

文章编号: 2096-1618(2017)06-0601-04

# 基于小波的超分辨率算法研究及 FPGA 实现

陈光拓<sup>1</sup>, 孙 洋<sup>1</sup>, 杜雨谔<sup>1</sup>, 杨学博<sup>2</sup>

(1. 成都信息工程大学电子工程学院, 四川 成都 610225; 2. 电子科技大学, 四川 成都 611731)

**摘要:** 由于超分辨率研究较火热, 且在各个方面应用广泛, 尤其是在家庭娱乐方面需求较大。如果利用纯软件的方法, 处理速度会过慢, 不适用于具体应用。利用基于小波效果较好的超分辨率算法, 提出硬件实现方式, 并给出各部分的实现方法以及设计实现。最后将整体以及各部分仿真, 利用 FPGA 验证其方法的正确性。利用此方法增加图像分辨率可以在每秒 30 帧的情况下处理  $1024 \times 1024$  大小图像, 然而  $1024 \times 1024$  大小的图像通过软件实现则需要较长时间。得到的硬件架构简练, 使其更加容易应用到实际中。通过升级(并行处理)可以处理 4 K 图像。

**关键词:** 超分辨率; 小波; 边缘估计; 边缘加强

**中图分类号:** TN911.73

**文献标志码:** A

**doi:** 10.16836/j.cnki.jcuit.2017.06.005

## 0 引言

图像超分辨率一直是较火热的研究方向。其主要目的是提高其分辨率。利用未知像素点相邻像素点的值估计出某未知像素点的值。超分辨率算法可以被应用于军事卫星遥感雷达成像、医疗图像处理以及家庭娱乐中的 4 K 电视等方面<sup>[1]</sup>。尤其是 4 K 电视的发展导致节目源数目有待提高。图像分辨率的提高主要有以下两种方法: 一种是利用硬件采集设备的升级; 另一种是利用软件处理的方法, 虽然在硬件设备上有着长足的发展, 但是这种方式在升级过程需要较为高昂的升级费用。然而软件的方法相对成本较为低廉, 同时可以得到较为满意的效果。由于现如今采集以及显示设备的像素点越来越多, 使在纯软件上执行速度远远不能满足现有需要。而 ASIC、FPGA 的发展使逻辑资源以及成本都可以满足现有需求。利用逻辑的并行以及流水等方法使处理速度大幅提高很好地解决了纯软件处理速度这一瓶颈。

采用基于小波的超分辨率算法, 提出硬件实现方式, 各部分的实现方法以及设计实现, 并且将整体以及各部分仿真, 最后利用 FPGA 验证其方法的正确性。利用此方法增加图像分辨率可以在每秒 30 帧的情况下处理  $1024 \times 1024$  大小图像, 然而  $1024 \times 1024$  大小的图像通过软件实现速度则只能达到 10 帧/秒。由于得到的硬件架构简练更容易用到实际应用中<sup>[2]</sup>。通过升级(并行处理)可以处理 4 K 图像。

## 1 超分辨率算法

目前提出的超分辨率算法已有很多。可以分为线

性方法和非线性方法。线性方法实现起来较简单, 一般由一个特定的插值的核, 如邻近插值法、双线性插值法以及双立方插值法。另一种为非线性插值法。大后验法(MAP)及由 Sauer 凸集投影法(POCS), 基于亚像元方法是由软件硬件协同方法, 小波算法的超分辨率算<sup>[3]</sup>。

由于小波可以将图像细节细化, 将其分到几个频率下。经过正交小波变换后其低频成分包含大多能量。而高频部分则是图像的细节。将图像利用小波变换分成不同频域进行处理, 后经小波反变换得到高分辨率的图像。

## 2 基于小波变换的超分辨率算法

首先经过二维小波变换将图像分解成 4 个不同频域的图像。二维小波变换是通过对图像的行列进行一维小波变换得到的<sup>[4]</sup>。

$$WT_f(a_0^j, kb_0) = \int_{-\infty}^{+\infty} f(t) \varphi_{j,k}^*(t) dt = \langle f, \varphi_{j,k}(t) \rangle \quad (1)$$

$$A \|f\|^2 \leq \sum_j \sum_k |\langle f, \varphi_{j,k} \rangle|^2 \leq B \|f\|^2 \quad (2)$$

$$f(t) = \frac{1}{A} \sum_j \sum_k WT_f(a_0^j, kb_0) \bar{\varphi}_{j,k} \quad (3)$$

离散小波变换可以由以上公式表示,  $b_0$  为离散化的平移因子,  $a_0$  为离散化的尺度因子。小波分解后的图像重构需要满足公式要求。图像就可以反变换为原图像。

