

# 自适应 AVS\_M 分数像素运动估计快速算法<sup>\*</sup>)

魏志强 李翠苹 刘敏 王岩 王莉  
(中国海洋大学计算机科学系 青岛 266100)

**摘要** 为了减小分数像素运动估计的计算量,提出了一种适用于 AVS\_M 的分数像素快速搜索算法。该算法采用更多的预测矢量提高分数预测矢量的准确性;在分数预测矢量不够准确时,利用自适应阈值的判断跳过不必要搜索过程,或者利用分析分数像素候选点的匹配误差关系排除可能性小的点。实验分析表明:对于不同运动特征的视频序列,该算法在保证图像质量和编码效率基本不变的同时,节省了 61.88%~87.06% 计算量。

**关键词** AVS\_M, 视频编码, 运动估计, 分数像素搜索

## Fast Fractional Pixel Motion Estimation Algorithm for AVS\_M

WEI Zhi-qiang LI Cui-ping LIU Min WANG Yan WANG Li  
(Department of Computer Science, Ocean University of China, Qingdao 266100, China)

**Abstract** A fast fractional pixel motion estimation algorithm is proposed for AVS\_M in order to reduce the computation of fractional pixel ME. More motion vector predictions are utilized to enhance the accuracy. It adopts adaptive threshold decision to stride over unnecessary search or analyzes the matching error relationship of the fractional pixel candidate points to exclude the little possible points, when the predicted motion vector is not accurate. Experimental results show that, to all kinds of video sequences, the proposed method can reduce 71.94%~87.06% computation cost, while keeping the same image quality and bit rate.

**Keywords** AVS\_M, Video coding, Motion estimation, Fractional pixel search

## 1 引言

AVS\_M<sup>[1]</sup>标准是中国自主制定的数字音视频编码系列标准(AVS)中的第七部分:移动视频编码标准,是我国拥有自主知识产权的面向下一代移动通信的视频编码标准。运动估计是 AVS\_M 视频编码的关键部分,也是计算量最大的部分。因此,快速运动估计算法一直都是研究的热点,许多学者对快速整像素运动估计算法进行了大量的研究<sup>[2-5]</sup>,加快了整像素运动估计的速度。在文献[4]中每个块的整像素的搜索点数降到了平均不到 6 个点。而每个块分数像素运动估计的全搜索算法则搜索 17 个点,所以分数像素运动估计在整个运动估计中所占的计算量已不可忽视了。因此,如何减小分数像素运动估计的计算量,提高分数像素搜索的速度,就成了一个重要的问题。

文献[6]中提出了一种基于分数像素搜索的抛物面预测算法(PPFPS),结合了抛物面预测和定向更新算法来估计其他分数像素的运动矢量。但是这种算法只有在整像素运动估计的搜索模型是钻石型,并且最佳整像素周围的四个钻石顶点的误差值都是有效的才能执行。文献[5]提出了中心偏置分数像素搜索算法(CBFPS),直接检测所预测的分数像素位置,并在其周围做精细搜索,可以解决文献[6]中算法的问题。但如果预测的分数像素位置不够准确,算法易陷入局部最小点。

本文提出一种分数像素快速搜索算法。实验结果显示该

算法既保证了分数像素搜索的准确性,又有效地减少了搜索点数。而且算法不需要测试最优整像素运动矢量周围点的匹配误差,可以与任意整像素搜索算法配合使用。

## 2 分数像素全搜索算法

分数像素的全搜索算法如图 1 所示。首先,计算整像素最佳匹配点 E 及其周围的 8 个 1/2 像素点 1, 2, 3, 4, 5, 6, 7, 8 的匹配误差,比较各点的匹配误差,求出具有最小匹配误差点,即最优 1/2 像素匹配点。假设 3 为最优 1/2 像素匹配点,搜索计算 3 周围的 8 个 1/4 像素点 a, b, c, d, e, f, g, h 的匹配误差,进行比较求得最优 1/4 像素匹配点。

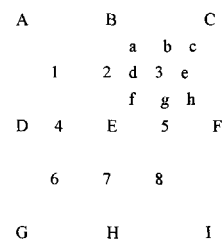


图 1 分数像素全搜索算法

为了更准确地选择分数像素运动向量, AVS\_M 定义了式(1),对分数像素运动估计的匹配误差进行计算,并选取最小匹配误差点为最优匹配点。

$$J(\vec{m} | \lambda_{motion}) = SATD(s, c(\vec{m})) + \lambda_{motion} \cdot R(\vec{m} - \vec{p}) \quad (1)$$

