

# 实时图像纹理替换算法

万金梁<sup>1</sup> 宋金宝<sup>2</sup> 叶 龙<sup>2</sup> 李淑红<sup>1</sup>

(河南财经政法大学计算机与信息工程学院 郑州 450002)<sup>1</sup>

(中国传媒大学信息工程学院 北京 100024)<sup>2</sup>

**摘要** 提出了一种实时图像纹理替换算法。使用区域纹理样本实现中等水平视觉的区域纹理替换。利用图像纹理替换潜在的数据并发特征,在集群环境下,设计并实现了混合 MPI 与 OpenMP 的图像纹理替换并行算法,给出了主从模式下的 MPI 并行设计、OpenMP 并行设计及负载均衡。实验结果表明,并行算法可行有效,保证重建图像符合人眼视觉要求,大大减少了图像重建所需数据量;有效降低了计算时间,能够满足硬实时系统的时间约束。

**关键词** 图像纹理替换,并行计算,MPI,OpenMP

**中图分类号** TP301.6 **文献标识码** A

## Real-time Image Texture Substitution Algorithm

WAN Jin-liang<sup>1</sup> SONG Jin-bao<sup>2</sup> YE Long<sup>2</sup> LI Shu-hong<sup>1</sup>

(College of Computer and Information Engineering, Henan University of Economics and Law, Zhengzhou 450002, China)<sup>1</sup>

(School of Information Engineering, Communication University of China, Beijing 100024, China)<sup>2</sup>

**Abstract** A real-time image texture substitution algorithm was proposed in this paper. The regional texture sample can be selected and be used to achieve medium level visual region texture substitution. The parallel image texture substitution algorithm with a hybrid MPI and OpenMP approach can be implemented on a dedicated cluster. The MPI parallelism, OpenMP parallelism and dynamic load balance under Master-Slave model were presented. Experimental results show parallel algorithm can ensure good performance of visual quality, greatly reduce the amount of data required by image reconstruction, effectively reduce the computing time and satisfy the time constraints of real-time application.

**Keywords** Image texture substitution, Parallel computing, MPI, OpenMP

## 1 引言

人眼是图像信息的主要接受源,因此可利用人类视觉系统的感知特性去除视觉感知冗余,改善图像编码的编码效果及计算效率。然而由于基于像素的图像与视频表征方法自身的局限性,使得进一步显著提高压缩效率的目标难以实现<sup>[1]</sup>。因此,如何找到一种更加有效的图像表征方法一直是图像编码领域的研究热点与难点问题。

Kunt 首先提出了利用人类视觉感知特性的第二代图像编码的思想,受到人们的广泛关注<sup>[2]</sup>。现实中存在的很多图像都包含大量的纹理如草、云、沙滩等。而这些图像通常构成图像或视频的静止背景。Ndjiki-Nya 等人<sup>[3]</sup>把图像中的纹理分为两大类:不重要主观细节纹理和重要主观细节纹理。由于人类视觉系统固有的缺陷,图像中的纹理通常是人眼不关注的那部分内容,纹理细节的变化不会影响对原始纹理的主观理解。人类视觉系统的这些特性可用于去除视觉感知冗余。

实际的编码算法设计还应考虑计算效率这个重要问

题<sup>[4]</sup>,而这在实时应用中尤为重要。随着中央处理器不断向多核方向发展及其应用越来越普及<sup>[5]</sup>,并行计算逐渐成为提高计算效率的主要途径。近几年来,混合 MPI 与 OpenMP 的并行算法设计框架以其兼顾多进程与多线程而越来越受到人们关注,为进一步提高并行计算效率提供了发展空间<sup>[6-9]</sup>。

本文提出了一种基于集群的混合 MPI 与 OpenMP 的图像纹理替换并行算法,其使用区域的轮廓结构与纹理信息简洁可靠地表征图像。实验结果表明,并行算法不仅能保证重建图像符合人眼视觉要求,大大降低图像重建所需的数据量,而且能有效减少图像纹理替换计算时间,满足硬实时系统的时间约束。

## 2 图像纹理替换

图像纹理替换是用区域轮廓结构特征和区域纹理样本实现对图像的解析与重建。首先对原始图像进行区域分割<sup>[10-12]</sup>,得到  $N$  个分割区域及其索引号列表,在此基础上分别提取各个分割区域的轮廓结构特征与纹理样本,实现各个区域的轮廓重建与区域纹理合成。最后把各个区域合成的区

到稿日期:2012-11-10 返修日期:2013-02-25 本文受国家自然科学基金重点项目(60832004),青年项目(61201236)资助。

万金梁(1980-),男,博士,讲师,CCF 会员,主要研究方向为数字图像处理与高性能计算,E-mail:wanqq\_2006@sohu.com;宋金宝(1980-),男,博士生,讲师,主要研究方向为图像与视频编码;叶 龙(1981-),男,博士,助理研究员,主要研究方向为数字信号处理与模式识别;李淑红(1972-),女,博士,教授,主要研究方向为数字图像处理与人工智能。

