

中图法分类号: TP391 文献标志码: A 文章编号: 1006-8961(2011)02-0191-06

论文索引信息: 邓慧萍, 喻莉, 钟刚, 王超, 熊玮. 基于运动矢量处理的帧率上转换算法 [J]. 中国图象图形学报, 2011, 16(2): 191-196

## 基于运动矢量处理的帧率上转换算法

邓慧萍, 喻莉, 钟刚, 王超, 熊玮

(华中科技大学电子与信息工程系, 武汉光电国家实验室光通信与智能网络研究部, 武汉 430074)

**摘要:** 帧速率上转换主要是用来实现两个不同帧率视频扫描格式之间的转换。它可以应用于 HDTV 和低比特码率编码等应用场合。将场景检测引入到帧率上转换方法中, 根据视频内容的变化来检测场景的变换, 然后自适应地选择内插(Interpolation)/外推(Extrapolation)的方法, 从而解决了传统运动补偿内插法针对序列剧烈运动或场景切换难以处理的问题。同时, 结合残差能量的大小和运动矢量的相关性对运动矢量的可靠性进行分类, 对不可靠运动矢量采用可变块大小运动估计进行重新搜索, 不断更新运动矢量, 逐步消除由于运动矢量的不准确带来的方块效应和边缘模糊现象。实验结果表明, 算法较传统的插值算法对内插图像的主观质量和客观质量都有所提高。

**关键词:** 帧率上转换; 运动补偿内插; 场景检测; 运动矢量处理

### Novel motion vector processing method for frame rate up conversion

Deng Huiping, Yu Li, Zhong Gang, Wang Chao, Xiong Wei

(*Electronics & Information Engineering Department, Huazhong University of Science and Technology,  
Wuhan National Laboratory for Optoelectronics, Wuhan 430074 China*)

**Abstract:** Frame rate up conversion (FRUC) is a conversion process between any two display formats with different frame rates. It can be used in HDTV application and in low bandwidth video coding. In this paper, a scene change detection method is proposed to detect whether the scene has changed. If the scene change occurs, the extrapolation technique will be used, otherwise, the interpolation will be chosen. Second, a motion vector processing method is adopted to correct and refine the unreliable motion vectors with variable block sizes, so the visual artifacts due to incorrect motion vectors can be eliminated and the object boundaries and structure information can be preserved. Experimental results show that the proposed algorithm provides good subjective and objective quality.

**Keywords:** frame rate up conversion; motion compensation frame interpolation; scene change detection; motion vector processing

## 0 引言

帧率上转换 (FRUC) 也叫帧率提升, 主要是用来实现两个不同帧率视频扫描格式之间的转换。FRUC 的一个应用场合是高清电视 (HDTV)。HDTV

的分辨率最高可以达到  $1920 \times 1080$ , 帧率可以达到 60 帧/s。而目前我国和欧洲电视所使用的 PAL (phase alternating line) 制帧率仅为 25 帧/s, 北美等国使用的 NTSC (national television system committee) 制为 30 帧/s。因此, 要将现有制式转换为 HDTV, 必须引入帧率上转换机制<sup>[1]</sup>。此外, FRUC 也可以

收稿日期: 2009-08-31; 修回日期: 2009-09-27

基金项目: 国家自然科学基金 (60872010, 60972016, 60903172); 国家高技术研究发展计划 (863) 项目 (2009AA01Z205); 新世纪优秀人才支持计划 (NCE070339); 国家实验室基金 (P080010); 湖北省杰出青年基金 (2009CDA150); 湖北省自然科学基金 (2008CDB329); 中芬国际合作项目 (2010DFB10570)。

第一作者简介: 邓慧萍 (1983—), 女。华中科技大学通信与信息系统专业博士研究生, 主要研究方向为视频编码技术。  
E-mail: pingp1223@163.com。

应用于低比特率视频编码中。它的思想是在编码端适当降低原始视频的帧率,然后在解码端引入 FRUC 机制让帧率提升到原始帧率大小。

运动补偿内插法(MCFI)是 FRUC 中一种效果较好,应用比较广泛的算法<sup>[2]</sup>。该算法利用相邻帧之间的相关性,假定中间帧的运动矢量是前后两帧运动矢量的一半,从而内插出中间帧来。但是,当视频序列剧烈运动或者存在场景切换时,相邻帧的相关性下降,这种内插的方法就会失效。同时,由于 MCFI 是一种基于块的算法,那么当某一分割块在物体边缘并且包含几个运动对象时,该块的运动矢量就不准确,从而导致内插出来的图像边缘模糊<sup>[3]</sup>。而且在遮挡区域和暴露区域内像素点找不到与之相匹配的像素,那么运动估计获得的运动矢量就不正确,这样的区域用 MCFI 方法就会内插出错误块,也就造成了方块效应<sup>[4]</sup>。

