

# 基于 H.264 的多参考帧运动估计快速算法

石迎波, 吴成柯

(西安电子科技大学综合业务网国家重点实验室, 西安 710071)

**摘要:** 提出一种应用于 H.264 中的多参考帧快速整像素运动估计算法, 该算法通过有效地预测多参考帧中的搜索起始点和将菱形算法扩展到多参考帧情形, 在很大程度上降低多参考帧运动搜索的计算量。试验结果表明, 与 H.264 参考软件 JM9.6 中的快速算法 UMHexagonS 相比, 该算法保持了较好的图像质量且码率几乎不变, 运动估计时间平均减少约 60%。

**关键词:** 多参考帧; 运动估计; 搜索起始点; 菱形搜索

## Fast Multi-frame Motion Estimation Algorithm Based on H.264

SHI Ying-bo, WU Cheng-ke

(National Key Lab on Integrated Service Networks, Xidian University, Xi'an 710071)

**【Abstract】** This paper proposes a novel fast multi-frame motion estimation algorithm based on H.264. The proposed algorithm can reduce the computational complexity by predic the initial search point prediction of the reference frames and extending the diamond search to multiple reference frames. Experimental results show that the algorithm can decrease the encoding time by 60% while maintaining nearly unchanged PSNR of pictures and bit rate compared to the fast motion estimation algorithm named UMHexagonS in H.264 reference software JM9.6.

**【Key words】** multi-frame; motion estimation; initial search point; diamond search

### 1 概述

H.264<sup>[1]</sup>是最新的视频编码国际标准,其目的是在所有的码率条件下都能够提供高质量的视频图像。H.264 编码器采用了很多新的编码技术,如多尺寸块编码、多参考帧预测等,这些新技术使H.264 具有很高的编码效率,但也极大地增加了编码器的计算复杂度,使其在实时通信应用方面受到了很多限制。

H.264 的高压缩性能是以提高运算复杂度为代价的,其中在帧间预测中采用的 7 种预测块模式、高精度运动矢量和多参考帧预测技术是引起运算量增加的主要因素。H.264 编码时,根据率失真准则对当前编码宏块选取最佳的编码模式,达到最佳的编码压缩效果。最简单的运动估计算法是全搜索,虽然该算法能得到最佳的匹配点,但运算量庞大,因此,人们提出了一系列的优化算法。这些算法主要是通过减少搜索的次数来寻找最佳的匹配点,如菱形搜索法(Diamond Search, DS)<sup>[2]</sup>、非对称十字形多层次六边形栅格运动搜索(UMHexagonS)<sup>[3]</sup>等。其中UMHexagonS算法已被 JVT采纳,可获得与全搜索算法几乎接近的图像PSNR性能。H.264 采用多参考帧技术,使得帧间预测的可选范围更大,预测也更为精确。试验表明,运动估计占总编码时间的 60%(1 个参考帧)~80%(5 个参考帧)以上<sup>[3]</sup>。文献[4]提出了一种H.264 多参考帧的快速选择算法。

上述算法在 H.264 编码中解决了一个参考帧的快速搜索或多参考帧的快速选择,并未将两者结合起来。本文提出一种基于 H.264 的多参考帧运动估计快速算法,该算法通过确定各参考帧的搜索起始点,将菱形算法扩展到多参考帧情形上,对各参考帧同时进行搜索,并在搜索匹配块的过程中同时考虑最佳参考帧的选择。试验表明整个算法的性能优于现

有算法,更利于实时应用。

### 2 H.264 多参考帧运动估计的基本原理

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,各块可使用不同的参考帧进行编码,当对 8×8 块模式进行运动估计时,可进一步独立地划分为 8×8、8×4、4×8 和 4×4 的亚宏块,但这 4 种分割块必须由同一参考帧预测编码。

H.264 的多参考帧运动估计是通过率失真优化准则(式(1))确定宏块的最优匹配模式,得到的结果即为该宏块的最终编码模式。

$$J_{\text{motion}}(s, c, \mathbf{mv}, \text{ref} | \lambda_{\text{motion}}) = \text{SAD}(s, c(\text{ref}, \mathbf{mv})) + \lambda_{\text{motion}}(R(\mathbf{mv} - \text{pred}) + R(\text{ref})) \quad (1)$$

