MorseGramVis: A Visualization and Analysis Tool for Segmented Granular Media

A PROJECT REPORT
SUBMITTED IN PARTIAL FULFILMENT OF THE
REQUIREMENTS FOR THE DEGREE OF
Master of Technology
IN
Faculty of Engineering

BY

RATHOD DINESH

Computer Science and Automation
Indian Institute of Science
Bangalore – 560 012 (INDIA)

July, 2023

Declaration of Originality

I, **Rathod Dinesh**, with SR No. **04-04-00-10-51-21-1-19252** hereby declare that the material presented in the thesis titled

MorseGramVis: A Visualization and Analysis Tool for Segmented Granular Media

represents original work carried out by me in the **Department of Computer Science and Automation** at **Indian Institute of Science** during the years **2022-2023**.
With my signature, I certify that:

- I have not manipulated any of the data or results.
- I have not committed any plagiarism of intellectual property. I have clearly indicated and referenced the contributions of others.
- I have explicitly acknowledged all collaborative research and discussions.
- I have understood that any false claim will result in severe disciplinary action.
- I have understood that the work may be screened for any form of academic misconduct.

Date: Student Signature

In my capacity as supervisor of the above-mentioned work, I certify that the above statements are true to the best of my knowledge, and I have carried out due diligence to ensure the originality of the report.

Advisor Name: **Prof. Vijay Natarajan** Advisor Signature

DEDICATED TO

All those who have supported and inspired me throughout my journey

Acknowledgements

I want to express my deepest gratitude to my parents for their unwavering support and encouragement throughout my academic journey. Their love and guidance have been invaluable to me. I am also thankful to my sister and brother-in-law for their constant encouragement and belief in my abilities. Their support has been a source of motivation for me. I extend my heartfelt appreciation to my nephews, whose innocent smiles and joyful presence brought light to my life during the challenging moments of this research.

I am grateful to my professors for their mentorship and guidance. Their expertise and insights have shaped my thinking and enriched my understanding of the subject. I want to acknowledge my project colleagues for their collaboration and assistance. Their contributions have greatly influenced the outcome of this work.

Finally, I sincerely thank all those who have supported me throughout my academic journey, including my friends, classmates, and well-wishers.

I am humbled and deeply appreciative of these remarkable individuals' unwavering support and encouragement. Their belief in me and continuous assistance have been invaluable throughout my journey.

Abstract

Granular materials, such as sand, soil, and powders, are ubiquitous in various fields, including geotechnical engineering, environmental science, and pharmaceuticals. In recent years, there has been increasing interest in understanding the properties and behaviour of granular media, particularly at the microscale. One crucial aspect of granular media is the arrangement and segregation of particles within the material. This can significantly impact the mechanical and transport properties of the material and can be influenced by various factors such as particle size, shape, and composition.

Researchers often use computed tomography (CT) scanning techniques to study the microstructure of granular media to acquire high-resolution 3D images of the material. However, manually analyzing and quantifying the microstructure of granular media from these images can take time and effort.

To address this issue, we developed a visualization tool to assist researchers in visualizing and quantifying the segmentation [23, 10] and arrangements of granular media from CT-scanned images. The tool utilizes segmentation results from work [23] and allows users to visualize and analyze the microstructure in 3D interactively. Additionally, the tool provides various quantification features, including particle size distribution, particle shape distribution, and segregation analysis.

The tool has been tested on various granular media, including sand and steel beads, and has demonstrated the ability to reconstruct the sample to visualize and quantify these materials' microstructures. This tool will be a valuable resource for researchers studying granular media, as it provides a fast and user-friendly way to analyze and understand the microstructure of these materials.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Visualization is the process of creating and displaying graphical representations of data in order to understand and analyze it more effectively. It is an important tool for researchers in a variety of fields, as it allows for the clear and concise representation of complex information. In recent years, the use of visualization techniques has become increasingly prevalent in research, as advances in technology have made it possible to create highly detailed and interactive visualizations.

There are many different approaches to visualization, ranging from simple bar graphs and pie charts to more complex techniques such as 3D modeling and virtual reality. These methods can be used to represent a wide variety of data types, including numerical, categorical, and spatial data. In addition, visualization techniques can be used to explore and analyze data in real-time, allowing researchers to gain a deeper understanding of the relationships and patterns within the data.

Overall, visualization is an essential tool for researchers, as it enables them to more effectively understand and analyze complex data sets and make informed decisions based on their findings.

1.1 Motivation

The development of a visualization tool to assist in the visualization and analysis of granular media has the potential to greatly impact the field of granular media research. By providing researchers with a tool that allows for the simultaneous display of multiple perspectives of the granular media, as well as the ability to quantify the segmentation and arrangements of the particles, this tool would greatly facilitate the understanding and analysis of these complex systems.

Furthermore, the inclusion of quantifiable data on the segmentation and arrangements of the granular material would enable researchers to more accurately and efficiently analyze their samples, leading to more accurate and reliable results. This, in turn, could lead to more significant and impactful research in the field of granular media, with potential applications in fields such as engineering, materials science, and geology.

Overall, the significance of this study lies in the potential for the development of a visualization tool to greatly enhance the understanding and analysis of granular media and facilitate further research in this field.

1.2 Challenges

Research on granular media analysis involves studying the behaviour and properties of collections of granular particles. This field poses several challenges that researchers must address to gain meaningful insights and advance the understanding of granular systems. In this context, we will discuss some key challenges in analysing granular media. These challenges include:

- **Providing multiple views of granular media at particle and ensemble levels:** The challenge lies in efficiently visualizing large-scale granular systems while maintaining a high level of detail and interactive exploration capabilities.

- **Computing fabric tensor-related statistics and shape descriptors:** Fabric tensor-related statistics capture the orientation and anisotropy of particles within the granular media. This challenge involves developing algorithms and methods to calculate fabric tensors and derive meaningful statistics. Additionally, shape descriptors, such as particle elongation or aspect ratios, can provide valuable insights into particle morphology and arrangement.

- **Allowing identification and marking of undersegmented particles for further analysis:** Undersegmented particles refer to those not correctly separated or identified in the segmentation process. It is essential to develop tools and functionalities that allow users to identify and mark these undersegmented particles for further analysis. This can involve manual correction techniques or automated algorithms to refine the segmentation results.

- **Enabling parallel processing:** Granular media simulations and analysis often involve large datasets and computationally intensive tasks. Parallel processing techniques can significantly accelerate the processing and analysis of these datasets by distributing the workload across multiple processors or computing resources. Implementing efficient parallel algorithms and frameworks that leverage the available computational resources is challenging.

These challenges require a combination of algorithmic development, software engineering, and computational techniques to overcome. Addressing these challenges would enhance the capabilities of granular media analysis, providing researchers with advanced visualization tools, quantitative metrics, and efficient processing methods for better understanding and analysis of granular systems.

1.3 Objectives and Research Questions

The analysis and understanding of granular material is an important area of research with applications in fields such as engineering, materials science, and geology. However, the complex nature of granular media, with its numerous individual particles and their intricate arrangements, presents a significant challenge for researchers attempting to study and understand these systems.

One of the primary challenges in studying granular media is the difficulty in visualizing and analyzing the arrangements and distributions of the individual particles. Traditional methods such as microscopy and physical inspection are limited in their ability to provide detailed, multi-perspective views of the granular material, and may be prone to errors and subjectivity.

As a result, there is a need for effective visualization tools that can assist researchers in understanding and analyzing the characteristics of granular media. Such tools should be able to provide multi-perspective views of the material, as well as quantifiable data on the segmentation and arrangements of the particles. The development of such a tool would greatly aid researchers in understanding the complex systems present in granular media and facilitate further research in this field.

In order to address this need, the following research questions will be addressed in this study:

1. How can a visualization tool be developed to assist in the visualization and analysis of granular media that has been segmented?
2. What features should be included in the visualization tool in order to enable researchers to effectively understand and analyze the arrangements and distributions of the granular material?

3. How can the visualization tool be used to quantify the segmentation and arrangements of the granular material, and how can this data be effectively presented to researchers?
4. How can the visualization tool be tested and evaluated to ensure its effectiveness and usefulness for researchers studying granular media?
5. How can undersegmentation in granular media analysis be addressed, and what techniques can be employed to identify and mark undersegmented particles for further analysis?
6. What impact could the development of this visualization tool have on the field of granular media research, and how can it be used to facilitate further study in this area?

1.4 Contributions

This thesis makes several significant contributions to the field of granular materials research. The main contributions are as follows:

1.4.1 Building Visualization and Analysis Tool

Figure 1.1: The MorseGramVis tool supports multiple features for data analysis and visualization. The splash screen shows the different features and serves as an interface for the user to select a feature.

One of the primary contributions of this thesis is the development of a novel visualization and analysis tool for studying granular materials as shown in the figure 1.1. It enables researchers to visualize and analyze the microstructure of granular materials in a user-friendly and interactive 3D environment. This tool provides valuable features such as particle size distribution analysis and segregation analysis. The tool has been successfully tested on various granular media, including sand and steel beads, demonstrating its accuracy and effectiveness in quantifying the microstructure of these materials.

1.4.2 Improvement in pyms3d Library

This thesis contributes to improving the pyms3d library [26]. The pyms3d library is used for the segmentation of granular media. The improvements made in this thesis enhance the library's capabilities to handle more extensive and more detailed datasets, allowing researchers to work with higher-resolution CT-scanned images of granular materials.

1.4.3 MorseGram extensions and performance improvements

Another significant contribution of this thesis is further developing the MorseGram segmentation pipeline [20]. The MorseGram segmentation method is robust for segmenting granular materials from CT-scanned images.

- This thesis extends the capabilities of the MorseGram segmentation pipeline by enabling multi-processing, which significantly improves speed and efficiency. By leveraging parallel computing techniques, the segmentation pipeline can process large datasets more quickly, allowing for faster analysis and exploration of granular microstructures.

- Fixed the bug which improved the segmented particle mesh reconstruction. The bug was due to not including all the surface points and as a result, reconstructed mesh was smaller than actual surface.

- Automatic persistence threshold selection was implemented using the Kneedle library [28] to determine an optimal threshold for persistence-based topological simplification.

- Undersegmentation detection and added capability to perform multi-level segmentation.

These contributions collectively advance the field of granular materials research by providing researchers with enhanced tools and methods for visualizing, analyzing, and quantifying granular microstructures. The developed visualization and analysis tool, the improvements in the pyms3d library, and the further development of the MorseGram segmentation pipeline collectively contribute to a more comprehensive and efficient approach to studying granular materials.

Chapter 2

Literature Review

> Visualization tools are software programs that are used to create visualizations of data, typically in the form of 2D or 3D images. These tools have a long history, with the first known visualization tool being developed in the 1970s.

Visualization tools are crucial in various fields, including scientific research, data analysis, and decision-making. They enable the exploration and analysis of complex data by representing it visually and intuitively. In this literature review, we will discuss some popular visualization tools along with examples of their applications.

- **Tableau [1]:** Tableau is a widely used data visualization tool with a user-friendly interface for creating interactive visualizations. It offers various chart types, such as bar charts, scatter plots, and maps, allowing users to explore and present data from different perspectives. For example, Tableau has been used in business intelligence to analyze sales data and identify trends and patterns.

- **D3.js [5]:** D3.js (Data-Driven Documents) is a JavaScript library that allows the creation of dynamic and interactive visualizations on the web. It provides a robust set of tools for manipulating and binding data to visual elements, enabling the creation of custom visualizations. D3.js has been used in various domains, including data journalism, where it has been used to create interactive visual stories and data-driven infographics.

- **ggplot2 [36]:** ggplot2 is an R package for data visualization that follows the Grammar of Graphics principles. It provides a flexible and layered approach to creating visualizations, allowing users to customize and modify plots easily. ggplot2 has been widely used in the statistical community for exploratory data analysis and generating publication-quality graphics.

- **Matplotlib [14]:** Matplotlib is a popular Python library for creating static, animated, and interactive visualizations. It provides various plotting functions and supports various plot types, including line plots, scatter plots, histograms, and 3D plots. Matplotlib has been extensively used in the scientific community for data analysis and visualization, including in physics, biology, and neuroscience.

- **ParaView [2]:** ParaView is an open-source visualization tool for large-scale scientific data analysis. It supports parallel computing and distributed memory architectures, making it suitable for visualizing complex simulations and large datasets. ParaView has been applied in various scientific domains, including computational fluid dynamics, climate modelling, and medical imaging.

- **Gephi [4]:** Gephi is a network visualization tool that explores and analyses graph data. It provides interactive features for visualizing networks, such as node-link diagrams, matrix views, and dynamic filtering. Gephi has been used in social network analysis, biological network analysis, and information visualization.

These examples highlight the diversity of visualization tools available and their applications across domains. Each tool offers unique features and capabilities, allowing users to gain insights and make informed decisions based on their data. The choice of a visualization tool depends on the specific requirements of the data, the desired level of interactivity, and the target audience.

2.1 Overview of granular materials

Granular materials [15] are collections of discrete solid particles, such as sand, soil, powders, grains, or granules, that exhibit unique behaviours and properties due to their particulate nature. These materials are encountered in various fields, including geotechnical engineering, environmental science, pharmaceuticals, food processing, etc.

Several key features characterize granular materials:

1. **Particle Interactions:** The behaviour of granular materials is primarily governed by the interactions between individual particles. These interactions can be influenced by particle size, shape, composition, surface properties, and the presence of interstitial fluids.

2. **Discrete Particle Nature:** Unlike continuous materials like fluids or solids, granular materials consist of discrete particles that can move and rearrange. This leads to unique phenomena such as particle flow, compaction, and segregation.

3. **Granular Packing:** Granular materials can form various packing arrangements, from loose and random to densely packed structures. The arrangement of particles significantly affects the mechanical and transport properties of the material, including its strength, permeability, and compressibility.

4. **Flow and Jamming:** Granular materials can flow like fluids under certain conditions, exhibiting characteristics of both solids and liquids. However, they can also undergo jamming, where particle interactions and geometric constraints lead to a solid-like behaviour and hinder flow.

5. **Segregation:** Granular materials often exhibit particle segregation, where particles of different sizes, shapes, or densities separate from each other due to various mechanisms such as vibration, shaking, or flowing. Segregation can impact the homogeneity and stability of granular assemblies.

Understanding the behaviour of granular materials is essential for numerous applications. In geotechnical engineering, knowledge of granular mechanics is crucial for designing foundations, slopes, and retaining structures. In pharmaceuticals, powders' flow and mixing behaviour affect drug formulation and manufacturing processes. In environmental science, studying granular media helps analyze pollutants' transport in soil or porous media's filtration properties.

