

引用格式:任向隆,田泽,张骏,等. Catmull-Rom 图像缩放算法的自适应结构设计及实现 [J]. 微电子学与计算机,2019,36(8):39-44. [REN X L,TIAN Z,ZHANG J,et al. Design and implementation of adaptive architecture for Catmull-Rom image scaling algorithm [J]. Microeletronics & Computer,2019,36(8): 39-44. ]

# Catmull-Rom 图像缩放算法的自适应结构设计及实现

任向隆<sup>1,2</sup>,田 泽<sup>1,2</sup>,张 骏<sup>1,2</sup>,韩立敏<sup>1,2</sup>,郑新建<sup>1,2</sup>,范飞虎<sup>1</sup>

(1 中国航空工业集团 西安航空计算技术研究所,陕西 西安 710068;

2 集成电路与微系统设计航空科技重点实验室,陕西 西安 710068)

**摘 要:**为解决传统 Catmull-Rom 缩放结构因行列计算量比例不协调导致行/列插值部件长期停顿而造成的性能低下问题,提出了两种自适应结构.在传统结构的基础上增加三个插值部件,并结合两种自适应策略,分配新增插值部件用于行或列插值,并对处于工作状态的插值部件的数量进行调整,得到两种自适应结构.实验表明:与传统结构相比,结构一各类资源占用为原来的 1~2.5 倍,而性能提高最大为原来的 3.99 倍,但仅在缩小时存在性能提升,适用于仅存在缩小的场合(如纹理的 MIPMAP);结构二各类资源占用为原来的 1.7~2.5 倍,而性能提高最大为原来的 3.98 倍,在缩小与放大时均存在性能提升,适用于放大缩小同时存在的场合(如图像的缩放).

**关键词:** Catmull-Rom; 缩放; 自适应结构; 插值部件

中图分类号: TP37

文献标识码: A

文章编号: 1000—7180(2019)08—0039—06

## Design and implementation of adaptive architecture for Catmull-Rom image scaling algorithm

REN Xiang-long<sup>1,2</sup>,TIAN Ze<sup>1,2</sup>,ZHANG Jun<sup>1,2</sup>,  
HAN Li-min<sup>1,2</sup>,ZHENG Xin-jian<sup>1,2</sup>,FAN Fei-hu<sup>1</sup>

(1 Aeronautical Computing Technique Research Institute,AVIC, Xi'an 710068, China;

2 Key Laboratory of Aviation Science and Technology on Integrated  
Circuit and Micro-System Design, Xi'an 710068, China)

**Abstract:** Two adaptive structures are proposed to conquer the low performance caused by long-term pause of the row and column interpolation components in traditional structures of Catmull-Rom, when operations between row and column not suitable. In the two adaptive structures, three interpolators were added on the basis of traditional structure, along with two adaptive strategies which could allocate interpolators between row and column and adjust the number of them at working. The result of experiment is as follows. Compared with the traditional structure, the first structure achieved up to 3.99 times performance improvement, while resources occupy between with 1 to 2.5 times, and the performance improvement achieved only when scaling down, suitable for cases of shrinking only, such as MIPMAP of texture; The second structure achieved up to 3.98 times performance improvement, while resources occupy between with 1.7 to 2.5 times, and the performance improvement achieved when scaling up as well as scaling down, suitable for occasions of scaling up and down, such as image zooming.

**Key words:** catmull-rom; scale; adaptive architecture; interpolation unit

## 1 引言

图形化界面广泛应用于 PAD、智能手机等消费类产品,直接影响用户的操作效率和体验指数. 目前大部分嵌入式操作系统及软件的 GUI 均基于 2D 技术实现<sup>[1]</sup>.

2D 图形加速器中,在源图像与目标图像大小不一致时,需对源图像进行缩小或放大处理以适应目标图像. 图像缩放是 2D 图形加速器的关键功能,对图形化界面的性能影响巨大.

