

# 基于 H. 264/AVC 视频标准的改进 UMHexagonS 算法

陈戈珩, 费东尼

(长春工业大学 计算机科学与工程学院, 长春 130012)

**摘要:** 基于 H. 264/AVC 视频编码标准, 通过分析非对称十字型多层次六边形格点搜索 (UMHexagonS) 算法的流程及存在的问题, 提出一种改进的分像素搜索算法, 并给出了改进的 UMHexagonS 搜索算法流程. 实验结果表明了改进算法的有效性, 在保证图像质量的同时, 显著减少了编码时间, 提高了编码效率.

**关键词:** 匹配; 估计; 编码; 搜索

**中图分类号:** TP309      **文献标志码:** A      **文章编号:** 1671-5489(2011)02-0283-06

## UMHexagonS Algorithm Based on H. 264/AVC Video Standard

CHEN Ge-heng, FEI Dong-ni

(College of Computer Science and Engineering, Changchun University of Technology, Changchun 130012, China)

**Abstract:** On the basis of H. 264/AVC video coding standard, an improved sub-pixel search algorithm was presented via the analysis of the current UMHexagonS algorithm processes and problems, optimization of the entire pixel search algorithm UMHexagonS with the improved UMHexagonS algorithm process given. Experimental results show the improved algorithm, on the premise of ensuring the quality of the images, significantly reduces the encoding time and improves the coding efficiency.

**Key words:** matching; estimation; coding; search

运动估计算法通常分为4类: 递归估计算法、光流法、贝叶斯估计算法和块匹配算法. 其中, 块匹配运动估计算法因其算法简单、便于VLSI实现等优点而得到广泛应用, 非对称十字型多层次六边形格点搜索 UMHexagonS (unsymmetrical-crossMulti-hexagon-grid search) 是一种常用的块匹配算法. 本文对该算法的初始搜索点及搜索窗<sup>[1]</sup>进行改进.

### 1 UMHexagonS 算法存在的问题

H. 264 视频标准中采用 UMHexagonS 整像素运动估计方法, 其算法流程可参见文献[2]. H. 264 视频标准进行整像素运动估计时, 首先依次进行中值预测区域搜索和原点预测区域搜索, 然后直接执行后续搜索处理, 如上层预测点区域搜索、网格搜索和钻石搜索等. 在后续搜索处理中, 包括对提前结束搜索条件的判断和由整像素运动估计过程得出的最优整像素点.

在获取最优整像素点后, 进行分像素的运动估计. 图1为AVC视频标准中采用全搜索的1/4像素运动估计示意图. 图1中大写字母(点A~I)代表整像素点位置, 其中A代表通过整像素运动估计搜

收稿日期: 2010-07-29.

作者简介: 陈戈珩(1961—), 女, 汉族, 硕士, 副教授, 从事数字信号处理的研究, E-mail: chengeheng@mail.ccut.edu.cn.

基金项目: 国家科技支撑计划项目基金(批准号: 2007BAQ00097).

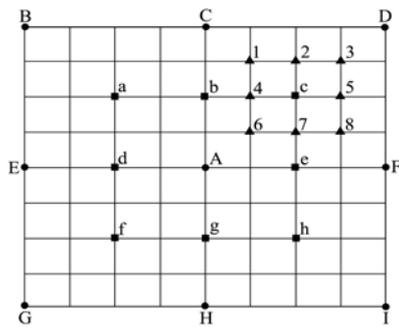


图1 分像素运动估计示意图

Fig.1 Points pixel motion estimation scheme

视频序列,由中值预测区域搜索或原点预测区域搜索得到的中值预测点或原点预测点与最优整像素点关联密切的情况占较大比例,而现有的整像素运动估计方法未考虑该因素,搜索完中值预测区域和原点预测区域后,直接执行后续搜索过程,导致不必要的搜索,使搜索点数过多,编码时间过长;并且上述分像素运动估计方法,未考虑不同宏块各自的运动特性,对不同运动特性的宏块按相同方式进行分像素搜索,也增加了编码时间。

