

一种时空权重和边缘自适应的扫描格式转换

丁 勇, 陆生礼, 时龙兴

(东南大学国家专用集成电路系统工程技术研究中心, 南京 210096)

[摘要] 为实现隔行扫描到逐行扫描的视频扫描格式转换, 提出了一种时空权重和边缘自适应的去隔行算法, 主要包括运动估计、小角度边缘搜索、时空权重计算、自适应插值等。该算法通过 4 场相邻像素的最大灰度变化与运动阈值比较实现对当前像素的运动估计, 采用自适应搜索半径和并行搜索树的方法实现小角度边缘检测, 并且增加对半像素边缘的考虑实现最小 6° 边缘的检测, 最后通过时空权重自适应的插值算法实现去隔行处理, 取得很好的处理效果。

[关键词] 扫描格式转换; 运动估计; 边缘搜索; 时空权重自适应插值; 去隔行

[中图分类号] TP752.1; TP751.1; TP391.41 [文献标识码] A [文章编号] 1009-1742(2007)10-0049-06

去隔行是实现视频信号从隔行到逐行扫描格式转换的处理技术^[1]。高质量的去隔行是视频处理的一个重要课题, 其目的是消除隔行扫描中的行间闪烁、行蠕动、并行和锯齿化等固有缺陷。去隔行技术大致可以分为线性滤波、运动自适应和运动补偿三类。线性滤波法^[2]主要包括行复制、行平均、场合并等, 结构简单, 行复制、行平均法并不能增加图像的垂直分辨率, 而且线性滤波法在处理运动图像时会带来锯齿、羽化等缺陷。自适应插值法^[3-5]是根据运动信息区分运动或静止, 对静止区域采取线性滤波法, 对运动区域采取场内非线性滤波的方法进行插值。这类方法的优点是能够在静止区域增加垂直分辨率并且能够消除运动区域的羽化现象, 缺点是无法提升运动区域的垂直分辨率, 而且由于运动估计的失误会带来插值错误等缺陷。运动补偿法^[6, 7]先进行精确的运动检测, 然后沿着运动轨迹进行插值。运动补偿法最大限度地利用了视频信号时间和空间上的相关信息, 是三类算法中性能最好的一类。但这类算法计算量最大, 对误差极其敏感, 可能由于错误运动估计产生局部失真^[8]。在实际应用中, 快速获得准确、可靠的运动矢量是较困难的, 而且运算复杂度很高、电路资源的消耗很大。

笔者提出一种时空权重和边缘自适应的去隔行算法, 主要包括 4 场同极性场运动估计、小角度边缘搜索、时空权重自适应插值。算法很好地实现视频运动的估计和小角度边缘的检测, 并且通过时空权重自适应的插值算法实现去隔行处理, 取得很好的处理效果。

1 运动估计

影响去隔行处理效果的决定性因素之一是正确地输入视频信号进行运动估计, 将视频图像划分为运动或静止区域, 采取相应的去隔行处理技术^[9]。

视频信号按奇偶场依次传输, 每隔一场极性相同。由于奇场和偶场的数据在场方向相同位置上不尽相同, 传统的采用相邻场(即不同极性场)的运动检测方法会带来一些问题, 尤其对于静止图像水平边缘的检测会产生错误的判断, 从而导致爬行效应^[10]。笔者利用待插补像素在 4 个连续视频场中的相邻像素, 计算它们在亮度信号(Y)上的最大灰度变化, 并与运动阈值比较实现对运动的检测, 其原理如图 1 所示。图中包括 $t-2$, $t-1$, t 和 $t+1$ 共 4 场图像, Z 为待插补像素点。

[收稿日期] 2006-06-12; 修回日期 2006-11-03

[基金项目] 江苏省专用集成电路设计重点实验室开放研究资助项目(JSICK0405)

