

基于 GPU 的工业 CT 三维图像 任意断面剖切算法

杨 涛, 赵 星, 张 朋

(首都师范大学 数学科学学院 检测成像实验室, 北京 100048)

摘 要:研究工业 CT 三维图像的内部信息是无损检测的重要途径。为了实时观察三维图像的任意断面, 获取数据信息, 提出一种基于 GPU 的体绘制任意断面剖切算法。该算法基于图形处理器(GPU)单指令多数据流(SIMD)计算方式, 通过并行计算内积的方法确定体素与切面的位置关系。当体素位于切面外侧时, 则将体素设置为完全透明, 使其对显示图像没有贡献, 实现了任意断面剖切。还可以实现多断面剖切和结合传输函数的剖切。试验证明, 该算法达到交互速度, 能够应用于工业无损检测过程中。

关键词:工业 CT; 无损检测; 直接体绘制; 任意断面剖切; 图形处理器

中图分类号: TG115. 28; TP391

文献标志码: A

文章编号: 1000-6656(2010)01-0014-05

GPU-based Arbitrary Clipping Rendering Algorithm for 3D Images of the Industrial Computerized Tomography

YANG Tao, ZHAO Xing, ZHANG Peng

(Laboratory of Computerized Tomography, School of Mathematical Sciences, Capital Normal University, Beijing 100048, China)

Abstract: In the process of non-destructive testing, it's an important approach to explore information from the 3D images of the industrial computerized tomography(ICT). In order to observing the arbitrary clipping and getting information of the 3D images, a GPU-based arbitrary clipping rendering algorithm is developed. By calculating dot metrix between the position of voxel and the parameter of clipping plane with the single instruction multiple data (SIMD) instruction of the graphics processing unit(GPU), this algorithm determines the relationship between voxel and the plane, and then modifies the voxel's transparence in parallel. According to the dot metrix value, the transparences of voxels outside the plane are set into zero, which makes them no contributions to the result, thus the arbitrary clipping plane is rendered finally. Our approach can also fulfill multiple plane clipping and clipping combined with transfer function. It's proved that this approach carries the point for interaction and satisfies the application of non-destructive testing process.

Keywords: Industrial computerized tomography (ICT); Non-destructive testing; Direct volume rendering; Arbitrary clipping; Graphics processing unit(GPU)

随着计算机断层成像(CT)技术的发展, 工业 CT(ICT) 已成为工业关键部件无损检测的重要途径。研究人员通过应用重建算法(如滤波反投影法)

得到反映工件构造信息的体数据, 再用直接体绘制方法对体数据进行三维可视化, 从而达到直观观察工件构造的目的。

目前常用的直接体绘制方法适宜可视化工件的外部形态。利用图像分割^[1]可帮助用户观察工件内部构造。但是体数据的分割算法实现复杂, 计算量大, 难以满足实时的要求。为了能够实时观察工件内部感兴趣区域的构造, 一个直接的想法就是对工

收稿日期: 2009-02-16

基金项目: 国家自然科学基金重点资助项目(60532080); 北京市科技新星资助项目(2005B49)

作者简介: 杨 涛(1982—), 男, 博士研究生, 研究方向为三维可视化。

件三维图像进行剖切。直接体绘制的剖切方法是目前的一个重要研究方向。

现阶段主要有四类三维图像剖切算法:基于体纹理的剖切、基于深度信息的剖切、基于构造表示的剖切和 OpenGL 传统剖切等。其中,基于体纹理的剖切又可细分为两种实现方法:通过直接消隐体素实现,称为直接体剖切^[2-3];应用辅助切割纹理间接消隐体素实现,称为间接体剖切^[4-5]。这类算法原理简单,易于实现多次剖切和复杂几何体的剖切。然而该类算法实现时整个体纹理都被重绘,断面绘制速度受体绘制速度的约束。另外直接体剖切方法需要切片重排实现,即要求采样平面与断面平行,具有局限性。而间接体剖切方法需额外占用显存。基于深度信息的剖切依据切割体的深度信息实现^[5-7]。该类算法优点在于:将体素与切面相对位置比较高维问题转化为仅比较深度的一维问题,能够提高绘制效率;同时由于绘制精度达到像素级别,能够绘制高质量的图像。然而在多个几何体和非凸几何体的切割的情况下,由于多切割体和非凸切割体的深度范围难以确定,因此这类算法难以实现。基于构造表示的剖切^[8]先计算外形轮廓,然后应用构造表示法(Constructive Solid Geometry, CSG)实现任意曲面的交互切割。这种算法可实现任意曲面的切割。缺点是实现断面剖切过程复杂,辅助计算量大。OpenGL 传统剖切^[9-10]通过限制视图体(frustum)实现场景的断面绘制。这类算法易于实现场景的剖切。缺点是视图体内的场景都参与剖切,无法实现三维图像的局部剖切。

