

基于海量存储的三维地理信息系统应用与研究

宋娟张宏

(哈尔滨理工大学 计算机科学与技术学院 黑龙江 哈尔滨 150080)

摘要: 针对海量地形数据存储空间大的瓶颈,提出了基于视域改进的 LRU 算法。为了实时提供基础的地理信息以及现场分布情况,支持海量地形数据存取,算法利用基础数据的大数据量、多数据源、多分辨率等特点,采用分页数据库组织、四叉树结构的 LOD 瓦片组织算法,构建金字塔三维模型;同时,采用动态调度、多缓冲多线程并行处理的方法,实现了多重纹理快速映射地形,达到了海量地形数据快速存取的效果。算法中海量数据存储方便,数据检索查询快速,配置要求低。系统实时提供基础地理信息、现场分布情况,支持海量数据和复杂图形符号,在使用过程中,使用者可直观地观察地图信息,根据实时传递的场景地形地貌进行分析,速度快可靠性高,提高了使用者的工作效率。

关键词: 多线程;缓冲区机制;海量地形数据;金字塔模型;纹理映射

中图分类号: TP302

文献标识码: A

文章编号: 1673-629X(2018)05-0144-05

doi: 10.3969/j.issn.1673-629X.2018.05.033

Application and Research of Three Geographic Information System Based on Mass Storage

SONG Juan ZHANG Hong

(School of Computer Science and Technology, Harbin University of Science and Technology, Harbin 150080, China)

Abstract: Aiming at the bottleneck of massive terrain data storage space, we propose a LRU algorithm based on improved horizon. In order to provide basic geographic information and field distribution in real time and support the accessing of massive terrain data based on the basic data of large volume, multi-source and multi-resolution, the algorithm uses the LOD tile organization algorithm with the paging database organization and four-tree structure to construct the pyramid 3D model. At the same time, it utilizes the method of dynamic scheduling and multi-buffering multi-thread parallel processing to achieve a multi-texture rapid mapping terrain, which finishes the fast accessing of the massive terrain data. In the algorithm, the mass data storage is convenient, the data retrieval query is quick and the configuration requirement is low. The system provides the real-time basic geographic information and field distribution and supports the massive data and complex graphical symbols. In the course, the user can visually observe the map information and analyze it according to real-time transmission of terrain topography, with fast speed and high reliability, rising the user's work efficiency.

Key words: multi-threading; buffer mechanism; massive terrain data; pyramid model; texture mapping

0 引言

近年来, GIS 软件的 3D 功能不断加强。国外 3D GIS 软件典型代表包括 ESRI 公司的 ArcView 的 3D Analyst 和 GEONOVA 公司的 DILAS。前者支持多维矢量转换和全球多分辨率数据的 3D 无缝可视化,后者支持 LOD 的 3D 建模、关系数据库存储与管理。由于显存和系统内存容量有限,实际应用中大规模地形数据采取最近最少利用(least recent use, LRU)原则的分页方式。针对大规模地形与纹理数据动态可视化的瓶颈,文中提出了多重纹理映射的大规模地形绘制方

法,实现了规则网格区域分割的分层(LOD)分块和四队列分页数据库调度、基于视域的 LRU 算法内存释放、多线程并行处理的场景优化策略。

1 四叉树结构的 LOD 瓦片组织算法

海量地形 3D 可视化,采用高精度的 DEM 作为基础数据,往往能达到几百万个三角形面,渲染每一帧不可能一次性显示,系统帧率无法达到实时渲染的要求。为解决大规模数据和系统显示速率的瓶颈,对海量地形和纹理数据采用基于四叉树结构的 LOD 简化^[1-3],

收稿日期: 2017-04-19

修回日期: 2017-08-29

网络出版时间: 2017-12-05

基金项目: 黑龙江省教育科学技术研究项目(12541177)

作者简介: 宋娟(1994-)女,研究方向为计算机应用;张宏教授,研究方向为计算机测控技术。

网络出版地址: <http://kns.cnki.net/kcms/detail/61.1450.TP.20171205.1432.084.html>

使纹理映射高效集成到地形 3D 可视化中, 遍历简单, 减少选中地形子块的时间。

1.1 四叉树分层分块的 LOD 算法

该系统预处理方法采用地形和纹理多分辨率建模 构建金字塔模型^[4]; 按照四叉树算法构建 LOD 地形和纹理树^[5]; 对纹理片与地形片进行有效压缩; 创建外存的金字塔模型数据库。多分辨率金字塔用网格表示统一区域, 相邻层精度为 1 : 4。按正常的视觉现象 透视投影规律: 视点距地面较远时, 绘制较少的三角形; 较近时, 绘制较多的三角形。采用四叉树算法将原始地形划分为多网格模型, 再分别进行递归分割, 直到最后一级分块为最高分辨率。

