

加权边沿自适应的场内插值去隔行方法*

马 斌, 丁 勇, 周 升, 严晓浪

(浙江大学 超大规模集成电路设计研究所, 杭州 310025)

摘要: 为实现隔行扫描图像到逐行扫描图像的扫描格式转换, 提出了一种基于空间加权和边沿自适应的场内插值去隔行算法。该算法包括 LRV(left right vertical) 评估和边沿自适应插值。相比传统的去隔行算法, LRV 评估可以有效地消除伪边沿插值; 在边沿自适应插值中增加预插值判断进行插值方向初步估计, 可以抑制反向插值错误; 在边沿方向搜索中增加空间权值, 可以提高插值的准确率。实验结果表明, 该算法在处理图片和视频流时的图像质量均要优于传统的算法。

关键词: 去隔行; 空间加权; 边沿自适应; 插值

中图分类号: TP391 **文献标志码:** A **文章编号:** 1001-3695(2010)08-3168-03

doi: 10.3969/j.issn.1001-3695.2010.08.098

De-interlacing with weighted edge adaptive intra-field interpolation

MA Bin, DING Yong, ZHOU Sheng, YAN Xiao-lang

(Institute of VLSI Design, Zhejiang University, Hangzhou 310025, China)

Abstract: To implement the scanning format conversion from interlaced image to progressive image, this paper proposed a high performance de-interlacing algorithm based on spatial weighted and edge adaptive. The algorithm included two steps: LRV evaluation and edge adaptive interpolation. Compared with previous algorithms, LRV evaluation could remarkably reduce the fake edge interpolation, the prejudgment of edge before edge adaptive interpolation could suppress the reversal interpolation error, considered spatial weighting in weighted edge evaluation could reduce the edge interpolation error. Extensive simulations conducted for images and video sequences show the efficacy of the proposed algorithm over the previous algorithms in terms of the objective and subjective image quality.

Key words: de-interlacing; spatial weighting; edge adaptive; interpolation

早期由于信道带宽的限制, 传统模拟信号电视都采用隔行扫描制式, 但隔行扫描制式会导致画面抖动、闪烁、图像质量降低等缺点。数字电视出现后, 随着人们对视频质量要求的提高, 逐行扫描逐步取代隔行扫描成为数字电视扫描方式的首选。去隔行技术是将隔行扫描信号转换为逐行扫描信号的一种视频格式转换技术, 通过该技术可以显著提高图像质量。

去隔行算法主要分为空间域和时间域去隔行算法两类, 空间域去隔行算法是时间域去隔行算法的基础。空间域去隔行算法使用单一场的信息进行插补, 如 LA(line average)^[1]、ELA(edge-based line average)^[2] 等方法。时间域去隔行算法则从时间上动态结合多个场的信息进行插补, 如运动检测^[5,6]、运动补偿^[7,9] 等方法。本文主要讨论空间域去隔行算法。

空间域去隔行的一种典型算法是场内行平均法 LA^[1], 采用同一场的上下两行取平均值的方法进行插值, 该算法实现简单, 但会导致锯齿状沿的产生, 对画面清晰度损伤较大。ELA^[2] 算法采用边沿插值方法进行去隔行操作, 有效解决了图像锯齿状沿问题, 但在显示画面细节方面略显不足。AELA(advanced edge-based line average)^[3]、DOI(direction oriented interpolation)^[4] 等算法对 ELA 算法进行了改进。AELA 算法增加了靠近水平方向的边沿搜索, 提高了插值后图像的质量, 但在处理中包含空间高频分量的图像时容易产生插值错误。

DOI 算法增加了边沿方向判断函数, 能有效地减少插值错误, 但并没有考虑在图像的细节丰富处的跨边沿插值错误问题。

本文的算法在前人的基础上增加 LRV 评估抑制伪边沿插值, 同时, 在计算边沿方向函数时增加了空间加权值, 抑制了边沿插值错误; 最后在边沿方向搜索时采用提前终止的方式, 消除了跨边沿错误插值。