域纹理填充到各自的重建轮廓中,实现图像纹理替换。图像纹理替换的串行流程如图1所示。下文主要对区域轮廓的结构特征提取与重建、区域纹理的样本选择与合成进行介绍。

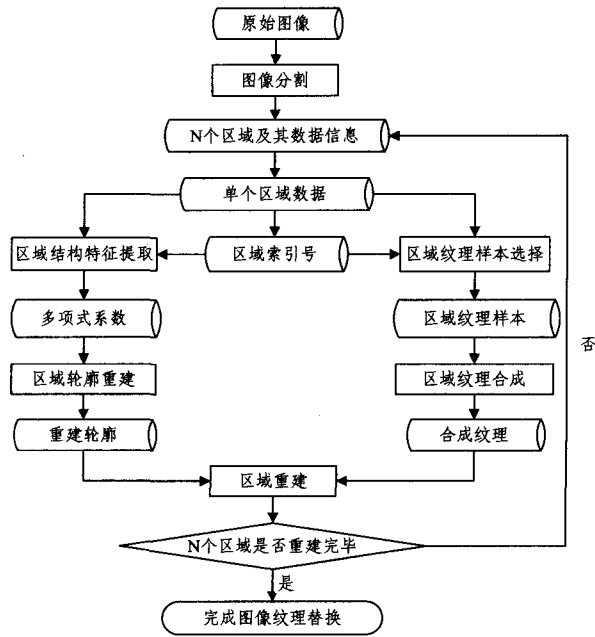


图1 图像纹理替换串行流程图

## 2.1 区域轮廓的结构特征提取与重建

区域轮廓的结构特征提取过程分为3个阶段:区域轮廓跟踪、下采样及分段迭代曲线拟合,得到代表区域轮廓结构特征的多项式系数  $p_0, p_1, \dots, p_m$ 。

首先,利用区域轮廓跟踪<sup>[13]</sup>得到一个由区域边界曲线上的像素点组成的有序序列。区域轮廓跟踪执行过程如下:

(1)以区域轮廓曲线上的起始点作为邻域的中心点,沿逆时针方向开始寻找下一个候补点。

(2)判断候补点是不是在对象区域中,如果在,就把该候补点作为邻域的中心点,重复步骤(1),直至得到轮廓上所有像素点的有序序列,即确定出区域轮廓曲线。

其次,对轮廓曲线上的所有像素点组成的有序序列,按照等长间隔或随机间隔进行有序抽样即下采样提取,得到代表区域轮廓的有序向量  $(x_s, y_s)$ ,  $s$  是指向量元素在轮廓曲线上的位置序号。

最后,利用分段迭代曲线拟合<sup>[14]</sup>提取代表区域轮廓结构特征的多项式系数  $p_0, p_1, \dots, p_m$ 。分段迭代曲线拟合的具体执行过程如下:

(1)找到区域轮廓曲线上相距最远的两个点  $A$  和  $B$ ,用直线段  $L_{AB}$  链接这两个点,把轮廓曲线分成两个曲线段  $AB$  和  $BA$ 。

(2)利用式(2)计算轮廓曲线上的点  $(x_s, y_s)$  到直线段  $L_{AB}$  的垂直距离。具体过程如下:设定轮廓点  $A$  和  $B$  的坐标分别是  $(x_A, y_A)$  和  $(x_B, y_B)$ , 直线段  $L_{AB}$  可由式(1)表示:

$$x(y_A - y_B) + y(x_A - x_B) + x_A y_B - x_B y_A = 0 \quad (1)$$

$$d_s = r_s / \Delta_{AB} \quad (2)$$

式中,  $d$  是指轮廓点  $(x_s, y_s)$  到直线段  $L_{AB}$  的距离;  $s$  是指该轮廓点沿着曲线段  $AB$  方向的位置序号;  $r_s$  由式(3)计算得到;  $\Delta_{AB}$  由式(4)计算得到。

$$r_s = x_s(y_A - y_B) - y_s(x_A - x_B) + x_A y_B - x_B y_A \quad (3)$$

$$\Delta_{AB} = \|A - B\| \quad (4)$$

因此,最大绝对误差  $MAE$  的计算公式如下:

$$MAE = \max_{s \in [A, B]} |d_s| \quad (5)$$

(3)如果轮廓点  $C$  的  $MAE$  大于分裂阈值,那么曲线段  $AB$  就应该分成两个子曲线段  $AC$  和  $CB$ ;同时轮廓点  $C$  就成为一个新的顶点。