图像使用的双线性插值如图 1 所示, A11-A22 为已知像素点而 P 则是所求像素点。所以可以通过下列公式求出像素点。

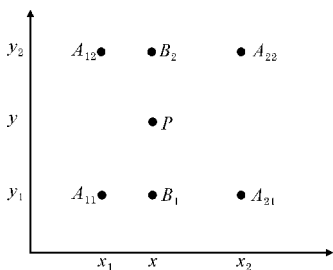


图 1 双线性插值示意图

$$f(B1) = \frac{x_2 - x}{x_2 - x_1}f(A_{11}) + \frac{x - x_1}{x_2 - x_1}f(A_{21}) \quad (4)$$

$$f(P) = \frac{y_2 - y}{y_2 - y_1}f(A_{11}) + \frac{y - y_1}{y_2 - y_1}f(A_{21}) \quad (5)$$

由图及公式可以看出,利用需要得到的像素点  $P$  四周的 4 个像素点就可以求出这一需要得到的像素点<sup>[9]</sup>。

算法整体框架如图 2 所示,具体实现方法:

- (1) 首先对图像进行插值得到高分辨率的图像。
- (2) 再对图像进行小波变换获取 4 个频率成分的图像( LH、HL、HH)。
- (3) 高分辨率的高频图像( LH1、HL1、HH1) 与低频成分( LL) 做小波反变换得到超分辨率图像。
- (4) 根据得到的超分辨率图像加以拉伸,得到最终的超分辨率图像。

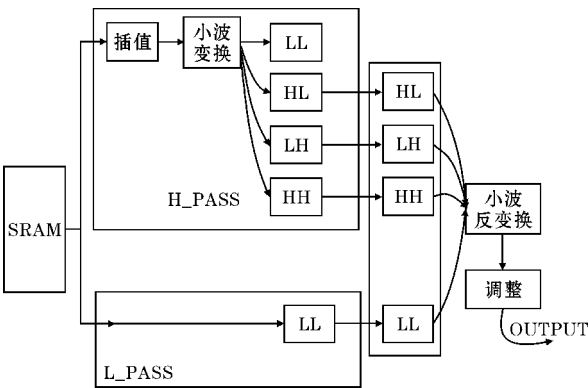


图 2 算法整体框架图

3 算法的硬件实现

由上述算法得到其硬件的实现,主要包括小波变换、反变换及双线性插值。

3.1 小波变换以及反变换硬件架构

小波变换方法有 Mallat 算法或者小波算法, Mallat 算法是在行与列进行交替滤波得到几个频率成分的图像,这种方法计算复杂度相对较高。

小波提升算法的算法复杂度相对较低。电路结构

相对简单,可以节省资源,如图 3 所示。而 5/3 小波变换对图像无损变换。

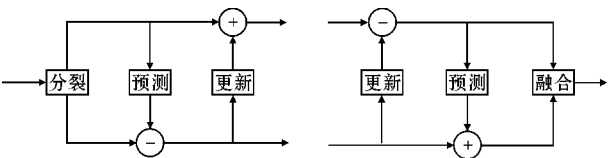


图 3 小波提升算法

提升小波变换可以分为:分裂、预测及更新 3 个步骤。

$$c(2n + 1) = x(2n + 1) + \lfloor \frac{x(2n) + x(2n + 2)}{2} \rfloor$$

$$d(2n) = x(2n) + \lfloor \frac{c(2n - 1) + c(2n + 1) + 2}{4} \rfloor \quad (6)$$

$$x(2n + 1) = x(2n + 1) - \lfloor \frac{x(2n) + x(2n + 2)}{2} \rfloor$$

$$x(2n) = d(2n) - \lfloor \frac{c(2n - 1) + c(2n + 1) + 2}{4} \rfloor \quad (7)$$

$\lfloor \rfloor$  为向上取整,由于二维小波变换是基于上面的提升算法,在行列各做一次提升小波变换<sup>[6]</sup>,其中  $x$  为在行小波变换时为图像的行像素值,在列小波变换的时候为图像的列像素值。 $c(2n + 1)$  为图像的低频部分经过更新得到的, $d(2n)$  为所获得图像的高频部分。由式(6)、(7)可以得到提升小波变换及逆变换。

在变换及逆变换过程中图像的边缘处理采用折叠方法,补齐所缺少的像素点的值。所以可以看出在实现提升算法的小波变换过程中所需要的运算量也较小。

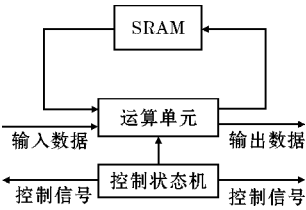


图 4 小波变换硬件框架

图 4 为二维小波算法的硬件架构图,首先通过提升算法做行一维提升小波变换。变换分为运算单元模块以及控制单元模块,控制单元主要判断像素点具体控制,运算单元做出相应运算以及相对应的数据流向。对于行运算后的列运算则利用 SRAM 缓存经过一次小波变换得到相应的数据。