其中,  $s$  为当前块像素值;  $\text{ref}$  为要选择的参考帧,  $\mathbf{mv} = (mv_x, mv_y)^T$  为参考帧  $\text{ref}$  的运动矢量;  $\text{pred}$  为运动矢量预测值;  $c(\text{ref}, \mathbf{mv})$  为在运动矢量  $\mathbf{mv}$  和参考帧  $\text{ref}$  下的重构视频信号;  $\lambda_{\text{motion}}$  为拉格朗日乘子; 码率  $R$  包括运动矢量与其预测值的差值编码比特和参考帧的编码比特;  $\text{SAD}$  (绝对误差和)为原始信号与重构信号之间的绝对值和; 采用式(2)计算:

$$\text{SAD}(s, c(\text{ref}, \mathbf{mv})) = \sum_{i=1, j=1}^{B_1, B_2} |s[i, j] - c[i - mv_x, j - mv_y]| \quad (2)$$

其中,  $B_1, B_2$  表示块的水平和垂直像素数,取值 16, 8 或 4。

**基金项目:** 国家自然科学基金资助重点项目(60532060); 国防科技重点实验室基金资助项目

**作者简介:** 石迎波(1981 -), 男, 博士研究生, 主研方向: 图像压缩编码, 视频通信; 吴成柯, 教授、博士生导师

**收稿日期:** 2007-06-30 **E-mail:** ybshi@mail.xidian.edu.cn

### 3 基于 H.264 多参考帧的运动估计快速算法

#### 3.1 搜索起始点预测

搜索起始点的精确预测可以有效地减少搜索点数,从而节约运动估计消耗的时间。文中设定可供选择的参考帧数为 5,此例很容易扩展到参考帧数为  $N$  的情况。设  $F(t)$  为当前编码帧,可供选择的参考帧为  $F(t-1) \sim F(t-5)$  等 5 个已编码重建帧,定义各参考帧的搜索起始点依次为  $MVP_{t-1} \sim MVP_{t-5}$ 。为确定各个参考帧中的搜索起始点位置,将每帧相对于前一帧的运动矢量都保存下来,即定义 5 个块存储单元  $M_i (0 \leq i < 5)$ ,记录帧  $F(t-i)$  中每个宏块的 7 种块类型在  $F(t-i-1)$  中预测得到的运动矢量  $MV_{t-i}$ 。

如图 1 所示,经过搜索得到当前编码块  $E$  (大小为  $B_1 \times B_2$ ) 在前一参考帧  $F(t-1)$  中的匹配位置,设运动矢量为  $MV_t$ ,令

$$MVP_{t-1} = MV_t \quad (3)$$

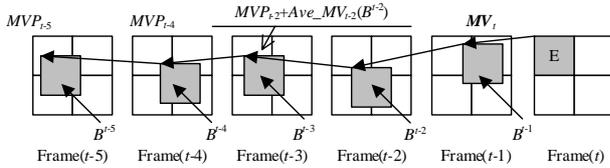


图 1 H.264 多参考帧搜索起始点示意图

然后确定其余 4 个参考帧的搜索起始点。由  $MV_{t-1}$  能够得到参考帧  $F(t-i)$  中预测匹配块  $B^{t-i}$  的每个像素点在前一参考帧  $F(t-i-1)$  中的运动矢量  $MV_{m,n,t-i}$ ,定义块  $B^{t-i}$  在  $F(t-i-1)$  中的平均运动矢量为

$$Ave\_MV_{t-i}(B^{t-i}) = \frac{1}{B_1 \times B_2} \sum_{m=0}^{B_1-1} \sum_{n=0}^{B_2-1} MV_{m,n,t-i} \quad (4)$$