图像缩放目前主要基于插值算法实现,如最近邻域插值、双线性插值、样条插值及基于边缘的插值等<sup>[2-8]</sup>. 其中近邻法使目标图像的像素点间方差较大,疵点较多;双线性法使目标图像更多地产生源图像不具有的像素值,导致图像细节丢失、不可逆转;样条法效果较好,但参考点选取较多,不适合仅适合小倍率缩放;边缘插值则需要根据边缘特征选择插值算法,实现复杂,难以做到实时处理.

除缩放算法本身和软件实现外,研究者也对图像缩放算法的硬件实现进行了研究<sup>[9-17]</sup>. 随邻近点数的增加,缩放效果会更加平滑,但硬件实现复杂度也随之增加.

在算法、硬件实现复杂度以及图像缩放质量之间进行折衷,许多文献<sup>[4,5,10,11]</sup>将 Catmull-Rom 三次样条插值作为研究对象. 文献[10]针对图像边缘缩放处理导致的边缘瑕疵和模糊现象,提出了一种边缘自适应 Catmull-Rom 图像插值算法并完成了该算法的硬件设计;文献[11]提出了 Catmull-Rom 样条插值算法的一种分步插值结构,并进行了实现. 在行列计算量不协调时,文献[10,11]的结构均存在流水线停顿问题,且最大吞吐率为每拍一个像素,若直接将其应用于 2D 加速器中,该结构会成为整个 2D 加速器的性能瓶颈.

为解决上述问题,本文在原有结构的基础上增加三个插值部件,并结合两种自适应的分配策略,根据缩放的实际情况动态地分配三个插值部件用于行或列插值运算,并对处于运算状态的部件数目进行调节. 实验结果表明本文提出的自适应结构能够大幅度提升其处理效率.

## 2 算法原理及存在问题

Catmull-Rom 算法中,每相邻的 4 个节点,就可构造出一段 Catmull-Rom 样条,且每段样条可以拼接构成 C 连续. Catmull-Rom 样条函数中每个节点

处 $(X_i, P_i)$ 的一阶导数可以由其相邻的前后 2 个节点计算出,其一阶导数定义如下:

$$p'_i = \tau (p_{i+1} - p_{i-1}) \quad (1)$$

式中, $\tau$ 称为形状因子,当用于插值用途时, $\tau = 0.5$ 为最佳值. 基于数字图像特殊性的考虑,定义一组步长为 1 的等距点列,任意一组相邻的 5 个节点依次为: $P_{i-2}, P_{i-1}, P_i, P_{i+1}$  和  $P_{i+2}$ . 在 $(X_{i-1}, P_{i-1})$ 与 $(X_i, P_i)$ 间可以构造出一段 3 次样条曲线,设插值点坐标为 $x \in [x_i, x_{i+1}]$ ,且其对应的浮点坐标为 $t = |x - x_i| \in [0, 1]$ ,那么该 3 次样条函数如下:

$$P(X) = c_0 + c_1 t + c_2 t^2 + c_3 t^3 = \sum_{k=0}^3 c_k t^k \quad (2)$$

根据已知各端点值可知,式(2)应满足以下条件:

$$\begin{cases} p(0) = c_0 = p_{i-1} \\ p(1) = c_0 + c_1 + c_2 + c_3 = p_i \\ p'(0) = c_1 = \tau(p_i - p_{i-2}) \\ p'(1) = c_1 + 2c_2 + 3c_3 = \tau(p_{i+1} - p_{i-1}) \end{cases} \quad (3)$$

通过式(3)解出 Catmull-Rom 样条函数的系数为:

$$\begin{cases} c_0 = p_{i-1} \\ c_1 = (-\tau)p_{i-2} + \tau p_i \\ c_2 = (2\tau)p_{i-2} + (\tau - 3)p_{i-1} + (3 - 2\tau)p_i + (-\tau)p_{i+1} \\ c_3 = (-\tau)p_{i-2} + (2 - \tau)p_{i-1} + (\tau - 2)p_i + (\tau)p_{i+1} \end{cases} \quad (4)$$

由上述(1)、(2)和(4)式得出 Catmull-Rom 样条函数插值矩阵如下:

$$p(t) = [1 \ t \ t^2 \ t^3] \begin{bmatrix} 0 & 1 & 0 & 0 \\ -\tau & 0 & \tau & 0 \\ 2\tau & \tau - 3 & 3 - 2\tau & -\tau \\ -\tau & 2 - \tau & -2 & \tau \end{bmatrix} \quad (5)$$

式中, $t \in [0, 1]$ , $\tau = 0.5$ .

