

Institute for Visualisation and Interactive Systems

Masterarbeit

A Morphological Visualization Analysis of Porous Structures in Phase Inversion Processes

Kevin Kelly

Course of Study:	InfoTech
Examiner:	Prof. Dr. Thomas Ertl
Supervisor:	Patrick Gralka, M. Sc., Stefan Scheller, M. Sc., Dr.-Ing. Manuel Hopp-Hirschler
Commenced:	12 November 2018
Completed:	12 May 2019

Abstract

Porous membranes are an important technology that continually become more established as steps in research lead to its better understanding. Pore formation within the porous membranes is an emerging field, and current research is focused on understanding their development in phase inversion processes. This thesis attempts to aid in the analysis of these types of simulations by providing a visualization tool to extract information about the material and its pores.

A morphological analysis was implemented on the structure of three simulated data sets. The analysis consisted of extracting the pore network from the data and creating sets of visualizations to interpret the structure and properties of the material. A special focus with regard to shape of the pores was applied when creating these visual tools. The results showed a good agreement with an intuitive visual analysis of the input data set and the resulting output, and they resulted in new visual tools to better understand the structure of porous membrane development and point to promising work for the future.

Contents

1	Introduction	15
2	Porous Media	17
2.1	Definition	17
2.2	Classification	17
2.2.1	Pore Size	17
2.2.2	Pore Accessibility	18
2.2.3	Pore Shape	19
2.3	Material Properties	20
2.4	Applications	21
2.5	Pore Networks	22
2.5.1	Data Preparation	22
2.5.2	Maximal Balls	23
3	Related Work	25
3.1	Pore Networks	25
3.2	Pore Network Visualization	26
3.3	Pore Formation	27
3.4	MegaMol	28
4	Methods	29
4.1	Maximal Balls	29
4.1.1	Overview	29
4.1.2	Data Preparation	29
4.1.3	Grid Line Sampling	30
4.1.4	Distance Field	32
4.1.5	Maximal Balls Algorithm	38
4.2	Pore Tracking	43
4.3	Pore Statistics	45
4.4	Visualization	47
5	Results	49
5.1	Performance	49
5.2	Data Set Visualization	50
5.2.1	Full Data Set Visualizations	50
5.3	Pore Tracking Visualization	52
5.3.1	Complete Pore Tracking	52
5.3.2	Individual Pore Visualizations	54

6 Discussion	57
6.1 Analysis	57
6.1.1 Performance Analysis	57
6.1.2 Visual Analysis	57
6.2 Limitations	60
6.3 Future Work	62
6.3.1 Pore Network Efficiency	62
6.3.2 Shape Analysis	63
6.3.3 Visualization	64
7 Conclusion	65
Citations	67

List of Figures

2.1	Closed pores: (a) Open pores: (b) (c) (d) (e) (f) Blind (open at one end): (b) (f) Through pores (open at two ends): (e). Adapted from [RAF+94].	18
2.2	Pore Shape Classification. Adapted from [ZCSJ07].	19
2.3	Tensor Pore Classification. Adapted from [CSCS16].	20
4.1	A continuous shape, where the red dots show the resulting indices of grid line sampling.	31
4.2	A continuous shape with Grid Line Sampling in each direction.	32
4.3	An example of the input sets created for the continuous shape with grid line sampling specified for each axis in <i>Figure 4.2</i>	33
4.4	An example of the input set sizes based on the results from <i>Figure 4.2</i>	33
4.5	An example of the initial distances calculated for the continuous shape in <i>Figure 4.2a</i>	37
4.6	An example of the initial distances calculated for the continuous shape in <i>Figure 4.2b</i>	38
4.7	An example of the intermediate results of the overall distance transform calculated for the continuous shape in <i>Figure 4.1</i>	38
4.8	A continuous boundary, where all void voxels within the boundary show the radius calculated using the distance transform.	39
4.9	A continuous shape in 2D that has been processed according to the Maximal Balls algorithm. <i>4.9a</i> shows the first step, where the voxel in red has the largest radius and is processed first. All voxels within its radius are assigned as children, and one voxel is shown to highlight when a voxel is completely included in its parent pore and will no longer be processed. <i>4.9b</i> shows the next largest radius being processed and the voxels that are correspondingly assigned as children. <i>4.9c</i> shows the final result, including the pore centers and voxels that have been assigned as throats with the corresponding throat center.	41
4.10	A continuous shape in 2D that has been processed according to the Maximal Balls algorithm. <i>4.10a</i> shows the pore centers and voxels and throat center and voxels before the throat watershed algorithm. <i>4.10b</i> shows the results after processing.	42
4.11	The MegaMol Configurator interface showing the Maximal Balls module within the data writing pipeline in addition to the parameters of the module [GKM+15].	47
5.1	Performance of implementation for Data Set 1 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.	49
5.2	Performance of implementation for Data Set 2 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.	50
5.3	Performance of implementation for Data Set 3 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.	50
5.4	The MegaMol Configurator interface showing the visualization pipeline of the Maximal Balls output [GKM+15].	51