1 边沿插值的基本原理

边沿插值的原理如图 1 所示。假设当前需要对点 $P(i, j)$ 进行插值操作, 上下两行像素点的对角线方向代表了可能的边沿方向, 通过计算、比较对角线上像素之间的绝对差值, 判断对角线对应像素与待插补像素的相关性, 绝对差值越小相关性越大, 最小差值对应的方向就是边沿方向, 然后通过边沿方向上的相关像素点插补出点 $P(i, j)$ 。为方便下面的讨论, 将点 $P(i-1, j+k)$ 和点 $P(i+1, j-k)$ 的对角线方向定义为点 $P(i, j)$ 的边沿方向 k 。

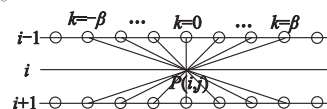


图1 边沿插值的基本原理

基于边沿插值的去隔行算法主要存在两类问题: a) 边沿

收稿日期: 2009-12-16; 修回日期: 2010-01-05 基金项目: 国家“863”计划资助项目(2009AA011706)

作者简介: 马斌(1984-), 男, 浙江绍兴人, 硕士研究生, 主要研究方向为集成电路设计和视频算法; 丁勇(1974-), 男, 山东青岛人, 副研究员, 博士, 主要研究方向为 VLSI 设计、视频图像处理、视频编码等(dingy@vlsi.zju.edu.cn); 周升(1985-), 男, 浙江衢州人, 硕士研究生, 主要研究方向为集成电路设计和验证; 严晓浪(1947-), 男, 浙江杭州人, 教授, 硕士, 主要研究方向为超大规模集成电路设计与制造。

搜索和插值运算时容易受到噪声干扰;b)在处理图像的细节丰富区域时容易产生边沿插值错误。本文算法对这两类问题进行了改进。

2 加权边沿自适应场内插值去隔行算法

本文提出的算法包括 LRV 评估、边沿自适应插值两个部分。边沿自适应插值又分成预插值方向判断、加权边沿方向评估和边沿方向搜索及插值三个步骤。图 2 为算法的流程图。首先进行 LRV 评估,判断当前待插值点 $P(i, j)$ 是否位于边沿上,如果不位于边沿上或边沿为垂直,则进行上下行平均插值,如果位于边沿上且边沿不为垂直,则进行边沿自适应插值。

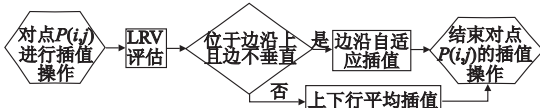


图2 加权边沿自适应去隔行算法流程图

2.1 LRV 评估

LRV 评估先计算点 $P(i, j)$ 的边沿左偏可能值 LP 、边沿右偏可能值 RP 、边沿垂直可能值 VP , 然后通过 LP 、 RP 、 VP 判断当前待插值点是否位于边沿上。LRV 评估筛选掉了不需要进行边沿自适应插值的像素点,有效抑制了伪边沿插值。

式(1)是 LP 、 RP 、 VP 的计算公式:

$$\begin{aligned} LP &= \sum_{k=-\alpha}^{\alpha} |P(i-1, j-1+k) - P(i+1, j+1+k)| \\ VP &= \sum_{k=-\alpha}^{\alpha} |P(i-1, j+k) - P(i+1, j+k)| \\ RP &= \sum_{k=-\alpha}^{\alpha} |P(i-1, j+1+k) - P(i+1, j-1+k)| \end{aligned} \quad (1)$$

其中: $\alpha \geq 0$, α 为预先设置的参数,表征的是参加 LRV 评估的像素点数。 α 越大,参加评估的像素点越多,评估的准确率越高,但相应的运算复杂度也会提高。根据实验的结果,本文采用折中的 $\alpha=1$ 进行 LRV 评估。图 3 是 $\alpha=1$ 的 LP 、 VP 、 RP 计算示意图。