(4)对分裂得到的子曲线段  $AC$  和  $CB$ ,重复(2)、(3)步操作,直至所有子曲线段的最大绝对误差都小于分裂阈值。

(5)对分段迭代得到的所有子曲线段进行多边形曲线拟合操作,得到每一个子分段曲线的多项式系数。

这里,我们仅以子曲线段  $AC$  为例来具体阐述曲线拟合过程。假定子曲线段  $AC$  由有序向量  $(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$  来表示。通常,一个多项式可以由若干个正交多项式相加得到。假定拟合多项式如下所示:

$$\begin{aligned} F_m(x) &= \sum_{i=0}^m C_i P_i(x) \\ &= C_0 P_0(x) + C_1 P_1(x) + \dots + C_m P_m(x) \\ &= p_0 + p_1 x + p_2 x^2 + \dots + p_m x^m \end{aligned} \quad (6)$$

正交多项式能够通过式(7)构造得到:

$$\begin{aligned} P_0(x) &= 1 \\ P_1(x) &= (x - A_1) P_0(x) \\ P_2(x) &= (x - A_2) P_1(x) - B_1 P_0(x) \\ &\dots \end{aligned} \quad (7)$$

$$P_{j+1}(x) = (x - A_{j+1}) P_j(x) - B_j P_{j-1}(x)$$

其中,  $j=0, 1, \dots, m$ 。

$$A_{j+1} = \frac{\sum_{i=1}^n x_i P_j^2(x_i)}{\sum_{i=1}^n P_j^2(x_i)}, B_j = \frac{\sum_{i=1}^n x_i P_j^2(x_i)}{\sum_{i=1}^n P_{j-1}^2(x_i)}, C_j = \frac{\sum_{i=1}^n y_i P_j^2(x_i)}{\sum_{i=1}^n P_j^2(x_i)} \quad (8)$$

最后,通过上述  $C_j$  和  $\{P_j(x)\}$  计算得到代表区域轮廓结构特征的多项式系数  $p_0, p_1, \dots, p_m$ 。

区域轮廓曲线能够通过多项式函数来描述,多项式函数定义如下:

$$y = p_0 + p_1 x + p_2 x^2 + \dots + p_m x^m \quad (9)$$

首先,依据两个端点的  $X$  方向的坐标,例如  $x_m$  和  $x_n$ , 得到横坐标向量  $\vec{x} = \{x_m, x_{m+1}, \dots, x_n\}$ , 向量中的每一个元素和轮廓曲线上的每一个点的  $X$  方向的坐标是一一对应的。其次,把式(9)中的横坐标向量  $\vec{x}$  中的每一个元素都替换成相应的纵坐标向量中的每一个元素,那么就可以得到轮廓曲线上每一个点的  $Y$  方向的坐标。最后,利用上述轮廓曲线上每一个点的横、纵坐标重建区域轮廓。为了保证重建的区域轮廓曲线是封闭的,本文算法中可采用一个简单的膨胀算法。

## 2.2 区域纹理的样本选择与合成

选择的区域纹理样本应该包含区域内部纹理的大部分局部及全局特征。利用二维自回归统计分析模型对区域内部纹理进行分析<sup>[15]</sup>。图像的自回归函数能被用于评估图像内部纹理的规则、细/粗糙程度。图像的自回归函数的定义如下所示:

$$\rho_{II}(x, y) = \frac{MN}{(M-x)(N-y)} \times \frac{\sum_{u=1}^{M-x} \sum_{v=1}^{N-y} I(u, v) I(u+x, v+y)}{\sum_{u=1}^M \sum_{v=1}^N I^2(u, v)} \quad (10)$$

对于规则的结构性纹理而言,函数将显示出波峰和波谷,故能容易地确定来自区域内部纹理的纹理基元的尺寸。一般情况下,样本应该包含 2~5 个纹理基元。而对于随机性纹理而言,超过 90% 的函数值都会大于 0.9,因此它的纹理样本尺寸应该被设置得小一些。

利用选取的区域纹理样本合成区域纹理<sup>[16,17]</sup>。最后,把合成的区域纹理填充到重建的区域轮廓中,实现区域恢复。图像中的所有区域都被成功地恢复,就意味着图像纹理替换完成。

### 3 混合 MPI 与 OpenMP 的图像纹理替换并行算法

基于以上理论可以看出,区域轮廓结构特征提取、区域轮廓重建、区域纹理样本选择、区域纹理合成 4 个阶段,在时间与操作上存在数据继承性。虽然每个阶段是流水执行,但不同区域之间的操作却无数据依赖性,这些特征为开发并行算法提供了可能。图像纹理替换的并行算法设计包括 MPI 并行设计与 OpenMP 并行设计。图 2 给出了混合 MPI 与 OpenMP 的图像纹理替换并行算法流程图。

