# 图像缩放的研究与FPGA设计

吕荣国,蒋 林,杨 飞
(西安邮电学院 电子工程学院,陕西 西安 710061)

**摘 要**:针对图像缩放的不同要求,提出一种分步插值图像缩放器结构。对行列在空间上分开,在时间上同步,在开始的K(选择不同的算法K值不同)行行列串行处理,先进行行缩放,再用新生成的像素点进行列操作,K行处理完后,行列并行处理。这种结构使得图像缩放时并行度得到提升,由于行列分开处理使得对于不同要求的场合行列可以采用相同或不同的算法以达到特定的要求。实验表明,该分步式插值电路架构既节省了资源、降低了开销,又提高了图像质量。

**关键词**:图像缩放;Catmull_Rom;分步插值
**中图分类号**:TP391.4 **文献标识码**:A **文章编号**:1673-629X(2011)12-0205-04

# Research and FPGA Design of Image Scaling

LÜ Rong-guo, JIANG Lin, YANG Fei
(School of Electronic Engineering, Xi'an University of Posts and Telecommunications, Xi'an 710061, China)

**Abstract**: Due to various request for image scaling, present a hardware design for an image scalar. The scalar allows fractional step interpolation. It is partitioned for two-stage processing with the first stage realizing horizontal scaling and the second stage realizing vertical scaling. Each stage in itself is a hardware pipeline. Horizontal scaling starts first. When sufficient interpolated rows are generated, vertical scaling will start and run in parallel with horizontal scaling. Both stages may use the same type of arithmetic components for interpolation although their control circuitry may be different. Experimental results indicate that this design is efficient.

**Key words**: image scaling; Catmull_Rom; fractional step interpolation

## 0 引 言

随着技术的发展,对于图像质量的要求越来越高,高清显示成为一种必然[1]。LCD显示屏的显示效果不仅和图像的像素有关,也和自身的分辨率有关,当这两者相同时显示的效果为最佳[2,3],所以要显示最好的效果,进行图像缩放就成了一种必然,因此图像缩放器的性能一定程度上会影响显示的效果。图像缩放中对图像的处理都是基于一定的插值算法,目前图像缩放常用的插值算法主要有最近邻插值、双线性插值、双三次插值等,其中最近邻和双线性插值本身并不适合图像缩放,缩放效果不理想[4,5]。

文中借助于Catmull_Rom算法设计了一种分步插值图像缩放器结构,经设计验证此处理后的图像清晰,设计的缩放器资源消耗少。

**收稿日期**:2011-05-08;**修回日期**:2011-08-15
**基金项目**:陕西省科技创新工程重大科技专项(13115)(2009ZDKG-43)
**作者简介**:吕荣国(1985-),男,陕西宝鸡人,硕士研究生,研究方向为集成电路设计;蒋 林,教授,博士,硕士生导师,主要研究领域为专用集成电路设计。

## 1 算法介绍

Catmull_Rom样条函数具有$c^1$上的连续性质,其通过前后四个节点上的像素值确定插值点上的像素。设输入图像分辨率为$m \times n$,$P_{i,j}(0 \leqslant i \leqslant m-1; 0 \leqslant j \leqslant n-1)$表示原图像中第$i$行、第$j$列像素的灰度值,它与平面上的二维点$(i,j)$相对应[6-8]。设有节点$t_0$, $t_1$, $t_2$,…$t_i$,…,对应的像素值为$p_0$, $p_1$,$p_2$,…, $p_i$,…,对节点$t_i$处的一阶导数定义如下:

$$p_i' = \tau(p_{i+1} - p_{i-1}) \tag{1}$$

式中$\tau$为形状因子,设原图像中相邻两个像素点之间的距离为1,$p_{i-1}$,$p_i$,$p_{i+1}$,$p_{i+2}$为相邻的四个像素点,设在$p_i$,$p_{i+1}$之间插值坐标为$x \in [x_i, x_{i+1}]$,则运算中使用的浮点坐标为$t = x - x_i$,构造的三次样条函数为[9,10]:

$$p_{(t)} = a_0 t^3 + a_1 t^2 + a_2 t + a_3 \tag{2}$$

(2)式满足的约束条件为:

$$\begin{cases} p_{(0)} = a_3 = p_i \\ p_{(1)} = a_0 + a_1 + a_2 + a_3 = p_{i+1} \\ p'_{(0)} = a_2 = \frac{1}{2}p_{i+1} - \frac{1}{2}p_i \\ p'_{(1)} = 3a_0 + 2a_1 + a_2 = \frac{1}{2}p_{i-1} - \frac{1}{2}p_i - \frac{1}{2}p_{i+1} + \frac{1}{2}p_{i+2} \end{cases}$$