[作者简介] 丁 勇(1974-), 男, 山东青岛市人, 东南大学博士生, 主要研究方向为 VLSI 设计、视频图像处理、视频编解码等

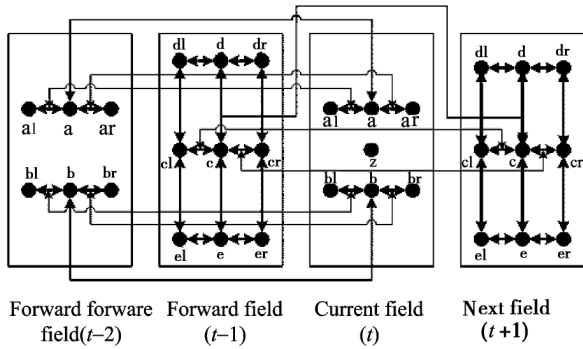


图1 4场运动检测

Fig.1 4-field motion detection

在4场运动检测中,相邻像素灰度的变化主要包括时间方向上的灰度变化(D)、空间水平灰度梯度(HD)和空间垂直灰度梯度(VD)三种类型,其计算如下:

1) 时间灰度变化

计算 t 场和 $t-2$ 场中的对应像素 a 的绝对差值

$$D_a(t, t-2) = |a(t) - a(t-2)| \quad (1)$$

其中 $a(t)$ 和 $a(t-2)$ 分别代表 t 和 $t-2$ 场中像素 a 的亮度分量值。

类似可计算:

$$\begin{aligned} D_b(t-1, t+2) &= |b(t) - b(t-2)|; \\ D_c(t-1, t+1) &= |c(t-1) - c(t+1)|; \\ D_{c_1}(t-1, t+1) &= |c_1(t-1) - c_1(t+1)|; \\ D_{c_r}(t-1, t+1) &= |c_r(t-1) - c_r(t+1)|. \end{aligned}$$

时间方向的最大灰度变化为

$$D_{\max} = \max 5(D_a(t, t-2), D_b(t, t-2), D_c(t, t-2), D_{c_1}(t-1, t+1), D_{c_r}(t-1, t+1)) \quad (2)$$

2) 空间水平灰度梯度

$$\begin{aligned} HD_{a_l}(t) &= |a(t) - a_l(t)|; \\ HD_{a_r}(t) &= |a(t) - a_r(t)|; \\ HD_{a_l}(t-2) &= |a(t-2) - a_l(t-2)|; \\ HD_{a_r}(t-2) &= |a(t-2) - a_r(t-2)|; \\ HD_{b_l}(t) &= |b(t) - b_l(t)|; \\ HD_{b_r}(t) &= |b(t) - b_r(t)|; \\ HD_{b_l}(t-2) &= |b(t-2) - b_l(t-2)|; \\ HD_{b_r}(t-2) &= |b(t-2) - b_r(t-2)|; \\ HD_{c_l}(t) &= |c(t) - c_l(t)|; \\ HD_{c_r}(t) &= |c(t) - c_r(t)|; \\ HD_{c_l}(t-2) &= |c(t-2) - c_l(t-2)|; \\ HD_{c_r}(t-2) &= |c(t-2) - c_r(t-2)|; \end{aligned}$$

分别计算 a, b, c 在 t 与 $t-2$ 场, $t-1$ 与 $t+1$ 场的水平总梯度:

$$HD_a(t, t-2) = |HD_{a_l}(t) - HD_{a_l}(t-2)| + |HD_{a_r}(t) - HD_{a_r}(t-2)| \quad (3)$$

$$HD_b(t, t-2) = |HD_{b_l}(t) - HD_{b_l}(t-2)| + |HD_{b_r}(t) - HD_{b_r}(t-2)| \quad (4)$$

$$HD_c(t, t-2) = |HD_{c_l}(t) - HD_{c_l}(t-2)| + |HD_{c_r}(t) - HD_{c_r}(t-2)| \quad (5)$$

