doi:10.6041/j.issn.1000-1298.2014.09.012

# 压缩感知苹果图像并行快速重构方法研究*

代　媛[1,2]　何东健[1]　杨　龙[2]

(1. 西北农林科技大学机械与电子工程学院, 陕西杨凌 712100; 2. 西北农林科技大学信息工程学院, 陕西杨凌 712100)

**摘要**: 针对压缩感知重构算法耗时长而影响苹果图像快速获取这一问题,分析了二维正交匹配跟踪重构算法的并行性,借助 GPU 通用并行计算平台,利用 CUDA 技术,设计其对应的并行化重构算法,从而得到快速的苹果图像重构方法。实验结果表明,并行化的算法可将苹果图像重构效率提高 16~35 倍,能在数秒内恢复原始图像,为压缩感知应用到果园远程实时监控及基于图像的苹果质量快速检测与分类等场合提供条件。

**关键词**: 苹果　图像重构　压缩感知　二维正交匹配跟踪算法　并行计算

**中图分类号**: TP399　　**文献标识码**: A　　**文章编号**: 1000-1298(2014)09-0072-07

## 引言

图像采集是机器视觉技术发展的关键技术之一[1],因此苹果图像采集是苹果果园远程监控系统对目标进行识别及苹果品质检测与分级等的重要基础[2]。目前苹果图像仍然依照香农采样定理的传统方法采集,图像数据量大,难以存储和传输问题突出[3]。2006 年,Donoho 等提出一种新的压缩感知(Compressed Sensing, CS)理论,为数据采集技术带来了革命性的突破[4],该理论表明利用较少的信号值可实现稀疏的或可压缩信号的精确重建。近年来,压缩感知在诸多领域受到了关注与重视[5-6],也逐渐应用到农业领域[7-8]。但是,这些研究都未在加速方面展开工作,随着样本尺寸增大及数目的增多,重构信号时间急剧增加,使其难以应用到需快速处理的场合。

近年来,计算机图形处理器(Graphics processing unit, GPU)以其性能、编程、价格等方面的优越性而受到青睐[9],且 GPU 越来越广泛地应用于图形学之外的其它通用计算领域,即基于 GPU 的通用计算(General purpose GPU, GPGPU)[10-11],GPGPU 的发展也为压缩感知重构算法执行效率的提高提供了新的技术途径[12]。因此,针对压缩感知苹果图像重构时间长的问题,本文研究压缩感知中重构算法这一关键技术,借助 GPU 通用计算平台,设计并行化的二维正交匹配跟踪算法(Two dimensional orthogonal matching pursuit, 2D-OMP),提高其执行效率以使其满足快速处理的需求,为压缩感知应用于苹果图像的采集提供更好的技术支持,同时也为解决传统采集方法因数据量大而带来的存储、传输及基于图像的后续各类应用效率低下问题奠定基础。

## 1　压缩感知理论

压缩感知理论主要包括信号的稀疏表示、编码测量和重构算法 3 个方面。其主要思想是首先对可稀疏表示或可压缩的原始信号进行稀疏变换,再利用随机传感矩阵把稀疏信号从高维度上投影到一个相对较低维度的随机弥散空间上,即以远低于奈奎斯特采样定理所要求的采样速率对该稀疏信号进行线性测量编码,最后利用信号重构算法在概率意义上实现信号的精确重构或者在误差允许范围内的近似重构。压缩感知理论下的信号压缩采样过程如图 1 所示。假设 $\boldsymbol{x}\in\mathbf{R}^n$ 是一个在 $\boldsymbol{\Psi}\in\mathbf{R}^{n\times n}$ 域内 $k$ 稀疏的一维信号,其中 $\boldsymbol{x}=\boldsymbol{\Psi z}$,并且在 $\boldsymbol{z}$ 中有 $k\ll n$ 个非零值。通过测量矩阵 $\boldsymbol{\Phi}\in\mathbf{R}^{m\times n}(k<m<n)$ 对 $\boldsymbol{x}$ 进行采样,可得到 $\boldsymbol{y}=\boldsymbol{\Phi x}=\boldsymbol{Az}\in\mathbf{R}^m$,其中 $\boldsymbol{A}=\boldsymbol{\Phi\Psi}$ 为传感矩阵,最后利用测量值,采用压缩感知重构算法就可高精度的恢复原始信号。

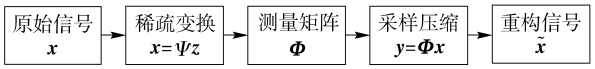


图 1　压缩感知及信号重构过程框图

Fig. 1　Process of compressed sensing and signal reconstruction

收稿日期: 2013-10-08　修回日期: 2013-11-17

*国家自然科学基金资助项目(61271280、61001100)和"十二五"国家科技支撑计划资助项目(2012BAH29B04-00)