1.1.1 分块四叉树

采用等间隔划分方法对大规模数据金字塔模型进行分层分块处理 根据由上而下的方法进行简化。以地形图为基础 进一步进行规则格网划分。例如一幅 1 : 2 000 图幅的大小为 1 000×800, 根据需要把部分单图幅按 2×2 等分, 即 500×400, 部分按照 4×4 等分, 即 250×200。以左上角为原点, 四叉树场景结构划分如图 1 所示。

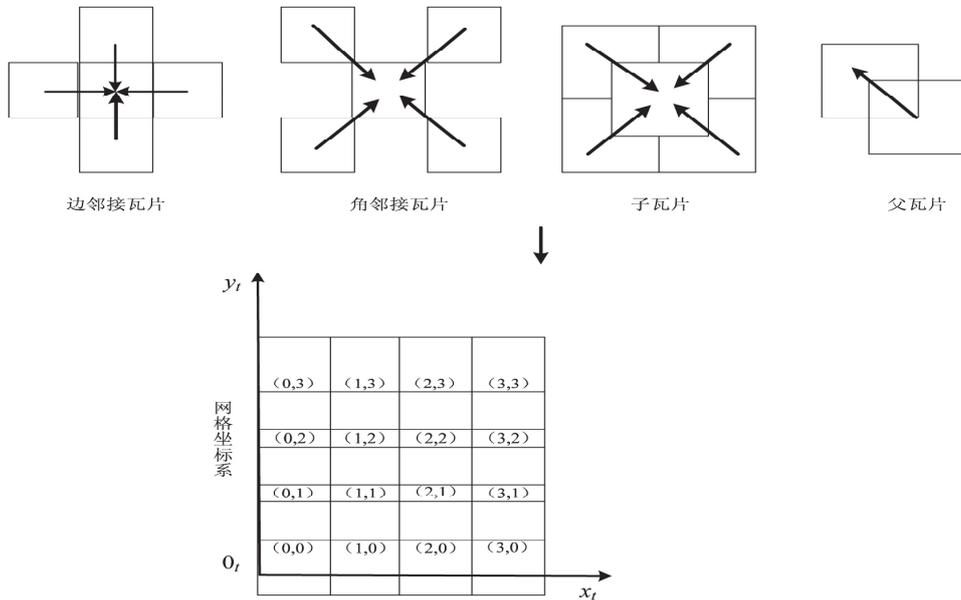


图 1 四叉树结构划分

模型每节点有两个 LOD 级别: 本身所在的粗糙网格模型级别和四等分后的子块级别。分层分块策略, 一次读写访问一节点, 且动态调度中只访问需要的节点。很大程度上缩短了数据访问时间又能表示多分辨率数据; 与四叉树结构相结合, 具备了快速数据索引的功能。

四叉树层与瓦片的层对应, 节点编号是层号和网格坐标, 拓扑关系如图 2 所示。上下层父子关系、同一层邻接关系。邻接包括角邻接、边邻接, 上下层包括向上继承、向下分解。

图 2 瓦片拓扑关系和网格坐标系

1.1.2 块结构与 LOD 建模编程实现

采用 hash map 的数据结构, 索引作为键, 指针内容作为值。采用块结构保存, 如下:

```

struct Chunk
{
    int chunkIndex; //块索引
    int swIndex, seIndex, neIndex, nwIndex; //4 个子节点索引
    int lod_level; //节点层
    int xIndex, yIndex; //节点块所在层的位置
    int min_h, max_h; //块的最小和最大高程
    int mesh_pos; //三角网数据位置
}

```

} 利用递归思想实现四叉树 LOD 模型建立的过程如下:

```

void UpdateQuadTr( int nXCenter, int nZCenter, int nSize, int
lod_level, int xIndex, int yIndex)
{
    .....
    //判断地形子节点是否需要分割 若是 则执行以下代码
    m_bQuadMat [ chunkIndex ] [ nXCenter + nZCenter * m_
nMapSize ] = true;
    UpdateQuadTr( nXCenter-nSize/2, nZCenter-nSize/2, nSize/