最大空间水平灰度梯度为

$$HD_{\max} = \max 3(HD_a(t, t-2), HD_b(t, t-2), HD_c(t, t-2)) \quad (6)$$

3) 空间垂直灰度梯度

$$VD_{d_l}(t-1) = |c_l(t-1) - d_l(t-1)|;$$

$$VD_{e_l}(t-1) = |c_l(t-1) - e_l(t-1)|;$$

$$VD_{d_l}(t+1) = |c_l(t+1) - d_l(t+1)|;$$

$$VD_{e_l}(t+1) = |c_l(t+1) - e_l(t+1)|;$$

$$VD_d(t-1) = |c(t-1) - d(t-1)|;$$

$$VD_e(t-1) = |c(t-1) - e(t-1)|;$$

$$VD_d(t+1) = |c(t+1) - d(t+1)|;$$

$$VD_e(t+1) = |c(t+1) - e(t+1)|;$$

$$VD_{d_r}(t-1) = |c_r(t-1) - d_r(t-1)|;$$

$$VD_{e_r}(t-1) = |c_r(t-1) - e_r(t-1)|;$$

$$VD_{d_r}(t+1) = |c_r(t+1) - d_r(t+1)|;$$

$$VD_{e_r}(t+1) = |c_r(t+1) - e_r(t+1)|;$$

分别计算 c_l, c_r, c 的垂直总梯度:

$$VD_{c_l}(t+1, t-1) = |VD_{d_l}(t+1) - VD_{d_l}(t-1)| + |VD_{e_l}(t+1) - VD_{e_l}(t-1)| \quad (7)$$

$$VD_{c_r}(t+1, t-1) = |VD_{d_r}(t+1) - VD_{d_r}(t-1)| + |VD_{e_r}(t+1) - VD_{e_r}(t-1)| \quad (8)$$

$$VD_c(t+1, t-1) = |VD_d(t+1) - VD_d(t-1)| + |VD_e(t+1) - VD_e(t-1)| \quad (9)$$

最大空间垂直灰度梯度为

$$VD_{\max} = \max 3(VD_{c_l}(t+1, t-1), V_{c_r}(t+1, t-1), V_c(t+1, t-1)) \quad (10)$$

选取式(6)、式(10)、式(14)的最大值为待插值像素 Z 的最大灰度变化

$$M_{\max} = \max 3(D_{\max}, HD_{\max}, VD_{\max}) \quad (11)$$

运动检测是通过 M_{\max} 与预先设置的运动阈值(T)比较实现的。如果 M_{\max} 大于 T ,则认为当前像素 Z 为运动像素,其运动信息 $MV = 1$;否则,认为当前像素为静止像素, $MV = 0$ 。即

$$MV = M_{\max} > T \quad (12)$$

在运动检测的同时,计算像素此 $c(t+1)$ 和

$c(t-1)$ 的绝对差值,作为当前像素的时域差值 TD (即 $D_c(t-1, t+1)$); 计算二者的平均值,作为当前像素的时域均值 PT,输出给后续的时空自适应插值模块。

$$TD = D_c(t-1, t+1) = |c(t-1) - c(t+1)| \quad (13)$$

$$PT = (c(t-1) + c(t+1))/2 \quad (14)$$

2 小角度边缘搜索

固然,在运动自适应的去隔行中,正确地区分运动和静止很重要。但是,对于运动图像中边缘信息的准确检测,并在空间插值滤波器中合理应用边缘信息对提高图像质量也相当关键。如果对边缘检测不准会导致锯齿等图像劣化现象。

基于边缘的行平均(ELA)算法在去隔行中得到广泛应用,它的缺陷是对水平边缘的检测能力很弱,只能检测 45° 、 90° 和 135° 边缘,对于小角度边缘则无能为力^[5, 11]。为此, Kuo 等人提出 $3+3$ taps 的 ELA 算法,并通过比较相邻像素的差值来改善对图像水平边缘的检测^[5], Lee 等人将 $3+3$ taps 的 ELA 算法扩展为 $5+5$ ^[12], Haan 等人则扩展为 $7+7$ taps^[13]。这些改进虽然增强了 ELA 对较小角度边缘的检测能力,但是仍然无法检测接近水平的边缘,即使是 $7+7$ taps 也只能检测到最小 18° 的边缘,而且固定抽头的滤波器设计方法对于硬件电路的设计也不是很好的方案。

