

一种改进的低成本自适应双三次插值算法及 VLSI 实现

庞志勇^{1,2} 谭洪舟¹ 陈弟虎²

摘要 提出了一种新型图像缩放算法,由自适应锐化滤波器和双三次插值组成.锐化滤波器减轻了双三次插值产生的模糊效应,自适应技术进一步提升了图像缩放质量.为了减少运算量,提出前置滤波和后置滤波技术.与其他几种算法相比较,本文的算法在主观和客观评价方面都明显胜出.为了实现实时低成本设计,提出了一种该算法的流水线超大规模集成电路(Very large scale integration, VLSI)架构.在现场可编程逻辑器件(Field-programmable gate array, FPGA)上实现,占用 695 个逻辑单元(Logic elements, LEs),时钟频率达到 165 MHz,减少了 36.8% 逻辑单元,图像质量平均峰值信噪比(Peak signal-to-noise ratio, PSNR)提升了 1.5 dB.

关键词 双三次插值, 图像缩放, 拉普拉斯变换, 自适应, 超大规模集成电路, 现场可编程逻辑器件

引用格式 庞志勇, 谭洪舟, 陈弟虎. 一种改进的低成本自适应双三次插值算法及 VLSI 实现. 自动化学报, 2013, 39(4): 407-417

DOI 10.3724/SP.J.1004.2013.00407

An Improved Low-cost Adaptive Bicubic Interpolation Arithmetic and VLSI Implementation

PANG Zhi-Yong^{1,2} TAN Hong-Zhou¹ CHEN Di-Hu²

Abstract A novel scaling algorithm is proposed which consists of a bicubic interpolation and an adaptive sharpening filter. The proposed sharpening filter is added to mitigate the blurring effects existing in bicubic interpolation methods. We also verify the scaling quality by taking into account the adaptive technique. Furthermore, we present both the procedures of filtering before and after interpolation in order to reduce the overall computing time. Compared with the previous reported techniques, our method performs better in terms of both quantitative evaluation and visual quality. To achieve the goal of real time and low cost, we describe a pipelined VLSI architecture for the implementation of the algorithm. The very large scale integration (VLSI) architecture of our image scaling processor contains 695 logic elements (LEs) and yields a processing rate of about 165 MHz by using field-programmable gate array (FPGA) technology. Our proposed architecture reduces the amount of gates by 36.8% while achieves an average peak signal-to-noise ratio (PSNR) increase of 1.5 dB in image quality.

Key words Bicubic interpolation, image scaling, Laplacian transform, adaptive, very large scale integration (VLSI), field-programmable gate array (FPGA)

Citation Zhi-Yong Pang, Hong-Zhou Tan, Di-Hu Chen. An improved low-cost adaptive bicubic interpolation arithmetic and VLSI implementation. *Acta Automatica Sinica*, 2013, 39(4): 407-417

数字图像缩放是指根据原始图像的离散像素对数字图像的大小进行调整的过程.图像缩放在多媒体^[1-2]、公共安全、消费电子^[3]以及医疗图像^[4]等

方面具有广泛应用.随着平板技术的发展,图像缩放更加成为了研究热点,研究人员提出了大量有效的图像算法^[1-21].

针对实时图像处理应用(如移动电话、数码相机等)图像缩放的实现,必须满足计算复杂度低和存储器资源占用少.虽然已经提出很多先进的图像缩放算法^[5-9, 19-21],目前应用最广泛的低复杂度算法仍然是最近邻插值^[10-11]、双线性插值^[12]和双三次插值^[10-13].最近邻插值算法是最简单的,要求最少计算量和存储器访问时间.然而,该方法的模糊效应和块效应较严重,因此研究的吸引力较小.双线性插值算法在计算复杂度和缩放质量方面进行了很好的折中,实现结构较简单,能够节省一定硬件资源.但是,该方法仍然有明显的模糊效应和块效应.双线性插值算法缩放图像的质量要优于最近邻插值算法,但

收稿日期 2012-06-14 录用日期 2012-11-06
Manuscript received June 14, 2012; accepted November 6, 2012
广东省战略新兴产业关键技术产业化专项(2011168014, 2011912004),广东省省部产学研重点科技项目(2011A090200037)资助
Supported by Strategic Emerging Industry Key Technology Special Project of Guangdong Province (2011168014, 2011912004) and the Provincial Ministry of Research Project of Key Science and Technology (2011A090200037)
本文责任编辑 刘成林
Recommended by Associate Editor LIU Cheng-Lin
1. 中山大学信息科学与技术学院 广州 510275 2. 中山大学物理科学与工程技术学院 广州 510275
1. School of Information Science and Technology, Sun Yat-sen University, Guangzhou 510275 2. School of Physics and Engineering, Sun Yat-sen University, Guangzhou 510275
该文的英文版同时发表在 *Acta Automatica Sinica*, vol. 39, no. 4, pp. 407-417, 2013.

比双三次插值效果差. 双三次插值^[13]是一种基于多项式的图像缩放算法, 图像缩放性能较高. 但双三次插值计算复杂度较高, 较难在实时图像缩放中应用, 但常用于对图像缩放质量要求较高时的应用.

目前, 很多研究都集中在图像缩放的算法上, 而在硬件实现方面的成果较少, 只有文献[1, 14–18, 22–24]给出了几种硬件实现的方法. 文献[22, 24]提出了双三次插值的现场可编程逻辑器件(Field-programmable gate array, FPGA)实现方法^[23], 提供了超大规模集成电路(Very large scale integration, VLSI)实现方法, 但都没有考虑双三次插值引起的模糊效应和块效应. 多项式(Polynomial)^[19]、自适应(Adaptive)^[20]、相关性(Correlative property)^[21]等改进方法可以克服这些问题, 但计算复杂度较高, 不适合硬件实现.

本文分别讨论了几种改进方法, 通过组合滤波器和拉普拉斯滤波器减少双三次插值带来的模糊效应和块效应; 采用自适应技术^[1]进一步改善图像缩放质量; 为了减少计算时间, 本文提出了插值前滤波和插值后滤波技术. 与其他方法^[1, 22–24]比较, 本文的方法在主观和客观评价上都有明显提高, 并且本文提出的VLSI实现比文献[24]减少了很多逻辑资源.

本文的组织结构如下: 第1节详细介绍了双三次插值算法; 第2节描述了假设的锐化滤波器和自适应技术; 第3节给出了实验结果, 包括峰值信噪比(Peak signal-to-noise ratio, PSNR)值和缩放效果图; 第4节介绍了本文提出的高性能VLSI架构; 第5节得出结论.

