

多对称性快速自适应插值滤波算法

李宏伟, 张捷, 吴成柯, 宋锐

(西安电子科技大学 ISN 国家重点实验室, 西安 710071)

摘要:提出了一种多对称性快速自适应插值滤波算法。该算法首先根据区域对称性,在多种的对称性插值滤波器中做出了初始选择,然后按照简化率失真模型计算出一种最优的对称性插值滤波器,最终大幅度地提高了编码性能。仿真结果表明:与复杂对称性自适应插值滤波算法相比,该算法在大幅度减少控制编码时间的同时,进一步提高了重建图像的峰值信噪比值,并降低了输出码率。

关键词:通信技术;多对称性;自适应;插值滤波器;率失真

中图分类号: TN919.81 **文献标志码:** A **文章编号:** 1671-5497(2011)05-1485-06

Fast multi-symmetry adaptive interpolation filter algorithm

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

(National Key Lab on ISN, Xidian University, Xi'an 710071, China)

Abstract: A Fast Multi-Symmetry Adaptive Interpolation Filter (FMS-AIF) algorithm was proposed. First, an initial subset of symmetry interpolation filters was picked out according to regional symmetric properties. Then, the most optimal symmetry interpolation filter was determined by the rate distortion model. Thus the encoding quality of the proposed algorithm was significantly improved. Simulation results show that compared with Flexible Symmetry Adaptive Interpolation Filter (FS-AIF) algorithm, the coding time is greatly reduced, the Peak Signal-to-Noise Ratio (PSNR) is further increased and the bit rate is reduced in this proposed FMS-AIF algorithm.

Key words: communication; multi-symmetry; adaptive; interpolation filter; rate distortion

在运动估计中,视频编码标准 H. 264 采用了恒定系数的插值滤波器作为标准插值滤波器,使运动搜索的范围确定在 1/4 像素级。因为标准滤波器对所有亚像素采用恒定系数,所以不能使运动估计获得更小的残差信号。针对这一问题,在下一代视频编码标准 H. 265 的提案中,很多方法被提出,如高精度插值滤波算法^[1], 1/8 像素插值滤波算法^[2]。其中以自适应插值滤波算法

AIF^[3-4]效果最为显著。做到了使预测残差达到最小;并减少了需要计算和传输的滤波系数的数量。但是 AIF 算法对所有区域都采用全对称滤波器,导致重建图像质量不能进一步提高。针对这个问题,复杂对称性自适应插值滤波算法 FS-AIF^[5]被提出,该算法采用 5 种不同的对称性滤波器,提高了重建图像质量,但是所花费的代价是运算量大幅增加。为了解决该问题并进一步提高

收稿日期: 2009-10-14.

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);长江学者和创新团队发展计划项目(PCSITR).

作者简介: 李宏伟(1982-),男,博士研究生.研究方向:视频编码与图像处理. E-mail: lhw3000@163.com

重建图像质量,本文提出了多对称性快速自适应插值滤波算法 FMS_AIF。该算法首先通过预处理有效地判定原始图像的对称性,然后根据不同的对称性选择相应的对称性自适应插值滤波器,并对某些滤波器采用简化率失真公式判别,最终选出最优的对称性自适应插值滤波器。

1 AIF 算法与 FS_AIF 算法分析

1.1 标准插值滤波器

如图 1 所示,为获得 1/4 像素级的所有亚像素值,H. 264 采用了标准插值滤波器。定义如下:

(1)对于 b, h 点采用 6-tap 恒定系数滤波器获得。例如

$$b = \{E - 5 \times F + 20 \times G + 20 \times H - 5 \times I + J\}$$

(2)对于 j 点,可以通过已获得的半像素点采用 6-tap 恒定系数滤波器获得。例如

$$j = \{cc - 5 \times dd + 20 \times h + 20 \times m - 5 \times ee + ff\}$$

(3)对于 a, c, d, f, n, l, k, q 点,通过邻近像素的内插而获得。例如 a 用 G 和 b 来获得。

(4)对于 e, g, p, r 点可以通过对角半像素点插值获得。例如 e 用 b 和 h 来获得。

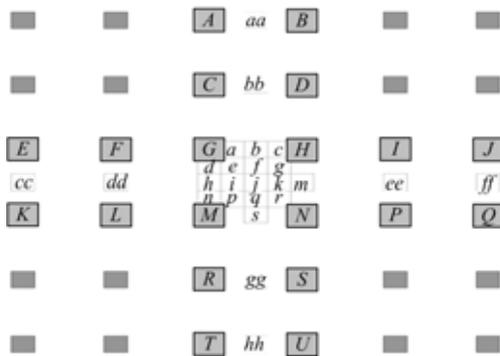


图 1 1/4 像素采样插值

Fig. 1 Quarter-pel sample interpolation

1.2 二维非独立自适应插值滤波器

二维非独立自适应插值滤波器根据预测残差最小化的原则为每一个亚像素点计算滤波系数。预测残差的具体计算公式为

$$(e^{SP})^2 = \sum_x \sum_y (S_{x,y} - \sum_i \sum_j h_{i,j}^{SP} P_{\tilde{x}+i, \tilde{y}+j})^2 \quad (1)$$

式中: $S_{x,y}$ 为原始图像整数像素点; $P_{\tilde{x}, \tilde{y}}$ 为在参考帧中的最佳匹配的整数像素点; $h_{i,j}^{SP}$ 为滤波参数。

$$\tilde{x} = x + \lfloor mvx \rfloor - \frac{filter_size}{2} + 1 \quad (2)$$

$$\tilde{y} = y + \lfloor mvy \rfloor - \frac{filter_size}{2} + 1 \quad (3)$$

为了使预测残差达到最小,计算最优的滤波系数,对 $(e^{SP})^2$ 进行二阶求导。计算公式为

$$\frac{(\partial e^{SP})^2}{\partial h_{k,l}^{SP}} = \frac{\partial}{\partial h_{k,l}^{SP}} \left[\sum_x \sum_y (S_{x,y} - \sum_i \sum_j h_{i,j}^{SP} P_{\tilde{x}+i, \tilde{y}+j})^2 \right] = \sum_x \sum_y (S_{x,y} - \sum_i \sum_j h_{i,j}^{SP} P_{\tilde{x}+i, \tilde{y}+j}) P_{\tilde{x}+k, \tilde{y}+l} = 0 \quad \forall k, l \in \{0;5\} \quad (4)$$

如图 1 所示,对于亚像素 a, b, c, d, h, n , 采用 6-tap 为每个点计算出 6 个滤波系数;对于其他位置亚像素,采用 6×6 -tap 为每个点计算出 36 个滤波系数;最终 15 个亚像素需要 360 个滤波系数。

1.3 全对称插值滤波器

如图 2 所示,假设所有滤波参数都是全对称的,那么 a, c, d, n 四个点的滤波系数是一样的; e, g, p, r 四个点的滤波系数是一样的; f, i, k, q 四个点的滤波系数是一样的; b 和 h 的滤波系数也是一样的。因此只需要计算亚像素点 a, b, e, f, j 的滤波系数就可以得到其他亚像素点的滤波系数。因为亚像素点 a 没有处于对称位置,所以需要计算 6 个滤波系数。而 b, e, f, j 处于对称位置,所以需要的滤波系数相对减少。例如, b 点需要 3 个滤波系数,计算公式为

$$b = h_{31} * E + h_{32} * F + h_{33} * G + h_{33} * H + h_{32} * I + h_{31} * J \quad (5)$$

最终 15 个亚像素需要的滤波系数从 360 个减少到 54 个。

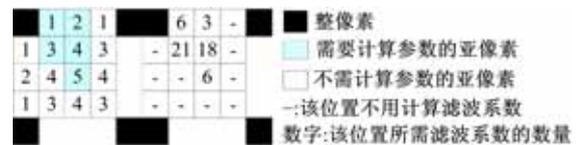


图 2 全对称插值滤波器及其滤波系数数目

Fig. 2 All symmetry interpolation filter and the number of coefficients

AIF 算法的具体步骤如下:

(1)采用 H. 264 的标准插值滤波器在参考图像中进行上采样,在 1/4 像素级找到运动矢量。

(2)按照已得到的运动矢量在最佳匹配位置通过二维非独立自适应插值滤波器和全对称滤波

器计算出亚像素值和滤波系数。

(3)将运动补偿后的预测残差和滤波系数一起传送到解码端。

FS_AIF 算法的具体步骤如下:

(1)采用 H.264 的标准插值滤波器在参考图像中进行上采样,在 1/4 像素级找到运动矢量。

(2)按照已得到的运动矢量在最佳匹配位置,



图 3 5 种对称插值滤波器

Fig. 3 5 kinds of symmetry interpolation filters

2 FMS_AIF 算法

由以上分析可以看出,AIF 算法首先采用了二维非独立自适应插值滤波器使预测残差最小化,又采用了全对称滤波器,缩减了需要计算和传输的滤波系数的数量,最终提高了编码质量。但是,原始图像并不是每个区域都是完全对称的,AIF 算法没有考虑到这点,而直接采用全对称插值滤波器,使得到的预测残差相对于直接使用二维非独立自适应插值滤波器得到的预测残差会有所增大,导致重建图像质量不能进一步提高。

针对以上问题,FS_AIF 算法提出,在以 5 种不同的对称性滤波器分别进行编码之后,通过率失真公式得到最优的对称性滤波器。因为 FS_AIF 算法只有 5 种对称性,不能使对称性滤波器的效果达到最好;又因为采用 5 种对称性滤波器分别进行编码,没有进行预判定,使编码运算量大幅增加。

针对 AIF 算法和 FS_AIF 算法问题,本文提出了 FMS_AIF 算法。该算法通过对称性的初始判定和简化率失真函数的计算,在有效控制运算量的同时,合理地减少了大量的需要计算和传输的滤波系数的数量,提高了编码质量。

算法步骤如下:

(1)采用 H.264 的标准插值滤波器在参考图像中进行上采样,在 1/4 像素级找到运动矢量。

(2)根据 6×6 大小的整像素块预测亚像素对称性, $P_{i,j}$ 为 6×6 整数像素点的值。

垂直对称满足条件为

通过二维非独立自适应插值滤波器和 5 种不同的对称性滤波器分别计算出 5 组亚像素值和滤波系数,如图 3 所示。

(3)将运动补偿后的 5 组预测残差和滤波系数一起传送到解码端。在得到实际 PSNR 值和比特率后,通过率失真函数计算出最优的对称性滤波器,并利用该对称性插值滤波器进行编码。

$$\sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j} - P_{7-i,j}| < \frac{1}{180} \times \sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j}| \tag{6}$$

水平对称满足条件为

$$\sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j} - P_{i,7-j}| < \frac{1}{180} \times \sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j}| \tag{7}$$

45°对称满足条件为

$$\sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j} - P_{7-i,7-j}| < \frac{1}{180} \times \sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j}| \tag{8}$$

135°对称满足条件为

$$\sum_{i=1}^6 \sum_{j=1}^6 |P_{i,7-j} - P_{7-i,j}| < \frac{1}{180} \times \sum_{i=1}^6 \sum_{j=1}^6 |P_{i,j}| \tag{9}$$

根据计算得到的 4 种对称性,在 16 种对称性插值滤波器中选择对应的对称性插值滤波器。图 4 为对应每种对称性插值滤波器所需计算的亚像素位置。图 5 为对应每种对称性每个亚像素所需计算的滤波系数的数目。例如,当计算得到的对称性插值滤波器同时满足水平对称,135°对称和 45°对称时,只需计算亚像素 a、b、e、f、j 的滤波系数,a 点需要计算 6 个滤波系数,b 点需要计算 3 个滤波系数,e 点需要计算 21 个滤波系数,f 点需要计算 18 个滤波系数,j 点需要计算 9 个滤波系数。所以,这种对称性插值滤波器一共需要计算 57 个滤波系数。

(3)如图 5 所示,全对称插值滤波器仅需 54 个滤波系数,采用该插值滤波器可以减少大量的



图 4 16 种对称插值滤波器

Fig. 4 16 kinds of symmetry interpolation filters



图 5 16 种对称性插值滤波器及其滤波系数数目

Fig. 5 16 kind of symmetry interpolation filters and their number of coefficients

附加比特。但是如果预测失误,则可能导致全对称插值滤波器的漏选,从而不能达到最好的编码质量。因此,当计算得到的对称性插值滤波器同时满足 3 种对称性或满足 $135^\circ+45^\circ$ 对称、垂直水平对称时,将这种对称性插值滤波器与全对称插值滤波器按式(4)分别计算出滤波系数,将运动补偿后的预测残差和熵编码后的比特率代入简化率失真函数进行比较,最终选取率失真最小的作为选定的对称插值滤波器。简化率失真计算公式为

$$J(aif) = D(f) + \lambda * (R_{\text{coeff}} + R_{\text{rem}}) \quad (10)$$

式中: $D(f)$ 为所选用滤波器的预测残差; R_{coeff} 为传输滤波器系数所需比特; R_{rem} 为宏块残差的比特。

3 实验结果与分析

为了测试本文算法的性能,在 KTA2.4^[6] 上做 2 种不同的仿真实验验证。实验计算机配置环境为 Pentium42.6 GHz,内存 1 GB,系统 Windows XP2。所有仿真均在此实验环境下进行。