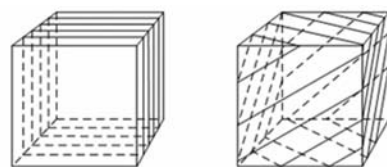
针对断面剖切在工件无损检测中的实时应用问题,笔者提出了一种基于直接体素消隐的任意断面剖切算法。该算法结合直接体绘制方法,充分利用 GPU 的 SIMD 计算方式,并行实现了任意断面的计算与渲染。相对于已有的基于体剖切的算法,该算法具有如下特点:直接剖切体纹理,无需额外的纹理内存消耗;并行计算体素与剖面的位置关系,然后消隐剖面外侧的体素,无需切片重排;结合基于纹理的体绘制方法实现,适合应用 GPU 加速实现,从而提高绘制速度;易于实时实现任意断面剖切、多断面的剖切和结合传输函数的剖切。

1 背景知识

1.1 基于三维纹理的直接体绘制

直接体绘制是体数据重构的一类重要算法。它的优点是生成中间几何图元,直接对体数据成

像,这使得体数据信息得以充分利用。目前常用的直接体绘制算法包括光线投射法、光线追踪法、错切变形法和纹理贴图法等。其中,纹理贴图法^[11-13]先用代理多边形(Proxy Polygon)采样体数据,再对采样数据进行纹理贴图,最后混合纹理贴图实现。该类算法按照实现方式可分为二维纹理贴图法和三维纹理贴图法。二维纹理贴图法应用垂直于数据轴向的代理多边形(图 1(a))进行采样,然后将采样数据作为二维纹理导入显存,要预先将三个垂直于轴向的采样数据同时导入显存,显存消耗大。而三维纹理贴图法则应用垂直于观察方向的代理多边形(图 1(b))进行采样,仅需要将一份数据作为三维纹理导入显存,显存消耗小。由于显卡存储量的限制,笔者应用三维纹理贴图的方法实现体重构。



(a) 垂直于轴向 (b) 垂直于观察方向

图 1 代理多边形

1.2 GPU 硬件加速技术

直接体绘制需要较大存储空间,且涉及大量插值运算和颜色混合操作,绘制速度慢。现阶段的可编程图形硬件(GPU)^[14]具有高存储量、高并行性和管线可编程性等特性。显卡的高存储量使得中小规模体数据可以作为三维纹理一次导入显存进行绘制,无需内存与显存的数据交换。硬件的 SIMD 计算方式具有高并行性,能提高绘制效率。同时,硬件管线的可编程性突破固定管线的限制,易于控制片元属性的计算。这些特性能满足对中小规模体数据实时剖切的要求,因此笔者采用基于 GPU 的可编程管线实现算法。

1.3 传统 OpenGL 裁切的实现原理

在实现过程中,笔者的算法根据向量内积判断体素消隐,这与 OpenGL 传统裁切方法的思路相近。为对比说明笔者的算法,先介绍 OpenGL 传统裁切的实现原理。OpenGL 传统裁切方法基于视坐标下的场景消隐实现,其裁切公式为:

$$(A, B, C, D)M^{-1}(x_e, y_e, z_e, w_e)^T \geq 0 \quad (1)$$

式中 (A, B, C, D) 是物体坐标系下的任意平面方程 $Ax + By + Cz + D = 0$ 的系数向量; M^{-1} 是当前绘制环境的模型视图矩阵的逆矩阵; $(x_e, y_e, z_e, w_e)^T$ 是视坐标系下的任意位置。