<sup>\*</sup>)国家自然科学基金(60602017),山东省中青年科学家基金(2005BS01001),山东省科技攻关项目(2004GG2005112)。魏志强 教授,博士生导师,主要研究方向为移动计算、图像处理与计算机视觉、视频编码和传输、智能机器人技术;李翠苹 研究生;刘敏 博士后;王岩 研究生;王莉 研究生。

其中,  $\vec{m}=(m_x, m_y)^T$  为当前正在考察的运动向量,  $\vec{p}=(p_x, p_y)^T$  为当前块的运动向量的预测值。  $R(\vec{m}-\vec{p})$  为编码运动向量所需要的比特数, 通过查表得到。  $\lambda_{\text{motion}}$  为运动估计的拉格朗日因子, SATD 按下面方式计算。

计算原始值和预测值之间的差值:

$$d(x, y) = s(x, y) - c(x - m_x, y - m_y) \quad (2)$$

对  $d$  进行二维的 Hadamard 变换, 得到  $dT$ :

$$\text{SATD}(s, c(\vec{m})) = (\sum_{x,y} |dT(x, y)|) / 2 \quad (3)$$

### 3 分数像素快速算法

#### 3.1 基于预测矢量的分数像素搜索

为了获得精确的预测运动矢量, AVS\_M 使用周围编码重建的相邻块的运动矢量对当前块进行预测, 获得预测矢量 ( $MV_{\text{pred}}$ )。 本文选取  $MV_{\text{pred}}$  作为当前块分数像素运动矢量的一个预测矢量。

AVS\_M 引入了 7 种不同的块大小 ( $16 \times 16, 16 \times 8, 8 \times 16, 8 \times 8, 8 \times 4, 4 \times 8, 4 \times 4$ )。 可以利用较大尺寸模式块的运动矢量来预测小尺寸模式块的运动矢量<sup>[3]</sup>, 称之为上层预测。 比如,  $16 \times 16$  为  $16 \times 8, 8 \times 16$  的上层模式,  $8 \times 16$  为  $8 \times 8$  的上层模式。 本文以上层预测得到的分数像素运动矢量 ( $MV_{\text{up-layer}}$ ) 作为另一个预测矢量。

为描述方便, 定义最优整像素运动矢量与  $MV_{\text{pred}}$  或者  $MV_{\text{up-layer}}$  整像素部分相同的块为预测块。

将预测运动矢量  $\text{pred\_mv}$  和整像素搜索得到的最优整像素运动矢量  $mv$  以  $1/4$  像素单位表示, 当前块为预测块时, 就可以按照式(4)提取预测分数矢量  $\text{frac\_pred\_mv}$ 。

$$\text{frac\_pred\_mv} = (\text{pred\_mv} - mv) \% 4 \quad (4)$$

为了验证预测块的预测分数像素运动矢量的准确性, 对预测分数像素运动矢量和分数像素全搜索后得到的最优分数像素运动矢量的偏差 ( $d$ ) 进行了统计。  $d = |x_b - x_p| + |y_b - y_p|$ , 其中  $(x_p, y_p)$  为预测分数像素运动矢量,  $(x_b, y_b)$  为分数像素全搜索得到的最优分数像素运动矢量。