算得 LP 、 VP 、 RP 后进行下列判断:

a) 判断 $(VP < LP) \&\& (VP < RP)$ 是否为真,如果为真,则判定点 $P(i, j)$ 位于垂直边沿,通过式(2)对 $P(i, j)$ 进行上下行平均插值,结束插值计算;

b) 判断 $(VP > LP) \&\& (VP > RP)$ 是否为真,如果为真,则判定点 $P(i, j)$ 并不位于边沿上,通过式(2)对点 $P(i, j)$ 进行上下行平均插值,结束插值计算;

c) 其余情况下判定点 $P(i, j)$ 位于边沿上,进入边沿自适应插值步骤。

式(2)是上下行平均插值的计算公式:

$$P(i, j) = (P(i-1, j) + P(i+1, j)) / 2 \quad (2)$$

2.2 边沿自适应插值

根据 LRV 评估的结果,对部分像素点进行边沿自适应插值。首先进行预插值方向判断,判断点 $P(i, j)$ 的插值大致方向为向左、向右或垂直。插值方向为垂直时使用式(2)进行上下行平均插值,结束插值计算。插值方向为向左或向右时首先进行加权边沿方向评估计算点 $P(i, j)$ 的加权边沿方向函数 $WEdge(k)$, 然后根据 $WEdge(k)$ 搜索得到点 $P(i, j)$ 的边沿方向 K , 根据边沿方向 K 上的像素点插补出 $Q(i, j)$, 最后对 $Q(i, j)$ 、 $P(i-1, j)$ 、 $P(i+1, j)$ 进行中值滤波,得到最终的插值结果。下面将详细论述边沿自适应插值的三个步骤:预插值方向判

断、加权边沿方向评估、边沿方向搜索及插值。

2.2.1 预插值方向判断

预插值方向判断引入了边沿方向值 $DIRECTION$ 的概念进行插值边沿方向的大致判断。 $DIRECTION$ 表征的是当前待插值点边沿的大致方向,如向左、向右或垂直。通过预先对插值方向的判断,可以有效地消除反向插值错误的产生,并使得加权边沿方向评估和边沿方向搜索的计算量减少一半。

式(3)是边沿方向值 $DIRECTION$ 的计算公式。预插值方向算法示意图如图 4 所示。

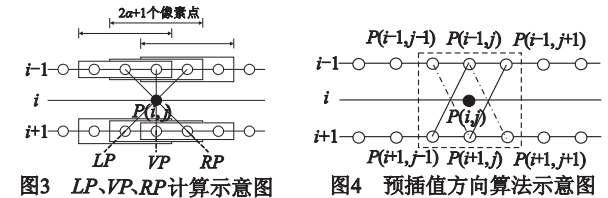


图3 LP 、 VP 、 RP 计算示意图

图4 预插值方向算法示意图

$$\begin{aligned} left &= |P(i-1, j) - P(i+1, j+1)| + |P(i-1, j-1) - P(i+1, j)| \\ right &= |P(i-1, j) - P(i+1, j-1)| + |P(i-1, j+1) - P(i+1, j)| \\ DIRECTION &= left - right \end{aligned} \quad (3)$$

算得 $DIRECTION$ 后进行下列判断:

a) 判断 $|DIRECTION| < T_direction$ 是否为真,其中, $T_direction$ 为预先设定的阈值,如果为真,则判定点 $P(i, j)$ 位于垂直边沿,使用式(2)对点 $P(i, j)$ 进行上下行平均插值,结束插值计算;

b) 如果 $|DIRECTION| \geq T_direction$ 且 $DIRECTION > 0$, 判定点 $P(i, j)$ 的边沿大致方向为向左; $|DIRECTION| \geq T_direction$ 且 $DIRECTION < 0$, 判定点 $P(i, j)$ 的边沿大致方向为向右。算法进入加权边沿方向评估步骤。