**作者简介**: 代媛,博士生,讲师,主要从事并行计算、压缩编码方向研究,E-mail: dy@nwsuaf.edu.cn

**通讯作者**: 何东健,教授,博士生导师,主要从事图像处理、智能监测及虚拟现实方向研究,E-mail: hdj168@nwsuaf.edu.cn

## 2　重构算法

目前常用的压缩感知重构方法主要是针对 $L_0$ 范数最小提出的一系列贪婪算法[13]，其具有计算量小、重建效果好的优点，因此应用广泛。其中正交匹配跟踪算法（Orthogonal matching pursuit，OMP）是一个典型的贪婪算法，该算法首先寻找字典中与残差最匹配的原子，将其添加到已选择的原子序列中去，并对已选择的原子进行 Schmidt 正交变换，之后将信号投影到正交原子构成的空间上，得到信号在已选原子的分量和迭代残差，反复迭代。该方法实现简单、效率较高，但是由于 OMP 算法是为一维信号设计的，而二维信号的恢复需通过二维的离散采样将其信号转换为一维信号，导致其存储空间较大并且解码端复杂度高。而 Fang 等在 2012 年提出一个二维正交匹配跟踪算法[14]，可直接对二维信号操作，其不同之处是在该算法中每个原子是一个矩阵，并且解码端将采样矩阵投影到二维原子空间去选择最匹配原子。2D-OMP 算法不仅在效率上优于 1D-OMP 算法还节省内存空间，因此，本文采用 2D-OMP 算法重构压缩感知的苹果图像。

令 $\boldsymbol{X}\in\mathbf{R}^{n\times n}$ 是一个在 $\boldsymbol{\Psi}$ 域内 $k$ 稀疏的二维信号，即 $\boldsymbol{X}=\boldsymbol{YZY}^{\mathrm{T}}$，并且在 $\boldsymbol{Z}$ 中有 $k\ll n^2$ 个非零值。通过测量矩阵 $\boldsymbol{\Phi}$ 对 $\boldsymbol{X}$ 进行采样可得到 $\boldsymbol{Y}=\boldsymbol{\Phi X\Phi}^{\mathrm{T}}=\boldsymbol{AZA}^{\mathrm{T}}\in\mathbf{R}^{m\times m}$，其中 $\boldsymbol{A}=\boldsymbol{\Phi\Psi}$。令 $\boldsymbol{A}=(\boldsymbol{a}_1,\cdots,\boldsymbol{a}_n)$，$\boldsymbol{a}_i$ 代表 $\boldsymbol{A}$ 的第 $i$ 列。在 2D-OMP 中，字典包含 $n^2$ 个原子，并且每个原子是一个 $m\times m$ 的矩阵。2D-OM 算法需要 2 个辅助变量，首先，利用 $(\boldsymbol{i},\boldsymbol{j})$ 标注将要选择原子的坐标，$\boldsymbol{i}$ 表示行，$\boldsymbol{j}$ 代表列；其次，$\boldsymbol{R}\in\mathbf{R}^{m\times m}$ 用来存储从 $\boldsymbol{Y}$ 拿走匹配原子后的剩余残差。算法开始将 $\boldsymbol{Y}$ 的值赋给 $\boldsymbol{R}$，此后在每次迭代中，解码端在字典中寻找和残差最为匹配的原子，并通过最小二乘更新已选择的原子集[14]。算法具体为：

Inputs：

$\boldsymbol{A}\in\mathbf{R}^{m\times n}$：采样矩阵

$\boldsymbol{Y}\in\mathbf{R}^{m\times m}$：样本

$k$：稀疏度

Output：

$\hat{\boldsymbol{Z}}\in\mathbf{R}^{n\times n}$：重构信号 $\boldsymbol{Z}$

Variables：

$\boldsymbol{R}\in\mathbf{R}^{m\times m}$：残差

$(\boldsymbol{i},\boldsymbol{j})$：标注将要选择原子的坐标，$\boldsymbol{i}$ 表示行，$\boldsymbol{j}$ 代表列。

Initialization：

$\boldsymbol{R}\leftarrow\boldsymbol{Y}$

$(\boldsymbol{i},\boldsymbol{j})\leftarrow\{(1,1),(1,2),\cdots,(n,n)\}$

For $t\leftarrow 1$ to $k$ do

$$(i_t,j_t)\leftarrow\arg\max_{(i',j')\in(i,j)}\frac{|\langle\boldsymbol{R},\boldsymbol{B}_{i',j'}\rangle|}{\|\boldsymbol{B}_{i',j'}\|_2}$$

$$(\boldsymbol{i},\boldsymbol{j})\leftarrow(\boldsymbol{i},\boldsymbol{j})\setminus(i_t,j_t)$$

$$\hat{\boldsymbol{u}}\leftarrow\arg\min_{u}\left\|\boldsymbol{Y}-\sum_{t'=1}^{t}u_{t'}\boldsymbol{B}_{i_{t'},j_{t'}}\right\|_2$$

$$\boldsymbol{R}\leftarrow\boldsymbol{Y}-\sum_{t'=1}^{t}\hat{u}_{t'}\boldsymbol{B}_{i_{t'},j_{t'}}$$

For $t\leftarrow 1$ to $k$ do

$$\tilde{z}_{i_t,j_t}\leftarrow\hat{u}_t$$

## 3　基于 GPU 并行计算的苹果图像重构

在压缩感知图像的重构过程中，其算法复杂度高，运算效率低。而 2D-OMP 算法包含大量的矩阵运算，可进行并行计算。GPU 具有良好的并行处理能力，且其在性能、编程及价格方面有着优越性，故采用 GPU 平台开发其相应的并行算法，以提高 2D-OMP 算法重构苹果图像的效率。

