Interactive 3D Visualization, Denoising, and Annotation Suite for Medical Data

Keyu Lu

Advisor: Adam Mally

University of Pennsylvania

ABSTRACT

Medical datasets, especially 3D ones like MRI or CT scans, hold a treasure of information, but their interpretation often relies on advanced tools for accurate diagnosis. This project aims to develop a sophisticated GPU-accelerated tool enabling medical professionals to explore, denoise, and annotate 3D medical datasets interactively. By integrating a customizable "degree of denoise" slider, users can adjust the level of noise reduction to their preference, ensuring optimal clarity. With the enhanced image clarity and real-time interactive features, the tool is anticipated to facilitate faster and more precise medical assessments.

Project Blog: https://keyulureels.com

Git Repo: https://github.com/uluyek/senior-capstone-project

1. INTRODUCTION

Problem Statement. The complexity and density of information in medical datasets demand efficient tools for exploration and analysis. However, noise and lack of interactive capabilities can hamper the interpretability of these images.

Motivation. With medical imaging becoming a linchpin in diagnostics, a tool that enhances the clarity and interactiveness of these images can greatly benefit the medical community by facilitating better insights and fostering academic discussions.

Proposed Solution. I propose to develop a VTK-based tool that will enable real-time visualization, denoising, and annotation of 3D medical datasets.

Contributions.

This project makes the following contributions:
- VTK-based real-time rendering of medical datasets.
- Interactive exploration features like zoom, pan, and rotation.
- Advanced denoising algorithm integration for clearer images with a customizable "degree of denoise" slider
- User-friendly annotation suite for medical professionals.

1.1 Design Goals

Target Audience: Medical professionals, students, and researchers.

Benefits: Improved understanding of medical datasets, efficient analysis, and collaborative discussions.

1.2 Projects Proposed Features and Functionality

1. Real-time 3D visualization of medical datasets.
2. VTK-based denoising algorithms with flexible degrees of denoise.
3. Interactive annotation suite.

2. RELATED WORK

Recent developments have seen several tools that focus on medical image visualization. Softwares like OsiriX and 3D Slicer offer advanced visualization features. However, real-time interactivity and VTK-based denoising are areas less explored. My project aims to bridge these gaps by providing a comprehensive solution.

2.1 Bibliography

Shi, L., Wen, L., Zhang, H., Xie, Y., & Wang, D. (2012, September 1). *A survey of GPU-based medical image computing techniques.* PubMed; National Institutes of Health. https://doi.org/10.3978/j.issn.2223-4292.2012.08.02

Yang, H., & Gu, L. (2005, January 1). *GPU-based Volume Rendering for Medical Image Visualization*. https://doi.org/10.1109/iembs.2005.1615635

Interactive GPU-Based Volume Rendering for Medical Image. (2009, October 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/5304991

Accelerating Medical Image Processing with NVIDIA DALI | NVIDIA Technical Blog. (2022, August 21). NVIDIA Technical Blog. https://developer.nvidia.com/blog/accelerating-medical-image-processing-with-dali/

Medical Image Denoising Using Convolutional Denoising Autoencoders. (2016, December 1). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/7836672

Sagheer, S. V. M., & George, S. N. (2020, August 1). *A review on medical image denoising algorithms*. Biomedical Signal Processing and Control; Elsevier BV. https://doi.org/10.1016/j.bspc.2020.102036

3. PROJECT PROPOSAL

The project aims to develop an interactive 3D visualization tool tailored for medical datasets, specifically focusing on DICOM data from modalities like CT and MRI scans.

3.1 Anticipated Approach

1. **Visualization:** Leveraging GPU capabilities, the visualization module will employ GPU shaders and advanced rendering techniques to allow real-time exploration of medical datasets. Different rendering modes will be made available, giving users the ability to visualize data from varying perspectives, such as volume rendering or multi-planar reconstruction.

2. **Denoising:** Medical datasets, particularly MRI ones, often contain inherent noise. This noise can obscure finer details, making diagnosis or study challenging. By integrating state-of-the-art denoising algorithms that take advantage of the GPU's parallel processing capabilities, we intend to enhance the clarity of these datasets in real-time. This will facilitate better interpretation and analysis of the images. A customizable "degree of denoise" slider will be incorporated into the interface, allowing users to fine-tune the level of noise reduction applied to the rendered medical images. This feature aims to provide flexibility and precision, facilitating more nuanced interpretation and analysis.