针对上述 MCFI 存在的问题分别提出相应的解决方法。首先通过场景检测来判断视频序列是否有场景变换发生。如果有场景变换产生,那么采用外推的方法来预测当前帧,利用前面两个相邻帧或者后面两个相邻帧来外推得到当前帧。如果序列没有发生场景变换,则采用内插的方法来预测当前帧,即利用一前一后两个相邻帧内插出当前帧。接着,对于初始运动矢量场(MVF),根据残差能量和运动矢量(MV)的相关性对其可靠性进行分类,然后对不可靠的运动矢量重新搜索,逐级校验并找到对应的遮挡区域和暴露区域。最后,采用自适应的插值滤波器进行插值获得待插帧。

## 1 本文算法

### 1.1 整体结构框图

图 1 是算法的整体结构框图。整个算法包括 4 个主要模块,分别是场景检测、对称运动估计、运动矢量处理和自适应插值滤波器。

对于输入的原始视频序列,首先利用场景检测功能来检测当前是否有场景变换发生。如果检测到当前场景发生变换,那么就采用外推的方法;若没有检测到场景变换,就采用内插的方法。然后通过双向运动估计,得到一个初始运动矢量场。尽管这个初始的 MVF 能够大致反映物体的运动轨迹,但是在物体的边缘区域、遮挡区域以及“暴露”区域,MV 有可能并不可靠,对不可靠的 MV 进行校验更正等处

理进行校正得到比较可靠的 MV。最后,选用自适应的插值滤波器得到待插帧输出。

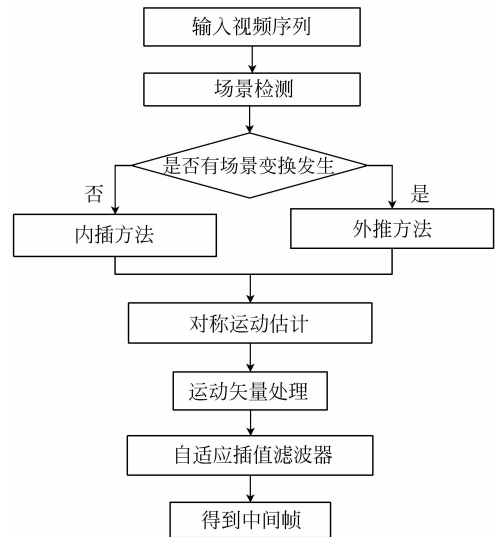


图 1 算法结构框图

Fig. 1 Block diagram for the proposed FRUC scheme

下面详细介绍这 4 个模块的实现方法。

### 1.2 整体结构框图

一般来说,序列的相邻帧之间存在着很大的相关性。如图 2 所示, $f_{n-3}$  帧和  $f_{n-1}$  帧之间, $f_{n+1}$  帧和  $f_{n+3}$  帧之间都存在很大的相关性。但是当出现镜头切换的情况时,处于切换点的两帧通常会存在极大的差异,比如  $f_{n-1}$  帧和  $f_{n+1}$  帧。此时如果仍然采用内插的方法用  $f_{n-1}$  帧和  $f_{n+1}$  帧对  $f_n$  帧进行预测,则无法得到有效的插帧。这种情况下若使用外推法,即使用  $f_{n-3}$  帧和  $f_{n-1}$  帧或者  $f_{n+1}$  帧和  $f_{n+3}$  帧来外推帧,往往能得到更好的效果。

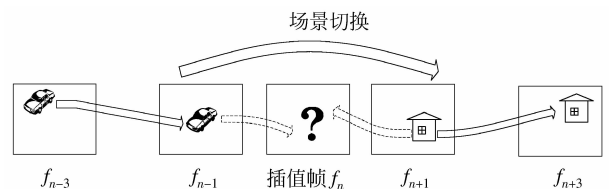


图 2 场景检测示意图

Fig. 2 Diagram of scene change detection

采用帧差法来检测场景的变换,通过统计 SAD (sum of absolute difference) 超过某一给定阈值的块的个数来判断是否有场景变换产生<sup>[5]</sup>。