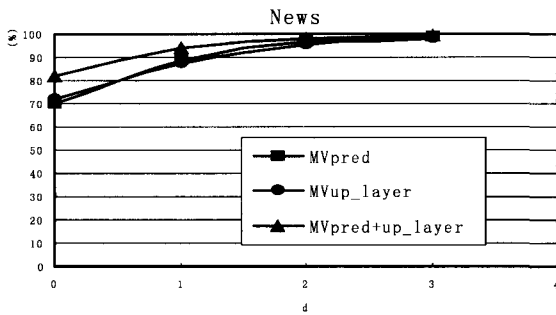


图2 News 预测分数像素运动矢量偏差

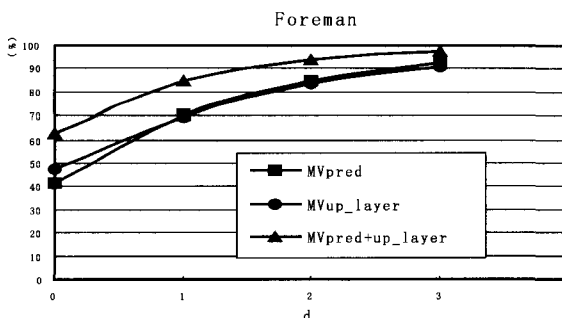


图3 Foreman 预测分数像素运动矢量偏差

从图 2 和图 3 可以看出, 同时利用  $MV_{\text{pred}}$  和  $MV_{\text{up-layer}}$  时, 预测矢量的准确性有明显的提高。 如果将  $MV_{\text{pred}}$  或者  $MV_{\text{up-layer}}$  作为初始搜索点, 做三步小菱形搜索, 准确率达到 97% 以上。 因此, 取具有最小匹配误差的分数像素预测运动矢量作为初始搜索点, 做三步小菱形搜索得到最终的运动矢量。

在小菱形搜索过程中, 进行有选择性的搜索。 如图 4 所示, 在新的菱形搜索中仅搜索与前一步小菱形中的最优和次优点位置相同的点。 当最小匹配误差的点位于菱形的中心或者搜索已经达到三步时, 算法终止。

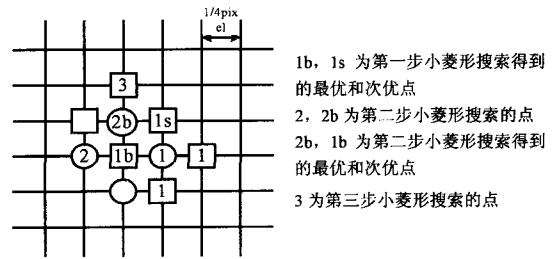


图4 有选择性的小菱形搜索

从图 2 和图 3 还可以看出, 所有待编码块中, 最终的分数像素运动矢量大约有 80%~60% 与预测矢量相同。 如果能够预测出这些编码块, 就可以跳过小菱形搜索过程, 省去大量不必要的计算。 为此, 设置自适应阈值  $TH1$ , 若匹配误差小于  $TH1$ , 表明分数像素预测运动矢量已经足够精确, 立即终止搜索过程。

#### 3.2 精细搜索

预测矢量的整像素部分与整像素运动搜索得到的最优运动矢量不同时, 预测的分数像素运动矢量不准确, 不能利用。 可以利用阈值判决跳过不必要的搜索过程, 或者利用整像素最佳匹配点周围分数像素点匹配误差的关系来排除一些可能性较小的点。

##### 3.2.1 阈值判决

对一些典型视频序列的整像素、分数像素全搜索后得到的运动矢量进行统计, 得到表 1。

序列	1/2 像素与整像素运动矢量相同的比例
Akiyo	87.07%
Silent	80.63%
Paris	78.05%
Tempete	67.82%

从表 1 可以看出, 运动程度较小的序列 Akiyo, 大概有 87.07% 左右的块,  $1/2$  像素精度的运动估计的结果和整像素运动估计的结果是基本一样的。 因此, 可以通过在整像素搜索之后设定阈值, 进行阈值判决跳过不必要的  $1/2$  像素精度搜索, 直接在整像素最佳匹配点周围进行  $1/4$  像素精度搜索。

由于自然物体运动的连续性, 编码块和相邻块及相邻帧中同一位置块具有很强的相关性。 对多种视频序列的统计分析表明, 当前编码块与相邻块及相邻帧中同一位置块的匹配误差仍具有很强的相关性。 为此, 本文算法采用了如下方法确定阈值:

$$TH2 = a \times \min(\text{Min}J_1, \text{Min}J_2, \text{Min}J_3, \text{Min}J_4) + b \quad (5)$$

其中  $\text{Min}J_1, \text{Min}J_2, \text{Min}J_3, \text{Min}J_4$  为当前块的左、上、右上 3 个相邻块和参考帧同一位置块在  $16 \times 16$  模式下的最小匹配

