

基于上下文的快速自适应运动估计算法*

田传艳, 沈承东, 李思昆

(国防科技大学 计算机学院, 湖南 长沙 410073)

摘要 高效的运动估计算法是实时视频编解码技术的研究重点。为降低视频编码中运动估计的高计算复杂度问题,提出了基于上下文的快速自适应运动估计算法。该算法在运动估计过程中引入了次最佳匹配点,并且使用了一种基于平行四边形模式的方向性搜索策略。该算法的基本思想是:在块运动矢量估计过程中,利用前面搜索步中得到的块运动矢量及次最佳匹配点与最佳匹配点之间的位置关系,自适应选择一种平行四边形搜索模式,并确定下一步搜索的方向,进行更精确运动矢量的搜索。实验结果表明,该算法有效降低了视频压缩编码中运动估计的运算复杂度。

关键词 视频编码;运动估计;搜索模式;匹配准则

中图分类号 TP391 **文献标识码** A

A Fast Adaptive Motion Estimation Algorithm Based on the Context

TIAN Chuan-yan, SHEN Cheng-dong, LI Si-kun

(College of Computer, National Univ. of Defense Technology, Changsha 410073, China)

Abstract The efficient motion estimation algorithms are the important research hotspot in the real-time video coding-decoding technology. In order to reduce the high computational complexity of motion estimation in the video-coding, a fast adaptive motion estimation algorithm based on the context is proposed. The novel algorithm uses a kind of parallelogram pattern to search directionally, and imports secondary minimum distortion point. The basic idea of the novel algorithm is: In the process of block-matching, this novel fast algorithm uses the former search step's motion vector and the relative location between the minimum distortion point and secondary minimum point to select adaptively one parallelogram model and decide the search direction of next step, and then keeps on searching for more accurate motion vector. Experimental results show that the proposed algorithm reduced the computational complexity of the motion estimation in video-coding effectively.

Key words video-coding; motion estimation; search pattern; matching criterion

块匹配运动估计是视频编码标准 MPEG-1/2/4^[1]和 H.261/263/264^[2]的重要组成部分。块匹配运动估计的目的在于去除两个相邻视频帧之间的时间冗余,降低编码比特率。块匹配算法就是从参考帧中寻找与当前块匹配最佳的块,匹配通常使用计算简单的最小化匹配准则 SAD。最佳匹配块与当前块在 x 与 y 方向上的距离组成当前块的运动矢量。研究表明,使用全搜索(FS)算法进行运动估计的计算量占整个编码计算量的 80%。

为降低视频编码中运动估计的高运算复杂度,人们提出了许多有效的快速运动估计算法^[3~8],其中有代表性的是三步搜索(TSS)、新三步搜索(NTSS)^[4]、四步搜索(4SS)^[5]、梯度搜索(BBGDS)^[6]、钻石搜索(DS)^[7]和六边形搜索(HEXBS)^[8]。在 TSS、NTSS、4SS 和 BBGDS 算法中,搜索模式使用的是不同尺寸的正方形搜索模式;DS 算法使用了钻石形搜索模式,在取得与 TSS、NTSS、4SS 和 BBGDS 算法相同的视频率失真情况下,DS 算法具有更快的运动估计速度;HEXBS 算法则采用了六边形搜索模式,该算法取得了比 DS 算法更好的视频压缩效果,降低了运动估计的运算复杂度,但其搜索模式单一,搜索易陷入局部极小。Zhibo Chen 等人^[9]提出了 UMHexagonS 快速运动估计算法,并被 H.264 视频压缩标准组织 JVT 所

* 收稿日期 2004-11-15

基金项目 国家自然科学基金资助项目(90207019)

作者简介 田传艳(1980—),女,硕士生。

采纳。UMHexagonS 算法是一种使用不对称“十”字形搜索模式、不均匀多六边形栅格搜索模式和扩展的六边形搜索模式的快速运动估计算法。不对称“十”字形搜索模式和不均匀多六边形搜索模式等多种搜索模式的混合使用,使得 UMHexagonS 算法取得了与全搜索非常接近的视频质量和低于全搜索的运算复杂度,但运动估计速度比 HEXBS 算法低。