3.2 插值电路硬件架构

利用插值模块实现双线性插值。图 5 为插值电路原理的示意图。

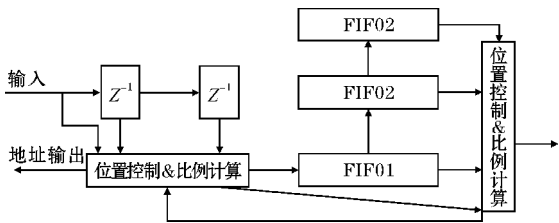


图 5 双线性插值原理图

插值模块主动取每一行的值,通过延时作为前一个像素的保存以此对行插值。对于列的插值利用 FIFO 存储前一行数据<sup>[7]</sup>。

位置控制 & 比例计算模块用来计算该像素点前一个像素点及后一个像素点的比例值。此比例值可以通过产生步长累加以用来输入像素点地址的产生。

由所要求的像素点前一个像素的值、当前像素值以及两个之间的距离(比例),就可以求出横向上所要求的像素点值,再通过纵向求出横向像素值的上一个像素值、当前像素值及两个之间的距离(比例)得到所要求的像素点值。

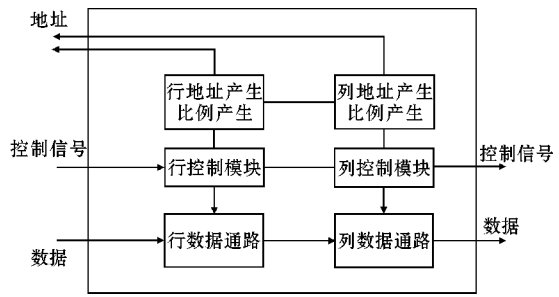


图 6 双线性插值硬件框架

图 6 为插值电路的硬件架构,共分为 6 个模块。其中行地址产生比例产生模块、列地址产生比例产生模块产生输入地址,通过行控制模块及列控制模块控制数据通路流处理数据。

3.3 FPGA 硬件验证架构

利用开发板平台验证仿真结果的正确性,由于平台的条件不足以提供足量的 sram,所以文中利用硬件平台简化 sram 的使用,同时在仿真平台同样处理,对比两个平台处理的数据,以此验证文中仿真结果正确。可以得到所需数据<sup>[8]</sup>。

如图 7 所示,通过 JTAG 将二进制配置文件写入 FPGA 中,同时也将需要处理的数据写到 FPGA 中。再经过需要被验证的系统处理,通过网口与上位机相连传输处理后的数据。最后将仿真数据通过硬件进行处理。

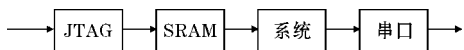


图 7 FPGA 验证系统图

4 仿真结果

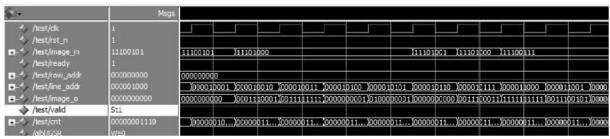


图 8 小波变换仿真图

图 8 为小波变换仿真图,可以看到由输入的图像数据得到输出图像数据的低频及高频分量。在图中输出中可以看到间隔的输出高频及低频像素点。

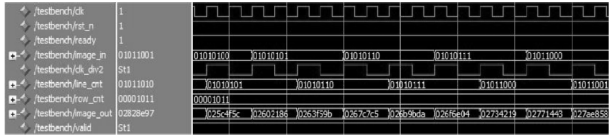


图 9 插值模块仿真图

图 9 中为插值电路的仿真图,可以看出输入信号由插值电路直接取得,通过计算得到输出像素点的值。

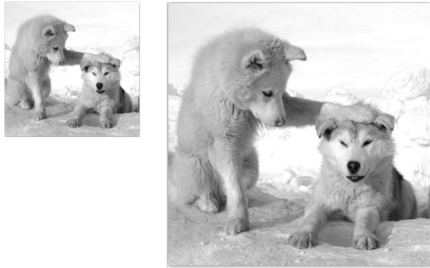


图 10 插值模块效果图

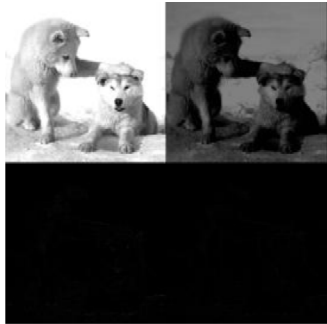


图 11 小波模块效果图

图 10 为双线性插值效果图,图 11 为小波变换的效果图。可以看出硬件模块可以很好地完成相应的运算得到所需要的结果。最后得到超分辨率的图像,如图 12 所示。结果表明,利用文中算法及硬件电路可以较为优秀的完成超分辨率过程。