以放大 2 倍为例来描述一幅图像进行样条插值的过程,具体如图 1 所示. 以 1/2 作为步长,对原图的行方向进行插值得到水平分辨率为原图 2 倍的中间图像(其中:1、2、3、4 为原已存在的列,其值经过重新计算;A、B、C、D 为新插入的列);接着对中间图像的列方向进行插值得到经过放大的最终图像(其中:5、6、7、8 为原已存在的行,其值经过重新计算;E、F、G、H 为新插入的行). 其余放大倍数与放大 2

倍的处理类似,只是步长不同. 对于图像缩小处理: 先计算行的缩小点值(一个缩小值替代了多个实际像素点),再计算列的缩小点值,以完成整幅图像的缩小.

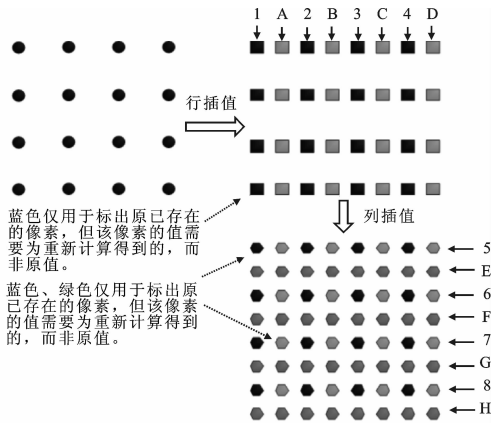


图 1 Catmull-Rom 样条插值过程

由 Catmull-Rom 样条插值算法可知:独立计算行(列)插值时,插值点与行(列)方向上的 4 个相邻像素存在数据相关性;整幅图像若先计算行(列)插值,后计算列(行)插值,则后计算的列(行)插值点与先计算出的行(列)插值点存在数据相关性.

文献[10-11]的结构中行列插值各采用一套插值部件,且使用 4 个 BANK(每 BANK 为  $2048 * 32$  bit 的 DPRAM)组成的存储存放行插值结果. Catmull-Rom 样条插值算法数据相关性的存在导致该结构在行列计算量不协调时流水线会长期处于停顿状态,整体性能有较大提升空间. 例如,在列缩小倍数为 4 时:首先,需等待图像前三行像素的行插值完成,并在第四行的第一个点行插值完成后,才可启用列插值,实现行列并行插值;其次,列插值产生目标图像的一行数据后,在列插值开始进行目标图像下一行的计算前,依据插值算法的数据相关性,存储的中间数据有三行需进行更新. 综合上述两种情况,开启下一行列插值时均需等待三行行插值的完成,该等待的存在,造成列插值出现长时停顿,整体性能有待提升.

3 自适应结构一

为避免行列计算量不协调时流水线出现长期停顿,本文提出一种优化结构:在文献[10-11]的基础上增加三个插值部件用于行插值,从而保证列插值不需要长期等待行插值.

3.1 自适应策略

该结构下,三个插值部件固定地用于行插值,处

于工作状态的插值部件的数量由硬件根据列缩放因子事先配置(一次缩放开始之前),其自适应策略如图 2 所示,其中  $f$  表示列缩放因子.