(3)

由(3)式解方程可得:

$$\begin{cases} a_0 = \frac{3}{2}p_i - \frac{1}{2}p_{i-1} - \frac{3}{2}p_{i+1} + \frac{1}{2}p_{i+2} \\ a_1 = p_{i-1} - \frac{5}{2}p_i + 2p_{i+1} - \frac{1}{2}p_{i+2} \\ a_2 = \frac{1}{2}p_i - \frac{1}{2}p_{i-1} \\ a_3 = p_i \end{cases} \quad (4)$$

由(4)式,像素值可表示为 $p(t) = \boldsymbol{T} \cdot \boldsymbol{M} \cdot \boldsymbol{G}$,其中 $\boldsymbol{T} = \frac{1}{2}[t^3, t^2, t, 1]$,$t(0 \leqslant t < 1)$,$\boldsymbol{M}$ 为一个固定矩阵,

$$\boldsymbol{M} = \begin{bmatrix} -1 & 3 & -3 & 1 \\ 2 & -5 & 4 & -1 \\ -1 & 0 & 1 & 0 \\ 0 & 2 & 0 & 0 \end{bmatrix}$$

$\boldsymbol{G}$ 为四个基点的灰度值向量 $\boldsymbol{G} = [p_{i-1}, p_i, p_{i+1}, p_{i+2}]^{\mathrm{T}}$。

如图 1 所示,通过 Catmull_Rom 曲线拟合可以得到曲线上任意点的像素值[11-13]。$t$ 一直累加,当其大于等于 1 时,去掉其整数部分,使用其小数部分;当 $t = 0$ 时,$p_1$ 即为所求的结果,当 $t = 1$ 时,使用 $t = 0$。

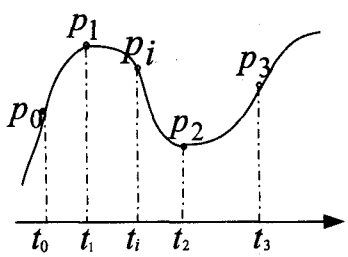



图 1 Catmull_Rom 样条插值

## 2 分步插值图像缩放原理与硬件实现

### 2.1 设计原理

设计原理如图 2 所示,空心圈是原图像的像素点,假设原图像相邻像素点间的距离为 1。四个相邻像素点生成一个新的像素点,每次使用的 4 个基点由浮点坐标系数 $t$ 决定,若 $t = 0$,则使用当前点前一个,后两个,及本身这 4 个像素值;若 $0 < t < 1$,则使用当前位置前两个,后两个这 4 个像素点;若 $t = 1$,则按 $t = 0$ 时处理;若 $t > 1$ 则使用 $t$ 的小数部分,4 个基点的采样如前所述。

由于图像缩放中边缘像素点的高频分量的影响,常常会导致缩放后边缘图像模糊和马赛克,对边缘像素点必须单独处理,设计中对于首行首列边缘像素点均复制,尾行和尾列的像素点要根据实际情况决定是否复制,图中实心圈是对图像边缘像素点值的复制,叉为插入的像素点,开始时行先插值,第三行插完第一个点后列插开始,此后行列同时进行,由于列插需要 4 行的像素,设计中提供 4 个 RAM 用于存储行插后的结果供列插使用。RAM 的深度可参数化,根据原图像和放大倍数的不同调整。