1 HEXBS 算法分析

HEXBS 算法是一种基于六边形搜索模式的快速运动估计算法。在求得相同的块运动矢量情况下,HEXBS 算法计算的探测点比 DS 算法更少。HEXBS 算法的六边形搜索模式的探测点分布如图 1(a)所示,可以看出 HEXBS 算法搜索模式含有 7 个探测点,其中 1 个探测点为搜索模式的中心探测点,其余 6 个探测点是边探测点,边探测点分布在离中心探测点不同距离的周围点处。图 1(b)和图 1(c)分别是边探测点的大六边形模式的搜索过程和小六边形模式的匹配搜索过程。假设探测点“2”是当前搜索步中满足匹配准则的最佳匹配点,则下一步的六边形搜索模式中,7 个探测点中有 3 个是新添加进来的,如图 1(b)所示“7”、“8”和“9”探测点,因此在下一步中只需计算这 3 个新探测点,若中心探测点“0”是当前搜索步的最佳匹配点,下一步匹配搜索则需要计算 4 个新探测点,如图 1(c)所示的“7”、“8”、“9”和“10”探测点。

从 HEXBS 算法的搜索过程,可以得到 HEXBS 算法在运动估计中探测点个数计算公式:

$$N_{HEXBS}(m_x, m_y) = 7 + 3n + 4 \tag{1}$$

其中 (m_x, m_y) 表示 HEXBS 算法求得的块运动矢量, n 为 HEXBS 搜索模式使用的次数。

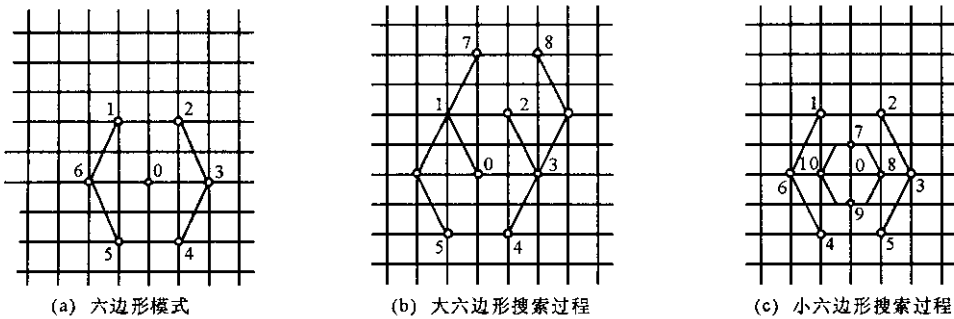


图 1 HEXBS 算法

Fig.1 HEXBS algorithm

假设在全局最小失真匹配点的邻域,块失真呈单调性变化,越接近全局最小失真匹配点,块失真越小。图 1(b)中“2”是上一步搜索中的最佳匹配点,“1”是次最佳匹配点,在下一步搜索中,若新探测点“9”是最佳匹配点,根据上面的假设,探测点“3”的块失真应比探测点“1”的块失真小,“3”应是上一步搜索中的次最佳匹配点,这与“1”是上一步搜索的次最佳匹配点相矛盾,因此,新探测点不可能是下一步搜索的最佳匹配点。图 1(c)中,假设上一步搜索中探测点“0”是最佳匹配点,而“3”是次最佳匹配点,根据前面的块失真单调性假设,下一步搜索的最佳匹配点不可能是“7”、“9”和“10”这 3 个新探测点。从上述分析可以得出,在运动估计过程中,六边形搜索模式的探测点之间存在点冗余,即存在一些对块运动矢量估计作用不大的探测点。

HEXBS 算法中存在上述探测点冗余,原因在于其搜索模式中探测点分布没有体现块的运动趋势。由此提出了基于上下文的快速自适应运动估计算法,简称 CAPS(Context-based Adaptive Parallelogram Search)算法。

2 CAPS 快速运动估计算法

2.1 PS 模式

PS(Parallelogram Search)模式存在 3 种形状,分别如图 2(a)(b)和(c)所示。PS 模式包含 4 个探测点,比 HEXBS 搜索模式少 3 个探测点。PS 模式中相邻探测点坐标差为 $(|\Delta x|, |\Delta y|) = (1, 2)$ 或 $(|\Delta x|,$