### 3.1　GPU 架构及 CUDA 技术

传统的 GPU 通用计算开发方式是将数据打包成纹理，将并行计算任务映射为图形学中的图像着色，使用图形学 API 进行开发，利用 GPU 中的可编程顶点着色单元和像素着色单元作为处理器完成通用计算任务[15]。其要求开发人员不仅熟悉需要移植具体应用的并行算法，还要精通计算机图形学和 GPU 架构才能完成 GPGPU 的开发，这样编程的灵活性就大打折扣。为了降低编程的难度和提高灵活性，NVIDIA 在 2007 年推出了一种将 GPU 作为数据并行计算设备的软硬件体系 CUDA（Computer unified device architecture，统一计算设备结构），它采用统一处理架构，使得 GPU 更加适合进行通用计算。它以扩展的 C 语言为基础，取代了以前基于 GPU 的程序设计模式。CUDA 构架分为 2 部分，主机 Host 和从设备 Device，通常 CPU 被看作是主机，而 GPU 则认为是从设备[16]。主程序是由 CPU 来执行，当遇到数据并行处理的部分，CUDA 就会将程序编译成 GPU 能执行的程序，并传送到 GPU。而从设备代码是使用 C 语言扩展并附有关键字写成的数据并行函数，称为核（kernel）函数，核函数通过一系列并行的线程以单指令多线程的模型来并行执行[11]。

### 3.2　重构算法的并行性分析

2D-OMP 算法主要分为 4 个步骤，下面对其各步骤进行分析，确定各步骤的复杂度及可并行部分。

（1）投影：在 2D-OMP 算法中，字典包含 $n^2$ 个

原子,而每个原子是一个 $m \times m$ 的矩阵,设 $\boldsymbol{B}_{i,j}$ 为第 $(i,j)$ 个原子并定义为:$\boldsymbol{B}_{i,j} = \boldsymbol{a}_i \boldsymbol{a}_j^{\mathrm{T}}$,$\boldsymbol{Y}$ 可用 $\boldsymbol{B}_{i,j}$ 的加权和表示,那么 $\boldsymbol{Y}$ 在 $\boldsymbol{B}_{i,j}$ 的投影为 $\langle \boldsymbol{Y}, \boldsymbol{B}_{i,j} \rangle / \| \boldsymbol{B}_{i,j} \|_2$,其中 $\langle \boldsymbol{Y}, \boldsymbol{B}_{i,j} \rangle \triangleq \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{Y} \boldsymbol{a}_j$,$\| \boldsymbol{B}_{i,j} \|_2$ 代表 $\boldsymbol{B}_{i,j}$ 的范数。设 $\boldsymbol{P}$ 是一个 $n \times n$ 的矩阵,并且 $\| \boldsymbol{B}_{i,j} \|_2$ 是其第 $(i,j)$ 个元素,那么投影这一步骤即可以这样方式实现:$\boldsymbol{A}^{\mathrm{T}} \boldsymbol{R} \boldsymbol{A} / \boldsymbol{P}$,因此当 $m < n$ 时,这一步骤的复杂度为 $O(mn^2)$,且矩阵相乘是这部分并行的核心。

(2) 选择最匹配原子:在未选择的原子中寻找绝对值最大的投影所对应原子。由于一共有 $n^2$ 个原子,而 $n^2 \gg k$,因此,该部分的复杂度为 $O(n^2)$,且矩阵元素求最大值可并行完成。

(3) 更新权重:用 $t$ 个已选择原子的加权和逼近 $\boldsymbol{Y}$,设

$$\boldsymbol{R} = \boldsymbol{Y} - \sum_{t'=1}^{t} u_{i_{t'},j_{t'}} \boldsymbol{B}_{i_{t'},j_{t'}} \tag{1}$$

则寻找最优的 $\boldsymbol{u} = (u_{i_1,j_1}, \cdots, u_{i_t,j_t})^{\mathrm{T}}$ 以求 $\boldsymbol{R}$ 的最小范数,等价于一个最小二乘问题[14]

$$\hat{\boldsymbol{u}} = \boldsymbol{H}^{-1} \boldsymbol{f} \tag{2}$$

令 $\langle \boldsymbol{B}_{i_{t'},j_{t'}}, \boldsymbol{B}_{i_{s'},j_{s'}} \rangle \triangleq \langle \boldsymbol{a}_{i_{t'}}, \boldsymbol{a}_{i_{s'}} \rangle \langle \boldsymbol{a}_{j_{t'}}, \boldsymbol{a}_{j_{s'}} \rangle$

则

$$\boldsymbol{H} = \begin{pmatrix} \langle \boldsymbol{B}_{i_1,j_1}, \boldsymbol{B}_{i_1,j_1} \rangle & \cdots & \langle \boldsymbol{B}_{i_1,j_1}, \boldsymbol{B}_{i_t,j_t} \rangle \\ \vdots & \vdots & \vdots \\ \langle \boldsymbol{B}_{i_t,j_t}, \boldsymbol{B}_{i_1,j_1} \rangle & \cdots & \langle \boldsymbol{B}_{i_t,j_t}, \boldsymbol{B}_{i_t,j_t} \rangle \end{pmatrix} \tag{3}$$