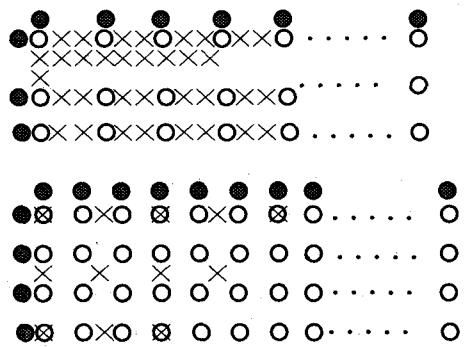


图 2 缩放原理图

### 2.2 缩放器硬件设计

设计框图如图 3 所示。功能模块包括输入模块、行缩放模块、行缓存模块、写控制模块(wr_ctrl)、读控制模块(rd_ctrl)、列缩放模块。

设计中要解决的几个问题:

(1)尾行、尾列边缘像素点在什么情况下需要复制。

(2)对放大/缩小倍数不同时需要的 4 个基点像素值的控制。

(3)浮点坐标系数 $t$ 的控制。

(4)RAM 的存储更新控制。

(5)设计资源消耗最少的数据流插值运算单元。

(6)间断性列插的控制。不同的缩小倍数产生的间断性列插的控制。

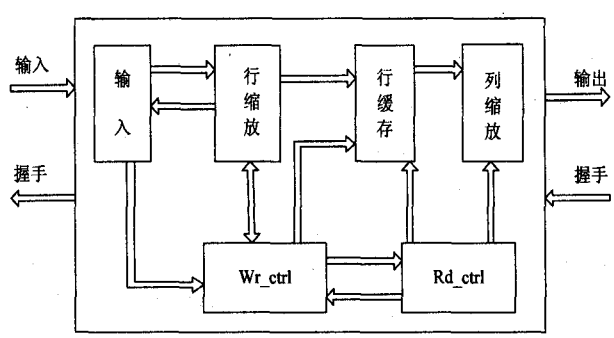



图 3 分步插值图像缩放器结构

输入模块主要用于浮点坐标系数 $t$,和 4 个基点像素值的控制,另外完成和前级的握手及读入原始图像的像素值;行/列缩放模块主要是一个数据流插值运算单元模块,主要用于计算像素值,同时要接收来自 wr_ctrl 模块的满指示信号,控制像素点重发,还要产生像素点的行号指示信号;基于所采用的插值算法的特性,行缓存模块采用 4 个 RAM 进行存储,这种交叉存储的方式控制较复杂,但却最大限度地节省了资源,第三行行插值完成后行列插值即可同时进行,提高了缩放的效率;wr_ctrl 模块主要完成 RAM 的交叉存储更

新,间断性列插控制及尾行/列复制控制功能;rd_ctrl 模块主要功能一是控制浮点坐标系数 $t$,二是从 RAM 中每次读取 4 个像素值,并将顺序整理好连同 $t$ 值一块送给列插值模块,同时也要完成同后级的握手。

输入模块对输入的原始像素点按行依次移位输入,行/列中第一个像素值处 $t=0$,$t$ 值累加产生,每次加放大/缩小倍数的倒数值,当 $t$ 大于等于 1 时,使用其小数部分计算。4 个基点控制随放大倍数不同而不同,$t$ 累加为整数时四个基点为左边一个,右边两个和当下位置的原像素值;当 $t$ 累加为小数时,四个基点平均分布于当下位置两侧。行/列缩放模块采用数据流的设计,运算过程中涉及浮点数和正负数计算,由于硬件浮点运算单元设计比较复杂,设计中自定义小数位宽、整数位宽和符号位,简化了电路,设计中使用移位和加法尽可能代替乘法,简化了电路,共用 5 个乘法器;另外由于插值函数本身的特性,若采样的四个基点为 0,或 255 计算所得的灰度值有可能为负,或超过 8 位所能表示的范围,所以设计中需有一块判断电路用来自纠错。

写控制模块功能是实现 RAM 的交叉存储更新、间断性列插控制及判断尾行/列复制。其中 A=[drop_count * enlarge_dao]-(hang-3),B=[(drop_count * enlarge_dao]。drop_count-已经列插的行数,englarge-放大倍数,englarge_dao-放大倍数的倒数,hang-当时像素点的行号,[ ]-取整数部分。

