

# 基于场景切变和内容特征检测的去隔行算法

朱小涛 李燕平 黄媛 黄倩

(河海大学计算机与信息学院 南京 211100)

**摘要** 基于对运动补偿去隔行算法的分析,提出了基于场景切变检测和内容特征检测的去隔行算法。该算法首先进行场景切变检测和视频内容特征检测,其次基于场景切变检测的结果进行优化的运动估计,然后对图像块进行局部区域划分,最后根据划分结果自适应选取不同的插值方式。实验结果表明,该算法不仅能够以较低的算法复杂度提升图像的垂直分辨率,而且对于不同视频内容的隔行视频序列,都能得到高质量的逐行序列。

**关键词** 场景切变,内容特征检测,去隔行

中图分类号 TP391 文献标识码 A DOI 10.11896/j.issn.1002-137X.2019.03.023

## Deinterlacing Algorithm Based on Scene Change and Content Characteristics Detection

ZHU Xiao-tao LI Yan-ping HUANG Yuan HUANG Qian

(College of Computer and Information, Hohai University, Nanjing 211100, China)

**Abstract** This paper proposed a deinterlacing algorithm based on scene change and content characteristics detection. Firstly, scene changes and video content characteristics were detected. Secondly, optimized motion estimation was performed based on scene change detection results. Thirdly, the image blocks were locally partitioned and different interpolation methods were applied. Experimental results show that the algorithm can not only improve the vertical image resolution with lower algorithm complexity, but also obtain high-quality progressive sequences for interlaced video sequences of different video content.

**Keywords** Scene change, Content characteristics detection, Deinterlacing

## 1 引言

受频带资源和刷新频率的限制,早期的模拟电视广播系统采用隔行扫描<sup>[1]</sup>格式进行传输和显示。数十年来,网络技术、多媒体技术和显示技术的发展催生了大量的视频去隔行研究。然而,随着新应用场景的不断出现,传统面向主观显示的研究路线已不能满足人们的需求,高性能、低成本的去隔行算法越来越受到关注。

目前主流的运动补偿去隔行算法<sup>[1-5]</sup>能更充分、准确地利用场内和场间的图像信息,实现更精确的插值,并弥补非运动补偿算法的边缘锯齿、运动模糊等缺点。但是运动补偿去隔行算法严重依赖运动估计过程中获得的最佳运动向量的精确度,并且对误差极其敏感,所以在硬件条件有限的情况下,复杂度较大的运动补偿去隔行算法无法广泛应用。

因此,现实应用中往往会将运动补偿去隔行算法与其他去隔行算法进行融合,或者结合去隔行算法与其他技术的优势,取长补短,从而提高算法实现的高效性和插值的准确性<sup>[6-12]</sup>。Gao等<sup>[7]</sup>将行复制、场合并和运动补偿相结合,得到

了较好的处理效果。Wang等<sup>[11]</sup>使用加权的最小二乘法误差准则将低分辨率的隔行图像转化为高分辨率的逐行图像,虽然矩阵运算等原因会增大算法的复杂度,但是主观和客观的对比结果均优于传统的去隔行算法。Jeon等<sup>[12]</sup>将模糊度量应用于双向滤波去隔行过程,不仅改善了图像的视觉效果,而且明显提高了测试序列的PSNR值。此外,不少算法从不同的角度提升了去隔行的性能<sup>[13-18]</sup>,但难以在高性能、高精度和低成本之间达到理想的平衡。文献<sup>[19]</sup>基于镜头检测提出了一种较好的解决方案,但对于存在较多静止区域的视频序列仍会产生冗余的计算量。

鉴于此,本文利用场景切变过程中的计算资源,增加了整体及局部区域的特征判断,提出了一种基于视频内容检测的自适应插值算法,该算法不仅能降低去隔行算法的复杂度,而且对于不同类型的隔行序列,都能得到清晰的逐行视频图像。

对于待插值的隔行视频序列,视频内容千变万化。因此,可以先对测试序列中的当前待插值场进行场景切变检测和视频内容特征检测,然后根据检测结果执行相应的去隔行步骤。传统的去隔行方法都采用基于整场的运动补偿插值方式,而

到稿日期:2018-07-16 返修日期:2018-09-19 本文受国家重点研发计划(2018YFC0407905),江苏省重点研发计划(BE2016904),国家自然科学基金(61502145,61300122),中央高校基本科研业务费专项资金(2017B42214)资助。

