

对基于 facet 模型的表面检测的加速技术研究

王 凯¹⁾ 张定华¹⁾ 张顺利¹⁾ 黄魁东¹⁾ 刘 晶²⁾

¹⁾ (西北工业大学现代设计与集成制造技术教育部重点实验室, 西北工业大学, 西安 710072)

²⁾ (华东理工大学机械与动力工程学院, 上海 200237)

摘 要 针对基于 facet 模型的亚体素表面检测算法计算量大的问题, 提出将 3 维 facet 模型的可分滤波器递归算法与感兴趣区域加速策略相结合的加速方案。可分滤波器递归算法通过在离散正交基下将 3 维卷积转换为 3 个 1 维卷积并使 1 维卷积递归执行, 使计算量与卷积核大小无关, 大大节省了计算时间。采用增量算法解决了可分滤波器递归算法内存消耗量大的问题。感兴趣区域加速策略采用图像分割后提取目标的分段包围盒作为有效区域, 从而大幅缩减待处理数据量。实验结果表明本文加速方案在保持原始算法精度的同时, 能取得很好的加速效果。

关键词 表面检测 亚体素 facet 模型 可分滤波器 感兴趣区域

中图法分类号: TP391.72 文献标识码: A 文章编号: 1006-8961(2009)02-0328-06

Research on Acceleration Techniques for Facet-model-based Surface Detection

WANG Kai¹⁾, ZHANG Ding-hua¹⁾, ZHANG Shun-li¹⁾, HUANG Kui-dong¹⁾, LIU Jing²⁾

¹⁾ (Key Laboratory for Contemporary Design and Integrated Manufacturing Technology, Ministry of Education, Northwestern Polytechnical University, Xi'an 710072) ²⁾ (School of Mechanical and Power Engineering, East China University of

Science and Technology, Shanghai 200237)

Abstract For large computation in facet-model-based surface detection methods, an acceleration scheme combining separable filter recursive algorithm for 3D facet model with region of interest strategy is proposed. The separable filter recursive algorithm implements the 3D convolution with three 1D convolutions and allows the 1D convolution to be implemented recursively. This significantly reduces the computation time by rendering the computation independent of the kernel size. To solve the subsequent high memory consuming problem of the separable filter recursive algorithm, an incremental method is employed. As for the region of interest strategy, objects' piecewise bonding box extracted after image segmentation is adopted as the valid region. This can greatly decrease the amount of data to be processed. Experiment results show the presented scheme achieved excellent acceleration performance with same accuracy.

Keywords surface detection, subvoxel, facet model, separable filter, region of interest

1 引 言

在基于 CT 的工业应用中, 常常需要从 CT 图像序列获得实体的 3 维高精度表面点云数据, 该环节

误差将直接影响后续检测和重建的精度。经典的边缘检测方法只能达到像素级精度, 而亚像素边缘检测算法基于局部灰度信息, 用连续函数或几何模型来近似边缘模型以获取连续的边缘位置, 达到亚像素级精度。

基金项目: 国家高技术研究发展计划(863)项目(20060104Z1097); “十一五”国家科技支撑计划重点项目(2006BAF04B02); 陕西省教育厅自然科学基金项目(07JK425)