误差,  $a=(H \times V) / 256$ ,  $H$  为当前块的行像素数,  $V$  为当前块的列像素数,  $b$  为值 128。

这种方法不受视频运动类型的影响, 具有较强的自适应性, 而且能准确确定恰当的阈值在搜索过程中进行阈值判决, 有助于以较小的搜索代价得到较好的搜索结果。

### 3.2.2 1/2 像素搜索策略

对于需要进行 1/2 像素精度搜索的块, 利用最优整像素运动矢量周围 1/2 像素点匹配误差的关系, 来排除一些可能性较小的点, 从而加速搜索过程。

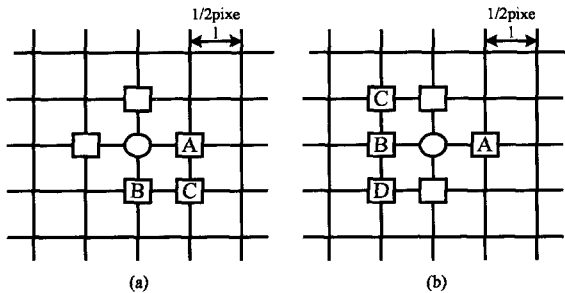


图 5 1/2 像素搜索示意图

以最优整像素运动矢量为搜索中心, 1/2 像素为单位做一步小菱形搜索, 保存最优和次优点的位置和匹配误差, 根据最优和次优点的位置决定其它需要测试的点。如图 5 (a) 所示, 如果最优和次优点 B, A 不在一条直线上, 那么只需再增加测试一点 C。在图 5 (b) 中, 最优和次优点 B, A 在一条直线上, 那么需要再增加测试最优点 B 相邻的两个点 C, D。以上述策略搜索完毕 1/2 像素运动矢量后, 以最小匹配误差点作为 1/4 像素精度的搜索中心, 1/4 像素为单位, 做一步小菱形搜索。

## 4 算法描述

根据上述研究和分析, 本文提出了一种分数像素快速运动估计算法。阈值  $TH2$  在 3.2.1 节中已定义,  $TH1$  等于  $TH2$ 。若不做特殊说明, 算法中的小菱形搜索均为 1/4 像素搜索。算法的主要步骤如下:

step1 计算最佳整像素点的匹配误差  $J$ 。如果当前块为预测块, 转到 step2; 如果  $J < TH2$ , 转到 step5, 否则转到 step4。

step2 计算  $MV_{pred}$  或  $MV_{up-layer}$  矢量点的  $J$ , 与最佳整像素点的  $J$  比较, 选择具有最小匹配误差  $MinJ$  的点。如果  $MinJ < TH1$ , 结束搜索, 转到 step6。否则, 以  $MinJ$  点为中心做一步小菱形搜索, 保存最优和次优点的位置, 若最优点是菱形的中心, 结束搜索, 转到 step6。

step3 以前一步小菱形搜索得到的最优点为新的菱形搜索的中心, 搜索新的小菱形中与前一步最优和次优点位置相同的点, 保存新的小菱形搜索中的最优和次优点的位置。如果最优点是新的小菱形的中心或者小菱形移动次数达到 2, 结束搜索, 转到 step6, 否则重复 step3。

step4 以小菱形搜索最佳整像素点周围的 4 个 1/2 像素点, 保存最优和次优点的位置和匹配误差。如果最优点为菱形的中心, 则停止 1/2 像素精度搜索。否则, 当最优点和次优点在同一直线上时, 增加搜索最优点相邻的两个点; 当最优点和次优点不在同一直线上, 增加搜索两点之间的那个点。

step5 以前一步得到的最优点为中心, 做一步小菱形搜索。

step6 当前最优点的运动矢量为最终的分数像素搜索的运动矢量, 分数像素搜索结束。

## 5 实验结果

把本文算法嵌入到 AVS\_M 校验模型 WM3.3a 中进行了测试。为了测试新算法在不同测试条件下的适用性, 选取 8 个具有不同运动特征的标准视频序列对算法性能进行了测试, 这些序列包括了不同的场景、不同的运动, 有的运动比较规则, 运动方式比较单一。有的比较剧烈, 包含各种运动方式。编码器主要参数包括: 搜索范围为 16, 参考帧为 1, 允许 RDO, Hadamard 变换开启, 帧率为 30, 量化参数为 28, 第一帧为 I 帧, 其余帧为 P 帧, 视频序列格式为 QCIF。实验中, 与分数像素全搜索算法和 CBFPS 算法在相同测试条件下进行了比较测试, 结果如表 2 所示。