## 2 改进 UMHexagonS 算法的流程及优点

针对 UMHexagonS 整像素搜索方法的缺陷,本文在进行整像素运动估计时,考虑到大量视频序列中值预测点或原点预测点与最优整像素点密切关联的特点,跳过 UMHexagonS 中不必要的搜索过程,并对原有 UMHexagonS 整像素搜索进行了优化,例如,优化了上层预测点区域搜索、相应块与相应帧搜索和非对称十字型搜索等,舍弃了  $5 \times 5$  的网关搜索;进行分像素搜索时,对不同运动特性宏块进行不同方式的搜索,在保证图像质量的同时,显著减少了编码时间<sup>[5-6]</sup>。改进的 UMHexagonS 算法流程如图2所示。

改进的 UMHexagonS 整像素搜索步骤如下<sup>[7]</sup>:

1) 进行中值预测区域搜索,获取当前最优匹配点。利用当前处理宏块的左、上、右3个相邻宏块的运动矢量中值作为当前块的预测值,获取中值预测点。进行中值预测时主要在中值预测点及其周围区域的像素点中进行搜索,获取匹配误差最小的点。通过绝对差分函数(sum of absolute difference, SAD)函数获取所述匹配误差。

2) 判断当前最优匹配点的匹配误差值是否满足提前终止条件,若满足,则跳过按 UMHexagonS 编码顺序后续的至少一步搜索过程,执行第一类型编码步骤;若不满足,执行步骤3)和4)。

提前终止条件包括提前终止条件2和条件1,根据量化参数QP和当前块的搜索模式确定提前终止条件1和条件2,其中,提前终止条件2的阈值小于提前终止条件1的阈值。

第一类型编码步骤如下:

① 若当前最优匹配点的匹配误差 SAD 值满足提前终止条件2,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行钻石搜索;

索出的最优整像素点;小写字母(点 a~h)代表  $1/2$  像素点位置;数字(1~8)代表  $1/4$  像素点位置。AVC 视频标准中采用的分像素搜索方法:首先搜索最优整像素点(点 A)及其周围  $3 \times 3$  范围内的8个  $1/2$  像素点(点 a~h),求出最优的  $1/2$  像素点(如点 c);然后,搜索最优的  $1/2$  像素点(点 c)周围的8个  $1/4$  像素点(点 1~8),得出最优的  $1/4$  像素点,将该最优  $1/4$  像素点作为运动估计的匹配最优点<sup>[3-4]</sup>。

事实上,UMHexagonS 的像素运动估计方法存在很多问题。如在整像素运动估计过程中,对大量

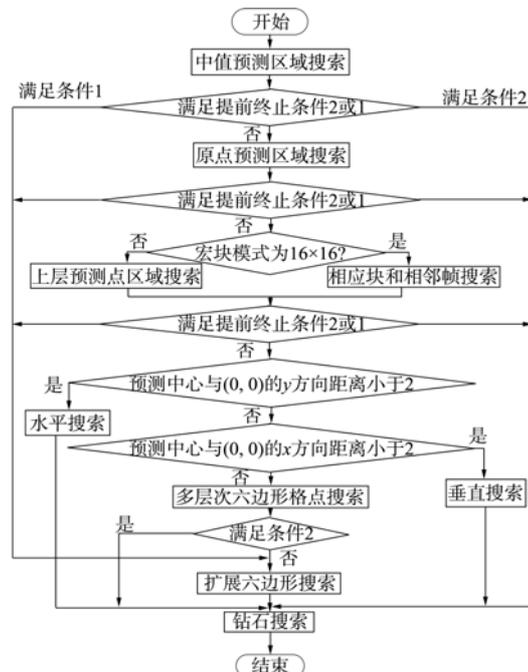


图2 改进的 UMHexagonS 算法流程

Fig.2 Improved UMHexagonS algorithm workflow

②若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 1,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行扩展六边形搜索和后续的钻石搜索。

3)进行原点预测区域搜索,获取当前最优匹配点。原点预测是对运动矢量为(0,0)处及其周围区域的像素点进行预搜索,从而获得当前最优匹配点。当前最优匹配点具有根据搜索过程不断更新的特点,当只进行中值预测时,当前最优匹配点为中值预测中搜索到的匹配误差最小的点;再进行原点预测区域搜索,这时获取的当前最优匹配点为中值预测和原点预测中搜索到的匹配误差最小的点,在搜索过程中当前最优匹配点会自动更新为当前已搜索过的所有像素点中匹配误差值最小的像素点。