3. **Annotation Suite:** Beyond simple visualization, the platform will offer an intuitive annotation suite. This suite will present tools that allow users to mark specific regions of interest, add textual comments, and even use color-coded highlights. This not only aids in collaborative efforts between medical professionals but also provides a platform for educational endeavors.

Reflecting on the Original Approach and Adaptation to VTK: The initial plan outlined above was ambitious, aiming to harness GPU capabilities for advanced visualization and denoising, along with a comprehensive annotation suite. However, as the project evolved, I decided to utilize the Visualization Toolkit (VTK), a choice that brought both new possibilities and constraints. This shift necessitated some modifications to the original approach. While the GPU-based visualization and denoising were limited due to VTK's compatibility challenges, I adapted by focusing on VTK's robust capabilities in rendering and data processing. I successfully integrated a multi-stage denoising pipeline and developed a functional annotation system within the VTK framework. Although this meant a departure from some of the anticipated GPU-intensive processes, it allowed me to explore the rich features of VTK, resulting in a versatile and effective tool for medical data visualization and analysis.

3.2 Target Platforms

1. Hardware:
NVIDIA Graphics cards, specifically those compatible with CUDA for parallel computation, will be used to leverage GPU acceleration. This choice ensures maximal performance, as NVIDIA GPUs are at the forefront of computational and rendering tasks.

2. Software:

Initially, the following software components were considered for the project:

OpenGL: A cross-platform graphics API that will facilitate the rendering of medical datasets.
Python: This will serve as the backbone of the project, enabling data processing, communication between modules, and integrating third-party libraries for tasks like DICOM data parsing.
Pydicom: A Python library that aids in reading, modifying, and writing DICOM files, essential for accessing medical datasets.

However, due to a strategic shift in approach, the final implementation diverged from these initial plans:

Instead of OpenGL, I pivoted to using **VTK (Visualization Toolkit)**, an alternative that provided a comprehensive suite of tools for 3D computer graphics, image processing, and visualization. VTK offered robust features for medical data visualization, which aligned well with the project's goals. **Python** remained the backbone of the project, facilitating the integration of VTK and other necessary functionalities. Its versatility and wide range of libraries proved invaluable throughout the development process. For handling DICOM files, **vtkDICOMImageReader** from VTK was utilized. This shift from Pydicom to vtkDICOMImageReader was a natural transition, ensuring seamless compatibility with VTK's ecosystem and streamlined handling of medical imaging data.

This evolution in the choice of software tools was driven by the need for a more cohesive and VTK-centric development environment, ultimately shaping the project's direction and capabilities.

3.3 Evaluation Criteria

1. **Rendering Speed:** One of the primary criteria will be the speed at which 3D datasets are rendered, especially when compared to existing medical imaging software. The goal is to achieve real-time, lag-free interactions, enabling efficient exploration of the data.

2. **Clarity Post-Denoising:** The denoising module's effectiveness will be assessed by comparing the quality of images before and after the denoising process. Quantitative metrics, such as the Signal-to-Noise Ratio (SNR) and qualitative evaluations from medical professionals, will be crucial.

3. **Ease of Use:** Given the varied user base – from medical professionals to students – the interface's intuitiveness and ease of utilizing annotation tools will be of paramount importance. Feedback from trial users and time taken for specific annotation tasks can serve as good evaluation metrics.

4. RESEARCH TIMELINE

Project Milestone Report (Alpha Version)

- Completed background research.
- Set up the software framework with basic visualization.
- Integration of initial denoising algorithms.

Project Final Deliverables

- Fully functional software with all features.
- Documentation detailing usage and technical aspects.
- Demonstrative videos showcasing the tool's capabilities.

Project Future Tasks

- Future enhancements can include integration with AI algorithms for predictive analysis, collaborative online features, and support for more medical image formats.