2.2.2 加权边沿方向评估

经过预插值方向判断后,获得了点 $P(i, j)$ 边沿的大致方向,如果插值方向为向左,则需要对方向 $k=0, 1, \dots, \beta$ 计算加权边沿方向函数 $WEdge(k)$, β 是预先设定的参数,表征的是需要进行评估的边沿方向范围,如果插值方向为向右,则需要对方向 $k=0, -1, \dots, -\beta$ 计算 $WEdge(k)$ 。

$WEdge(k)$ 的计算步骤如下:

a) 如图 5 所示,通过式(4)计算 $EL(k)$ 、 $Em(k)$ 、 $Er(k)$:

$$\begin{aligned} EL(k) &= \sum_{l=-\gamma}^{\gamma} |P(i-1, j+k-1+l) - P(i+1, j-k-1+l)| \\ Em(k) &= \sum_{l=-\gamma}^{\gamma} |P(i-1, j+k+l) - P(i+1, j-k+l)| \\ Er(k) &= \sum_{l=-\gamma}^{\gamma} |P(i-1, j+k+1+l) - P(i+1, j-k+1+l)| \end{aligned} \quad (4)$$

其中: γ 为预先定义的参数, γ 表征的是参与边沿方向评估的像素点数,方向 k 上的 $2\gamma+1$ 个像素点组成的像素块参与了计算,相比于传统算法中使用单个像素点进行计算,提高了边沿方向评估的精确度。

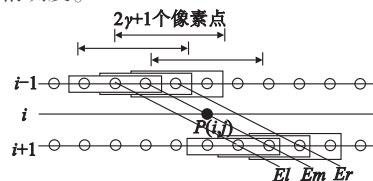


图5 $EL(k)$ 、 $Em(k)$ 、 $Er(k)$ 的计算示意图

b) 对 $EL(k)$ 、 $Em(k)$ 、 $Er(k)$ 进行中值滤波得到边沿方向函数 $Edge(k)$:

$$Edge(k) = \text{median}(EL(k), Em(k), Er(k)) \quad (5)$$

通过对 $EL(k)$ 、 $Em(k)$ 、 $Er(k)$ 进行中值滤波消除图像中噪

声点对边沿方向评估的干扰。

c) 计算加权边沿方向函数 $WEdge(k)$ 。计算公式为

$$WEdge(k) = Edge(k) + a_1 \times |k| + a_2 \times |k|^2 + \dots + a_n \times |k|^n \quad (6)$$

其中: a_1, a_2, \dots, a_n 为预先设定的参数。 $a_1 \times |k| + a_2 \times |k|^2 + \dots + a_n \times |k|^n$ 的项数越多, 对边沿方向评估实际情况的拟合也会越好, 但相应地也会提高算法的复杂度。根据实验的结果, 本文采用:

$$\begin{cases} a_n = 0.8 (n=1) \\ a_n = 0 (n > 1) \end{cases}$$

相比传统算法的边沿方向评估, 本文提出的加权边沿方向评估将具有以下优势: a) 通过比较上下两行的 $2\gamma + 1$ 的像素块消除噪声像素点带来的影响; b) 在传统边沿自适应插值方法单边沿上下行相似度比较的同时, 引入相邻同一方向的两条边沿进行中值滤波, 更精确地实现了边沿评估; c) 随着 β 的增大, 边沿方向的搜索范围 $-\beta \leq k \leq \beta$ 也会相应增大, 边沿方向的搜索更加准确, 但相应也会引入大的 k 值, k 值越大, 点 $P(i-1, j+k)$ 、点 $P(i+1, j-k)$ 和当前待插值点 $P(i, j)$ 的距离也会越远, 它们之间的相关性也会越小, 容易引入错误的边沿插值。本算法对边沿方向函数 $Edge(k)$ 进行空间加权, 随着 k 的增大, $a_1 \times |k| + a_2 \times |k|^2 + \dots + a_n \times |k|^n$ 也会相应变大, 抑制了 k 变大所带来错误边沿插值的可能性。