则当前块  $E$  在  $F(t-i) (i=2,3,4,5)$  的搜索起始点为

$$MVP_{t-i} = MVP_{t-i+1} + Ave\_MV_{t-i+1}(B^{t-i+1}) \quad (5)$$

#### 3.2 算法描述

在 H.264 参考模型 JM9.6<sup>[5]</sup> 中的 UMHexagonS 算法采用非对称的十字搜索和不均匀的多层次六边形栅格搜索相结合的思想,搜索模式多样化,这样虽然可以保证较高的搜索精度,但与其他搜索算法相比,计算量明显提高,并且不易于硬件实现。所以,在本文的算法中采用的是菱形搜索算法,如图 2 所示,即有 9 个搜索点的大模板 LDSP 和 5 个搜索点的小模板 SDSP。一次搜索的 9 点或 5 点中的最小块误差点即为最佳点。令 LDSP 每 3 个相邻的搜索点为一组,共 8 组,即 123, 234, 345, 456, 567, 678, 781 和 812,依次对其编号为 1~8。定义 SAD 之和最小的一组为  $Group_{min}$ ,其对应的 3 个 SAD 之和为  $SAD_{min}$ 。

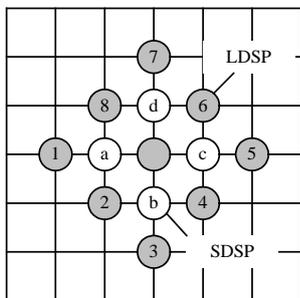


图 2 DS 算法基本搜索点示意图

本文的快速算法是当前编码块  $E$  同时在所有可供选择的参考帧中进行运动搜索,在搜索过程中根据各个参考帧的匹配情况,去除和当前编码块相关性最小的参考帧,进而最终得到最佳参考帧和对应的运动矢量。算法步骤具体如下:

Step1 将所有可供选择的参考帧标记为可选,并令各参考帧的  $Group_{min}$  初始位置编号为 0。用 ARPS 快速算法<sup>[6]</sup> 搜索出参考帧  $F(t-1)$  的匹配位置,即得到  $MV_t$ 。

Step2 对参考帧  $F(t-1)$  用图 3 所示的搜索模式,以式(3)为中心点进行搜索,对其余可选参考帧依次用 LDSP 进行搜索,分别以式(5)为搜索中心点,则可以得到编码块在各个参考帧的最佳点。如果在某些(1 个或多个)参考帧中的最佳点是中心点,执行 Step3;如果各个参考帧中的最佳点都不是中心点,执行 Step4。

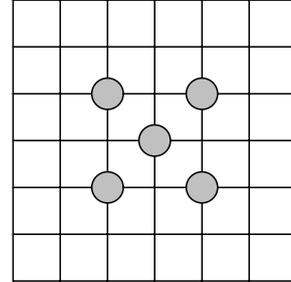


图 3 参考帧  $F(t-1)$  的一种搜索点示意图

Step3 记录最佳点为中心点的坐标,  $Group_{min}$  的位置编号(0~8)及对应的参考帧,并将该参考帧标记为不可选。如果目前没有标记为可选的参考帧,执行 Step5;否则,执行 Step4。

Step4 在标记为可选的参考帧中,以各自对应的最佳点为中心点,进行 LDSP 搜索,此时计算出编码块在这些参考帧中的各自的最佳点。如果在某些(1 个或多个)参考帧中的最佳点是中心点,执行 Step3;如果各个参考帧中的最佳点都不是中心点,并且可选的参考帧为 1 个以上,计算各参考帧对应的  $SAD_{min}$ ,将其中最大的  $SAD_{min}$  对应的参考帧标记为不可选,执行 Step4;如果最佳点不是中心点,且仅有 1 个可选的参考帧,执行 Step4。

Step5 在 Step3 中记录的参考帧中,以各自的最佳点为中心点,进行菱形的 SDSP 的快速搜索。在某个参考帧内,如果  $Group_{min}$  编号为 1,3,5 或 7,再搜索 2 个点,如图 2 所示,当 1,2,3 为  $Group_{min}$ ,则继续搜索 a, b 两点;如果  $Group_{min}$  编号为 2,4,6 或 8,则再搜索 1 个点,如 2,3,4 为  $Group_{min}$  继续搜索 b。依次可得到 Step3 中所记录下的所有参考帧中所对应的最佳搜索点,进行 Step6。

Step6 依次计算在 step5 中得到的所有参考帧中对应的最佳搜索点的率失真代价  $J_{motion}$ ,根据  $J_{motion}$  最小原则,选取当前编码块的最佳参考帧和对应的运动矢量。