传统裁切方法的坐标变化过程如图 2 所示,其中裁切操作在视坐标系下执行。执行裁切操作时,绘制满足式(1)的视坐标位置的场景,而不满足式(1)的场景不参与绘制,最终达到绘制断面的目的。

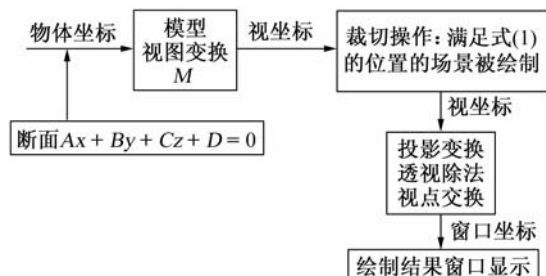


图 2 OpenGL 裁切的坐标变换流程图

2 任意断面剖切算法

应用式(1)实现裁切主要存在两个问题:一是 M^{-1} 作用于视坐标系下所有的位置,这使得场景中的位置都参与裁切,难以满足仅裁切三维图像的要求;二是切面与场景中的位置在不同坐标系定义,导致裁切需要额外的变换操作,实现繁琐。

2.1 算法的数学原理

产生上述问题的主要原因是传统方法无法控制裁切坐标范围并且切面与体素的位置所在的坐标系不统一。为克服上述局限,给出在世界坐标系下的体素通透修正公式,进而提出基于该公式的任意断面剖切算法。其中世界坐标系以体数据包围盒的中心为原点,以数据的三个轴向按照右手系建立。则世界坐标系下的体素通透修正公式为:

$$(A, B, C, D)(x_w, y_w, z_w, w_w)^T \geq 0 \quad (2)$$

式中 $Ax + By + Cz + D = 0$ 是定义在世界坐标系下的任意平面; (A, B, C, D) 是其系数向量; $(x_w, y_w, z_w, w_w)^T$ 是该坐标系下体素的齐次坐标。

该算法在世界坐标系下计算断面剖切,将所在位置不满足式(2)的体素的通透度置为零,即使该体素完全透明,消隐该体素对绘制结果的影响,从而实现断面绘制。具体的坐标变换过程如图 3 所示。

体素的消隐操作如图 4 所示。图 4 是 8 个体素依据式(2)进行消隐的效果,圆球表示体素,黑球表示被消隐的体素。

2.2 算法实现

算法采用独立计算每个体素与切面之间的相对位置,因此适宜并行计算。算法实现充分应用 GPU 的并行计算能力,并结合基于三维纹理的直接体绘制方法,其过程如图 5 所示。图中“纹理坐标”和“切