Researchers employ various experimental and computational techniques to study granular materials, including CT scanning, microscopy, numerical simulations, and physical testing. These approaches aim to investigate granular microstructure, particle-scale interactions, flow behaviour, compaction, segregation mechanisms, and other properties to develop models and theories that can describe and predict the behaviour of granular materials.

2.2 Previous studies on granular materials

Granular materials have been extensively studied in various scientific disciplines. Previous research has contributed to a deeper understanding of their behaviour and properties. Here, we highlight a few notable studies in the field:

- In geotechnical engineering, Terzaghi and Peck conducted pioneering research on soil mechanics, including the behaviour of granular soils [34]. Their work laid the foundation for analyzing the stability and settlement of foundations and the mechanics of earth structures.

- Jaeger and Nagel investigated granular flow and jamming phenomena through experiments using photoelastic disks [16]. Their work revealed the critical role of particle interactions in granular flow and established a framework for understanding the transition from flowing to jammed states.

- Cates and Wittmer explored the concept of granular segregation, demonstrating how size and density differences between particles can lead to self-sorting and stratification in granular materials [6]. Their research shed light on the mechanisms underlying segregation and its impact on granular assemblies' structural and mechanical properties.

These studies represent only a fraction of the extensive research on granular materials. The collective body of work has contributed to a comprehensive understanding of granular behaviour, paving the way for advancements in numerous fields and applications.

2.3 Related work on visualization and analysis tools for granular materials

2.3.1 Visualization of Particle Interactions in Granular Media

The work[19] introduces a two-scale homogenization approach for granular materials to reduce computational costs and gain insights into discontinuous materials. The main contribution is a novel 2D visualization tool that supports micro and macro-scale visualization, enabling practical analysis of particle behaviour. Interactive rose diagrams are developed to represent dynamic contact networks on the micro-scale in a condensed and efficient manner.

2.3.2 FEM

FEM (Finite Element Method) is a numerical technique for solving engineering problems by dividing a complex structure into small finite elements. ABAQUS[11] and ANSYS[3] are popular software packages that implement FEM for analyzing and simulating structural and mechanical behaviours of materials and systems. These tools provide capabilities for modelling, meshing, applying boundary conditions, solving equations, and visualizing results, making them widely used in various industries for design, analysis, and optimization.

2.3.3 DEM

DEM (Discrete Element Method) is a numerical simulation technique used to model and analyze the behaviour of granular materials, particles, or discontinuous materials. Itasca[13] is a software that specializes in implementing the DEM approach. It provides tools for modelling and simulating the interactions between individual particles, considering their shape, size, and material properties. Itasca is commonly used in geotechnical engineering, mining, and other fields where the behaviour of granular materials is of interest.

2.4 Background

Consider a smooth, twice differentiable scalar function $f : \mathbb{R}^3 \to \mathbb{R}$. A point $p_c \in \mathbb{R}^3$ is called a critical point of f if the gradient of f at p_c is zero ($\nabla f = \mathbf{0}$). The critical point p_c is called non-degenerate if its Hessian, which is the matrix of second partial derivatives at p_c, is non-singular. A function f is called a Morse function if all its critical points are non-degenerate and have distinct function values. It is worth noting that any smooth function f can be infinitesimally perturbed into a Morse function and refer the reader to the books on this topic for further understanding [7].

In the context of Morse theory, the number of negative eigenvalues of the Hessian corresponds to the Morse index of a critical point. For a three-dimensional function f, critical points can be classified into four types: minima (index-0), 1-saddles (index-1), 2-saddles (index-2), and maxima (index-3).

In the discrete case, the concept of Morse theory can be modeled using a piecewise continuous function. Instead of considering a smooth function over a continuous domain, we work with a discrete domain where the function is defined over a set of discrete points. The critical points [8] and their indices can still be identified based on the behavior of the function values at these points.

By applying Morse theory to the discrete case, we can analyze the topological features and critical points of a scalar field defined on a discrete grid. This allows us to gain insights into the connectivity, structure, and behavior of the scalar field, even in cases where the function is not smooth.

Overall, Morse theory provides a powerful framework for understanding the critical points and topology of scalar functions, both in the continuous and discrete domains. It enables us to study the behavior of functions, identify important features, and extract meaningful information from complex datasets.

2.4.1 Morse-Smale Complex

The Morse-Smale Complex (MSC) is a mathematical construct used in the field of computational topology to analyze and describe the topological features of scalar fields. It is based on the Morse theory, which studies the critical points (such as local minima, maxima, and saddle points) of a scalar function defined on a manifold.

The MSC provides a hierarchical decomposition of the domain into regions, each associated with a different critical point. It captures the connectivity and flow patterns of the scalar field by considering the gradient vector field and the stable and unstable manifolds associated with the critical points. These manifolds are defined as the sets of cells where the gradient vector points towards or away from the critical point, respectively.

The MSC consists of a collection of critical points, separatrices (trajectories connecting pairs of critical points), and cells that represent regions of the domain bounded by separatrices. The cells are labeled with the indices of the critical points they contain, indicating the flow behavior within each region. The MSC is computed from the work [30, 31].

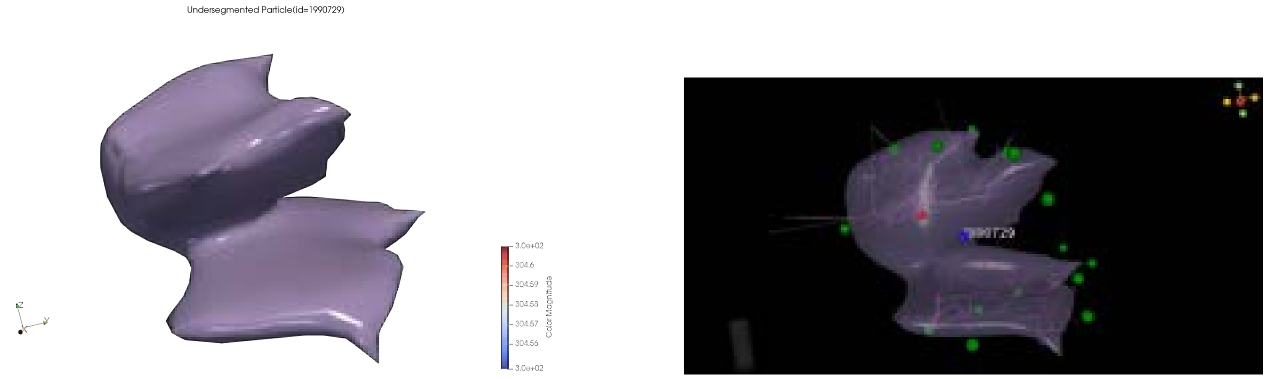
By analyzing the MSC, various topological features of the scalar field can be identified, such as ridges, valleys, basins, and boundaries between different regions. This information is useful for understanding the structure, connectivity, and behavior of scalar fields in diverse applications, including data analysis, image processing, computer graphics, and scientific visualization.

2.4.2 Simplified Saddle Graph

Scientific data is often noisy, manifesting as more critical points. Topological simplification identifies noisy topological features and removes them in a controlled manner, often as a sequence of critical point pair cancellation operations and resulting in a simpler MSC [9, 8]. The notion of persistence typically directs the simplification.

The segmentation algorithm performs persistence-driven cancellation, which cancels 2-saddle-maximum pair connected by an arc in the ms complex and reconnects the neighbourhood of both nodes.

As a result, every cancelled or simplified 2-saddle is connected to 2 maximums forming a triplet(maximum-(2-)saddle-maximum)—a collection of all triplets form a simplified saddle graph and an example of such a graph in shown in the figure 2.1.



(a) Undersegmented Particle (b) Simplified saddle graph

Figure 2.1: Simplified saddle graph(where pink balls represent 2-saddles). The right figure illustrates a graph showing a specific pattern, where two subgraphs are connected by an edge.

2.5 Details of Dataset used

- **ODO** - Oedometric compression (confined one dimensional compression)
- **Chess** - Cornell High Energy Synchrotron Source

Chapter 3

Methodology

3.1 Segmentation algorithm and implementation

Morse theory-based framework [23] is employed to segment 3D X-ray CT images of granular ensembles and compute their particle arrangement. The framework includes an algorithm for segmentation, a data structure for storing the segmentation and connectivity network, and interactive visualizations of topological descriptors. Compared to watershed transform-based approaches, the Morse theory-based framework achieves higher-quality segmentation and improved accuracy in the connectivity network. It also enables efficient computation of distribution statistics, facilitating a comprehensive understanding of granular ensemble behaviour across multiple length scales.

3.2 Input data acquisition and preprocessing

Input data acquisition and preprocessing are critical steps in any data analysis workflow. In this work, the data is obtained from the morsegram segmentation pipeline [20], which generates several output files: labels.vtp, contacts.vtp, connectivity network.vtp, and maxima centers.vtp. These files contain important information for further analysis. Here is a brief description of each file:

- **labels.vtp**: This file contains the segmented labels or regions of interest from the morsegram segmentation. Each label represents a distinct granular particle in the material.

- **contacts.vtp**: The contacts file stores information about the contact between particles in the granular material. It provides data on the location and region of contacts, which are crucial for understanding particle interactions.

- **connectivity network.vtp**: This file contains information about particles connectivity where each particle is represented as node in the connectivity graph. It provides a visual and structural representation of particle-particle connectivity within the granular material.
- **maxima centers.vtp**: This file contains the positional information (x,y,z) of maxima critical point associated with individual particles in the granular material. It includes each particle's coordinates or spatial data, allowing for spatial analysis and visualization.

These files serve as further analysis and visualization inputs, enabling a detailed exploration of the granular material's microstructure and particle interactions.

3.3 MorseGramVis Overview

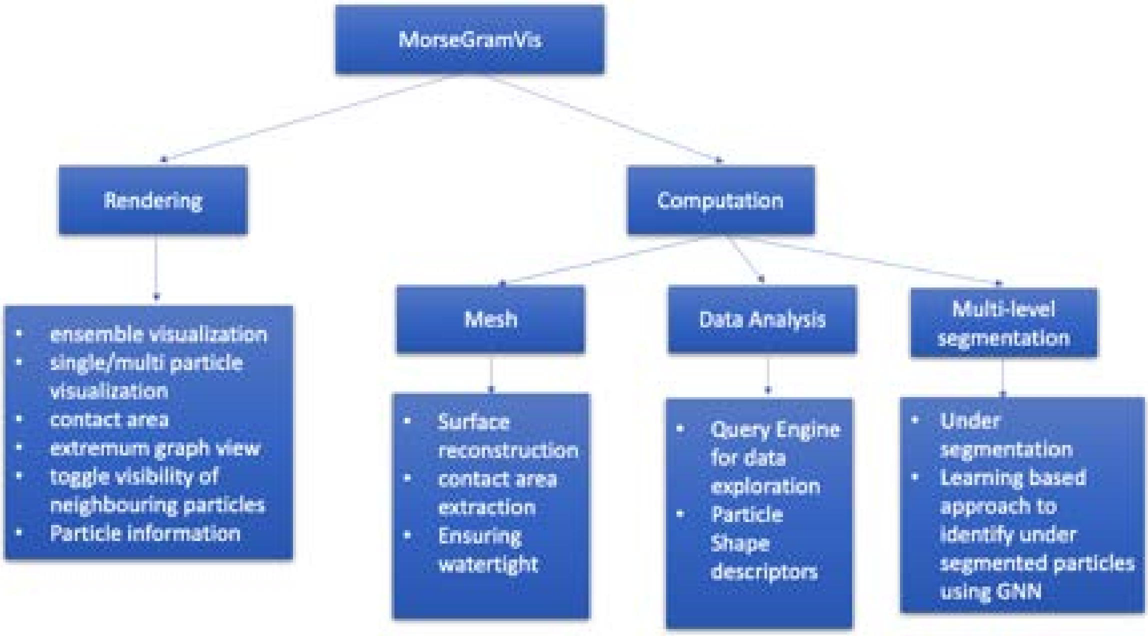


Figure 3.1: System Design Chart

The tool provides various functionalities for rendering and computation of particle-based data as shown in the figure 3.1. The key features are as follows:

1. **Rendering**
    - **Ensemble Visualization:** Enables visualization of multiple particles together.
    - **Single/Multi Particle Visualization:** Supports visualization of individual particles or groups of particles.

- **Contact Area Visualization:** Allows visualization of contact regions between particles.
- **Extremum Graph Visualization:** Provides visual representation of extremum graphs.

2. **Computation**

    - **Mesh:** Includes functionalities for mesh generation and surface reconstruction.
    - **Data Analysis:** Supports various data analysis techniques, such as PCA and shape descriptors.
    - **Multi-Level Segmentation:** Offers capabilities for segmenting data at multiple levels.

These features make the tool versatile and suitable for tasks related to particle visualization, analysis, and segmentation.

3.3.1 Computational aspects of the MorseGramVis

3.3.1.1 Alpha complex based methods

In order to create a 3D visualization of the granular media, it was necessary to compute a mesh for each segmented particle in the media. This was achieved through the use of library Visualization ToolKit (VTK) [29] and an illustration is shown in the figure 3.2.

A vtkDelaunay3D filter of the VTK library performs surface reconstruction by constructing a 3D Delaunay triangulation from a set of input points. The filter generates an unstructured grid dataset as output, typically as a tetrahedral mesh.

By default, the filter produces a tetrahedral mesh but also supports using an "alpha" value. If a non-zero alpha distance is specified, only tetrahedra, triangles, edges, and vertices within the alpha radius are included in the output. This allows for more control over the resulting mesh, enabling the generation of combinations of tetrahedra, triangles, lines, and vertices.