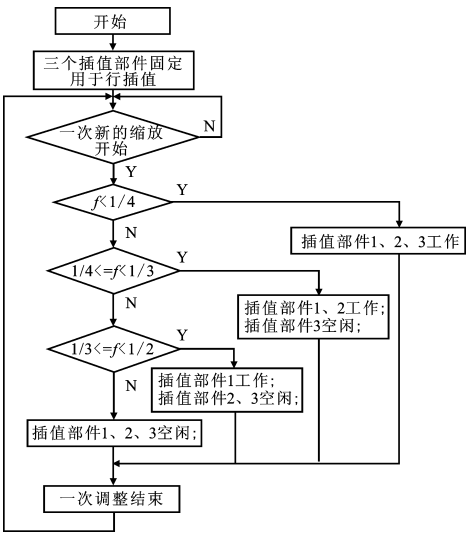


图 2 自适应结构一的自适应策略

3.2 自适应结构

第一种结构的实现如图 3 所示,其中包括自适应策略模块、读像素控制模块、乒乓 RAM、读乒乓 RAM 控制模块、五个插值部件(行、列插值部件各一个,及额外三个插值部件)、行缓冲、写行缓冲控制(wr\_ctr)和读行缓冲控制(rd\_ctr)模块. 首先,当自适应策略模块接收到启用缩放操作的信号时,锁存缩放相关数据信息. 其次,使能读像素控制模块从 DDR2 中读取像素信息,并存储在内部的乒乓 RAM 中;然后,由自适应策略模块使能读乒乓 RAM 控制模块将像素信息从乒乓 RAM 中读出发送给行插值部件以及三个添加的插值部件(由自适应策略模块决定是否工作),行插值部件以及三个添加的插值部件将计算的结果由写行缓冲控制(wr\_ctr)写入行缓冲中;再次,由自适应策略模块和写行缓冲控制(wr\_ctr)模块控制使能读行缓冲控制(rd\_ctr)模块读取行缓冲中的插值结果发送给列插值部件;最后,列插值部件将运算的结果输出.

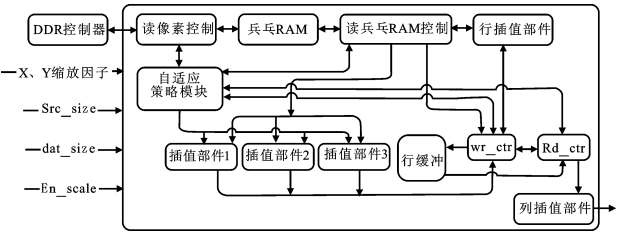


图 3 自适应结构一

4 自适应结构二

文献[10-11]结构和上文提出的自适应结构一，流水线充满时，每拍最多只能输出一个像素；此外，结构一的性能提升仅存在于列方向缩小的情况。因此，在结构一的基础上，提出第二种结构。

4.1 自适应策略

为进一步提高缩放单元的吞吐量和适应性，提出另一种自适应策略。具体为：不再将添加的三个插值部件固定地分配给行插值，而是在缩放过程中根据插值运算的实际情况，不断地在行插值和列插值中动态进行调整（三个作为一个整体，同一时刻均用于行或列），并对处于工作状态的插值部件的数量动态进行调整，该自适应策略如图 4 所示。其中： $r$  表示行插值运算已产生的行数； $f$  表示列缩放因子； $v$  表示列插值部件需要的行号（初始值为 0）。

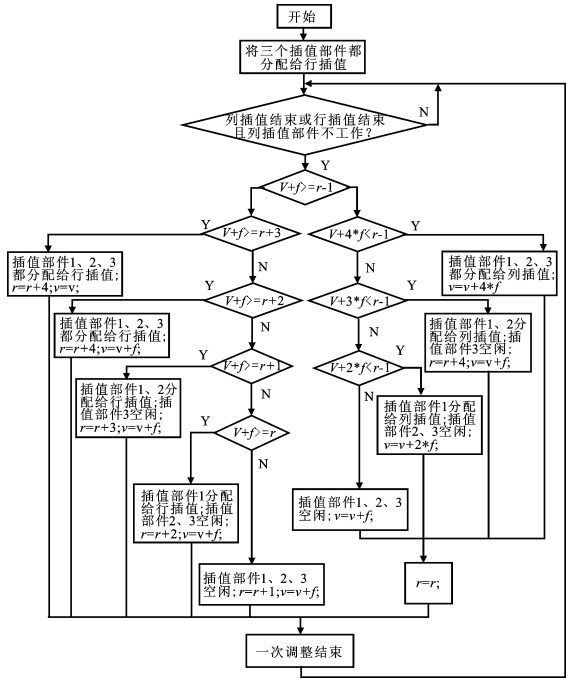


图 4 自适应结构二的自适应策略

4.2 自适应结构

第二种结构的实现如图 5 所示，在第一种结构的基础上，自适应策略模块进行了修改，并增加了一个列缓冲模块和两个多路选择器，其处理过程与结构一类似。相对于结构一，结构二中自适应策略模块更为复杂。