5.5	Initial visualizations of Data Set 1 over time. t corresponds to the time frame of the image.	52
5.6	Initial visualizations of Data Set 2 over time. t corresponds to the time frame of the image.	52
5.7	Initial visualizations of Data Set 3 set over time. t corresponds to the time frame of the image.	53
5.8	Full outputs of the Maximal Balls method for the final time frame of each data set.	53
5.9	Output of only pores and throats from the Maximal Balls method for the final time frame of each data set.	53
5.10	Pore visualizations of Data Set 1 over time. t corresponds to the time frame of the image.	54
5.11	Pore visualizations of Data Set 2 over time. t corresponds to the time frame of the image.	54
5.12	Pore visualizations of Data Set 3 set over time. t corresponds to the time frame of the image.	55
5.13	Visualization of pore tracking as pores develop between time frames.	55
5.14	Pore tracking of a randomly selected pore over time. t corresponds to the time frame of the image.	56
5.15	A comparison of a single pore result calculated according to the Maximal Balls method to a direct volume visualization of the input corresponding to the same area.	56
6.1	A visualization of a lamella pore being split over multiple time frames.	59
6.2	The histograms corresponding to Data Set 1. The x-axis corresponds to the bins and the y-axis corresponds to the time frame.	59
6.3	The histograms corresponding to Data Set 2. The x-axis corresponds to the bins and the y-axis corresponds to the time frame.	59
6.4	The histograms corresponding to Data Set 3. The x-axis corresponds to the bins and the y-axis corresponds to the time frame. Note: Due to an error in processing, only the longest to intermediate ratios are available.	60
6.5	The results of single pore tracking and volume mesh analysis show good agreement with the calculation of the Maximal Balls method.	61
6.6	Results near the boundary condition that lead to many small pores in what appears to be one pore.	61
6.7	An example of a boundary pore that has a high variance in assignment.	62

List of Tables

- 5.1 Performance results in minutes per submethod of Maximal Balls implementation. 51
- 5.2 Average results in minutes per time frame of Maximal Balls implementation. . . 51

List of Algorithms

4.1	Distance Field Data Set Arrangement	34
4.2	Distance Field Data Set Access	35
4.3	Distance Field Phase 1	36
4.4	Distance Field Phase 2	37
4.5	Maximal Balls Algorithm	40

List of Abbreviations

DPD Dissipative Particle Dynamics. 27

EDT Euclidean Distance Transform. 30

SDPD Smoothed Dissipative Particle Dynamics. 27

SPH Smoothed Particle Hydrodynamics. 27

XCT X-Ray Computed Tomography. 22

1 Introduction

Porous media is significant for many fields. Originally, porous membranes and membrane processes were used in research settings, such as chemical and biomedical laboratories [SGD06]. This period was the emergence of membrane technology and a time of discovery for the applications for which it could be used. In general, the applications were largely unknown and research showed promise of porous membranes as a future technology [SGD06].

Not long after, porous membranes were rapidly integrated into industrial products and methods, such as microfiltration, ultrafiltration, reverse osmosis, and gas separation [WDBF96]. The sudden increase in usability and application was a clear sign that porous membranes were an effective new tool and required more research to develop. The history of porous media quickly developed from being used primarily in a research setting to establishing porous membranes as a practically useful science, which led to the further discovery of its usefulness across industries [SGD06].

Today, porous media is a large focus in material science and other fields. In particular, technological developments have led to the ability to more accurately and efficiently represent and process porous media, both in high-resolution captures of real material and in computer modeling of simulated media [ARW05]. This has led to even further advances in understanding porous media and the ability to use it in more advanced processes. The progress of porous media is continually increasing and will likely continue to remain a target of research as more is learned and new steps enable further discoveries of its capabilities.

In addition to research in understanding porous material properties of existing material, further research is focused on understanding the formation process of porous material. This is because the modern level of understanding of the formation of porous membranes is mostly based on experimental observations [ZHHN18]. New research is beginning to focus on theoretically understanding and predicting the porous membrane development process based on applied physics [HHN18]. This will lead to results that are directly applicable to industry because the majority of porous membrane production in commercial use is based on phase inversion processes [HHN18].

This research is made possible due to the many previous advances in porous membrane research. These include the research in physical processes concerning thermodynamics and precipitation kinetics to determine pore formation, as well as research regarding the theories behind the formation of specific pore structures [HHN18]. Understanding the mechanisms behind these complex processes will allow the improved development of porous membranes by being able to control the production process. These advances form the foundation for current research and lead to new results with promising future applications.

The purpose of this thesis is to structurally analyze sets of porous material using results of new works that model the formation of porous morphologies [HHN18] [ZHHN18]. There will be a focus with respect to pore networks and pore shapes in order to determine their development over time. This will allow better analysis during their formation and will lead to aiding researchers to be

able to more accurately and efficiently analyze results of numerical simulations. The analysis of this thesis will focus on the visualization of the pore formation over time, while focusing on creating a visual representation of the shapes and statistics produced by the analysis.

2 Porous Media

2.1 Definition

Porous media, as a general definition, applies to a wide range of material. It is best defined as a granular material that is distinguished by two states. The first state is the solid component of the material, and the second is the empty or void space between the material [GVTA10]. The qualities of the two states as they relate to each other are of interest as it can define many properties about the material [SSA05].