表 2 本文算法与其他算法的比较

测试序列	算法	平均搜索 点数	PSNR (dB)	BitRate (kbit/s)
Akiyo	Full	17.00	40.83	69.60
	CBFPS	7.48	40.82	69.55
	Proposed	2.20	40.81	70.37
News	Full	17.00	39.22	130.51
	CBFPS	7.55	39.23	131.09
	Proposed	2.87	39.22	130.83
Silent	Full	17.00	38.46	145.05
	CBFPS	7.86	38.44	144.36
	Proposed	3.45	38.45	144.92
Foreman	Full	17.00	37.91	218.59
	CBFPS	9.32	37.88	218.22
	Proposed	4.77	37.91	220.53
Pairs	Full	17.00	36.92	326.46
	CBFPS	7.90	36.91	325.74
	Proposed	3.91	36.90	327.89
Bus	Full	17.00	36.68	605.95
	CBFPS	10.22	36.66	605.89
	Proposed	6.15	36.67	606.60
Tempete	Full	17.00	36.14	648.53
	CBFPS	8.77	36.12	648.69
	Proposed	4.96	36.11	650.29
Football	Full	17.00	36.48	968.72
	CBFPS	9.80	36.47	970.88
	Proposed	6.48	36.46	972.11

从表 2 可以看出, 与分数像素全搜索算法相比, 本文分数像素快速搜索算法减少了 61.88%~87.06% 的搜索点数, 而平均 PSNR 下降 0.01dB。与 CBFPS 算法相比, 本文算法减少了 33.88%~70.59% 的搜索点数, 平均 PSNR 基本相当。同时, 本文算法码率的变化也很微小。从表 2 中的 8 个测试序列的结果还可以看出, 本文算法具有很好的鲁棒性, 对不同运动特征的视频序列均可以获得很好的效果。

结束语 本文对 AVS\_M 的分数像素预测运动矢量进行了分析, 并提出了一种分数像素快速搜索算法。如果预测运动矢量的整像素部分等于整像素运动搜索得到的最优矢量, 那么利用预测的分数像素运动矢量作为搜索起点开始搜索。否则, 在预测矢量不够准确时采用精细搜索的策略。实验表明, 对于不同运动特征的序列, 本文算法都可以保持很好的信噪比和码率, 同时明显地降低了计算量。而且本文算法不需

(下转第 271 页)

一种是配置冗余,即配置多样化,从多种配置中选一种具有最大可存活性的配置。文献[16]讲述了通过依赖关系矩阵来选择,获得各种服务之间依赖关系的最大可存活性,从而确定服务之间的依赖关系。文献[17]通过设置服务复本的多少及位置,实现当服务受到攻击后,复本可以继续提供服务,而且可重新配置,使得原本存放服务 A 的复本空间根据需要存放服务 B 的复本。

另外一种是通过重新分配资源给关键任务来保证攻击成功后,系统仍然能够完成关键的服务。这种方法首先要确定哪些是关键任务,哪些为非关键任务。文献[18]提供了 ERAS 的方法,在系统受到攻击后,通过剥夺非关键任务的资源,将其重新分配给关键任务来实现系统的可存活性。文献[19]则通过 Cactus 方法,使得系统在运行时刻可以动态地改变自身的行为和配置来实现可存活性。

#### 4.3.3 实现方法的局限性

现有的方法无论是设计体系结构还是重新配置,都默认了一个前提:攻击是事先可知的,或者说是事先假定好的。系统的可存活能力只在某一种或者几种类型的攻击发生时有效,对于突然出现的(emerging)、事先不清楚的攻击无能为力。

此外,同分析方法类似,方法多而杂,且实用性不足。

**结束语** 近年来,针对系统的蓄意攻击给国家资源和基础设施(如医疗、银行等)带来的损失愈来愈大,系统的可存活性变得愈发重要。什么是可存活性的本质以及怎样描述可存活性,是可存活性研究领域两个最基本的问题。如何分析现有系统的可存活性,以及如何提高系统的可存活性成为当前研究的两大课题。

由于系统的可存活性至今尚无一个统一的定义,这两大课题的发展面临着巨大的挑战——方法种类繁多,但具有代表性的经典方法不多;研究重在理论,实用性不强;研究偏重某一方面,全局性不足。将来的研究应着手从多方面考虑系统可存活性的分析与实现,少些假设,多些实用。

### 参 考 文 献

[1] Kyamakya K. Security and Survivability of Distributed Systems: an Overview // IEEE MILCOM 2000. Los Angeles, California, 2000; 449-454

[2] Ellison R, Fisher D, Linger R, et al. Survivable Network Systems: An Emerging Discipline [R]. CMU/SEI-97-TR-013. 1997

[3] Westmark V R. A Definition of Information System Survivability // Proceedings of the 37th Hawaii International Conference on

System Sciences. Hawaii, 2004

[4] Lin Xuegang, Xu Rongsheng, Zhu Miao liang. Survivability Computation of Networked Information Systems. CIS, 2005, Part II; 407-414