理论上，随插值部件和存储资源的增加，其效率可以线性增长，但 Catmull-Rom 算法计算列的一个点时需要行的 4 个点，即需要 4 行中间数据，所以在原有 1 个插值部件的基础上，即增加 3 个插值部件

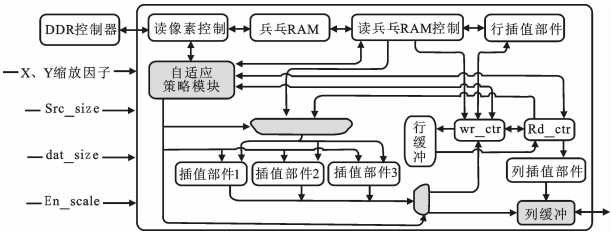


图 5 自适应结构二

是合适的，超过 3 个则会造成资源浪费。采用上述设计，在列缩放系数大于等于 2 时，缩放模块能够保证每拍最多输出 4 个像素。

5 验证结果与分析

为对提出的自适应结构进行功能、性能和代价分析，使用 Verilog 实现了文献[10-11]、方案 1 和方案 2 的缩放结构，并构建了基于 RTL 的仿真验证平台。

5.1 功能验证

使用源分辨率为  $650 \times 520$  的图像，将目标分辨率设置为  $163 \times 130$ ，在仿真验证平台上分别对文献[10-11]、结构 1 和结构二进行仿真验证。为排除人眼对细微差别无法分辨的情形，使用工具对 3 种结构产生图像的数据文件进行了对比，结构一、结构二与文献[10-11]结构产生的目标图像数据没有任何差别，验证了本文所提结构功能的正确性。

5.2 性能分析

为验证本文提出自适应缩放结构的性能，采用一幅源分辨率为  $200 \times 150$  的图像，设置 8 种不同的目标分辨率，形成 8 个测试用例，如表 1 所示。

表 1 性能测试用例

|      | 源分辨率/像素 |     | 目标分辨率/像素 |       | 缩放倍数 |      | 目标像素数量/个  |
|------|---------|-----|----------|-------|------|------|-----------|
|      | x       | y   | x        | y     | x    | y    |           |
| 用例 1 | 200     | 150 | 48       | 36    | 0.24 | 0.24 | 1 728     |
| 用例 2 | 200     | 150 | 60       | 45    | 0.3  | 0.3  | 2 700     |
| 用例 3 | 200     | 150 | 80       | 60    | 0.4  | 0.4  | 4 800     |
| 用例 4 | 200     | 150 | 160      | 120   | 0.8  | 0.8  | 19 200    |
| 用例 5 | 200     | 150 | 300      | 225   | 1.5  | 1.5  | 67 500    |
| 用例 6 | 200     | 150 | 400      | 300   | 2    | 2    | 120 000   |
| 用例 7 | 200     | 150 | 800      | 600   | 4    | 4    | 480 000   |
| 用例 8 | 200     | 150 | 1 600    | 1 200 | 8    | 8    | 1 920 000 |

对于每个用例分别使用文献[10-11]和结构一、结构二对图像进行缩放操作，测量从开始缩放到完成缩放各结构所需的时钟周期数。对于各个用例，从

每种结构完成缩放所需周期,可以看出,随目标像素数量的增加,三种结构所需的周期数据均相应增加,但结构二与其他两种相比,增加最为缓慢。

为更加清晰地对三种结构的性能进行对比,以各测试用例下文献[10-11]所需的周期数目作为基准,对结构一、结构二的所需周期数进行归一化处理,结果表明:

(1)对于结构一,在图像缩小时,有性能提升,本文的用例中最大可提升 3.99 倍;在图像放大时,与文献[10-11]相比没有性能提升;

(2)对于结构二,在图像缩小和放大时均有性能提升,且在用例中最大可提升 3.98 倍;

(3)性能的提升不是一个固定的数值,而与图像缩放比例有关。

本文测试用例依据自适应算法的区间选取,不同区间工作的插值部件数量有所调整,与原本 1 个插值部件相比,工作的插值部件数量的增加带来相应倍数的性能提升。实验证明实测性能提升与理论分析值接近。

此外,选用 Xilinx 的 FPGA 芯片,在 ISE 中对本文结构一和结构二进行综合,其最大频率均可达到 268 MHz。

### 5.3 代价分析

在 ISE 中对论文[10-11]和本文提出的两种自适应 Catmull-Rom 缩放结构进行了综合,布局布线后各自的资源使用报告如图 6 所示。

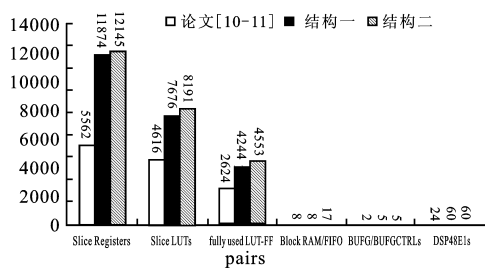


图 6 三种结构各自的资源使用情况

为更加清晰地比较,以文献[10-11]结构所使用的各项资源作为基准,对本文两种结构的资源使用情况进行了归一化处理,结果如图 7 所示。可以看出:

对于 Slice Registers、BUFG/BUFGCTRLs、DSP 资源,本文所提两种结构基本相同,与文献[10-11]所占资源相比,为其 2.1 到 2.5 倍;对于 Slice LUTs、fully used LUT-FF pairs 资源,本文所提两种结构基本相当,与文献[10-11]所占资源相比,为其 1.6 到 1.8 倍;而对于 Block RAM/FIFO 资源,

结构一与文献[10-11]相同,结构二为前两者的 2.1 倍。

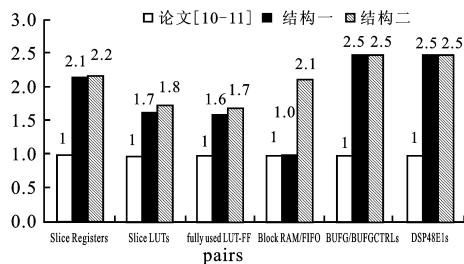


图 7 归一化的资源使用对比

## 6 结束语

根据 Catmull-Rom 算法的数据相关性,在文献[10-11]的基础上,增加了三个插值部件,并结合两种自适应分配策略提出了 Catmull-Rom 图像缩放算法的两种自适应结构,可对新增插值部件用于行或列插值,以及处于工作状态的插值部件的数量进行调整。使用 verilog 对本文的设计进行实现,构建基于 RTL 的仿真验证平台,并选用 Xilinx 的 FPGA 芯片对本文结构一和结构二进行综合。实验表明:与传统结构相比,结构一各类资源占用为原来的 1~2.5 倍,而性能提高最大为原来的 3.99 倍,但仅在缩小时存在性能提升,适用于仅存在缩小的场合(如纹理的 MIPMAP);结构二各类资源占用为原来的 1.7~2.5 倍,而性能提高最大为原来的 3.98 倍,在缩小与放大时均存在性能提升,适用于放大缩小同时存在的场合(如图像的缩放)。目前这两种自适应结构已应用于我们自主研发的图形处理器之中。