The void space is, therefore, typically further divided into two categories to better define the material. The first set is the pores themselves, which consist of the larger spaces in the void space, and the second set are throats, which are the thinner restrictions that connect pores [GMH12]. Therefore, pores are a distribution of differently sized and shaped spaces within the material that are interconnected by throats.

The development of porous media has many applications across a wide range of fields. Their properties and characteristics led them to quickly go from a research topic to an important part of many industries. The versatility of porous media has led them to be used in many different settings, one of which may most importantly be to combat the increasing levels of carbon dioxide in the atmosphere.

2.2 Classification

2.2.1 Pore Size

The previous description of a porous material is very generic and applies to a wide range of materials. As such, this has led to many differing views on further classification of pores [ZZY00]. Classification of pores has received considerable attention in research, and for good reason. A porous material that contains the same porosity as another may behave very differently under similar conditions because of its pore sizes and distribution [ZCSJ07].

For instance, pore sizes can vary largely between and even within the same material. It is clearly important that a categorization of pores based on sizes exists and can be easily implemented to cover the wide range of possibilities. In a review of pore classification methods, Zdravkov et al. [ZCSJ07] created a summary of the most used pore size classifications. However, they found it difficult to identify a consistent classification because even the most common classification systems can have large variances in the ranges they use to classify a similarly named range [ZCSJ07].



Figure 2.1: Closed pores: (a)
Open pores: (b) (c) (d) (e) (f)
Blind (open at one end): (b) (f)
Through pores (open at two ends): (e). Adapted from [RAF+94].

For the work of this thesis, sizes of pores will simply be calculated according to their volume and will be compared to one another. While pores will not be classified based on size, it is important to note that the pores being processed are on a mesoscopic scale. The resolution of the data being processed is 15.6 *nm*.

2.2.2 Pore Accessibility

While a pore may be defined by its topology within a pore network, it can also be classified according to its surroundings relative to the material. *Figure 2.1* shows an example of a widely accepted classification of pores as they relate to their position within the material [RAF+94].

Closed pores are of interest when assessing mechanical properties of a material, but otherwise do not contribute to fluid flow and adsorption processes [RAF+94]. Open pores are accessible to the surroundings and are “availab[le] to an external fluid” [RAF+94]. Open pores are further subdivided into Blind and Through pores depending on the number of openings to the surroundings [RAF+94]. This classification is useful in determining pores’ relationships to each other as well as to the outer surroundings and how this can influence the behavior of the material under different conditions. Despite the importance, in general, of a universal pore size classification scheme to exist, it is not necessary in the scope of this thesis to categorize pores based on their accessibility.

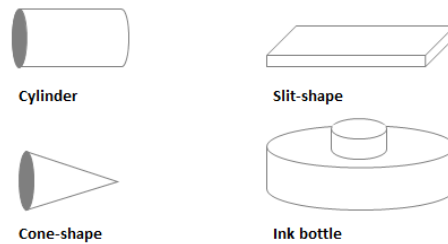


Figure 2.2: Pore Shape Classification. Adapted from [ZCSJ07].

2.2.3 Pore Shape

The shape of a pore can also vary widely, and is one of the most difficult parameters of a pore to describe [CSCS16]. In their review, Zdravkov et al. [ZCSJ07] determined a shape classification that was mostly similar compared to other previous works. This classification describes pores as belonging to any of five different categories and is shown in *Figure 2.2*. However, they also reference works that extend this list of shapes to better describe more specific geometries [ZCSJ07].

Claes et al. [CSCS16] offered a new classification system that breaks from previous works' use of pre-defined shapes. Instead, they propose classifying the shape of a pore based on the mechanical moment of each pore as a discrete object [CSCS16]. This results in a tensor that fully describes the extents of the pore and its orientation using an eigendecomposition of the tensor. In other words, each pore is described by an ellipsoid with axes corresponding to the lengths and direction of the longest axis of the pore in addition to the lengths and directions of the two axes orthogonal to it. This means that this classification focuses on shape parameters based on the pore geometry rather than rock textures and genesis as is the focus of most previous works such as A. Choquette and C. Pray [ACCP70].

From this flexible classification, pores can be described completely. It also allows to directly compute different shape parameters, which can be roughly categorized into rod like shapes, blade like shapes, cuboid like shapes, plate like shapes, and cube like shapes as seen in *Figure 2.3* [CSCS16]. In addition, Claes et al. [CSCS16] are able to use the orientation of the shapes within the structure when classifying a pore.

For the work in this thesis, the classification according to Claes et al. [CSCS16] will be used. The reason for using this classification is to take advantage of the authors' consideration of three-dimensionality when creating their classification. Many earlier classifications were largely based on two-dimensional slices of a porous media or on two-dimensional interpretations [CSCS16]. However, this classification uses five shape classes, rod, blade, plate, cuboid, and cube, that can be used at every resolution and provide information about the orientation of the pores [CSCS16]. This is ideal for pore geometries that have large differences in shapes and sizes of the pores and for the methods used in this thesis, as automatic classification across highly varying porous material types is desired.