Experiments showed that using an alpha value of 3 produced better results. The alpha value defines the radius within which elements are considered for inclusion in the output. When set to 3, only tetrahedra within a distance of 3 units from the input points are included. This choice balances the level of detail in the surface reconstruction while maintaining a smooth and visually pleasing representation. This filter is configured to skip the lower dimensional cells like triangles and lines in the final output to avoid edges sticking out from the surface and triangle strip creation. Finally boundary surface is extracted from the tetrahedral mesh, and

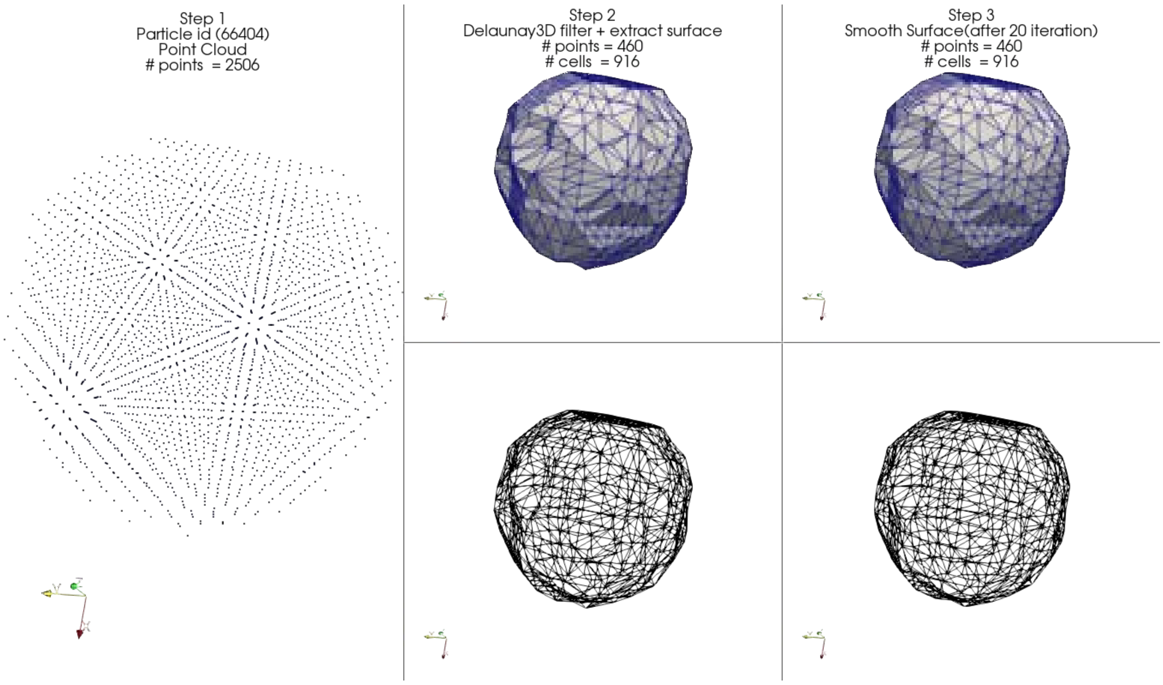


Figure 3.2: Alpha complex based methods

then smoothing is applied to enhance the visual appearance of the surface. Smoothing helps to reduce noise and irregularities in the mesh, resulting in a more aesthetically pleasing and visually coherent representation. The choice of the smoothing algorithm and the number of smoothing iterations can be tailored to the application's specific requirements and the desired surface smoothness level. Care should be taken to avoid over-smoothing, which can lead to losing essential surface details.

Algorithm 1 Alpha complex based methods

```
Require: Input point cloud dataset
  Initialize a VTK point cloud object using the input dataset
  Apply the Delaunay 3D triangulation filter (vtkDelaunay3D) to the point cloud
  Set the alpha value for the Delaunay filter to control the output geometry
  Disable the output of lines and triangle strips to obtain only tetrahedra
  Extract the boundary surface using the vtkDataSetSurfaceFilter
  Apply a smoothing filter (e.g., vtkSmoothPolyDataFilter) to refine the surface
  return smooth surface (collection of triangles)
```

3.3.1.2 Voxel modelling based methods

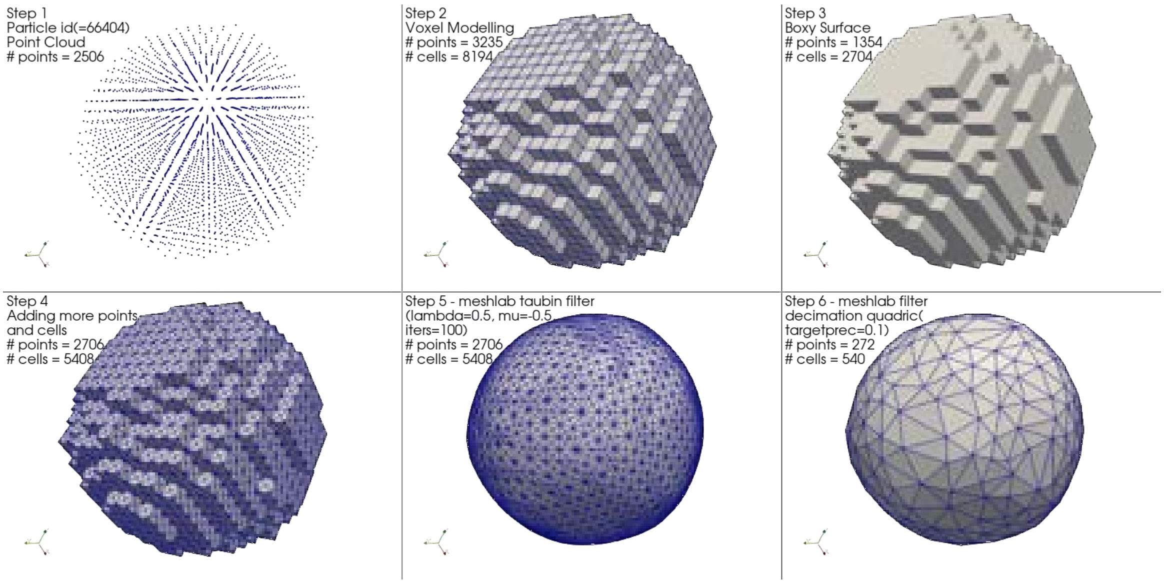


Figure 3.3: Voxel modelling based methods

To illustrate the concept, reader is referred to the figure 3.3. Inspired from the work [22], this method takes an input point cloud dataset as its input. It starts by creating a voxel grid from the input point cloud dataset. A voxel grid is a 3D grid where each grid cell represents a small volume element (voxel). The voxel grid discretizes the space occupied by the point cloud. Next, each point in the input dataset is replaced with a voxel in the voxel grid. This step transforms the continuous point cloud into a discrete representation using voxels.

Next, it extracts the boxy surface from the voxel grid. The boxy surface refers to the surface that encloses the voxels and represents the approximate shape of the original point cloud.

A strategy can be employed to improve the mesh surface obtained from the boxy surface, where new points are added to the existing surface. These new points are computed as the centres of each cell of the boxy surface. The boxy surface is initially extracted from the voxel grid representation of the object. Each cell of the boxy surface corresponds to a voxel in the grid. By computing the centre of each cell, a new point is generated. Once these new points are obtained, the next step is to replace each cell of the boxy surface with four triangles. This is done by creating triangles connecting the newly added point with the vertices of the cell. This process effectively refines the mesh surface, increasing its resolution and improving the overall quality. By adding new points and replacing cells with triangles, the mesh surface becomes more detailed and captures the finer features of the object. This refinement step is crucial for achieving a smoother and more accurate representation of the underlying surface.

Finally, the mesh obtained from previous strategy is refined and smoothed using external tools such as MeshLab. MeshLab provides various smoothing techniques that can be used to get visually appealing surfaces.

```
Algorithm 2 Voxel modelling based methods
Require: Input point cloud dataset
  1. Create a voxel grid from the input point cloud dataset by replace each point in the dataset
  with a voxel in the grid
  2. Extract the boxy surface from the voxel grid
  3. Add more points(center of quad cell) and replace quad face with triangular cells.
  4. Apply smoothing techniques (e.g., using MeshLab) to refine the mesh
  5. Apply decimation quadric filter (e.g., using Meshlab) to reduce size of mesh
  return smooth surface (collection of triangles)
```

3.3.1.3 Particle Statistics

In order to facilitate the analysis of granular media, it is important to be able to compute various particle statistics. These statistics can provide valuable information about the characteristics of the particles and their arrangements within the media and corresponding interface is shown in the figure 3.4a.

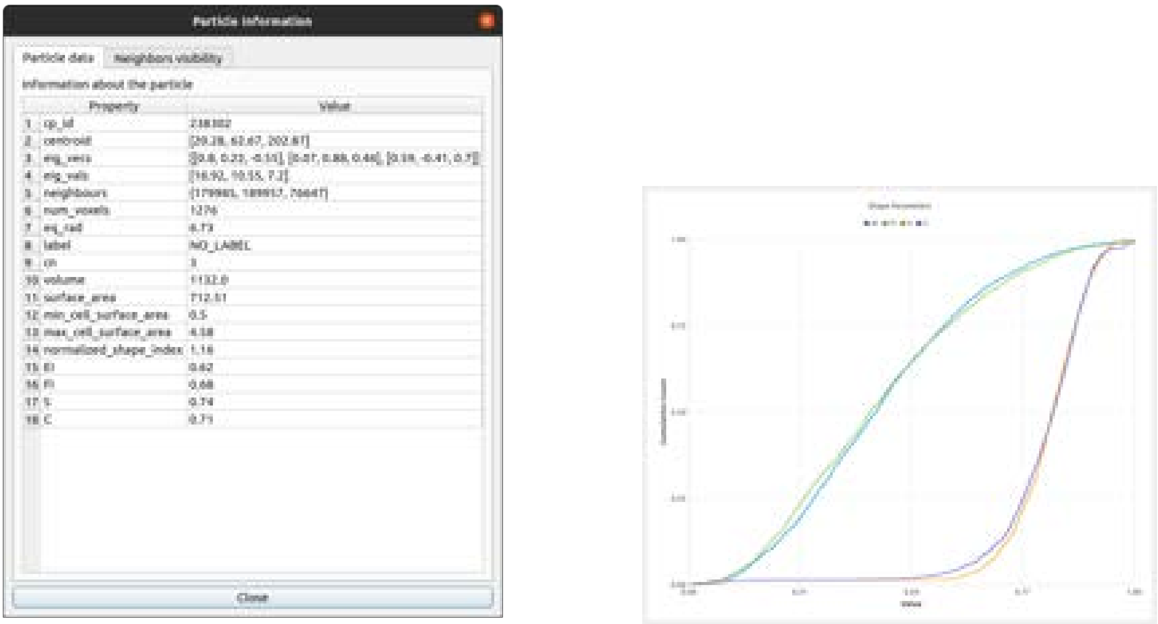
One example of a particle statistic that can be useful for analysis is the eigenvalues and eigenvectors of the particles. These quantities describe the shape and orientation of the particles, and can be used to understand how they fit together within the media.

Another important statistic is the equivalent radius of the particles, which is a measure of their size. This can be used to understand the distributions of particle sizes within the media and their potential effects on the overall properties of the system.

In addition to these statistics, it can also be useful to compute the volume of the particles, as well as an adjacency list that specifies which particles are adjacent to each other. These statistics can provide valuable information about the arrangements and distributions of the particles within the media.

To enable data analysis, it is important to store these statistics in a format that is easily accessible and manipulatable, such as a csv file. This allows researchers to easily import the data into a spreadsheet or statistical software for further analysis.

Overall, the computation and storage of particle statistics such as eigenvalues, equivalent radius, volume, and adjacency list can greatly aid in the analysis and understanding of granular media, enabling researchers to more effectively.



Information about the particle

	Property	Value
1	cp_id	238302
2	centroid	[29.28, 62.67, 202.87]
3	eig_vecs	[[0.8, 0.23, -0.53], [0.07, 0.88, 0.46], [0.59, -0.41, 0.7]]
4	eig_vals	[16.92, 10.55, 7.2]
5	neighbours	[179983, 189937, 76647]
6	num_voxels	1276
7	eq_rad	6.73
8	label	NO_LABEL
9	cn	3
10	volume	1132.0
11	surface_area	712.51
12	min_cell_surface_area	0.5
13	max_cell_surface_area	4.58
14	normalized_shape_index	1.16
15	EI	0.62
16	FI	0.68
17	S	0.74
18	C	0.71

(a) Particle Statistics Information Window

(b) Cumulative distribution of shape parameters for ODO dataset

Figure 3.4: Particle Shape Descriptors

3.3.1.4 Contact Statistics

In addition to particle statistics, it can also be useful to compute contact statistics for granular media. These statistics provide information about the interactions between particles and can be used to understand the behavior and properties of the media.

Some examples of contact statistics that can be computed include the number of quadrilaterals per contact per particle, eigen values and vectors, and the contact area distribution. These statistics can provide valuable insights into the forces and interactions within the media and how they influence the overall behavior of the system.

Like particle statistics, it is important to store contact statistics in a format that is easily accessible and manipulatable, such as a csv file. This allows researchers to easily import the data into a spreadsheet or statistical software for further analysis.

Overall, the computation and storage of contact statistics can greatly aid in the analysis and understanding of granular media, enabling researchers to more effectively study the complex interactions within these systems.

3.3.1.5 Multi-level Segmentation

One common problem that can arise in the segmentation of granular media is undersegmentation, in which individual particles are not properly distinguished and are instead treated as a single entity. This can make it difficult to accurately analyze and understand the characteristics of the media.

To overcome the problem of undersegmentation, it is often necessary to use a multi-level segmentation approach. This involves using multiple levels of segmentation, each with a different level of detail, in order to more accurately distinguish individual particles as shown in the figure 3.5.

For example, a multi-level segmentation approach might involve using a coarse level of segmentation to identify the general shape and location of the particles, followed by a finer level of segmentation to distinguish individual particles and their features.

By using a multi-level segmentation approach, it is possible to more accurately and reliably segment granular media and overcome the problem of undersegmentation.

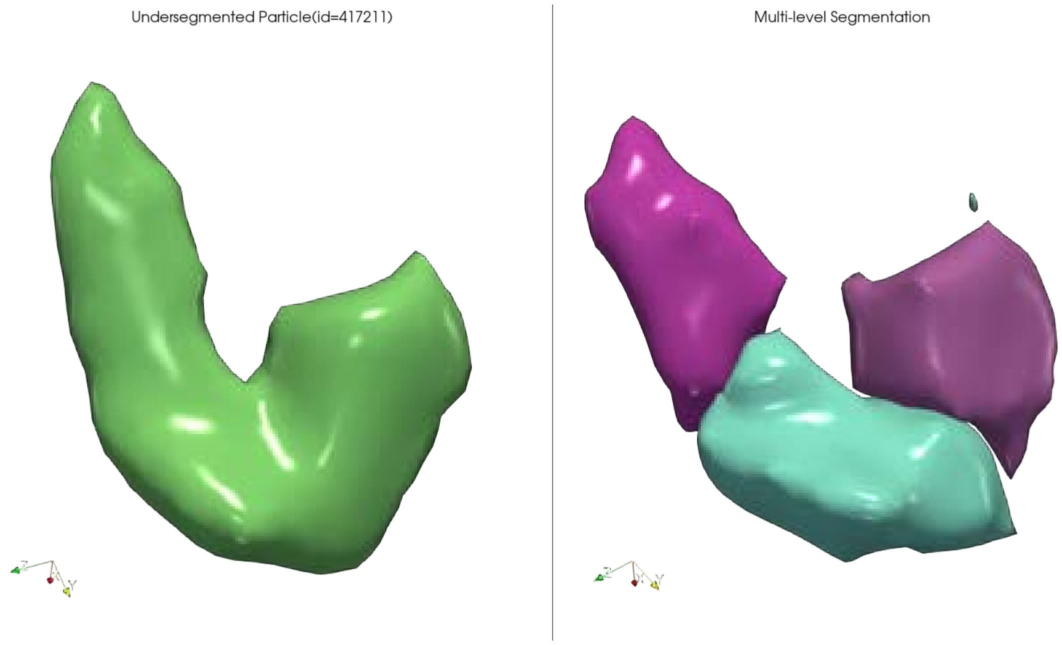


Figure 3.5: Multi-level Segmentation

3.3.2 Rendering aspects of the MorseGramVis

3.3.2.1 Ensemble Visualization

In order to facilitate the visualization and analysis of the granular media, the rendering tool included a feature for ensemble visualization. This feature allowed researchers to view the granular media as a whole, rather than individual particles, enabling them to more easily understand the overall arrangements and distributions of the particles as shown in the figure 3.6.