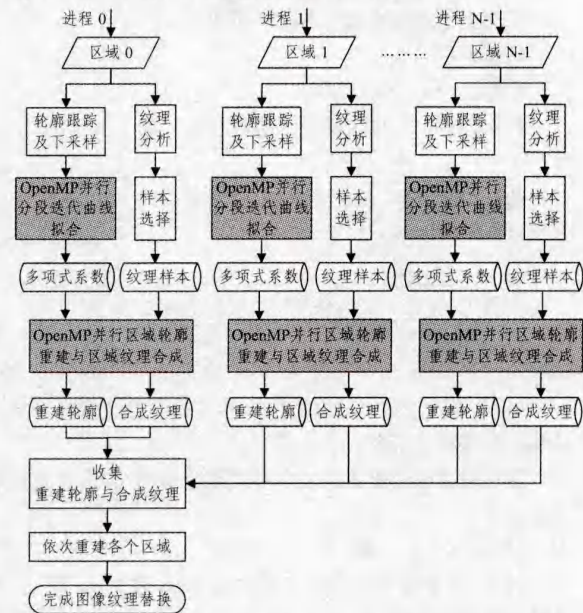


图 2 混合 MPI 与 OpenMP 的图像纹理替换并行算法流程图

#### 3.1 MPI 并行设计

本文根据区域之间潜在的并发特性,采用基于 Ian Foster 描述的任务/通道模型设计方法<sup>[18,19]</sup>开发基于数据并行的分布式 MPI 并行设计。将并行计算表示为一系列任务,任务之间通过使用通道发送消息进行相互通信。

为了阐述方便,假定原始图像的区域个数是  $N$ ,则每个区域的索引号分别是  $0, 1, 2, \dots, N$ 。在 MPI 并行执行过程中,每个区域映射为一个计算任务,每个计算任务对应一个进程。主进程的标识号是 0,其他进程的标识号分别是  $1, 2, \dots, N$ 。负责某个区域计算任务的进程的标识号和该区域的索引号必须是相同的,例如进程 0 负责索引号为 0 的区域。同时,为了减少通信消耗、提高运算速度,本文采用同步通信策略。为了最小化通信代价,本文采用 MPI 组通信完成进程间数据通信。

首先,使用 MPI\_Bcast() 将进程 0 得到的原始图像数据与区域索引列表分发给每一个进程。根据区域索引列表

表,每一个进程提取所负责区域的轮廓结构特征并实现区域轮廓重建;选取所负责区域的纹理样本并合成区域纹理。其次,使用 MPI\_Gather() 把重建的区域轮廓与合成的区域纹理收集到进程 0。最后,进程 0 依次重建各个区域,实现图像纹理替换。

本文使用任务/通道示意图直观地描述基于主从模式的 MPI 并行设计的进程间通信过程,如图 3 所示。带箭头的虚线是通道,用于进程 0 发送原始图像数据和区域索引列表。加粗箭头是通道,用于进程 0 收集重建的区域轮廓与合成的区域纹理。显而易见,其他进程之间没有通信开销。

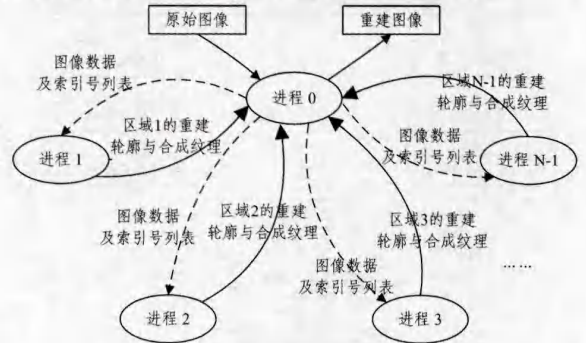


图 3 MPI 并行任务/通道示意图

#### 3.2 OpenMP 并行设计

为了充分利用图像纹理替换的潜在并行特性,本文采用基于 fork / join 模型设计方法<sup>[18]</sup>开发基于数据并行与功能并行的多线程 OpenMP 并行设计。基于共享存储模型的 OpenMP 并行执行过程中,线程之间不会产生通信开销。

由 2.1 节可知,分段迭代拟合曲线在区域轮廓结构特征提取过程中起着关键的作用,因此它的执行时间不容忽视;而且随着图像尺寸的增加,执行分段迭代曲线拟合所消耗的时间必将增加,这必然会影响到并行算法的效率。为了解决这个问题,本文开发了分段迭代曲线拟合的 OpenMP 并行设计。

由第 2 节可知,区域轮廓重建与区域纹理合成是图像纹理替换的两个不同的功能模块,可把它们分配到两个线程中并行执行,因此本文开发了区域轮廓重建与区域纹理合成的 OpenMP 并行设计。图 4 给出了两个 OpenMP 并行设计的伪代码。

```

.....
#pragma omp parallel sections
{
    #pragma omp section
    {
        /* 计算子曲线段的多项式系数 */
    }
    #pragma omp section {
        /* 计算子曲线段的多项式系数 */
    }
    .....
    #pragma omp section
    {
        /* 计算子曲线段的多项式系数 */
    }
}
.....

