

# 对偶算法的有效改进

刘德春

(南阳理工学院计算机科学与技术系, 南阳 473000)

**摘要** 运动估值的快速算法普遍存在着精度过低的缺陷。本文应用块匹配法的基本原理,通过对块匹配规律的分析,找到了改进对偶快速算法搜索路径的有效方法,提出了实用快速算法——最大梯度及多重循环对偶法。该算法在兼顾准确性的同时,保留了算法路径简单、易于实现的优点,其匹配结果仍具有快速算法的速度且更加接近全匹配算法的精度。

**关键词** 图像编码 图像处理 运动估值

中图分类号: TN919.8 文献标识码: A 文章编号: 1006-8961(2006)04-0465-04

## The Effective Improvement of Conjugate Search Algorithm

LIU De-chun

(Department of Computer Science and Technology, Nanyang Institute of Technology, Nanyang 473000)

**Abstract** Generally fast algorithms for motion estimation have low precision. Based on the basic theory and rule of block matching, a new fast motion estimation algorithm which uses maximum grads and more cycle conjugate search algorithm to improve searching approach, is presented. Its speed is comparable to typical fast algorithm and its precision is comparable to the exhaustive searching algorithm, meanwhile it has veracity, simple algorithm path and is easy to be realized. The results of the comparison are also presented.

**Keywords** image coding, image processing, motion estimation

## 1 引言

在保持较高恢复精度的前提下提高运动图像的压缩倍数,是动态序列图像压缩的一个重要课题。为此许多有关的压缩标准如 H. 261 ~ H. 263、MPEG1 ~ MPEG4 都采用了运动补偿技术。而运动补偿的效果主要取决于运动估计或运动匹配的具体算法。全匹配搜索算法是各种估计算法中精度最高的,但由于运算量太大,估值速度慢,不适于在实时性要求较强的场合使用。为解决这一问题,人们研究并提出了很多运动估值的快速算法,典型的如三步法、对数法、对偶法等。实践表明,这些快速算法具有匹配速度快的突出优点,一般比全匹配搜索算法约快一个数量级,但同时又暴露出匹配精度差的

明显缺点。以典型对偶算法为例,对 SUSIE 标准序列图像(352 × 240 单波段)前 11 幅图做压缩实验,并选第 1 帧作为基础预测帧(16 × 16 宏块,32 × 32 搜索窗),其余补偿帧均假设用匹配到的最小失真宏块去替换当前宏块,则估值恢复图像相对于原标准图像的平均信噪比分别为 33.19dB,而用全匹配算法的匹配结果为 35.8dB,相差 2 个多分贝。运动图像在压缩时,对估值的宏块,或保存运动矢量,或保存运动矢量及误差值等,如果估值误差大,不仅影响图像的恢复精度,同时也影响图像的压缩倍数。综观快速算法,其速度主要快在大量减少搜索窗内的搜索点数,而精度差在简化后的搜索路径不够合理,很多宏块不能找到最佳匹配位置,而这一位置可能实际存在。如果把减少搜索点数和合理选择路径进行折中考虑,就有可能获得意外收获,譬如,把对

收稿日期:2004-02-26;改回日期:2005-06-09

第一作者简介:刘德春(1963 ~ ),男,副教授。1998 年毕业于北京航空航天大学获工学硕士学位。从事计算机应用、图像处理等方面的研究。E-mail: chunqiu@371.net

偶算法进行这样的研究和改进,就能使其匹配效果得到很大改善。

## 2 块匹配法与对偶算法

在运动图像编码中 2 维运动估值被广泛应用于运动补偿帧间预测、帧间内插中。帧间运动可以近似为本帧(第  $k$  帧)内一个区域或多个区域相对于参考帧(如第  $k-1$  帧)的逐段平移。块匹配法是一种基于模式匹配的位移估值法。其基本思想可以描述如下:首先,本帧(第  $k$  帧)图像被均匀划分成  $M \times N$  像素的矩形块(称为子块),并假定位于同一子块内的所有像素具有同样的位移,则对每一子块只需要计算一个位移矢量。每一子块在参考帧内预先确定的一个搜索窗内的所有位置上进行位移匹配(互相关)运算。这里搜索窗大小为  $(M+2h) \times (N+2v)$ ,其中  $v, h$  分别为水平和垂直方向的最大位移(整像素数)。对每一位移  $[(i, j): -h < i < h, -v < j < v]$ ,在给定的匹配准则下,计算出相应的代价(失配函数值)  $D(i, j)$ ,而相应于使  $D(i, j)$  为最小的  $(i, j)$  给出了该块的位移矢量。