To enhance the realism and detail of the ensemble visualization, the rendering tool also incorporated ambient occlusion. Ambient occlusion [25] is a shading technique that simulates the way light is absorbed and scattered by objects in a scene. It creates the illusion of depth and volume by adding shadows and highlights to the surfaces of the particles, making the ensemble visualization more visually appealing and easier to interpret.

Overall, the ensemble visualization with ambient occlusion feature of the rendering tool provided researchers with a powerful tool for understanding and analyzing the granular media, allowing them to more easily understand the relationships and patterns within the data.

Figure 3.6: Ensemble Visualization(enabled ambient occlusion) of Steel Beads Dataset

Algorithm 3 Construct Ensemble Data

```
ensembleData $\leftarrow$ None {Initialize ensemble data as empty}
for each particle point cloud in the segmentation output do
    1. retrieve the particleId from point cloud
    2. Read the mesh data from file particleId.vtp
    3. Append the mesh data to ensembleData
end for
Write ensembleData to a file (ensemble.vtu) {Unstructured grid data}
```

3.3.2.2 Single/Multi Particle Visualization

In addition to the ensemble visualization feature, the rendering tool also included a single particle visualization feature as shown in the figure 3.7a, which allowed researchers to view individual particles in the granular media along with their neighbors as shown in the figure 3.7b. This feature was particularly useful for studying the arrangements and distributions of the particles at a more detailed level.

The single particle visualization feature was implemented using a sliding window approach, in which a user could select a single particle and view a specified number of its neighbors. This allowed researchers to more easily understand the local arrangements and distributions of the particles, as well as their relationships to their neighbors.

Overall, the single particle visualization feature provided researchers with a valuable tool for studying the granular media at a more detailed level, enabling them to more easily understand the complex relationships and patterns within the data.

(a) Single Particle View

(b) Neighbours View

Figure 3.7: Single/Multi Particle Visualization of Steel Beads Dataset

3.3.2.3 Extremum Graph Visualization

Extremum graphs are a type of graph that are used in computational topology to represent the topological structure of a dataset. An example is shown in the figure 3.8. The idea behind extremum graphs is to represent the topological features of a dataset by connecting local extrema of the data. These extrema can be maxima, minima, or saddle points, and the edges of the graph connect pairs of extrema that are in some sense close to each other. The resulting graph captures the topological structure of the data, and can be used to analyze features such as connected components, loops, and voids.

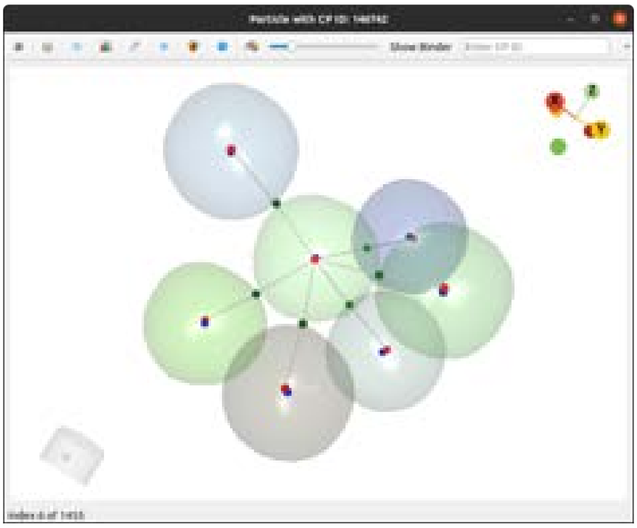


Figure 3.8: Extremum Graph Visualization of Steel Beads Dataset

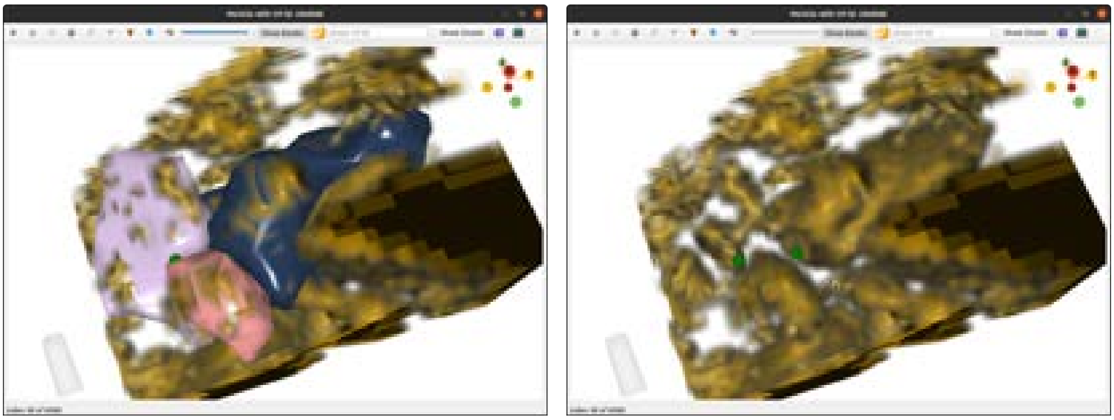
3.3.3 Binder Visualization

In binder visualization using volume rendering [17], the goal is to represent the presence of binder material surrounding a particle visually as shown in the figure 3.9. Here's a brief explanation of how volume rendering can help achieve this.

Volume rendering is a technique that allows the visualization of volumetric data, such as 3D particle structures and their surroundings. In the case of binder visualization, the binder material is typically represented as a different density or material from the particles themselves.

Using volume rendering, the binder material can be assigned a specific color or opacity, distinct from the particles. This enables the visualization of the binder as a surrounding medium, helping to identify its distribution and penetration within the particle structure.

Different visual effects can be achieved by adjusting the transfer function in the volume rendering process. For example, increasing the opacity of the binder material can highlight regions with a higher concentration of binder, while reducing the opacity can reveal internal particle structures.



(a) Three particles along with volume rendering of scalar field extracted using booud-ing box.

(b) volume rendering of scalar field, green sphere represent the point of contact between particles and surrounding white space indicates presence of no binder.

Figure 3.9: Binder Visualization

The volume rendering technique provides a valuable tool for visualizing and analyzing the presence and distribution of binder material surrounding particles. It aids in gaining insights into the binder-particle interaction, identifying areas of binder accumulation or depletion, and assessing the quality of binder distribution within the sample.

By refining the volume rendering settings, researchers and scientists can effectively examine and evaluate the behaviour and characteristics of the binder material within the particle system, facilitating further analysis and optimization of binder distribution in various applications.

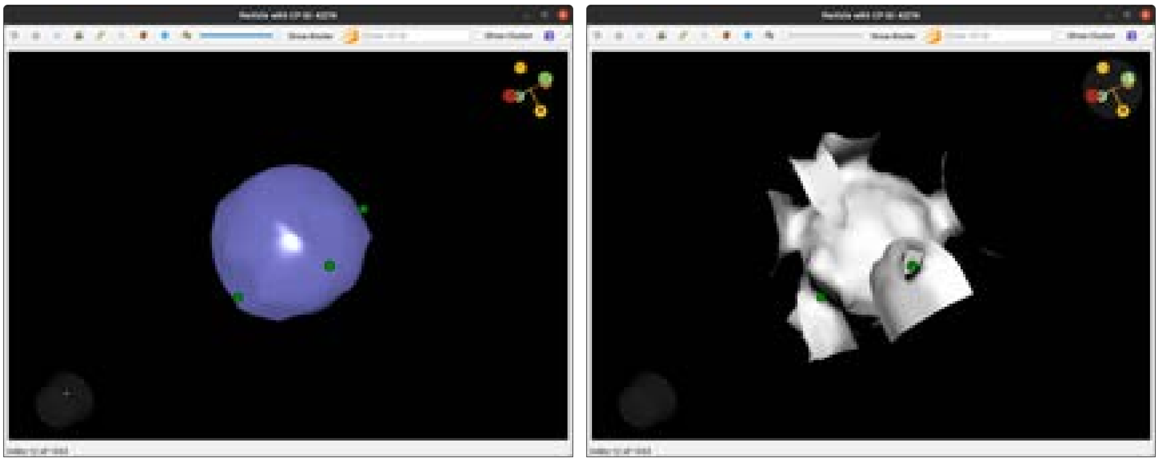
3.3.4 Isosurface Visualization

An isosurface [18] is a three-dimensional (3D) surface representing a specific value (known as the isovalue) within a volumetric dataset. An illustration is provided in the figure 3.10. It is a visualization technique commonly used in scientific and engineering fields to extract and visualize regions of interest within volumetric data.

Each point in the dataset has an associated value in a volumetric dataset, such as a 3D scalar field or a medical image. The isosurface represents the set of points where the value equals the specified isovalue. It forms a continuous surface that separates regions with different values.

The isosurface visualization within the single/multi-particle visualization tool is helpful for debugging and analyzing particle segmentation and contact regions. Here's how it can assist in the debugging process:

- **Particle Segmentation:** The isosurface allows you to visualize the surface of individual particles in the 3D space. By adjusting the threshold value of the isosurface, you can control the boundaries of segmented particles. This helps you visually inspect and verify the accuracy of the particle segmentation process. You can identify undersegmented or oversegmented regions, ensuring particles are appropriately separated.
- **Contact Regions:** The isosurface visualization also helps identify contact regions between particles. By adjusting the threshold value, you can isolate and visualize the areas where particles come into contact with each other. This is particularly useful for debugging cases where particles are incorrectly merged or separated, leading to contact regions that should not exist or missing contact regions that should be present.



(a) Steel Bead Particle (b) Isosurface(isoval=0)

Figure 3.10: Isosurface Visualization

By examining the isosurface visualization, you can more effectively identify and debug potential issues in particle segmentation and contact regions. This visual feedback allows you to gain insights into the quality of the segmentation process and identify areas that require improvement. It aids in fine-tuning segmentation parameters and refining the algorithm to enhance the accuracy of particle segmentation and contact detection.

3.3.5 Data Analysis

The following data analysis techniques were employed to extract meaningful information from the dataset:

3.3.5.1 Statistical Analysis of Dataset

Statistical analysis was conducted on the dataset, involving the following aspects:

1. Principal Component Analysis (PCA) of Point Clouds: PCA was performed on the point clouds to identify the principal directions of variation and capture the most significant features of the dataset.

2. Analysis of Contact Regions: The contact regions between particles were analyzed to determine their characteristics, such as size, shape, and distribution. This analysis provided insights into the contact network and its influence on the behaviour of the granular material.

3. Volume Analysis: The volume occupied by the granular material was computed to quantify its overall size and assess any variations within the dataset.

4. Dominant Direction of Particles and Contacts: The dominant directions of particles and their contacts were determined to understand the granular assembly's preferential orientations and alignment patterns.

3.3.5.2 Particle Shape Descriptors

To characterize the shape of individual particles, the following shape descriptors were computed and for more details, reader is referred to [37] and an illustration is provided in the figure 3.11 and its distribution is illustrated in the figure 3.4b:

1. Elongation Index(EI): This index quantifies a particle's elongation or elongation tendency, indicating its degree of elongation along a specific direction.

2. Flatness Index(FI): The flatness index measures the flatness or planarity of a particle, representing the extent to which it deviates from a perfectly spherical shape.

3. Sphericity(S): Sphericity measures how closely a particle resembles a sphere. It provides insights into the roundness or compactness of the particle shape.

4. Compactness(C): The compactness of a particle indicates how tightly packed its geometry is, reflecting its density or degree of densification.

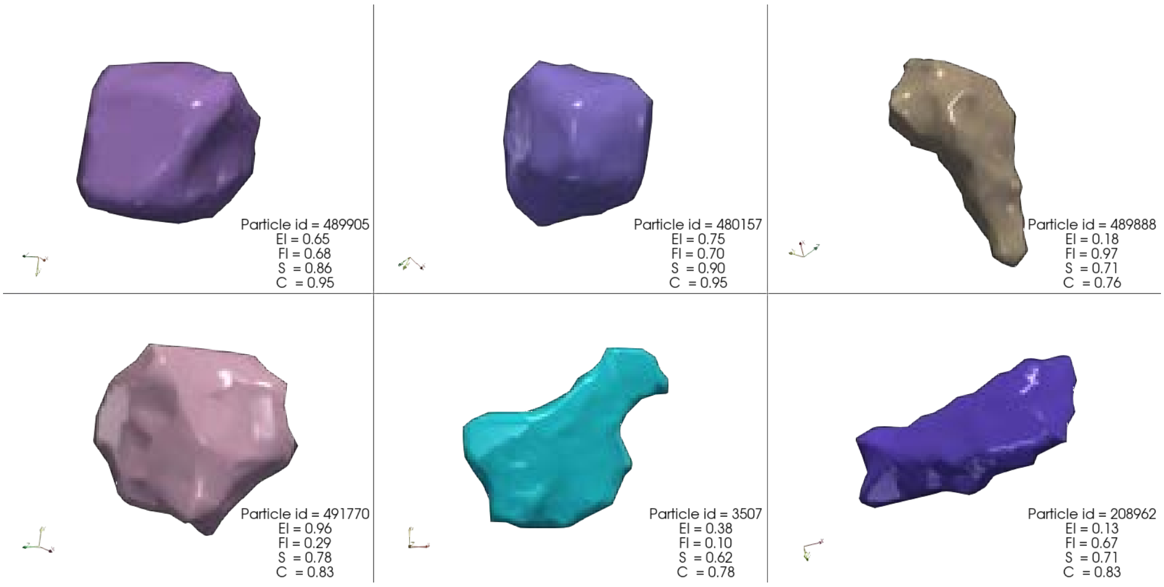


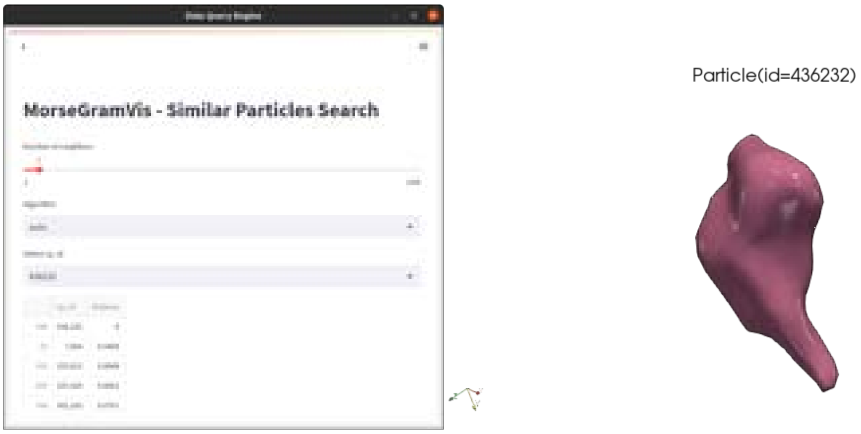
Figure 3.11: Shape descriptors

3.3.5.3 Similar Particle Search Feature

A search algorithm was employed to identify particles with nearly identical shape descriptors. This analysis aimed to find particles within the dataset that exhibit similar geometric characteristics, allowing for the identification of particle subsets with comparable properties and behaviour. An illustration of such interface is shown in the figure 3.12a along with reference particle is shown in the figure 3.12b.