朱小涛(1991-),女,硕士,主要研究方向为视频处理,E-mail:1695669216@qq.com;李燕平(1993-),女,硕士生,CCF学生会会员,主要研究方向为视频处理、机器学习;黄媛(1994-),女,硕士生,CCF学生会会员,主要研究方向为视频处理、机器学习;黄倩(1981-),男,博士,副研究员,CCF高级会员,主要研究方向为多媒体计算、大数据分析、机器人教育等,E-mail:huangqian@hhu.edu.cn(通信作者)。

本文提出的算法利用场景切变检测的计算资源对视频内容进行局部区域特征检测,将当前场内的每个图像块划分为不同运动类型的区域,然后结合内容特征对不同区域采用不同的插值方式,从而降低了算法的整体复杂度。

## 2 去隔行过程的描述

### 2.1 算法执行流程的描述

算法的执行流程如图1所示。首先,将当前场的每一个图像块与前后场的对应图像块进行比较,得到绝对图像块误差  $AIBD$ ,如式(1)所示:

$$AIBD = \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} |F_n(i, j) - F_{n\pm 1}(i\pm 1, j)| \quad (1)$$

其中,  $F_n(i, j)$  为当前块的某个像素点的值;  $F_{n\pm 1}(i\pm 1, j)$  为前后场对应位置的像素值;  $M, N, i, j, n$  分别为运动估计块高度、运动估计块宽度、相对垂直坐标、相对水平坐标和场图像序号。

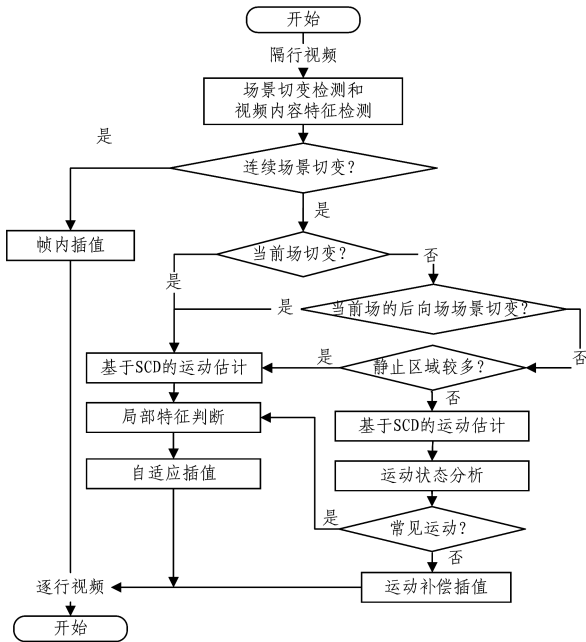


图1 算法执行流程

Fig. 1 Execution flowchart of proposed algorithm

接着,在文献[19]提出的镜头切变检测的基础上引入对图像块的局部区域类型判断。在得到场景切变的检测结果后,若当前场产生连续场景切变,则使用改进的帧内插值方式;否则,根据场景切变检测情况进行双向或者单向的运动估计,对于当前场发生场景切变、当前场的后向场发生场景切变、当前场没有发生场景切变且当前场与相邻场相比静止区域较多这3类场图像,待插值的像素与时空域相邻像素存在很大的相关性。因此,可以根据局部特征的判定结果(即当前待插值图像块是属于静止区域,或常见运动区域,还是复杂运动区域)进行自适应插值。但是对于没有产生场景切变的运动区域较多的场图像,最佳运动向量的获取是关键步骤,因此,在场景切变检测和整体视频内容特征判定之后,需要进行运动估计和运动状态分析。若当前场图像的常见运动区域较多,则仍可以结合局部区域特征判断进行分类插值;若复杂运动区域较多,则采用基于整场的运动补偿插值方式。