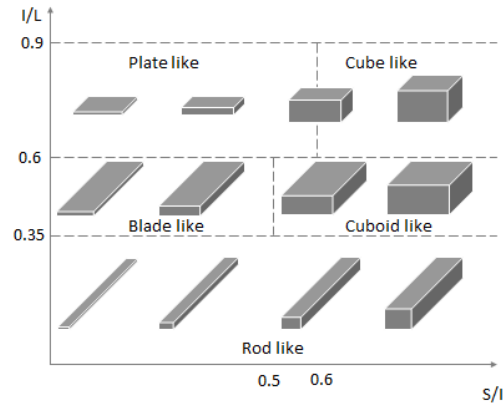


Figure 2.3: Tensor Pore Classification. Adapted from [CSCS16].

Rather than exactly using their terminology for classification, pores will generally be referred to in elliptical terms. Additionally, pore shapes will be discussed more generally as three class shapes to be consistent with the naming convention of Zander et al. [ZHHN18] and Hopp-Hirschler and Nieken [HHN18]. These are finger-like, sponge-like, and lamella-like pore shapes. Each of these classes can be compared to the shapes as adapted from Claes et al. [CSCS16] in *Figure 2.3*. As their name implies, finger-like pores are long, skinny pores where one dimension is larger and the remaining two are similar in size. This corresponds to the rod like pores. Sponge-like pores have a similar range of length in each dimension and correspond to cuboid or cube like pores. Lastly, lamella pores are planar pores that have two dimensions that are larger than the third. This corresponds to the remaining blade and plate like pores.

Pore shape is an important part of the analysis of this thesis, and therefore describing and categorizing pores into these categories will be of importance.

2.3 Material Properties

While porous media differs widely in material and application due to differences in pore types, sizes, and distributions, membranes have become widely adopted due to certain shared characteristics. These most important characteristics are that using membranes as a form of separation occurs at ambient temperatures and does not require chemically altering the materials being subjected to the porous media, properties of the porous material can be changed according to the desired use, membrane processes tend to require less energy than other separation techniques, and membrane processes can be used equally well in large scale applications or in smaller use cases [SGD06]. These characteristics have shown porous media to be useful across many fields and applications. Therefore, in general, a significant amount of research is focused on determining the properties of a material and how to interpret them.

One important property of porous materials is its porosity. Porosity is simply the ratio of the volume of the void space of the material compared to the overall volume [Nim13]. Therefore, porosity represents the amount of space within the material and how much of a material is available to hold liquid.

Another important property of porous materials is its permeability. Permeability is “the capacity of a reservoir to deliver hydrocarbons” [RJK06]. In other words, it is the ability of a porous material to transmit fluids.

Permeability is a difficult parameter to measure within porous media, despite its importance as a measure of porous media [GMH12]. The relationship between porosity, permeability, and pore and throat shapes and sizes has been of great interest in research with many works using different methods to find the appropriate relationship [GMH12]. In particular, Rezaee et al. [RJK06] were able to show that the best predictor for determining permeability in carbonate samples is directly related to median throat radius. This means that determining the throats of a material based on the pore network is sufficient to predict a very important property of porous materials.

In addition, pore networks have been shown to predict other important factors and properties related to morphology, thermodynamics, and kinetics [HHN18]. It is clear that pore networks are extremely valuable in characterizing a material by determining and predicting many different properties.

For this reason developing pore networks for a material can help to understand its properties. Further, as is the goal of this thesis, obtaining this information can help to affirm or reject expected outcomes for the experimental formation of porous material. In either case, it helps to gain a better overall understanding of the process.

2.4 Applications

Porous media is generically useful across several disciplines due to its versatility from the common characteristics defined previously. For example, in construction, porosity is important in determining the weight of materials, as well as other properties such as drainage capacity and their degradation rate. Material science, in general, is concerned with factors such as internal erosion that can weaken the structural integrity of a material [HBW+13].

In biology and medicine, porosity is an important indicator for bones and is capable of determining medical diagnoses [GVTA10]. Porous membranes are, after all, a natural occurring material that has existed long before their scientific development, whether in biology or for example the use of charcoal as a medical tool [ZCSJ07]. Further in the medical field, porous membranes are essential for developing artificial organs and drug delivery systems due to their ability to operate at body temperature “without chemically altering the constituents of a mixture” [SGD06].

Radioactive waste management is another pressing issue that benefits greatly from the use of porous media due to its sorption affinity [XBJ16]. In addition, several materials that act as barriers for nuclear waste management are porous, and their rate of degradation must be able to be predicted to ensure the safety of operation [JO12].

One specific application of porous media that has grown in interest recently is storing carbon dioxide in subsurface geologic formations. Due to the increasing levels of carbon dioxide in the earth’s atmosphere and the corresponding destructive effects on the climate, there is a great

focus on decreasing the levels of atmospheric carbon dioxide [UMW+12]. One way this has been accomplished is by injecting carbon dioxide into water-saturated porous sandstone. While this has been effective and requires low energy, research has been focused on ensuring the safety of this method and identifying the properties of potential formations [ZFS+18]. The further research and development of geologic sequestration of carbon dioxide is an important application of porous media that can have a significant impact in the reduction of increasing atmospheric carbon dioxide levels.