对于输入序列的每两个相邻帧,按下式计算对应宏块的 SAD 值

$$\text{SAD}[\mathbf{B}_{ij}] = \sum_{s \in \mathbf{B}_{ij}} |f_{n-1}[s] - f_{n+1}[s]| \quad (1)$$

式中  $f_{n-1}, f_{n+1}$  是两个相邻帧,  $\mathbf{B}_{ij}$  是帧内的一个宏块。

统计 SAD 大于一定阈值的宏块数目  $\text{SAD\_num}$ 。

假设  $f_{n-3}$  与  $f_{n-1}$  之间 SAD 大于阈值的宏块数目为  $\text{SAD\_num}_0$ ,  $f_{n-1}$  与  $f_{n+1}$  之间为  $\text{SAD\_num}_1$ ,  $f_{n+1}$  与  $f_{n+3}$  之间为  $\text{SAD\_num}_2$ 。

如果  $\text{SAD\_num}_1 < n$  (实验中取  $n$  为整幅图像宏块数目的 1/2), 那么说明相邻帧的相关性很大, 于是采用内插的方法。如果  $\text{SAD\_num}_1 > n$ , 那么判定这两个相邻帧之间存在场景的切换。此时采用外推的方法对中间帧  $f_n$  进行估计。

外推时, 如果满足

$$\begin{cases} \text{SAD\_num}_0 < n \\ \text{SAD\_num}_1 > n \\ \text{SAD\_num}_2 > n \end{cases} \quad (2)$$

那么就利用前面两帧  $f_{n-3}, f_{n-1}$  外推当前帧  $f_n$ 。

如果满足下式

$$\begin{cases} \text{SAD\_num}_0 > n \\ \text{SAD\_num}_1 > n \\ \text{SAD\_num}_2 < n \end{cases} \quad (3)$$

那么就利用后面两帧  $f_{n+1}, f_{n+3}$  外推当前帧  $f_n$ 。

如果  $\text{SAD\_num}_0, \text{SAD\_num}_1, \text{SAD\_num}_2$  都大于  $n$ , 那么说明存在镜头连续切换, 这时不管是内插还是外推都不能得到很好的效果, 针对这种情况, 采用跳帧的方法。这样既保证了每幅图像的质量, 而且对于整段视频来说, 画面仍然非常流畅。

### 1.3 对称运动估计

帧率转换中通常采用传统的运动估计, 该方法对两个已知相邻帧, 以其中一帧图像的一个块为基准, 在另一帧图像中搜索最佳匹配块, 从而得到运动矢量(MV), 然后取  $\text{MV}/2$  作为中间帧的运动矢量。这种方法通常会导致了重叠和漏洞现象的出现<sup>[6]</sup>。

对称运动估计能很巧妙地解决上述问题<sup>[7]</sup>。它是以内插帧中待插块的坐标为基准, 然后在两个参考帧中的一定搜索范围内对称的移动对应匹配块的位置, 按照式(4)的匹配准则, 选出差别最小的那对块作为最佳匹配块。图3是对称运动估计的示意图。这种对称的运动估计使得待插帧中每个块是“无缝”拼接的, 因此很好地解决了直接运动估计带来的漏洞和重叠问题。

$$\text{SAD\_inter}[\mathbf{B}_{ij}] = \sum_{s \in \mathbf{B}_{ij}} |f_{n-1}[s - \mathbf{MV}] - f_{n+1}[s + \mathbf{MV}]| \quad (4)$$

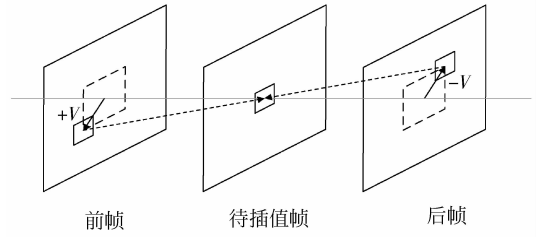


图3 对称运动估计框图

Fig.3 Diagram of bilateral motion estimation

但是, 依照式(4)的匹配准则得到的 MV 是视频编码中为了提高编码效率, 使残差最小时的 MV, 并不是真正意义上的运动轨迹。

在 FRUC 中为了使 MV 更接近真实的运动轨迹, 同时考虑帧内的相关性, 修正了运动估计的匹配准则, 使用式(5)作为匹配准则。

$$\text{SAD}[\mathbf{B}_{ij}] = \text{SAD\_inter}[\mathbf{B}_{ij}] + \lambda \text{SAD\_intra}[\mathbf{B}_{ij}] \quad (5)$$

$$\text{SAD\_intra}[\mathbf{B}_{ij}] = \sum_{h=0}^{N-1} \sum_{l=0}^{N-1} |x_{kl} - x'_{kl}| \quad (6)$$