目前常用的匹配准则有均方误差(MSE)、平均绝对值误差(MSD)和绝对值误差(SAD)准则等。其表示形式为

$$E_{\text{MSE}}(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|^2$$

$$E_{\text{MSD}}(i, j) = \frac{1}{MN} \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|$$

$$E_{\text{SAD}}(i, j) = \sum_{m=1}^M \sum_{n=1}^N |f_k(m, n) - f_{k-1}(m+i, n+j)|$$

其中,  $f_k$  表示第  $k$  帧灰度值,  $i, j$  分别为水平、垂直方向的偏移量。SAD 准则由于在计算上的方便性得到了广泛应用,后面介绍的对偶法及其改进算法均采用此准则。显然最优匹配应是全匹配搜索法,即对窗内的每一点都计算 SAD 值,最小的  $E_{\text{SAD}}(i, j)$  值即对应于最优匹配。它共需计算  $(2h+1) \times (2v+1)$  个 SAD 值。该方法的优点是能够保证全局最优。

对偶算法可以描述为,从搜索区域(窗)中某一起始点(通常选与被预测块对应位置)开始,先在横向( $X$  向)上确定最小失真的方向,接着沿这方向进行连续的比较匹配,找到匹配最小失真点(位置);然后再以此点为始,在纵向( $Y$  向)确定最小失真方

向,接着执行与  $X$  向同样的搜索过程,该过程结束时所找到最小失真点即被认为是最终估值点。设若用全匹配方法需要搜索  $x \times y$  个点的话,则用对偶法时至多搜索  $(v_x + v_y + 2)$  个点(注: $v_x, v_y$  分别为  $X$  和  $Y$  向最大绝对位移,  $v_x < x, v_y < y$ ),所以搜索要快得多。对偶法的优点表现在路径直接,易于实现,大致反映了匹配数据的分布规律。但是由于其搜索的步数过于有限(只有两步),使相当一部分块不能找到最佳匹配位置,所以匹配精度与全匹配算法相比损失较大。从块匹配的一个实际过程来说明这一点,选标准序列 SUSIE 中被预测帧距离基础帧较近(3 帧)和较远(10 帧)两种情况,分别如图 1 和图 2 所示(注:图中为截取的  $32 \times 32$  搜索窗内的不同匹配位置上的  $E_{\text{SAD}}$  数值)。

619	618	592	580	594	572	606	562	638	733
590	588	583	570	550	532	519	444	503	684
601	571	599	574	<u>473</u>	453	<u>346</u>	384	539	727
547	552	555	512	479	404	388	498	650	768
559	552	554	507	481	410	500	600	722	761
531	530	519	503	499	537	606	<u>678</u>	718	770
556	538	522	510	553	583	613	646	682	779
575	550	541	539	564	599	642	700	709	800

图 1 距离基础帧 3 帧的某块匹配过程

Fig. 1 Matching process of the block which has 3 frame distance to base frame

7926	8124	8845	9774	10791	11839	12922	13997	15029	
840	5373	5550	6529	7735	8018	10297	11552	12776	
5210	4053	3212	<u>3103</u>	4404	5823	7306	8793	9225	
5873	4536	3119	1930	<u>1120</u>	2756	4483	6130	7714	
6873	5356	5280	4379	3053	2214	3602	4831	6493	
7269	6230	5367	4467	3445	2351	2725	4133	5661	
9339	<u>8863</u>	8367	7538	6626	5410	3928	<u>3766</u>	4513	
9985	11565	11093	10401	9578	8488	7106	5552	4983	