2.5 Pore Networks

The morphology of the pores and throats can be modeled as a network throughout the material that can be used to understand the connections and distribution of pores and their sizes, as well as be used to determine certain properties of the material [ARW05]. The qualities of porous media, therefore, are highly dependent upon the internal structure of the material. Features such as pore type, size, and orientation, as well as the interconnection of pores and throats are all extremely important in determining the characteristics of a porous material [JHE+13]. Such properties of pores that can be determined include thermal, electrical, and acoustic characteristics of the material [JWC+07].

To determine pore networks, several methods exist. While other approaches for generating pore networks exist [RH10], these methods can be roughly sorted into two main methods that have been established using medial axis methods or morphological methods based on the geometry of the material [SSA05]. Each uses its own methods to determine and categorize the pores and throats contained in the material.

2.5.1 Data Preparation

In any case, either pore network method must first process the data used for the image analysis. Typically, the type of data falls into two categories.

The first category is images of real material. For the analysis of porous media, these are usually high-resolution X-Ray Computed Tomography (XCT) scans in three dimensions that typically range from micro to nano scales [DB09]. Advances in image capture technology allow extremely high-resolution images that can be directly captured and used to analyze real material in a non-destructive way [SSA05]. Using real images allows for analyzing complex and unique structures.

The alternative to real XCT scans is using simulations of material. Using computer reconstructions typically allows for quicker realizations of test data and allows for testing more generalized criteria, which allows more comprehensive results that can be conceptualized for a broader range of situations [ARTW03].

The data used in this thesis comes from computer simulations. More specifically, the data is the result of a numerical precipitation experiment, wherein the porous material is produced using a phase separation process [ZHHN18]. Further, during the simulation of the formation of the pores, [ZHHN18] introduce a moving front to orientate the fluid phases. The result is an unstable material

that forms pores over time. The goal of this thesis is to analyze the development of the pore network over time for this data set. In addition, an analysis of the shapes of pores over time will be carried out.

2.5.2 Maximal Balls

As mentioned above, pore network extraction is typically split between medial axis methods and morphological methods. The prior methods function by calculating the medial axis transform to reduce the three-dimensional pores to its skeleton of maximal distances from the walls of the void space [DB09]. The pore network is extracted from the medial axis method by assigning pores to the connection points of the skeleton and assigning the connections and local minima as throats [SSA05]. An importance of this method, is a focus on the appropriate analysis and merging functions that come with post-processing of the medial axis result to achieve proper results for pore and throat assignment [SSA05]. This method, while intuitive and clearly defined, results in inconsistent outcomes [SP06]. In addition, it can result in connections that geometrically do not exist [SSA05].

The main morphological method focused on the geometry of the void space is the Maximal Balls method. This method functions by calculating for each voxel located within the void space, the maximal inscribed sphere that is centered at that voxel and its radius extends to the boundary between material and void space [DB09]. In the classical Maximal Balls methods, this calculation was performed by inflating and deflating spheres until the maximum radius was found [SP06]. Also in the classical Maximal Balls all spheres are processed to determine a hierarchy within the void space [AH17]. This hierarchy is then reduced using a depth-first search and it is then possible identify the pores and throats accordingly [AH17].

The main issue with this approach are that it is sensitive to noise, as fluctuations in the definition of the boundary can result in large differences during the calculation of the spheres, particularly if they are rounded to the nearest integer grid [SSA05]. As the issue with using a morphological approach is that it is susceptible to noise, there is no concern with using such an approach here. The data used in this thesis comes from simulations and, therefore, contains exact results. In addition, medial-axis methods result in a direct mapping of the pore network, and is more focused with the topology of the pore network and less concerned with the shape of the pores [DB09]. As one of the goals of this thesis is to determine shapes of pores over time, the Maximal Balls method was chosen to better implement those goals.

A known issues with pore network models is that they are inflexible when it comes to adaptations in the network [SHCZ18]. This is because pore network models assume a static shape during analysis. This comes from the extent of research mostly focusing on single images that come from either statistical simulations or real samples that can be imaged using XCT [AKB07]. However, these types of works do not consider the effects that can take place in a porous medium over time. These effects can include deformations of the material due to stress to causes changes in the properties of the material [SHCZ18]. A novel approach of this thesis is to efficiently apply a network analysis technique over a range of time in a changing environment. This will allow analysis of porous material while regarding changes in the network.

The examples throughout *Chapter 2* clearly show the wide range of applications of porous media. The development of porous membranes and the resulting research over time has led to significant advances in understanding its properties and abilities. Due to an increased understanding of pore networks and their implications, porous media has been able to be implemented in a variety of fields and application. Research is beginning to focus more on the formation of porous media to understand how to create desired properties [HHN18]. Instead of being used solely as an analysis of materials, membrane research is beginning to focus more on the development of porous materials to improve membranes for manufacturing [KHS+14]. This thesis attempts to add to the understanding of porous material and its development in phase inversion processes.