```

```

.....
#pragma omp parallel sections
{
    #pragma omp section
    {
        /* 重建区域轮廓 */
    }
    #pragma omp section
    {
        /* 合成区域纹理 */
    }
}
.....

```

图4 OpenMP 并行设计伪代码

### 3.3 负载均衡分析

混合 MPI 与 OpenMP 的图像纹理替换并行算法能实现有效的负载均衡。

一方面,在 MPI 并行执行过程中,各个区域具有相同的纹理样本尺寸与纹理合成参数,因此区域纹理的样本选择与合成所用时间是一样的。而各个区域的轮廓结构是不同的,因此区域轮廓的结构特征提取与重建所用时间是不相同的。但是,在实际测试中,由于后者所用时间少于前者所用时间,因此进程之间的总执行时间的差异是小的。

另一方面,虽然每个区域的轮廓结构是不同的,但是分段迭代曲线拟合的 OpenMP 并行执行时间几乎是相同的。而区域轮廓重建与区域纹理合成的 OpenMP 并行执行时间是

相同的。因此,负责各个区域的进程之间的总执行时间的差别将变得更小。OpenMP 并行设计有利于负载均衡。

## 4 实验结果与分析

本节将评估混合 MPI 与 OpenMP 的图像纹理替换并行算法的性能,并对串行算法与并行算法的性能作比较。本文使用包含大量纹理的自然场景图像作为测试图像。在图像分割时,为了使图像同质检测发挥良好的性能,需设置 3 个参数<sup>[11]</sup>: $TQUAN=-1$ , $NSCALE=-1$ , $Threshcolor=0.8$ 。在分段迭代曲线拟合阶段,分裂阈值通常在 5 至 10 之间取值。JPEG2000 的版本是 Kakadu\_V2.2.3。

### 4.1 实验环境

本算法用 C 语言、MPI 及 OpenMP 实现。实验环境是高性能计算集群系统,配置了符合 MPI-2 标准的消息通信库。该集群系统有 8 个计算节点,每个节点的配置为:2 个 Intel Quad-core Xeon X5430(2.66GHz)处理器、16 GB ECC Fully buffered DDR2 667MHz 内存、320GB SATA II 3.0Gb/s 硬盘、Infiniband 网络连接接口卡、双千兆以太网接口、操作系统为 Windows HPC Server 2008 与 RedHat Linux。

### 4.2 实验结果

图像纹理替换的目的不是为了使重建图像与原始图像完全一致,而是尽可能使人眼看起来自然、不令人讨厌。实验结果(见图 5(a)-(o))表明,图像纹理替换结果存在一定程度的视觉损伤。而区域的非纹理细节信息的丢失是引起视觉损伤的主要原因。