4)判断当前最优匹配点的匹配误差值是否满足提前终止条件,若满足,则跳过按 UMHexagonS 编码顺序后续的至少一步搜索过程,执行第一类型编码步骤;若不满足,执行第二类型编码步骤。

在编码过程中,若该步骤得到的匹配误差值小于上一步得到的匹配误差值,则自动对上一步得到的匹配误差值进行更新,当原点预测中搜索所有像素点的匹配误差值大于中值预测获取的当前最优匹配点的匹配误差值时,则不需再对是否满足提前终止条件进行判断,这时,原点预测后的当前最优匹配点仍为中值预测的当前最优匹配点,不满足提前终止条件,可执行第二类型编码。

当在原点预测更新了当前最优匹配点时,判断当前最优匹配点的匹配误差 SAD 值是否满足提前终止条件 2 或 1。若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 2,则跳过 UMHexagonS 编码的中间搜索步骤,直接进行钻石搜索;若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 1,则跳过 UMHexagonS 编码的中间搜索步骤,直接进行扩展六边形搜索和后续的钻石搜索。当原点预测点的匹配误差 SAD 值不能满足提前终止条件 2 或 1 时,执行第二类型编码<sup>[8]</sup>。

第二类型编码步骤如下:

①判断当前宏块的宏块模式是否为  $16 \times 16$ ,若是,则进行相应块和相邻帧搜索,获取当前最优匹配点;若否,则进行上层预测点区域搜索,获取当前最优匹配点。H.264 视频标准中宏块分为  $16 \times 16$ ,  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$  四种模式,其中: $16 \times 16$  为  $16 \times 8$  和  $8 \times 16$  的上层块; $16 \times 8$  为  $8 \times 8$  的上层块。上层预测是用当前块的上层块运动矢量作为当前块的预测值,所以只有  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$  三种模式具有上层预测,对  $16 \times 16$  无需进行上层预测。

在步骤①中,首先对是否为  $16 \times 16$  的宏块模式进行判断,对  $16 \times 16$  模式的宏块,跳过上层预测区域搜索。

②根据当前最优匹配点的匹配误差值判断是否满足提前终止条件,若满足,则跳过按 UMHexagonS 编码顺序后续的至少一步搜索过程,执行第一类型编码步骤;若不满足,执行步骤③。

对于  $16 \times 16$  模式的宏块,根据相应块和相邻帧预测搜索获取当前最优匹配点的 SAD 值,判断是否满足提前终止条件 2 或 1。若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 2,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行钻石搜索;若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 1,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行扩展六边形搜索和后续的钻石搜索。

对于  $16 \times 8$ ,  $8 \times 16$ ,  $8 \times 8$  三种模式的宏块,根据上层预测搜索获取的当前最优匹配点的匹配误差 SAD 值判断是否满足提前终止条件 2 或 1。若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 2,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行钻石搜索;若当前最优匹配点的匹配误差 SAD 值满足提前终止条件 1,则跳过 UMHexagonS 编码的中间搜索步骤,直接执行扩展六边形搜索和后续的钻石搜索。

③判断获取的当前最优匹配点与参考帧中和当前宏块位置相同的宏块之间垂直方向距离是否小于预定像素点数,若小于预定像素点数,则在当前最佳匹配宏块所处的宏块行中进行搜索,再进行钻石搜索;若不小于预定像素点数,则执行步骤④。

通过步骤③对原有的 UMHexagonS 编码中非对称十字型搜索进行了改进,其中预定像素点数值选为 2<sup>[9]</sup>。当前最优匹配点与参考帧中和当前宏块位置相同的宏块(即运动矢量为(0,0)处)之间垂直方向(二维坐标的 y 方向)的像素个数小于 2 时,在参考帧与当前最优宏块所处的宏块行中进行搜索,然

后执行钻石搜索, 结束搜索过程, 其中, 当前最佳匹配宏块为当前最佳匹配点所属的宏块, 以下同理.