$|\Delta y|)=(2,0)$ 。在块全局最佳匹配点的搜索过程中,根据前一步搜索的最佳匹配点和当前步搜索中的最佳匹配点及次最佳匹配点,选择下一步搜索的搜索模式并确定块搜索模式的前进方向。若最佳匹配点在前后两步的搜索模式中不变,则下一步的搜索模式选择扩展的 PS 模式,图 2(d)和(e)为扩展的 PS 模式。图 2(d)为最佳匹配点和次最佳匹配点坐标差为 $(|\Delta x|, |\Delta y|)=(2,0)$ 的扩展 PS 模式,图 2(e)为最佳匹配点和次最佳匹配点坐标差为 $(|\Delta x|, |\Delta y|)=(1,2)$ 的扩展 PS 模式。CAPS 算法的搜索每前进一步,新增加的探测点个数为 2 或 3。

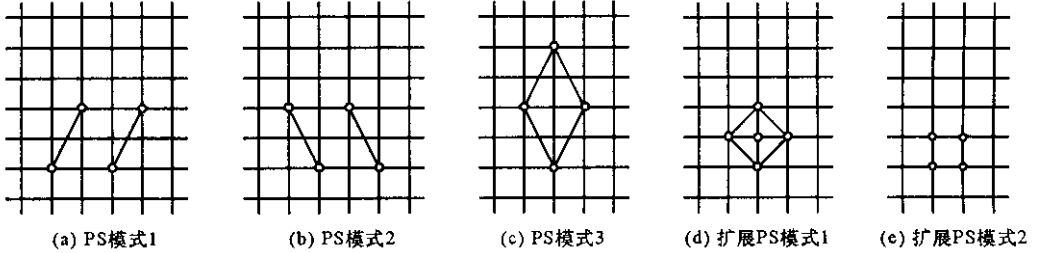


图 2 CAPS 算法

Fig.2 CAPS algorithm

2.2 CAPS 快速运动估计算法

CAPS 算法在运动估计过程中引入了次最佳匹配点,并且使用 PS 模式的方向性搜索策略。为降低块运动矢量估计过程中不必要的搜索,在保证视频质量的同时,设置了提前结束搜索的阈值 T_0 , T_0 定义方法与文献[9]相同。该算法的步骤是:

(1) 确定当前块的初始运动矢量和相应初始匹配点,由相邻的块的运动矢量使用预测公式(2)计算当前块的初始运动矢量,并由此得出相应初始匹配点:

$$mv = \sum_{i=1}^n p_i mv_i \quad (2)$$

其中, mv_i 表示与当前块相邻的参考块 i 的运动矢量, p_i 为初始运动矢量的估计中 mv_i 的权值, n 表示参考块的个数,一般取 3。

(2) 根据上一步得出的运动矢量及次最佳匹配点与最佳匹配点之间的相对位置,选择某一种 PS 模式并确定搜索前进方向,进行搜索模式中探测点的匹配计算,得出当前搜索步的最佳匹配点和次最佳匹配点。若最佳匹配点满足匹配准则,则搜索结束;若最佳匹配点与上一步的最佳匹配点重合,则执行(3),否则重复执行(2)。

(3) 根据上一步得出的最佳匹配点和次最佳匹配点相对位置关系,选择一种扩展的 PS 模式,进行该搜索模式中探测点的匹配计算,得出最佳匹配点,求得块运动矢量,搜索结束。

2.3 CAPS 算法分析

图 3 和图 4 分别是 CAPS 算法和 HEXBS 算法的运动矢量 $(1, -5)$ 的搜索过程。图中较大的长方形黑块表示最后求得的最佳匹配点,而较小的长方形黑块表示搜索过程中某一搜索步中的最佳匹配点,方形黑块表示搜索过程中某一搜索步中的次最佳匹配点。从图中可以看出,为求得运动矢量 $(1, -5)$, HEXBS 算法共计算 23 $(7+3+3+3+4+3)$ 个探测点,而 CAPS 算法共计算 17 $(7+2+2+3+3)$ 个探测点,与 HEXBS 快速算法相比,节省了 6 个探测点的匹配计算。

运动估计过程中使用的探测点个数决定了运动估计的运算复杂度,由 2.2 节可以得出 CAPS 算法在每个块的运动估计过程中使用的探测点个数计算公式:

$$N_{CAPS}(m_x, m_y) = 7 + 2n_1 + 3n_2 \quad (5)$$

其中 (m_x, m_y) 为 CAPS 算法求得的块运动矢量, n_1 为块运动矢量估计过程中,图 2(a)和(b)搜索模式使用的次数, n_2 为块运动矢量估计过程中,图 2(c)(d)和(e)搜索模式使用的次数。