为提高水平边缘的检测能力,提出了小角度边缘搜索的改进算法,其搜索窗如图 2 所示。

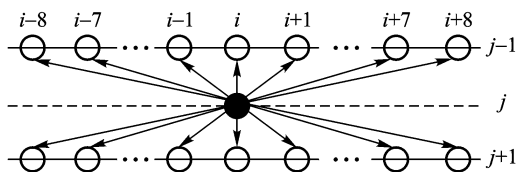


图 2 小角度边缘搜索窗

Fig. 2 Search window of low angle edge detection

图 2 中黑点代表待插补的像素 Z (用 $Y(i, j)$ 表示), 上下两行的像素点的对角线方向代表可能的边缘方向。其基本原理是通过计算、比较对角线上的像素绝对差值,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缘方向。在计算中采用并行处理架构,将上下行的对角像素分为 3 部分:

只有 90° 一个方向的两个像素 $Y(i, j-1)$, $Y(i, j+1)$; **b.** 向左搜索部分,包括 $j-1$ 行上 $Y(i, j-1)$ 右边的像素和 $j+1$ 行上 $Y(i, j+1)$ 左边的像素; **c.** 向右搜索部分,包括 $j-1$ 行上 $Y(i, j-1)$ 左边的像素和 $j+1$ 行上 $Y(i, j+1)$ 右边的像素。同时计算这三部分对角像素的绝对差值,分别得到中间部分绝对差值 D_m , 左边部分绝对差值 L_1, L_2, \dots, L_8 和右边部分绝对差值 R_1, R_2, \dots, R_8 , 即

$$D_m = |Y(i, j-1) - Y(i, j+1)| \quad (15)$$

$$L_n = |Y(i+n, j-1) - Y(i-n, j+1)|, \quad n = (1, 2, \dots, 8) \quad (16)$$

$$R_n = |Y(i-n, j-1) - Y(i+n, j+1)|, \quad n = (1, 2, \dots, 8) \quad (17)$$

其中 $Y(i, j)$ 表示像素 $x(i, j)$ 的亮度分量值,其他同义。

在小角度边缘搜索中,为减少计算量和硬件资源的消耗,采取自适应搜索半径的检测方法,在进行向左或向右顺序搜索的过程中每一级计算结束后都要将差值与上一级进行比较(第一级的差值与 D_m 进行比较),如果绝对差值一直保持增大趋势或减小趋势则继续进行搜索,直到 $\text{tap} = 8$,如果出现相反的变化趋势,则停止搜索。然后比较这些差值选择出最小的绝对差值 D ,此最小差值对应的对角方向就是大致的边缘方向。

$$D = \min(D_m, L_1, \dots, L_n, R_1, \dots, R_n) \quad (18)$$

通过最小绝对差值 D 的选择方法可以检测到 90° , 45° (135°), 26.57° (123.43°), 18.43° (161.57°), 14° (164°), 11.3° (168.7°), 9.46° (170.54°), 8.13° (171.87°), 6.34° (173.66°) 的边缘,检测到的最小边缘为 6.34° 。

为了得到更准确的边缘方向,实现更接近水平方向的检测精度,增加对半像素边缘的搜索。在通过最小绝对差值 D 得到大致边缘方向的基础上,增加对大致边缘方向邻近像素的考虑,如图 3 所示。

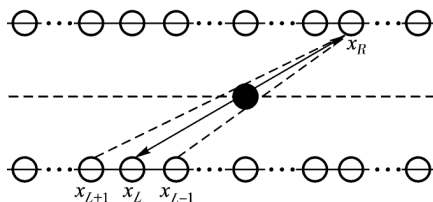


图 3 考虑邻近像素的搜索

Fig. 3 Edge detection considering neighbor pixels

如果初步选定的最小绝对差值 D 对应的方向是 $X_R - X_L$, 则增加对邻近像素 X_{L-1} 和 X_{L+1} 计算的考虑, 分别计算 $X_R - X_{L-1}$ 和 $X_R - X_{L+1}$ 的绝对差值 D_{L-1} 和 D_{L+1} , 求三者的最小值 ND , ND 对应的方向就是边缘的方向。

$$ND = \min(D_{L-1}, D_{L+1}, D) \quad (19)$$

同时, 计算这两个像素的差值为当前像素的空域差值 SD 和它们的平均值为当前像素的空域均值 PS , 输出给时空权重自适应插值模块。

$$SD = \begin{cases} |X_R - X_{L-1}|, & ND = D_{L-1} \\ |X_R - X_{L+1}|, & ND = D_{L+1} \\ |X_R - X_L|, & ND = D_L \end{cases} \quad (20)$$

$$PS = \begin{cases} (X_R + X_{L-1})/2, & ND = D_{L-1} \\ (X_R + X_{L+1})/2, & ND = D_{L+1} \\ (X_R + X_L)/2, & ND = D_L \end{cases} \quad (21)$$