3 Related Work

The following related work is relevant to the work of this thesis. Previous work in pore networks was critical in determining the appropriate method to use for this thesis, as well as finding a method that is efficient enough to handle processing across multiple time frames. The work in pore formation is what directly led to the work of this thesis. The data used in the methods of this thesis comes from the results of these works. All processing and visualizations produced in this thesis were carried out using the MegaMol framework [GKM+15].

3.1 Pore Networks

Pore networks' significance in classifying porous materials has been understood as a useful tool in determining properties of a material, and has therefore been researched to better analyze pore networks. Al-Raoush et al. [ARTW03] developed an analysis of two network generation techniques that were state-of-the-art for the time in 2003. Both networks were able to extract the network structure correctly for regular packed structures, but struggled with handling random packings [ARTW03].

The method implemented by Sheppard et al. [SSA05] is an example of an advanced medial axis technique for its time that implemented elegant post-processing to determine appropriate merging and morphology. Their implementation was also able to be fully parallelized to allow up to one million nodes to be generated on micro-XCT images. Overall, their results were promising, but they did notice issues with robustness when comparing differing networks between two similar images [SSA05].

Dong and Blunt [DB09] were able to implement a Maximal Balls method on micro-XCT images in 2009 based on the work by Silin and Patzek [SP06] and extended to handle the input. They achieved good agreement in accordance with benchmark tests and performed a study on andstone, carbonate, and sand pack samples [SP06].

The work by Jivkov et al. [JHE+13] proposed a novel solution by arguing that a truncated octahedral is the best representation of a pore on average. By experimenting on XCT images of limestone samples, they were able to show not only accurate permeability measures, but also “a strategy for calculating the evolution of permeability with damage-related pore space changes” [JHE+13], where they were able to efficiently calculate mechanical properties of the material over the simulated process of degradation. While this method considered the structure of a pore network beyond a single point in time, it was more important for this thesis to completely recalculate pore shapes than calculate pore properties based on changes.

Homberg et al. [HBW+13] presented a novel solution of determining the pore structure for micro-XCT images that considered previous works, but implemented a unique solution. Their process was to define elements based on a distance transform with a calculation of critical points, a voxel-based process to determine the network skeleton, and a merging algorithm to avoid high space fragmentation.

The work by Arand and Hesser [AH17], which is the work followed for the methodology of this thesis, is an excellent implementation of the Maximal Balls method. As will be outlined throughout *Chapter 4*, Arand and Hesser [AH17] were able to meaningfully improve on previous methods in terms of both efficiency and memory, in addition to improving accuracy. Due to improvements in every aspect, this method was chosen to be implemented for the pore network creation section of this thesis.

3.2 Pore Network Visualization

While pore networks have received considerable attention for their ability to determine pore properties, it was found that pore network visualization has received noticeably less attention during the literature review for this thesis. This could show that most researchers are mainly concerned with finding properties of porous material and are not concerned as much with a visual analysis. Due to the goal of increasing understanding of complex data and aiding in researchers to better analyze simulated data, this thesis makes an effort to create visualizations to display the results.

Despite lacking an overarching focus on visualizations, some works display good results of visualizations of pore networks. For instance, Al-Raoush et al. [ARTW03] generated images of the medial axis skeleton and the skeleton combined with the locations of pores. Sheppard et al. [SSA05] created visualizations of networks for their medial axis method. While the results are good for their sandstone images, there is a high amount of clutter in their network for limestone. Despite that, their use of abstraction to visualize pores simply as spheres and the connecting throats as cylinders is easy to understand the topology of the network.

Silin and Patzek [SP06] provide interesting visualizations of multiple types. The first is simply the pore space as calculated by their method. While this does not inherently provide any topological information, it is interesting to see the complete difference between the material and void space. They also provide abstracted visualizations called stick-and-ball diagrams using the centers of the pores as spheres with lines connecting them to display the topology of their network. Overall, they provide a considerable amount of images for visualization results.

The work by Grau et al. [GVTA10] provide a full explorative study of porous structures and provides many excellent visualizations and methods for producing them. For example, they provide a pore network abstraction visualization with the option to select factors such as specific pores and/or volume ranges and accordingly use a combination of color highlighting and opacity to properly visualize the results of interest while to an extent avoiding occlusion. In addition to abstracting the network by mapping pores to a physical 3D location and connections using cylinders, they also completely abstract the network to a bubble graph to statistically visualize the topology of the network. There were also attempts to give a sense of depth to a pore network visualization by mapping the depth within the image to a color. This particular choice of visualization must be criticized, however, for its use of a rainbow color map. A better selected color map to map color

more linearly to perception may have produced good results, but otherwise does not have a great effect in creating visual depth. However, later on they use an opacity ramp within a visualization that creates a strong sense of depth, and would likely be very effective with a proper color map. Their work further visualizes other exploratory visualizations that are overall interesting and well-suited for researching the effects of different visualizations in pore network analysis [GVTA10].

3.3 Pore Formation

As stated above, the simulation data and motivation for this thesis comes from [HHN18] and [ZHHN18]. These methods are both state-of-the-art research in understanding and predicting the morphological formation of pores. They both develop physical models to simulate the pore formation process using a phase inversion process that accurately reflects real-world dynamics.

