 等经典算法进行比较。分别采用上述算法以 H. 264 中的基本档次对不同序列进行编码, 编码序列采用 IPPP 格式, 即第 1 帧为 I 帧, 其后的帧使用 P 帧, 不使用 B 帧。帧率设置为 30fps, 量化参数 QP 为 28。以亮度 PSNR 和搜索点数作为编码效果的比较标准。选择具有代表性的 7 个测试序列进行比较。其中 Claire, News, Container 3 个序列运动量非常小, 细节少; Saleman, Foreman, Carphone 3 个序列运动量较大, 细节较多; 而 Football 序列细节多, 运动剧烈。仿真结果如表 2 和表 3 所示。

表 2 不同算法编码质量的比较
Tab. 2 Comparison of PSNR by different algorithms
单位: dB

序列	全搜索	菱形搜索	MVFAST	本文算法	比较全搜索	比较 MVFAST
Claire	38.90	38.80	38.81	38.82	-0.08	0.01
News	36.10	36.06	36.03	36.07	-0.03	0.04
Container	35.68	35.68	35.67	35.65	-0.03	-0.02
Salesman	35.23	35.24	35.24	35.26	0.03	0.02
Foreman	34.81	34.79	34.80	34.85	0.04	0.05
Carphone	35.96	35.97	35.98	35.95	-0.01	-0.03
Football	37.40	37.34	37.37	37.44	0.04	0.07

表 3 不同算法搜索点数的比较
Tab. 3 Comparison of search point by different algorithms

序列	全搜索	菱形搜索	MVFAST	本文算法	比较 MVFAST (%)
Claire	43 016 589	548 559	129 899	119 778	7.79
News	31 912 056	396 234	157 239	117 278	25.41
Container	32 235 489	388 467	153 270	147 269	3.92
Salesman	48 083 706	578 376	254 339	204 966	19.41
Foreman	42 800 967	775 431	408 161	271 139	33.57
Carphone	40 860 369	633 915	345 724	198 192	42.67
Football	37 518 228	1 265 616	637 704	408 091	36.01

从表 2 和表 3 可以看出, 全搜索算法的图像质量最好, 而搜索点数也最多。本文算法与全搜索算法相比, 亮度 PSNR 基本无变化, 7 个序列平均约有 0.01dB 的下降; 与菱形搜索算法相比, 亮度 PSNR 有较大的提高; 而与 MVFAST 算法相比, 亮度 PSNR 平均有 0.02dB 的上升。这是因为本文算法与 MVFAST 算法都属于预测快速搜索算法, 加入对起始点预测技术后图像质量明显改善, 因此亮度 PSNR 优于未采用预测技术的菱形搜索算法。在搜索点数上本文算法与 MVFAST 算法相比也有了不同程度的减少, 并且随着序列运动复杂性的增强, 节约的点数更多。对运动缓慢的 Claire, News, Container 3 个序列平均有 12.37% 的减少, 而对另外 4 个运动较剧烈的序列则平均有 39.12% 的减少。这是因为 MVFAST 在任何方向都使用小菱形模板反复搜索, 而本文算法加入了对预测矢量方向性的判断, 在判断之后使用方向性模板只进行一次搜索, MVFAST 算法中若使用小菱形模板 2 次就需要搜索 8 个点, 而本文的模板至多只搜索 7 个点, 从而有效地降低了运算量。特别是对复杂的图像序列, 这种

优势更加明显。总体来看,本文算法在图像质量与全搜索算法相差微小的情况下非常有效地降低了运算量。

4 结 论

在充分考虑了图像序列各自的运动特性基础上,提出了一种基于预测矢量方向判断的自适应快速运动估计算法,引入自适应阈值设定和方向性模板等技术使本文算法可以很好地适应不同运动特性的图像序列。实验结果表明,该算法与全搜索算法以及其他一些经典快速算法相比图像质量相当或略有提高,同时大大减少了计算量,可以明显提高运算速度。

参考文献 (References)

1 Koga T, Linuma K, Hirano A. Motion compensated interframe coding for video conferencing [A]. In: Proceedings of IEEE National Telecommunications Conference [C], New Orleans, LA, USA, 1981: G5.3.1 ~ 5.3.5.

2 Jain J R, Jain A K. Displacement measurement and its application in interframe image coding [J]. IEEE Transactions on Communications, 1981, 29 (12): 1799 ~ 1808.

3 Zhu S, Ma K K. A new diamond search algorithm for fast block matching motion estimation [J]. IEEE Transactions on Image

Processing, 2000, 9 (2): 287 ~ 290.

4 Zhu C, Lin X, Chau L P. Hexagon-based search pattern for fast block motion estimation [J]. IEEE Transactions on Circuits and System for Video Technology, 2002, 12 (5): 349 ~ 355.

5 Hosur P I, Ma K K. Motion vector field adaptive fast motion estimation [A]. In: Proceedings of Second International Conference on Information, Communications and Signal Processing [C], Singapore, 1999: 7 ~ 10.

6 Hosur P I, Ma K K. Report on Performance of Fast Motion Estimation Using Motion Vector Field Adaptive Search Technique (MVFAST) [EB/OL]. <http://www.ntu.edu.sg/home/ekkma/Publications.htm>, 1999-12.

7 Ma K K, Hosur P I. Performance Report of Motion Vector Field Adaptive Search Technique (MVFAST) [EB/OL]. <http://www.ntu.edu.sg/home/ekkma/Publications.htm>, 2000-03.

8 Tourapis A M, Au O C, Liou M L. Fast Block-matching Motion Estimation Using Predictive Motion Vector Field Adaptive Search Technique (PMVFAST). [EB/OL]. <http://users.softlab.ece.ntua.gr/~tourapis/navigate.htm>, 2000-03.

9 Jia H, Zhang L. Directional diamond search pattern for fast block motion estimation [J]. Electronic Letters, 2003, 39 (22): 1581 ~ 1583.

10 Ahmad I, Zheng W, Luo J, et al. A fast adaptive motion estimation algorithm [J]. IEEE Transactions on Circuits and System for Video Technology, 2006, 16 (3): 420 ~ 438.

11 Hosur P I. Motion adaptive search for fast motion estimation [J]. IEEE Transactions on Consumer Electronics, 2003, 49 (4): 1330 ~ 1339.