2.2.3 边沿方向搜索及插值

加权边沿方向评估算得的 $WEdge(k)$ 是边沿方向搜索及插值的依据。算法要得到点 $P(i, j)$ 最终的插值结果依赖于以下两个步骤:

a) 假设当前待插值点 $P(i, j)$ 的预插值方向评估的结果为左偏, 需要在 $k=0, 1, \dots, \beta$ 中选择一个正确的边沿方向 K 。计算 K 的具体算法如下:

```

WEdge( $\beta + 1$ ) =  $\infty$ ;
for ( $k=0; k < \beta; k++$ )
    if ( $WEdge(k) < WEdge(k+1)$ )
        break;
end
K = k;
    
```

K 便是最终选定的边沿方向。在边沿方向搜索的过程中, 采取了提前终止的方式: 即随着 k 从 0 变到 β , 当 $WEdge(k)$ 从递减变为递增时, 就认为已经检测到了相应的边沿。相比传统算法从所有 $WEdge(k)$ 中找出最小值对应的 k , 这种方式可以消除跨边沿错误插值的可能性, 即 $WEdge(k)$ 反复递增、递减, 提高了边沿搜索的准确率。

b) 插值运算。

通过式(7)计算出方向 K 上的插值:

$$Q(i, j) = (P(i-1, j+K) + P(i+1, j-K)) / 2 \quad (7)$$

通过式(8)对 $Q(i, j)$ 、 $P(i-1, j)$ 、 $P(i+1, j)$ 进行中值滤波, 得到最终的插值结果:

$$P(i, j) = \text{median}(P(i-1, j), P(i+1, j), Q(i, j)) \quad (8)$$

3 实验结果及分析

3.1 图像实验

为评估本文算法对图像的场内插值去隔行效果, 在实验中采用“Baboon”“Barbara”“Boat”“Airplane”“Lena”“Man”“Pepper”“Mandrill”“Tiffany”等测试图像。参与比较的传统去隔行算法包括 LA、ELA、AELA、DOI。本文所提出的算法与 LA、ELA、AELA、DOI 等去隔行算法处理后的 PSNR (peak signal-to-

noise ratio, 峰值信噪比) 进行了比较, 如表 1 所示。

表 1 本文算法与传统去隔行算法的 PSNR 比较

测试图像	LA	ELA	A-ELA	DOI	Proposed
Baboon	23.38	22.87	23.90	23.47	23.97
Barbara	32.11	25.11	25.11	28.52	30.57
Boat	35.35	32.35	32.64	34.66	34.79
Airplane	31.87	31.12	30.92	31.89	32.07
Lena	37.67	35.99	35.87	37.94	38.08
Man	31.98	30.83	31.50	32.03	32.81
Pepper	33.98	34.23	33.96	33.38	34.20
Mandrill	21.62	22.98	22.95	23.18	23.48
Tiffany	29.52	30.22	30.20	30.47	30.69

从表 1 可以看出, 本文算法的 PSNR 值除在处理“Barbara”“Boat”图片时略低于 LA 算法外, 其他情况下都要优于传统的去隔行算法。与 DOI 算法相比, PSNR 平均提高了 0.6 dB。

图 6 是本文算法与 LA、ELA、AELA、DOI 等传统算法对“Barbara”进行去隔行处理后的截图主观效果比较。其中, (a) 为原图, (b) 为 LA 效果图, (c) 为 ELA 效果图, (d) 为 AELA 效果图, (e) 为 DOI 效果图, (f) 为本文算法的效果图。

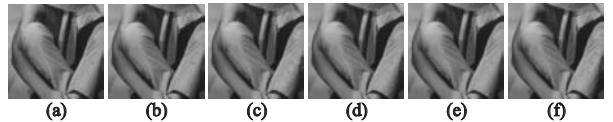


图 6 各种去隔行算法处理效果比较

从图 6 可以看出, ELA、AELA、DOI 这三种基于边沿插值的去隔行算法在处理“Barbara”裤子这个区域时发生了大量的插值错误, 而本文算法有效抑制了这些插值错误, 在裤子的这个区域也能较好地插补出原样。