并有

$$\boldsymbol{f} = (\langle \boldsymbol{Y}, \boldsymbol{B}_{i_1,j_1} \rangle, \cdots, \langle \boldsymbol{Y}, \boldsymbol{B}_{i_t,j_t} \rangle)^{\mathrm{T}} \tag{4}$$

计算 $\boldsymbol{H}$ 和 $\boldsymbol{f}$ 的复杂度分别是 $O(mt^2)$ 和 $O(tm^2)$,计算 $\boldsymbol{H}^{-1}$ 的复杂度是 $O(t^3)$,且计算 $\boldsymbol{H}^{-1}\boldsymbol{f}$ 的复杂度为 $O(t^2)$,该部分矩阵运算均可并行执行。

(4) 更新残差:该部分主要是矩阵向量相乘,其复杂度依赖于 $t$,而 $t \leqslant k \ll n^2$,因此,它的计算时间只占总时间很少一部分。

根据以上分析,当 $k$ 较小时,2D-OMP 算法的复杂度主要由投影部分决定,其复杂度为 $O(mn^2)$。当 $k$ 增大时,更新权重中的复杂度迅速增加。因此,加速更新权重和投影是提高整个算法的关键。

### 3.3 矩阵运算的并行计算

由上述分析可知,更新权重和投影是 2D-OMP 算法中复杂度高的部分,而这两部分主要包含矩阵矩阵相乘和矩阵向量相乘运算,此外,更新残差部分也包含矩阵向量相乘运算,而选择最匹配原子则是求最大值问题。因此,主要任务就是在 GPU 上加速矩阵矩阵相乘和矩阵向量相乘的运算,以提高整个算法的计算效率。

#### 3.3.1 矩阵矩阵相乘并行设计

假设计算 2 个矩阵 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的乘积,$\boldsymbol{A}$ 的大小为 $M \times P$,$\boldsymbol{B}$ 的大小为 $P \times N$。在相乘的过程中,$\boldsymbol{A}$ 的每行和 $\boldsymbol{B}$ 的每列中的元素都要反复使用,因此,在此过程中需对 GPU 的全局存储器反复访问。由于全局存储器的访问速度非常慢,导致计算效率很低。所以,将 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 的数据移植至能高速访问的共享存储器,这将会在很大程度上提高访问速度。然而,共享存储器的存储空间有限,无法同时存储 2 个较大矩阵。因此,引入一个分块算法[17],以使 2 个矩阵分批移植至共享存储器并分块计算。

图 2 展示了采用分块法实现矩阵矩阵相乘的方

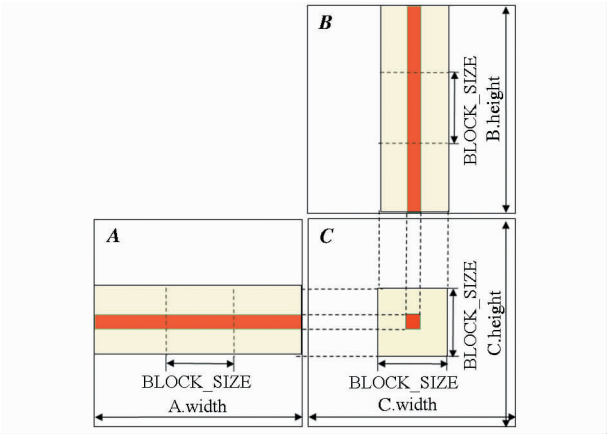


图 2 采用分块法的矩阵-矩阵相乘示意图

Fig. 2 Matrix-matrix multiplication using blocking algorithm

案。将 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 分成互不重叠的子矩阵块,其大小为 BLOCK_SIZE × BLOCK_SIZE,矩阵 $\boldsymbol{C}$ 存储 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 乘积的结果,$\boldsymbol{C}$ 中的每个元素都是 $\boldsymbol{A}$ 的行和 $\boldsymbol{B}$ 的列相乘并相加的结果,其红色小块代表其中一个元素,是 $\boldsymbol{A}$ 中红色行和 $\boldsymbol{B}$ 中红色列运算的结果。为子矩阵块定义大小为 BLOCK_SIZE × BLOCK_SIZE 的共享存储器,并指派 GPU 每个线程计算 $\boldsymbol{C}$ 中每个元素值。首先将 $\boldsymbol{A}$ 的最顶行和 $\boldsymbol{B}$ 的最左列的子矩阵块加载至共享存储器,则可计算 $\boldsymbol{C}$ 中第 1 个子矩阵块的结果。当每个线程计算完之后,再加载第 2 批 $\boldsymbol{A}$ 和 $\boldsymbol{B}$ 子矩阵块去计算 $\boldsymbol{C}$ 的第 2 个子矩阵,按照这样的方式分批加载并分块计算,即可在共享存储器中完成矩阵的乘积运算,且在计算的过程中不需要访问全局存储器,这样大大提高了并行计算的效率,其整个并行代码为:

```
_shared_float matA[BLOCK_SIZE][BLOCK_SIZE];
_shared_float matB[BLOCK_SIZE][BLOCK_SIZE];
int tidc = threadIdx.x;
int tidr = threadIdx.y;
int bidc = blockIdx.x * BLOCK_SIZE;
int bidr = blockIdx.y * BLOCK_SIZE;
float results = 0;
float comp = 0;
for (int j = 0; j < P; j + = BLOCK_SIZE)
```

```
{
  matA[tidr][tidc] = A[(tidr + bidr) * P + tidc +
j];
  matB[tidr][tidc] = B[(tidr + j) * N + tidc + bidc];
    _syncthreads();
  for(int i = 0; i < BLOCK_SIZE; i + +)
  {
    float t;
    comp - = matA[tidr][i] * matB[i][tidc];
    t = results - comp;
    comp = (t - results) + comp;
    results = t;
  }
  _syncthreads();
}
C[(tidr + bidr) * N + tidc + bidc] = results;
```