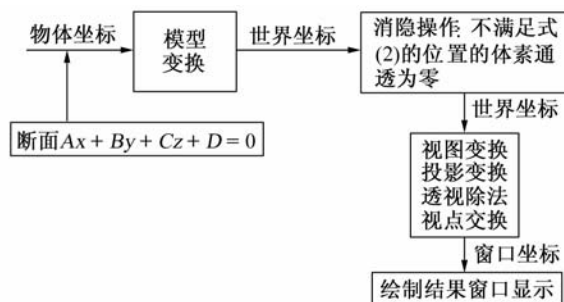


图 3 算法的坐标变换流程图

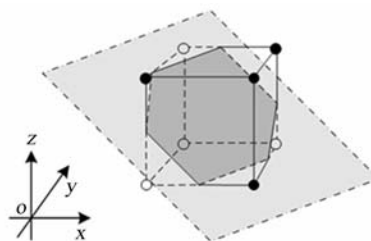


图 4 体素消隐示意图

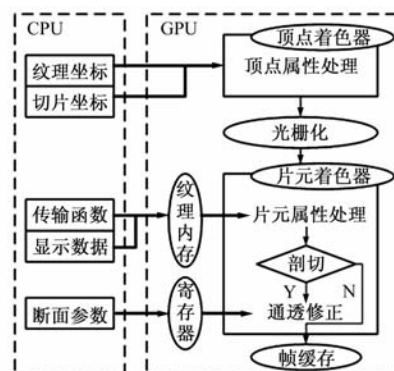


图 5 基于 GPU 的算法实现流程图

片坐标”表示三维纹理贴图所需的坐标,根据体数据的尺寸计算得到。它们被用来实现三维纹理贴图,并被顶点着色器的“顶点属性处理”过程处理。“传输函数”是体绘制所需要的数据光学特征映射函数表,可通过手动调制相应的控件获取,并作为二维纹理导入“纹理内存”。“显示数据”表示要绘制的体数据,作为三维纹理导入“纹理内存”。“断面参数”表示世界坐标系下的任意平面的系数向量,作为参数存入“寄存器”,用于“通透修正”计算。“顶点属性处理”主要计算断面绘制所需的位置坐标及纹理坐标。“光栅化”后通过“片元属性处理”进行纹理采样等操作。“通透修正”依据式(2)修正体素的透明度。绘制结果最终输出到“帧缓存”,并在显示器上显示。任意断面绘制的具体实现过程主要分为以下三个步骤。

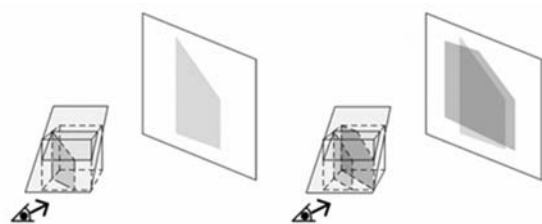
2.2.1 预处理数据

预处理过程应用 CPU 主要完成两项工作:一是通过对体数据进行滤波、采样等操作得到绘制所需的“显示数据”,并计算数据直方图;二是根据数据大小按照 2.1 节介绍的方法建立世界坐标系,计算体包围盒的顶点坐标,然后计算纹理贴图所需要的“纹理坐标”和“切片坐标”。

2.2.2 剖切 CT 三维图像

先将“显示数据”和“传输函数”导入“纹理内存”,再应用 GPU 实现体素消隐,然后进行体素的绘制操作,最终实现三维图像的断面剖切。在实现体素消隐时,首先在顶点着色器下获得访问显示数据所需要的纹理坐标和进行断面绘制所需的体素位置。它们经过“光栅化”后,进入片元着色器。然后根据纹理坐标先获得体素值,这些值通过三线性插值得到,再查找传输函数获得数据的光学特征。接着把光栅化后的体素位置和“寄存器”中的断面系数按照式(2)进行计算,将不满足式(2)的位置对应的体素透明度置零。这就完成体素的通透修正,达到消隐体素的目的。

在体素消隐操作后,需要绘制体素以显示三维图像的任意断面。由于算法实现结合基于三维纹理的体绘制方法,因此经过通透修正后的体素的绘制过程主要包括纹理贴图操作和混合操作。以任意切面剖切为例,图 6(a)表示对消隐操作后的体素进行纹理贴图的情况,图 6(b)表示对纹理贴图进行混合的情况。



(a) 纹理贴图操作 (b) 混合操作

图 6 本算法绘制示意图

2.2.3 交互设置断面参数、调制传输函数

通过断面参数设置控件调整断面方向和断面位置,以改变绘制的断面。通过传输函数控件调整更新传输函数,以便实现结合传输函数的任意断面绘制。完成断面参数和传输函数的设置后,再执行前两步,就实现断面绘制的交互。

2.3 算法分析

由于应用式(2)实现剖切,因此仅需限定参与剖

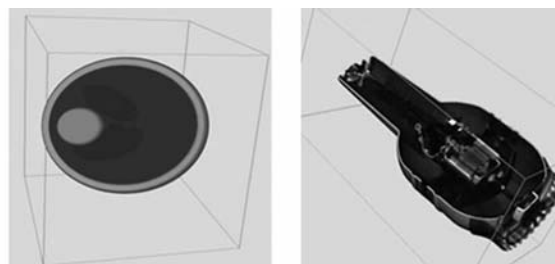
切的坐标范围,就可以实现任意局部的剖切。默认情况下参与剖切的坐标范围是体数据的包围盒在世界坐标系下的范围。由于剖面与体素位置在世界坐标系下定义,因此算法的实现无需额外的变换操作,而且更加简捷、直观。综上分析,本算法能够解决传统裁切存在的问题。

