

文章编号: 1001-0920(2012)07-1087-05

最优对称性的快速自适应环路滤波算法

李宏伟, 吴成柯, 张捷, 宋锐

(西安电子科技大学 综合业务网理论及关键技术国家重点实验室, 西安 710071)

摘要: 为了进一步提高基于块与四叉树的自适应环路滤波算法(BQ_ALF)的编码性能, 提出了最优对称性的快速自适应环路滤波算法(FALF_OS). 首先根据区域对称性和平均绝对差值门限确定最优的对称性滤波器; 然后在I帧中按照基于块与四叉树的方法得到需要滤波的区域, 在P或B帧中利用运动矢量、运动估计和率失真模型A(RDO_A)得到需要滤波的区域; 最后采用得到的最优对称性滤波器对这些区域进行滤波. 仿真结果表明, 与BQ_ALF算法相比, 所提出的算法在保证重建图像质量的同时, 编码时间大幅度下降.

关键词: 对称性; 环路滤波; 运动矢量; 率失真

中图分类号: TN919.81

文献标识码: A

Fast adaptive loop filter algorithm with optimal symmetry

LI Hong-wei, WU Cheng-ke, ZHANG Jie, SONG Rui

(The State Key Laboratory of Integrated Services Networks, Xidian University, Xi'an 710071, China. Correspondent: LI Hong-wei, E-mail: lhw3000@163.com)

Abstract: In order to further improve the coding performance of block-based and quadtree-based adaptive loop filter(BQ_ALF) algorithm, fast adaptive loop filter algorithm with the optimal symmetry(FALF_OS) is proposed. Firstly, this algorithm determines the optimal symmetry filter according to area symmetry and average sum of absolute difference. Then the filter areas are obtained by using the block-based and quadtree-based method in I frame, and the filter areas are obtained through motion vector, motion estimation and rate distortion optimization model A(RDO_A) in P or B frame. Finally, the filter areas are filtered by the optimal symmetry filter. Simulation results show that compared with BQ_ALF algorithm, the proposed algorithm reduces the coding time greatly, mean while, the reconstructed picture quality it kept.

Key words: symmetry; loop filter; motion vector; RDO

1 引言

在视频编码过程中, 为了消除图像的块效应, 视频编码标准H.264^[1]采用了去块效应滤波算法^[2-3]. 但是, 该算法没有考虑到原始图像和重建图像的相关性, 只是利用宏块类型和运动矢量差值等参数消除重建图像的块效应, 不能进一步提高视频编码质量. 针对这一问题, 在下一代视频编码标准H.265的提案中, 文献[4,5]提出了基于块与四叉树的自适应环路滤波算法(BQ_ALF), 该算法作为整个编码框架中的一个新部分而用于去块效应滤波之后.

BQ_ALF算法首先通过原始图像和去块效应滤波后的图像统计出 5×5 tap维纳滤波器, 并采用该滤波器对去块效应滤波后的图像进行维纳滤波, 同时将维纳滤波后图像与原始图像的绝对差值和(SAD)作

为率失真代价; 其次根据该滤波器和SAD按照基于块与四叉树的方法计算出最优的滤波区域; 再分别重新统计出 7×7 tap维纳滤波器和 9×9 tap维纳滤波器, 并根据率失真模型A(RDO_A)选出最优大小的维纳滤波器; 最后采用该滤波器对需要滤波的区域进行滤波. 因为BQ_ALF算法运算量过大, 导致编码时间过长; 又因为BQ_ALF算法没有考虑到滤波器的对称性, 使编码质量不能进一步提高^[6-7].

针对以上问题, 本文提出了最优对称性的快速自适应环路滤波算法(FALF_OS). 该算法首先根据区域对称性和平均绝对差值门限确定最优的对称性滤波器; 然后通过快速算法确定需要滤波的区域; 最后采用得到的最优对称性滤波器对这些区域进行滤波. 相对于BQ_ALF算法, 本文算法在保证重建图像质量的

收稿日期: 2010-12-04; 修回日期: 2011-03-07.