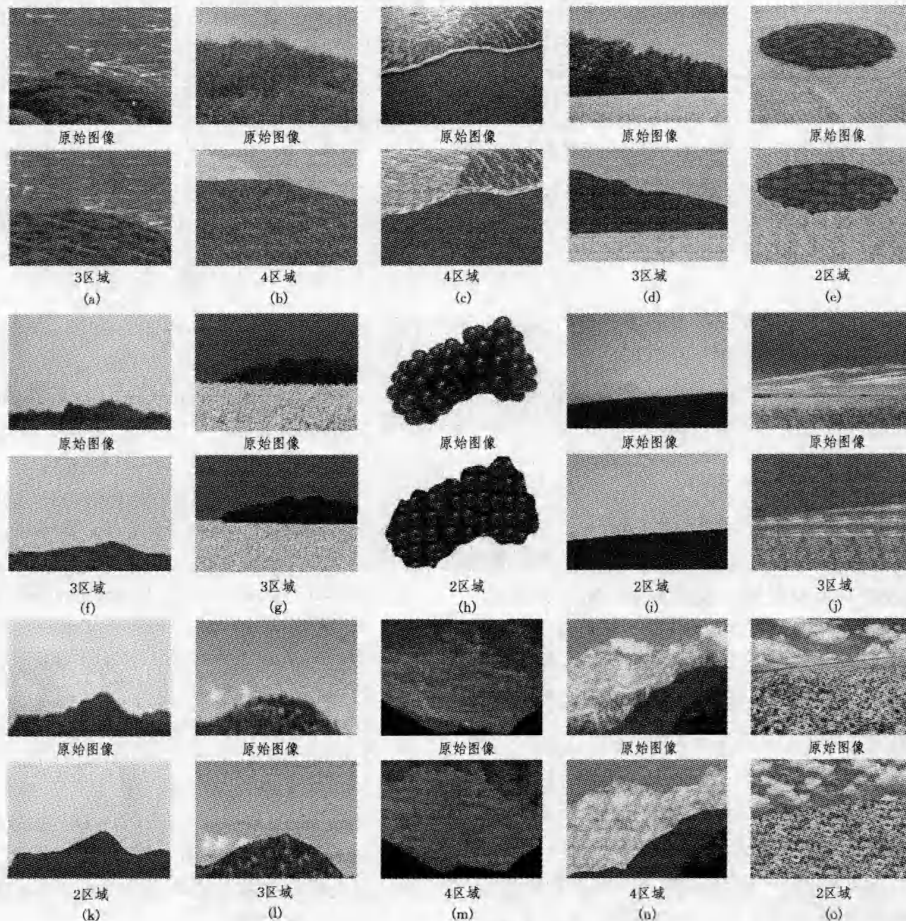


图5 图像纹理替换结果

但是,由人类视觉感知理论可知,人眼对平滑区域的敏感性远高于纹理密集区域,从而在一定程度上降低了观察者对重建图像的主观视觉误差。同时,如果这些由大量纹理构成的重建图像被用于构成图像或视频的静止背景,而图像的前景是骏马、苍鹰或美女等,视频的前景是飞驰的猎豹、飞翔的苍鹰或游泳的美女等,那么人眼的视觉注意力<sup>[20]</sup>肯定集中于上述前景,而静止背景的视觉关注度是低的,尤其是纹理静止背景。

图像纹理替换所需的平均数据量(字节)的计算公式如下:

$$\Omega = \sum_{i=1}^N (Q_i + T_i) \quad (11)$$

式中, $\Omega$ 是指图像纹理替换所需的平均数据量(字节); $N$ 是指区域个数; $Q_i$ 是指表征索引号为 $i$ 的区域结构特征的多项式系数所需的字节数,每一个系数为8字节; $T_i$ 是指在索引号为 $i$ 的区域中选取的纹理样本所需的字节数。

表1给出了图像纹理替换所需的平均数据量(字节)。显而易见,图像纹理替换能大大减少图像重建所需的数据量。

表1 图像纹理替换所需数据量(单位:字节)

原始图像	JPEG2000	本文算法		
		$\Omega$	$Q$	$T$
图5(c)4区域	25538	12340	2944	9396
图5(d)3区域	14481	6361	736	5625
图5(e)2区域	29475	8799	1248	7551
图5(g)3区域	12553	4935	1344	3591
图5(h)2区域	30988	15542	1664	13878
图5(j)3区域	11155	6128	512	5616
图5(k)2区域	5364	2992	832	2160
图5(m)4区域	7354	5540	2432	3108
图5(o)2区域	17028	6038	320	5718

### 4.3 效率对比

为了衡量混合MPI与OpenMP的图像纹理替换并行算法的性能,我们通过加速比与并行效率这两个指标来分析。

加速比 $S_p$ 定义如下:

$$S_p = \frac{T_s}{T_p} \quad (12)$$

式中, $T_s$ 为最优串行算法在单处理机上的运行时间; $T_p$ 为并行算法在集群上使用 $p$ 个处理器时的运行时间。

并行效率 $E_p$ 用来度量并行算法对处理器的利用效率,以补偿加速比评价标准的不足。并行效率定义如下:

$$E_p = \frac{S_p}{p} \quad (13)$$

式中, $S_p$ 为加速比; $p$ 为处理器个数。

使用图5(c)和图5(e)中的原始图像作为测试并行算法的执行时间、加速比和并行效率的实验图像。同时考虑到原始图像的实际形状与纹理特征,图5(c)中的原始图像的区域个数为2,3,4和5。图5(e)中的原始图像的区域个数为2,3,4,5,6,7和8。并行算法的线程数是可以任意设置的,实际测试时设定线程数为2。测试结果如图6所示。

通过对测试结果的对比分析,发现并行算法的执行时间明显少于串行算法,而且区域个数越多,节省的执行时间也越多。并行算法能得到高的加速比,而且区域个数越多,加速比就会越大。然而当处理器的个数增加时,并行效率将降低,因此对于图像纹理替换的并行算法设计仍需进一步研究和改进。综上所述,混合MPI与OpenMP的图像纹理替换并行算