**3.3.2**　矩阵向量相乘并行设计

2D-OMP 算法中包含大量的矩阵向量相乘运算，因此设计一个快速并行的矩阵向量相乘方案非常重要。假设计算矩阵 $\boldsymbol{E}$ 和向量 $\boldsymbol{f}$ 的乘积，如果指派 GPU 每个线程计算 $\boldsymbol{E}$ 中每行和 $\boldsymbol{f}$ 的乘积运算，则访问 $\boldsymbol{E}$ 的这种模式会导致非合并访问。合并访问是影响 CUDA 程序效率的一个关键的因素，一个合并访问意味着硬件可以合并很多线程的访问请求而一次性进行，这样就可减少通信次数以提高访问效率。因此，可以对矩阵 $\boldsymbol{E}$ 做一些变换，使得其满足合并访问：采用 $\boldsymbol{E}$ 的转置矩阵去乘以 $\boldsymbol{f}$，这样，每个线程则计算 $\boldsymbol{E}$ 的每列与 $\boldsymbol{f}$ 的乘积，那么在第一次计算中，线程 0 就访问 $\boldsymbol{E}[0]$，线程 1 就访问 $\boldsymbol{E}[1]$，线程 $j$ 就访问 $\boldsymbol{E}[j]$，后续的计算都是按照这样的模式。因此，每个线程束中的线程都访问相邻的地址，访问请求就可合并一次进行，有效地提高访问速度。

此外，程序还可以进一步优化，考虑到共享存储器的访问速度高，可尝试将向量 $\boldsymbol{f}$ 加载至共享存储器以提高访问速度。由于共享存储器的存储空间有限，无法加载大向量，因此将向量分成多块，每次向共享存储器加载一块，当该块数据计算完成后，加载下一块，以此类推。因此，当利用数据合并访问及共享存储器后，可获得一个快速并行的矩阵向量相乘运算方案。

### 3.4　苹果图像的并行快速重构

苹果图像压缩感知及并行重构过程如图 3 所示，最初输入原始苹果图像对其进行稀疏变换，变换矩阵采用多贝西小波变换矩阵，再对信息进行观测编码，观测矩阵采用随机高斯测量矩阵，从而获得观测数据。此后，使用 2D-OMP 算法在 GPU 平台上从观测数据中对苹果图像进行并行快速重构：①在 GPU 上给变量分配内存空间，输入参数包括 $m \times m$ 的采样矩阵 $\boldsymbol{Y}$，$m \times n$ 的测量矩阵 $\boldsymbol{A}$，输出 $\boldsymbol{Z}$ 是一个 $n \times n$ 的矩阵。在算法的核函数执行之前，$\boldsymbol{Y}$ 需从 CPU 中拷贝至 GPU，$\boldsymbol{A}$ 是只读参数，因此可将其放入 GPU 的纹理存储器以提高读的速度。2D-OMP 算法执行 $k$ 次迭代，在每次迭代中，首先计算投影 $\boldsymbol{A}^{\mathrm{T}}\boldsymbol{RA}/\boldsymbol{P}$，使用 2 个核函数分别计算两次矩阵相乘同时完成点除运算。②调用 CUBLAS 库函数求矩阵最大值以获得最匹配原子。③调用 4 个核函数求解矩阵逆。④更新残差中包括矩阵向量相乘运算，调用一个核函数可完成其并行计算。待算法计算完后，恢复矩阵 $\boldsymbol{Z}$ 再从 GPU 中拷贝至 CPU。

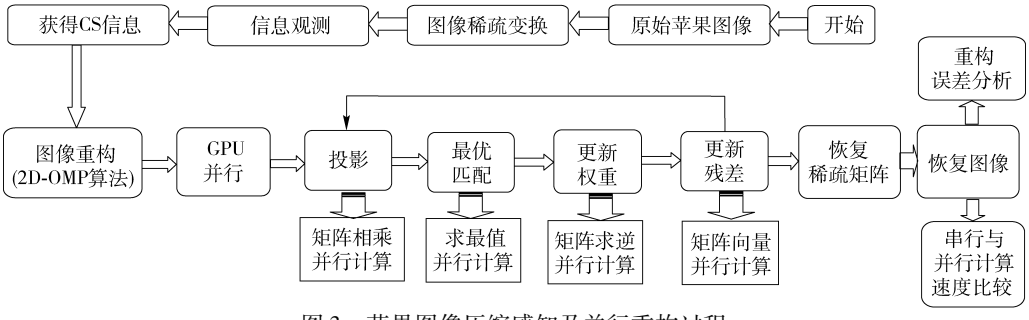


图 3　苹果图像压缩感知及并行重构过程

Fig. 3　Compressed sensing and parallel reconstruction of the apple image

## 4　实验结果

实验测试在英特尔 3.07 GHz 的 i7 核 CPU 上进行，CPU 串行 C 代码使用 $O_2$ 级优化。GPU 为 NVIDIA 公司具有 CUDA、性能为 2.0 的 GTX480，有 15 个多重流处理器（SMs），每个多重流处理器包括 32 个流处理器（SPs），一共有 480 个处理核。每个 SM 有 32 KB 的寄存器和 48 KB 的共享存储器，可提供高速的数据访问。此外，还有 64 KB 的常数存储器和纹理存储器。