In the work by [HHN18], they propose a prediction for the formation of porous structures based on density-gradient theory. They give an in depth view of the theory behind membrane formation, which is based on the interactions of thermodynamics, before simplifying the models to reduce computational complexity. The method is implemented using Smoothed Particle Hydrodynamics (SPH) to solve the partial differential equations, where SPH is a hydrodynamic method developed in 1977 as a mesh-free computational method for simulating particle motion using Newtonian equations of motion [GM77]. SPH is recognized as a useful tool in simulation science with a wide range of applications due to its ability to focus on molecular interactions while retaining properties of the macroscopic equations [Mon12]. They are able to show, for a simulation domain consisting of a reduced binary fluid mixture driven into a symmetric miscibility gap, that phase separation with an induced moving front can predict the formation of pores. Their results show that they are able to predict the formation of certain shapes of pores based on diffusive mass transport.

In the work by Zander et al. [ZHHN18], they use a similar approach by applying Smoothed Dissipative Particle Dynamics (SDPD) to also study the formation of pores based on a moving front in a phase separation process. SDPD is a method combining SPH with Dissipative Particle Dynamics (DPD), a method for simulating the properties of complex fluids that is well-suited for operations at the mesoscopic scale [ER03]. The work by Zander et al. [ZHHN18] makes use of SDPD in combination with thermal fluctuation modeling to extend previous works that did not consider thermal noise. Their results in including thermal fluctuations to their model results in good agreements with analytical results and the development of different morphological structures.

These two works on pore formation are the basis for this thesis, and provide all simulation data that is processed. They provide a solid platform for the development of a visualization tool that can calculate the shapes of pores over time.

3.4 MegaMol

MegaMol is a “cross-platform visualization prototyping framework” [GKM+15] created by the Visualization Research Center of the University of Stuttgart and is the platform on which this thesis was created. MegaMol is a middleware used to visualize point-based data sets and is an open source project available on GitHub [GKM+15]. MegaMol was installed and setup on a Windows machine, and this thesis was developed as a standalone plugin using Microsoft Visual Studio.

The MegaMol architecture consists primarily of Modules and Calls. Modules are the functional components of the framework and provide access points to other modules. The Maximal Balls method implemented for this thesis is implemented as a module in MegaMol that provides two access points through the form of a Call. A Module uses a Call as an access point to allow that Module to interact with other Modules. This loose coupling improves modularity within the architecture and makes an application easier to understand, develop, and test. A Module either exposes its functionality as an available Call or requests functionality from other Modules as a Call. The maximal balls Module had one Call to request the necessary data from downstream and one Call to offer the resulting network to a Module upstream that could write the data to a file [GKM+15].

The other important function of MegaMol for the maximal balls Module is the use of parameters. The use of parameters allows a user to adjust input variables to modules before each simulation run. For example, the Maximal Balls module implementation can easily change modes to track individual pores over time or output various statistics. The MegaMol framework is the foundation for the implementation of this thesis.

4 Methods

4.1 Maximal Balls

4.1.1 Overview

The Maximal Balls method is a robust algorithm for extracting pore networks from porous materials. More specifically, it is used to extract pore and throat centers and sizes, as well as the connections of each throughout the entire material. Previously, thinning algorithms were the primary algorithm for extracting information from porous materials. These are the basis of the methods that make up the medial axis classification of pore network techniques [AH17]. However, the Maximal Balls method was first introduced by Silin and Patzek [SP06] to extract the network of pores and throats from tomographic images more robustly. For example, thinning methods are known to have inconsistent results, even on the same data sets. The results of Maximal Balls were promising and led to much more research in improving the method. As research has led to significant increases in processing times of the method, it is now one of the two most used methods for extraction of pore network information [SHCZ18].

As several variations exist for the implementation of the Maximal Balls algorithm, the method according to Arand and Hesser [AH17] was chosen to be used in this thesis. This decision was made after a thorough literature review of possible methods. The methodology, data structures, and detailed concepts of this implementation are specifically improved upon compared to previous methods in terms of accuracy and efficiency. As the methodology of the Maximal Balls algorithm is discussed in more detail, all naming conventions will follow those chosen by [AH17]. The implementation of this method for this thesis is discussed in more detail as follows.

4.1.2 Data Preparation

The data used is a volumetric data set in a Cartesian grid consisting of $256 \times 64 \times 64$ voxels with a cell size of $15.6nm$. The resolution was found from the data set as the calculation of the average distance between voxels in each direction. Each voxel consists of voxel coordinates $\mathbf{x}^i \in \mathbb{Z}^3$ with voxel values

$$f^i = f(\mathbf{x}^i), f^i \in \mathbb{R}.$$

Following the Maximal Balls method according to Arand and Hesser [AH17], the data was prepared by initially separating it into two partitions called void space and material, where void voxels are defined as

$$\mathbf{x}_{void}^i = \{\mathbf{x}^i \mid f(\mathbf{x}^i) < f_t\},$$

using a segmentation threshold f_t that defines the difference between material and void voxel values. The segmentation threshold was implemented as an adjustable parameter in the MegaMol Configurator interface. This thesis ignored the other pre-processing steps such as using a non-local means filter to remove noisy voxel data used by Arand and Hesser [AH17], as the data used here is from simulated data and not reconstructed from XCT scanners. Therefore, the data processed in this thesis does not contain noise like data captured in real images.