基金项目: 国家自然科学基金项目(60802076); 高等学校创新引智计划项目(B08038); 中央高校基本科研业务费专项基金项目(72115046).

作者简介: 李宏伟(1982-), 男, 博士, 从事视频编码与图像处理的研究; 吴成柯(1938-), 男, 教授, 博士生导师, 从事视频与图像编码等研究.

同时,大幅度减少了编码时间。

2 BQ_ALF 算法

2.1 5×5tap 维纳滤波器的计算

根据原始图像和去块效应滤波后的图像,按照维纳滤波的原理统计出最优的5×5tap 维纳滤波器,并且为了减少传输的比特数,该滤波器采用45°对称结构.利用该滤波器对去块效应滤波后的图像中每个像素进行滤波,以得到与原始图像相对应的重建图像。

2.2 基于块与二叉树方法的滤波区域的确定

1) 如图(1)所示,按光栅扫描顺序遍历整个图像中所有的基础块,每个基础块采用自顶向底的四叉树形结构.根据率失真模型A^[8](RDO_A),计算出需要滤波的区域,其中四叉树形结构最小遍历块为4×4块. RDO_A为

$$J_f(D) = \text{SAD}_f(D) + \frac{\text{SIZE}_D}{\text{SIZE}_{\text{frame}}} \lambda_{\text{MODE}} R_{\text{coef}}. \quad (1)$$

式中: $J_f(D)$ 为采用5×5tap 维纳滤波器对 D 块进行滤波后的率失真代价, $\text{SAD}_f(D)$ 为采用5×5tap 维纳滤波器对 D 块进行滤波后的重建图像与原始图像的绝对差值和, R_{coef} 为滤波器所占的比特数, λ_{MODE} 为率失真参数, SIZE_D 和 $\text{SIZE}_{\text{frame}}$ 分别为 D 块的大小和整幅图像的大小. 如果 $J_f(D)$ 小于 $\text{SAD}_{nf}(D)$, 则进行滤波, 否则不进行滤波, 其中 $\text{SAD}_{nf}(D)$ 为 D 块的去块效应滤波后的图像与原始图像的绝对差之和。

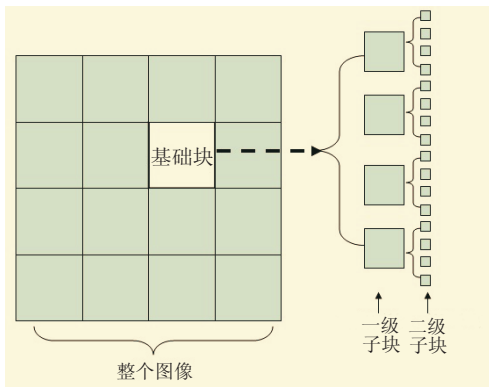


图1 基于块与二叉树的方法

2) 采用128×128, 96×96, 64×64, 48×48, 32×32, 24×24, 16×16, 8×8, 8种不同大小的块作为基础块,按照上述方法依次确定一幅图像中需要滤波的区域,并根据已得到的5×5tap 维纳滤波器对这8种不同区域分别进行滤波,最终计算出8种5×5tap 维纳滤波后重建图像与原始图像的SAD,并选用SAD最小的基础块所确定的滤波区域作为该帧需要滤波的区域。

2.3 最优大小的滤波器的确定

首先,根据原始图像和去块效应滤波后的图像重新统计出45°对称的7×7tap和9×9tap两种维纳滤波器,再加上已确定的45°对称的5×5tap 维纳滤

波器,共有3种滤波器;然后,利用这3种滤波器对该帧需要滤波的区域进行滤波;最后,按率失真模型B(RDO_B)分别计算其代价, RDO_B为

$$J_f(N_{\text{tap}}) = \text{SAD}_f(N_{\text{tap}}) + \lambda_{\text{MODE}} R_{\text{coef}}(N_{\text{tap}}). \quad (2)$$

