

## 基于并行运算的双层图像锐化方法

张巍<sup>1\*</sup>, 贺星<sup>1</sup>, 霍颖翔<sup>1</sup>, 滕少华<sup>1</sup>, 滕毅<sup>2</sup>, 李日贵<sup>1</sup>

(1. 广东工业大学 计算机学院, 广州 510006; 2. 香港理工大学 纺织及成衣系, 香港 999077)

(\* 通信作者电子邮箱 weizhang@gdut.edu.cn)

**摘要:**针对低清晰度照片或图像放大后边界模糊、画质差及人们对高清图像的实际需求,基于统一计算架构(CUDA)环境,提出了一个两层结构的图像并行锐化方法,设计并实现了一个基于GPU的并行锐化算法:第一层采用并行线性插值法,反复对图像非边界部分进行计算以及边缘区域锐化处理;第二层采用改进的梯度法对图像进一步优化。放大后的图像经该方法处理后,基本上可消除图像边缘区域的锯齿,使图像画质平滑、自然、清晰。经实验验证,设计的基于GPU的并行锐化算法在效率和画质上都优于目前流行的算法,提出的方法可应用于现有图像及照片放大后处理。

**关键词:**图像放大; 图像锐化; 并行设计; 双层锐化算法; 梯度优化

**中图分类号:** TP391.41 **文献标志码:** A

### Bi-level image sharpening method based on parallel computing

ZHANG Wei<sup>1\*</sup>, HE Xing<sup>1</sup>, HUO Yingxiang<sup>1</sup>, TENG Shaohua<sup>1</sup>, TENG Yi<sup>2</sup>, LI Rigui<sup>1</sup>

(1. School of Computer Science and Technology, Guangdong University of Technology, Guangzhou Guangdong 510006, China;

2. Institute of Textile and Clothing, The Hong Kong Polytechnic University, Hong Kong 999077, China)

**Abstract:** A parallel bi-level image sharpening methodology in Compute Unified Device Architecture (CUDA) circumstance was proposed especially for the improvements on fuzzy boundaries and poor quality when enlarging low resolution photos or images. A GPU-based parallel sharpening algorithm with two stages was designed and implemented. Firstly, the parallel linear interpolation algorithm was repeatedly adopted by the calculation of non-edge region and the sharpening treatments of edge area. Secondly, an improved gradient method was utilized for the further optimized images. The jagged edges of the enlarged images were basically eliminated by the proposed method, making the images much more smooth, natural, and legible. The experimental results prove that the GPU-based parallel image sharpening algorithm is superior to the currently popular algorithms in calculation efficiency and image quality, and it can be applied in sharpening images and amplifying photos.

**Key words:** image zooming; image sharpening; parallel design; bi-level sharpening algorithm; gradient optimization

## 0 引言

随着大版面海报、广告的日益流行,高清显示器、蓝光技术的发展,以及计算机处理能力的日新月异,人们对高清影像的要求越来越高。与此同时,高清晰度的素材却不近人意,除现有的一大批非高清晰度的素材外,人们还在大量产生非高清晰度的素材。尤其对于许多无线终端,其分辨率低、画面小,经放大后,有些画面会产生严重失真。这个问题一直在困扰着人们,限制了图像编辑者的取材范围,导致一些大版面的印刷品图像不清晰或边缘出现毛刺现象,影响了画面的美观和质感。

图像锐化是通过补偿图像轮廓、突出图像的边缘信息以使图像更加清晰,即图像锐化的实质是增强原始图像的高频分量,同时抑制、降低甚至剔除图像的噪声。但常见的图像锐化算法在放大原始图像高频分量的同时,也会伴随着噪声的放大,如何处理噪声就成为图像锐化算法的主要研究问题。人们已经开发了许多图像锐化算法,如:线性插值法<sup>[1-3]</sup>、非线性逆扩散方程去噪法<sup>[1,4-5]</sup>、拉普拉斯锐化算法<sup>[6]</sup>、反锐化