图 12 最终处理效果图

利用文中方法得到的图像更加清晰,细节成分更加明显,得到的图像效果优秀。

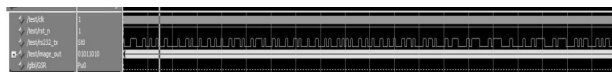


图13 插值输出和串口输出的FPGA仿真图



图14 chipscope抓取的串口值

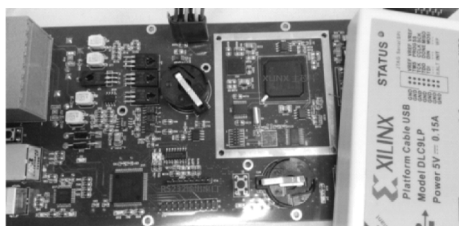


图15 算法实现的硬件电路板

从图13可以看到插值输出的信号及串口输出的信号 modesim 仿真。经过验证其结果正确。

图14为chipscope抓取串口通信信号,由上位机串口助手接收后与仿真的数据对比,验证了其硬件上实现的正确性。图15为最后实现的硬件平台。

## 5 结束语

所提出的基于小波的超分辨率算法,不仅可方便于硬件实现,还可以在每秒30帧的情况下处理 $1024 \times 1024$ 大小图像,然而利用软件处理需要更长的时间(在AMD A6-3430MX处理器4GB内存的pc平台采用Matlab2014完成算法需要38 min)。经过其改进,加入并行处理,可以在满足30 f/s视频流下处理4 K图像。

## Research and FPGA Implementation of Super-resolution Algorithm based on Wavelet Transform

CHEN Guang-tuo<sup>1</sup>, SUN Yang<sup>1</sup>, DU Yu-ming<sup>1</sup>, YANG Xue-bo<sup>2</sup>

(1. College of Electronic Engineering, Chengdu University of Information Technology, Chengdu 610225, China; 2. University of Electronic Science and Technology of China, Chengdu 611731, China)

**Abstract:** The research of super-resolution is hot and it is widely used in many fields, especially in family entertainment. If you use a pure software approach, the processing speed will be too slow, so it is not suitable for specific applications. In this paper, a method of hardware implementation is proposed based on the super-resolution algorithm with better effect, and the implementation method and design implementation of each part are given. Finally, the whole and each part of the design are simulated, and the correctness of the method is verified by FPGA. This method can process  $1024 \times 1024$  size images at 30 frames per second when increasing image resolution. However,  $1024 \times 1024$  size images require a much more longer time by software implementation. The hardware architecture of this article is concise, making it easier to apply to practice. 4 K images can be processed by upgrading (parallel processing).

**Keywords:** super resolution; wavelet; edge enhancement; interpolation

## 参考文献:

- [1] 钟宝江,陆志芳,季家欢.图像插值技术综述[J].数据采集与处理,2016,31(6):1083-1096.
- [2] 刘刚,戴明.基于区域分割自适应的超分辨率算法[J].微电子学与计算机,2012,29(1):76-79.
- [3] 汤晓莉,韩睿,郭若杉.基于边缘指导的快速动态超分辨率算法[J].微电子学与计算机,2016,33(10):167-172.
- [4] Akbarzadeh S, Ghassemian H, Vaezi F. An efficient single image super resolution algorithm based on wavelet transforms[C]. Iranian Conference on Machine Vision and Image Processing. IEEE, 2015: 111-114.
- [5] 公茂法,王志文,于江,等.基于FPGA小波变换核的设计[J].电测与仪表,2014,51(8):103-106.
- [6] 赵阳,李国林,谢翔,等.一种邻域嵌入超分辨率算法[J].微电子学与计算机,2017,34(2):10-14.
- [7] 谭会生.多级二维整数小波变换的FPGA实现研究[J].电子测量与仪器学报,2009(7):29-34.
- [8] 刘怡,黄自力,王经纬,等.FPGA双线性插值图像变换系统的设计与实现[J].中国测试技术,2008,34(3):65-67.
- [9] Matsuo Y, Iwasaki S, Yamamura Y, et al. Wavelet domain image super-resolution from digital cinema to ultrahigh definition television by dividing noise component[C]. Visual Communications and Image Processing (VCIP), 2012 IEEE. IEEE, 2012:1-6.