The analysis involved comparing each particle's computed shape descriptors, such as elongation index, flatness index, sphericity, and compactness. By applying the k-NN algorithm, particles with shape descriptors that closely match or are within a specified threshold of similarity were identified. An illustration of the result of this feature is shown in the figure 3.13.

These data analysis techniques provide valuable insights into the properties and behaviour of the granular material, enabling a comprehensive understanding of its characteristics and facilitating further investigations in the field of granular materials research.



(a) Similar Particle Search Window

(b) Reference Particle

Figure 3.12: Similar Particle Search Example

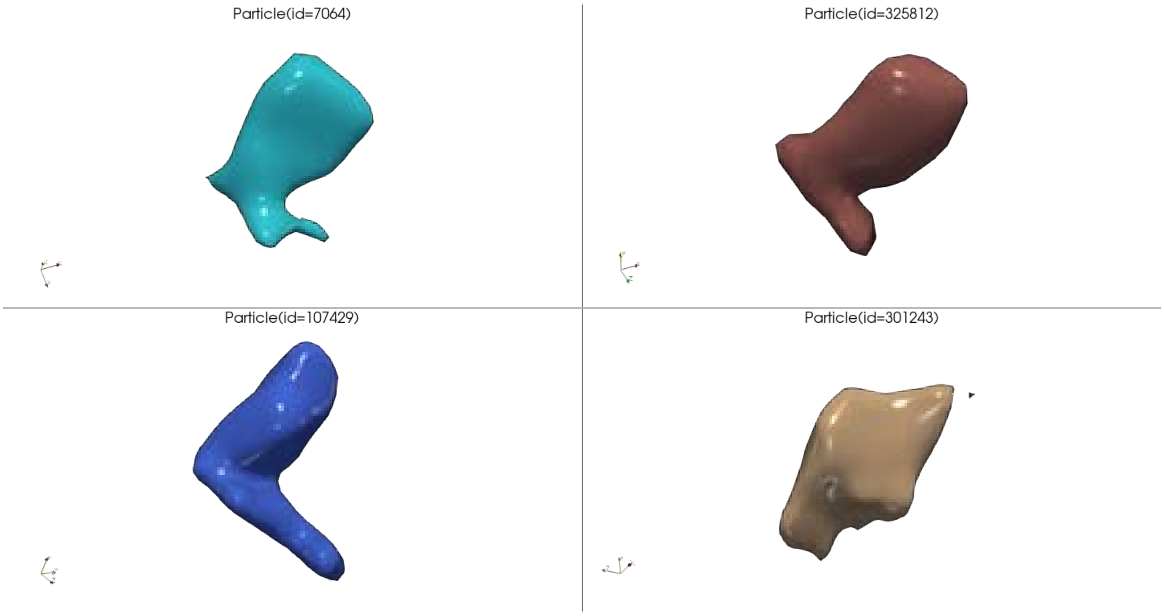


Figure 3.13: Particles similar to reference particle

3.3.6 Undersegemented Particle Detection

undersegmentation occurs when multiple particles are merged or grouped, leading to a loss of distinction between individual particles.

Various factors, including the complexity of particle shapes, overlapping boundaries, particle size or composition variations, and noise or artefacts in the image or data, can cause undersegmentation. It is particularly challenging to achieve accurate segmentation in granular materials due to particle arrangement's irregular and non-uniform nature.

The consequences of undersegmentation can be significant. It can lead to inaccurate measurements of particle properties such as size, shape, and spatial distribution. It can also affect the analysis of particle interactions, packing density, and other granular material characteristics. Understanding and addressing undersegmentation is crucial for obtaining reliable and meaningful results in studying granular materials.

Using metrics such as fabric tensor quantities or shape characteristics to comment on undersegmentation can be challenging due to the complex nature of granular materials. So, there is a need to explore other methods to address the issue of undersegmentation detection.

3.3.6.1 Graph Neural Networks(GNN)

Inspired by previous research[32], a dimensionless parameter called alpha is introduced to distinguish between actual contact and neck regions within granular particles. The computation of alpha is performed as follows:

$$\alpha = 1 - \frac{\text{scalar value of 2-saddle}}{\text{min(adjacent maxima's scalar value)}}$$

this alpha value is calculated for each triplet in the simplified saddle graph.

In this work, a graph neural network (GNN) approach is utilized to detect under-segmented particles. Two different approaches are employed:

1. Classification Model Trained on Simplified Saddle Graphs: A GNN-based classification model is trained using simplified saddle graphs. The model learns to distinguish between under-segmented particles and properly segmented particles based on the computed alpha values.

2. Classification Model Trained on Surface Point Clouds of Particles: Another GNN-based classification model is trained, but this time using the surface point clouds of particles. The model learns to identify under-segmented particles by analyzing the geometric properties and spatial relationships of the surface points.

By leveraging the power of GNNs and utilizing either the simplified saddle graphs or the surface point clouds, these learning-based approaches aim to effectively detect and classify under-segmented particles in granular materials. This provides a valuable tool for improving the accuracy and reliability of particle segmentation and analysis in various applications involving granular media.

3.4 Technology used

The Visualization and Analysis tool developed was implemented using the Python programming language [27]. Specifically, the Visualization Toolkit (VTK) [29] library was used for rendering the granular media, while PySide6 [35] was used for window handling and user interface development.

VTK is a widely-used open-source software library for the visualization of scientific data, and has a rich set of features for creating high-quality visualizations. It was well-suited for the development of the rendering tool due to its ability to handle large volumes of data and its robust rendering capabilities.

PySide6, on the other hand, is a Python binding of the Qt framework, which provides tools for the development of user interfaces. It was used in the development of the rendering tool to provide a user-friendly interface for researchers to interact with the tool and view the visualizations.

MeshLab [24] is an open-source, cross-platform software tool for processing and editing 3D triangular meshes. It provides many features and filters for mesh manipulation, cleaning, repair, visualization, and analysis. MeshLab supports various mesh file formats and offers a user-friendly interface for working with meshes.

Pymeshlab [21] is a Python library that provides a programming interface to access and utilize the functionality of MeshLab. It allows developers to integrate MeshLab's filters and operations into their own Python scripts or applications. Some notable filters in MeshLab and Pymeshlab include:

- Smoothing Filter (e.g., Taubin [33]): This filter applies a smoothing algorithm to the mesh, reducing noise and refining surface details while preserving important features. Taubin smoothing is known for its ability to preserve sharp edges and corners.

- Decimation Filter (e.g., Quadric [12]): The decimation filter reduces the number of triangles in the mesh while preserving its overall shape. The Quadric decimation algorithm simplifies the mesh by collapsing triangles based on geometric error, resulting in a lower-resolution mesh with fewer polygons.

These libraries were essential in the development of the tool, allowing for the creation of high-quality visualizations and an intuitive user interface for researchers to use.

3.5 Multiprocessing

Multiprocessing refers to the execution of multiple processes or tasks simultaneously on a computer system that has multiple processors or cores. It allows for parallel execution of tasks, thereby improving overall performance and efficiency.

During surface reconstruction, multiprocessing can be employed to leverage the computational power of multiple processors or cores to expedite the processing time. Surface reconstruction involves complex calculations and algorithms, which can be time-consuming, especially when dealing with large datasets or intricate geometries.

The surface reconstruction algorithm can distribute the workload across multiple processors or cores by multiprocessing, enabling concurrent execution of tasks. Each processor or core can independently handle some data, performing computations simultaneously. This parallel processing significantly reduces the overall computation time compared to sequential execution.

The need for multiprocessing during surface reconstruction arises due to the following reasons:

- Handling large datasets: Surface reconstruction of large datasets can be computationally demanding. With multiprocessing, the dataset can be divided into smaller chunks and processed concurrently, effectively utilizing the available computational resources and reducing memory constraints.
- Real-time or interactive applications: It is crucial to obtain results quickly in specific scenarios, such as interactive visualization or real-time analysis. Multiprocessing enables faster surface reconstruction, facilitating real-time or near-real-time processing and ensuring smooth user experiences.
- Utilizing available resources: Many modern computer systems have multiple processors or cores. Multiprocessing allows efficient utilization of these resources, ensuring that all available computational power is harnessed to expedite surface reconstruction tasks.

Overall, multiprocessing offers a significant advantage in terms of computational efficiency and improved performance during surface reconstruction. Parallelizing the computation across multiple processors or cores enables faster processing of large datasets and facilitates real-time or interactive applications.

3.6 Performance evaluation

System Details:

- **System**: Linux
- **Release**: 5.4.0-149-generic
- **Architecture**: x86_64
- **Processor**: x86_64

- **Python Version**: 3.8.10
- **Number of cores**: 16
- **CPU model**: Intel(R) Xeon(R) CPU E5-2650 0 @ 2.00GHz

Table 3.1: Performance Evaluation for Surface Reconstruction

Dataset	Resolution	Number of Particles	Time(VTK) (s)	Time(Voxel) (s)
ODO Dataset	153x153x324	1814	33.86	808.99
Steel Beads Dataset	181x176x251	1454	26.21	611.44
Chess Dataset	305x305x721	6988	384.65	5455.23

The table 3.1 presents the performance evaluation results for surface reconstruction using two different methods: VTK and voxel modelling. The dataset consists of three different samples: ODO Dataset, Steel Beads Dataset, and Chess Dataset.

The "Resolution" column indicates the dimensions of the dataset in terms of width, height, and depth. The "Number of Particles" column represents each dataset's total number of particles.

The following two columns, "Time (VTK)" and "Time (Voxel)," show the time taken for surface reconstruction using the VTK method and voxel modelling method, respectively. The time values are given in seconds.

From the table, we can observe that the VTK method generally performs faster than voxel modelling for all three datasets. The ODO Dataset has the lowest number of particles and the shortest reconstruction time for both methods. On the other hand, the Chess Dataset, with the highest resolution and a more significant number of particles, requires significantly more time for surface reconstruction using both methods.

Overall, the table compares the computational efficiency of VTK and voxel modelling techniques for surface reconstruction, showcasing the varying performance based on dataset characteristics.

3.6.1 Comparison between VTK and Voxel Modelling based Surface Reconstruction

When comparing VTK and voxel modelling-based surface reconstruction methods, one notable difference is how they handle void spaces within the granular media as shown in the figure 3.14.

VTK may produce triangles across void spaces, resulting in a surface representation that does not accurately represent the actual surface of the granular material. This can be problematic when analyzing or visualizing the material, especially in scenarios where void spaces play a significant role in the overall behaviour or properties of the media.

On the other hand, voxel modelling-based surface reconstruction methods tend to provide better results in such scenarios. Voxel modelling involves representing the granular media as a grid of volumetric elements (voxels) and reconstructing the surface based on the occupancy or presence of particles within each voxel. This approach can better capture the actual surface by accurately representing the boundaries between the granular material and void spaces.

By utilizing voxel modelling-based surface reconstruction techniques, researchers can obtain more accurate and reliable surface representations, which can be crucial for analyzing and understanding the behaviour of granular media, particularly in situations where void spaces are significant.

It is important to note that both VTK and voxel modelling-based approaches have their strengths and limitations, and the choice of method should be based on the specific requirements of the analysis or visualization task at hand.

Below image helps in understanding the above information.

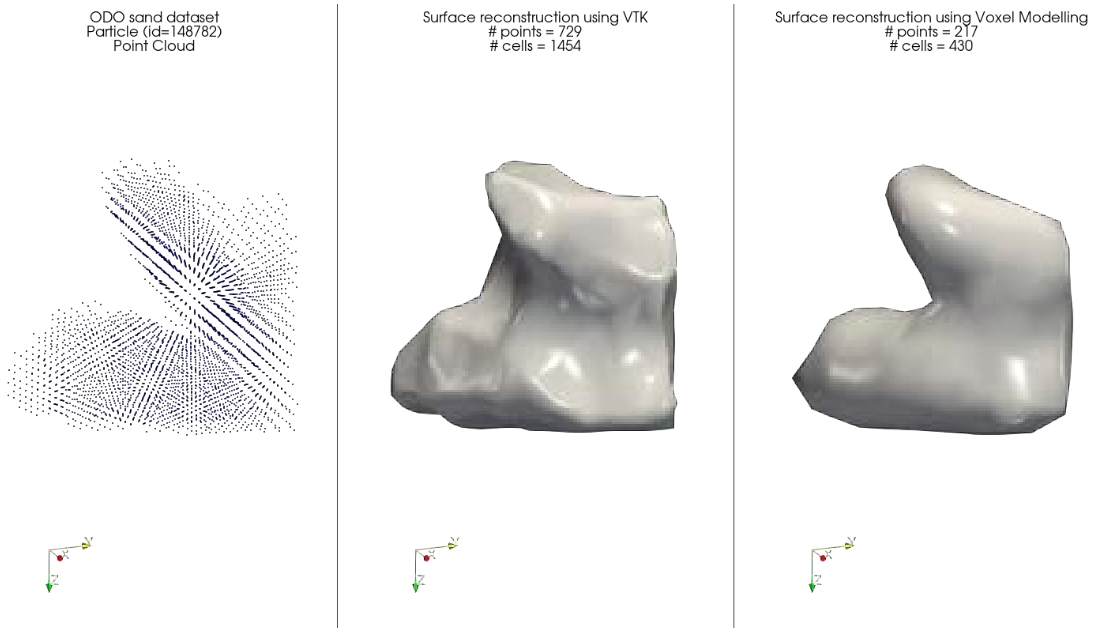


Figure 3.14: Surface Reconstruction Comparison

3.6.2 Limitation:

One limitation of surface reconstruction using VTK is its need for more robustness when dealing with large point cloud datasets. When the size of the point cloud increases significantly, it may not terminate within a reasonable time frame for large datasets. As the number of points increases, the computational complexity of the surface reconstruction algorithm grows, which can lead to significantly longer processing times.