采用图 4 所示的 6 幅不同 256 × 256 BMP 格式的苹果灰度图像进行模拟测试（左边为原始图，右

边为重构图),变换矩阵采用多贝西小波变换矩阵,测量矩阵均采用 128 ×256 的随机高斯测量矩阵,即采样率 $m/n = 0.5$ 的情况下进行采样(采样的图像大小为 128 ×128,其采样的数据量为原图的 1/4),采用 2D-OMP 重构算法分别在 CPU 和 GPU 上进行重构测试,其执行时间如表 1 所示。

**表 1 苹果图像重构在 CPU 和 GPU 上执行效率对比**

**Tab. 1 The performance of the reconstruction for apple images on CPU and GPU**

| 图像 | 迭代次数 | CPU 执行时间/ms | GPU 执行时间/ms | 加速比 | PSNR |
|---|---|---|---|---|---|
| 实例 1 | 1 494 | 84 954 | 4 919 | 17. 2 | 31. 32 |
| 实例 2 | 1 555 | 88 101 | 5 262 | 16. 7 | 30. 77 |
| 实例 3 | 1 750 | 105 924 | 6 660 | 15. 9 | 29. 29 |
| 实例 4 | 628 | 26 847 | 920 | 29. 1 | 36. 81 |
| 实例 5 | 1 680 | 99 370 | 6 063 | 16. 3 | 29. 08 |
| 实例 6 | 512 | 22 175 | 634 | 34. 9 | 38. 63 |

由表 1 可以看出,不同图像达到所给定的误差所需迭代的次数不同(当 $R$ 的均方误差小于 0.03 时停止迭代),在室外自然条件下的苹果图像(实例 1、实例 2、实例 3、实例 5)背景较复杂,达到给定误差需要迭代的次数多,并且重构图像的峰值信噪比(Peak signal-noise ration, PSNR)较低,而在室内环境下的苹果图像(实例 4、实例 6),背景简单,需要迭代的次数少,且重构图像的峰值信噪比较高。另外,从表 1 可看出,并行 2D-OMP 算法对于图 4 不同的苹果图像在 GPU 上可获得 16 ~35 倍的加速。由于加速比与图像的内容无关,仅与图像的尺寸和迭代次数有关,因此,针对同一幅图像(实例 1)在不同的迭代次数进行测试,其结果如图 5 所示,由此可看出同一幅图像在不同的迭代次数下并行 2D-OMP 算法可获得 17 ~42 倍的加速,且当算法迭代次数较多时,在 GPU 上的加速比较小,而迭代次数较少时,加速比较大。最终基于 GPU 并行计算的苹果图像的重构均可在数秒内完成。