写控制算法的描述如下所示:

Wr_ctrl Algorithm (enlarge, A, B)

//enlarge 为放大倍数,A 为放大时跳出等待的指示,B 为缩小时跨越所用的行号

```
begin
copy first row memory to RAM1 and RAM2;
if(copy not finish)
    {continue;}
else
  begin
     memory next row to RAM3;
     memory next row to RAM4;
  if((enlarge<=1)&&(row finish))
     {//缩小处理
     if(hang_num! =B)
{memory to RAM1 and wait for lie cha;
if(row memory not finish)
  {continue;}
else
  {交叉存储;}
  //判断是否是最后一行
  if(hang_num==last_hang)
      {copy last row;}
  else
      continue;
}
else
{memory to RAM1 and lie cha ;
{交叉存储;}
}
else if(enlarge>1)
      {//放大处理
      if(A>=1)
        {//判断是否倒数第二行
if(count_hang<width-1)
  {交叉存储;}
else
  { copy last row;
    if(copy not finish)
       continue;
    else
       finish;
}
}
else  {wait for jump current state;}
end
    end
```

行缓存模块是将经行变换处理后的像素值暂时存储起来,待列插时取出。开始时经过判断将第一行中需要存储的像素存入 RAM1 和 RAM2 中,第二行、第三行像素点存入 RAM3 和 RAM4 中,以后各行的像素点按行循环存入 RAM 中。

读控制从 RAM 中取出列插所需的 4 个基点 P1,P2,P3,P4 并控制浮点坐标系数 $t$ 的值,同时要完成和下级的握手。4 个基点及 $t$ 的控制原理同输入模块,这里不再赘述。列插所需像素点都从 RAM 中读取,每次从 4 个 RAM 中分别读出一个像素点,计算时 4 个像素点有一定的顺序,开始时 RAM 号的顺序就是像素点的顺序,以后对读出的像素值需要一个顺序变换电路。顺序变换原理如图 4 所示,若是缩小的情况,由于每个源像素点最多使用一次所以从 RAM1 中读数的同时就可以对其读取的地址更新;若是放大的情况,有些源像素点要使用多次,行 RAM1 读完后不能马上更新,需判断决定何时对 RAM1 更新,待可对 RAM1 更新时 4 个基点的顺序左移,P2,P3,P4 变为 P1,P2,P3,RAM1 中读出的数据为 P4。

基点的顺序变换如图 4 所示。

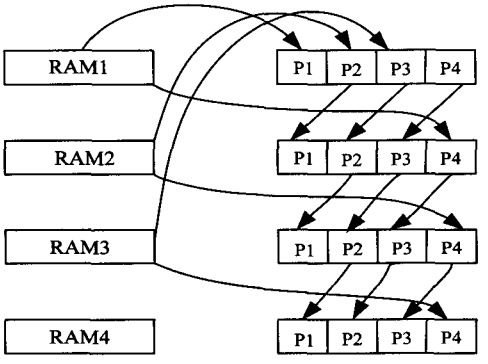



图 4 基点顺序变换图

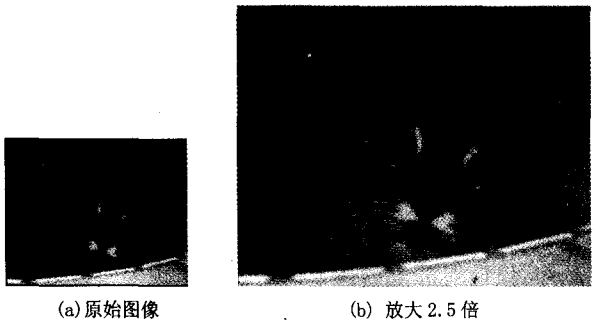
## 3 功能实现

对分步插值图像缩放器结构采用 Verilog 语言描述电路，使用 Synopsys 公司的 VCS 进行功能仿真。仿真结果如表 1 所示：

表 1 仿真结果对比

| 像素点 | 浮点坐标 t | 理论值 | 计算值 | 误差 | 误差百分比 |
|---|---|---|---|---|---|
| (188,163,126,99) | 0.75 | 135.3264 | 135 | +0.3264 | 0.5% |
| (163,126,99,88) | 0.75 | 115.5782 | 115 | +0.5782 | 0.5% |
| (126,99,88,85) | 0 | 99 | 99 | 0 | 0 |
| (99,88,85,87) | 0.25 | 87.324 | 87 | +0.324 | 0.37% |