式中: N 为滤波器的大小, $J_f(N_{\text{tap}})$ 为以 $N \times N_{\text{tap}}$ 滤波器对图像中需要滤波的区域进行滤波后的率失真代价, $\text{SAD}_f(N_{\text{tap}})$ 为以 $N \times N_{\text{tap}}$ 滤波器对图像中需要滤波的区域进行滤波后重建图像与原始图像的绝对差值和, $R_{\text{coef}}(N_{\text{tap}})$ 为 $N \times N_{\text{tap}}$ 滤波器所占的比特数. 最终在 $J_f(5_{\text{tap}})$, $J_f(7_{\text{tap}})$ 与 $J_f(9_{\text{tap}})$ 中选出代价最小的滤波器作为最终大小的维纳滤波器,并采用该滤波器对需要滤波的区域进行滤波。

3 FALF_OS 算法

BQ_ALF 算法采用了基于块与二叉树的方法确定了需要滤波的区域,并利用 RDO_B 确定的最优大小的维纳滤波器对这些区域进行滤波,使重建图像质量有所提高.但是该算法有以下缺点:

1) 没有考虑到图像对称性,对图像中所有像素都采用45°对称滤波器,从而使重建图像质量不能进一步提高。

2) 没有考虑到帧间相关性,对每帧图像重新算一次滤波参数,并且在确定基础块大小和滤波区域的过程中作了大量的率失真比较,使编码时间大幅度增加。

针对以上问题,本文提出了 FALF_OS 算法.对于 I 帧,该算法首先通过区域对称性预测出滤波器的最优对称性,并统计出3种不同大小的滤波器;然后通过平均图像绝对差值(SAD_{ave})门限确定最终的滤波器;最后采用该滤波器根据 BQ_ALF 中基于块与二叉树的方法确定需要滤波的区域,并对其进行滤波;对于非 I 帧,首先根据 SAD_{ave} 门限和在 I 帧中得到3种不同大小的滤波器确定所需的滤波器;然后采用该滤波器根据运动矢量、运动估计和 RDO_A 确定需要滤波的区域,并对其进行滤波.具体步骤如下:

Step 1 为了进一步提高视频编码质量,选择最优的对称性滤波器.遍历 I 帧图像的每个像素,按5×5块统计出4种对称系数。

垂直对称系数为

$$C_{\text{hor}} = \sum_{x=3}^{X-3} \sum_{y=3}^{Y-3} \sum_{i=x}^{x+1} \sum_{j=y}^{y+4} |P_{i,j} - P_{4-i,j}|; \quad (3)$$

水平对称系数为

$$C_{\text{ver}} = \sum_{x=3}^{X-3} \sum_{y=3}^{Y-3} \sum_{i=x}^{x+4} \sum_{j=y}^{y+1} |P_{i,j} - P_{i,4-j}|; \quad (4)$$

45° 对称系数为

$$C_{45} = \sum_{x=3}^{X-3} \sum_{y=3}^{Y-3} \sum_{i=x}^{x+1} \sum_{j=y}^{y+1} |P_{i,j} - P_{4-i,4-j}|; \quad (5)$$

135° 对称系数为

$$C_{135} = \sum_{x=3}^{X-3} \sum_{y=3}^{Y-3} \sum_{i=x}^{x+1} \sum_{j=y}^{y+1} |P_{i,4-j} - P_{4-i,j}|. \quad (6)$$

式中: P 为像素值, X 和 Y 为图像的高和宽, x 和 y 为图像的行列坐标. 选用对称系数最小的对称方式作为滤波器的对称结构, 这种对称结构在每个 GOP 的 I 帧被更新.

Step 2 根据已确定的滤波对称结构, 对 I 帧的原始图像和去块效应滤波后的图像进行统计, 按照维纳滤波的原理^[9]计算出 5×5 tap, 7×7 tap 和 9×9 tap 三种不同大小的维纳滤波器, 以备当前 GOP 中的 I 帧和非 I 帧在其中选择最优的滤波器进行编解码. 维纳滤波器的具体计算方法为: 采用 $N \times N$ 的矩阵, 根据维纳滤波器的计算公式, 计算出图像中以每个像素为中心点的 $N \times N$ 维纳滤波器; 然后对这些滤波器进行加权平均得到最终的 $N \times N$ 自适应维纳滤波器, 维纳滤波器的计算公式为

$$H = R_{xx}^{-1} R_{xd}. \quad (7)$$

式中: H 为维纳滤波器, R_{xx} 为原始图像的自相关矩阵, R_{xd} 为原始图像和去块效应滤波后的图像的互相关矩阵. 这 3 种不同大小的维纳滤波器在每个 GOP 的 I 帧被重新计算.