掩模法<sup>[7-8]</sup>、马尔可夫随机场法<sup>[9]</sup>以及量子遗传算法<sup>[10]</sup>等。

最邻近插值法<sup>[1]</sup>是一种简单、高效的方法,具体操作如下:对任意放大后未知的像素点 $P$ ,在原始图像上找到其对应点 $Q$ ,以曼哈顿距离计算出 $Q$ 点的最邻近像素点 $A、B、C、D$ ,取 $P$ 点的像素值为最靠近的邻近点像素值。该方法通过简单复制原始图像像素来获得放大效果,放大后的图像有明显马赛克效应,像素值会出现突然跳跃,导致图像明显失真。双线性插值法<sup>[2]</sup>可以避免方块效应,使放大后的图像更平滑,视觉效果相对较好,且运算量较小,容易实现;缺点是应用双线性插值法放大的图像边缘会模糊,且随着放大倍数增加模糊程度将加剧,从而影响图像效果。双立方插值法<sup>[4]</sup>被插值点的值是由16个邻近点计算获得的,由它得到的图像比双线性插值的结果更平滑。双立方插值算法与双线性插值算法的本质区别不仅在于扩大了影响点的范围,还采用了高级的插值算法——非线性插值算法。双立方插值算法在现有的一些常用图像处理软件中得到了应用,如Photoshop、After Effects、Avid、Final Cut Pro等。小波双立方配比插值法<sup>[4]</sup>比双立方插值法含有更多的高频成分,其消除马赛克效果比双立方插值更好,

收稿日期:2013-03-01;修回日期:2013-05-08。

**基金项目:**国家自然科学基金资助项目(61273036);教育部重点实验室基金资助项目(110411);广东省自然科学基金资助项目(10451009001004804,9151009001000007);广东省科技计划项目(2012B091000173);广州市科技计划项目(2012J5100054)。

**作者简介:**张巍(1964-),女,江西萍乡人,副教授,硕士,CCF会员,主要研究方向:数据挖掘、网络安全、协同计算、数学建模;贺星(1987-),男,湖南株洲人,硕士研究生,主要研究方向:算法设计、图像处理;霍颖翔(1989-),男,广东广州人,主要研究方向:算法设计与实现;滕少华(1962-),男,江西南昌人,教授,博士,CCF会员,主要研究方向:协同计算、数据挖掘、网络安全;滕毅(1986-),女,江西南昌人,博士研究生,主要研究方向:数据建模、计算机仿真;李日贵(1989-),男,广东茂名,主要研究方向:算法设计。

但边缘仍存在锯齿现象。文献[5]采用 B-样条插值法使物体的轮廓更加清晰,边缘的锯齿现象大大减弱,图像感受更为自然。郝玉峰等<sup>[6]</sup>将各向异性逆扩散算法用于指纹图像的锐化去噪,克服了非线性扩散方程对大曲率边缘点的模糊效应,兼顾了去噪和保持边界这一矛盾的两方面。张玲等<sup>[7]</sup>提出了将粗糙集和小波反锐化掩模相结合的方法来实现图像增强,运用粗糙集中的近似及等价属性关系将知识化后的图像划分成不同的区域,再根据估计好的阈值进行数据约简,实现噪声的有效去除;运用小波反锐化掩模法对图像的轮廓及细节信息进行增强处理,完成图像的最终增强;最后用实验检验了算法的效果。王利平等<sup>[8]</sup>采用受限拉普拉斯锐化算法,有效地抑制了图像的噪声加强,使图像边缘清晰,保护了图像的细节。刘芳等<sup>[9]</sup>将 Thin Plate 先验能量函数引入到反锐化掩膜方法中,给出了一种基于马尔可夫随机场理论的图像锐化方法,并用实验证明了该方法能有效抑制噪声,突出图像边缘细节。姜桂圆等<sup>[11]</sup>采用尺度不变特征变换方法提取局部特征点、生成局部特征描述器,研究了数据分配、特征调整、数据块划分与数据发送方式,给出了基于消息传递机制的并行图像处理与优化策略,并用实验说明了算法的加速效果。王刚等<sup>[12]</sup>提出了基于脊小波 (ridgelet) 变换域的模糊自适应图像增强算法,利用脊小波变换在表示图像线性奇异边缘时具有独特的优越性,达到突出边缘和抑制噪声的目的,并用实验验证了算法的效果。钱惟贤等<sup>[13]</sup>给出了一种既可以对图像进行锐化滤波,又不降低图像信噪比的算法,该算法采用模式识别的相关理论,基于隶属度和概率松弛技术对红外图像中由真实边缘和由各种噪声引起的亮度数值变化进行区分,对不同区域采用不同的锐化处理,并用实验证明该算法的效果。Xu 等<sup>[14]</sup>通过从不同缩放率中取得的锐化参数来修改图像缩放所需的像素值,在近似梯度方法上采用简单的高斯分布来保障运算速度,从而锐化放大后的图像。Pattanasethanon 等<sup>[15]</sup>把直方图均衡化和边缘检测程序应用到原始图像,在锐化过程中,均值、方差、噪声率信号被用来对明亮度、分辨率做统计参数,经验证,输出的图像亮度和对比度得到增强,最后该文指出这种方法适合历史图像、卫星图像和捕获图像三类。Lee 等<sup>[16]</sup>提出了一个新颖的锐化掩模技术,该技术在锐化图像时,既不使图像失真又不会放大噪声,给出的算法模仿了人类可视化系统的特点,用于控制锐化的级别,同时使用能量分散过滤来排除图像色彩失真,最后通过实验验证了算法的效果。Nooraliei 等<sup>[17]</sup>利用 IJA 随机学习自动机检测噪声和  $\alpha$  参数调优,该参数被用于大气扩散模型图的锐化。该方法被用于自动机和自适应方式的灰度图处理时,表明 IJA 自动机能检测到噪声并用邻近点的相应特征值代替,作局部图像恢复,最后作者通过实验给出了与采用其他方法作图像锐化的比较。Tseng 等<sup>[18-19]</sup>使用 Hermite 插值法设计分数延迟滤波器,给出了两种基于 Hermite 插值的分数延迟滤波器的实现方法,而滤波器被广泛用于数字合成和图像锐化过程的去噪;另外,用分数阶导数和 Mach 带效应进行图像锐化,使用 Grünwald-Letnikov 导数设计分数阶数字微分器,进而将 Mach 带效应和分数阶数字微分器用于构造图像锐化算法,最后用例子表明了方法的有效性。Gui 等<sup>[20]</sup>提出了一个增强图像的空间局部对比度的非线性模块,该方法利用线性反锐化掩模技术和非抽样离散模糊逻辑来降低噪声的敏感程度,获得了感知愉悦的结果。进而,Gui 等<sup>[21]</sup>将非线性模块结合图像的灰度差异用于 X-射线图像的锐化,用实验表明了方法的效果。