④ 判断获取的当前最优匹配点与参考帧中和当前宏块位置相同的宏块之间水平方向距离是否小于预定像素点数, 若小于预定像素点数, 则在当前最佳匹配宏块所处的宏块列中进行搜索, 再进行钻石搜索; 若不小于预定像素点数, 则进行多层次六边形格点搜索, 获取当前最优匹配点, 执行步骤⑤.

通过步骤④对原有的 UMHexagonS 编码中非对称十字型搜索进行了改进, 其中预定像素点数值选为 2. 当前最优匹配点与参考帧中和当前宏块位置相同的宏块(即运动矢量为(0,0)处)之间水平方向(二维坐标的  $x$  方向)的像素个数小于 2 时, 在参考帧与当前最优宏块所处的宏块列中进行搜索, 然后执行钻石搜索, 结束搜索过程.

⑤ 判断多层次六边形格点搜索获得的当前最优匹配点的匹配误差值是否满足提前终止条件 2, 若满足, 则进行钻石搜索; 若不满足, 则依次进行扩展六边形搜索和钻石搜索.

由于步骤⑤与搜索过程的最后步骤扩展六边形搜索和钻石搜索相邻, 所以无需再对是否满足提前终止条件 1 进行判断, 当多层次六边形格点搜索获得的当前最优匹配点的匹配误差 SAD 值满足提前终止条件 2 时, 进行钻石搜索; 否则, 依次进行扩展六边形搜索和钻石搜索, 结束搜索过程.

图 3 为改进的分像素搜索示意图. 由图 3 可见, 实施的方法舍弃了  $5 \times 5$  的网格搜索, 由上述步骤, 可得到最优的整像素匹配点.

如图 3(A)所示, 当运动矢量的水平方向运动矢量或垂直方向运动矢量取余于 4 的余数为 0 时, 只进行十字型方向的分像素搜索; 如图 3(B)所示, 当所述运动矢量的水平方向运动矢量或垂直方向运动矢量取余于 4 的余数都不为 0 时, 只进行 X 字型方向的分像素搜索.

例如, 若水平方向运动矢量或垂直方向运动矢量分别为  $\text{pred\_mv\_x}$ ,  $\text{pred\_mv\_y}$  时, 且  $x = \text{pred\_mv\_x} \% 4$ ,  $y = \text{pred\_mv\_y} \% 4$ , 则当  $x$  或  $y$  为 0 时, 只进行十字型方向的分像素搜索; 当  $x$  或  $y$  都不为 0 时, 只进行 X 字型方向的分像素搜索.

由于 AVC 的分像素搜索采用  $1/4$  精度, 所以采用了水平方向运动矢量或垂直方向运动矢量取余于 4 的方法<sup>[10]</sup>.

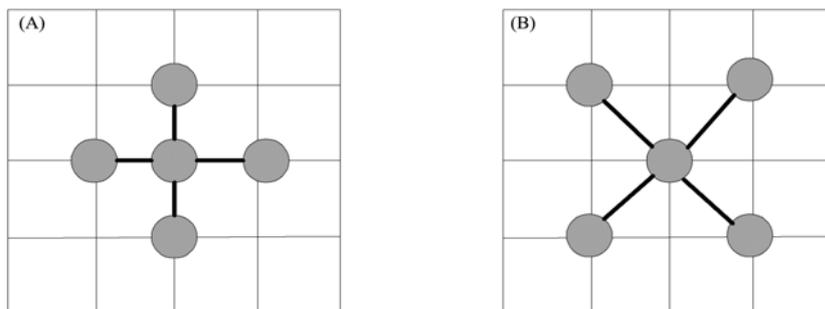


图 3 改进的分像素搜索示意图

Fig.3 Improved points pixel search scheme

### 3 改进算法的验证

通过将 PC 上运行的 H.264 代码 DSP 化, 在 DSP 上实现 H.264 的编解码算法. 为充分利用 DSP 的各种性能, 采用优化(代码的项目级优化、C 程序级优化和线性汇编程序级优化)的 X264 算法测试平台<sup>[11-12]</sup>, 对改进后的整像素 UMHexagonS 搜索算法结合改进的  $1/4$  像素算法进行验证. 实验参数设置如下:

- 1) 6 个典型不同运动特性和纹理特性的 cif 格式测试序列 (football, template, foreman, mobile, new, paris), 帧率 30 帧/s, 其中, football 和 template 编码 250 帧, 其他序列编码 300 帧;
- 2) 编码结构为 IPPP, 每秒插入两个 I 帧;
- 3) 参考图像为 2 帧;
- 4) 使用 Hadamard 变换;

- 5) 使用率失真优化;
- 6) 使用环路滤波;
- 7) 快速运动估计, 搜索范围为 32 个像素点;
- 8) 量化参数取 28, 32, 36 和 40.