1 双三次插值

双三次插值是三次插值在二维空间上的一种扩展. 三次插值的核是基于三次多项式对理想抽样函数 sinc 在 $[-2, 2]$ 的卷积插值的近似^[13]. 通常, 三次插值基于三阶多项式 (1) 产生插值系数:

$$h_c(s) = \begin{cases} 1 - (c+3)|s|^2 + (c+2)|s|^3, & 0 \leq |s| < 1 \\ -4c + 8c|s| - 5c|s|^2 + c|s|^3, & 1 \leq |s| < 2 \\ 0, & 2 \leq |s| \end{cases} \quad (1)$$

上式中, s 代表插值像素和参考像素之间的距离, c 是一个可调参数, Keys^[13]采用 $c = -0.5$ 取得了最好的效果, 这时式 (1) 简化为

$$h_c(s) = \begin{cases} 1 - 2.5|s|^2 + 1.5|s|^3, & 0 \leq |s| < 1 \\ 2 - 4|s| + 2.5|s|^2 - 0.5|s|^3, & 1 \leq |s| < 2 \\ 0, & 2 \leq |s| \end{cases} \quad (2)$$

双三次插值要求采用 16 个像素, 16 个浮点数操作,

类似窗口处理. 因此, 空间和时间的并行处理技术是双三次插值 VLSI 实现的理想解决方案^[23]. 图 1 表示用 4×4 邻域像素计算插值像素 F 的过程. 双三次插值过程通常分解为水平和垂直两个一维的插值, 这样可以减少计算复杂度. 图 1 显示了双三次插值的过程. 首先进行垂直插值, 根据原像素和虚拟像素的距离插值得到 4 个虚拟像素 ($F_{h1}, F_{h2}, F_{h3}, F_{h4}$); 然后由这 4 个虚拟像素进行水平插值得到插值像素 F . 双三次插值由式 (3) 进行描述, 其中, C_1, C_2, C_3, C_4 是垂直系数, L_1, L_2, L_3, L_4 是水平插值系数, x, y 是插值像素的坐标.

$$\begin{aligned} F_{h1} &= C_1P_1 + C_2P_5 + C_3P_9 + C_4P_{13} \\ F_{h2} &= C_1P_2 + C_2P_6 + C_3P_{10} + C_4P_{14} \\ F_{h3} &= C_1P_3 + C_2P_7 + C_3P_{11} + C_4P_{15} \\ F_{h4} &= C_1P_4 + C_2P_8 + C_3P_{12} + C_4P_{16} \\ F &= L_1F_{h1} + L_2F_{h2} + L_3F_{h3} + L_4F_{h4} \\ F(x, y) &= L_1(C_1P_1 + C_2P_5 + C_3P_9 + C_4P_{13}) + \\ &L_2(C_1P_2 + C_2P_6 + C_3P_{10} + C_4P_{14}) + \\ &L_3(C_1P_3 + C_2P_7 + C_3P_{11} + C_4P_{15}) + \\ &L_4(C_1P_4 + C_2P_8 + C_3P_{12} + C_4P_{16}) \quad (3) \end{aligned}$$

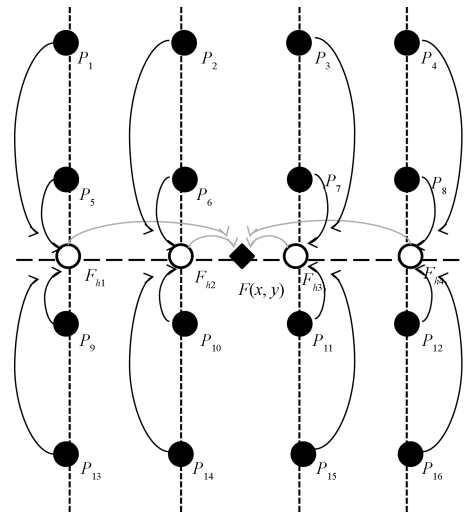


图 1 双三次插值原理图

Fig. 1 Schematic diagram illustrating bicubic interpolation

2 锐化滤波器和自适应技术

本文提出了几种低复杂度技术解决双三次插值带来的高频损失问题, 并且分别进行仿真测试. 锐化滤波器可以增强图像边缘部分或者突出图像的细节, 因此可以减少双三次插值的模糊化问题. 自适应技术^[1]可以进一步提高锐化滤波器改善的效果. 为了减少处理时间, 本文提出了插值前和插值后滤波方法. 第3节给出了这些图像缩放算法的仿真结果.

2.1 组合滤波器

组合滤波器包括一个 3×3 箝位滤波器^[25-26] (Clamp filter) 和一个 3×3 锐化滤波器^[27-28]. 箝位滤波器属于低通滤波器, 作用是平滑不连续边缘和减少块效应. 锐化滤波器属于高通滤波器, 可以减轻块效应. 为了减少存储器资源, 组合滤波器被简化成了一个 5×5 矩阵, 如下所示:

$$P'_{(i,j)} = \frac{P_{(i,j)} \begin{bmatrix} 1 & 1 & 1 \\ 1 & C & 1 \\ 1 & 1 & 1 \end{bmatrix}}{C+8} \times \frac{\begin{bmatrix} -1 & -1 & -1 \\ -1 & S & -1 \\ -1 & -1 & -1 \end{bmatrix}}{S-8} = P_{(i,j)} \times \frac{\begin{bmatrix} -1 & -2 & -3 & -2 & -1 \\ -2 & -2-C+S & -4-C+S & -2-C+S & -2 \\ -3 & -4-C+S & -8+SC & -4-C+S & -3 \\ -2 & -2-C+S & -4-C+S & -2-C+S & -2 \\ -1 & -2 & -3 & -2 & -1 \end{bmatrix}}{(C+8) \times (S-8)} \quad (4)$$

上式中, C 代表箝位参数, S 代表锐化参数. 式 (4) 表示了滤波结果被滤波增益放大, 在这种情况下, 滤波结果应该修改成缩放的平均亮度值, 从而近似原像素的亮度值. 组合滤波器的增益为

$$gain_{\text{Combined sharpening filter}} = (C+8) \times (S-8) \quad (5)$$

2.2 拉普拉斯锐化滤波器

组合滤波器先进行平滑滤波再进行锐化滤波. 由于双三次插值会对缩放图像产生模糊效应, 本文设想等价于平滑滤波. 因此, 本文设想只采用拉普拉斯锐化滤波器进行锐化滤波, 可以进一步提升图像缩放质量.

拉普拉斯锐化滤波器是一种各向同性线性滤波器, 可以增强图像边缘, 但对噪声比较敏感. 拉普拉斯锐化滤波器增强了图像的突变区域同时降低灰度缓慢变换的区域. 为了获得一个锐化的图像 $g(x, y)$, 通常将原始图像 $f(x, y)$ 和拉普拉斯锐化后的图像进行相加. 在实际应用中, 采用一个掩膜进行拉普拉斯锐化滤波和相加运算. 原始图像 $f(x, y)$ 的拉普拉斯变换定义为

$$\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2} \quad (6)$$

上式的离散形式如下:

$$\nabla^2 f = [f(x+1, y) + f(x-1, y) + f(x, y+1) + f(x, y-1) - 4f(x, y)] \quad (7)$$

将原始图像 $f(x, y)$ 和拉普拉斯锐化后的图像进行相加, 得到增强图像 $g(x, y)$

$$g(x, y) = f(x, y) + \nabla^2 f(x, y) \quad (8)$$

通过仿真实验, 本文得到的最优拉普拉斯掩膜为

$$Kernel_{\text{LAPLACIAN}} = \begin{bmatrix} 0 & -1 & 0 \\ -1 & S+4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad (9)$$

参数 S 可以根据图像特点和锐化程度进行调节, 拉普拉斯锐化滤波器的增益为 S .

拉普拉斯锐化滤波器结构简单, 比较适合硬件实现. 同时, 仿真实验表明: 拉普拉斯锐化滤波器对缩放图像的性能提升优于其他滤波器. 自适应滤波技术^[1] 也应用到拉普拉斯锐化滤波器上, 进一步改善了图像缩放质量.

2.3 自适应技术

自适应技术的目的是进一步提升锐化滤波器的效果. 图 2 表示前置滤波方案, 锐化滤波的结果反馈回线性存储器作为新的像素输入.