收稿日期: 2007-01-19; **改回日期:** 2007-08-14

第一作者简介: 王 凯(1980 ~), 女。西北工业大学航空宇航制造专业博士研究生。主要研究方向为工业 CT、计算机图形图像处理。

E-mail: lwlwk@yahoo.com.cn

目前国内外学者对 2 维亚像素边缘检测算法的研究较多^[1-5],对 3 维亚体素表面检测讨论相对较少。3 维亚体素表面检测算法能够充分利用相邻层的信息,可获得比 2 维亚像素边缘检测更高的精度和更好的抗噪性,更适合 CT 切片图像序列的检测。现有的亚体素表面检测算法主要是将 2 维边缘检测方法直接扩展到 3 维。如:罗立民推广了 Lyvers^[1]的 2 维空间矩算法,提出了利用一、二阶空间矩拟合理想表面模型确定表面点位置和方法^[6],但由于其采用理想阶梯边缘模型,定位精度受图像模糊度影响较大。Wu 等人将 Haralick 的 2 维方向导数边缘检测器^[2]扩展到了 3 维,通过三次多项式拟合 facet 模型,再由该模型求得梯度方向上的二阶方向导数过零点,确定亚体素表面点位置^[7]。该方法需计算 facet 模型的 20 个系数,每一个系数都需要将一个 3 维模板与图像卷积,计算量很大。

Ji 等人采用可分滤波器及其递归实现加速了 2 维 facet 模型系数计算,通过将 2 维卷积转换为 2 个 1 维卷积并使 1 维卷积递归执行,使计算量与卷积核大小无关^[8]。本文将该技术应用到 3 维 facet 模型系数计算上,并研究了随之产生的存储空间问题。最后与感兴趣区域(ROI)加速策略相结合,减少待检测的像素数目,获得了很好的加速效果。

2 基于 facet 模型的表面检测及其加速

2.1 基于 facet 模型的亚体素表面检测算法

Facet 模型是 Haralick^[9]于 1981 年提出的一种图像表示方法,通过被称为 facet 的分段连续函数来近似表示每个图像像素的一个邻域。基于 2 维 facet 模型,Haralick^[2]实现了亚像素边缘检测,Wu 等人将该方法推广,采用 3 维 facet 模型进行亚体素表面检测^[7]。

设 S 为某一像素的 3 维邻域体素集合, $I(x, y, z)$ 为 $(x, y, z) \in S$ 处的灰度值, $G = \{g_1(x, y, z), \dots, g_N(x, y, z)\}$ 为一组 3 维多项式基函数,则图像灰度拟合函数 f 可被表示为

$$f(x, y, z) = \sum_{i=1}^N K_i g_i(x, y, z) \quad (1)$$

这里, $K_i (i=1, \dots, N)$ 为模型系数,通过最小二乘拟合使目标函数 ε^2 最小:

$$\varepsilon^2 = \sum_{(x, y, z) \in S} [I(x, y, z) - f(x, y, z)]^2 \quad (2)$$

基于 facet 模型的表面检测常采用三次多项式

facet 模型, G 可取一般多项式基^[7]:

$$G = \{1, x, y, z, x^2, y^2, z^2, xy, xz, yz, xyz, xy^2, xz^2, x^2y, yz^2, x^2z, y^2z, x^3, y^3, z^3\}$$

此时计算系数 $K_i (i=1, \dots, 20)$ 的卷积模板为 20 个 3 维模板; G 也可采用 3 维离散正交多项式基^[8],通过 3 个 1 维离散正交多项式集合的张量积构造,这时任一系数的 3 维卷积模板可转换为 3 个 1 维模板,并使 1 维卷积递归执行。