```

```

2 lod_level+1 xIndex yIndex); //分割西南节点
    UpdateQuadTr( nXCenter+nSize/2 ,nZCenter-nSize/2 ,nSize/
2 lod_level+1 xIndex yIndex); //分割东南节点
    UpdateQuadTr( nXCenter+nSize/2 ,nZCenter+nSize/2 ,nSize/
2 lod_level+1 xIndex yIndex); //分割东北节点
    UpdateQuadTr( nXCenter-nSize/2 ,nZCenter+nSize/2 ,nSize/
2 lod_level+1 xIndex yIndex); //分割西北节点
}

```

1.2 地形数据库管理策略

在纹理叠加到地形的过程中,根据地形和纹理数据的大小,根据不同层不同块分辨率大小及粗糙程度,对纹理和地形不断进行 LOD 处理、计算并叠加。不同层由不同分辨率块与纹理片组成。把同时打包纹理的地形块保存为三维模型写入磁盘,表示多段数据块的组合体,以常用的“.ive”格式,把场景树保存为二进制文件,按不同层次数据块存储。

每数据层有一特定的标识名。命名方式如下:

FILENAME = 地名 + 数据类型 + 数据层 + 行文件夹 + 数据文件

地块模型命名为 name_root/subtile(数据类型)_L(层)_r(行号)_c(列号).ive,如 beijing_subtile_L3_X0_Y1.ive。由于系统中每个文件夹容纳的文件数量有限,通过对层做分级,可避免文件夹文件过多。分级后,在外存中,3D 地形纹理库存储管理图如图 3 所示。

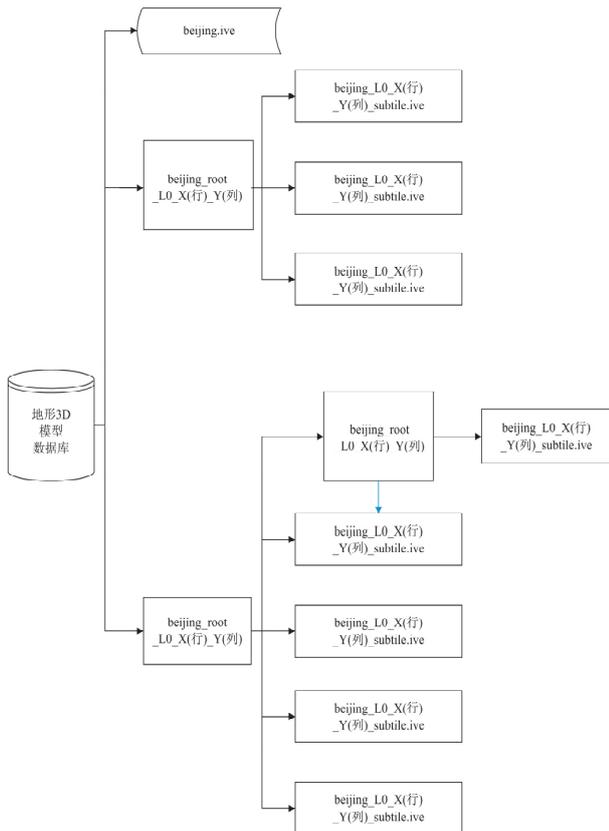


图 3 地形纹理数据库存储管理图

构建 3D 地形库,有效组织特定区域相关的数

据^[6]。通过地理坐标关联视域范围的数据,统一分配空间位置索引,快速调度 3D 库中的地形纹理数据,提高纹理地形节点的访问效率和场景裁剪效率,达到整个大规模地形的无缝渲染漫游。

2 动态调度策略算法

2.1 动态调度内存释放方法

三维场景实时浏览时,根据视点的位置,计算视域内块的索引,根据空间索引快速计算与上一步建立的四叉树各块中心点的距离,并根据事先指定的空间阈值判断是否需要加载或卸载数据,并根据视点的变化,不断调整内存中模型的细节层次(LOD)级别。

2.1.1 基于视域的 LRU 算法

LRU 页面置换算法在管理内存时移出短时间内不用的数据块,提高了空间利用率。采取虚拟页存储管理,将所需空间划分为多个页面,内存中存放当前所需页面,其余页面放入外存。基于视点的加载过程如图 4 所示。