Step 3 为了快速在 3 种不同大小的滤波器中选择出最优的滤波器, 需对 BQ-ALF 算法进行实验. 如图 2 所示, 对 3 个运动剧烈程度不同的图像序列 foreman, coastguard 和 football 进行测试, 采样 100 帧的 SAD_{ave} , 发现 SAD_{ave} 越大, 则需要的滤波器越大.

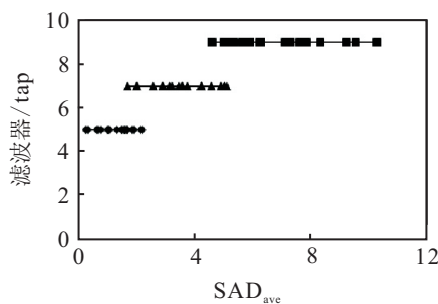


图 2 BQ-ALF 算法中不同 SAD_{ave} 下的最优大小的滤波器

可以根据 SAD_{ave} 门限代替 RDO_B 去预测最优大小的滤波器, 当 $SAD_{ave} > Th1$ 时, 选用 9×9 tap 滤波器; 当 $SAD_{ave} < Th2$ 时, 选用 5×5 tap 滤波器; 否则选用 7×7 tap 滤波器. 根据上述实验结果, 设置经验值 $Th1 = 5$, $Th2 = 2$.

Step 4 采用以上步骤得到的最优的滤波器, 根据 BQ-ALF 中基于块与二叉树的方法, 确定 I 帧中需要滤波的区域, 并对这些区域进行滤波.

Step 5 因为 1 个 GOP 中的图像具有较强的帧间相关性, 所以对于非 I 帧, 首先通过 Step 3 的方法确

定对于当前帧的最优滤波器的大小; 然后在 I 帧中已确定的 3 种不同大小的滤波器中选择同样大小的滤波器作为该帧的滤波器; 同时, 将该滤波器的索引发送到解码端, 以用于对该帧进行解码.

Step 6 对于非 I 帧中的 B, P 宏块中的每个子块, 通过运动估计和运动矢量确定最优参考帧和该子块在最优参考帧中对应的区域, 如果这个区域被滤波, 则对该子块进行滤波; 否则不进行滤波. 当该子块在参考帧中跨越滤波区域和非滤波区域时, 根据 RDO_A 确定该子块的失真代价 $J_f(D)$, 如果 $J_f(D) > SAD_{nf}(D)$, 则进行滤波; 否则不进行滤波. 该方法得到的 P 帧, 也作为参考帧, 且 P 帧的滤波区域与 I 帧一样, 被非 I 帧中的 B, P 宏块中的每个子块所参考.

Step 7 对于非 I 帧中的 I 宏块中的每个子块, 根据 RDO_A 确定该子块的失真代价 $J_f(D)$, 如果 $J_f(D) > SAD_{nf}(D)$, 则进行滤波; 否则不进行滤波.