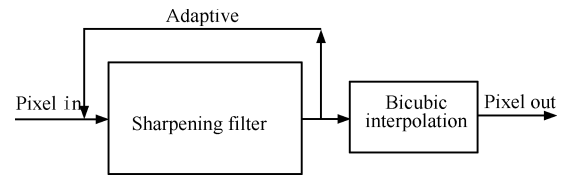


图 2 自适应技术

Fig. 2 Adaptive technology

图 3 表示提出的自适应拉普拉斯滤波器, 滤波结果反馈到第三个线性存储器. 因此, 5 个原始像素和 4 个自适应滤波像素一起组成 3×3 拉普拉斯锐化滤波器的输入像素.

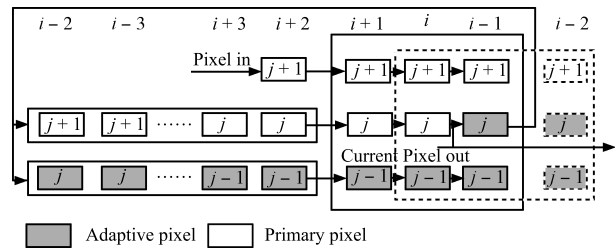


图 3 拉普拉斯自适应滤波过程

Fig. 3 The flow of Laplacian adaptive filter

由于反馈, 锐化滤波器由原来的 FIR 滤波器变成了 IIR 滤波器. 这里通过计算拉普拉斯自适应锐化滤波器的 z 变换证明其稳定性. 拉普拉斯自适应锐化滤波器的脉冲响应的 z 变换为

$$H(z_1, z_2) = \frac{A(z_1, z_2)}{B(z_1, z_2)} = \frac{8 - z_1^{-1} - z_2^{-1}}{4 + z_1 + z_2} \quad (10)$$

因此, $B(z_1, z_2)$ 为

$$B(z_1, z_2) = 4 + z_1 + z_2 \quad (11)$$

由于 $|z_1| = 1, |z_2| \leq 1$, 有:

$$B(z_1, z_2) = 4 + z_1 + z_2 > 0 \quad (12)$$

对于 $|z_1| \leq 1$, 有:

$$B(z_1, 0) = 4 + z_1 > 0 \quad (13)$$

从上面的推导过程可以得出结论, 自适应拉普拉斯锐化滤波器满足稳定条件^[29-30], 这种 IIR 滤波器是稳定的.

2.4 前置滤波和后置滤波

为了减少计算时间, 本文提出前置滤波 (插值前滤波) 和后置滤波 (插值后滤波), 并且仿真验证是否能够改善图像缩放效果. 当图像放大时, 滤波在图像放大前进行; 相反, 当图像缩小时, 滤波在图像缩小后进行, 这样可以减少处理时间. 前置滤波和后置滤波如图 4 所示.

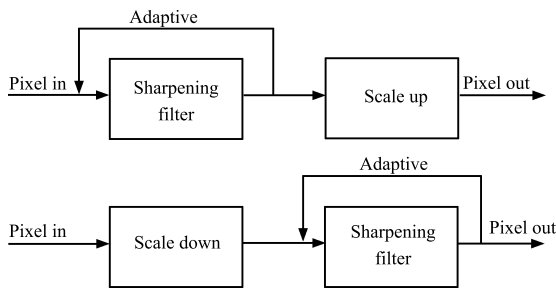


图 4 前置滤波和后置滤波
Fig. 4 Pre-filter and post-filter

3 实验结果

采用 8 幅 512 像素 × 512 像素标准测试彩色图像对提出的算法从主观和客观两个方面进行评估. 测试图像如图 5 所示.

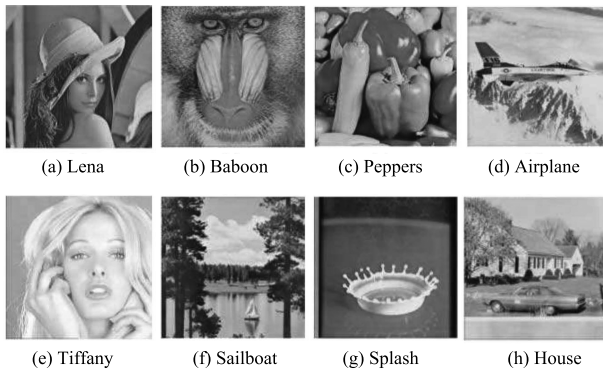


图 5 8 个测试图像
Fig. 5 Eight sample images

3.1 参数 C 和 S 的选择

为了选择拉普拉斯锐化滤波器参数 S 的最优值, 通过实验得到了图 5 中 8 张标准测试图片的峰值信噪比 PSNR 和 $(S + 4)$ 的关系图, 如图 6 所示. 从图 6 可以得出, 对于 8 张标准测试图片的变化趋势是一致的, 最优的 $S + 4$ 变化范围是 12~22. 表 1 分别列出了 PSNR 的最大值时 $S + 4$ 的值和 $S + 4 = 20$ 时的 PSNR 值. 注意这里取 $S = 16$, 是 2 的整数次幂, 既保证了较高的 PSNR 值, 又简化了硬件实现. 基于这个原因, 后面本文的实验采取了 $(S + 4) = 20$ 来进行.

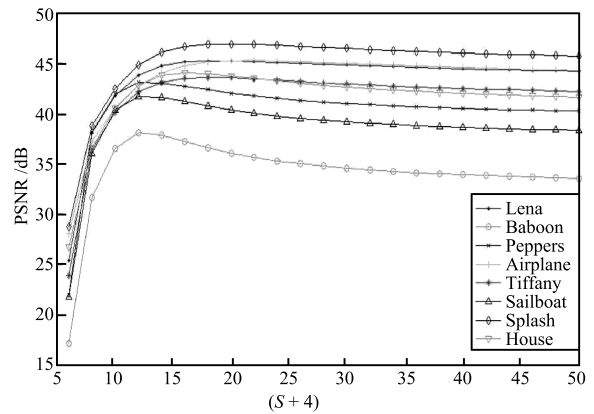


图 6 PSNR 和 S 的关系图

Fig. 6 PSNR with variable S

表 1 PSNR 峰值比较

Table 1 Comparison of peak PSNRs

Image	$S + 4$	Peak PSNR	$S + 4$	PSNR (dB)	PSNR 差值 (dB)
Lena	20	45.26	20	45.26	0
Baboon	12	38.11	20	36.07	2.04
Peppers	12	43.10	20	42.07	1.03
Airplane	22	45.30	20	45.28	0.02
Tiffany	18	43.62	20	43.57	0.05
Sailboat	12	41.74	20	40.39	1.35
Splash	16	48.05	20	47.90	0.15
House	16	44.06	20	43.76	0.30

从表 1 可以看出, 对于 $S + 4 = 20$ 并不能满足对所有的图像都是最优值. 例如, Baboon 图像的误差达到了 2.04 dB. 参数 S 的值可以根据图像的特点适当调整. 对于组合滤波器, 由于涉及到两个参数 C 和 S 同时调整, 因此比拉普拉斯锐化滤波器误差更大. 实验也证明了对于组合滤波, 参数 C 和 S 不存在最优值, $C = 28, S = 32$ 是比较好的选择. 因此, 本文提出的拉普拉斯锐化滤波器更稳定且更容易硬件实现.