当 $t=0.75$，而送入的四点分别为 188，163，126，99 时计算的理论像素值为 135.3264，由于像素值为 0 ~ 255 的整数，所以取 135，误差 0.5%；而当 $t=1.75(t>10)$ 时，使用其小数部分 $t=0.75$，此时四个点为 163，126，99，88 计算的理论结果为 115.5782，此处取 115，误差 0.5%；当 $t=0$，而送入的四点分别为 126，99，88，85 时计算的理论像素值为 99，计算值 99，没有误差。设计中使用了 5 个 8bit 乘法器。



(a) 原始图像 (b) 放大 2.5 倍

图 5 缩放效果图

设计中采用 verilog 硬件描述语言，FPGA 的开发环境为 Quartus II 9.0，选用的 FPGA 芯片型号为 ALTERA 公司生产的 Stratix Ⅲ 系列的 EP3SL150F1152C2。参数化设计，可对图像进行任意倍数的放大或缩小，如图5所示，在处理分辨率为163 * 130这幅图像时，将其放大 2.5 倍时整幅图像清晰，边缘不存在马赛克。以上这些验证结果说明了设计中对于边缘像素点的处理是正确的，提出的分步插值图像缩放器结构是正确的。

## 4 结束语

文中针对目前的研究热点图像处理，对其中较重要的缩放问题，提出分步插值图像缩放器结构，并借助 verilog 加以实现，VCS 仿真结果说明了设计中给出的算法的设计思路满足精度的要求，借助 JDK 对处理后的数据进行了转化，结果说明此种电路结构的缩放器性能满足对质量的要求，同时由于设计中对算法的实现采用单独模块化的思路使得设计具有了一定的可移植性。可为高清图像及视频缩放器后续研究提供一定的参考。

**参考文献：**

[1] 林 媛. 图像缩放算法研究及其 FPGA 实现[D]. 厦门：厦门大学，2006.

[2] 蔡泽锋，闾晓晨，郑学仁. 基于行列不同插值算法的图像缩放器引擎的设计[J]. 液晶与显示，2009，24(4)：12-13.

[3] 詹 毅. 自适应图像插值改进算法[J]. 计算机工程，2010，16(8)：21-23.

[4] 徐敏杰. 基于彩色图像势能的非线性缩放算法[J]. 计算机科学，2010，10(10)：45-53.

[5] 申利平，李开宇. 基于误差修正的自适应边缘保持图像插值[J]. 计算机技术与发展，2007，17(2)：116-119.

[6] Singh M，Patel P，Khosla D，et al. Segmentation of Functional MRI by K-means Clustering[J]. IEEE Transactions on Nuclear Science，1996，43(6)：2030-2036.

[7] Ahmed M N. A Modified Fuzzy C-Means Algorithm for Bias Field Estimation and Segmentation of MRI Data[J]. IEEE Transactions on Medical Imaging，2002，21(3)：193-199.

[8] Liew A W C，Yan Hong. An Adaptive Spatial Fuzzy Clustering Algorithm for 3-D MR Image Segmentation[J]. IEEE Transactions on Medical Imaging，2003，22(9)：1063-1075.

[9] Yin Qin. Design and Verification of Digital Video Scaling IP Core[D]. Xi'an：Xi'an Jiaotong University，2006.

[10] 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)：20-24.

[11] 林家鸣，钱 盟，程树林. 基于 Catmull_Rom 插值算法的二维图像的三维显示[J]. 计算机技术与发展，2007，17(10)：17-19.

[12] 张雪阳，黄迪明，张德源. DOS 环境下 BMP 图像的快速旋转及无敌缩放[J]. 计算机应用研究，1999，8(8)：87-89.

[13] 冈萨雷斯，伍 兹. 数字图像处理[M]. 北京：电子工业出版社，2003：50-51.