表 1 列出了本文提出的改进算法和参考模型 X264 中算法编码性能的对比结果. 其中正数表示增加, 负数表示减少.

表 1 改进算法和参考模型编码的性能对比

Table 1 Coding performance comparison between improved algorithm and reference model

测试对象	量化参数	峰值信噪比/dB			码率变化/%	总编码时间 变化/%
		亮度分量 Y	色度分量 U	色度分量 V		
足球	28	-0.038 2	0.000 1	-0.016 2	0.568	-25.7
	32	-0.045 4	-0.009 5	-0.019 9	0.711	-27.2
	36	-0.047 2	0.005 1	-0.018 8	0.848	-28.5
	40	-0.059 9	-0.021 4	-0.030 7	0.768	-29.9
	平均值	-0.047 7	-0.006 4	-0.021 4	0.734	-27.8
模板	28	-0.045 8	0.004 2	-0.000 8	0.770	-32.8
	32	-0.049 3	0.002 7	0.000 5	0.709	-32.1
	36	-0.051 8	-0.000 7	-0.002 8	0.696	-33.7
	40	-0.051 8	-0.001 5	0.002 1	0.600	-35.3
	平均值	-0.049 7	0.001 2	-0.000 3	0.694	-33.5
领班	28	-0.053 4	-0.004 8	-0.012 8	0.432	-33.0
	32	-0.050 5	-0.007 9	-0.006 2	0.479	-33.9
	36	-0.049 8	-0.011 9	-0.006 2	0.313	-36.1
	40	-0.060 4	-0.011 2	-0.004 6	0.241	-36.8
	平均值	-0.052 5	-0.009 0	-0.007 5	0.366	-35.0
汽车	28	-0.038 0	-0.000 5	0.002 6	0.541	-31.9
	32	-0.041 3	-0.002 3	0.002 3	0.456	-32.3
	36	-0.045 0	-0.007 4	-0.003 2	0.284	-32.7
	40	-0.047 9	-0.000 9	0.001 3	0.189	-35.3
	平均值	-0.043 1	-0.002 8	0.000 8	0.367	-33.1
新闻	28	-0.021 9	-0.004 1	-0.004 1	0.371	-40.3
	32	-0.028 2	-0.006 1	-0.003 3	0.322	-41.1
	36	-0.024 8	-0.013 2	-0.010 0	0.246	-40.2
	40	-0.020 2	-0.004 8	-0.004 8	0.236	-41.4
	平均值	-0.023 8	-0.007 1	-0.005 6	0.294	-40.8
巴黎	28	-0.017 2	-0.002 9	-0.004 1	0.726	-35.8
	32	-0.018 9	0	0.000 9	0.713	-38.6
	36	-0.021 1	-0.002 6	-0.002 8	0.603	-41.1
	40	-0.016 6	-0.003 9	0.000 7	0.694	-39.2
	平均值	-0.018 5	-0.002 4	-0.001 3	0.684	-38.7
所有测试序列的平均值		-0.039 2	-0.004 4	-0.002 9	0.523	-34.8

由表 1 可见, 对不同运动特性和纹理特性的序列, 快速算法的性能不同. 对新闻和巴黎这种背景相对固定的序列, 快速模式选择的精度最高, 编码速度提高幅度最大, 低码率下达到了 41% 以上; 对汽车和模板这些纹理丰富并且前景和背景都有复杂运动的序列, 编码速度提高幅度稍小, 但也达到 32% 以上. 对于分别有整体剧烈运动和头部剧烈运动并且存在场景切换的足球和领班, 快速算法的亮度信噪比降低相对较明显.

根据上述实验数据可见, 改进后的算法在平均亮度信噪比降低了 0.04 dB, 平均色度信噪比几乎不变, 平均输出比特率增加 0.6% 的情况下, 编码速度平均提高了 34.8%.