由于单层锐化效果不理想,本文提出了一个两层结构的图

像并行锐化方法,设计并实现了一个基于图形处理器 (Graphic Processing Unit, GPU) 的并行锐化算法,该算法第一层采用并行线性插值法,反复对图像非边界部分和边缘区域进行锐化处理,第二层采用改进的梯度法对图像进一步优化。这既可以使图像的边缘趋于平滑、自然,减少模糊,又不致平滑过渡区域产生色调分离;另外,由于采用 GPU 的行级并行技术进行算法设计,本文提出的方法具有高效性。实验表明:本文提出的算法在图像质量和减少噪声方面取得了较好的效果。

## 1 双层图像并行锐化方法

针对现有图像锐化算法存在的问题,为了降低噪声干扰,提高图像锐化速度,本文提出了一个双层图像并行锐化方法,并设计、实现了相应的算法。该方法的第一层对图像进行并行锐化,第二层则在锐化的基础上进行优化。

### 1.1 双层并行锐化算法体系结构

基于并行运算的双层图像放大锐化算法的体系结构如图 1 所示:第一层采用线性插值法,提取原图像所有像素点,计算相似度,以原图像每一像素点为中心作  $n$  条切线,求出拟合度最高的切线,以此对图像进行锐化;第二层采用改进的梯度法对边界进行优化和锐化处理。整个锐化过程采用并行计算设计。

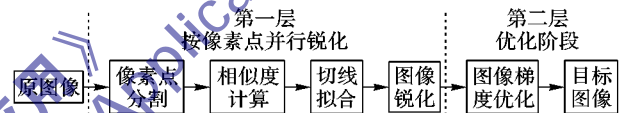


图1 双层图像并行锐化算法体系结构

### 1.2 相似度与拟合切线

图像锐化的关键是边缘信息清晰,在同一颜色区域中,图像上各点 RGB 三通道的值是相同或相近的。本文的相似度计算正是基于此原理。

提取原图像的所有像素点,通过各点 RGB 对应通道的值来确定点与点之间的相似度。具体如下:

**定义 1** 图像上任意两点  $P$  和  $Q$  的绝对相似度定义为 RGB 各分量之差的绝对值之和。

**定理 1** 设图像上任意两点  $P, Q$  的 RGB 分别为  $(x_1, y_1, z_1)$  和  $(x_2, y_2, z_2)$ , 则其绝对相似度  $AbsSim$  按下式计算:

$$AbsSim = |x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2| \quad (1)$$

**性质 1**  $AbsSim$  的取值范围为  $[0, 765]$ 。

**证明** 设  $P, Q$  为图像上任意两点,其 RGB 分别为  $(x_1, y_1, z_1)$  和  $(x_2, y_2, z_2)$ 。因为每一通道的取值范围为  $[0, 255]$ , 则有:

$$0 \leq x_1, y_1, z_1 \leq 255 \quad (2)$$

和

$$0 \leq x_2, y_2, z_2 \leq 255 \quad (3)$$

由式(1)、(2)和(3)立即得证。

**性质 2** 最大绝对相似度为  $\text{Max}(AbsSim) = 765$ 。

考虑到  $AbsSim$  变化范围较大,为处理方便,本文采用相对相似度来计算。

**定义 2** 图像上任意两点  $P$  和  $Q$  的相对相似度定义为绝对相似度与最大绝对相似度之比。

**定理 2** 设图像上任意两点  $P, Q$  的 RGB 分别为  $(x_1, y_1, z_1)$  和  $(x_2, y_2, z_2)$ , 则其相对相似度  $RelSim$  按下式计算:

$$RelSim = 1 - \frac{P \text{ 和 } Q \text{ 点的绝对相似度}}{\text{最大绝对相似度}} = 1 - \frac{|x_1 - x_2| + |y_1 - y_2| + |z_1 - z_2|}{3 \times 255} \quad (4)$$

由定理 2,可以得到如下性质:

性质 3  $RelSim$  的取值范围为 $[0, 1]$ 。

性质 4 两点之间的颜色相近与  $RelSim$  之值成正比。

由定理 2 和性质 4, 不难得到:

- 1) 当  $RelSim$  越大, 两点之间的色差越小, 近似程度越高;
- 2) 当  $RelSim = 1$  时, 两点的颜色相同;
- 3) 当  $RelSim = 0$  时, 两点颜色相异度达到最大。

定义 3  $r$  为过图像上任意点  $P$  的切线,  $r$  的拟合度定义为  $r$  上点  $Q(Q \neq P)$  与  $P$  点的最大相对相似度和最小相对相似度之差。

定理 3 过图像上任意点  $P$  的切线为  $r$ ,  $r$  的拟合度  $Fitting$  按下式计算:

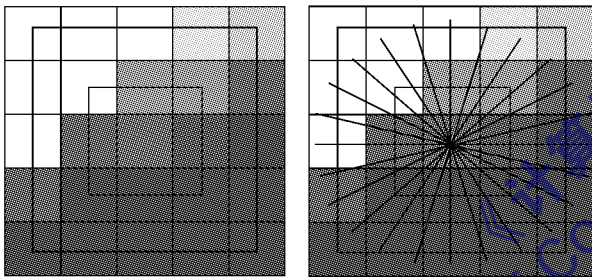
$$Fitting = \text{最大相对相似度} - \text{最小相对相似度} = \text{Max}(RelSim) - \text{Min}(RelSim) \quad (5)$$

定义 4 图像上任意一点  $P$  的拟合切线定义为拟合度  $Fitting$  取最小值所在的切线。

由定义 4, 拟合切线刻画了与点  $P$  拟合程度最高的切线, 则  $P$  点的颜色可以取拟合切线上的点作近似。

### 1.3 基于像素点的并行锐化

以每个像素点为中心, 过该点以相等间隔向  $n$  个方向作切线, 如图 2 所示, 每点的切线方向均为  $[0, \pi]$  之内的数, 在切线上取若干个, 计算其相对相似度  $RelSim$  和拟合度  $Fitting$ , 求取拟合切线, 如图 3 所示, 进而以此对图像进行锐化。



(a) 原图像和像素点 (b) 像素点与过该点的 12 条切线  
图 2 原图像、像素点和切线

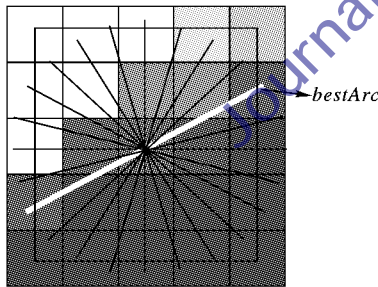


图 3 拟合切线

锐化算法: 以每一个像素点为中心, 平均取  $n$  (如  $n = 12$ ) 个方向作  $n$  条切线, 计算该像素点的切线方向与拟合度。

输入 像素点和过该像素点的每条切线上  $m$  (如  $m = 10$ ) 个点的 RGB 值, 最大取样直径距离默认设定为 4.5 个像素, 可按需求重新设定。

输出 图像各像素点的切线方向与拟合切线。

算法步骤如下:

- 1) 任取一像素点  $P$ ;
- 2) 设: 过  $P$  点的切线方向  $i = 0$   
最佳拟合率  $bestRelSim = 0$   
最差拟合率  $worstRelSim = 1$   
最佳拟合角度  $bestArc = 0$   
 $Fitting = 0$   
 $bestFitting = 1$