Alternative surface reconstruction using voxel modelling does well and terminates within a reasonable time, thus making a preferable choice of surface reconstruction.

3.6.3 Scalability of Surface Reconstruction Methods

One crucial aspect to consider in the context of granular materials is scalability, which refers to the ability of a method or technique to handle more extensive and complex datasets efficiently.

Regarding scalability in terms of the number of particles, VTK demonstrates good performance. It can handle datasets with many particles without significant degradation in processing time. This makes it suitable for applications that analyze ensembles or systems with a high particle count.

However, VTK's scalability is limited regarding the particles' size or the point cloud's density. As the size of particles increases, or the point cloud becomes denser, VTK's surface reconstruction may encounter challenges in generating a proper mesh representation. This limitation can result in incomplete reconstructions and impact the system's overall performance.

On the other hand, voxel modelling offers scalability in both the number and size of particles. By representing the space as a grid of voxels, voxel modelling can efficiently handle large datasets with varying particle sizes. The use of voxels allows for flexible adaptation to different particle sizes and densities, making voxel modelling a scalable approach for granular material analysis.

With voxel modelling, the computational cost primarily depends on the resolution of the voxel grid rather than the size or density of individual particles. This enables voxel modelling to handle small and large particles effectively, making it suitable for analyzing granular materials with diverse particle characteristics.

Overall, while VTK provides scalability regarding the number of particles, voxel modelling offers scalability in the number and size of particles. The choice between the two approaches should consider the application's specific requirements, including the dataset size, particle characteristics, desired accuracy, and available computational resources.

Chapter 4

Results and Analysis

This research demonstrates the effectiveness of the developed visualization and data analysis features in understanding and interpreting the particle segmentation and particle surface reconstruction process. The visualization features allow users to comprehensively examine the segmentation results by providing multi-perspective views both at the ensemble and particle levels. Furthermore, data analysis is crucial in computing metrics, statistics, and other quantitative measures to gain insights from the segmented data. It enables researchers and practitioners to explore and understand the data's characteristics, properties, and relationships.

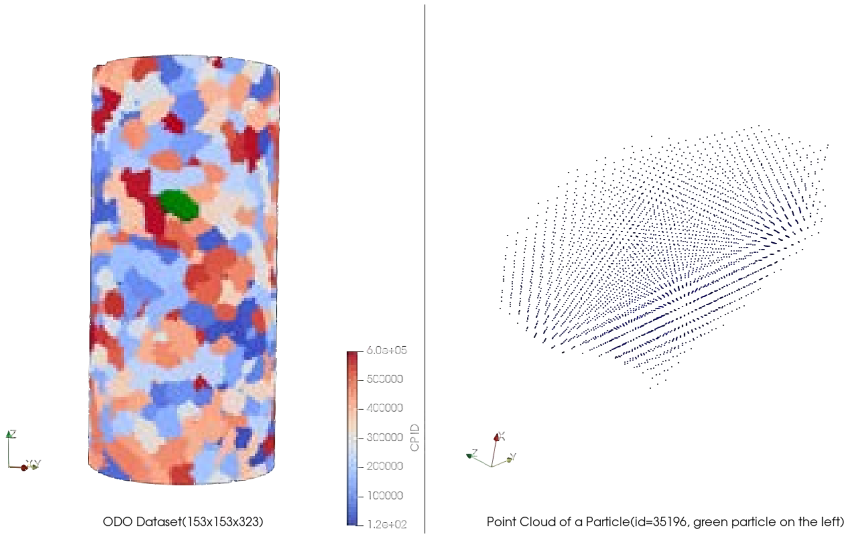
Researchers can obtain meaningful quantitative information about the segmented particles by applying various data analysis techniques, such as calculating particle size distribution, measuring particle shape parameters, or quantifying spatial arrangements. These metrics provide valuable insights into the particles' distribution, variability, and overall characteristics, allowing for a deeper understanding of the system under investigation.

Data exploration through a query engine is another essential aspect of data analysis. It involves exploring and visualizing the segmented data from different perspectives to uncover patterns, trends, and outliers. Through interactive visualizations, researchers can interact with the data, filter and aggregate information, and dynamically explore different subsets or aspects of the segmented particles.

Additionally, data analysis enables comparing different segmentation results or experimental conditions. By computing metrics and statistics across multiple datasets or variations, researchers can identify differences, similarities, or trends, providing valuable information for optimizing segmentation algorithms or studying the effects of various factors on the particle segmentation process.

Overall, the results and analysis of this research highlight the significance of visualization and data analysis features in gaining a comprehensive understanding of particle segmentation. These capabilities contribute to a comprehensive understanding of the segmented particles, aid decision-making processes, and drive further research and optimization efforts.

4.1 Overview of datasets used for testing



ODO Dataset(153x153x323)

Point Cloud of a Particle(id=35196, green particle on the left)

Figure 4.1: Labelled ensemble file

In the given figure(left) 4.1, the output of the segmentation process is a collection of 3D points. Each point represents a voxel or a small volume element/body centre of a voxel in the 3D space. These points are labelled with unique identifiers that signify the membership of each point to a specific particle.

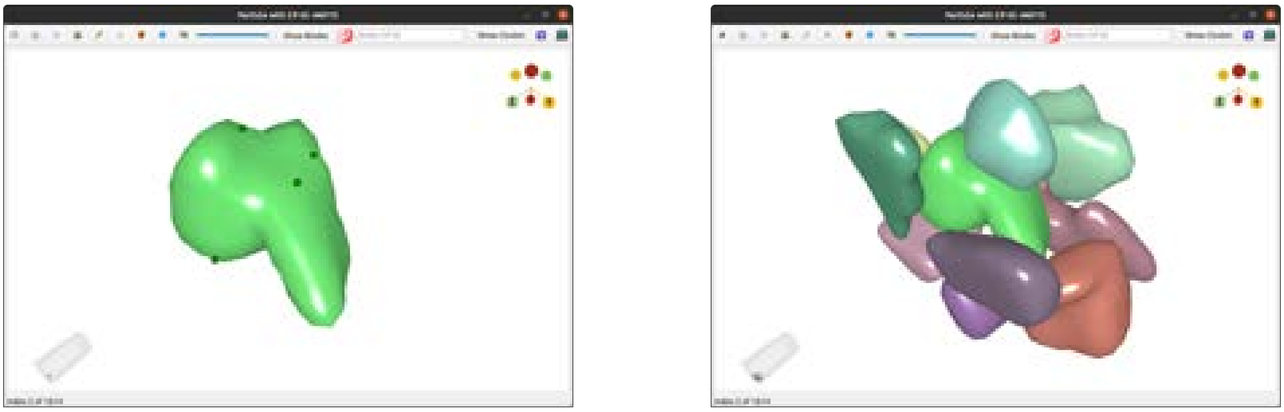
A particle refers to a distinct object or entity within the 3D space, such as a sand particle or a specific material. All the points belonging to the same particle are assigned the same unique identifier. This allows for easy identification and grouping of points that belong to the same object.

In the figure(left) 4.1, different colours are used to represent different particles. Each colour corresponds to a unique identifier, indicating separate particles within the scene. When zoomed in, a specific particle is highlighted in green, which is shown in the figure(right) 4.1.

Using these unique identifiers makes it possible to track and analyze individual particles within the segmented 3D data. This information is valuable for various purposes, such as studying particle interactions, measuring particle properties, or performing further analysis and simulations on specific particles of interest.

4.2 Visualization results

Figure 4.2: Ensemble View of ODO dataset

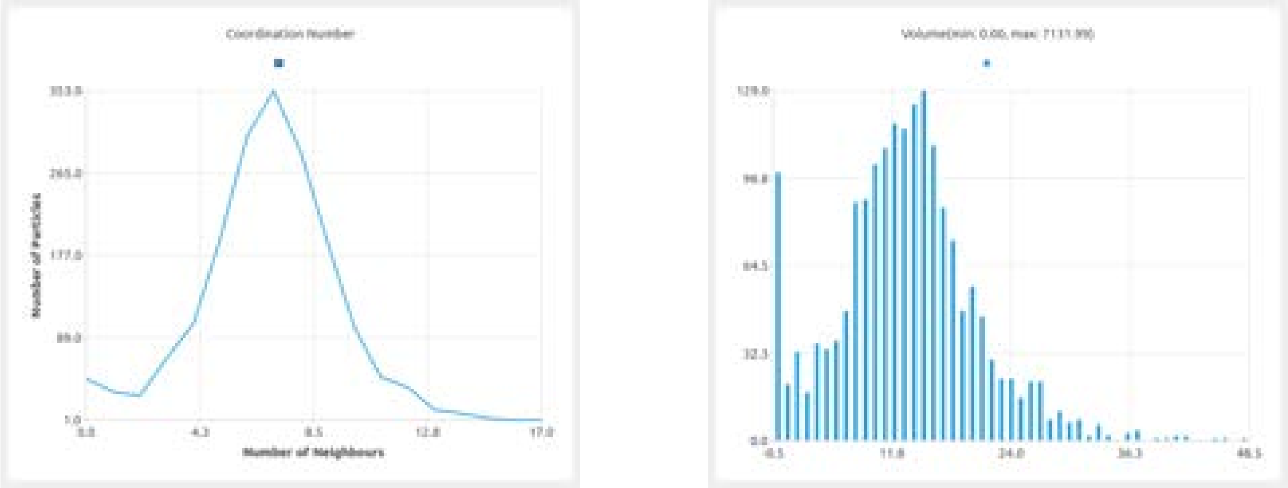


(a) Single Particle View

(b) Neighbours View

Figure 4.3: Single/Multi Particle Visualization of ODO dataset

4.3 Analysis of particle packing density, size distribution, and segregation



(a) Coordination Number graph

(b) Volume distribution of Particles(in voxels)

Figure 4.4: ODO Dataset

The coordination number refers to the number of distinct neighbours a particle has in a given system. It provides information about the local environment and connectivity of particles within a structure or material.

A coordination graph 4.4a can be created to analyze the coordination number distribution in a system. The coordination graph plots the coordination number on the y-axis and the number of particles with a specific coordination number on the x-axis.

The coordination number refers to the number of distinct neighbours that a particle has in a given system. It is a measure of the level of connectivity or interaction between particles.

In the coordination graph, the y-axis represents the coordination number, which can range from 0 to a specific maximum value depending on the system. The x-axis represents the number of particles in the system with a corresponding coordination number.

Analyzing the coordination graph can reveal necessary information about the system's clustering behaviour, particle packing, or spatial arrangement. As the system size changes, it can help identify patterns, trends, or anomalies in the coordination number. This knowledge can be helpful in various fields, such as materials science, chemistry, physics, and biology, where understanding the connectivity and interactions between particles is crucial for characterizing and predicting system behaviour.

The volume distribution graph 4.4b provides valuable insights into the distribution of particle volumes within a dataset. It plots the number of particles falling within different volume ranges (bins) on the y-axis, while the x-axis represents the bins.

This graph is significant in several ways:

- Identification of volume trends: By observing the distribution of particle volumes, you can identify any prominent trends or patterns. For example, suppose you notice a peak or cluster of particles in a specific volume range. In that case, it may indicate the presence of a distinct particle subgroup or a particular phenomenon related to volume.
- Error detection: The graph allows you to identify particles with volumes that deviate significantly from the expected distribution. Observing a disproportionately high or low number of particles in particular volume bins could indicate measurement errors, anomalies, or outliers in the data. These error particles with shallow volumes, as mentioned in the use case, can be identified and investigated further.
- Statistical analysis: The volume distribution graph provides a statistical overview of the dataset, allowing you to assess the overall distribution shape (e.g., normal, skewed, bimodal). This information is crucial for understanding the characteristics and behaviour of the particle system under study.
- Comparison and benchmarking: You can compare volume distributions across different datasets or experiments to identify variations or similarities. This helps benchmark and assess particle volumes' consistency under various conditions or experimental setups.

The volume distribution graph is a powerful tool for analyzing particle volumes, detecting errors or anomalies, and gaining insights into the underlying distribution characteristics. It aids in understanding the system behaviour and facilitates comparisons for research, quality control, and optimization purposes.

Chapter 5

Discussion and Conclusion

5.1 Summary of key findings

This work focuses on developing a comprehensive visualization tool for granular media, enabling users to interactively view ensemble and single particle-level data. The tool facilitates in-depth analysis of the granular media by computing shape metrics and fabric tensors to gain insights into their characteristics.

In addition, a query engine has been implemented, allowing users to explore and interact with the data flexibly and intuitively. This feature enhances the ability to extract valuable information from granular media and facilitates data-driven investigations.

Furthermore, a novel approach based on Graph Neural Networks has been employed to tackle the challenge of detection of under-segmented particles. This approach leverages the power of machine learning to enhance the segmentation accuracy and improve the understanding of the granular media microstructure.

Overall, this project provides a comprehensive set of tools and techniques for the visualization, analysis, and exploration of granular media, contributing to a better understanding of their properties and behaviour.

5.2 Conclusions and Future Work

In conclusion, this thesis presents a visualization tool that helps researchers to visualize and quantify the microstructure of segmented granular materials. The tool allows for interactive visualization and analysis of the microstructure in 3D. It also provides various quantification features, including particle statistics. The tool has been tested on different types of granular materials and has demonstrated the ability to accurately visualise and quantify the microstructure of these materials.

This section discusses potential avenues for future work and improvements to the current project:

- **Improvement in Mesh Surface Reconstruction of Point Cloud**
  Another area for future improvement is enhancing the mesh surface reconstruction of point clouds. This could involve exploring advanced algorithms or techniques to improve the accuracy and quality of the reconstructed surfaces. Additionally, optimizing the performance and efficiency of the reconstruction process can be a vital aspect to consider.

- **Development of Web Application and Android App**
  One possible direction for future work is developing a web application and an Android app based on the developed visualization and analysis tool. This would provide greater accessibility and convenience for researchers and users who prefer to work on web platforms or mobile devices.