综上所述,本文通过分析 UMHexagonS 算法的流程及存在的问题,提出一种改进的分像素搜索算法,并给出了改进的 UMHexagonS 算法流程.实验证明了改进算法的有效性,所提出的改进算法在保证图像质量的同时,显著减少了编码时间,提高了编码效率.

### 参 考 文 献

- [ 1 ] Sullivan G J, Topiwala P, Luthra A. The H. 264/AVC Advanced Video Coding Standard: Overview and Introduction to the Fidelity Range Extensions [J]. SPIE Conference on Applications of Digital Image Processing XXVII, 2004, 5558(1): 454-474.
- [ 2 ] DING Chao, CHEN Tao. Research on Motion Estimation Algorithm in X264 [J]. Applied Science and Technology, 2009, 36(11): 41-45. (丁超,陈涛. X264 的运动估计算法研究 [J]. 应用科学与技术, 2009, 36(11): 41-45.)
- [ 3 ] XIAO Min-lian. Comparative Research on Several Classical Rapid Algorithms of Block-Matching Motion Estimation [J]. Computer Knowledge and Technology, 2010, 6(32): 9152-9154. (肖敏连. 几种经典快速块匹配运动估计算法的比较研究 [J]. 电脑知识与技术, 2010, 6(32): 9152-9154.)
- [ 4 ] Mallat S, Hwang W L. Singularity Detection and Processing with Wavelets [J]. IEEE Trans Info Theory, 1992, 38(2): 617-643.
- [ 5 ] ITU Telecom. Standardization Sector of ITU, Advanced Video Coding for Generic Audiovisual Services [J]. ITU-T Recommendation H. 264, 2005(3): 87-92.
- [ 6 ] CHEN Zhen-xing, SONG Yang, Ikenaga T, et al. Adaptive Search Range Algorithms for Variable Block Size Motion Estimation in H. 264/AVC [J]. IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences, 2008, E91-A(4): 1015-1022.
- [ 7 ] WANG Li-li, HUANG Xiao-ge, GAN Tao. Adaptive Cross-Square Algorithm for Fastmotion Estimation [J]. Application Research of Computers, 2009, 26(2): 540-542. (王莉莉,黄晓革,甘涛. 一种自适应十字-方形运动估计快速算法 [J]. 计算机应用研究, 2009, 26(2): 540-542.)
- [ 8 ] YANG Yu-hong, XU Xuan, JI Xiao-yong. An Improvement on the Motion Estimation Algorithm: UMHexagonS [J]. Computer Engineering and Applications, 2006(11): 52-54. (杨育红,徐烜,季晓勇. 快速运动估计 UMHexagonS 算法的探讨与改进 [J]. 计算机工程与应用, 2006(11): 52-54.)
- [ 9 ] LI Ren-xing, ZENG Bing, Liou M L. A New Three-Step Search Algorithm for Block Motion Estimation [J]. IEEE Transactions on Circuits and Systems on Video Technology, 1994, 4(4): 438-442.
- [ 10 ] ZHU Shan, MA Kai-kuang. A New Diamond Search Algorithm for Fast Block Matching [J]. IEEE Transactions on Image Processing, 2000, 9(2): 287-290.
- [ 11 ] YANG Hong-sheng, YANG Guang, MAO Hong-yu, et al. H. 264/SVC Fast Motion Estimation Algorithm for Sample Video [J]. Journal of Jilin University: Engineering and Technology Edition, 2010, 40(6): 1710-1714. (杨红生,杨光,毛宏宇,等. 针对样品视频的 H. 264/SVC 的快速运动估计算法 [J]. 吉林大学学报:工学版, 2010, 40(6): 1710-1714.)
- [ 12 ] WU Xiao-lan, WANG Shi-gang, JI Teng-fei, et al. Video Object Extraction and Its DSP Realization Based on Multi-frame Difference [J]. Journal of Jilin University: Engineering and Technology Edition, 2008, 38(1): 219-223. (吴晓岚,王世刚,纪腾飞,等. 基于多帧间的差的视频对象提取方法及其在 DSP 上的实现 [J]. 吉林大学学报:工学版, 2008, 38(1): 219-223.)