[5] McDermott J. Attack-Potential-Based Survivability Modeling for High-Consequence Systems // Proceeding of Third IEEE International Workshop on Information Assurance. College Park, Maryland, 2005

[6] Fung C, et al. Survivability Analysis of Distributed Systems Using Attack Tree Methodology // Military Communications Conference. IEEE, 2005; 17-20

[7] Yi Xun, Zhang Yanchun. Survivability of Information System. Information, Communications and Signal Processing, 2005; 1551-1555

[8] Cloth L, Haverkort B R. Model Checking for Survivability. Quantitative Evaluation of Systems, 2005; 145-154

[9] Jha S, Wing J, Linger R, et al. Survivability Analysis of Network Specifications // International Conference on Dependable Systems and Networks. New York, USA, 2000

[10] Krings A W, Azadmanesh M H. A Graph Based Model for Survivability Analysis: Technical Report, UI-CS-TR-02-024. 2004

[11] Voas J M, Ghosh A K. Software Fault Injection for Survivability // DARPA Information Survivability Conference and Exposition. 2000; 338-346

[12] Harrison W S, Krings A W, Hanebutte N. On the Performance of a Survivability Architecture for Networked Computing Systems. System Sciences, 2002; 2534-2542

[13] Xiao Kun, Chen Niannen, Ren Shangping, et al. A Workflow-based Non-intrusive Approach for Enhancing the Survivability of Critical Infrastructures in Cyber Environment. Software Engineering for Secure Systems, 2007; 4-10

[14] Quek B K, Ibanez-Guzman J, Lim K W. A Survivability Framework for the Development of Autonomous Unmanned Systems. Control, Automation, Robotics and Vision, 2006; 1-6

[15] Quek B K, Ibanez-Guzman J, Lim K W. Attaining Operational Survivability in an Autonomous Unmanned Ground Surveillance Vehicle // IEEE Industrial Electronics, IECON. 2006; 3969-3974

[16] Lu Tun, Gu Ning. Survivability-Aware Configuration Management of Service-Oriented System Based on Service Dependency. Theoretical Aspects of Software Engineering, 2007; 421-432

[17] Wells D, Ford S, Langworthy D, et al. Software Survivability // DARPA Information Survivability Conference and Exposition. vol. 2, 2000; 241-255

[18] Wang Jian, Wang Huiqiang, Zhao Gaosheng. ERAS—An Emergency Response Algorithm for Survivability of Critical Services. Computer and Computational Sciences, 2006; 2; 97-100

[19] Hiltunen M A, Schlichting R D, Ugarte C A, et al. Survivability through Customization and Adaptability: The Cactus Approach // DARPA Information Survivability Conference and Exposition. vol. 1, 2000; 294-307

(上接第 252 页)

要最优整像素运动矢量周围点的匹配误差,可以与任意整像素搜索算法配合使用,更易于推广,具有很强的实用性。

### 参 考 文 献

[1] AVS 工作组. 信息技术——先进音视频编码,第七部分:移动视频(报批稿)[S]. 2006

[2] Tourapis A M, Au O C, Liou M L. Fast block matching motion estimation using predictive motion vector field adaptive search technique (PMVFAST). ISO/IEC JTC1/SC29/WG11 MPEG99/m5866, Noordwijkerhout, the Netherland, Mar. 2000

[3] Tourapis A M. Enhanced predictive zonal search for single and multiple frame motion estimation // Proceedings of Visual Communications and Image Processing 2002 (VCIP-2002). San Jose,

CA, January 2002; 1069-1079

[4] Zhou Z, Sun M T, Hsu Y F. Fast variable block-size motion estimation algorithms based on merge and split procedures for H. 264/MPEG-4 AVC [C] // IEEE International Symposium on Circuits and Systems. Vancouver, Canada, 2004; 725-728

[5] Chen Z B, Zhou P, He Y. Fast integer pel and fractional pel motion estimation for JVT[Z]. ftp://standards. po lycom. com/ imtc\_jvtexperts/2002\_12\_Awaji/JVT-F017. zip

[6] Chen Z B, Du C, Wang J H, et al. PPFPS: A paraboloid prediction based fractional pixel search strategy for H. 26L // Proc. of ISCAS 2002. May 2002; 9-12

[7] Yang Libo, Yu Keman, Li Jiang. Prediction-based directional fractional pixel motion estimation for H. 264 video coding [C] // IEEE International Conference on Acoustics, Speech and Signal Processing. Philadelphia, PA, 2005; 901-904