下面给出 3 维离散正交多项式基下系数模板的计算。对于一元三次多项式,在对称域 R 上定义的 4 个 Chebyshev 正交基函数为

$$P_0(r) = 1 \quad (3)$$

$$P_1(r) = r \quad (4)$$

$$P_2(r) = r^2 - \frac{\mu_2}{\mu_0} \quad (5)$$

$$P_3(r) = r^3 - \frac{\mu_4}{\mu_2} r \quad (6)$$

式中, $\mu_t = \sum_{\mu \in R} \mu^t, t=0, 2, 4$ 。

由式(3)~(6)可知,通过张量积构造出的三元三次离散正交多项式基为

$$G = \{P_0(x)P_0(y)P_0(z), P_1(x)P_0(y)P_0(z), P_0(x)P_1(y)P_0(z), P_0(x)P_0(y)P_1(z), P_2(x)P_0(y)P_0(z), P_0(x)P_2(y)P_0(z), P_0(x)P_0(y)P_2(z), P_1(x)P_1(y)P_0(z), P_1(x)P_0(y)P_1(z), P_0(x)P_1(y)P_1(z), P_1(x)P_1(y)P_1(z), P_1(x)P_2(y)P_0(z), P_1(x)P_0(y)P_2(z), P_2(x)P_1(y)P_0(z), P_0(x)P_1(y)P_2(z), P_2(x)P_0(y)P_1(z), P_0(x)P_2(y)P_1(z), P_3(x)P_0(y)P_0(z), P_0(x)P_3(y)P_0(z), P_0(x)P_0(y)P_3(z)\}$$

在该正交基下,利用基函数的正交性,无需求解线性方程组,直接化简即可得到使目标函数 ε^2 (见式(2))最小的 K_i 。

$$K_i = \frac{\sum_{(x, y, z) \in S} g_i(x, y, z) I(x, y, z)}{\sum_{(x, y, z) \in S} g_i^2(x, y, z)} \quad (7)$$

$i = 1, \dots, 20$

根据式(7)可求得 20 个 3 维系数模板,将这些模板与图像卷积获得 facet 模型系数,建立局部区域灰度的连续表达式。对该连续表达式,求梯度方向上的二阶方向导数过零点,获得亚体素表面点位置,具体参见文献[7]。

2.2 基于可分滤波器的加速算法

前面已经指出,若 G 采用正交多项式,3 维系数模板可以转换为 1 维模板。设式(7)中的 $g_i(x, y, z) =$

$P_u(x)P_v(y)P_w(z)$, $i = 1, \dots, 20$ 是定义在 $S = S_x \times S_y \times S_z$ 上的 3 维正交多项式基, 其中 $S_x = \{-m_x, -m_x + 1, \dots, 0, \dots, m_x\}$, $S_y = \{-m_y, -m_y + 1, \dots, 0, \dots, m_y\}$, $S_z = \{-m_z, -m_z + 1, \dots, 0, \dots, m_z\}$, m_x, m_y 和 m_z 为正整数, 则式(7)可表示为

$$K_i = \frac{\sum_{z=-m_z}^{m_z} \sum_{y=-m_y}^{m_y} \sum_{x=-m_x}^{m_x} g_i(x, y, z) I(x, y, z)}{\sum_{z=-m_z}^{m_z} \sum_{y=-m_y}^{m_y} \sum_{x=-m_x}^{m_x} g_i^2(x, y, z)}$$

$$= \frac{\sum_{z=-m_z}^{m_z} P_w(z)}{\sum_{z=-m_z}^{m_z} P_w^2(z)} \cdot \frac{\sum_{y=-m_y}^{m_y} P_v(y)}{\sum_{y=-m_y}^{m_y} P_v^2(y)}$$

$$\frac{\sum_{x=-m_x}^{m_x} P_u(x) I(x, y, z)}{\sum_{x=-m_x}^{m_x} P_u^2(x)} \quad (8)$$

由式(8)可见, 计算系数 K_i 的 20 个 3 维模板可转换为 20 个 X 方向 1 维模板, 20 个 Y 方向 1 维模板和 20 个 Z 方向 1 维模板。每个方向上的 20 个 1 维模板只有 4 个是互不相同的, 对应 4 个 1 维 Chebyshev 正交基函数。这样就将 20 个 3 维卷积模板减少到 12 个 1 维模板。时间复杂度由原始的 $O(\tilde{N}^2 \cdot \tilde{m}^3)$ 降低到 $O(\tilde{N}^2 \cdot \tilde{m})$, \tilde{N}^2 为图像大小, \tilde{m} 为 3 维邻域长度。