另外,针对中小规模的体数据,该算法适合 GPU 硬件加速实现,能克服传统体剖切方法存在的局限。现阶段的显存容量能够满足直接处理中小规模体数据的要求,因此重绘将不影响剖切的交互。而 GPU 的可编程性和并行计算能力使得该算法能够突破基于切片重排的剖切方法的局限。

3 试验结果与分析

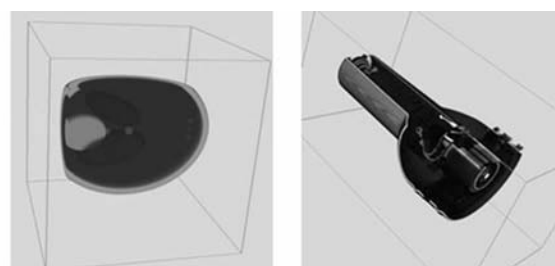
试验利用三维 Shepp-Logan 模型(简称为 3D-SL)的模拟数据和实际的吹风机 CT 重建数据测试该算法。数据的存储类型均为无符号短整型,数据大小分别为 $256 \times 256 \times 256$ 和 $256 \times 256 \times 512$ 。试验的内容包括任意断面剖切、多断面剖切和结合传输函数的剖切,并且给出不同情况的绘制速度。试验运行计算机硬件环境为: Intel Xeon 5120(主频 1.83 GHz,双核)处理器,DDR2 内存(4 GB),NVIDIA Qurdro FX4600 显卡(显存 768 MB,核心频率 500 MHz)。

试验效果如图 7~9 所示。可见,应用本算法能够突显被扫描物体的大部分内部结构。然而,为了



(a) 3D-SL (b) 吹风机

图 7 任意断面剖切



(a) 3D-SL (b) 吹风机

图 8 多断面剖切

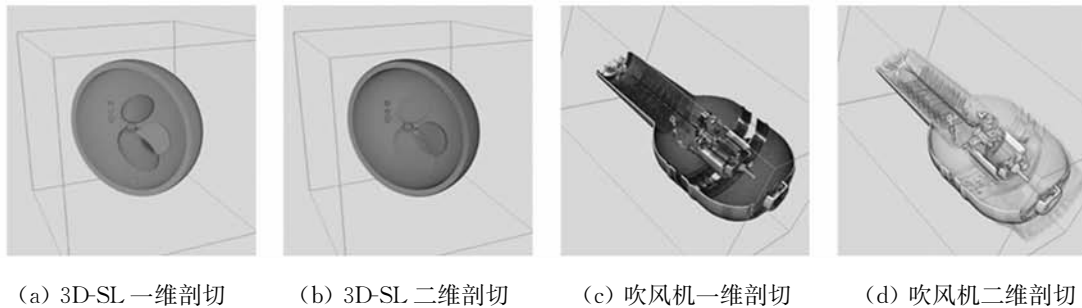


图9 结合一维、二维传输函数的剖切

更加清晰地显示被剖切图像的结构分布,提出结合传输函数剖切三维图像的思路。通过对比绘制效果可见,内部结构信息被进一步突显。图中显示的线框为体数据包围盒。

为了说明算法的交互性能优势,表1列出基于体纹理的几种剖切算法的绘制速度。其中输出窗口大小为 $1257\text{ 像素}\times 803\text{ 像素}$,绘制速度用每秒重绘的帧数描述(fps)。对比试验中断面个数为二并且未开启传输函数,辅助纹理与数据纹理具有相同的维度。

由表1可知,在相同效果断面的绘制情况下,基于GPU的直接体剖切算法相对已有的两种基于体纹理的剖切算法具有性能优势。相对于基于辅助纹理的剖切,该算法实现不需要额外的显存消耗,因此性能优势较大。与基于切片重排的剖切相比虽然占用相同的显存数量,但由于减少了一些比较计算,因此同样具有速度优势。另外基于切片重排的方法纹理采样方向受剖面方向限制,难以同时绘制不同方向断面的切面,因此本文算法优于基于切片重排的剖切算法。