3.2 结果比较

本文采用 Matlab 仿真软件从定量的 (客观) 和可视的 (主观) 两个方面, 对提出的改进双三次插值算法进行仿真验证. 均方误差 (Mean square error, MSE) 可以用来定量分析原始图像和插值图像的近似程度. 对于大小为 $M \times N$ 的图像, 图像处理后的均方误差定义为

$$MSE = \frac{1}{MN} \sum_{M=0}^{M-1} \sum_{N=0}^{N-1} [P(i, j) - P'(i, j)]^2 \quad (14)$$

$P(i, j)$ 和 $P'(i, j)$ 分别是坐标为 (i, j) 位置上的原始图像像素值和图像缩放后的像素值. 另外, PSNR 也常用来定量评估图像缩放的性能, 定义为

$$PSNR = 10 \times \lg \frac{MAX^2}{MSE} \quad (15)$$

由于本文中图像的像素是 8 位的, 因此, 式 (15) 中, MAX 的值为 255.

在仿真实验中, 组合滤波器^[1]、拉普拉斯锐化滤波器、前置滤波、后置滤波、自适应滤波都分别进行了测试. 这样对于每个测试图像都进行了 5 种方法的仿真实验, 包括: 1) 双三次插值, 2) 前置滤波双三次插值, 3) 前置自适应滤波双三次插值, 4) 后置滤波双三次插值, 5) 后置滤波自适应双三次插值. 这 5 种方法与自适应双线性 (A-Bilinear)^[1]、扩展线性 (Extend linear, EL)^[22] 进行了比较.

对图 5 中的标准测试图像进行两种测试. 1) 测试 A. 用上述 5 种方法, 分别将大小为 512×512 的源图像放大到 1024×1024 或者缩小到 256×256 , 然后再用 Matlab 仿真软件提供的双三次插值方法把放大或缩小后的图像恢复至源图像的大小 512×512 . 表 2 列出了不同方法的 PSNR 结果. 2) 测试 B. 用 Matlab 仿真软件提供的双三次插值方法, 分别将大小为 512×512 的源图像放大到 1024×1024 或者缩小到 256×256 , 然后再把放大或缩小后的图像用上面提到的五种方法恢复至源图像的大小, 即 512×512 . 表 3 列出了不同方法的 PSNR 结果. 对彩色测试图像的三个颜色分量分别采用式 (15) 计算 PSNR 值, 然后再取平均值作为 PSNR 值.

表 2 和表 3 列出了每种插值算法的 PSNR 值. 性能从高到低依次为: 拉普拉斯自适应双三次插值 (36.563 dB), 拉普拉斯非自适应双三次插值, 组合滤波自适应双三次插值, 组合滤波非自适应双三次插值, 双三次插值 (BC)^[23] (35.038 dB), 扩展线性插值 (EL)^[22], 自适应双线性插值 (A-Bilinear)^[1]. 拉普拉斯自适应双三次插值比双三次插值平均 PSNR 值提升了 1.5 dB. 组合滤波自适应双三次插值较拉普拉斯自适应双三次插值性能有所下降, 但仍比双三次

插值改善很多. 并且, 当进行图像放大时, 前置滤波性能要优于后置滤波, 当进行图像缩小时, 后置滤波优于前置滤波, 这样可以减少计算时间.

表 2 图像放大 512×512 到 1024×1024 和图像缩小 512×512 到 256×256 的 PSNR 值 (dB)

Table 2 The values of PSNR scaling up from size 512×512 to 1024×1024 and scaling down from size 512×512 to 256×256 (dB)

Sample images	Scaling up 512×512→1024×1024		Scaling down 512×512→256×256	
	A	B	A	B
Lena				
A-Bilinear ^[1]		38.55		31.48
EL ^[22]		40.25		32.75
BC ^[23]		43.08		33.36
CAF	43.97	43.98	33.55	33.56
CF	43.93	43.98	33.55	33.55
LASF	45.26	44.07	33.56	33.74
LSF	44.97	44.11	33.57	33.70
Baboon	A	B	A	B
A-Bilinear ^[1]		29.45		22.34
EL ^[22]		29.42		22.65
BC ^[23]		32.32		23.11
CAF	33.95	33.43	23.23	23.27
CF	33.93	33.44	23.23	23.27
LASF	36.07	33.59	23.24	23.43
LSF	35.94	33.68	23.25	23.42
Peppers	A	B	A	B
A-Bilinear ^[1]		36.47		29.71
EL ^[22]		36.33		30.31
BC ^[23]		39.24		30.63
CAF	40.37	40.12	30.73	30.75
CF	40.35	40.13	30.73	30.74
LASF	42.07	40.32	30.74	30.83
LSF	41.86	40.38	30.74	30.82
Airplane	A	B	A	B
A-Bilinear ^[1]		38.57		29.51
EL ^[22]		39.89		30.55
BC ^[23]		43.04		30.98
CAF	44.35	44.11	31.12	31.15
CF	44.29	44.12	31.12	31.14
LASF	45.28	44.08	31.13	31.26
LSF	44.99	44.13	31.13	31.23
Tiffany	A	B	A	B
A-Bilinear ^[1]		37.00		29.99
EL ^[22]		37.93		30.49
BC ^[23]		41.03		30.75
CAF	42.28	42.07	30.83	30.82
CF	42.24	42.07	30.83	30.82
LASF	43.57	42.17	30.84	30.87
LSF	43.32	42.22	30.84	30.85
Sailboat	A	B	A	B
A-Bilinear ^[1]		34.40		27.22
EL ^[22]		34.31		27.96
BC ^[23]		37.22		28.39
CAF	38.48	38.18	28.53	28.57
CF	38.46	38.19	28.53	28.57
LASF	40.39	38.37	28.54	28.73
LSF	40.19	38.45	28.55	28.71
Splash	A	B	A	B
A-Bilinear ^[1]		41.55		32.50
EL ^[22]		41.58		33.13
BC ^[23]		44.93		33.44
CAF	46.52	46.08	33.54	33.55
CF	46.49	46.10	33.54	33.54
LASF	47.90	46.15	33.54	33.62
LSF	47.76	46.23	33.54	33.60
House	A	B	A	B
A-Bilinear ^[1]		36.96		27.24
EL ^[22]		37.17		27.99
BC ^[23]		40.22		28.55
CAF	42.14	41.54	28.72	28.78
CF	42.10	41.56	28.72	28.77
LASF	43.76	41.58	28.72	28.95
LSF	43.61	41.68	28.74	28.93

注: A: Filtering before scaling; B: Scaling before filtering; LASF: Laplacian adaptive sharpening filter; LSF: Laplacian sharpening filter; CF: Combined filter; CAF: Combined adaptive filter.