### 参考文献:

- [1] 袁扬智,韦明,曾献军. 2D 图像硬件加速引擎的设计优化[J]. 微电子学与计算机, 2009, 26(4): 241-248.
- [2] 郭莹,李伦,王鹏. 基于 Lanczos 核的实时图像插值算法[J]. 通信学报, 2017, 38(6): 142-147.
- [3] 陈宝国,李宁. 一种基于 Sobel 算子和混合有理插值的图像缩放方法[J]. 微电子学与计算机, 2012, 29(2): 150-152.
- [4] 陈利平. 自适应 Catmull-Rom 样条图像放大[J]. 计算机辅助设计与图形学学报, 2013, 25(2): 200-207.
- [5] 江巨浪. 基于改进 Catmull-Rom 样条的图像缩放算法[J]. 计算机技术与发展, 2007, 17(4): 211-217.
- [6] 王平,全吉成,赵柏宇. 基于双线性插值的图像缩放在 GPU 上的实现[J]. 微电子学与计算机, 2016, 33(11): 129-132.
- [7] 王翔. 数字图像缩放及图像质量评价关键技术研究

[D]. 杭州:浙江大学, 2012.

[8] 王玮. 几种最新的图像缩放方法的比较性研究[D]. 大连: 大连理工大学, 2013.

[9] 邹学瑜, 刘昌禄, 胡敬营. 基于双线性插值算法的缩放 IP 核设计[J]. 计算技术与自动化, 2017, 36(3): 113-117.

[10] 魏美荣, 蒋林. 一种自适应 Catmull-Rom 样条图像缩放算法的设计实现[J]. 微电子学与计算机, 2011, 28(12): 63-67+71.

[11] 吕荣国, 蒋林, 杨飞. 图像缩放的研究与 FPGA 设计[J]. 计算机技术与发展, 2011, 21(12): 205-208.

[12] 莫迪涵, 杜慧敏, 沈绪榜. 基于线性插值法图像缩放的设计与 FPGA 实现[J]. 西安邮电大学学报, 2013, 18(3): 52-55.

[13] 冉峰, 李天, 季渊, 等. 基于 FPGA 的图像缩放算法设计[J]. 光电子技术, 2017, 37(1): 21-26.

[14] 武文斌, 高俊雄, 王耘波. 基于叠加的图像缩放优化及 FPGA 实现[J]. 微电子学与计算机, 2017, 34(11): 128-130.

[15] 王中博, 周远远, 黄鲁. 基于多相位插值的视频缩放系统 FPGA 实现[J]. 微型机与应用, 2016, 35(24): 46-49.

[16] 丁家隆, 杜慧敏, 曹广界, 等. 基于最优移位双线性插值的图像缩放旋转硬件加速研究[J]. 微电子学与计算机, 2016, 33(7): 115-124.

[17] 吴以凯. 基于 FPGA 的视频缩放的设计与实现[D]. 镇江: 江苏大学, 2017.

作者简介:

**任向隆** 男, (1982-), 博士, 高级工程师. 研究方向为图形处理器系统结构、微处理器系统结构.  
E-mail: r\_x\_l2004@163. com.

**田 泽** 男, (1965-), 博士, 研究员. 研究方向为 VLSI 设计、SoC 设计、航空专用集成电路设计.

**张 骏** 男, (1978-), 博士后, 高级工程师. 研究方向为图形处理器系统结构、微处理器系统结构.

(上接第 38 页)

[9] XU L, HUIJSING J H, MAKINAWA K A A. A 12uW NPN-based Temperature Sensor With a 18. 4pJ • K<sup>-2</sup> FOM in 0. 18um BCD CMOS[C]//International Workshop on Advances in Sensor and Interfaces. IEEE, 2017: 180-182.

作者简介:

**陈贵宝** 男, (1993-), 硕士研究生. 研究方向为 CMOS 集成

温度传感器设计.

**郭仲杰(通讯作者)** 男, (1982-), 教授, 研究员. 研究方向为数模混合集成电路设计. E-mail: zjguo@xaut. edu. cn.

**李 婷** 女, (1985-), 博士, 高级工程师, 研究方向为数模混合集成电路设计.

**魏海龙** 男, (1976-), 博士, 研究员, 硕士生导师. 研究方向为模拟集成电路设计.