原始 3 维模板, 可一次卷积获得待求 facet 模型系数, 而可分滤波器算法, 对 3 个方向的 1 维卷积需串行进行, 后一步卷积需要前一步结果作为输入, 因此, 需要额外空间存储中间计算结果。若要保存图像序列所有层 Y 向卷积后结果, 再进行 Z 向滤波, 内存耗费量太大。因此考虑在 Z 向计算时采用增量算法: 对当前待处理图像及其上下相邻 m_z 层图像进行 X 向和 Y 向卷积, 将 Y 向卷积后结果保存在内存中, 再对这 $(2m_z + 1)$ 层结果做 Z 向卷积得到当前待处理图像的所有系数 K_i , 根据这组系数进行后续的亚体素表面点参数计算, 得到最终的表面点位置, 至此, 当前待处理图像处理完毕。在对下一层待处理图像处理时, 由于内存中已保留该层相邻的 $-m_z$ 到 $m_z - 1$ 层 Y 向卷积后结果, 而内存中最顶层 Y 向卷积后结果在后续处理中不再需要, 因此将内存中除顶层数据均上移一层(交换指针), 最顶层数据被覆盖, 最底层用来存储新读入的第 m_z 层图像在 Y 向卷积后的结果, 这样就可以计算当前待处理

图像 Z 向卷积结果及表面点位置, 如此逐层处理。采用增量算法在整个处理过程中始终仅需保存 $(2m_z + 1)$ 层的中间数据, 大大节省了内存空间。以处理 1024^3 大小的图像为例, 参数 $m_z = 2$, 中间结果采用 float 类型保存, 不采用增量算法计算 20 个系数共需要 $20 \times 1024^3 \times 4\text{byte} = 80\text{ GB}$, 采用增量算法需要 $20 \times 5 \times 1024^2 \times 4\text{byte} = 400\text{ MB}$ 空间, 对于处理通常尺寸的图像, 一般计算机配置均能满足要求。

2.3 1 维卷积递归实现

采用可分滤波器可将时间复杂度降低到 $O(\tilde{N}^2 \cdot \tilde{m})$, 但计算量仍会随卷积模板的增大而增加。文献[8]推导了 1 维模板卷积的递归实现, 使计算量与模板尺寸无关, 复杂度为 $O(\tilde{N}^2)$ 。这里对文献[8]中未进行归一化处理的递归公式加以修正, 具体推导过程不再赘述。

设沿某一轴向上的 1 维卷积模板大小为 $2m + 1$, a_1, \dots, a_m 为输入数据。根据式(8), 输入数据与该轴向上的 4 个 1 维模板的卷积可表示为

$$A_n = \frac{\sum_{r=-m}^m 1 \cdot a_{n+r}}{\sum_{r=-m}^m 1^2}, B_n = \frac{\sum_{r=-m}^m r a_{n+r}}{\sum_{r=-m}^m r^2}$$

$$C_n = \frac{\sum_{r=-m}^m (r^2 - a) a_{n+r}}{\sum_{r=-m}^m (r^2 - a)^2}, D_n = \frac{\sum_{r=-m}^m (r^3 - br) a_{n+r}}{\sum_{r=-m}^m (r^3 - br)^2}$$

式中, $a = \frac{\mu_2}{\mu_0}; b = \frac{\mu_4}{\mu_2}$ 。

则 1 维卷积运算的递归公式为

$$A_{n+1} = A_n + \frac{1}{Q_0} [a_{n+m+1} - a_{n-m}] \quad (9)$$

$$B_{n+1} = B_n - \frac{Q_0}{Q_1} A_{n+1} + \frac{1}{Q_1} [(m+1)a_{n+m+1} + ma_{n-m}] \quad (10)$$

$$C_{n+1} = C_n - \frac{Q_0}{Q_2} A_{n+1} - 2 \frac{Q_1}{Q_2} B_{n+1} + \frac{(m+1)^2 - a}{Q_2} a_{n+m+1} - \frac{m^2 - a}{Q_2} a_{n-m} \quad (11)$$

$$D_{n+1} = D_n - (3a - b + 1) \frac{Q_0}{Q_3} A_{n+1} - 3 \frac{Q_1}{Q_3} B_{n+1} - 3 \frac{Q_2}{Q_3} C_{n+1} + \frac{(m+1)^3 - b(m+1)}{Q_3} a_{n+m+1} + \frac{m^3 - bm}{Q_3} a_{n-m} \quad (12)$$