表1 三种体剖切算法的速度

数据集(数据维数)	基于辅助纹理	基于切片重排	直接消隐体素
3D-SL($256\times 256\times 256\times 16$)	31.7	46.7	48.3
吹风机($256\times 256\times 512\times 16$)	17.4	31.5	32.9

4 结语

提出一种基于直接体素消隐的任意断面剖切算法。相对于已有的同类算法,该算法通过在体绘制下直接消隐体素实现,避免额外的显存消耗。同时,由于应用GPU并行计算内积的方法消隐体素,克服了传统直接体剖切对断面方向的限制。由试验结果可知,针对中小规模的体数据,该算法达到实时交

互的绘制速度,其绘制效果有助于对CT三维图像的结构进行研究,能够满足工业CT实时观察三维图像任意断面以辅助无损检测的要求。

另外,还需进一步研究的工作有:①实现该算法需将体数据一次导入显存,限制了参与绘制的数据量。需研究数据分块、空数据块剔除及LOD技术,根据断面参数导入参与显示的非空数据块,提高可绘制的数据量和绘制速度。②传输函数针对全局的体数据进行调制,无法针对局部数据调制,影响感兴趣结构的突显。需设计适用于任意断面剖切的传输函数,使得切面一侧的信息进一步突显。

参考文献:

- [1] 卢艳平,王珏,覃仁超.一种剥皮算法在工业CT图像分割中的应用[J].无损检测,2005,27(5):235-237.
- [2] Schulze J P, Niemeier R, Lang U. The perspective shear-warp algorithm in a virtual environment[C] // Proceedings of Visualization'01. Amsterdam, Netherlands: Elsevier Science Publishers B V, 2001: 207-213.
- [3] He Hui Guang, Tian Jie, Lin Yao, et al. Virtual cutting and its application in medical imaging[J]. Chinese Journal of Medical Physics, 2002, 19(2): 72-78.
- [4] 杨慧,鲍旭东.基于硬件加速体绘制的切割方法[J].生物医学工程研究,2007,26(2):183-190.
- [5] 孙刚,杨杰.基于硬件纹理体重建的切割算法[J].上海交通大学学报,2006,40(5):783-786.
- [6] Weiskopf D, Engel K, Ertl T. Volume Clipping via Per - Fragment Operations in Texture-Based Volume Visualization[C] // Proceedings of Visualization'02. Washington, DC, USA: IEEE Computer Society, 2002: 93-100.
- [7] Weiskopf D, Engel K, Ertl T. Interactive clipping techniques for texture-based volume visualization and volume shading[J]. IEEE Transactions on Visualization and Computer Graphics, 2003, 9(3): 298-312.

(下转第21页)

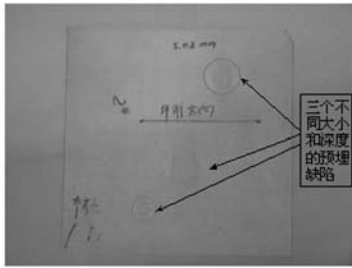


图4 复合材料样品图

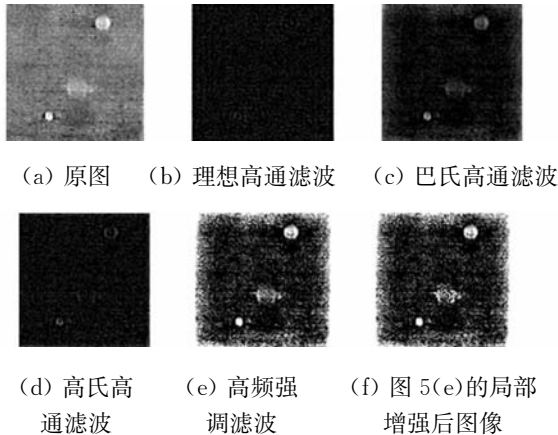


图5 高频强调滤波法处理的增强效果图

高斯高通滤波图滤波效果最差,缺陷被湮没在背景和噪声中。

为检验红外探伤图像的增强效果,采用峰值信噪比(PSNR)^[5]进行评价,其数值表达式为:

$$\text{PSNR} = 10 \lg \left\{ \frac{(L-1)^2}{\sum_{i=0}^{M-1} \sum_{j=0}^{N-1} [f_N(x,y) - f(x,y)]^2} \right\} \quad (5)$$