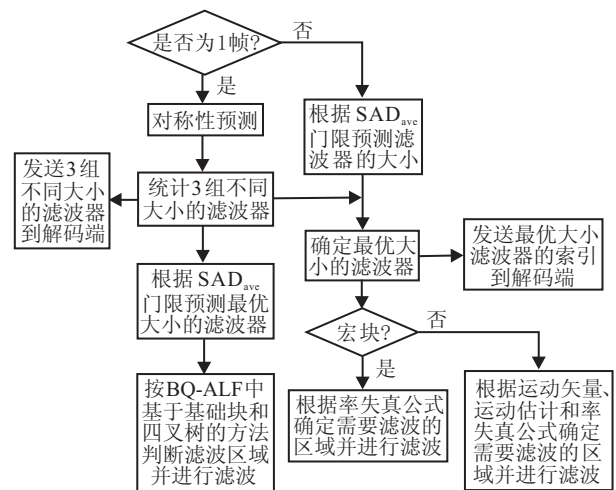


图 3 本文算法流程图

图 3 为 FALF_OS 算法的流程图, 该算法通过 Step 1 ~ Step 4 对 I 帧进行滤波; 通过 Step 5 ~ Step 7 对非 I 帧进行滤波. 相对于 BQ-ALF 算法, FALF_OS 算法在 I 帧中, 先通过选择最优的对称性滤波器进一步提高了视频编码质量, 再通过预测滤波器的大小减少了编码时间; 在非 I 帧中先进行最优滤波器的预测选择, 然后根据运动矢量、运动估计和 RDO_A 确定需要滤波的区域, 并对这些区域进行滤波, 在保证重建图像质量的情况下, 大幅度地减少了编码时间. 对于 FALF_OS 算法的鲁棒性, 因为 1 个 GOP 中的图像具有较强的帧间相关性, 所以非 I 帧可以在上一个临近 I 帧所确定的 3 种对称性滤波器中作选择; 而对于非 I 帧的 3 种不同大小的滤波器的快速选择, SAD_{ave} 门限是根据 3 种运动程度不同的测试序列得到的最优值. 因此对于任何序列, FALF_OS 算法的鲁棒性都得以保留, 并提高了编码性能.

4 实验结果与分析

为了测试本文算法的性能,在KTA2.6^[10]上作2种不同的仿真验证.

1) 不同量化参数(QP)的编码性能比较.

在固定QP下比较H.264算法, H.264 + BQ_ALF算法和H.264 + 本文算法的峰值信噪比(PSNR)、编码时间和比特率. 实验中采取标准的QCIF测试序列, H.264基本层算法, 帧率30 fps, GOP为15, 共编码100帧. 编码时间、PSNR和比特率对比的实验结果如表1所示, 其中PSNR值增量 ΔPSNR , 编码时间增量 ΔT 和比特率增量 ΔB 的计算如下:

$$\Delta\text{PSNR} = \text{PSNR}_{\text{other}} - \text{PSNR}_{\text{H.264}}, \quad (8)$$

$$\Delta T = (T_{\text{other}}/T_{\text{H.264}} - 100\%) \times 100\%, \quad (9)$$

$$\Delta B = (B_{\text{other}}/B_{\text{H.264}} - 100\%) \times 100\%. \quad (10)$$

式中: $\text{PSNR}_{\text{other}}$ 为H.264 + 本文算法的PSNR或H.264 + BQ_ALF算法的PSNR, $\text{PSNR}_{\text{H.264}}$ 为H.264的PSNR; T_{other} 为H.264 + 本文算法的编码时间或H.264 + BQ_ALF算法编码时间, $T_{\text{H.264}}$ 为H.264的编码时间; B_{other} 为H.264 + 本文算法的比特率或H.264 + BQ_ALF算法的比特率, $B_{\text{H.264}}$ 为H.264的比特率.

表1 不同QP的编码性能比较

序列名称	QP	H.264 + BQ_ALF算法			H.264 + 本文算法		
		$\Delta\text{PSNR}/\text{dB}$	$\Delta B/\%$	$\Delta T/\%$	$\Delta\text{PSNR}/\text{dB}$	$\Delta B/\%$	$\Delta T/\%$
fore-	20	0.09	0.29	10.07	0.32	0.18	1.08
man	30	0.38	1.90	11.95	0.19	0.43	1.35
(IPPP)	40	0.31	2.65	8.45	0.20	0.15	1.51
coast-	20	0.12	0.31	8.98	0.31	0.04	0.92
guard	30	0.10	0.42	10.24	0.20	0.35	1.15
(IPPP)	40	0.06	1.76	7.19	0.18	0.06	1.17
foot-	20	0.12	0.9	10.2	0.32	0.2	3.7
ball	30	0.21	1.2	9.8	0.19	0.1	3.1
(IPPP)	40	0.18	1.2	8.6	0.18	0	4.7
fore-	20	0.08	0.29	12.10	0.28	0.45	2.50
man	30	0.31	1.55	11.46	0.19	0.94	2.66
(IBBP)	40	0.25	1.44	8.79	0.23	2.54	2.71
coast-	20	0.09	0.06	12.29	0.29	-0.13	1.85
guard	30	0.15	0.98	10.39	0.21	0.60	2.20
(IBBP)	40	0.08	0.99	7.88	0.2	1.13	1.84
foot-	20	0.06	0.08	9.62	0.32	0.64	3.49
ball	30	0.12	0.63	11.22	0.18	1.10	2.86
(IBBP)	40	0.08	0.08	8.86	0.17	1.82	2.17
平均		0.15	0.93	9.89	0.23	0.59	2.27