式(6)中的  $\text{SAD\_intra}$  表示当前块的上水平边界和左垂直边界与它们相邻边界的绝对差,  $N$  是一个块的宽度, 块的大小为  $N \times N$  的大小。  $\lambda$  是加权系数, 取 0 ~ 16 的整数。它是一个经验值, 对于运动较剧烈的序列,  $\lambda$  取值应该偏小, 对运动较平缓的序列,  $\lambda$  取值应该偏大。

### 1.4 运动矢量处理

对称运动估计后就能得到一个初始的 MVF, 这个 MVF 里面大部分 MV 是可靠的, 但是仍然存在部分不可靠的 MV。不可靠的 MV 会导致对应块在插帧时发生错误, 进而严重影响主观效果。因此, 运动矢量处理是 MCFI 中一个非常重要的部分。

一般来说, SAD 较大也就是匹配误差较大, 这样的块对应的 MV 往往是不可靠的, 因此, SAD 是判断 MV 是否可靠的一个基本原则。但是对于有些块虽然 SAD 很小, 但是其 MV 也不可靠。这种情况产生的原因是一幅图像里有很多相似或者重复的区域, 一个块可能对应几个最佳匹配块, 依据 SAD 最小, 得到的 MV 可靠性很难保证。这样的 MV 可以通过 MV 的相关性来判断进而进行处理<sup>[8-9]</sup>。

本文就是结合 SAD 和 MV 的相关性来对 MV 的可靠性进行分类。

设  $(w, h)$  是当前块在一幅图像所有块中的位置,  $w$  是当前块所在的行块数,  $h$  是当前块所在的列块数,  $(x_{w,h}, y_{w,h})$  是当前块的运动矢量。

当前块与之周围的 8 个相邻块之间的相关系数定义为

$$C_{w,h} = \frac{\frac{1}{8} \sum_{i=-1}^1 \sum_{j=-1}^1 \sqrt{(x_{w,h} - x_{w+i,h+j})^2 + (y_{w,h} - y_{w+i,h+j})^2}}{\frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 \sqrt{x_{w+i,h+j}^2 + y_{w+i,h+j}^2}} \quad (7)$$

当前块与周围相邻块的平均相关系数定义为

$$C_{w,h}^{\text{avg}} = \frac{1}{9} \sum_{i=-1}^1 \sum_{j=-1}^1 C_{w+i,h+j} \quad (8)$$

将一幅图像的所有运动矢量分为 3 类,第 1 类是由于 SAD 过大引起的不可靠 MV,标记为  $L_1$ ;第 2 类是由于当前块的 MV 与周围块的 MV 相关性不大引起的不可靠 MV,标记为  $L_2$ ;剩下的第 3 类为可靠的 MV,标记为  $L_3$ 。

用式子表示如下

$$\text{MVreli}_a\text{-flag} = \begin{cases} L_1 & \text{SAD} > T_1 \\ L_2 & c_{w,h} > c_{w,h}^{\text{avg}} \& c_{w,h} > 0.5 \\ L_3 & \text{其他} \end{cases} \quad (9)$$

式中的 SAD 是式(5)求出来的结果。

对运动矢量分类后,对于标记  $L_1$  和标记  $L_2$  的不可靠 MV 要分别采取不同的方法进行处理。运动矢量的处理过程如下:

1) 将 MV 标记为  $L_1$  的对应块分割为子块。因为这样的块往往是物体的边缘,所以它可能包含两个不同的物体。因此将块划分为小块,每一块包含一个物体,这样能使物体的边缘更清晰;

2) 将相邻的标记为  $L_2$  的对应块合并为大块。因为  $L_2$  类的 MV 往往是出现在周围运动矢量都很平滑的区域,而只有被标记的这个 MV 不规则,因此要与周围的块合并成一个大块;

3) 经过前面的分割和合并之后,块的大小已经不是初始的固定大小了,变成了可变块大小。对这些可变块要重新运动估计,得到一个新的 MVF;

4) 对更新后的 MV 的可靠性重新分类;

5) 重复以上的 4 个过程,直到  $L_1$  类别对应块的 SAD 小于某一固定阈值或者是最小分割块达到  $2 \times 2$  的块大小;