- **Integration with Virtual Reality**
  Integrating the developed visualization and analysis tool with virtual reality (VR) technology is an exciting possibility for future work. This integration would allow researchers to immerse themselves in a virtual environment to interact with and visualize the granular media more effectively and intuitively. This can provide a unique perspective and enhance the understanding and analysis of the granular materials' microstructure.

- **Creating Dataset for Learning undersegmented Particles**
  To tackle the undersegmentation of granualr media, a valuable future direction is to create a dataset specifically designed for learning and training models to handle undersegmented particles. This dataset can include diverse examples of granular media with varying levels of undersegmentation, enabling the development and evaluation of machine learning-based approaches to address this issue.

Overall, these future directions aim to enhance the usability, accuracy, and capabilities of the developed tools and methods, paving the way for further advancements in visualizing and analyzing granular materials.

Chapter 6

User Experience

6.1 Introduction to User Experience Evaluation for the MorseGramVis

The user experience evaluation was conducted to assess the usability and effectiveness of the tool. The evaluation aimed to gather feedback from users to understand their experiences with the tool, identify strengths and areas for improvement, and guide further development. The evaluation aimed to measure user satisfaction, understand the ease of use and navigation, evaluate the rendering and visualization capabilities, assess the computation and analysis features, and gather suggestions for future enhancements.

The evaluation involved three participants with diverse backgrounds in geotechnical engineering, micromechanics of cohesive granular materials, and soil mechanics. Their expertise in these fields provided a comprehensive evaluation from different perspectives, contributing valuable insights to the research.

The evaluation followed a questionnaire-based methodology. Each participant was provided with structured questions related to their experience with the tool. The questionnaire covered various aspects such as general information, previous experience with similar tools, understanding of the tool's functionalities, user interface evaluation, rendering and visualization assessment, computation and analysis capabilities, performance and efficiency evaluation, suggestions for improvements, overall satisfaction rating, and recommendation.

The participants responded based on their individual experiences and perceptions of the tool. The questionnaire allowed them to rate different aspects on a scale, provide detailed explanations, and suggest additional features or improvements. Their responses provided valuable insights into the surface reconstruction tool's usability, performance, and potential enhancements.

The user experience evaluation aimed to gather subjective feedback from the participants, enabling the development team to understand the strengths and weaknesses of the tool. The evaluation outcomes would be used to inform future iterations of the tool, enhance its user-friendliness, address any performance issues, and incorporate suggested features to improve user experience.

6.2 Questionnaire

6.2.1 General Information

1. Name: ______________________________

2. Occupation/Field of Study: ______________________________

6.2.2 Experience with other tools

1. Have you used any surface reconstruction tools before?

2. If yes, please mention the tools you have used and briefly describe your experience with them.

3. If no, please proceed to the next question.

6.2.3 Introduction to the MorseGramVis

1. How would you rate the ease of understanding the tool's functionalities and features ?

2. Did you find the user interface intuitive and easy to navigate ?

3. Were you able to quickly grasp the purpose and potential applications of the tool ?

6.2.4 Rendering and Visualization

1. How would you rate the quality and clarity of the rendered visualizations ?

2. Did the tool provide sufficient options for ensemble visualization, single/multi-particle visualization, contact area visualization, and extremum graph visualization ?

6.2.5 Computation and Analysis

1. Did the mesh computation feature of the tool perform efficiently and accurately ?
2. How useful and informative did you find the data analysis capabilities of the tool ?
3. Were the multi-level segmentation functionalities effective in identifying and analyzing particle structures ?

6.2.6 Performance and Efficiency

1. How would you rate the overall speed and responsiveness of the tool during your usage ?
2. Did you encounter any significant delays or performance issues while using the tool?

6.2.7 Future Scope and Improvements

1. Based on your experience, what additional features or functionalities would you suggest for the tool? Are there any specific areas where you think the tool could be improved for better user experience or performance?

6.2.8 Overall Satisfaction

1. On a scale of 1 to 10, how satisfied are you with the tool overall ?
2. Would you recommend this tool to others in your field or related domains ?

Thank you for your participation! Your feedback and responses will be valuable for our thesis report and further development of the tool.

6.3 Analysis of User Experience Feedback

Based on the questionnaire results, here is an analysis of the user experience with the tool:

The questionnaire results provided valuable insights into the participants' experience with the MorseGramVis tool. Participants had prior experience with tools such as MATLAB, ParaView, and Tomoviz for surface reconstruction, visualization, and simulation purposes. When introduced to MorseGramVis, participants found the tool's functionalities and features easy to understand and navigate. The user interface was considered intuitive and user-friendly.

Regarding rendering and visualization, participants highly rated the quality and clarity of the rendered visualizations. They appreciated the options available for ensemble visualization, single/multi-particle visualization, contact area visualization, and extremum graph visualization.

In terms of computation and analysis, participants found the mesh computation feature accurate and efficient in capturing detailed particle shapes. They also found the data analysis capabilities of the tool useful and informative for extracting geometric properties and characterizing micro-scale properties.

Participants reported positive feedback on the performance and efficiency of the tool. They noted that the tool demonstrated good speed, responsiveness, and no significant delays or performance issues were encountered during usage.

For future improvements, participants provided valuable suggestions. These included automating the selection of under-segmented particles, adding options for exporting high-resolution image files, implementing linked camera views for comparative analysis, and providing a user manual or tutorial to enhance usability for users from different backgrounds.

Overall, participants expressed high levels of satisfaction with the MorseGramVis tool, rating it between 8 and 9 out of 10. They also indicated their willingness to recommend the tool to others in their respective fields.

The feedback received from the participants provides valuable insights for further enhancing the tool's usability, features, and overall user experience.

6.4 Conclusion

The user experience evaluation of the MorseGramVis yielded valuable insights from participants with varied backgrounds and experience with similar tools. Here are the key findings:

- **Ease of Use:** Participants found the tool's functionalities and features easy to understand and navigate. The user interface was generally intuitive, although some participants suggested more explicit instructions for first-time users.

- **Rendering and Visualization:** The quality and clarity of the rendered visualizations received positive feedback. The tool provided valuable options for ensemble visualization, single/multi-particle visualization, contact area visualization, and extremum graph visualization.

- **Computation and Analysis:** Participants were satisfied with the efficiency and accuracy of the mesh computation feature. The data analysis capabilities of the tool were considered valuable and informative, providing essential geometrical properties for post-processing.

- **Performance and Efficiency:** The tool exhibited good speed and responsiveness, with no significant delays or performance issues reported.
- **Suggestions for Improvement:** Participants suggested adding features like automatic selection of under-segmented particles, exporting visualized 3D views to high-resolution image formats, and providing a user manual for users from different backgrounds.

6.4.1 Overall Assessment

The user experience evaluation indicates that the MorseGramVis offers a positive user experience. Participants expressed high satisfaction with the tool's ease of use, rendering and visualization capabilities, computation and analysis features, and overall performance.

6.4.2 Implications and Potential Actions

The feedback received provides valuable insights for further improvements. Addressing the suggestions for adding new features, such as automated selection of under-segmented particles, and enhancing the user manual (added in appendix chapter, 7) can enhance the tool's usability for a broader range of users. Additionally, implementing the option to export visualized 3D views to high-resolution image formats would improve the tool's versatility.

It is recommended to incorporate the suggestions for improvement into future iterations of the tool, considering the needs and preferences of the target users. Regular updates based on user feedback and continuous testing can enhance the tool's usability and ensure a positive user experience.

Overall, the surface reconstruction tool has been well-received, providing a user-friendly interface, efficient computation, and valuable visualization and analysis capabilities. The positive user feedback and recommendations for future use indicate the tool's potential to effectively serve researchers and professionals in surface reconstruction and image segmentation.

Chapter 7

Appendix

7.1 User manual for the MorseGramVis

7.1.1 System Requirements

To run the software effectively, your system must meet the following requirements:

- Operating System: Ubuntu 20.04.6 LTS or compatible Linux distribution.
- Memory: 8 GB RAM or higher.

Please note that these requirements are specific to Ubuntu 20.04.6 LTS. The software may also be compatible with other Linux distributions, but it is tested on Ubuntu 20.04.6 LTS.

Before installing the software, ensure that your system meets these requirements. Failure to meet these requirements may result in degraded performance or compatibility issues.

7.1.2 Installation Instructions

To install and set up the visualization and analysis tool, follow these steps:

1. Create a virtual environment for Python using venv by running the following command in your terminal:

   ```
   python -m venv myenv
   ```

   This will create a new virtual environment named `myenv`.

2. Activate the virtual environment by running the appropriate command based on your operating system:

   - For Windows:

     ```
     myenv\Scripts\activate
     ```

- For macOS and Linux:

```
source myenv/bin/activate
```

3. Clone the repository from the GitHub repository by running the following command in your terminal:

```
git clone https://github.com/your/repository.git
```

4. Navigate to the cloned repository directory:

```
cd repository
```

5. Install the required dependencies by running the following command:

```
pip install -r requirements.txt
```

6. Finally, start the application by running:

```
python start.py
```

7.2 User Interfaces(UI)

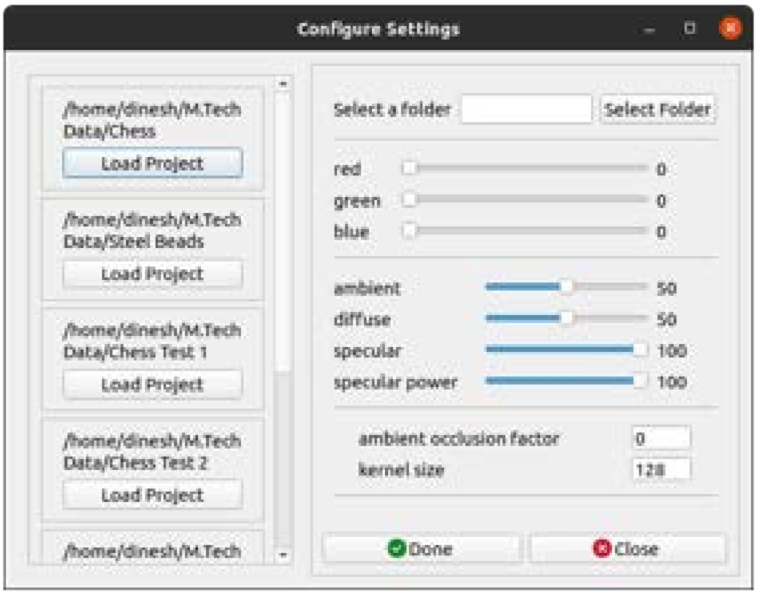
Figure 7.1: Start Window containing various features

7.2.1 Getting Started with Visualization

Here are the steps to follow for visualization:

1. Choose the dataset you want to work with by accessing the configure window 7.2.

2. Select the folder containing the dataset and adjust the parameters as needed.



Select a folder

red 0

green 0

blue 0

ambient 50

diffuse 50

specular 100

specular power 100

ambient occlusion factor 0

kernel size 128

Figure 7.2: Configure Window

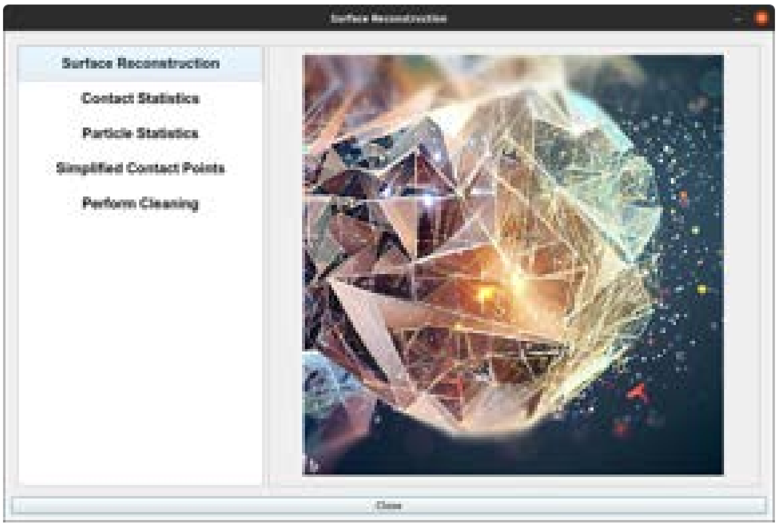


Figure 7.3: Surface Reconstruction Window

3. Open the surface reconstruction window 7.3.
4. From the navigation options, choose the surface reconstruction option.
5. Click the start button to begin the surface reconstruction process.
6. Once the surface reconstruction is completed, click the merge button.
7. After merging all the surface meshes, the ensemble is created.

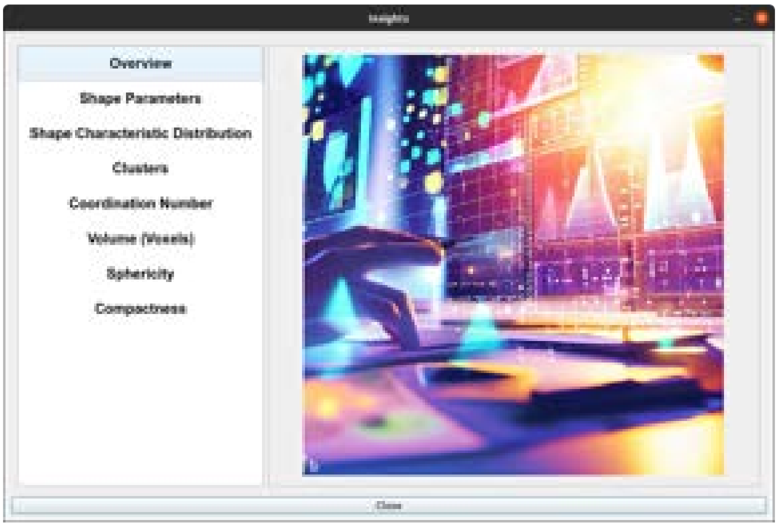


Figure 7.4: Insights Window

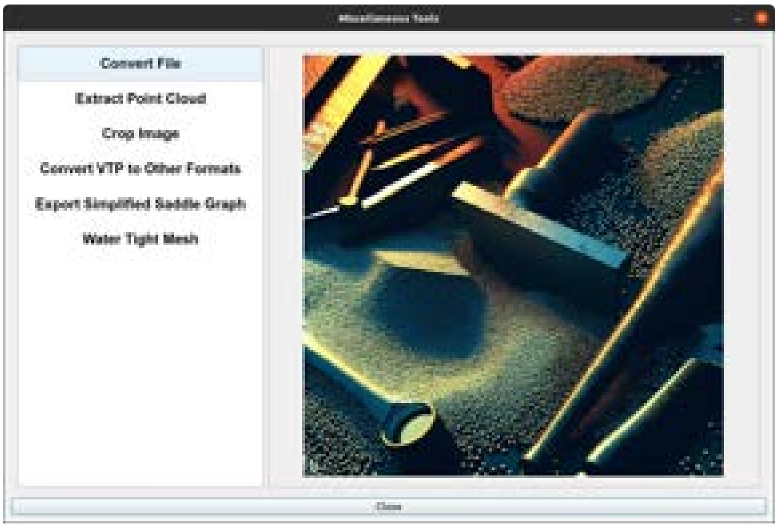


Figure 7.5: Miscellaneous Tools Window

8. Close the surface reconstruction window.
9. Finally, open the ensemble visualization window to view and analyze the ensemble or open the single/multi-particle visualization window to view and analyze at the particle level.