在 Step1 中,ARPS 快速算法利用已编码块运动矢量,设置多个搜索参考起始点,然后不断运用 SDSP 寻找匹配块,同时保证了搜索精度和搜索速度。搜索后的运动矢量  $MV_t$  不仅要作为参考帧  $F(t-1)$  的搜索起始点,还要保存至存储单元  $M_0$  中,以备后续帧编码时用于参考帧搜索起始点的预测。在 Step2 中对参考帧  $F(t-1)$  采用的图 3 所示的搜索模式不存在  $Group_{min}$ ,即搜索后的  $Group_{min}$  编号仍为 0。

在完成多参考帧的整数点运动估计之后,还要进行分数像素级的运动搜索,本文采用参考模型 JM9.6 所用算法,即基于中心点的亚像素点快速搜索(CBFPS)<sup>[3]</sup>。

### 4 试验与分析

通过修改 JM 9.6 实现本文提出的多参考帧运动估计快速算法,然后同 JM9.6 中所采用的快速全搜索算法 FastFS 和快速搜索算法 UMHexagonS 进行性能比较。编码参数设置如下:

采用 CABAC 熵编码, 运用率失真优化模式判别方法; 允许码率控制, 目标码率分别为 64 Kb/s(QCIF)和 128 Kb/s(CIF); 采用 IPPP 序列编码方式, 共编码 100 帧, 运动估计搜索范围为 16, 1/4 像素精度, 支持 7 种块模式, 参考帧数为 5, 帧频为 10 fps。

由表 1 可以看出, 使用 UMHexagonS 算法与使用 FastFS 算法的编码质量基本相当, 本文算法与 FastFS 相比编码质量略有下降, PSNR 下降不超过 0.1 dB, 对运动较小的序列 PSNR 变化完全可以忽略, 平均码率增加不到 1%。同时, 本文算法显示出了更大的编码速度优势, 采用本文算法可以明显降低编码器的运动搜索时间, 运动估计平均编码时间仅为 FastFS 算法的 13%。

表 1 新算法与参考模型性能比较

序列	格式	平均码率/(Kb·s <sup>-1</sup> )						亮度分量平均 PSNR/dB			运动估计时间/s		
		FastFS		UMHex		本文算法		FastFS		UMHex		本文算法	
		FastFS	UMHex	FastFS	UMHex	FastFS	UMHex	FastFS	UMHex	FastFS	UMHex	FastFS	UMHex
Container	QCIF	64.05	64.08	64.12	42.41	42.36	42.34	2.26	0.47	0.17			
Akiyo	CIF	128.04	127.92	127.98	44.49	44.48	44.47	9.21	2.10	0.77			
Silent	QCIF	64.11	64.08	64.16	41.54	41.51	41.48	2.22	0.55	0.21			
Foreman	CIF	128.13	128.12	128.25	36.80	36.77	36.72	9.60	3.18	1.34			
Football	QCIF	64.10	64.10	64.20	30.12	30.08	30.03	2.41	1.05	0.45			
Mobile	CIF	128.13	128.18	128.34	28.44	28.40	28.34	9.17	3.41	1.47			

与 UMHexagonS 算法相比, 本文算法在搜索速度和图像编码质量之间取得了更好的折中, 因而其整体性能更优: 在多参考帧运动搜索时间减少为原来 1/3 左右的同时, 重建图像质量仅略有下降。这是由于相比 UMHexagonS 算法采用的多参考帧搜索起始点的简单的线性预测模型, 本文算法保存了每帧相对于前一帧的运动矢量, 这对于多参考帧情形下的快速搜索很有帮助, 利用该运动矢量, 可以精确地预测当前编码块在每一参考帧的搜索起始点; 同时本文算法可以对所有的参考帧同时进行搜索, 并在搜索过程中去除相关性最小的参考帧, 而 UMHexagonS 算法是对各参考帧进行逐一搜索, 没有考虑最佳参考帧的选取。

(上接第 206 页)

可以看出, 用 OSV 恢复出的结构图像中出现了明显的阶梯现象, 如图 2(b)所示; 用 DN 模型恢复出的结构图像中阶梯现象明显减少, 如图 2(d)所示。图 2(b)、图 2(d)表明, OSV 模型分解出的结构成分仍然含有纹理成分, 而 DN 模型能很好地分离结构和纹理部分。

#### 4 结束语

本文提出了一种将图像分解为结构和纹理部分的新模型, 数值实验表明, 该模型在保持图像边缘的同时, 能很好地减少阶梯现象, 并将纹理和结构分开。但由于该模型的参数比较多, 它们的选择对实验结果的影响非常大, 因此本文只对这一方面做了简单的考虑, 下一步将进行全面深入的研究。

#### 参考文献

[1] Meyer Y. Oscillating Patterns in Image Processing and Nonlinear Evolution Equations[M]//University Lecture Series. Boston, USA: American Mathematical Society, 2001.