式中,

$$Q_0 = \sum_{r=-m}^m 1^2, Q_1 = \sum_{r=-m}^m r^2$$

$$Q_2 = \sum_{r=-m}^m (r^2 - a)^2, Q_3 = \sum_{r=-m}^m (r^3 - br)^2$$

采用 1 维卷积的递归实现,初始像素卷积结果通过模板运算获得,而后每一个像素结果根据式(9)~式(12)从前一像素结果推出,这样若图像比较大,处理过程采用的精度不够高,就可能带来较大的累积误差。图 1 为对一幅 512×512 大小的图像分别采用 float 类型、double 类型变量进行递归计算检测到的边缘图像,易见采用 float 类型累积误差很大,影响最终的边缘检测结果,而采用更高精度的 double 类型,累积误差对边缘基本没有影响。虽然采用 double 类型可有效地降低累计误差,但内存耗用量将至少为采用 float 类型的两倍,若要处理较大尺寸图像,就需配置较高的内存,如邻域窗口取 5×5 时,处理 1024^3 大小的图像序列,需 1 GB 左右内存。

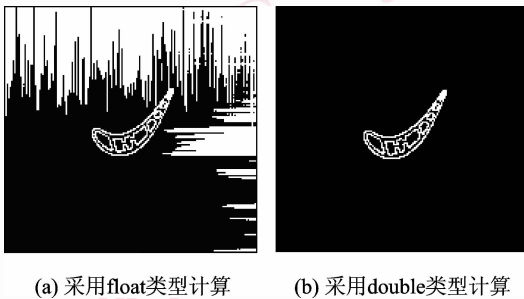


图 1 累积误差对表面检测结果影响

Fig. 1 Influence of accumulated error on surface detection results

2.4 整体加速方案

考虑将以上可分滤波器递归加速算法与感兴趣区域技术相结合,实现了两种算法的串联加速,从而达到了很高的加速比。

基于感兴趣区域的加速策略是通过缩小图像中待处理区域的范围,仅处理感兴趣区域内的像素,以达到缩减图像总体处理时间的目的。感兴趣区域越小,总体处理时间就越少,因此,通常将边缘及其邻域像素定义为感兴趣区域,把这种感兴趣区域称为基于边缘的 ROI,可以通过以下步骤获得:首先采用较为简单的图像分割算法(如:Sobel、Prewitt、LOG、Canny 等边缘检测算子或 Otsu、迭代阈值分割算法)

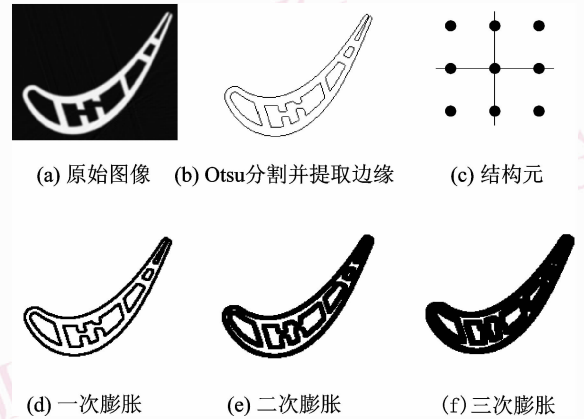


图 2 基于边缘的 ROI 获取

Fig. 2 Acquisition of edge-based ROI

获取边缘点(如图 2(b)所示);然后采用形态学膨胀算法将边缘扩展,得到边缘点及其邻域像素(如图 2(d)所示),这些像素就构成了基于边缘的 ROI。为保证获得的 ROI 的完备性,即包含所有可能的边缘像素,可通过增大膨胀算法的结构元尺寸,或采用相同结构元进行多次膨胀运算,进一步扩大候选边缘像素范围(如图 2(e)、(f)所示),但同时增加的候选像素也会延长图像处理时间。