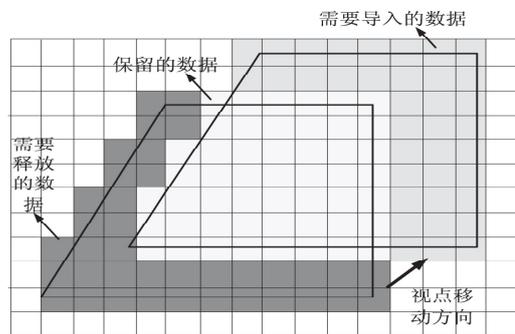


图 4 基于视点的数据加载卸载

文中在 LRU 算法^[7]的基础上,采用基于可视范围的数据内存释放算法,内存达到上限时,计算下一帧视域外的地形块,根据优先级高低将地形数据从内存缓冲区卸载。

2.1.2 动态调度实现

大规模地形 3D 场景浏览,有效采用数据库分页技术。分页调度划分空间页面,页面为地形基本调度单位,场景浏览时动态加载。整个地形按照四叉树构建原则进行划分,自顶而下、逐级建立各级 R+树空间索引。空间索引方法是为了加速场景遍历,最大限度地缩短数据加载与卸载的响应时间。

根据 LOD 技术,利用视角高度与瓦片尺寸的比,判定当前显示的瓦片是否需要进一步分解。为了利用 GPU 图形处理能力,对每一地形瓦片,采用 Chunlod 网格(Mesh)模型算法,利用后台线程加载网格数据,GPU 对其进行批处理运算,调整地形几何网格块的 LOD 层。

地形动态调度和可视化流程如图 5 所示。

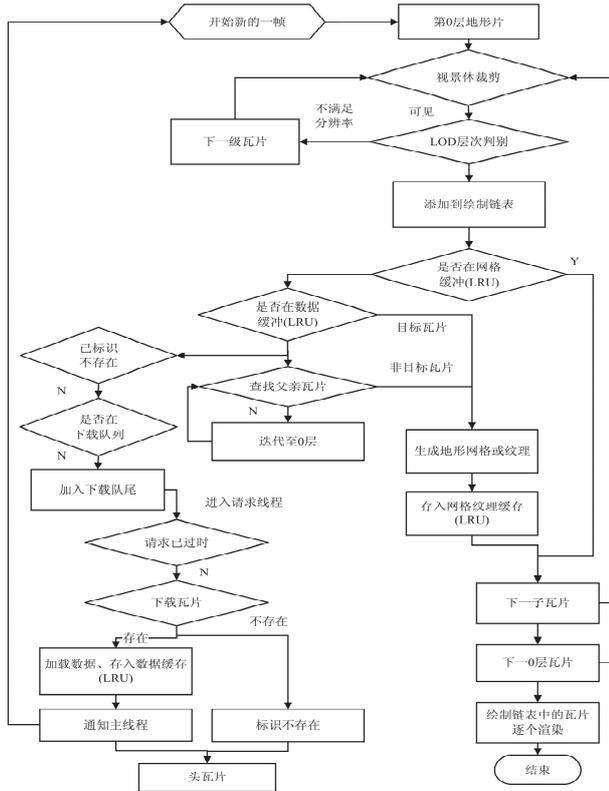


图 5 地形数据动态调度和可视化流程

3D 场景漫游中,计算和渲染视域内的网格数据^[8-10]在很大程度上降低了运行时间复杂度,高效地实现了大规模地形数据的构建与绘制。

2.2 四队列多线程调度机制

通过上述调度策略,基本实现了大规模地形 3D 场景的实时浏览与交互。为进一步提高场景渲染效率,采用缓存区四队列机制、多线程加载、区域索引的场景裁切和纹理共享。

2.2.1 缓冲区机制

缓冲区包括内存和外存缓存。内存缓存中,已加载的数据直接用于场景绘制,实现高速缓存。外存缓存用于预存储数据。高速缓存大小设有上限,控制瓦片数。内外存置换采用所述的 LRU 算法。漫游常在部分场景附近进行,刚渲染的数据不应立即卸载。当后台线程需要获得渲染数据时,先在缓冲区中查找,如未找到则在外存中继续搜索。

2.2.2 四队列调度

数据动态调度机制^[11-12]:显示当前可视范围时,预判断下一步可能载入的数据;判断在最近时间内不可能出现在可视范围内的数据;正确地进行数据加载、卸载、预编译和合并处理。保证场景浏览的稳定性和畅通性。数据调度流程如图 6 所示。