2) 不同比特率下的性能比较.

对于H.264算法, H.264 + BQ_ALF算法和H.264 + 本文算法, 测试foreman, coastguard和football三个标准的QCIF视频序列在不同比特率下的PSNR值. 如图4所示, 实验中采用H.264基本层控制, 帧率30 fps, GOP长度为15帧, 编码100帧.

从实验结果可以看出, 相对于H.264算法, H.264 + BQ_ALF算法虽然编码质量有所提高, 但是平均编

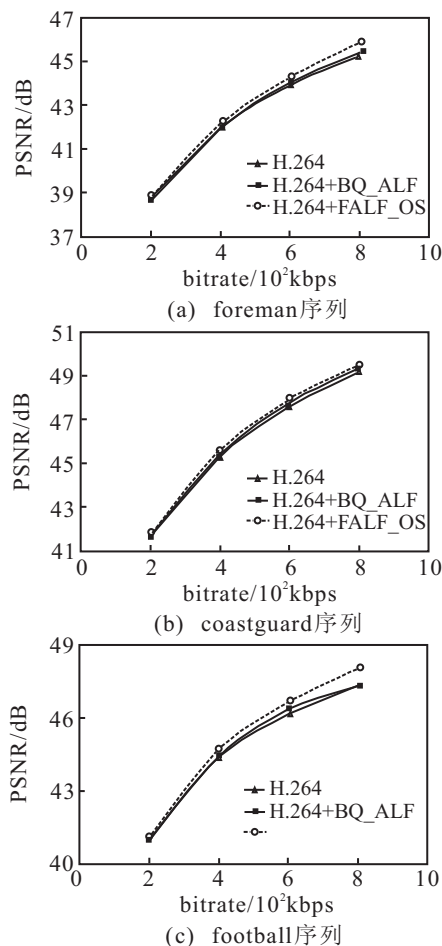


图4 不同比特率下的PSNR值比较

码时间却增加了9.89%. H.264 + 本文算法, 在I帧中, 先预测选取最优的对称性滤波器, 以提高编码质量; 再根据 SAD_{ave} 的门限提前确定滤波器的大小, 从而缩短了编码时间. 在非I帧中, 先通过 SAD_{ave} 的门限和已得到的3种不同大小的滤波器快速选取最优的滤波器; 然后通过运动矢量、运动估计和率失真模型快速而准确地确定需要滤波的区域. 对于本文算法的鲁棒性, 最终相对于H.264 + BQ_ALF算法, H.264 + 本文算法在保证编码质量的情况下, 大幅度缩短了编码时间.

5 结论

通过对BQ_ALF算法的分析发现, BQ_ALF算法首先对图像中所有像素都采用 45° 对称滤波器, 从而使重建图像质量不能进一步提高; 其次对每帧图像重新算一次滤波参数, 并且在确定基础块大小和滤波区域的过程中作了大量的率失真比较, 使编码时间大幅度增加. 针对这一问题, 本文提出了FALF_OS算法. 该算法在根据区域对称性和 SAD_{ave} 门限得到最优的对称性滤波器之后, 在I帧中按照基于块与四叉树的方法确定需要滤波的区域, 在非I帧中利用运动矢量、运动估计和 RDO_A 确定需要滤波的区域; 最终采用得到的最优对称性滤波器对这些区域进行滤波.