表 3 图像放大 256×256 到 512×512 和图像缩小 1024×1024 到 512×512 的 PSNR 值 (dB)

Table 3 The values of PSNR scaling up from size 256×256 to 512×512 and scaling down from size 1024×1024 to 512×512 (dB)

Sample images	Scaling up $256 \times 256 \rightarrow 512 \times 512$		Scaling down $1024 \times 1024 \rightarrow 512 \times 512$	
	A	B	A	B
LENA				
A-Bilinear ^[1]		31.79		37.69
EL ^[22]		32.75		40.25
BC ^[23]		33.36		43.08
CAF	33.56	33.52	43.98	43.93
CF	33.55	33.52	43.98	43.88
LASF	33.74	33.47	44.07	45.04
LSF	33.70	33.47	44.11	44.75
Baboon				
A-Bilinear ^[1]		22.61		27.97
EL ^[22]		22.65		29.42
BC ^[23]		23.11		32.32
CAF	23.27	23.21	33.43	33.94
CF	23.27	23.22	33.44	33.93
LASF	23.43	23.22	33.59	36.05
LSF	23.42	23.22	33.68	35.92
Peppers				
A-Bilinear ^[1]		29.98		35.47
EL ^[22]		30.31		36.33
BC ^[23]		30.63		39.24
CAF	30.75	30.68	40.12	40.35
CF	30.74	30.68	40.13	40.33
LASF	30.83	30.63	40.32	41.99
LSF	30.82	30.63	40.38	41.78
Airplane				
A-Bilinear ^[1]		29.81		37.12
EL ^[22]		30.55		39.89
BC ^[23]		30.98		43.04
CAF	31.15	31.00	44.11	44.24
CF	31.14	31.00	44.12	44.18
LASF	31.26	30.87	44.08	44.87
LSF	31.23	30.86	44.13	44.60
Tiffany				
A-Bilinear ^[1]		29.84		36.47
EL ^[22]		30.49		37.93
BC ^[23]		30.75		41.03
CAF	30.82	30.78	42.07	42.22
CF	30.82	30.78	42.07	42.18
LASF	30.87	30.73	42.17	43.32
LSF	30.85	30.72	42.22	43.06
Sailboat				
A-Bilinear ^[1]		27.70		33.10
EL ^[22]		27.96		34.31
BC ^[23]		28.39		37.22
CAF	28.57	28.50	38.18	38.46
CF	28.57	28.50	38.19	38.44
LASF	28.73	28.48	38.37	40.34
LSF	28.71	28.48	38.45	40.13
Splash				
A-Bilinear ^[1]		32.75		40.10
EL ^[22]		33.13		41.58
BC ^[23]		33.44		44.93
CAF	33.55	33.42	46.08	46.44
CF	33.54	33.42	46.10	46.41
LASF	33.62	33.31	46.15	47.66
LSF	33.60	33.31	46.23	47.52
House				
A-Bilinear ^[1]		27.74		34.76
EL ^[22]		27.99		37.17
BC ^[23]		28.55		40.22
CAF	28.78	28.64	41.54	42.09
CF	28.77	28.64	41.56	42.05
LASF	28.95	28.56	41.58	43.57
LSF	28.93	28.57	41.68	43.42

以测试图像 Baboon 作为例子, 图 7 给出了测试方法 A 中各种算法图像放大效果图. 为了进行主观比较, 图 8 给出了原始图像和各种算法放大图像的细节区域对比效果图. 可以看出, 当进行图像放大时, 拉普拉斯自适应算法优于其他算法.

3.3 滤波器频谱

本节分别给出了箱位滤波器、锐化滤波器、组合滤波器、自适应组合滤波器、拉普拉斯滤波器和拉普拉斯自适应滤波器的频谱, 并进行理论分析.

如图 9 所示, 箱位滤波器 (图 9(a)) 属于低通滤波器, 保留了图像的低频分量, 滤除了高频分量. 因此, 可以滤除部分高频噪声. 锐化滤波器 (图 9(b)) 属于高通滤波器, 抑制了低频分量, 保留了图像的高频分量, 同时带来了高频噪声. 组合滤波器 (图 9(c)) 是上述两种滤波器的组合, 有效地保留了更多图像的信息, 降低了高频噪声^[4]. 组合自适应滤波比组合滤波保留了更多频率分量, 如图 9(d) 所示. 拉普拉斯滤波和拉普拉斯自适应滤波的频谱如图 9(e) 和 9(f) 所示. 可以看出, 自适应拉普拉斯滤波器保留的最多有用的频率分量, 也即更多的高频噪声被滤除, 同时, 更多的有用频谱被保留.

4 VLSI 架构

本文提出的拉普拉斯自适应缩放算法的计算复杂度较低, 只需要一个拉普拉斯自适应锐化滤波器和一个双三次插值处理器. 因此, 比较适合低成本 VLSI 实现. 本节给出了 VLSI 架构及 FPGA 实现.

4.1 系统结构

如图 10 所示, 整个系统结构包括一个拉普拉斯自适应滤波器、一个双三次插值器和三个多路选择器. 多路选择器的作用是根据缩放比例系数 (Scale ratio, SR) 来决定滤波和插值的顺序. 基于前面的仿真结果, 当 $SR > 1$ 时, 滤波操作要在插值前, 即前置滤波. 相反, 当 $SR < 1$, 插值操作在滤波前, 即后置滤波.

4.2 拉普拉斯自适应锐化滤波器

基于第 3.1 节的仿真结果, 锐化参数 $S = 16$, 既可以保证 IIR 滤波器的稳定性, 又利于硬件实现. 基于拉普拉斯模板, 新的像素值计算如下:

$$P'_{i,j} = \frac{-P'(i-1, j) - P'(i, j-1) + 20P(i, j) - P(i, j+1) - P(i+1, j)}{16} \quad (16)$$

拉普拉斯自适应滤波由 3 个线性存储器、9 个寄存器和 1 个计算单元组成, 如图 11 所示. 每个线性存储器存储一行图像像素. 为了实现流水线处理, 每个时钟更新一次寄存器数据. 同时, 计算单元获得一个新的像素发送给线性存储器, 并存储回计算单