除了以上描述的基于边缘的 ROI,还可采用目标的矩形包围盒作为感兴趣区域,我们称之为基于包围盒的 ROI。对于原始的基于 facet 模型的亚体素表面检测算法,使用边缘 ROI 能最大程度缩减待处理像素数,因此采用边缘 ROI 作为感兴趣区域。对于可分滤波器递归加速算法,边缘 ROI 不适合该算法。因为在使用递归公式(式(9)~式(12))计算当前像素的卷积时,需用到前一个像素的卷积结果,若该值不存在,就要采用 1 维模板计算;而基于边缘的 ROI 由于区域形状复杂,在卷积方向上常会出现待处理像素不连续的情况,需要在断开处采用模板计算卷积值,影响递归算法的加速效果。以图 3 所示的图像沿 X 方向进行卷积为例,灰色像素表示 ROI 区域,阴影覆盖的灰色像素表示需模板计算卷积的区域。从图中可见,边缘 ROI 虽然整体处理像素数比包围盒 ROI 少,但采用模板计算卷积的像素数目却较多,两种包围盒加速效果在很大程度上受到实际目标形状的影响。考虑到包围盒 ROI 提取和描述都较为简单,易于编程实现,将其与可分滤波器递归算法结合进行加速。对于序列图像,3 维包围盒 ROI 通过以下步骤获得:首先采用较为简单的

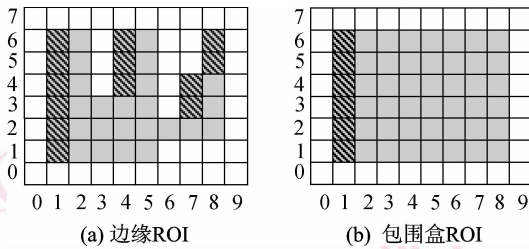


图 3 采用不同 ROI 的可分滤波器递归算法
Fig. 3 The separable filter recursive algorithm under different ROI

图像分割算法对所有切片图像进行分割,并记录各层目标像素沿 X、Y 向的最大最小值,获得每一层的 2 维包围盒;然后从这些 2 维包围盒中抽取目标在 X、Y、Z 方向的最大和最小坐标,得到 3 维整体包围盒。若图像中目标截面变化较大,可相邻若干层计算一个包围盒,即分段包围盒。

采用的亚体素表面检测算法的整体加速方案的处理流程如图 4。

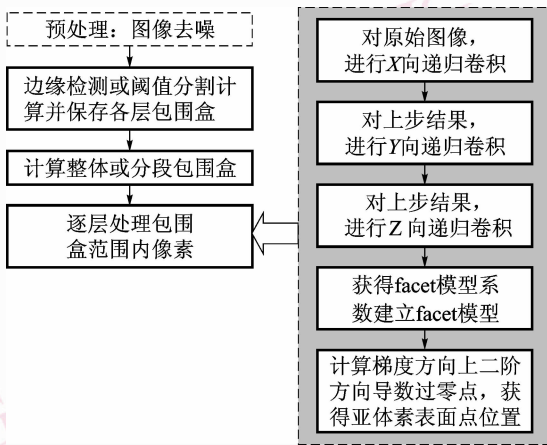


图 4 整体加速方案处理流程图
Fig. 4 Flow chart of global acceleration scheme

3 实验及结果

实验采用西北工业大学现代设计与集成制造技术教育部重点实验室自主研发的锥束 CT 仿真重建系统,从原始 UG/CAD 模型出发,仿真并重建出 409 层大小为 512 × 512 的 bmp 格式切片图像。对该组图像分别采用表 1 中各算法进行处理并比较算法性能,其中递归算法均采用 double 类型变量处理。