5. METHOD

In this project, I developed a comprehensive medical imaging application focusing on image denoising, annotation, and advanced visualization techniques. The primary methodology involved integrating several key features:

Medical Imaging Display: Leveraging VTK and PyQt, I created an interactive interface for displaying medical images, supporting various formats including DICOM.

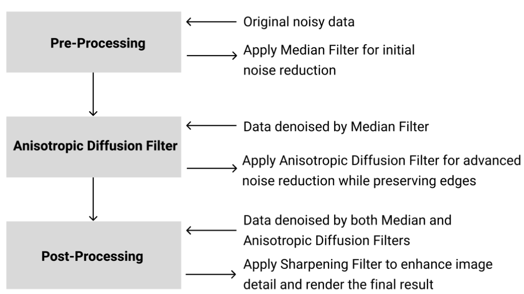
Annotation: Implemented functionality allowing users to annotate specific areas within the images, crucial for marking regions of interest or abnormalities.

Slice View: Enabled viewing individual slices from 3D medical scans, aiding in detailed analysis of specific cross-sections of the scanned data.

Bounding Box: Integrated a bounding box feature to define and focus on specific regions within the medical images, enhancing analysis efficiency.

Denoiser Pipeline: Initially, I intended to develop my denoising solution using OpenGL/CUDA. However, due to compatibility issues with VTK, which I chose as my primary toolkit, I had to revise my approach. To seamlessly integrate the denoiser into the VTK framework and enable real-time processing, I implemented a three-stage denoising pipeline. This pipeline effectively combines various image processing techniques to enhance medical scan clarity while maintaining compatibility with the VTK environment:

1. Median Filtering: Applied to reduce random noise while preserving edges.
2. Anisotropic Diffusion Filtering (ADF): Employed for further noise reduction and edge enhancement.
3. Edge Sharpening: Utilized a convolutional matrix to enhance and sharpen the edges, making the finer details more pronounced.



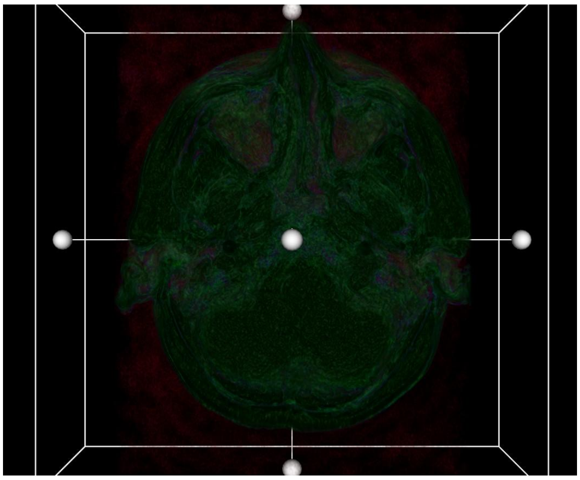
Each feature was carefully crafted, ensuring a balance between functionality and user-friendliness. Rigorous testing was conducted to ensure the accuracy and reliability of the features.

6. RESULTS

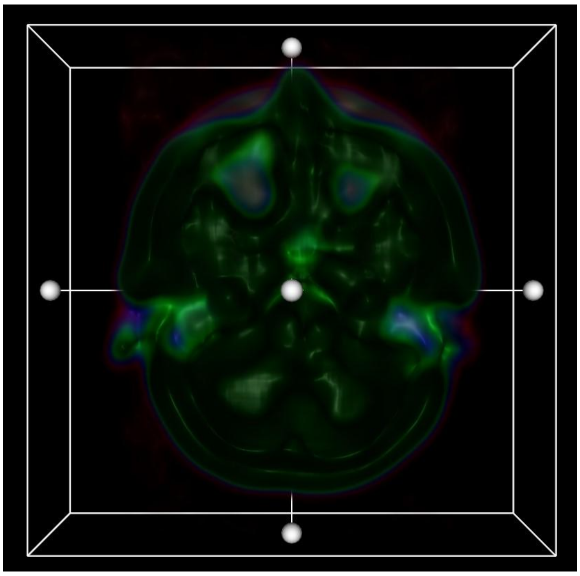
The application demonstrated excellent capability in processing and visualizing medical images. Key results include

Enhanced Image Quality: The denoising pipeline significantly improved the clarity of images, making it easier to identify and analyze critical features.