由公式(1)和公式(5)得出,在求得相同块运动矢量 (m_x, m_y) 下, CAPS 算法相对 HEXBS 算法可以节

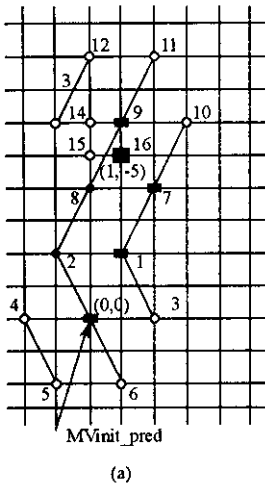


图3 CAPS快速算法搜索过程

Fig.3 The searching process of CAPS algorithm

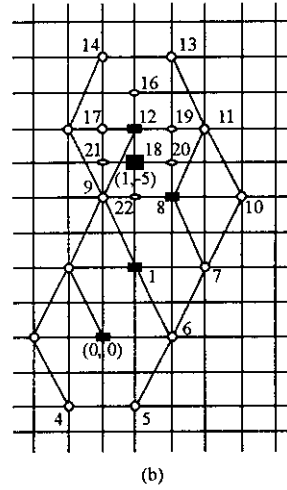


图4 HEXBS快速算法搜索过程

Fig.4 The searching process of HEXBS algorithm

省的探测点个数 ΔN 可用公式表示如下:

$$\Delta N = N_{\text{HEXBS}} - N_{\text{CAPS}} \quad (6)$$

与 HEXBS 算法相比较, CAPS 算法的运动估计速率提高的百分比 SIR (Speed Improvement Ratio) 可表示为:

$$SIR = \frac{N_{\text{HEXBS}} - N_{\text{CAPS}}}{N_{\text{HEXBS}}} \quad (7)$$

由公式(6)和(7)可得出, 块运动矢量越大, ΔN 越大, SIR 也越大; 因此, 视频中场景的变化越剧烈, CAPS 算法节省的探测点个数越多, CAPS 算法的运动估计速率提高的百分比 SIR 值越大。

3 实验与结果分析

为评价 CAPS 算法的性能, 在主频 1GHz、内存 256MB 的 Pentium III PC 机上使用 H.264 参考软件 JM7.6^[10] 实现了 FS 算法、TSS 算法、UMHexagonS 算法和 CAPS 算法。

实验参数如表 1 所示, 将 PSNR (峰值信噪比)、视频压缩编码时间和运动估计时间作为实验结果输出, PSNR 定义如下:

$$PSNR = 10 \lg \left(\frac{255^2}{\text{MSE}} \right) \quad (8)$$

式中, MSE 为重构帧与原始帧之间的均方差。PSNR 值的大小表示运动估计精度的高低。

表1 主要实验参数

Tab.1 The main parameters of experiments

视频序列 视频格式, 图像速率 帧数)	帧序列类型	QP	搜索窗口	搜索步长	其它
Mother_Daughter (cif 30, 150)	IPPP...	28	32 × 32	2pels	失真最优化 (RDO) 方法
Flower (sif 30, 150)		32			Hadamard 变换
Foreman (cif 30, 150)		36			
Mobiler (cif 30, 150)		36			
Stefan (sif 30, 150)		40			

表 2 列出了 5 种视频序列的平均 PSNR 值, 表 3 列出了 CAPS 算法相对于 HEXBS 算法、UMHexagonS 算法和 FS 算法的运动估计速率提高的百分比 (SIR)。

从表 2 中看出, 对于“Mother_Daughter”视频序列, CAPS 算法的 PSNR 值为 35.0367dB, 在相同的比特率下, 与 UMHexagonS 算法相比, PSNR 值提高了 0.017dB。对于“Foreman”视频序列, CAPS 算法的

PSNR 值为 32.4825dB ,与 UMHexagonS 算法相比 ,PSNR 值提高了 0.015dB ;与 HEXBS 算法相比 ,PSNR 值提高了 0.02dB。CAPS 算法与 FS 算法相比 ,视频序列的平均 PSNR 值下降也非常小。

表 2 平均峰值信噪比 (PSNR) (dB)

Tab.2 The average PSNR (dB)