图 2 距离基础帧 10 帧的某块匹配过程

Fig. 2 Matching process of the block which has 10 frame distance to base frame

图 1 中最小失真点在  $E_{\text{SAD}} = 346$  处,实际执行从  $E_{\text{SAD}} = 678$  处开始搜索,所得匹配位置为  $E_{\text{SAD}} = 473$  处;图 2 中最小失真点在  $E_{\text{SAD}} = 1120$  处,实际执行时从  $E_{\text{SAD}} = 8863$  开始,匹配所得位置为  $E_{\text{SAD}} = 3766$ ,误差较大。另外注意到,对偶法算法对于搜索时先  $X$  后  $Y$  向和先  $Y$  后  $X$  向在很多情况下的匹配结果是不一

样的,甚至相差较多。例如现改为先Y后X向搜索,则在图1中可得最佳匹配位置  $E_{\text{SAD}} = 346$ ,并且较前方法搜索步数更少;而在图2中搜索得  $E_{\text{SAD}} = 3103$ ,较前方法所得的3766更接近最小失真点,但要找到最佳匹配点,仍需在X和Y向再循环搜索一次才行。由此可见,要合理选择X与Y向的先后搜索顺序,并加大搜索范围,才能使更多的宏块搜索到最佳匹配位置,从而获得好的估值效果。

### 3 最大梯度及多重循环搜索法

对典型运动图像序列全匹配搜索所得的SAD值,进行误差散布分析和统计发现,在预测帧离基础帧较近的情况下(此时运动位移较小,新的技术也强调在压缩中应适时引入近参考帧),搜索窗范围内各个搜索块的SAD值分布基本属于单极值,所以在快速算法中为了实现方便,假设搜索窗内的SAD值有最小值且为单极值。例如在图1情况,  $\min E_{\text{SAD}} = 346$ ,此单极值即为最小误差点。可以想象,在此极值周围的SAD值呈立体抛物面状,如用切割法在X、Y向上连续求多个截断面上极值的办法就可逐步逼近,直至得到该极值,从2维搜索窗内看即是沿X和Y向连续查找搜索,直至找到SAD的最小值。XY的切割顺序不同,代表不同的求值路径,如果在XY向只各求一次单向极值的话,结果可能不同,或均未达到最终极值或其中一种求法未达到。对SUSIE序列图像前21帧的统计表明,用对偶法时平均至少有24.6%的块不能搜索到最小误差点,这主要是算法的深度不够造成的。所以,无论搜索采用先X向形式还是先Y向形式,应增加搜索步数,由于位移通常毕竟有限,所以增加的步数也不可能很大,直到重复对偶不能进行下去为止。为提高搜索的效率,在开始搜索时,先求出X向和Y向的下降梯度,选择梯度大的方向先搜索,例如,在图1中,从  $E_{\text{SAD}} = 648$ 处始,Y正下降梯度最大,如选择在Y向先操作,就比先进行X向操作能更快地搜索到极值点。算法操作如图3所示。

具体概括如下:

(1) 决定XY模式 在搜索窗中与预测帧的对应位置块O点开始,先在X向左右两点计算SAD值,找出SAD值递减最大的方向,并记下递减值  $dx$ (取正值),再找出Y向上下递减最大的方向,并记下递减值  $dy$ (取正值)。比较  $dx$ 、 $dy$ ,如果  $dx > dy$ ,

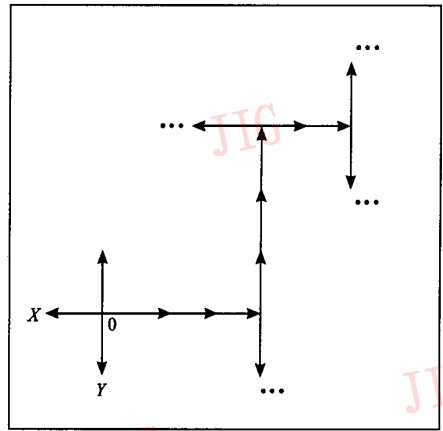


图3 最大梯度多重循环搜索法(X模式)

Fig. 3 Maximum grads and multi-cycle searching arithmetic (X mode)