经过仿真实验, 相对于BQ-ALF算法, 本文算法在保证重建图像质量的同时, 编码时间大幅度减少。

参考文献(References)

- [1] Richardson I E G. H.264 and MPEG-4 video compression[M]. 2004: 159-164.
- [2] ITU-T Rec. H.264 ISO/IEC 14496-10: 2003, Draft ITU-T recommendation and final draft international standard of joint video specification[S].
- [3] List P, Joch A, Lainema J, et al. Adaptive deblocking filter[J]. IEEE Trans on Circuits and Systems for Video Technology, 2003, 7(13): 614-619.
- [4] Takeshi C, Goki Y, Naofumi W, et al. Block-based adaptive loop filter[DB/OL]. (2008-07-17)[2009-12-1]. <http://wftp3.itu.int/av-arch/video-site/0807-Ber/VCEG-AI18.zip>.
- [5] Takeshi C, Naofumi W, Takashi W, et al. Specification and experimental results of quadtree-based adaptive loop filter[DB/OL]. (2009-05-02)[2009-12-1]. <http://wftp3.itu.int/av-arch/video-site/0904-Yok/VCEG-AK22.zip>.
- [6] Takeshi C, Goki Y, Naofumi W, et al. Improvement of block-based adaptive loop filter[DB/OL]. (2008-10-10)[2009-12-1]. <http://wftp3.itu.int/av-arch/video-site/0810-San/VCEG-AJ13.zip>.
- [7] Takeshi C, Naofumi W, Goki Y. Quadtree-based adaptive loop filter[DB/OL]. (2009-10-26)[2009-12-1]. <http://www.itu.int/md/T09-SG16-C-0324/en>.
- [8] Shukla R, Dragotti P L, Do M N, et al. Rate-distortion optimized tree structured compression algorithms for piecewise smooth images[J]. IEEE Trans on Image Proceedings, 2005, 5(14): 343-359.
- [9] Chiu Y J, Xu L. Adaptive (wiener) filter for video compression[DB/OL]. (2008-4-22)[2009-12-1]. <http://www.itu.int/md/T05-SG16-C-0437/en>.
- [10] JVT. H.264 reference software KTA2.6[DB/OL]. (2009-11-12) [2009-12-1]. <http://iphome.hhi.de/suehring/tml/download/KTA/jm11.0kta2.6r1.zip>.
-
- (上接第1081页)
- [7] Arumugam M S, Rao M V C. On the improved performances of the particle swarm optimization algorithms with adaptive parameters, cross-over operators and root mean square(RMS) variants for computing optimal control of a class of hybrid systems[J]. Applied Soft Computing, 2008, 8(1): 324-336.
- [8] Coelho L S. A quantum particle swarm optimizer with chaotic mutation operator[J]. Chaos, Solitons & Fractals, 2008, 37(5): 1409-1418.
- [9] Tavakkoli M R, Azarkish M, Sadeghnejad B A. A new hybrid multi-objective pareto archive PSO algorithm for a bi-objective job shop scheduling problem[J]. Expert Systems with Applications, 2011, 38(9): 10812-10821.
- [10] Eberhart R C, Shi Y H. Evolving artificial neural networks[C]. Proc of the Int Conf on Neural Networks and Brain. Beijing, 1998: 5-13.
- [11] Afentakis P, Gavish B. Optimal lot-sizing algorithms for complex product structures[J]. Operations Research, 1986, 34(2): 237-249.
- [12] Kennedy J, Eberhart R C. A discrete binary version of the particle swarm algorithm[C]. Proc of the World Multiconference on Systemics, Cybernetics and Informatics. Piscataway, 1997: 4104-4109.
- [13] Shi Y, Eberhart R. Parameter selection in particle swarm optimization[C]. Proc of the 7th Annual Conf on Evolutionary Programming. Washington DC, 1998: 591-600.
- [14] García V A, Pastor R. Introducing dynamic diversity into a discrete particle swarm optimization[J]. Computers & Operations Research, 2009, 36(3): 951-966.