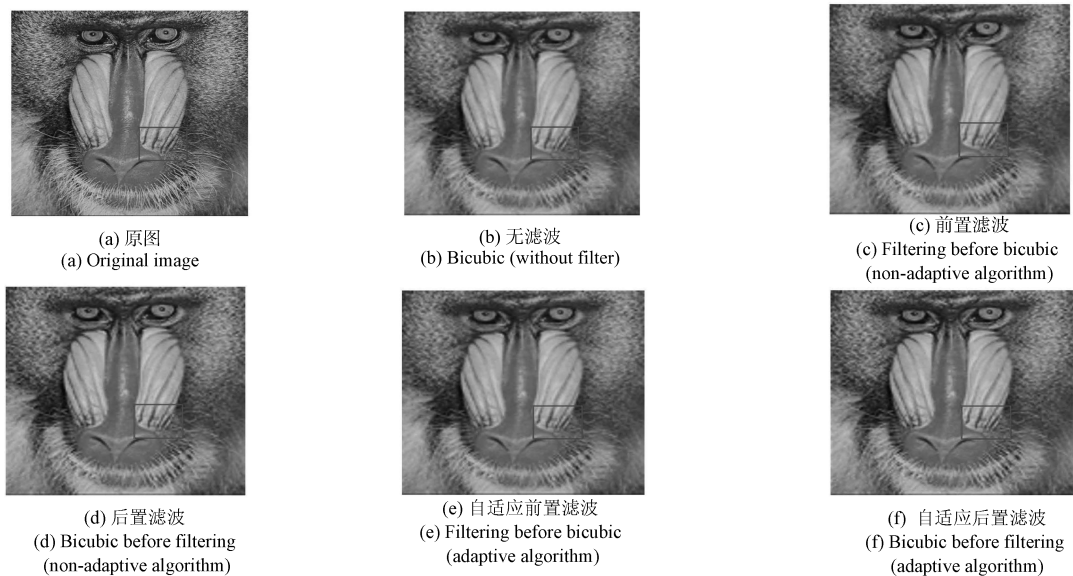


图 7 不同方法放大的图像
Fig. 7 Images with different up-scaling methods

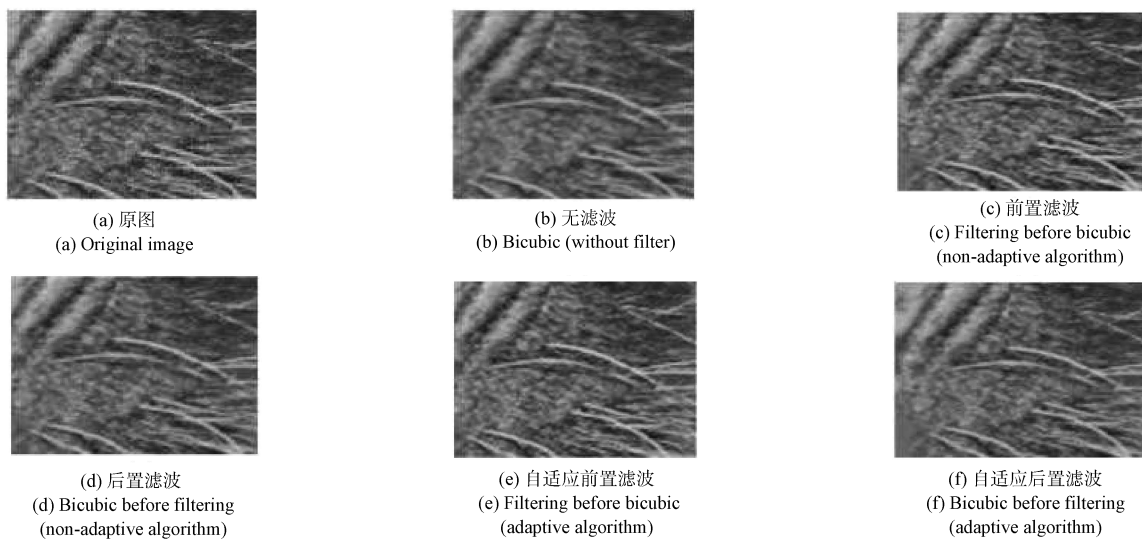
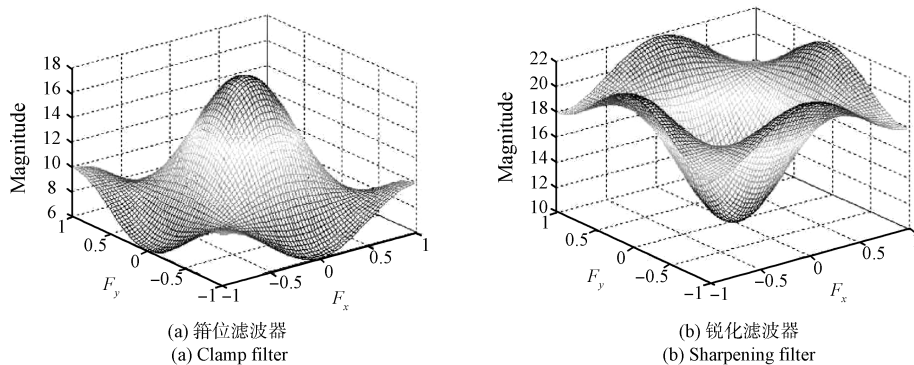


图 8 不同方法放大图像细节
Fig. 8 Image details processed with different up-scaling methods



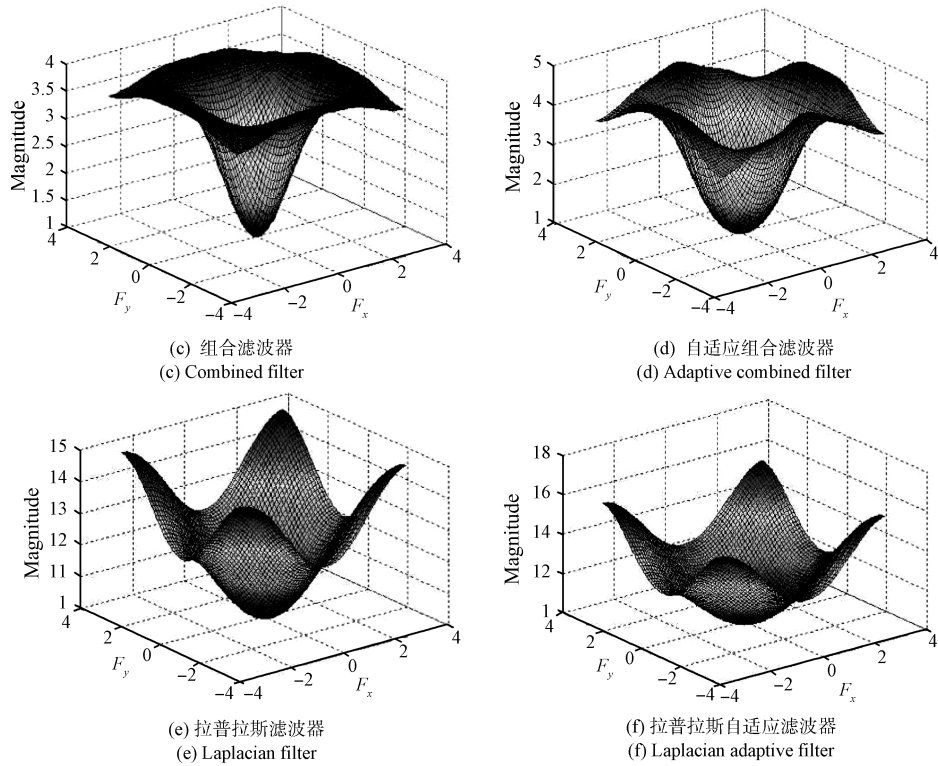


图 9 滤波器频谱

Fig. 9 Spectra of filters

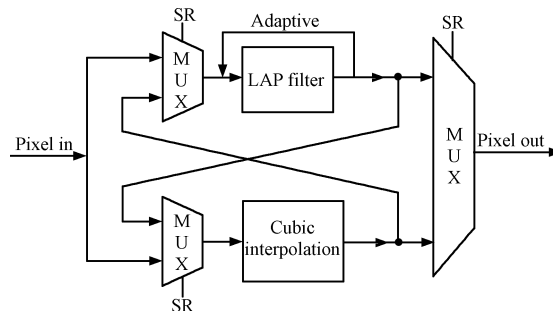


图 10 硬件体系结构框图

Fig. 10 Block diagram of the proposed hardware architecture

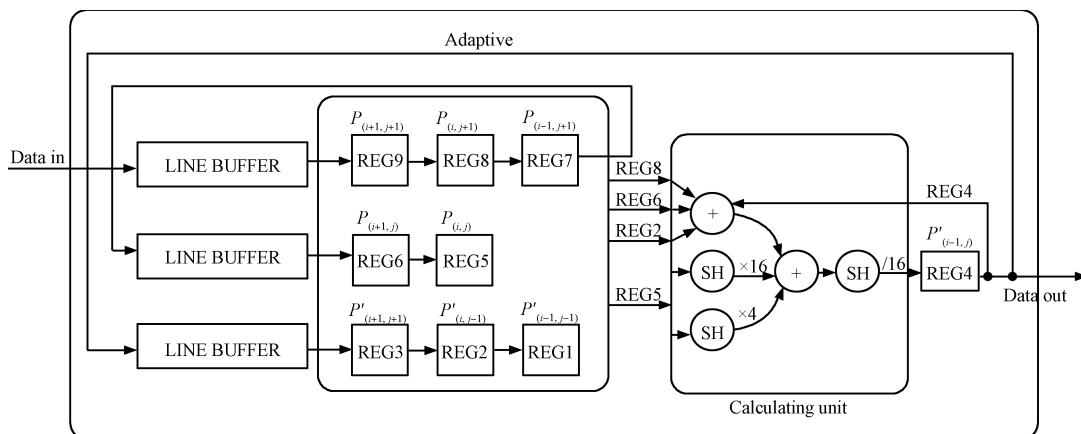


图 11 拉普拉斯锐化滤波器结构框图

Fig. 11 The architecture of the Laplacian sharpening filter

元. 注意计算单元每个时钟产生一个新的像素. 3×3 滤波器的硬件结构很容易映射到式 (9) 中的卷积参数, 卷积表达式如式 (16) 所示. 拉普拉斯自适应滤波器电路的计算单元 (CU) 是最耗资源的部分, 用移位寄存器代替乘法器, 用 3 个移位寄存器和 2 个加法器就可以实现, 有效地降低了整个系统的资源.