式中 $f(x,y)$ 为红外探伤原图; $f_N(x,y)$ 为含噪声图像; L 为灰度级数; M, N 为图像的行和列。上述四

表1 各种高通滤波器的峰值信噪比

名称	峰值信噪比(PSNR)
理想高通滤波	-72.883 2
Butterworth 高通滤波	-68.661 6
高斯高通滤波	-69.451 0
高频强调滤波	-66.734 9
局部增强后图像	-66.688 9

种高频滤波的 PSNR 值如表 1 所示。

通过计算四种滤波器的峰值信噪比,局部增强后的高频强调滤波图像的 PSNR 最大,理想高通滤波的 PSNR 最小,与定性分析基本吻合。

4 结论

采用局部增强后的高频强调滤波法对红外探伤图像进行了处理。在突出高频分量的同时,能保持经过高通滤波后丢失的低频分量,对比度得到明显增强,峰值信噪比变大,图像缺陷判读变得更加容易。

参考文献:

- [1] 王爱玲,叶明生,邓秋香. 图像处理技术与应用[M]. 北京:电子工业出版社,2007:343-346.
- [2] 刘加伶,余成波. 基于人体手指静脉特征提取算法的研究[J]. 计算机科学,2008,35(8):218-230.
- [3] 胡广书. 数字信号处理——理论、算法与实现[M]. 北京:清华大学出版社,2003:272-277.
- [4] 郭永刚. 红外热波图像的畸变校正与拼接技术研究[D]. 北京:首都师范大学,2006.
- [5] Damara-Venkata N, et al. Image quality assessment based on a degradation model[J]. IEEE Trans on Image Process,2004,9(4):636-650.
- [6] Meißner M, Hoffmann U, Straßer W. Enabling Classification and Shading for 3D Texture-based Volume Rendering Using OpenGL and Extensions[C] // Proceedings of Visualization'99. New York, NY, USA: ACM Press, 1999:207-214.
- [7] Rezk-Salama C, Engel K, Bauer M, et al. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-texturing and Multi-stage Rasterization[C] // Proceedings of Euro graphics/SIGGRAPH Workshop on Graphics Hardware. New York, NY, USA: ACM Press, 2000:109-118.
- [8] 李衷怡,张浩,李利军,等. 体绘制的任意曲面切割方法[J]. 计算机工程与应用,2005,41(33):60-63.
- [9] Woo M, Neider J, Davis T, et al. OpenGL Programming Guide Third edition[M]. Massachusetts: Addison Wesley,1999:118-119.
- [10] Gaurav K, Bijaya B K. Visualization of 3D scientific datasets based on interactive clipping[R]. WSCG 2005 Short Papers proceedings, Plzen, Czech Republic; UNION Agency-Science Press,2005:33-33.
- [11] Westermann R, Ertl T. Efficiently using graphics hardware in volume rendering applications[C] // Proceedings of the ACM SIGGRAPH'98. New York, NY, USA: ACM Press,1998:169-178.
- [12] Meißner M, Hoffmann U, Straßer W. Enabling Classification and Shading for 3D Texture-based Volume Rendering Using OpenGL and Extensions[C] // Proceedings of Visualization'99. New York, NY, USA: ACM Press, 1999:207-214.
- [13] Rezk-Salama C, Engel K, Bauer M, et al. Interactive Volume Rendering on Standard PC Graphics Hardware Using Multi-texturing and Multi-stage Rasterization[C] // Proceedings of Euro graphics/SIGGRAPH Workshop on Graphics Hardware. New York, NY, USA: ACM Press, 2000:109-118.
- [14] 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.