法是切实可行的。随着并行计算技术的进一步发展,本文并行算法必将获得更好的性能。

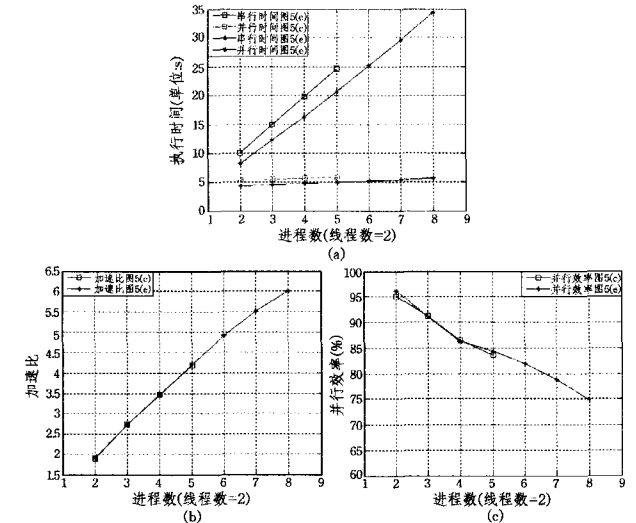


图6 并行算法的执行时间、加速比与并行效率

**结束语** 本文提出的混合MPI与OpenMP的图像纹理替换并行算法能选用最小的区域纹理样本,并使用区域的轮廓结构与纹理信息简洁可靠地表征图像。实验结果表明,并行算法能获得符合人眼视觉要求的、恰当的重建图像,大大减少了图像重建所需的数据量;并行算法的执行时间明显低于串行算法,而且区域个数越多,并行算法节省的图像重建时间就越多,加速比就越大。图像纹理替换并行算法的设计还有很多问题有待研究,今后的工作将围绕以下几个方面展开:

(1)设计非纹理区域的功能模块,并把该功能模块融入到图像纹理替换中,以构建更加成熟的编码框架,重建更加逼真的图像;

(2)通过集群计算机系统来实现图像纹理替换并行算法虽然便于对算法的验证,但系统对资源的利用率不高。下一步拟采用可编程逻辑器件(FPGA)进一步提高系统性能。

### 参考文献

- [1] Sikora T. Trends and perspectives in image and video coding [J]. Proceedings of the IEEE, 2005, 93(1): 6-17
- [2] Kunt M, Ikononopoulos A, Kocher M. Second-generation image coding techniques [J]. Proceedings of the IEEE, 1985, 73(4): 549-574
- [3] Ndjiki-Nya P, Makai B, Blattermann G. Improved H. 264/AVC coding using texture analysis and synthesis [C]// IEEE International Conference on Image Processing. 2003, 3: 849-852
- [4] Sullivan G J, Wiegand T. Video Compression-From Concepts to the H. 264/AVC Standard [J]. Proceedings the IEEE, Special Issue on Advances in Video Coding and Delivery, 2005, 93(1): 18-31
- [5] Geer D. For programmers, multicore chips mean multiple challenges [J]. Computer, 2007, 40(9): 17-19
- [6] Jones M D, Yao R, Bhole C P. Hybrid MPI-OpenMP Programming for Parallel OSEM PET Reconstruction [J]. IEEE Transactions on nuclear science, 2006, 53(5): 2752-2758
- [7] 陈荣鑫, 廖湖声, 陈维斌. 基于并行子树构建的XML解析方法 [J]. 计算机科学, 2011, 38(3): 191-194

(下转第299页)

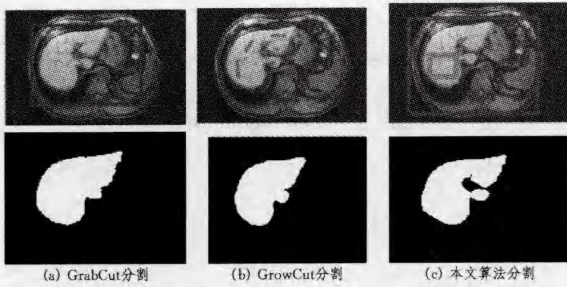


图9 肝脏图像分割3种算法分割效果比较

图8(a)和图9(a)是采用 GrabCut 算法分割的效果图,图8(a)中 GrabCut 分割无法区分出每颗牙,且上下牙之间的部分也作为前景提取了出来;图9(a)的分割轮廓不够清晰。图8(b)和图9(b)是采用 GrowCut 方法分割的结果,由于初始种子定义的不同,导致分割效果出现误差,如图8(b)中牙齿图像的左上部、图9(b)中肝脏中间的沟回分割都不够精确。图8(c)和图9(c)是采用本文提出的改进方法进行分割的结果,其灰度直方图和种子模板分别如图10和图11所示。由于利用灰度直方图自动生成种子模板,使得用户标记区域内的像素点均可以根据其类别进行标记,从而得到较好的分割效果。