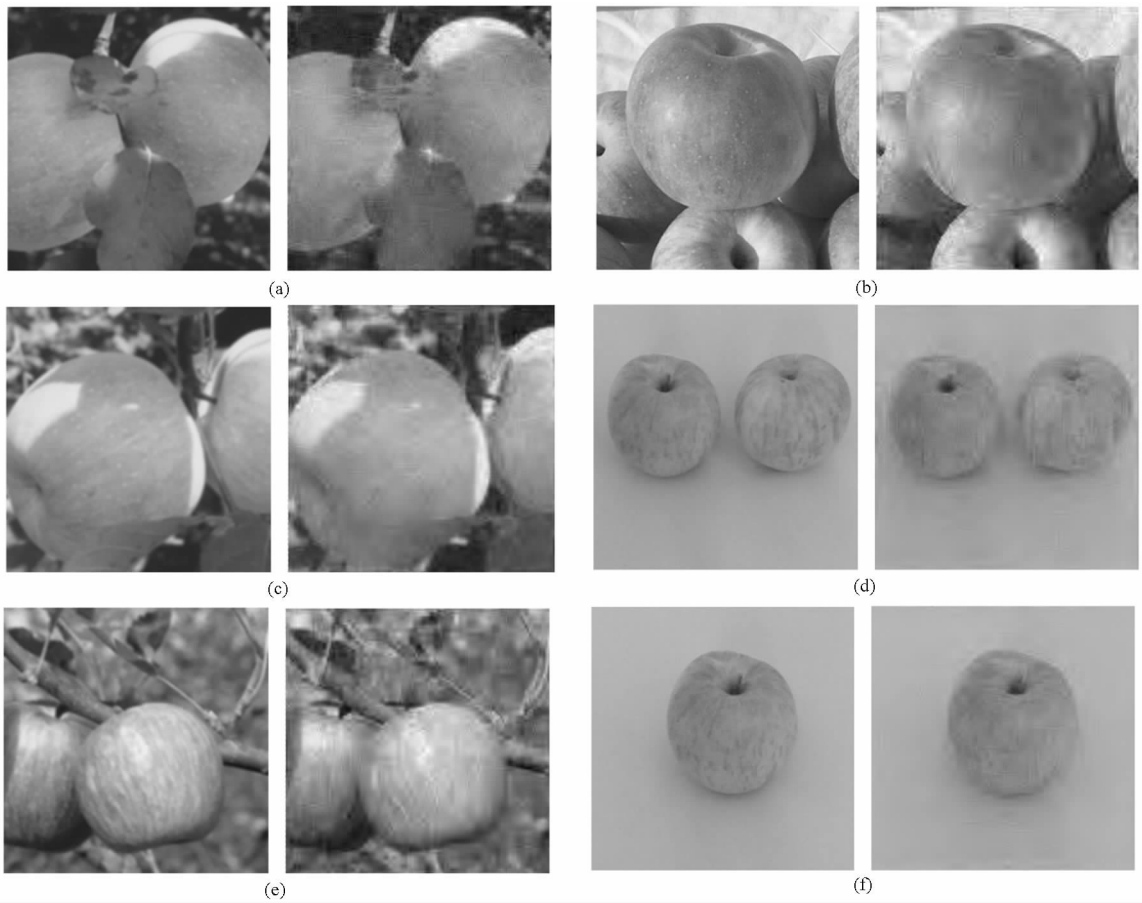


图 4 采样率 $m/n = 0.5$ 时,苹果图像重构效果

Fig. 4 When $m/n = 0.5$, the reconstruction result of apple images

(a) 实例 1 (b) 实例 2 (c) 实例 3 (d) 实例 4 (e) 实例 5 (f) 实例 6

## 5 结论

(1)提出一个并行快速的苹果重构方法。在 GPU 通用并行计算平台上利用 CUDA 技术实现 2D-OMP 算法中大量的矩阵并行运算,从而得到并行快速的苹果图像重构方法。实验结果表明,在 GTX480 平台上,并行 2D-OMP 算法重构苹果图像比原始算法快 16 ~35 倍,能在数秒内恢复 256 ×256

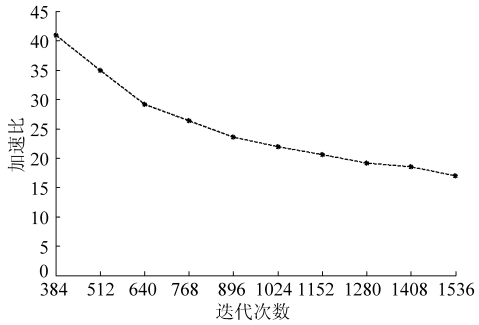


图 5　实例 1 在不同迭代次数下在 GPU 上的加速比

Fig. 5　Speedup of example 1 on GPU with various iterations

的苹果灰度图像，并且需要的数据量为原来图像的 1/4。

(2)并行重构算法大大提高图像恢复的速度，解决目前苹果图像使用压缩感知技术面临复杂度高而耗时长的问题，为苹果生长过程远程监控系统实时采集图像及苹果质量快速检测与分类等应用提供条件。

## 参 考 文 献


1　Alexander Hornberg. Handbook of Machine Vision [M]. Weinheim: Wiley-VCH, 2006.

2　司永胜，乔军，刘刚，等. 基于机器视觉的苹果识别和形状特征提取[J]. 农业机械学报，2009，40(8)：161 - 165.
Si Yongsheng, Qiao Jun, Liu Gang, et al. Recognition and shape features extraction of apples based on machine vision [J]. Transactions of the Chinese Society for Agricultural Machinery, 2009, 40(8): 161 - 165. (in Chinese)

3　戴琼海，付长军，季向阳. 压缩感知研究[J]. 计算机学报，2011，34 (3)：425 - 434.
Dai Qionghai, Fu Changjun, Ji Xiangyang. Research on compressed sensing [J]. Chinese Journal of Computer, 2011, 34 (3): 425 - 434. (in Chinese)

4　Donoho D L. Compressed sensing [J]. IEEE Transactions on Information Theory, 2006, 52 (4): 1289 - 1306.

5　邵文泽，韦志辉. 压缩感知基本理论：回顾与展望[J]. 中国图象图形学报，2012，17 (1)：1 - 12.
Shao Wenze, Wei Zhihui. Advances and perspectives on compressed sensing theory [J]. Journal of Image and Graphics, 2012, 17 (1): 1 - 12. (in Chinese)

6　霍迎秋，唐晶磊，尹秀珍，等. 基于压缩感知理论的苹果病害识别方法[J]. 农业机械学报，2013，44(10)：227 - 232.
Huo Yingqiu, Tang Jinglei, Yin Xiuzhen, et al. Apple disease recognition based on compressive sensing[J]. Transactions of the Chinese Society for Agricultural Machinery, 2013, 44(10): 227 - 232. (in Chinese)

7　蔡骋，张明，朱俊平. 基于压缩感知理论的杂草种子分类识别[J]. 中国科学：信息科学，2010，40(增刊)：160 - 172.
Cai Cheng, Zhang Ming, Zhu Junping. Weed seeds classification based on compressive sensing theory [J]. Science China: Information Science, 2010, 40(Supp.): 160 - 172. (in Chinese)

8　杨小青. 基于压缩传感的水果分级系统的研究[D]. 西安：陕西科技大学，2012.
Yang Xiaoqing. Study of fruit classification system based on compressed sensing [D]. Xi'an: Shaanxi University of Science and Technology, 2012. (in Chinese)

9　Ho Tze-Yui, Lam Ping-Man, Leung Chi-Sing. Parallelization of cellular neural networks on GPU [J]. Pattern Recognition, 2008, 41(8): 2684 - 2692.