- 3) Repeat
  - ①取角度  $i$ , 对以角度为  $i$  的切线 Repeat
    - a) 取切线上一样本点  $Q$ , 计算  $Q$  点与  $P$  点的相对相似度  $RelSim$
    - b) If  $RelSim > bestRelSim$  then  
 $bestRelSim = RelSim$ ;
    - If  $RelSim < worstRelSim$  then  
 $worstRelSim = RelSim$ ;
    - c) 取下一个样本点  $Q$
  - Until 没有下一个样本点
  - ②  $i = i + \frac{\pi}{16}$
  - ③  $Fitting = bestRelSim - worstRelSim$
  - ④ 保存  $Fitting$
  - ⑤ if  $Fitting < bestFitting$  then  $bestFitting = Fitting$
- 4) Until  $i \geq \pi$
- 5) 得到所有像素点的拟合度  $Fitting$  以及拟合度为  $bestFitting$  的拟合切线

### 1.4 梯度优化和锐化处理

本文采用改进的梯度法对未知像素进行插值, 目标是使图像边界趋于平滑, 最大限度地减少图像的模糊性, 同时避免锐化处理区域产生色调分离。

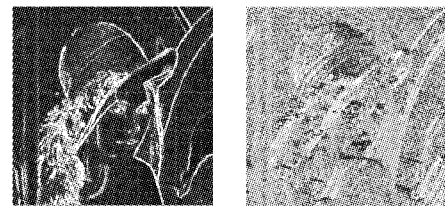
改进的梯度算法: 输入原图像, 每线程处理若干像素点, 参考切线向量场计算, 得到目标像素点向量场; 判断目标点取色偏移方向及大小, 得到目标点取色, 生成目标图。

输入 原图像的每像素点 RGB 值, 每线程处理若干像素点;

输出 目标生成色, 放大锐化后的图像。

算法步骤如下:

- 1) 首先用原图像像素点  $P$  邻近的  $s$  条 (如 4 条) 切线通过线性插值求出目标点的切线方向以及拟合切线。
- 2) 依据拟合切线方向, 求出其法线方向  $n$ , 然后通过拟合度, 确定取样距离, 在经过目标点的法线上, 找到距离目标点取样距离最近的两个点  $r_1$  和  $r_2$ , 通过线性插值分别求得其颜色。取目标点通过线性插值后的颜色为  $P_c$ 。拟合度图及方向图见图 4。



(a) 拟合度图 (b) 方向图  
图 4 拟合度和方向图

3) 求出未知像素点到颜色渐变线分界点距离 (图 5 中  $AF$  法线方向计算公式如下):

$$arc = AC * \sin \left( \frac{1}{AC} \left[ (AB + BC * \tan(bestArc)) * |\cos(bestArc)| \right] \right) \quad (6)$$

其中: 点  $C$  为原图像像素点;  $A$  为放大后图像的未知像素点。

4) 用未知像素点  $A$  到颜色渐近线分界点距离, 以及  $r_1, r_2$  与  $P_c$  的颜色相似度, 可以判断出目标点取色偏移方向。假如切线左上角像素的原图颜色与切线所在的原图像素的颜色相似程度较高, 而与右下角的颜色相差较大, 则切线需要往右下偏移。

5) 如切线需要往右下偏移, 此时切线需要计算像素的右

下方,故此像素判定为属于切线左上方的范围,它的颜色需要从更左上方的位置取。具体要往左上偏移多少度取色则取决于拟合度。

6)对每一个像素均进行以上处理。由于边缘地方拟合度较大,  $r_1, r_2$  样本点取值距离也比较大,进行的压缩也较大;对非边缘的地方,拟合度很小,  $r_1, r_2$  样本点取值距离也比较小,压缩也较小。从而边缘切线方向会更加平滑,法线方向会更加锐化,将获得更适宜于视觉清晰度的边界效果,非边缘地方也能得到相应的优化。

式(6)的证明如下:

$$\text{证明 } (AB + BC * \tan(\text{bestArc})) = AE$$

$$AE * |\cos(\text{bestArc})| = AD$$

$$a \cos(AE/AC) = \beta$$

$$AC * \sin \beta = CD$$

$$AF = CD$$

注意:在颜色计算过程中,当求得的坐标为非整数时,一律使用线性插值以获得更加近似的颜色。

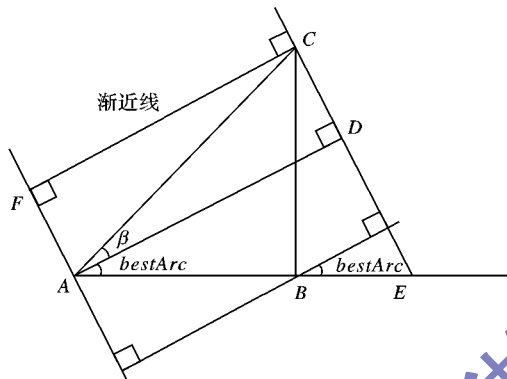


图 5 AF 法线方向

### 1.5 算法的并行设计

并行算法设计如下:

- 1)为每个线程分配若干像素点的处理任务;
- 2)每个 GPU 运行数千个并发线程,可处理一张图片;
- 3)本文提出的算法采用统一计算架构(Compute Unified Device Architecture, CUDA)开发。

4)CUDA 线程管理大致分为三层, grid、block (块)和 thread(线程)。grid 中有若干个 block,每个 block 中有相同数量的 thread。对于运算能力为 1. x 的设备,block 最高可支持二维,thread 最高可支持三维;对于运算能力 2. x 或以上的设备,block 的最高维度提升至三维。

图 6 是本程序使用的结构: block 和 thread 均采用二维。这个结构可以刚好将图片切割成若干等份,这样的分割就可采用并行设计来实现计算。

### 1.6 图像质量评价

参照常用的图像相似度测量方法<sup>[21-24]</sup>,需要计算均方误差(Mean Squared Error, MSE)<sup>[21-23]</sup>、峰值信噪比(Peak Signal-to-Noise Ratio, PSNR)<sup>[21-23]</sup>、模糊系数  $K_{\text{blur}}$ <sup>[24]</sup> 和质量指标  $Q$  值<sup>[21]</sup>,其公式如下:

$$MSE = \frac{\sum_{i=1}^M \sum_{j=1}^N [g(i, j) - f(i, j)]^2}{M \times N} \quad (7)$$

$$PSNR = 10 \lg \left[ \frac{255^2 \times M \times N}{\sum_{i=1}^M \sum_{j=1}^N [g(i, j) - f(i, j)]^2} \right] \quad (8)$$

$$Q = \frac{4\sigma_{xy}}{(\sigma_x^2 + \sigma_y^2)[(\bar{x})^2 + (\bar{y})^2]}; Q \in [-1, 1] \quad (9)$$

$$\begin{cases} \bar{x} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N x_{ij} \\ \bar{y} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N y_{ij} \end{cases} \quad (10)$$

$$\begin{cases} \sigma_x^2 = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \bar{x})^2 \\ \sigma_y^2 = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (y_{ij} - \bar{y})^2 \end{cases} \quad (11)$$

$$\sigma_{xy} = \frac{1}{MN} \sum_{i=1}^M \sum_{j=1}^N (x_{ij} - \bar{x})(y_{ij} - \bar{y}) \quad (12)$$

其中:  $M$  和  $N$  分别是图像长度和宽度上的像素个数,  $f(i, j)$  和  $g(i, j)$  分别是原始图像和处理后的图像在点  $(i, j)$  处的灰度值。均方误差和峰值信噪比是用于比较被评价图像与原图像质量的参数,峰值信噪比数值越大,图像质量越好<sup>[22]</sup>;  $K_{\text{blur}}$  为输出的边缘能量和输入边缘能量比值,  $K$  值越接近 1, 图像清晰度越高,结果表现越鲜明<sup>[24]</sup>;  $Q$  值越接近 1, 图像质量越好,一般  $Q$  不小于 0<sup>[21]</sup>。

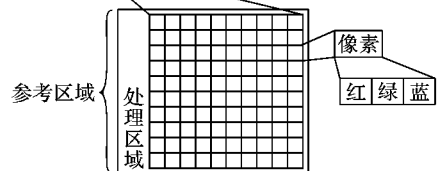
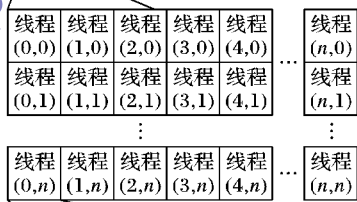
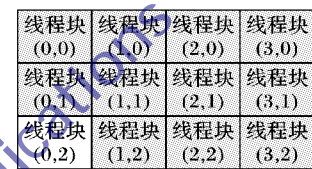


图 6 并行线程示意图

## 2 实验及结果分析

本文算法已在 Microsoft Visual Studio 2010 + CUDA + Parallel Nsight 开发平台进行了一组实验。

实验 1 CPU + GPU 与纯 CPU 方式比较。

实验数据说明:

1)原图像采用分辨率为 512 × 512 像素,颜色深度为 24 位真彩的图片。

2)放大后的目标图片分辨率有两种: 8 192 × 8 192 和 1 024 × 1 024 像素。

3)硬件配置: ① GPU 显卡: GeForce550Ti, GeForce 9600GT; ② CPU: Core 2 Quad Q6600 3.0 GHz OC, AMD 速龙 64 X2 6000 +。

测试结果如表 1 所示,可以看出,采用本文的方法,其数据速度比仅采用 CPU 的处理效率提高了 2 个数量级,最大增速倍数接近 100 倍。

实验 2 本算法与类似算法比较实验数据说明:

1)原始图像文件为 Lena. bmp, 112 × 104 像素。

2)图像放大 16 倍后, 1 792 × 1 664 像素。

对原始图像分别采用直接放大、双线性插值(Bilinear)、

双立方插值(Bicubic)和本文算法四种插值方法放大 8 倍后,经各种算法的锐化处理,得到 5 个 1 792 × 1 664 的图像及其局部截图,如图 7 所示,其效果说明如表 2 所示。从图 7 及表 1、2 分析可知,本文算法在效率上有优势,在图像画质上 Bilinear 和 Bicubic 比本文方法更接近 S-Spline 方法。

表 1 本文 CPU + GPU 方法与 CPU 方法耗时比较 ms

硬件设备	图像	
	512 × 512 放大到 8 192 × 8 192	512 × 512 放大到 1 024 × 1 024
GPU GetForce550Ti	2 153	3 206
GPU GeForce 9600GT	8 112	12 199
CPU Intel Core2 Quad Q6600 3.0 GHz OC	57 221	227 684
CPU AMD 速龙 64 X2 6000 +	71 200	306 625

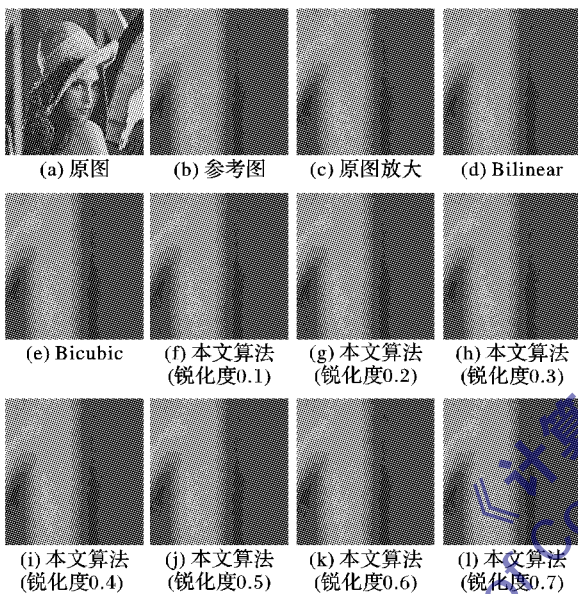


图 7 本文算法与类似算法效果比较

表 2 以 S-Spline 算法作参考进行比较

算法	PSNR/dB	$K_{Blur}$	Q
最邻近点插值算法	38.166 1	1.356 0	0.374 1
Bilinear 算法	51.355 3	0.939 9	0.875 2
Bicubic 算法	52.573 0	1.030 2	0.910 7
本文算法锐化度 0.1	49.579 6	1.211 3	0.831 0
本文算法锐化度 0.2	49.573 0	1.212 8	0.830 6
本文算法锐化度 0.3	49.571 5	1.212 0	0.829 9
本文算法锐化度 0.4	49.595 0	1.209 2	0.831 6
本文算法锐化度 0.5	49.573 9	1.214 1	0.830 6
本文算法锐化度 0.6	49.542 3	1.213 8	0.829 9
本文算法锐化度 0.7	49.399 5	1.221 2	0.829 1

### 3 结语

针对低清晰度照片或放大后边界模糊、画质差的图像,考虑人们对高清图像需求实际,利用 GPU 的并行运算能力,本文提出了一个双层结构的图像并行锐化方法,设计并实现了一个并行锐化算法。第一层采用并行线性插值法,反复对图像非边界部分进行计算以及对边缘区域进行锐化处理;第二层采用改进的梯度法对图像作进一步优化。放大后的图像经本文算法处理后,基本上可消除图像边缘区域的锯齿,使图像画质平滑、自然、清晰。经实验验证,本文设计的基于 GPU 的并行锐化算法,在效率和画质上都优于目前流行的算法。