实验 1 采用算法 1 ~ 6 对 409 层切片图像进行处理,邻域窗口大小为 5 × 5 × 5,并比较各算法检测精度,如表 2 所示。误差为提取的表面点与原始 CAD

表 1 原始及加速算法

Tab. 1 Original and acceleration algorithms

类型	算法	名称	描述
单独加速	1	基于 facet 模型的亚体素表面检测算法	正交多项式基下 20 个 3 维模板对整幅图像处理
	2	边缘 ROI 的加速算法	正交多项式基下 20 个 3 维模板对边缘 ROI 处理
	3	可分滤波器非递归算法	正交多项式基下 12 个 1 维模板对整幅图像非递归处理
	4	可分滤波器递归算法	正交多项式基下 12 个 1 维模板对整幅图像递归处理
整体加速	5	整体包围盒 ROI 可分滤波器递归算法	正交多项式基下 12 个 1 维模板对整体包围盒 ROI 递归处理
	6	分段包围盒 ROI 可分滤波器递归算法	正交多项式基下 12 个 1 维模板对分段包围盒 ROI 递归处理

表 2 5 × 5 × 5 窗口下各算法精度比较

Tab. 2 Algorithm accuracy comparison with processing window 5 × 5 × 5

算法	平均误差 (pixel)	均方根误差 (pixel)	最大误差 (pixel)	表面点个数
1	0.011 283	0.141 903	1.106 450	472 074
2	0.011 283	0.141 903	1.106 450	472 074
3	0.011 283	0.141 903	1.106 450	472 074
4	0.011 283	0.141 903	1.106 450	472 074
5	0.011 283	0.141 903	1.106 450	472 074
6	0.011 283	0.141 903	1.106 450	472 074

模型的距离,分别计算所有点的平均误差、均方根误差和最大误差以衡量算法精度。由表 2 可见,各加速算法处理结果与原始算法结果相同,精度不变,采用 double 类型变量的递归算法的累积误差对检测结果没有影响。算法 2 采用的是 Otsu 阈值分割后进行两次膨胀(结构元素为 3 × 3 的正方形,原点在正方形中心,如图 2(c)所示)获得的边缘 ROI。在实际处理中,膨胀次数过少可能造成 ROI 不能完全包含所有的表面体素,检测到的表面点仅为原始算法检测到的点的子集,精度有所改变;但膨胀次数较多又会增加处理时间,影响加速效果。

实验 2 采用算法 1 ~ 6 对 409 层切片图像进行处理,邻域窗口尺寸分别取 5 × 5 × 5、7 × 7 × 7、9 × 9 × 9,比较各算法处理速度受窗口大小的影响程度,如

表3所示。时间为处理409幅图像的总时间,加速比为给定窗口下原始算法(算法1)处理时间与任一加速算法处理时间的比值。易见原始算法耗时最长,速度受窗口尺寸影响最大;算法2和3受窗口大小影响次之;各递归算法(算法4,5,6)速度基本不受窗口大小影响。本文提出的整体加速方案中的算法6速度最快,不同窗口下耗时均在2分钟左右。

表3 不同窗口尺寸下各算法速度比较

Tab.3 Algorithm speed comparison with different processing window

算法	5×5×5		7×7×7		9×9×9	
	时间(s)	加速比	时间(s)	加速比	时间(s)	加速比
1	4 175.264 0	1.00	10 412.128 0	1.00	21 622.704 0	1.00
2	674.032 0	6.19	896.528 0	11.61	1 262.992 0	17.12
3	840.896 0	4.97	1 070.048 0	9.73	1 314.944 0	16.44
4	674.528 0	6.19	641.344 0	16.23	627.936 0	34.43
5	134.128 0	31.13	134.624 0	77.34	132.704 0	162.94
6	108.320 0	38.55	114.112 0	91.24	118.224 0	182.90

注:1. 各算法在 P4 2.8 GHz 处理器、1 G 内存 PC 上进行验证;