执行先X后Y向模式(简称X模式),否则,执行先Y后X向模式(简称Y模式)。

(2) 循环匹配 设执行X模式,沿X向SAD值递减梯度最大的方向,连续计算比较下去,直至递减终止,记下末值及位置,再从此位置开始计算并求出Y向SAD值递减梯度最大的方向,沿这方向计算并比较下去,直至递减终止,记下末值及位置;依此位置再重新开始X模式,直至X、Y向递减均不能成立或搜索越界为止。Y模式的执行过程与此类似。最终匹配所得即为预选最小失真块SAD。

用该算法执行图1的搜索,匹配搜索过程为  $678 \rightarrow 600 \rightarrow 498 \rightarrow 384$ (为判别前进模式而计算的附加点未列);执行图2的匹配搜索过程为  $8863 \rightarrow 6230 \rightarrow 5356 \rightarrow 4536 \rightarrow 4053 \rightarrow 3212 \rightarrow 3103 \rightarrow 1930 \rightarrow 1120$ 。

该算法的特点表现在对于单极值的情况肯定能找到最小失真点,而无论是X模式还是Y模式(图1);而对于多极值情况,采用最大梯度选择模式后,有可能避开非最小值的局部极值,找到最小失真点(如图2),现行算法避开了用对偶法时所匹配到的局部极值  $E_{\text{SAD}} = 3766$ ,找到了最小失真点。

### 4 实验结果与结论

选择SUSIE标准序列图像,用正向预测方法,并限制最大预测距离为10帧,图像大小为  $352 \times 240$ ,选宏块为  $16 \times 16$ ,搜索窗口为  $32 \times 32$ ,在Pentium 4微机上实现算法。为比较不同算法运动

估值的准确性,规定当前被预测帧的宏块全部用在预测帧所匹配到的最小误差宏块来替换,而无论大小。算法用 BC3.1 编程,且采用相同的一般实现技术。有关算法的匹配精度和匹配时间如表 1 所示。其中,全匹配算法采用传统顺序全匹配。

表 1 相关算法指标比较

Tab. 1 Compare data of correlative arithmetic

指标	方法				
	全匹配	2 维对数	3 步搜索	对偶法	改进方法
平均每帧估值时间(s)	3.83	0.26	0.23	0.16	0.27
替换帧平均恢复信噪比(dB)	35.885	33.276	33.144	33.193	34.860
复杂度(W 为最大绝对偏移量)	$(2W+1)^2$	$2+7\log_2 W$	$1+8\log_2 W$	$2(W+1)$	$2(W+1)$

对快速算法来说,算法的复杂度只能反映算法的一个侧面,不能完全说明算法的实际效果,因为影响算法速度的既有搜索点数的多少,还有算法在实现上的容易程度等,而算法的精度又直接和搜索路径的合理性有关,所以,算法的具体效果还要由实际实验结果而定。从以上结果来看,改进算法的匹配

结果比改进前在精度上有了较多提高,近 2dB,而速度降低不多;算法更接近全匹配算法的精度,且仍具有快速算法的速度,为提高运动图像的压缩速度提供了一个可供参考的方法。算法的缺点是在程序实现上增加了搜索方向控制的难度和时间开销,另外,仍存在典型对偶算法中当遇到局部极值(非最小值)算法终止的不足。由于算法加大了搜索深度,为进一步提高速度,还可以考虑采用增加匹配阈值的办法。

### 参考文献(References)

- 1 Li Wei. Successive elimination algorithm for motion estimation[J]. IEEE Transactions on Image Processing, 1995, 4(1): 105 ~ 107.
- 2 Ghanbari M. The cross-search algorithm for motion-estimation[J]. IEEE Transactions on Communication, 1990, 38(7): 112 ~ 116.
- 3 Liu De-chun. Moving Image Compression Arithmetic[D]. Beijing: Image Center Beijing University of Aeronautics & Astronautics, 1998: 15 ~ 38. [刘德春. 动图压缩方法研究[D]. 北京:北京航空航天大学图象中心, 1998:15 ~ 38.]
- 4 Ma Xiao-hu. Multimedia Data Compression Standard and Realization [M], Beijing: Tsinghua University Press, 1995: 386 ~ 391. [马小虎. 多媒体数据压缩标准及实现[M]. 北京:清华大学出版社, 1995: 386 ~ 391.]