通过增加大致边缘邻近像素的考虑, 可以使检测到的边缘更加接近水平, 比最小绝对差值法增加了 63.4° (116.6°), 33.7° (146.3°), 21.8° (158.2°), 15.9° (164.1°), 12.5° (167.5°), 10.3° (169.7°), 8.7° (171.3°), 7.6° (172.4°), 6.7° (173.3°), 6.0° (174°) 等角度边缘的检测能力, 检测的最小边缘为 6° , 而且将 $6 \sim 90^\circ$ 之间划分得更加细, 从而提高了边缘检测的精度。这一检测精度是已有文献报道所没有达到的。

在所提出的小角度边缘检测的改进中, 主要包括两个方面: **a.** 增大搜索范围(半径), 左右最大各搜索 8 个点, 并且采用自适应的设计方法, 从而减小硬件的复杂度; **b.** 在得到大致边缘方向的基础上, 增加对相邻像素的考虑, 实现半像素精度, 提高边缘检测的精度。

3 时空权重自适应插值

时空权重自适应去隔行的流程图如图 4 所示。

对于 4 场运动估计得到的运动信息 $MV = 0$ 的区域, 即静止区域, 采用场合并 (weave) 的方式进行插值, 提高图像的垂直分辨率, 恢复视频图像原有的帧结构; 对于运动信息 $MV = 1$ 的区域, 即运动区域, 采用时空权重自适应插值, 有效克服运动估计误差和边缘检测不准带来的插值错误, 自动融合场合并和边缘自适应场内插值的优点。

时空权重自适应插值如图 4 虚线框中所示, 主要步骤为:

Step 1 时空权重计算单元利用前后场对应像素

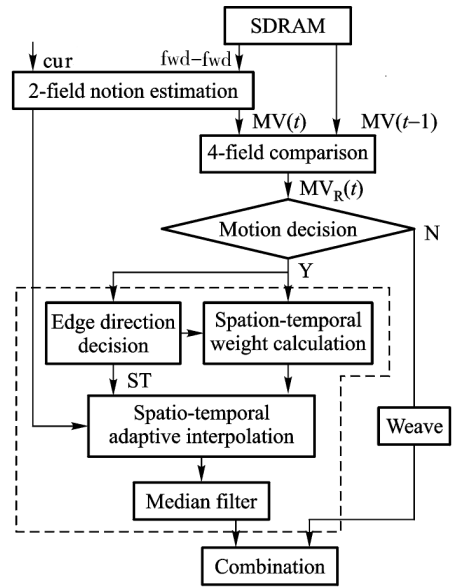


图 4 时空和边缘自适应去隔行流程图

Fig. 4 Flowchart of motion and edge adaptive de-interlace

的绝对差值——时域差值 (TD) 和场内边缘方向的上下行对角像素的差值——空域差值 (SD) 分别计算出时域权重系数 k_T 和空域权重系数 k_S :

$$k_T = SD / (SD + TD) \quad (22)$$

$$k_S = 1 - k_T \quad (23)$$

Step 2 按照 Step 1 得到的权重系数对 PT 和 PS 进行加权平均得到待插补点的估计值

$$P' = PTk_T + PSk_S \quad (24)$$

Step 3 为了消除视频图像中椒盐噪声, 避免插值后的像素值与上下行的像素值相差太过悬殊引起的图像突兀变化和点闪烁现象, 需要对当前场像素点 $Y(i, j+1)$, $Y(i, j-1)$ 和待插补像素点 Z 的估计值 P' 进行中值滤波处理, 得到最终的插补像素

$$P = \text{median}(a(t), b(t), P') \quad (25)$$

4 实验结果及分析

去隔行的实验处理结果比较如图 5 所示。图 5a 为源信号中的偶场图像, 图 5b 为源信号中的奇场图像, 图 5c 传统场内插值去隔行算法得到的处理效果, 图 5d 传统运动自适应去隔行算法得到的处理效果, 图 5e 为所提出的去隔行算法的处理效果。