### 2.2 局部区域类型判断

在场景切变检测的过程中,基于式(1)定义的  $AIBD$  值判断视频内容特征和局部区域特征,而且这些操作都在插值之前进行,可以为插值算法的选取奠定基础。

局部区域特征的判断方式如式(2)和式(3)所示:

$$Local\_forward = \begin{cases} 1, & AIBD\_forward \geq BlocksInWidth \times \\ & BlockInHeight \times C_1 \\ -1, & AIBD\_forward \leq BlocksInWidth \times \\ & BlockInHeight \times C_2 \\ 0, & \text{其他} \end{cases} \quad (2)$$

$$Local\_backward = \begin{cases} 1, & AIBD\_backward \geq BlocksInWidth \times \\ & BlockInHeight \times C_1 \\ -1, & AIBD\_backward \leq BlocksInWidth \times \\ & BlockInHeight \times C_2 \\ 0, & \text{其他} \end{cases} \quad (3)$$

其中,  $Local\_forward$  和  $Local\_backward$  分别表示每个图像块与前、后场对应图像块的比较结果;  $BlocksInWidth$  和  $BlocksInHeight$  分别为行和列中运动估计图像块的数量;  $C_1 = 12$  和  $C_2 = 2$  为经验阈值。

当前图像块与前后场对应图像块的比较结果可以分为3种类型:  $Local\_forward = 1$  表示当前图像块与前向参考块不同;  $Local\_forward = -1$  表示当前图像块与前向参考块相似;  $Local\_forward = 0$  表示其余不确定情况。同理,  $Local\_backward$  取值为 1, -1, 0 时分别表示当前图像块与后向参考块不同、相似、不确定的情况。前向绝对图像块误差  $AIBD\_forward$  和后向绝对图像块误差  $AIBD\_backward$  的计算方法见式(1)。

在场景切变检测的过程中,首先计算每个图像块的  $AIBD\_forward$  和  $AIBD\_backward$ ,然后与经验阈值进行比较。前向绝对图像块误差  $AIBD\_forward$  和后向绝对图像块误差  $AIBD\_backward$  只需要计算一次,但是其可以用于统计与前后场图像不同检测类型的图像块数量,从而继续进行后续的场景切变判断以及整体视频内容特征判断,以简化局部静止区域、相对前向或后向静止区域的插值算法的执行步骤,最终降低整个自适应插值算法的复杂度。

上述对于局部图像块的判断中,不同、相似、不确定3种情况分别对应复杂运动区域、静止区域和常见运动区域,下同。

### 2.3 自适应插值过程

自适应的插值过程仍然是先对当前场进行场景切变检测,然后将检测结果分为4类:

(1) 若  $CF\_diff$  和  $CB\_diff$  都大于  $CC_3$ , 则当前场和后向场同时发生场景切变;

(2) 若  $CF\_same$  大于  $CC_1$  且  $CB\_diff$  大于  $CC_3$ , 则当前场的后向场发生场景切变;

(3) 若  $CB\_same$  大于  $CC_1$  且  $CF\_diff$  大于  $CC_3$ , 则当前场发生场景切变;

(4) 若为其余情况,则当前场没有发生场景切变。

对于当前场和后向场同时发生场景切变(即连续场景切

变)的情况,采用帧内插值方式。对于其他3类情况,首先结合场景切变的检测结果进行整场视频内容特征的判断和分类,然后再进行场内图像块的局部特征判断,最后进行分类插值。 $CF\_same$ 、 $CF\_diff$ 和 $CF\_uncertain$ 分别表示将当前场与前向参考场相比较得出的相似、不同和不确定3个类型的图像块数量。类似地, $CB\_same$ 、 $CB\_diff$ 和 $CB\_uncertain$ 分别表示将当前场与后向参考场,比较得出的相似、不同、不确定3个类型的图像块数量。下面分别介绍这3种类型的插值过程。

### 2.3.1 当前场没有发生场景切变

若当前场不存在场景切变,则当前场与相邻场存在时空相关性,插值可分为以下3种情况:

(1)若 $CF\_same > CC_1$ 且 $CB\_same > CC_1$ ,则当前待插值场的静止区域较多,在使用基于镜头切变检测的运动估计方式以后,无需进行运动状态分析,直接根据局部区域特征的判定结果,采用下文描述的双向时空中值滤波算法。当图像块被判断为静止区域时,进行时空域滤波插值;若与前向或者后向场相比被判断为运动区域,则进行单向或者双向运动补偿插值。

(2)若不满足情况(1),但 $\max\{CF\_same, CF\_diff, CF\_uncertain, CB\_same, CB\_diff, CB\_uncertain\} < CC_2$ ,则表示当前场与相邻场相比,可能因为场内图像的亮度信息跨度较小而导致场景切变检测结果判断错误或噪声等因素,使得局部区域特征的判定结果的可靠性不高,需要进行双向运动估计和运动状态分析,然后根据所求的最佳运动向量进行运动补偿插值(即无场景切变情况下的运动补偿插值算法<sup>[19]</sup>)。