2.2.3 多线程处理

每一层单独设置线程负责不同 LOD 数据的加载卸载,可提高大规模场景反复加载卸载的性能。原理

如下:

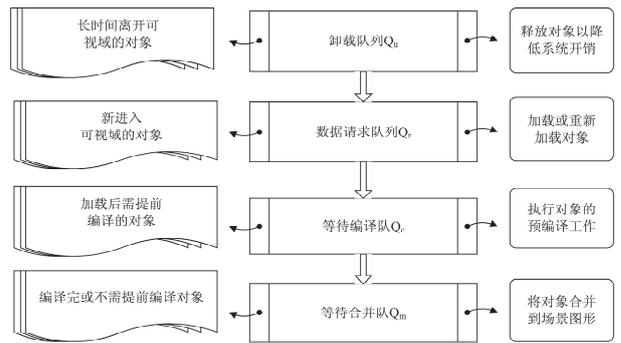


图 6 数据调度处理流程

主线程:地形数据获取,LOD 简化,视域剔除,实现实时绘制和渲染。

数据预调度线程:根据当前视点计算可见域和预可见域,若需要地形数据块未在内存中,载入队列加入地形快索引,从外存库动态读取地形块,且剔除不可见节点以避免资源浪费,同时进行预可见数据调度。

事件处理线程:接收事件,计算相机参数,设置相机视图矩阵、漫游器、视口等。

3 3D GIS 平台应用

地形/影像三维模型处理工具,实现多重纹理映射地形构建金字塔模型,按层在本地存放组织,实现多重纹理映射。为减小渲染负担,将多个原始影像分别预处理成不同分辨率纹理,使用 OSG 中 Mipmap 纹理映射技术^[13-15]。2D 纹理宽度高度是 2 的幂次方,单位为纹素(texel)。一般的,较小纹理图像分辨率是下一层 4 个纹理片(单元)的平均值。

如图 7 所示,纹理使用多幅分辨率不同的遥感影像(10 m 分辨率的普通遥感影像、1 m 分辨率的城区高分辨率遥感影像)叠加拼接而成。场景中明显地展现了不同分辨率纹理的效果,很好地实现了多幅遥感影像的无缝拼接,可在尽量减少大规模数据且保证较高的加载渲染效率的条件下,使场景研究主题更突出。

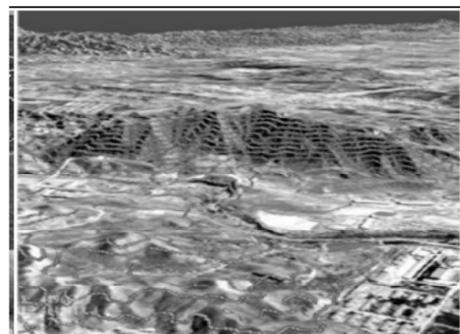


图 7 三维地理信息系统

4 结束语

针对 Mipmap 纹理的大规模三维地形可视化技术

进行了研究,并应用于 3D GIS 平台。基于四叉树分层分块的 LOD 算法,建立多分辨率金字塔模型并进行有效的组织管理;引入四叉树空间索引的动态调度机制,对场景中地形数据进行纹理映射,并进行动态调度可视化;通过场景裁剪和纹理共享,实现地形场景的快速优化。加快了存取速度,降低了裁剪计算的复杂度,实现了高效渲染。该课题开发的 3D GIS 平台为一个基础应用架构,并未充分利用 2D 矢量数据的拓扑关系,网络分析、邻域分析、缓冲区分析等空间分析应用功能需进一步完善。

参考文献:

- [1] 岳利群,夏青,柳佳佳,等.全球海量地形数据组织管理方法的研究[J].测绘科学,2009,34(3):121-123.
- [2] 戴晨光,张永生,邓雪清.一种用于实时可视化的海量地形数据组织与管理方法[J].系统仿真学报,2005,17(2):406-409.
- [3] 王璐锦,唐泽圣,唐龙.基于三角形二叉树的地表模型动态简化算法[J].清华大学学报:自然科学版,2002,42(1):92-95.
- [4] 张恒国,王合龙,吴清香.基于 ROAM 算法的数字地面模型研究与实现[J].信息技术,2008,32(1):95-97.
- [5] 杜莹,武玉国,王晓明,等.全球多分辨率虚拟地形环境的金字塔模型研究[J].系统仿真学报,2006,18(4):955-958.
- [6] MINOUX C. Enabling virtual-globe browsing on memory-constrained platforms visualization and exploration of geospatial data [R]. [s.l.]: [s.n.], 2007.
- [7] SUN M, LYU G L, LEI C. Large-scale vector data displaying for interactive manipulation in 3D landscape map [R]. [s.l.]: [s.n.], 2008.
- [8] BLOW J. Terrain rendering at high levels of detail [C]// Game developers conference. [s.l.]: [s.n.], 2000: 119-124.
- [9] DAI C G, ZHANG Y S, YANG J Y. Rendering 3D vector data using the theory of stencil shadow volume [R]. [s.l.]: [s.n.], 2008.
- [10] 申闫春,王锐,翟春丽.海量地形的 GIS 特征数据实时渲染算法研究[J].计算机仿真,2011,28(10):223-227.
- [11] 马柳青,宋关福,郭会,等.一种海量地形影像数据的快速漫游算法[J].地球信息科学学报,2009,11(5):604-609.
- [12] 宋敏,申闫春.基于空间数据库 VRGIS 的研究与应用[J].测绘与空间地理信息,2012,35(8):47-52.
- [13] BHATFACHARJEE S, PATIDAR S, NARAYANAN P J. Real-time rendering and manipulation of large terrains [C]// Proceedings of 6th Indian conference on computer vision, graphics and image processing. Washington DC, USA: IEEE Computer Society, 2008: 551-559.
- [14] 黄超超,凌永顺,吕相银.地形纹理映射方法研究[J].计算机仿真,2005,22(1):209-212.
- [15] 姜翰青,王博胜,章国锋,等.面向复杂三维场景的高质量纹理映射[J].计算机学报,2015,38(12):2349-2360.

(上接第 143 页)

参考文献:

- [1] 周静,杨永明,何为.心音信号的分析及其特征提取方法的研究[J].中国生物医学工程学报,2005,24(6):685-689.
- [2] 成谢锋,马勇,刘陈,等.心音身份识别技术的研究[J].中国科学:信息科学,2012,42(2):237-251.
- [3] 赵治栋,赵知劲,张嵩,等.心音自动分段算法研究[J].航天医学与医学工程,2004,17(6):452-456.
- [4] 屠志海.心音信号的分析方法研究[D].上海:华东师范大学,2011.
- [5] ZHOU J, HE W, DAN C, et al. Feature extraction and recognition of heart sound [C]// Automation congress. [s.l.]: [s.n.], 2008.
- [6] 全海燕,王威廉.基于小波多分辨分析的第一第二心音提取[J].北京生物医学工程,2004,23(1):64-66.
- [7] LIANG H, SAKARI L, IIRO H. A heart sound segmentation algorithm using wavelet decomposition and reconstruction [C]// Proceedings of the international conference of engineering in medicine and biology society. [s.l.]: [s.n.], 1997: 1630-1633.
- [8] GUO X, CHEN J, XIAO S. Heart sound recognition algorithm based on mathematical morphology [J]. Journal of Biomedical Engineering, 2004, 21(5): 832-835.
- [9] ZHAO Z, ZHAO Z, CHEN Y. Time-frequency analysis of heart sound based on HHT [Hilbert-Huang transform] [C]// International conference on communications, circuits and systems. [s.l.]: [s.n.], 2005: 929.
- [10] 谭江平,何为,张占龙.心音信号采集与分析系统[J].重庆工商大学学报:自然科学版,2004,21(5):465-467.
- [11] 欧阳春娟,杨群生,欧阳迎春.基于小波变换的自适应模糊阈值去噪算法[J].计算机工程与应用,2006,42(5):82-84.
- [12] 魏文畅,杨俊杰,蔡建立.基于小波变换的半软阈值参数算法研究[J].计算机工程与应用,2009,45(1):73-76.
- [13] 刘刚,屈梁生.自适应阈值选择和小波消噪方法研究[J].信号处理,2002,18(6):509-512.
- [14] 刘智.改进的正交小波变换阈值去噪算法研究[J].计算机仿真,2011,28(8):272-275.
- [15] VARGHEES V N, RAMACHANDRAN K I. A novel heart sound activity detection framework for automated heart sound analysis [J]. Biomedical Signal Processing and Control, 2014, 13: 174-188.