These steps will guide you through the process of performing ensemble visualization using the tool.

7.2.2 Getting rid of erroneous particles

To perform cleaning and remove erroneous particles, please follow these steps:

1. Open the Surface Reconstruction window 7.3.
2. From the navigation options on the left side of the window, select ”Perform Cleaning.”
3. Wait for the cleaning process to finish. This process may take some time, depending on the size and complexity of the dataset.
4. Once the cleaning process is complete, a volume histogram will be displayed. Please take note of the volume threshold or cutoff value and its units. All particles with a volume (measured in the number of voxels/ or points in the point cloud) less than the chosen cutoff are considered erroneous particles.
5. Provide the volume cutoff value in the input field present in the cleaning window.
6. Start the cleaning process by pressing the designated button.
7. Wait for the cleaning process to finish execution.
8. After the process is finished, all erroneous particle IDs are stored in the ”$data/clean_data.pkl$” file.
9. A new button will be enabled, indicating the option to create a clean and new ensemble file. Press this button to generate a new ensemble file that is free from erroneous particles.

By following these steps, you will successfully perform cleaning and remove erroneous particles from the dataset, ensuring a more accurate and reliable analysis.

7.2.3 Getting Started with Query Engine

Here are the steps to follow:

1. Make sure to perform surface reconstruction task 7.2.1 for all the particles in the dataset.
2. Once the project is configured, open the surface reconstruction window 7.3.
3. From the navigation options on the left, select ”Particle Statistics”.
4. check noisy particle checkbox to compute statistics for all particles including noisy too, else uncheck to ignore. Then, click on the ”Compute Particle Statistics” button.

5. Wait for the computation to complete.
6. After the computation is finished, close the surface reconstruction window.
7. Finally, open the Query Engine window by accessing the start window.

By following these steps, you will ensure that the surface reconstruction is performed for all particles in the dataset and then proceed to compute particle statistics. Once the computation is complete, you can close the surface reconstruction window and open the Query Engine window for further analysis and exploration of the data.

7.2.4 Getting Started with Insights / Analysis

please follow these steps:

1. Ensure that the previous task 7.2.3 is complete.
2. Open the Surface Reconstruction window 7.3.
3. From the navigation options on the left side of the window, select "Contact Statistics."
4. Wait for the process to finish, as it may take some time depending on the size of the dataset.
5. Once the surface reconstruction process is complete, close the Surface Reconstruction window.
6. Open the Insights/Analysis window.
7. In the Insights/Analysis window, you will find various plots and information related to the dataset based on the option chosen from the navigation options.

By exploring the different navigation options in the Insights/Analysis window, you can access valuable insights and analysis of the granular material dataset.

7.2.5 Visualization of contact region

Here are the steps to follow:

1. Ensure to perform surface reconstruction task 7.2.1 for all the particles in the dataset.
2. After configuring the project, open the surface reconstruction window 7.3.

3. Choose "Contact Statistics" in the navigation options on the left, then click the "Compute Contact Statistics" button.
4. Wait until the computation ends.
5. Finally, after finishing the computation, close the surface reconstruction window.
6. Now, the contact region visualization feature should be available for use.

By following these steps, you can ensure that the surface reconstruction is performed for all particles in the dataset. After configuring the project and opening the surface reconstruction window, choose "Contact Statistics" to compute the contact statistics. Once the computation is complete, close the surface reconstruction window. With the contact region visualization feature now enabled, you can proceed with visualizing and analyzing the contact regions in the granular material.

7.2.6 Update environment parameters like background color and lighting conditions

The given instructions outline a three-step process for adjusting the environment parameters through a setting dialog. Here is a summary of the steps:

1. Click on the "gear" icon located in the toolbar of the ensemble/single visualization window as displayed in the following figures 4.2 3.6 3.7 4.3.
2. A setting dialog will open, presenting various options and widgets to adjust the environment parameters. Use the appropriate widgets to make the desired adjustments.
3. After making the necessary changes, close the setting dialog to apply the adjusted environment parameters.

These steps provide a clear and concise guide for users to access and modify the environment parameters through the setting dialog.

7.2.7 Computing Simplified Saddle Graph

Here is a set of instructions to compute a simplified saddle graph:

1. Ensure that the surface reconstruction task 7.2.1 for all particles is complete.
2. Open the surface reconstruction window 7.3 and navigate to the options menu.

3. Choose the ”Simplified Saddle Graph” option from the navigation menu.
4. Click on the ”Start” button to initiate the computation process.
5. Wait for the process to finish, as it may take some time depending on the complexity of the data.
6. Once the process is complete, close the surface reconstruction window.
7. Open the single/multi-particle visualization window.
8. In the toolbar, locate and select the ”Show Simplified Saddle” icon.

Following these instructions will allow users to compute and visualize the simplified saddle graph for the surface reconstruction.

7.2.8 Manually labelling particles

To manually label particles, follow these steps:

1. Ensure that the surface reconstruction task 7.2.1 is complete.
2. Open the single/multi-particle visualization window.
3. By default, you will see an orange-colored sand particle icon in the toolbar, indicating that the particle is not labeled.
4. Press the ”c” key to label the particle as correctly segmented. The color of the icon discussed in the previous instruction will change to green.
5. Press the ”i” key to label the particle as not correctly segmented. The color of the sand particle icon in the toolbar will change to red.
6. All the label information will be stored in a pickle file located in the project folder under ”data/labels.pkl”.

By following these steps, you can create labeled particles, where the labels indicate whether the particles are correctly segmented or not. The pickle file will contain the necessary label information for further analysis.

7.2.9 Creating dataset for learning undersegmented particles

To create a dataset for learning undersegmented particles, follow these steps:

1. Ensure that you have completed labeling of particles 7.2.8.
2. Open the miscellaneous tools window 7.5 and select the "export simplified saddle graph" option from the navigation menu.
3. Depending on the type of classification model you are using, choose the folder containing the mesh or saddle graph.
4. Provide the location of the pickle file that contains the information on the labels of the particles.
5. Specify the target location for storing the correctly labeled particles and the not correctly labeled particles.
6. Click the "export" button to start the export process.
7. Wait for the process to finish. The dataset containing the undersegmented particles, along with their corresponding labels, will be saved in the specified target locations.

By following these steps, you will generate a dataset that can be used for training a classification model to identify undersegmented particles. The dataset will consist of the particles, their associated features (from the mesh or saddle graph), and their labels indicating whether they are correctly segmented or not.

7.2.10 Training Models for learning undersegmented particles

To train the Graph Neural Network (GNN) using the dataset generated from the previous steps, follow these instructions:

1. Make sure that you have completed the task 7.2.9.
2. Run the appropriate classification model implemented in the Jupyter notebook. Locate the notebook file in the "notebooks" folder of the software source code repository.
3. Ensure that the preprocessed data is stored in the "notebook/data/preprocessed" directory or the appropriate location specified by the notebook.

4. Start the training process by executing the notebook cells. The model will begin training on the preprocessed dataset.
5. Wait for the training process to complete. This may take some time depending on the complexity of the model and the size of the dataset.

During the training process, the GNN model will learn to classify undersegmented particles based on the provided features and labels. The duration of training will vary depending on the specific model architecture and the computational resources available. Once the training is finished, you can evaluate the performance of the trained model on test data or use it for predictions on new, unseen data.

Bibliography

[1] Tableau. `https://www.tableau.com`. Accessed on May 16, 2023. 6

[2] James Ahrens, Berk Geveci, and Charles Law. *Visualization Handbook*, chapter ParaView: An End-User Tool for Large Data Visualization, pages 717–731. Elsevier Inc., Burlington, MA, USA, 2005. URL `https://www.sciencedirect.com/book/9780123875822/visualization-handbook`. 7

[3] Ansys. `http://www.ansys.com/`, 2008. 9

[4] Mathieu Bastian, Sebastien Heymann, and Mathieu Jacomy. Gephi: An open source software for exploring and manipulating networks. 2009. URL `http://www.aaai.org/ocs/index.php/ICWSM/09/paper/view/154`. 7

[5] Michael Bostock, Vadim Ogievetsky, and Jeffrey Heer. D3.js. `https://d3js.org/`, 2011. Accessed: May 16, 2023. 6

[6] ME Cates, JP Wittmer, J-P Bouchaud, and Ph Claudin. Jamming, force chains, and fragile matter. *Physical review letters*, 81(9):1841, 1998. 9

[7] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022. 10

[8] Herbert Edelsbrunner, John Harer, and Afra Zomorodian. Hierarchical morse complexes for piecewise linear 2-manifolds. In *Proceedings of the seventeenth annual symposium on Computational geometry*, pages 70–79, 2001. 10, 11

[9] Herbert Edelsbrunner, John Harer, Vijay Natarajan, and Valerio Pascucci. Morse-smale complexes for piecewise linear 3-manifolds. In *Proceedings of the nineteenth annual symposium on Computational geometry*, pages 361–370, 2003. 11

[10] Matthieu Faessel and Dominique Jeulin. Segmentation of 3d microtomographic images of granular materials with the stochastic watershed. *Journal of microscopy*, 239(1):17–31, 2010. ii

[11] Abaqus FEA. `http://www.simulia.com/`, 2008. 9

[12] Michael Garland and Paul S Heckbert. Surface simplification using quadric error metrics. In *Proceedings of the 24th annual conference on Computer graphics and interactive techniques*, pages 209–216, 1997. 30

[13] Itasca Consultants GMBH. `http://www.itasca.de/`, 2008. 9

[14] J. D. Hunter. Matplotlib: A 2d graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007. doi: 10.1109/MCSE.2007.55. 7

[15] K Hutter and KR Rajagopal. On flows of granular materials. *Continuum Mechanics and Thermodynamics*, 6:81–139, 1994. 7

[16] Heinrich M Jaeger, Sidney R Nagel, and Robert P Behringer. Granular solids, liquids, and gases. *Reviews of modern physics*, 68(4):1259, 1996. 9

[17] Marc Levoy. Display of surfaces from volume data. *IEEE Computer graphics and Applications*, 8(3):29–37, 1988. 23

[18] William E Lorensen and Harvey E Cline. Marching cubes: A high resolution 3d surface construction algorithm. In *Seminal graphics: pioneering efforts that shaped the field*, pages 347–353. ACM, 1998. 24

[19] Holger A Meier, Michael Schlemmer, Christian Wagner, Andreas Kerren, Hans Hagen, Ellen Kuhl, and Paul Steinmann. Visualization of particle interactions in granular media. *IEEE transactions on visualization and computer graphics*, 14(5):1110–1125, 2008. 9

[20] IISc MorseGram, VGL. `https://bitbucket.org/vgl_iisc/morsegram/`, 2022. 5, 13

[21] Alessandro Muntoni and Paolo Cignoni. Pymeshlab, January 2021. 30

[22] Kazutaka Nakashima and Takeo Igarashi. Extraction of a smooth surface from voxels preserving sharp creases. In *ACM SIGGRAPH 2015 Posters*, pages 1–1. ACM, 2015. 17

[23] Karran Pandey, Talha Bin Masood, Saurabh Singh, Ingrid Hotz, Vijay Natarajan, and Tejas Murthy. Morse theory-based segmentation and fabric quantification of granular materials. *Granular Matter*, 24, 02 2022. doi: 10.1007/s10035-021-01182-7. ii, 13

[24] Guido Ranzuglia Paolo Cignoni, Alessandro Muntoni and Marco Callieri. Meshlab. 30

[25] Georgios Papaioannou, Maria Lida Menexi, and Charilaos Papadopoulos. Real-time volume-based ambient occlusion. *IEEE Transactions on Visualization and Computer Graphics*, 16(5):752–762, 2010. 21

[26] IISc pyms3d, VGL. `https://bitbucket.org/vgl_iisc/mscomplex-3d`, 2012. 5

[27] Python Software Foundation. Python. `https://www.python.org/`, 2021. Accessed: September 2021. 29

[28] Ville Satopaa, Jeannie Albrecht, David Irwin, and Barath Raghavan. Finding a" kneedle" in a haystack: Detecting knee points in system behavior. In *2011 31st international conference on distributed computing systems workshops*, pages 166–171. IEEE, 2011. 5

[29] Will Schroeder, Ken Martin, and Bill Lorensen. *The Visualization Toolkit (4th ed.)*. Kitware, 2006. ISBN 978-1-930934-19-1. 15, 29

[30] Nithin Shivashankar and Vijay Natarajan. Parallel computation of 3d morse-smale complexes. *Computer Graphics Forum*, 31(3pt1):965–974, 2012. doi: https://doi.org/10.1111/j.1467-8659.2012.03089.x. URL `https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8659.2012.03089.x`. 11

[31] Nithin Shivashankar, M Senthilnathan, and Vijay Natarajan. Parallel computation of 2d morse-smale complexes. *IEEE Transactions on Visualization and Computer Graphics*, 18 (10):1757–1770, 2011. 11

[32] Quan Sun, Junxing Zheng, and Cheng Li. Improved watershed analysis for segmenting contacting particles of coarse granular soils in volumetric images. *Powder Technology*, 356: 295–303, 2019. 29

[33] Gabriel Taubin. A signal processing approach to fair surface design. In *Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 351–358, 1995. 30

[34] Karl Terzaghi, Ralph B Peck, and Gholamreza Mesri. *Soil mechanics in engineering practice*. John wiley & sons, 1996. 8

[35] The Qt Company. Pyside6. `https://doc.qt.io/qtforpython/`, 2021. Accessed: September 2021. 29

[36] Hadley Wickham. *ggplot2: Elegant Graphics for Data Analysis*. Springer-Verlag New York, 2016. ISBN 978-3-319-24277-4. URL `https://ggplot2.tidyverse.org`. 6

[37] Budi Zhao and Jianfeng Wang. 3d quantitative shape analysis on form, roundness, and compactness with μct. *Powder technology*, 291:262–275, 2016. 26