(3)其余情况表示当前场图像没有明显的整场特征判断,此时根据运动估计和运动状态的分析结果,再结合每个图像块的局部特征自适应选择对应的插值方式。插值算法为下文描述的简化的混合时空中值滤波。

上文提到的 $CC_1$ 、 $CC_2$ 和 $CC_3$ 为经验阈值,本文中其取值分别为每幅图像中运动估计图像块数量的0.55、0.5和0.6倍。

### 2.3.2 后向场发生场景切变

若 $CF\_same > CC_1$ 且 $CB\_diff > CC_3$ ,则后向场发生场景切变,当前场与前向参考帧存在时空相关性。在运动补偿插值过程中,只利用当前场和前向参考帧的像素值作为参考信息。插值算法为下文描述的前向时空中值滤波。

### 2.3.3 当前场发生场景切变

若 $CB\_same > CC_1$ 且 $CF\_diff > CC_3$ ,当前场发生场景切变,则当前场与后向参考帧存在时间和空间相关性。则在运动补偿插值过程中,只利用当前场和后向参考帧的像素值作为参考信息。插值算法为下文描述的后向时空中值滤波。

## 2.4 自适应插值算法

下面介绍当前场没有发生场景切变、后向场发生场景切变和当前场发生场景切变这3种情况下用到的插值算法的具体实现细节。

### 2.4.1 双向时空中值滤波

(1)若当前图像块为边界区域,插值方法如式(4)所示:

$$F(i, j) = \text{median} \left( \frac{F_n(i-1, j+offset) + F_n(i+1, j-offset)}{2}, \right.$$

$$\left. \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, A, B, \frac{A+B}{2} \right) \quad (4)$$

其中, $A = F_{n-1}(i+mv_y, j+mv_x)$ 和 $B = F_{n-1}(i-mv_y, j-mv_x)$ 分别为前向和后向的参考像素; $offset$ 用于确定边缘方向。

(2)若 $Local\_forward = Local\_backward = -1$ ,则表示当前图像块与前后向参考场对应位置的图像块相似,即为静止区域,只使用前向后参考帧中对应位置的像素值以及当前场中待插值像素点的相邻位置的像素值作为中值滤波的候选值,具体插值方式如式(5)所示:

$$F(i, j) = \text{median} \left( \frac{F_n(i-1, j+offset) + F_n(i+1, j-offset)}{2}, \right.$$

$$\left. \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, F_n(i-1, j), F_n(i+1, j), \right.$$

$$\left. F_{n-1}(i, j), F_{n+1}(i, j) \right) \quad (5)$$

(3)若 $Local\_forward = -1$ 且 $Local\_backward = 1$ ,表示当前图像块与前向参考场对应位置的图像块相似,则只使用前向参考帧对应位置的像素值、运动向量补偿得到的前向参考位置的像素值以及当前待插值场已知相邻像素点的像素值作为中值滤波的候选值,具体插值方式如式(6)所示:

$$F(i, j) = \text{median} \left( A, A, F_{n-1}(i, j), \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, \right.$$

$$\left. F_n(i-1, j), F_n(i+1, j) \right) \quad (6)$$

(4)若 $Local\_forward = 1$ 且 $Local\_backward = -1$ ,类似可得式(7)所示的插值方式:

$$F(i, j) = \text{median} \left( B, B, F_{n+1}(i, j), \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, \right.$$

$$\left. F_n(i-1, j), F_n(i+1, j) \right) \quad (7)$$

(5)其他情况,将运动向量补偿得到的前后向参考位置的像素以及当前场的上、下相邻位置的像素作为中值滤波的候选值,具体插值方式如式(8)所示:

$$F(i, j) = \text{median} \left( A, B, \frac{A+B}{2}, \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, \right.$$

$$\left. F_n(i-1, j), F_n(i+1, j) \right) \quad (8)$$

### 2.4.2 前向时空中值滤波

由于是当前场的后向场发生了场景切变,因此在运动估计以及插值过程中不考虑后向临时完整帧的参考信息,故自适应插值方式可以分为以下3类:

(1)若当前图像块为边界区域,则插值方法与式(4)类似,将公式中的 $B$ 替换为 $A$ 即可。

(2)若 $Local\_forward = -1$ ,表示当前图像块与前向参考场对应位置的图像块相似,即为静止区域,则只使用前向参考帧中对应位置的像素值以及当前待插值场已知的相邻像素点的像素值作为中值滤波的候选值,具体插值方式与式(5)类似,去掉后向参考帧对应位置的像素值即可。

(3)其他情况,利用运动向量补偿得到的前向参考位置的像素以及当前场的上、下相邻位置的像素值作为中值滤波的候选值,具体插值方式与式(8)类似,只需将式(8)中的 $B$ 替换为 $A$ 即可。

### 2.4.3 后向时空中值滤波

因为是当前场发生场景切变,所以在运动估计以及插值过程中不考虑前向已插值帧的参考信息,故自适应插值方式可以分为以下3类:

(1)若当前图像块为边界区域,则插值方法与式(4)类似,将式(4)中的  $A$  替换为  $B$  即可。

(2)若  $Local\_forward = -1$ ,表示当前图像块与后向参考场对应位置的图像块相似,即为静止区域,则只使用后向临时完整帧中对应位置的像素值以及当前待插值场已知的相邻像素点的像素值作为中值滤波的候选值,具体插值方式与式(5)类似,去掉前向参考帧对应位置的像素值即可。

(3)其他情况,利用运动向量补偿得到的后向参考位置的像素值以及当前场的上、下相邻位置的像素值作为中值滤波的候选值,具体插值方式与式(8)类似,将式(8)中的  $A$  替换为  $B$  即可。

### 2.4.4 简化的混合时空中值滤波

(1)若当前图像块为边界区域,插值方法同式(4)。

(2)若  $Local\_forward = Local\_backward = -1$ ,表示当前图像块与前后向参考场对应位置的图像块相似,即为静止区域,则只使用前向参考帧中对应位置的像素值、当前待插值场已知相邻像素点的像素值和运动向量补偿得到的前后向参考位置的像素值作为中值滤波的候选值,具体插值方式如式(9)所示:

$$F(i, j) = \text{median}(F_n(i-1, j), F_n(i+1, j), \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, A, B, \frac{F_{n-1}(i, j) + F_{n+1}(i, j)}{2}, F_{n-1}(i, j), F_{n+1}(i, j)) \quad (9)$$

(3)若  $Local\_forward = -1$  且  $Local\_backward = 1$ ,与式(6)类似,去掉一个候选值  $A$  即可。

(4)若  $Local\_forward = 1$  且  $Local\_backward = -1$ ,与式(7)类似,去掉一个候选值  $B$  即可。

(5)其他情况,利用运动向量补偿得到的前后向参考位置的像素值以及当前场的上、下相邻位置的像素值作为中值滤波的候选值,若为运动一致性区域,则使用式(10)进行计算,否则使用式(8)进行计算。

$$F(i, j) = \text{median}(A, B, \frac{F_n(i-1, j) + F_n(i+1, j)}{2}, F_n(i-1, j), F_n(i+1, j)) \quad (10)$$

## 3 实验结果

### 3.1 测试序列

本文提出的算法主要针对不同视频内容的自适应插值算法,因此选择了频繁发生场景切变的测试序列、视频相邻帧相似但帧内亮度跨度不大的测试序列、局部区域缓慢运动的测试、变化不确定的测试序列(相邻帧局部区域发生旋转等复杂运动的测试序列)。用于测试的隔行视频都是由逐行视频采样得到的,然后将去隔行后生成的逐行视频与原始的逐行视频进行比较,从而得出主观评价和客观评价。BG\_16336, BG\_16337, BG\_16343, BG\_36501, BG\_37309 是由 TRECVID 2007 测

试视频去除前几帧不清晰的序列得到的。而 Comb288i, Comb480i, Comb576i, Comb720i 是由相同分辨率的不同视频取连续的9帧或10帧拼接而成的。在局部区域的检测过程中,选用宽和高都为8的图像块进行对比和判断。

### 3.2 客观评价

从表1可以看出,对于不同的测试序列,相比于文献[19]提出的基于场景切变检测的去隔行算法,本文算法在大多数情况下都能得到更好的PSNR比对结果;此外,算法在执行过程中利用了场景切变检测的计算资源来进行视频特征的判断,从而简化了静止区域的插值过程,降低了整个算法的执行复杂度,因此该算法相较于文献[19]报道的算法,执行时间明显缩短。