(1) 不同 QP 下的性能比较

首先在不同 QP 下比较 AIF 算法、FS_AIF 算法和 FMS_AIF 算法的编码性能。实验中采用 H.264 基本层控制,选取标准的 QCIF 测试序列,包括 2 个运动剧烈序列 coastguard、football 和 2 个运动平缓序列 carphone、foreman,帧率为 30 帧/s,GOP 长度为 15 帧,IPPP 格式,共编码 100

帧,每行宏块为一个基本单元。实验结果如表 1 所示。其中, $\Delta\text{Bit Rate}$ 和 ΔTime 的计算式为

$$\Delta\text{Bit Rate} = \frac{\text{Bit Rate}_{\text{other}}}{\text{Bit Rate}_{\text{AIF}}} - 100\% \quad (11)$$

$$\Delta\text{Time} = \frac{\text{Time}_{\text{other}}}{\text{Time}_{\text{AIF}}} - 100\% \quad (12)$$

式中: $\text{Bit Rate}_{\text{other}}$ 和 $\text{Time}_{\text{other}}$ 为其他算法的比特率和编码时间; $\text{Bit Rate}_{\text{AIF}}$ 和 Time_{AIF} 为 AIF 算法的比特率和编码时间。

从实验结果可以看出,通过预处理和简化率

失真函数, FMS_AIF 算法比 FS_AIF 算法减少了大量的编码时间;相对于 AIF 算法, FMS_AIF 算法由于采用了 16 种对称性滤波器,所以更加有效合理地减少了需要计算和传输的滤波系数,提高了编码性能。如表 1 所示, FMS_AIF 算法的编码时间相对于 FS_AIF 算法有大量的减少。相对于 AIF 算法,不但平均 PSNR 值有 0.17 dB 的提高,而且平均比特率有 2.94% 的下降。

(2) 不同比特率下的性能比较

如图 6 所示,实验中采用 H. 264 基本层控

表 1 不同 QP 下的性能比较

Table 1 Encoding quality and encoding efficiency under different QP

图像名称	QP	PSNR/dB			$\Delta\text{Bit Rate}$		ΔTime	
		AIF	FS_AIF	FMS_AIF	FS_AIF	FMS_AIF	FS_AIF	FMS_AIF
Coastguard	20	40.75	40.77	40.87	-3.00	-3.79	98.59	5.39
	30	32.93	32.95	33.03	-4.39	-5.32	96.31	4.22
	40	27.05	27.14	27.21	-3.63	-4.60	96.78	4.38
Football	20	41.47	41.49	41.59	-1.14	-2.11	98.67	5.30
	30	33.86	33.90	33.98	-1.36	-2.26	98.03	5.15
	40	27.67	27.72	27.79	-0.50	-1.48	97.51	4.97
Carphone	20	43.05	43.06	43.17	-2.53	-3.36	98.99	5.54
	30	36.12	36.23	36.32	-1.77	-2.65	96.73	4.63
	40	29.69	29.88	29.95	-0.76	-1.01	98.21	5.34
Foreman	20	41.53	41.54	41.65	-3.33	-4.16	98.82	5.64
	30	34.68	34.77	34.86	-2.32	-3.18	97.17	4.61
	40	28.67	28.91	28.98	-0.48	-1.41	96.98	4.65
平均		34.78	34.86	34.95	-2.10	-2.94	97.73	4.98