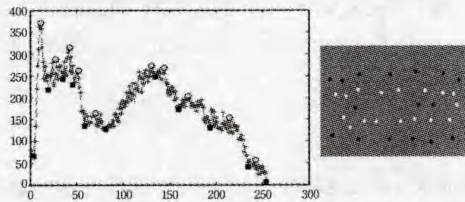


图10 牙齿斑口腔图像的多阈值直方图和自动种子模板

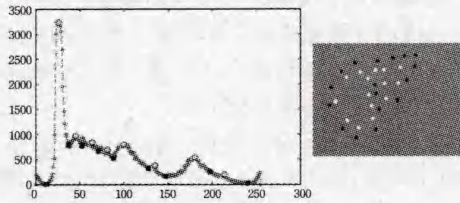


图11 肝脏图像的多阈值直方图和自动种子模板

以上实验结果表明,利用图像的多阈值直方图自动生成

初始种子模板可以达到和手工设置种子模板相一致的效果,但却节省了人工设置种子点的多次交互操作,且利用改进的迭代算法可以达到更好的分割结果,提高了算法的鲁棒性。

**结束语** 本文针对 GrowCut 算法中初始种子标记要求用户较多工作量的问题,提出了简化标记、自动生成初始种子模板的多阈值优化细胞自动机图像分割方法。算法结合了图像的像素信息,简化了用户的操作,提高了算法的分割精度,对边界模糊的医学图像分割具有良好的效果。与自动分割方法相比,交互式图像分割可以融合用户的先验知识,具有更高的研究价值和更广的应用领域,不过交互式图像分割的研究目的是通过尽可能简便而直观的人机交互,快速而准确地提取出图像中具有高层语义的前景物体。因此,尽可能使用户通过简明的交互方式对高分辨率图像进行准确分割是未来进一步深化研究的重点。

## 参考文献

- [1] Mortensen E N, Barrett W A. Interactive segmentation with intelligent scissors[J]. Graphical Models and Image Processing, 1998, 60(5): 349-384
- [2] Kass M, Witkin A, Terzopoulos D. Snakes: Active contour models[J]. International Journal of Computer Vision, 1988, 1(4): 321-331
- [3] Osher S, Fedkiw R. Level Set Methods and Dynamic Implicit Surfaces[M]. Springer-Verlag, New York, 2002
- [4] Li Yin, Sun Jian, Tang C K, et al. Lazy Snapping [J]. ACM Transaction on Graphics, 2004, 24(3): 303-308
- [5] Boykov Y, Jolly M P. Interactive graph cuts for optimal boundary and region segmentation of objects in N-D images[C]//IEEE International Conference on Computer Vision, 2001: 102-115
- [6] Rother C, Kolmogorov V, Blake A. Grabcut-Interactive Foreground Extraction using Iterated Graph Cuts[J]. ACM Transaction on Graphics, 2004, 24(3): 309-314
- [7] Von Neumann J. Theory of Self-Reproducing Automata [M]. Theory of Self-Reproducing Automate, 1966
- [8] Thomas C D. Evolution of Cellular Automata for Image Processing[D]. University of Birmingham, April 2000
- [9] Vezhnevets V, Konouchine V. GrowCut-interactive multi-label ND image segmentation by cellular automata[C]//Proceeding of Graphicon, 2006: 231-234

(上接第 292 页)

- [8] Chen L, Zhou J, Wang D, et al. Parallel Image Registration with OpenMP and MPI [C]//International Conference on Computational Intelligence and Software Engineering, 2009: 1-4
- [9] 李锦, 钟诚. 异构机群上高效可扩展的 Motif 发现并行算法[J]. 计算机科学, 2012, 39(3): 279-282
- [10] Deng Y, Manjunath B S, Shin H. Color image segmentation [C]//IEEE Conference on Computer Vision and Pattern Recognition, 1999: 446-451
- [11] Deng Y, Manjunath B S. Unsupervised segmentation of color-texture regions in images and video [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2001, 23(8): 800-810
- [12] Jing F, Li M, Zhang H J, et al. An efficient and effective region-based image retrieval framework [J]. IEEE Transactions on Image Processing, 2004, 13(5): 699-709
- [13] Adamek T, O'Connor N. Efficient contour-based shape representation and matching [C]//Proc. 5th ACM SIGMM Int. Workshop on Multimedia Information Retrieval, 2003: 138-143
- [14] Sun K, Yang Y Y, et al. Image restoration using piecewise itera-

tive curve fitting and texture synthesis [C]//International Conference on Intelligent Computing, 2008, 2: 1056-1063

- [15] Chen C H, Wang P S P. Handbook of pattern recognition and computer vision [M]. World Scientific Publishing Co., 1999: 218-222
- [16] Efros A A, Freeman W T. Image quilting for texture synthesis and transfer [C]//ACM SIGGRAPH, 2001: 341-346
- [17] Liang L, Liu C, Xu Y Q, et al. Real-time texture synthesis by patch-based sampling [J]. ACM Transactions on Graphics, 2001, 20(3): 127-150
- [18] Quinn M J. Parallel Programming in C with MPI and OpenMP [M]. McGraw-Hill Companies, Inc., 2004
- [19] 都志辉. 高性能计算并行编程技术—MPI 并行程序设计[M]. 北京:清华大学出版社, 2001
- [20] Guo C L, Ma Q, Zhang L M. Spatio-Temporal Saliency Detection using Phase Spectrum of Quaternion Fourier Transform [C]//IEEE Conference on Computer Vision and Pattern Recognition, 2008: 1-8