### 参考文献:

- [1] 章毓晋. 图像工程: 中册[M]. 北京: 清华大学出版社, 2005.
- [2] CASTLEMAN K R. 数字图像处理[M]. 朱志刚, 林学闫, 石定机, 等译. 北京: 电子工业出版社, 2002.
- [3] 郭海霞, 郭海龙, 解凯. 基于边缘信息改进的双线性插值算法[J]. 计算机过程与应用, 2011, 47(31): 171 - 174.
- [4] 浦利, 金伟其, 刘玉树, 等. 基于小波双立方插值的图像插值放大算法研究[J]. 红外技术, 2006, 28(8): 453 - 455.
- [5] 李开宇. 基于 B-样条插值的图像边缘检测[J]. 南京航空航天大学学报, 2007, 39(2): 198 - 203.
- [6] 郝玉峰, 袁春伟. 基于各向异性逆扩散方程的指纹图像锐化去噪方法[J]. 数据采集与处理, 2005, 20(3): 258 - 262.
- [7] 张玲, 黄粉平, 郑恩让. 基于粗糙集与小波反锐化掩模的图像增强[J]. 光子学报, 2008, 37(6): 1285 - 1288.
- [8] 王利平, 陈钱, 顾国华, 等. 基于 DSP + FPGA 的红外图像锐化算法的实现[J]. 南京理工大学学报, 2006, 30(6): 764 - 766.
- [9] 刘芳, 刘东升, 桂志国. 基于马尔科夫随机场的图像锐化方法[J]. 计算机工程, 2011, 37(1): 224 - 225, 228.
- [10] 邵明省, 李伟. 基于改进量子遗传算法的图像锐化研究[J]. 激光与光电子学进展, 2012, 49(1): 56 - 62.
- [11] 姜桂圆, 张桂玲, 张大坤. SIFT 特征分布式并行提取算法[J]. 计算机研究与发展, 2012, 49(5): 1130 - 1141.
- [12] 王刚, 肖亮, 贺安之. 脊小波变换域模糊自适应图像增强算法[J]. 光学学报, 2007, 27(7): 1184 - 1190.
- [13] 钱惟贤, 陈钱, 顾国华, 等. 一种具有噪声抑制功能的红外图像锐化算法[J]. 光学学报, 2009, 29(7): 1807 - 1811.
- [14] XU W, KIM Y-T. An image sharpening algorithm for high magnification image zooming [C]// Proceedings of the 2010 Digest of Technical Papers International Conference on Consumer Electronics. Piscataway: IEEE, 2010: 27 - 28.
- [15] PATTANASETHANON P, ATTACHOO B. A unified histogram and Laplacian based for image sharpening [C]// ISCIT 2009: Proceedings of the 9th International Symposium on Communications and Information Technology. Piscataway: IEEE, 2009: 1125 - 1130.
- [16] LEE K W, KIM Y S, PARK S H, et al. Effective color distortion and noise reduction for unsharp masking in LCD[J]. IEEE Transactions on Consumer Electronics, 2008, 54(3): 1473 - 1477.
- [17] NOORALIEI A, IRAJI, R. Noise reduction and image sharpening using IJA stochastic learning automaton [C]// Proceedings of the 2010 Second International Conference on Computer Research and Development. Piscataway: IEEE, 2010: 790 - 794.
- [18] TSENG C-C, LEE S-L. Design of fractional delay filter using Hermite interpolation method [J]. IEEE Transactions on Circuits and Systems, 2012, 59(7): 1458 - 1471.
- [19] TSENG C-C, LEE S-L. Digital image sharpening using fractional derivative and mach band effect [C]// Proceedings of the 2012 IEEE International Symposium on Circuits and Systems. Piscataway: IEEE, 2012: 2765 - 2768.
- [20] GUI Z G, LIU Y. An image sharpening algorithm based on fuzzy logic [J]. International Journal for Light and Electron Optics, 2011, 122(8): 697 - 702.
- [21] GUI Z G, ZHANG P C, ZHANG J H, et al. A X-ray image sharpening algorithm using nonlinear module [J]. Imaging Science Journal, 2012, 60(1): 3 - 8.
- [22] 丁绪星, 朱日宏, 李建欣. 一种静止图像质量评价指标[J]. 南京理工大学学报, 2004, 28(5): 507 - 510.
- [23] 张聪, 桂志国. 噪声控制的图像非线性锐化方法[J]. 计算机工程与应用, 2011, 47(4): 154 - 156, 164.
- [24] 杨嘉琛, 侯春萍, 沈丽丽, 等. 基于 PSNR 立体图像质量客观评价方法[J]. 天津大学学报, 2008, 41(12): 1448 - 1452.
- [25] 黄文辉, 陈仁雷, 张家谋. 数字视频图像质量客观测量方法的改进与实现[J]. 北京邮电大学学报, 2005, 28(4): 87 - 90.