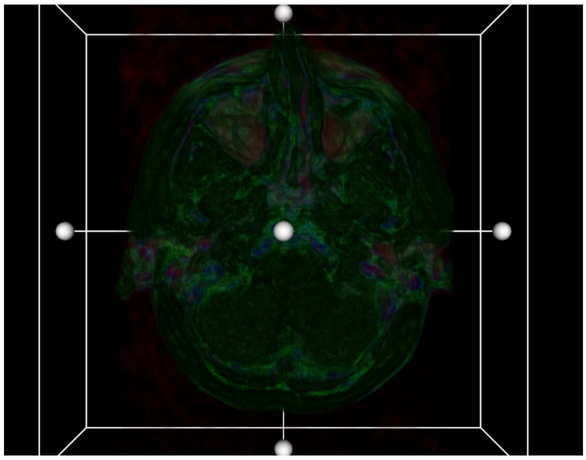
Graph 1: Undenoised Data



Graph 2: Denoised with Simple Gaussian Blur



Graph 3: Denoised with the three-stage denoising pipeline



Effective Annotation: Users could successfully mark and annotate areas of interest, facilitating better communication and record-keeping for medical analysis.

Intuitive Slice Navigation: The slice view feature provided a seamless experience in navigating through different layers of the scans, offering detailed insights into the internal structures.

Precise Region Identification: The bounding box tool allowed for precise isolation of areas within the scans, enabling focused analysis.

7. CONCLUSIONS AND FUTURE WORK

Conclusions:

This project successfully integrates vital features for medical image analysis, offering a robust and user-friendly tool for healthcare professionals. The denoising pipeline, in particular, stands out as a significant enhancement, providing clearer and more detailed images for diagnostic purposes.

Future Work:

Future enhancements could include:

- **Machine Learning Integration:** Implementing AI algorithms for automatic anomaly detection and diagnosis prediction.
- **3D Reconstruction:** Expanding capabilities to include 3D reconstruction of scans for a more comprehensive spatial understanding.
- **Telemedicine Features:** Adding functionality for remote sharing and collaboration on scans, catering to the growing field of telemedicine.

- **Performance Optimization:** Focusing on optimizing the processing speed and resource usage, especially for handling larger datasets.
- **Extended File Support:** Broadening the range of supported file formats and integrating more advanced image processing techniques.

By continuously evolving and integrating these advanced features, the application can significantly contribute to the field of medical imaging and diagnostics.

Gantt Chart Outline for Interactive 3D Visualization, Denoising, and Annotation Suite for Medical Data

1. Project Initialization
- Duration: 2 weeks
- Tasks:
  - Setting up the development environment
  - Reviewing initial literature and related work
  - Gathering necessary datasets

2. 3D Visualization Module Development
- Duration: 3 weeks
- Tasks:
  - Configuring GPU shaders for medical data rendering
  - Implementing interactive features (rotate, zoom, pan)
  - Testing initial visualization on sample datasets

3. Denoising Algorithm Integration
- Duration: 2 weeks
- Tasks:
  - Researching and selecting suitable denoising algorithms
  - Adapting chosen algorithms for GPU acceleration
  - Testing denoising on various datasets for effectiveness

4. Annotation Suite Development
- Duration: 1 week
- Tasks:
  - Designing user interface for annotation tools
  - Implementing mark, comment, and highlight functionalities
  - Integrating annotation data with the 3D visualization module

5. User Interface & Usability Testing
- Duration: 1 week
- Tasks:
  - Finalizing the user interface design
  - Conducting usability testing with sample users
  - Gathering feedback and making necessary adjustments

6. Performance Evaluation & Optimization
- Duration: 1 week
- Tasks:
  - Assessing rendering speed and denoising clarity
  - Making optimizations based on evaluations
  - Refining the tool based on user feedback

7. Documentation & Finalization
- Duration: 1 week
- Tasks:
  - Documenting software functionalities, user guides, etc.
  - Final tests and quality assurance
  - Preparing for final presentation/demo