“钟摆”源图像可以分为钟摆的悬杆、符号“OK”和图像右下角旋转的图标三部分。“OK”可以用来检验运动检测的准确度; 钟摆的悬杆和旋转的图标用于检验小角度边缘的搜索能力和边缘自适应

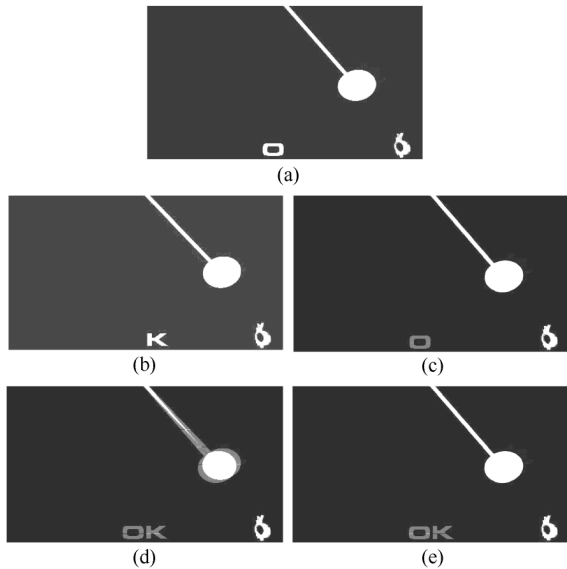


图5 处理效果比较

Fig. 5 Comparison of proposed and typical de-interlace methods

的插值精度。

符号“OK”由只存在于偶场的“O”和只存在于奇场的“K”组成,也就是说,“O”和“K”分别存在于不同极性的场中,所以如果采用相邻场来进行运动检测,将会把此处判断为运动区域而选择场内插值算法,从而丢失“K”或者“O”,如图5e所示的处理结果。正确的处理方法应当是采用同极性场的运动检测,判断此处为静止区域而选择场合并的方法进行插值,效果如图5e所示。

钟摆的悬杆和旋转的图标都是快速运动的区域,需要采取场内插值的算法。但是,如果运动检测不准,将其误判为静止区域,或者在场内插值中小角度边缘的搜索能力和插值精度不足,将大大影响去隔行的效果,如图5d所示。图5d中由于场内插值的失误引起钟摆悬杆出现明显的重影现象。

由图5e可以看出,时空权重和边缘自适应的去隔行算法融合了场合并和场内边缘自适应插值的优点,克服了传统算法的缺陷,取得好的处理效果。有效地消除了重影现象,图像中的边缘更加锐利。

综上所述,时空权重和边缘自适应的去隔行方法克服了传统算法对运动图像和小角度边缘插值等方面的缺陷和不足,取得比较理想的处理效果。此算法和VLSI电路已经被成功地嵌入基于中芯国际0.18 μm CMOS工艺的数字视频扫描格式转换芯片(SFC)中,芯片版图如图6所示。

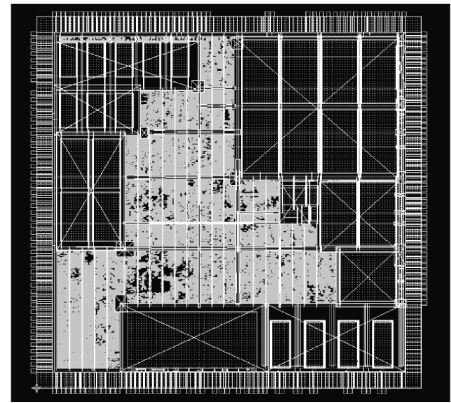


图6 基于0.18 μm工艺的SFC版图

Fig. 6 The layout of SFC based on SMIC 0.18 μm process

5 结语

所提出的一种时空权重和边缘自适应的去隔行方法,通过4场运动估计得到比较精确的运动信息,进行运动自适应插值。对于静止区域采取场合并的插值处理,对于运动区域采取时空权重自适应的插值处理,通过小角度边缘搜索检测最佳空间插值方向。该算法的优点是自动融合场内差值和场合并的优点,电路实现简单,得到的视频图像处理效果好。