10　吴恩华. 图形处理器用于通用计算的技术、现状及其挑战[J]. 软件学报，2004，15(10)：1493 - 1504.
Wu Enhua. State of the art and future challenge on general purpose computation by graphics processing unit [J]. Journal of Software, 2004, 15(10): 1493 - 1504. (in Chinese)

11　Sanders J, Kandrot E. CUDA by example: an introduction to general-purpose GPU programming[M]. 1st ed. Addison-Wesley Professional, 2010.

12　吴恩华，柳有权. 基于图形处理器(GPU)的通用计算[J]. 计算机辅助设计与图形学学报，2004，16(5)：601 - 612.
Wu Enhua, Liu Youquan. General purpose computation on GPU [J]. Journal of Computer Aided Design & Computer Graphics, 2004, 16(5): 601 - 612. (in Chinese)

13　李树涛，魏丹. 压缩传感综述[J]. 自动化学报，2009，35(11)：1369 - 1377.
Li Shutao, Wei Dan. A survey on compressive sensing [J]. Acta Automatica Sinica, 2009, 35(11): 1369 - 1377. (in Chinese)

14　Fang Yong, Wu Jiaji, Huang Bormin. 2D sparse signal recovery via 2D orthogonal matching pursuit [J]. Science China: Information Science, 2012, 55(4): 889 - 897.

15　Kirk David B, Hwu Wen-mei W. Programming massively parallel processors: a hands-on approach[M]. Amsterdam: Elsevier, 2010.

16　Boyer V, El Baz D, Elkihel M. Solving knapsack problems on GPU [J]. Computers & Operations Research, 2012, 39(1): 42 - 47.

17　NVIDIA. NVIDIA CUDA C programming guide [M]. Santa Clara, CA: NVIDIA Corporation, 2010.

18　冯鑫，王晓明，党建武，等. 基于改进非下采样轮廓波的图像融合算法[J]. 农业机械学报，2012，43(12)：192 - 196.
Feng Xin, Wang Xiaoming, Dang Jianwu, et al. Image fusion algorithm based on improved nonsubsampled contourlet [J]. Transactions of the Chinese Society for Agricultural Machinery, 2012, 43(12): 192 - 196. (in Chinese)

# Parallel and Fast Reconstruction Algorithm for Compressed Sensing Apple Image

Dai Yuan[1,2] He Dongjian[1] Yang Long[2]

(1. *College of Mechanical and Electronic Engineering*, *Northwest A&F University*, *Yangling*, *Shaanxi* 712100, *China*
2. *College of Information Engineering*, *Northwest A&F University*, *Yangling*, *Shaanxi* 712100, *China*)

**Abstract**: With the emerging of compressed sensing (CS), it is possible to overcome the storage and transmission difficulty of the mass data sampled by traditional methods. It also provides a new way for machine vision applied to apple image sampling. However, the major shortcoming of the reconstruction algorithms for CS signals is the expensive computing time, which limits its applications to the occasions requiring fast processing. Aiming at this problem, two dimensional orthogonal matching pursuit algorithm with parallel computing is proposed for apple image reconstruction. The parallelism of the algorithm is analyzed and the parallel algorithm using CUDA technology on GPU is designed in order to achieve a fast reconstruction algorithm. Experimental results show that the parallel algorithm improves the recovery efficiency by 16 to 35 times and the apple image can be recovered in several seconds. This method provides a new technical support to remote monitoring in real time for apple garden. It can be used in the fast apple quality detection based on image as well.

**Key words**: Apple Image reconstruction Compressed sensing Two dimensional orthogonal matching pursuit Parallel computing

(上接第 65 页)

# Positioning Method for Tea Picking Using Active Computer Vision

Zhang Hao Chen Yong Wang Wei Zhang Guolu

(*College of Mechanical and Electronic Engineering*, *Nanjing Forestry University*, *Nanjing* 210037, *China*)

**Abstract**: For the intelligent tea picking machine, the positioning of tea tips was a difficult procedure. A positioning method based on active computer vision and corresponding visual system were proposed to solve this problem. First, according to the characteristics of the tea picking surface, a cross light path of projection and camera was designed; then the recognition approach of tea tips based on the color factor was developed in natural environment; furthermore, fringe projection profilometry was studied to acquire the height information of tea tips. In fringe projection profilometry, the temporal phase unwrapping method was used to obtain the phase maps, the invalidity identification framework and morphological filter were designed to remove noise points, and a polynomial approximation method which can reduce the nonlinear error was applied to calculate the height. The experimental results show that the positioning system for tea tips based on fringe projection technology can effectively recognize the tips and extract their three-dimensional information, which can help new tea picking machine to realize the positioning function.

**Key words**: Tea Positioning method Active computer vision Fringe projection profilometry