4.3 双三次插值

图 12 为双三次插值的硬件结构框图, 分别由下列模块组成: 坐标标定单元 (Coordinate orientation unit)、系数生成器 (Coefficient generator)、垂直插值单元 (Vertical interpolation unit)、虚拟像素缓存 (Virtual pixel buffer) 和水平插值单元 (Horizontal interpolation unit)^[18, 23]. 符号 vs 和 hs 分别表示参考像素和插值像素的垂直和水平距离.

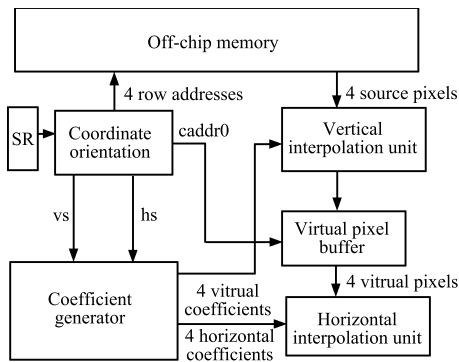


图 12 双三次插值结构框图

Fig. 12 Block diagram of the proposed bicubic hardware architecture

4.3.1 系数生成器

双三次插值的主要瓶颈是动态产生插值系数. 在一维插值中, 计算垂直系数和水平系数的方法是一样的. 根据图 13, 式 (2) 可以改写为

$$\begin{aligned} vc_0 &= \frac{1}{2} (-x^3 + 2x^2 - x) = \frac{1}{2} (-x^3 + 2x^2 - x) \\ vc_1 &= \frac{1}{2} (3x^3 - 5x^2 + 2) = \\ &= \frac{1}{2} (2x^3 + x^3 - 4x^2 - x^2 + 2) \\ vc_2 &= \frac{1}{2} (-3x^3 + 4x^2 + x) = \\ &= \frac{1}{2} (-2x^3 - x^3 + 4x^2 + x) \\ vc_3 &= \frac{1}{2} (x^3 - x^2) = \frac{1}{2} (x^3 - x^2) \end{aligned} \quad (17)$$

x 是原像素 P_1 和虚拟插值像素 F 之间的距离. vc_0, vc_1, vc_2, vc_3 分别是原像素 P_0, P_1, P_2, P_3 的垂直权重系数. 对于芯片实现来说, 乘法器占用的逻辑

资源要远大于移位寄存器. 因此, 在本文的缩放电路中, 用移位运算代替乘法运算. 为了进一步减少硬件资源, 在图 14 中, 采用了流水线技术和硬件共享技术.

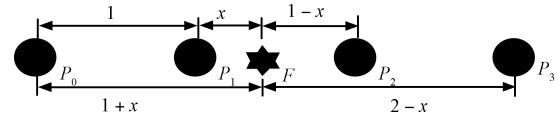


图 13 一维三阶多项式插值

Fig. 13 One-dimension of third-order polynomial interpolation

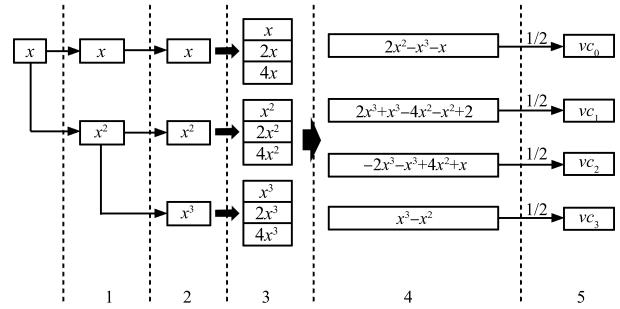


图 14 系数产生器结构框图

Fig. 14 Architecture of the coefficient generator

4.3.2 其他模块

双三次插值其他模块, 如坐标标定单元, 垂直插值单元, 虚拟像素缓存和水平插值单元, 都与文献 [23] 中的设计一样.

4.4 硬件资源比较

整个 VLSI 电路采用 FPGA EP2C70F896C6 进行实现. 表 4 列出了与文献 [23-24] 比较结果, 包括逻辑单元 (Logic elements, LEs), 平均 PSNR 和操作频率.

从表 4 中可以看出, 本文提出的结构可以比文献 [24] 减少 36.8% 逻辑单元, 图像质量 PSNR 比文献 [23-24] 提升超过 1.5 dB. 因此, 整个 VLSI 实现达到了低成本高性能、高质量的目标, 适合许多视频和图像缩放的应用.

表 4 与其他双三次插值设计的比较

Table 4 Comparison with othe bicubic designs

	Bicubic ^[24]	Bicubic ^[23]	Proposed
Computing Resources	32 mult	12 mult	12 mult
Process	26 add	11 add	24 add
LEs (FPGA)	FPGA	FPGA	FPGA
Frequency	1 100	529	695
PSNR	167 M	164 M	165 M
	35.038	35.038	36.563

5 结论

本文提出了几种改进的双三次插值算法, 通过仿真实验证明了拉普拉斯自适应双三次插值算法是最优的. 选择双三次插值是因为它的图像缩放质量较高. 拉普拉斯滤波器结构简单易于硬件实现, 应用到双三次插值中, 有效地解决了双三次插值产生的模糊效应和块效应. 自适应技术进一步提升了图像缩放的质量. 当进行图像放大时, 采用前置滤波, 当进行图像缩小时, 采用后置滤波, 大大减少了处理时间. 改进算法采用一种低成本 VLSI 结构进行了实现, 逻辑单元减少了 36.8%, 图像质量 PSNR 提升了 1.5 dB.