[2] Vese L A, Osher S J. Modeling Textures with Total Variation Minimization and Oscillating Patterns in Image Processing[J]. Journal of Scientific Computing, 2003, 19(11): 553-572.

#### 5 结束语

本文提出了一种有效的基于 H.264 多参考帧的运动估计快速算法, 该算法给出了多参考帧搜索起始点的预测方法, 各参考帧采用菱形搜索, 同时在搜索过程中去除相关性最小的参考帧, 使得多参考帧运动搜索所需的个数大大减少, 有效地减少了运动估计的时间, 并且该算法易于硬件并行实现。试验结果表明本文提出的算法大大降低了整像素运动估计的运算复杂度, 与 JM9.6 采用的快速运动估计算法 UMHexagonS 相比, 在保持较好的图像编码质量的同时, 节省了编码时间。

#### 参考文献

[1] Wiegand T, Sullivan G J, Bjontegaard G, et al. Overview of the H.264/AVC Video Coding Standard[J]. IEEE Trans. on Circuits and Systems for Video Technology, 2003, 13(7): 560-576.

[2] Zhu Shan, Ma Kaikuang. A New Diamond Search Algorithm for Fast Block-matching Motion Estimation[J]. IEEE Trans. on Image Processing, 2000, 9(2): 287-290.

[3] Chen Zhibo, Zhou Peng, He Yun. Fast Integer Pel and Fractional Pel Motion Estimation for JVT[C]//Proc. of the 6th Meeting on Joint Video Team(JVT) of ISO/IEC MPEG & ITU-T VCEG. Awaji, Island, JP: [s. n.], 2002-12.

[4] Ting C W, Po L M, Cheung C H. Center-biased Frame Selection Algorithm for Fast Multiple-Frame Motion Estimation in H.264[C]//Proc. of IEEE International Conference on Neural Networks and Signal Processing. Nanjing, China: [s. n.], 2003.

[5] H.264 Reference Software. JM9.6[Z]. (2005-06-30). [http://iphome.hhi.de/suehring/tml/download/old\\_jm/jm96.zip](http://iphome.hhi.de/suehring/tml/download/old_jm/jm96.zip).

[6] Ma Kaikuang, Qiu Gang. An Improved Adaptive Rood Pattern Search for Fast Block-matching Motion Estimation in JVT/H.26L[C]//Proc. of IEEE International Symposium on Circuits and Systems. Bangkok, Thailand: [s. n.], 2003: 708-711.

[3] Osher S, Sole A, Vese L. Image Decomposition and Restoration Using Total Variation Minimization and the H<sup>1</sup> Norm[J]. Journal of Multiscale Modeling and Simulation, 2003, 1(3): 349-370.

[4] Le T, Vese L. Image Decomposition Using the Total Variation and div(BMO)[J]. Journal of Multiscale Modeling and Simulation, 2005, 4(2): 390-423.

[5] Aujol J F, Aubert G, Blanc-Feraud L, et al. Image Decomposition: Application to Textured Images and SAR Images[R]. INRIA, Technical Report: I3S/RR-2003-01-FR, 2003.

[6] Chambolle A. An Algorithm for Total Variation Minimization and Applications[J]. Journal of Mathematical Imaging and Vision, 2004, 20(1/2): 89-97.

[7] Ekeland I, Temam R. Convex Analysis and Variational Problems [M]. Amsterdam, Netherlands: North-Holland Publishing Co., 1976.

[8] Chambolle A, Lions P. Image Recovery via Total Variation Minimization and Related Problems[J]. Numer. Math., 1997, 76(2): 167-188.

[9] Chen Yunmei, Levine S, Rao M. Variable Exponent, Linear Growth Functionals in Image Processing[J]. SIAM Journal on Applied Mathematics, 2006, 66(4): 1383-1406.