	FS	UMHexagonS	HEXBS	CAPS
Mother_ Daughte(CIF)	35.0575	35.035	35.0375	35.0367
Foreman(CIF)	32.5025	32.4675	32.4625	32.4825
Mobile(CIF)	28.9125	28.91	28.9075	28.9083
Flower(SIF)	28.1125	28.1075	28.1075	28.1075
Stefan(SIF)	30.09	30.08	30.0625	30.0825

表 3 CAPS 算法运动估计平均速度提高百分比 (%)

Tab.3 The average SIR in motion estimation by CAPS algorithm (%)

	CAPS(/FS)	CAPS(/UMHexagonS)	CAPS(/HEXBS)
Mother_ Daughte(CIF)	99.0345	39.591	0.68
Foreman(CIF)	98.6655	66.862	12.276
Mobile(CIF)	98.691	74.156	16.103
Flower(SIF)	98.465	71.836	1.025
Stefan(SIF)	98.5865	72.857	9.555

表 3 中 ,“ CAPS(/FS)”表示与 FS 算法比较 ,CAPS 算法的平均运动估计速率提高百分比 ;“ CAPS(/UMHexagonS)”表示与 UMHexagonS 算法比较 ,CAPS 算法的平均运动估计速率提高百分比 ;“ CAPS(/HEXBS)”表示与 HEXBS 算法比较 ,CAPS 算法的平均运动估计速率提高百分比。对 “ Foreman ” (cif ,150 帧 ,30fps)视频序列 ,CAPS 算法与 FS 算法相比平均运动估计速度提高了 98.6655% ,与 UMHexagonS 算法相比运动估计速度提高了 66.862% ,与 HEXBS 算法相比运动估计速度提高了 12.276%。对于“ Mobile ” (cif ,150 帧 ,30fps)视频序列 ,CAPS 算法与 FS 算法相比平均运动估计速度提高了 98.691% ,与 UMHexagonS 算法相比平均运动估计速度提高了 74.156% ,与 HEXBS 算法相比平均运动估计速度提高了 16.103%。从 CAPS(/HEXBS)的 SIR 值得出 ,视频图像运动越剧烈 ,CAPS 算法的运动估计速率提高越明显。

4 结论

提出了一种基于上下文的快速自适应运动估计算法 ,该算法有效降低了视频压缩编码中运动估计的高计算复杂度 ,同时视频质量损失非常小。

参考文献 :

- [1] Information Technology- Generic Coding of Audio- Visual Objects[S]. Part2 :Visual ,ISO/IEC 14 496 - 2 (Mpeg-4 Video) , 1999 .
- [2] ITU-T Rec. H.264/ISO/IEC 11496 - 10 , Advanced Video Coding[S]. Final Committee Draft , Document JVT-G050 , March 2003 .
- [3] Ghanbari M. The Cross-search Algorithm for Motion Estimation[J]. IEEE Trans. Commun. , 1990 , 38 : 950 - 953 .
- [4] Li R , Zeng B , Liu M L. A New Three-step Search Algorithm for Block Motion Estimation[J]. IEEE Trans. Circuits Syst. Video Technol. , 1994 , 4 : 438 - 442 .
- [5] Po L M , Ma W C. A Novel Four-step Search Algorithm for Fast Block Motion Estimation[J]. IEEE Trans. Circuit Syst. Video Technol. , 1996 , 6 : 313 - 317 .
- [6] Liu L K , Feng E. A Block-based Gradient Descent Search Algorithm for Block Motion Estimation in Video Coding[J]. IEEE Trans. Circuits Syst. Video Technol. , 1996 , 6 : 419 - 423 .
- [7] Zhu S , Ma K K. A New Diamond Search Algorithm for Fast Block-matching Motion Estimation[J]. IEEE Trans. Image Processing , 2000 , 9 : 287 - 290 .
- [8] Zhu C , Lin X , Chau L. Hexagon-based Search Pattern for Fast Block Motion Estimation[J]. IEEE Trans. Circuits Syst. Video Technol. , 2002 , 12 : 349 - 355 .
- [9] Fast Integer Pel and Fractional Pel Motion Estimation for JVT[P]. ISO/IEC JTC1/SC29/WG11 and ITU-T SG16 Q.6 , December , 2002 .
- [10] JM7.6 , Reference Software of JVT[R]. http://bs.hhi.de/~suehring/tml/download/old_jm/ , 2004 .