3.2 视频流实验

为评估算法对视频流的去隔行效果, 在实验中采用了“foreman”“flower”的前 100 帧图像进行了测试。参与比较的传统去隔行算法包括 LA、ELA、AELA、DOI。本文算法与 LA、ELA、AELA、DOI 等去隔行算法的平均 PSNR 比较如表 2 所示。

表 2 本文算法与传统去隔行算法对视频流的平均 PSNR 比较

图像	LA	ELA	AELA	DOI	Proposed
flower	22.15	21.79	21.83	22.09	22.28
foreman	33.65	34.44	34.87	34.89	35.24

从表 2 可以看出, 本文算法的平均 PSNR 要优于传统的去隔行算法。图 7、8 分别是本文算法和传统算法处理“foreman”“flower”每一帧图像时的 PSNR 值比较。

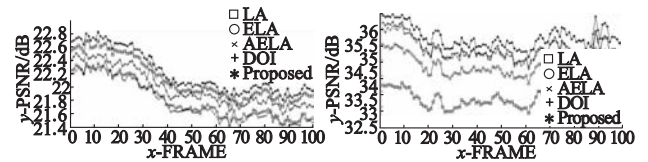


图 7 处理“flower”视频流时各种算法的 PSNR 比较

图 7、8 中的 X 轴为图像的帧数, Y 轴为每一帧图像的 PSNR 值, 单位 dB。从图 7、8 可以看出, 本文算法在处理“flower”“foreman”图像时的 PSNR 值要优于其他传统算法。

4 结束语

针对边沿插值方法存在的问题, 本文提出了一种基于空间加权和边沿自适应的空间域去隔行算法。实验结果表明, 本文算法有效地改善了上述两个问题, 提高了去 (下转第 3175 页)

$$\theta(t + \Delta t) = \theta(t) + \dot{\theta}(t + \Delta t) \quad (24)$$

每一帧动画利用式(22)(24)重复计算,从而得到振动回复过程中草叶所有关节的弯曲和扭转角度。

2.2.2 草丛 billboard 近似形变

对于精度不高的中距离局部草场,通过移动布告板上端顶点的方法模拟草丛形变。由于场景中多种表达方式,必须考虑几何草体形变与布告板形变之间的平滑过渡。为此本文采用一套形变数据对应两种表达方式,即利用部分几何形变数据作用于布告板顶点,使得两种变形所产生的草叶姿态尽量接近。

由几何数据所得到的布告板形变向量可表示为

$$M = (L \cdot \sin\left(\frac{\sum_{i=0}^n \theta}{n}\right) \cdot (\text{normal}(V) \cdot x, L \cdot \cos\left(\frac{\sum_{i=0}^n \theta}{n}\right) - L \cdot \sin\left(\frac{\sum_{i=0}^n \theta}{n}\right) \cdot (\text{normal}(V) \cdot y) \quad (25)$$

其中: n 为单根草叶的分节节数; L 为单节草叶长度; θ 为各节草叶关节弯曲角度; V 表示风速向量; $\text{normal}(V)$ 表示对风速向量单位化。每个布告板下方两顶点保持不动,上方两顶点使用相同的三维形变向量 M 进行形变叠加计算。

这种方法不仅可以满足不同表达方式间形变的平滑过渡,同时也大大增加了布告板级别草体风吹动画的生动性,使得布告板所表示的草丛同样可以展现出实时可变风场影响下的动态特性,并支持对风场消失后草丛振荡回复过程的表现。

3 实验结果与分析

本文利用 OpenGL,在配置为 Intel 奔腾双核 E5300,2.0 GB 内存,NVIDIA 9800GT 显卡的 PC 机上对本文方法进行了实验。图6分别展示了利用混合式表达方式构建的大面积静态草地场景,以及爆炸风、旋风作用下的效果。



图6 各种情况下的草场效果

(上接第3170页)隔行扫描转换的图像处理效果。在本文中,对视频的去隔行转换进行了初步的实验,但是并没有在转换中考虑时间域上的信息。在下一步的研究工作中,笔者将把本文提出的基于空间加权和边沿自适应的空间域去隔行算法,结合时间域上的运动检测理论,用于视频流的去隔行转换。