表1 PSNR和执行时间的比较

Table 1 Comparison of PSNR and execution time

测试视频	测试帧数	PSNR/dB		执行时间/s	
		文献[19]算法	本文算法	文献[19]算法	本文算法
BG_16336	300	33.66	33.87	32.68	30.31
BG_16337	300	39.64	39.76	31.56	29.20
BG_16343	300	34.83	35.00	32.80	31.40
BG_36501	300	31.16	31.03	31.38	28.77
BG_37309	300	38.66	39.37	32.15	30.48
Comb288i	300	35.70	35.99	38.51	37.62
Comb576i	90	35.19	35.28	46.85	44.19
Comb480i	50	45.82	45.92	23.43	21.49
Comb720i	100	36.05	35.97	115.11	113.32
平均值	/	36.75	36.91	42.72	40.75

表2列出了本文提出的自适应去隔行算法与组成该自适应算法各成分算法的对比结果。从表2可以看出,本文提出的自适应插值算法优于各成分算法,尤其是对于静止区域较多的测试序列,客观PSNR值的提升非常明显。

表2 本文自适应去隔行算法与其成分算法的PSNR比较

Table 2 Comparison of PSNR among proposed adaptive algorithm and its components

测试视频	测试帧数	PSNR/dB				
		帧内插值	文献[19]算法	时空线性滤波	文献[20]算法	本文算法
BG_16336	300	32.01	33.66	33.21	29.96	33.87
BG_16337	300	36.38	39.64	39.41	31.25	39.76
BG_16343	300	29.08	34.83	33.35	25.71	35.00
BG_36501	300	30.07	31.16	27.07	25.35	31.03
BG_37309	300	28.11	38.66	36.23	26.10	39.37
Comb288i	300	31.17	35.70	32.81	27.67	35.99
Comb576i	90	31.53	35.19	31.53	30.25	35.28
Comb480i	50	35.33	45.82	33.53	40.29	45.92
Comb720i	100	35.94	36.05	33.01	31.62	35.97
平均值	/	32.19	36.75	33.37	29.8	36.91

### 3.3 主观评价

图2给出了本文提出的自适应去隔行算法的主观效果。对于不同类型的隔行序列,本文算法能够根据各个视频内容的特征自适应地选择最合适的插值方式,从而获得清晰的主观视觉效果。

图2(a)、图2(b)与图2(d)相比于前后相邻的图像,大多数图像块处于静止区域,因此经整体视频内容的判定后,以时空域滤波为主进行插值。图2(c)和图2(e)的整体亮度偏低,图2(c)的部分区域还存在复杂运动,若使用基于像素的场景

切变检测算法,检测结果的可信度不高,因此主要采用双向运动补偿的插值方式。图2(f)中常见运动区域较多,采用混合时空域滤波与运动补偿相结合的方式,通过局部内容特征判断来选择最优的插值算法,以保证插值的准确度。



图2 去隔行帧图像的主观评价

Fig. 2 Subjective evaluation for deinterlaced frames

**结束语** 本文算法在基于场景切变检测的去隔行算法的基础上,加入了整体和局部视频内容检测模块,能够根据视频内容特征进行分类插值。所提算法不仅能够提高去隔行后视频序列的清晰度,还能利用场景切变过程中的计算资源降低插值算法的复杂度。但是,场图像中存在复杂运动的整体以及局部特征判断方式还存在一些不足,实验中会出现运动特征错判的情况,从而导致部分插值准确度下降。下一步将致力于优化运动区域的判断准则以降低去隔行算法的复杂度,并进一步提升插值的准确度。