References

- Chen S L, Huang H Y, Luo C H. A low-cost high-quality adaptive scalar for real-time multimedia applications. *IEEE Transactions on Circuits and Systems for Video Technology*, 2011, **21**(11): 1600–1611
- Hwang I, Kang B, Gerard J. High-resolution image scaler using interpolation filter for multimedia video applications. *IEEE Transactions on Consumer Electronics*, 1997, **43**(3): 813–818
- Doswald D, Hafliger J, Blessing P, Felber N, Niederer P, Fichtner W. A 30-frames/s megapixel real-time CMOS image processor. *IEEE Journal of Solid-State Circuits*, 2000, **35**(11): 1732–1743
- Lehmann T M, Gonner C, Spitzer K. Addendum: B-spline interpolation in medical image processing. *IEEE Transactions on Medical Imaging*, 2001, **20**(7): 660–665
- Battiato S, Gallo G, Stanco F. A locally adaptive zooming algorithm for digital images. *Image and Vision Computing*, 2002, **20**(11): 805–812
- Chen M J, Huang C H, Lee W L. A fast edge-oriented algorithm for image interpolation. *Image and Vision Computing*, 2005, **23**(9): 791–798
- Shi J Z, Reichenbach S E. Image interpolation by two-dimensional parametric cubic convolution. *IEEE Transactions on Image Processing*, 2006, **15**(7): 1857–1870
- Chen J L, Chang J Y, Shieh K L. 2-D discrete signal interpolation and its image resampling application using fuzzy rule-based inference. *Fuzzy Sets and Systems*, 2000, **114**(2): 225–238
- Blu T, Thevenaz P, Unser M. Linear interpolation revitalized. *IEEE Transactions on Image Processing*, 2004, **13**(5): 710–719
- Meijering E H W, Zuiderveld K J, Viergever M A. Image reconstruction by convolution with symmetrical piecewise nth-order polynomial kernels. *IEEE Transactions on Image Processing*, 1999, **8**(2): 192–201
- Caselles V, Morel J M, Sbert C. An axiomatic approach to image interpolation. *IEEE Transactions on Image Processing*, 1998, **7**(3): 376–386
- Jensen K, Anastassiou D. Subpixel edge localization and the interpolation of still images. *IEEE Transactions on Image Processing*, 1995, **4**(3): 285–295
- Keys R G. Cubic convolution interpolation for digital image processing. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1981, **29**(6): 1153–1160
- Eadie D, Shevlin F P, Nisbet A. Correction of geometric image distortion using FPGAs. In: Proceedings of SPIE Conference on Optical Metrology, Imaging and Machine Vision. Galway, Ireland, 2002, **4877**: 28–37
- Hudson R D, Lehn D I, Athanas P M. A run-time reconfigurable engine for image interpolation. In: Proceedings of the 1998 IEEE Symposium on FPGAs for Custom Configurable Computing Machines. Napa, California: IEEE, 1998. 88–95
- Skarabot A, Ramponi G, Buriola L. FPGA architecture for a videowall image processor. In: Proceedings of the 2001 SPIE International Symposium on Electronic Imaging 2001. San Jose, California, 2001
- Gibbon K T, Johnston C T, Bailey D G. A real-time FPGA implementation of a barrel correction algorithm with bilinear interpolation. In: Proceedings of the 2003 Image and Vision Computing New Zealand. Palmerston North, New Zealand, 2003. 408–413
- Lin C C, Sheu M H, Liaw C, Chiang H K. Fast first-order polynomials convolution interpolation for real-time digital image reconstruction. *IEEE Transactions on Circuits and Systems for Video Technology*, 2010, **20**(9): 1260–1264
- Andrews S, Harris F. Polynomial approximations of interpolants. In: Proceedings of the 33rd Asilomar Conference on Signals, Systems, and Computers. Pacific Grove, CA, USA: IEEE, 1999. **1**: 447–451
- Shezaf N, Abramov-Segal H, Sutskov I, Bar-Sella R. Adaptive low complexity algorithm for image zooming at fractional scaling ratio. In: Proceedings of the 21st IEEE Convention of the Electrical and Electronic Engineers. Tel Aviv, Israel: IEEE, 2000. 253–256
- Kim H C, Kwon B H, Choi M R. An image interpolator with image improvement for LCD controller. *IEEE Transactions on Consumer Electronics*, 2001, **47**(2): 263–271
- Lin C C, Sheu M H, Chiang H K, Tsai W K, Wu Z C. Real-time FPGA architecture of extended linear convolution for digital image scaling. In: Proceedings of the 2008 IEEE International Conference on ICECE Technology. Taipei, China: IEEE, 2008. 381–384
- Lin C C, Sheu M H, Chiang H K, Liaw C, Wu Z C. The efficient VLSI design of BI-CUBIC convolution interpolation for digital image processing. In: Proceedings of the 2008 IEEE International Symposium on Circuits and Systems. Seattle, WA: IEEE, 2008. 480–483
- Nuno-Maganda M A, Arias-Estrada M O. Real-time FPGA-based architecture for bicubic interpolation: an application for digital image scaling. In: Proceedings of the 2005 IEEE International Conference on Reconfigurable Computing and FPGAs. Puebla City: IEEE, 2005

- 25 Ridella S, Rovetta S, Zunino R. IAVQ-interval-arithmetic vector quantization for image compression. *IEEE Transactions on Circuits and Systems Part II: Analog and Digital Signal Processing*, 2000, **47**(12): 1378–1390
- 26 Shi B E, Chua L O. Resistive grid image filtering: input/output analysis via the CNN framework. *IEEE Transactions on Circuits and Systems, Part I: Fundamental Theory and Applications*, 1992, **39**(7): 531–548
- 27 Schaller S, Wildberger J E, Raupach R, Niethammer M, Klingenbeck-Regn K, Flohr T. Spatial domain filtering for fast modification of the tradeoff between image sharpness and pixel noise in computed tomography. *IEEE Transactions on Medical Imaging*, 2003, **22**(7): 846–853
- 28 Wang Q, Ward R. A new edge-directed image expansion scheme. In: *Proceedings of the 2001 International Conference on Image Processing*. Thessaloniki: IEEE, 2001. **3**: 899–902
- 29 Goodman D. Some stability properties of two-dimensional linear shift-invariant digital filters. *IEEE Transactions on Circuits and Systems*, 1977, **24**(4): 201–208
- 30 Chang H, Aggarwal J K. Design of two-dimensional semi-casual recursive filters. *IEEE Transactions on Circuits and Systems*, 1978, **25**(12): 1051–1059



庞志勇 中山大学信息科学与技术学院博士研究生, 主要研究方向为图像处理, IC 设计.

E-mail: stspzy@mail.sysu.edu.cn

(**PANG Zhi-Yong** Ph.D. candidate at the School of Information Science and Technology, Sun Yat-sen University. His research interest covers image

processing and IC design.)



谭洪舟 中山大学信息科学与技术学院教授. 1998 年获得电子工程博士学位. 在 1998 年~2004 年, 在北美几所大学和 IT 公司工作. 主要研究方向为信号处理, RFID 和 IC 设计. 本文通信作者. E-mail: isssthz@sysu.edu.cn

(**TAN Hong-Zhou** Professor at the School of Information Science and

Technology, Sun Yat-sen University. He received his Ph.D. degree in electronic engineering in 1998. From 1998 to 2004, he was with several universities and IT companies in North America. His research interest covers signal processing, RFID, and IC design. Corresponding author of this paper.)



陈弟虎 中山大学教授. 1986 和 1989 分别获得四川大学学士学位和硕士学位, 2000 年获得香港中文大学电子工程系博士学位. 2000 年至 2001 年到香港中文大学做博士后. 主要研究方向为电子器件, IC 设计和工艺.

E-mail: stscdh@mail.sysu.edu.cn

(**CHEN Di-Hu** Professor at Sun

Yat-sen University. He received his bachelor and master degrees in semiconductor physics from Sichuan University in 1986 and 1989, respectively, and he received Ph.D. degree in solid state electron from Department of Electronic Engineering, Chinese University of Hong Kong in 2000. From 2000 to 2001, he worked in Chinese University of Hong Kong as a postdoctoral fellow. His research interest covers electronic devices, IC design, and design methodology.)