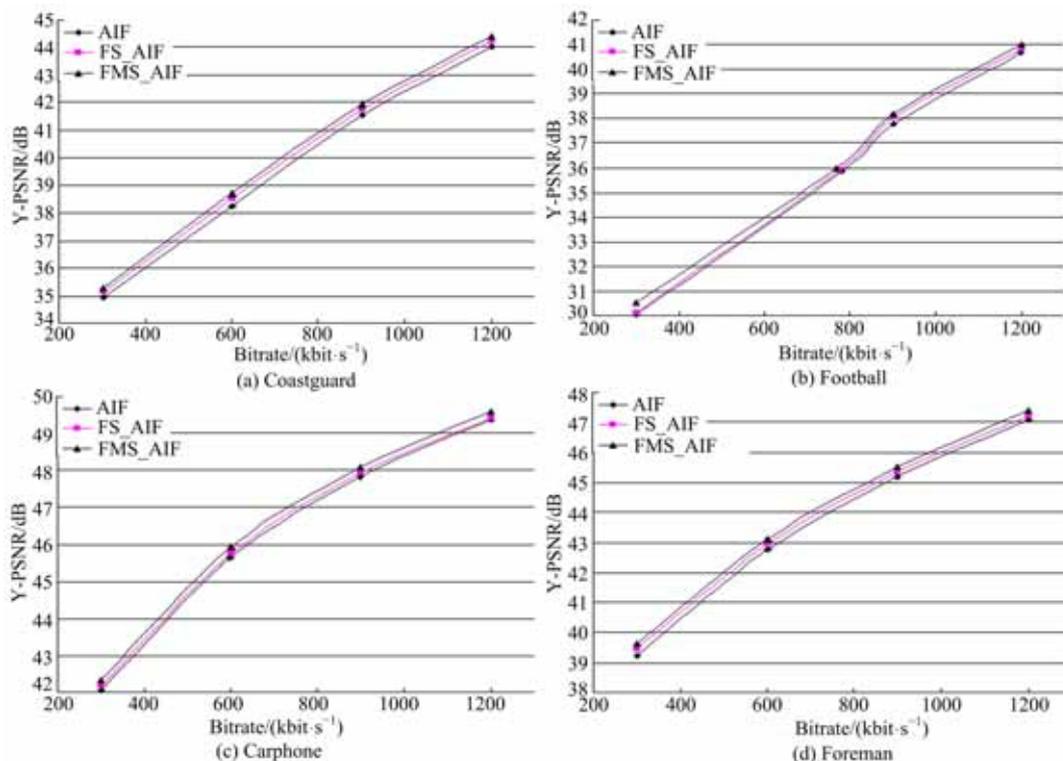


图 6 PSNR-QP 实验结果

Fig. 6 PSNR-QP experiment result

制,帧率为30帧/s,GOP长度为15帧,IPPP格式,编码100帧,每行宏块为一个基本单元。结果表明,在同样比特率情况下,FS_AIF算法的PSNR值优于AIF算法,而FMS_AIF算法的PSNR值又优于FS_AIF算法。

4 结束语

AIF算法没有考虑区域对称性,而直接采用全对称滤波器,不能有效地提高编码图像质量;FS_AIF算法没有预先判定区域对称性,而采用5种对称性滤波器分别编码,导致运算量过大。针对这两个问题,本文提出了FMS_AIF算法。该算法首先通过预测区域对称性,在16种对称性插值滤波器中做了初始的选定,然后根据简化率失真函数确定了最终的对称性自适应插值滤波器,最终大幅度地提高了编码性能。仿真结果表明,与FS_AIF算法相比,FMS_AIF算法的编码时间大幅度减少;与AIF算法相比,FMS_AIF算法使重建图像的PSNR值提高了0.17 dB,输出码率降低了2.94%。

参考文献:

- [1] Ye Yan, Chen P S, Marta K. High precision interpolation and prediction[DB/OL]. [2009-09-20]. http://wftp3.itu.int/av-arch/video-site/0807_Ber/VCEG-AI33.zip.
- [2] Matthias N. Motion compensated prediction with 1/8-pel displacement vector resolution [DB/OL]. [2009-09-20]. http://wftp3.itu.int/av-arch/video-site/0610_Han/VCEG-AD09.zip.
- [3] Yuri V, Bernd E, Dieu T N, et al. Two-dimensional non-separable adaptive wiener interpolation filter for H. 264/AVC[DB/OL]. [2009-09-20]. http://wftp3.itu.int/av-arch/video-site/0504_Bus/VCEG-Z17r1.doc.
- [4] Yuri V, Joern O. Prediction of P-and B-Frames using a two-dimensional non-separable adaptive wiener interpolation filter for H. 264/AVC[DB/OL]. [2009-09-20]. http://wftp3.itu.int/av-arch/video-site/0610_Han/VCEG-AD08r1.zip.
- [5] Kemal U, Jani L, Moncef G. Interpolation filters with different symmetry properties [DB/OL]. [2009-09-20]. http://wftp3.itu.int/av-arch/video-site/0701_Mar/VCEG-AE21.zip.
- [6] H. 264 Reference software. KTA2. 4 [DB/OL]. [2009-09-20]. <http://iphome.hhi.de/suehring/tml/download/KTA/jm11.0kta2.4>.

[1] Ye Yan, Chen P S, Marta K. High precision inter-