参考文献

- [1] Thomas G. A Comparison of Motion-compensated interlace-to-progressive conversion methods [J]. Signal Process: Image Commun, 1998, 12: 209 ~ 229
- [2] Haan G de, Bellers E B. Deinterlacing—an overview [J]. Proceedings of the IEEE, 1998, 86(9): 1839 ~ 1857
- [3] Han D, Shin C Y, Choi S J, et al. A motion adaptive 3-D de-interlacing algorithm based on the brightness profile pattern difference [J]. IEEE Trans Consum Electron, 1999, 45(3): 690 ~ 697
- [4] Lin S F, Chang Yulin, Chen L G. Motion adaptive de-interlacing with horizontal motion detection and ELA with median [A]. Proceedings of the International Symposium on Circuits and Systems [C]. Bangkok Thailand, 2003. 696 ~ 699
- [5] Kuo C J, Liao C, Lin C C. Adaptive interpolation technique for scanning rate conversion [J]. IEEE Trans Circuits Syst Video Technol, 1996, 6(3): 317 ~ 321
- [6] Jung Y Y, Choi B T, Park Y J, et al. An effective de-interlacing technique using motion compensated interpolation [J]. IEEE Trans Consum Electron, 2001, 46(3): 460 ~ 466
- [7] Sugiyama K, Nakamura H. A method of de-interlacing with motion compensation interpolation [J]. IEEE Trans Consum

- Electron, 1999, 45(3): 611 ~ 616
- [8] Li Renxiang, zeng Bing, Liou Mingli. Reliable motion detection/compensation for interlaced sequence and its applications to deinterlacing [J]. IEEE Trans Circuits Syst Video Technol, 2000, 10(1): 23 ~ 29
- [9] Nguyen A, Dubois E. Spatio-temporal adaptive interlaced-to-progressive conversion [A]. Dubois E, Chiariglione L Eds. Signal Processing of HDTV [M]. New York: Elsevier, 1993. 749 ~ 756
- [10] Chang Y L, Lin S F, Chen C Y, et al. Video de-interlacing by adaptive 4-field global/local motion compensated approach [J]. IEEE Transactions on Circuits and System for Video Technology, 2005, 15(12): 1569 ~ 1582
- [11] Lee S G, Lee D H. A motion-adaptive de-interlacing method using an efficient spatial and temporal interpolation [J]. IEEE Trans Consum Electron, 2003, 49(4): 1266 ~ 1271
- [12] Lee H Y, Park J W, Bae T M, et al. Adaptive scan rate up-conversion system based on human visual characteristics [J]. IEEE Trans Consum Electron, 2000, 46(4): 999 ~ 1006
- [13] Haan G de, Lodder R. De-interlacing of video data using motion vectors and edge information [A]. IEEE Int Conf Consum Electron [C]. 2002. 70 ~ 71

A Video Scan Format Conversation Algorithm Based on Spatio-temporal Weight and Edge Direction

Ding Yong, Lu Shengli, Shi Longxing

(National Engineering Research Center for Application Specific Integrated Circuit System, Southeast University, Nanjing 210096, China)

[**Abstract**] De-interlace is to convert interlaced images to progressive ones. In this paper, a de-interlacing algorithm based on spatio-temporal weight and edge direction is presented. It consists of motion detection, low-angle edge detection, and spatio-temporal weight adaptive interpolation. It uses total 4 fields information captured in field storage to detect the presence of motion. And it detects low-angle edge by an adaptive searching radius in which the 6° edge is obtained. The proposed method using spatio-temporal adaptive interpolation can obtain quite good display quality and satisfies real-time process of HDTV sequences.

[**Key words**] scan format conversation; motion estimation; edge detection; spatio-temporal adaptive interpolation; de-interlace