### 1.5 自适应运动补偿

MCFI 实际上是一个线性滤波的过程<sup>[10]</sup>,插值结果可用下式表示

$$\hat{f}_n[s, v] = \lambda f_{n-1}[s - \text{MV}] + (1 - \lambda) f_{n+1}[s + \text{MV}] \quad (10)$$

式中,  $f_{n-1}$  和  $f_{n+1}$  是已知的两相邻帧,  $\hat{f}_n$  是预测出来的帧。  $s$  表示一个分割块,  $\text{MV}$  是该块对应的运动矢量。权值  $\lambda$  一般取 0.5。但是这种固定权值的滤波器往往会局部模糊。如图 4 所示, Foreman 序列人物的嘴巴部分比较模糊。造成这种模糊的原因是在前一帧中人的嘴巴是闭合的,而在后一帧中,嘴巴是张开的,因此,按式(10)中  $\lambda$  取 0.5,就相当于把这两个块进行加权平均,必然会导致模糊现象。因此需要检测出序列内容的变化来自适应地调整  $\lambda$  的取值。按以下原则选择  $\lambda$ 。



图 4 Foreman 序列第 91 帧主观效果对比图

Fig. 4 Experiment results of frame 91 of foreman sequence

$$\lambda = \begin{cases} 0.5 & \text{SAD} < T_2 \\ 1 \text{ 或 } 0 & \text{其他} \end{cases} \quad (11)$$

如果当前块的 SAD 超过了某一阈值,那么采用前向帧或者后向帧对当前帧进行插补,否则,使用双向帧进行填补。

## 2 实验结果分析

实验对象选取的是 YUV 格式的 CIF 大小的视频序列,测试的方式是先将 30 帧/s 的原始视频序列去掉偶数帧,裁剪成 15 帧/s 的序列,然后经过本方法又转换为 30 帧/s 的序列。

对同一实验,采用如图 1 所示的处理流程来验证所提出的算法。对输入序列逐步进行场景检测、运动矢量处理和自适应运动补偿,得到最终的实验结果。主观效果分别选取了部分图像对比分析算法的效果。客观效果给出了单帧 PSNR 和平均 PSNR。

主观结果如图 4—图 6 所示。图 4(a) 是 Foreman 序列使用第 90 帧和 92 帧采用内插的方法

得到的 91 帧的结果,方块效应非常明显。(b)使用第 88 帧和第 90 帧外推出来的 91 帧的结果。显然外推方法比内插方法的结果又很大的改善。

图 5(a)(b)分别是使用运动矢量处理前的结果和使用运动矢量后的结果。在两幅图中的铁塔处纹理较强,很多块很相似,因此运动估计的时候一个块与周围多个块都很匹配,因此会有不规则的运动矢量。对运动矢量不加处理就会造成(a)所示的结果。

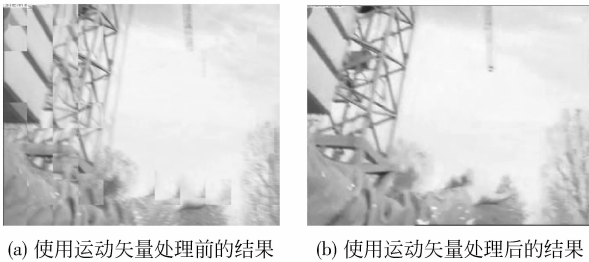


图 5 Foreman 序列第 197 帧主观效果对比图

Fig. 5 Experiment results of frame 197 of Foreman sequence

图 6 是自适应运动补偿的对比结果。图 6(b)较(a)在人的嘴巴部分更清晰。



图 6 Foreman 序列第 53 帧主观效果对比图

Fig. 6 Experiment results of frame 53 of Foreman sequence

为了验证所提出的帧速率上转换算法的客观效果,进行了测试实验,将提出的方法与 MCFI 方法进行比较。本实验选用了 Bus、Foreman、Football、Flower 4 个序列测试客观效果,分别取这 4 个序列的前 50 个插值帧计算 PSNR。

图 7 是 Foreman 序列采用本文方法与 MCFI 方法的单帧 PSNR 对比结果。表 1 是 4 个序列各自平均 PSNR 对比结果。结果表明算法对序列的客观质量也有一定的提升。