2. 算法6采用的分段包围盒 ROI 的段长为50层。

实验3采用算法6对409层切片图像进行处理,邻域窗口尺寸分别取 $5 \times 5 \times 5$ 、 $7 \times 7 \times 7$ 、 $9 \times 9 \times 9$,比较分段包围盒 ROI 的段长参数对算法速度的影响,处理结果如图5所示。从图中可见,合理地选取段长可以进一步提高算法速度,对本文的实验对象,当段长约为50层时,处理时间最短。最佳段长与实体形状相关,当截面变化速度快,段长应取得较小;反之,段长应取得较大。

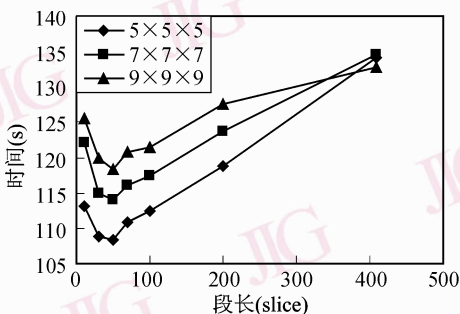


图5 段长对算法速度的影响

Fig.5 Influence of section length on algorithm speed

4 结论

针对工业 CT 切片图像序列,研究了亚体素表面检测算法的可分滤波器递归加速算法,将时间复杂度从 $O(\tilde{N}^2 \cdot \tilde{m}^3)$ 降低到 $O(\tilde{N}^2)$ 。采用增量算法解决了可分滤波器算法从2维推广到3维过程中产生的存储空间问题。将以上算法与感兴趣区域加速策略相结合,采用分段包围盒 ROI 缩减待处理数据量,从而快速有效地实现了产品3维表面点云的精确提取,对工业 CT 高精度实时检测具有现实意义。

参考文献 (References)

- Lyvers E P, Mitchell O R, Akey M L, *et al.* Subpixel measurements using a moment-based edge operator [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1989, **11** (12): 1293-1309.
- Haralick R M. Digital step edges from zero crossing of second directional derivatives [J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 1984, **6**(1): 58-68.
- Ghosal S, Mehrotra R. Orthogonal moment operator for subpixel edge detection [J]. Pattern Recognition, 1993, **26**(2): 295-306.
- He Zhong-hai, Wang Bao-guang, Liao Yi-bai, *et al.* Subpixel algorithm using a curve fitting method [J]. Chinese Journal of Scientific Instrument, 2003, **24**(2): 195-197. [贺忠海, 王宝光, 廖怡白等. 利用曲线拟合法的亚像素提取算法 [J]. 仪器仪表学报, 2003, **24**(2): 195-197.]
- Qu Ying-dong, Cui Cheng-song, Chen Shan-ben, *et al.* A fast subpixel edge measurement method based on Sobel-Zernike moment operator [J]. Opto-electronic Engineering, 2003, **30**(5): 59-61. [曲迎东, 崔成松, 陈善本等. 利用 Sobel-Zernike 矩算子的快速亚像素边缘检测方法 [J]. 光电工程, 2003, **30**(5): 59-61.]
- Luo Li-min. A moment-based three-dimensional edge operator [J]. Pattern Recognition and Artificial Intelligence, 1991, **4**(1): 36-42. [罗立民. 一个基于矩的三维边缘检测算子 [J]. 模式识别与人工智能, 1991, **4**(1): 36-42.]
- Wu X G, Wee W G. Surface detection with subvoxel accuracy using 3D directional derivatives [A]. In: Proceedings of SPIE Nondestructive Evaluation of Aging Aircraft, Airports, and Aerospace Hardware III 1999 [C], Newport Beach, California, USA, 1999: 319-329.
- Ji Q, Haralick R M. Efficient facet edge detection and quantitative performance evaluation [J]. Pattern Recognition, 2002, **35**(3): 689-700.
- Haralick R M, Watson L. A facet model for image data [J]. Computer Graphics and Image Processing, 1981, **15**(3): 113-129.