参考文献:

- [1] HAAN G, BELLARS E B. Deinterlacing: an overview[J]. *Proceedings of the IEEE*, 1998, 86(9):1839-1857.
- [2] DOYLE T. Interlaced to sequential conversion for EDTV applications [C]//Proc of the 2nd International Workshop on Signal Processing of HDTV. 1988:412-430.
- [3] KUO C J, LIAO C, LIN C C. Adaptive interpolation technique for scanning rate conversion[J]. *IEEE Trans on Circuits and Systems for Video Technology*, 1996, 6(3):317-321.
- [4] YOO H, JEONG J. Direction-oriented interpolation and its application to de-interlacing[J]. *IEEE Trans on Consumer Electronics*,

在不考虑风场的情况下,整个静态草地场景实时漫游平均帧速为 70 fps。当在场景中加入风场后,平均帧速在 1018 fps,其变化主要取决于风场的大小以及视点的位置。同时,在实验中本文测试了可变风场对草体动画的影响。结果表明,无论风场如何变化,整个动画效果非常连贯,当风场忽然消失时,草叶受回复力作用振动的效果较为逼真。

4 结束语

本文提出了一种风场作用下大规模草地场景动态模拟方法,该方法不仅能逼真模拟出草体在各种风场作用下的动态效果,而且对草体受回复力作用时所表现出的振动效果也提供了较好支持。由于局部草场采用了基于物理的模拟方法,计算开销相对较大,下一步工作主要考虑引入模拟细节层次的方法,进一步简化物理计算。

参考文献:

- [1] PERBET F, CANI M P. Animating prairies in real-time[C]// Proc of Symposium on Interactive 3D Graphics. New York: ACM, 2001: 227-232.
- [2] GUERRAZ S, PERBET F, RAULO D, et al. A procedural approach to animate interactive natural sceneries[C]// Proc of the 16th International Conference on Computer Animation and Social Agents. New York: ACM, 2003:73-78.
- [3] PELZER K. Rendering countless blades of waving grass[EB/OL]. (2008-07-15)[2009-12-05]. http://developer.nvidia.com/GPU-Gems/gpugems_ch07.html.
- [4] BEZRATI A. Technical report nature scene[EB/OL]. (2005-08-15)[2009-12-05]. http://developer.nvidia.com/object/nature_scene.html.
- [5] BAKAY B. Animating and lighting grass in real-time[D]. Vancouver: University of British Columbia, 2003.
- [6] 王长波. 基于物理模型的自然景物实时绘制[D]. 杭州:浙江大学, 2006.
- [7] BANISCH S, CHARLES A W. Making grass and fur move[J]. *Journal of WSCG*, 2006, 14(3):213-220.
- [8] 2002, 48(4):954-962.
- [5] GILLIES D, PLANTHOLT M, WESTERKAMP D. Motion adaptive field rate upconversion algorithms for 900 lines/100 Hz/2:1 displays [J]. *IEEE Trans on Consumer Electronics*, 1990, 36(2):149-160.
- [6] WANG Y, MITRA SK. Motion/pattern adaptive interpolation of interlaced video sequences [C]// Proc of Acoustics Speech and Signal Processing. 1991:2829-2832.
- [7] HARGREAVES D, VAISEY J. Bayesian motion estimation and interpolation in the interlaced video sequences[J]. *IEEE Trans on Image Processing*, 1997, 6(5):764-769.
- [8] PATTI A, SEZAN M, TEKALP A. Robust methods for high-quality stills from interlaced video in the presence of dominant motion [J]. *IEEE Trans on Circuits and Systems for Video Technology*, 1997, 7(2):328-341.
- [9] SUGIYAMA K, NAKAMURA H. A method of de-interlacing with motion compensation interpolation [J]. *IEEE Trans on Consumer Electronics*, 1999, 45(3):611-616.