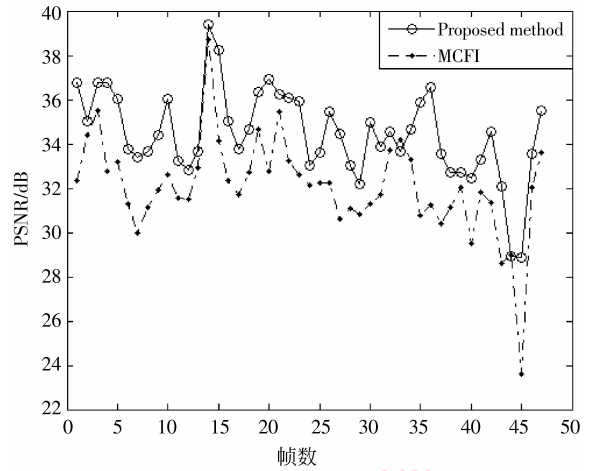


图 7 Foreman 序列前 50 帧的 PSNR 对比图

Fig. 7 PSNR comparisons between MCFI method and the proposed method of Foreman sequence

表 1 采用 MCFI 方法和本文所提出方法的平均 PSNR/dB 对比

Tab. 1 Average PSNR comparisons between MCFI method and the proposed method

序列名称	本文方法	MCFI
Bus	25.36	25.568
Foreman	33.63	30.275
Football	23.85	23.34
Flower	31.23	28.099

### 3 结 论

在帧率转换的 3 个不同的阶段分别采取了不同的方法来提升插值图像的主观效果。首先,利用场景检测来判断是使用用内插的方法还是外推的方法。然后对运动估计得到的初始 MVF 进行更新校验,以得到最可靠的 MV。最后自适应地选择插值滤波器的权值  $\lambda$  进一步消除局部模糊现象。

场景检测采用帧差法,在帧率转换应用中简单实用。因为它只需统计 SAD 超过某一阈值的大小的个数,其计算复杂度与运动估计的复杂度比较起来非常小,对整体复杂度的影响不大。因此采用的场景检测算法不仅能提高内插帧的质量,同时也适合实时应用。

### 参考文献 (References)

[ 1 ] Patti A J, Sezan M I, Tekalp A M. Digital video standards conversion in the presence of accelerated motion [ C ]

- //Proceedings in IEEE International Conference on Acoustics, Speech, and Signal Processing. Adelaide, SA, Australia; IEEE Press, 1994;225-228.
- [ 2 ] Choi B, Lee S, Ko S. New frame rate up-conversion using Bi-directional Motion Estimation [ J ]. IEEE Transactions on Consumer Electronics, 2000, 46(3) : 603-609.
- [ 3 ] Ojo O A, Haan G. Robust motion-compensated video upconversion[J]. IEEE Transactions on Consumer Electronics, 1997, 43(4) :1045-1056.
- [ 4 ] Chen T. Adaptive temporal interpolation using bidirectional motion estimation and compensation [ C ]//Proceedings in IEEE International Conference on Image Processing. Rochester, NY: IEEE Press, 2002;313-317.
- [ 5 ] Hong B, Eom M, Choe Y. Scene change detection using edge direction based on intra prediction mode in H.264/AVC compression domain [ C ]//Proceedings in IEEE Region 10 Conference on TENCON. Hong Kong: IEEE Press, 2006; 1-4.
- [ 6 ] Song H B, Men A D, Shi G J. A Method for halo artifact reduction in MEMC [ C ]//Proceedings in IEEE International Conference on Consumer Electronics. Hong Kong: IEEE Press, 2009; 1-2.
- [ 7 ] Choi B, Han J, Kim C, et al. Motion-compensated frame interpolation using bilateral motion estimation and adaptive overlapped block motion compensation [ J ]. IEEE Transactions on Circuits and Systems for Video Technology, 2007, 17(4) : 407-416.
- [ 8 ] Huang A M, Truong N. Correlation-based motion vector processing for motion compensated frame interpolation [ C ] // Proceedings in IEEE International Conference on Image Processing. San Diego, California, USA: IEEE Press, 2008: 1244-1247.
- [ 9 ] Huang A M, Truong N. Correlation-based motion vector processing with adaptive interpolation scheme for motion-compensated frame interpolation [ J ]. IEEE Transactions on Image Processing, 2009, 18(4) : 740-752.
- [ 10 ] Lee S H, Kwon O, Park R H. Weighted-adaptive motion-compensated frame rate up-conversion [ J ]. IEEE Transactions on Consumer Electronics, 2003, 49( 3 ):485-492.