## 参考文献

- [1] YANG L, XU W P. A novel adaptive de-interlacing algorithm based on motion compensation [J]. *Microelectronics & Computer*, 2002, 19(9): 4-6. (in Chinese)  
杨柳,徐维朴.一种新型基于运动补偿的自适应去隔行算法[J]. *微电子学与计算机*, 2002, 19(9): 4-6.
- [2] HUANG Q, GAO W, ZHAO D, et al. An Efficient and Robust Adaptive Deinterlacing Technique [J]. *IEEE Transactions on Consumer Electronics*, 2006, 52(3): 888-895.
- [3] MOHAMMADI H M, LANGLOIS P, SAVARIA Y. A Five-field Motion Compensated Deinterlacing Method Based on Vertical Motion [J]. *IEEE Transactions on Consumer Electronics*, 2007, 53(3): 1117-1124.
- [4] HUANG Q, ZHAO D, MA S, et al. Deinterlacing Using Hierarchical Motion Analysis [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010, 20(5): 673-686.
- [5] MOHAMMADI H M, SAVARIA Y, LANGLOIS J M P. Enhanced Motion Compensated Deinterlacing Algorithm [J]. *IET Image Processing*, 2013, 6(8): 1041-1048.
- [6] LEE G G, LIN H Y, WANG M J, et al. A High-Quality Spatial-temporal Content-adaptive Deinterlacing Algorithm [C] // *Proceedings of IEEE International Symposium on Circuits and Systems*. New York: IEEE Press, 2008: 2594-2597.
- [7] GAO X, GU J, LI J. De-interlacing Algorithms Based on Motion Compensation [J]. *IEEE Transactions on Consumer Electronics*, 2005, 51(2): 589-599.
- [8] PARK S J, JEON G, JEONG J. Deinterlacing Algorithm Using Edge Direction From Analysis of the DCT Coefficient Distribution [J]. *IEEE Transactions on Consumer Electronics*, 2009, 55(3): 1674-1681.
- [9] CHO H J, LEE Y S, OH S H, et al. A Voting-based Intra Deinterlacing Method for Directional Error Correction [J]. *IEEE Transactions on Consumer Electronics*, 2010, 56(3): 1713-1721.
- [10] ZHANG H, WANG R, LIU W, et al. Fusion-Based Edge-sensitive Interpolation Method for Deinterlacing [J]. *Multimedia Tools and Applications*, 2015, 74(18): 7643-7659.
- [11] WANG J, JEON G, JEONG J. De-Interlacing Algorithm Using Weighted Least Squares [J]. *IEEE Transactions on Circuits and Systems for Video Technology*, 2014, 24(1): 39-48.
- [12] JEON G, KANG S H, LEE J K. A Robust Fuzzy-bilateral Filtering Method and Its Application to Video Deinterlacing [J]. *Journal of Real-Time Image Processing*, 2016, 11(1): 223-233.
- [13] KWON O, SOHN K, LEE C. Deinterlacing Using Directional Interpolation and Motion Compensation [J]. *IEEE Transactions on Consumer Electronics*, 2003, 49(1): 198-203.
- [14] LIN S F, CHANG Y L, CHEN L G. Motion adaptive de-interlacing by horizontal motion detection and enhanced ELA processing [C] // *Proceedings of IEEE International Symposium on Circuits and Systems*. New York: IEEE Press, 2003: II-696-II-699.
- [15] ZHANG X, WANG D. An Efficient Motion Adaptive Deinterlacing Algorithm Using Improved Edge-based Line Average Interpolation [C] // *Proceedings of IEEE International Conference on Signal Processing*. New York: IEEE Press, 2016: 659-662.
- [16] WANG J, WU Z, WU J. Efficient Adaptive Deinterlacing Algorithm Using Bilateral Filter [C] // *MATEC Web of Conferences*. Les Ulis Cedex A: EDP Sciences, 2016.
- [17] LEE K, LEE C. High Quality Deinterlacing Using Content Adaptive Vertical Temporal Filtering [J]. *IEEE Transactions on Consumer Electronics*, 2010, 56(4): 2469-2474.
- [18] WANG A, JEONG J. Deinterlacing Using Multi Direction Detection with Fixed Adaptive Tap Interpolation Filter [C] // *Proceedings of IEEE International Symposium on Broadband Multimedia Systems and Broadcasting*. New York: IEEE Press, 2012: 1-5.
- [19] ZHU X, HUANG Q, YE F, et al. Motion-compensated deinterlacing based on scene change detection [C] // *Proceedings of Pacific-Rim Conference on Multimedia*. Berlin: Springer-Verlag, 2017: 397-406.
- [20] JANG S M, PARK J H, HONG S H. Deinterlacing method based on edge direction refinement using weighted median filter [C] // *Proceedings of IEEE International Conference on Signal and Image Processing Applications*. New York: IEEE Press, 2009: 227-230.