4.1.3 Grid Line Sampling

The first contribution of Arand and Hesser [AH17] that differs from previous methods is to extract further geometrical information from the data in order to use grid line sampling to implement the Euclidean Distance Transform (EDT) from Lindblad and Sladoje [LS15] to calculate a distance field. The reasoning here is to efficiently calculate a more accurate Maximal Ball at each \mathbf{x}_{void}^i . A Maximal Ball MB^i centered at any point \mathbf{x}_{void}^i is formally defined as “a sphere centered at \mathbf{x}_{void}^i , with radius r^i chosen maximal such that MB^i is completely contained in Ω_{void} ” [AH17], where Ω_{void} is the set of all void voxels. Previous works calculated such a Maximal Ball for each void voxel by constructing a ball until it hits a boundary between the void space and material or by using a two-part inflating and deflating search [DB09]. Using grid line sampling and a distance field, Arand and Hesser [AH17] are able to efficiently calculate Maximal Balls to a sub-voxel accuracy, and the Maximal Balls for each \mathbf{x}_{void}^i are immediately available from the calculation of the distance field.

In order to use this method, the data was first reconstructed into an integer grid. First the resolution of the data was calculated as the average difference between voxels in each direction. This was found to be $15.6nm$. Using the known information that the data is a Cartesian grid with the calculated cell size resolution, an integer grid could be constructed using the formulas

$$\begin{aligned} particle_x^i &= \left\{ \text{floor} \left(\frac{\mathbf{x}_x^i}{\text{resolution}} \right) \mid x \in (0, \dots, 256) \right\}, \\ particle_y^i &= \left\{ \text{floor} \left(\frac{\mathbf{x}_y^i}{\text{resolution}} \right) \mid y \in (0, \dots, 64) \right\}, \\ particle_z^i &= \left\{ \text{floor} \left(\frac{\mathbf{x}_z^i}{\text{resolution}} \right) \mid z \in (0, \dots, 64) \right\}, \end{aligned}$$

A particle was implemented here as a subclass of the grid that contains the x , y , and z coordinates of voxel \mathbf{x}^i , the voxel value f^i , and whether the particle is contained in Ω_{void} or $\Omega_{material}$. The particles were then arranged in a container, which was implemented as a linear data structure and indexed using the row-major format based on the x , y , and z coordinates.

With the integer grid properly constructed, it was then possible to implement a grid line sampling distance transform as defined by [LS15]. Whereas grid intersection sampling functions by identifying the integer-valued indices nearest to a continuous shape, a grid line sampling scheme functions by using a real-valued index to exactly identify the location on the grid lines where the continuous shape intersects them. By using a real-valued index for one axis and integer-valued indices for the remaining axes, it is possible to more accurately describe shapes on a grid.

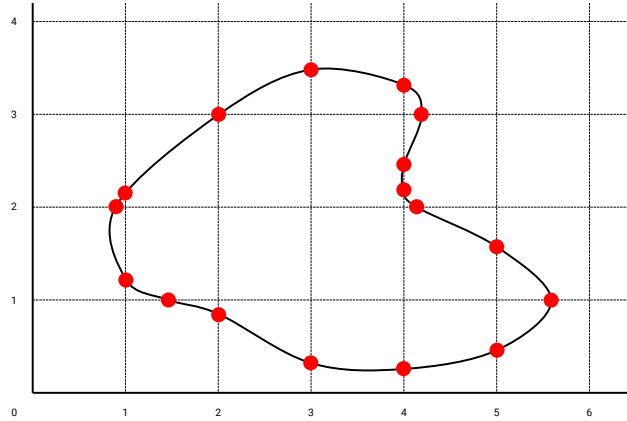


Figure 4.1: A continuous shape, where the red dots show the resulting indices of grid line sampling.

An example of a continuous shape that has been grid line sampled in 2D can be seen in *Figure 4.1*. This continuous shape will be used as an illustration of processing steps in all following examples. Any references to results are not indicative of actual results of this thesis, but rather are only meant to provide a use case as an example.

To create a grid line sampled data set, an algorithm similar to Marching Cubes was implemented to calculate the boundary points of the grid. Marching Cubes is an algorithm to extract the isosurface from a scalar field [LC87]. Boundary points are defined as the position on a line between two grid points where the value of the boundary point is equal to the segmentation threshold f_t [LS15]. Boundary points, therefore, contain a real-valued index for one axis that indicate a sub-voxel accurate border between Ω_{void} and $\Omega_{material}$ with $k \in \{1, \dots, K\}$, where K is the total number of boundary points. This was achieved by iterating through all particles, analyzing all particles in the 6-neighborhood of the current particle, and linearly interpolating between the current particle and neighboring particles if they differed in terms of Ω_{void} or $\Omega_{material}$.

While \mathbf{x}_{bound}^k consists of points in which any axis could be the real-valued one, the algorithm proposed by Lindblad and Sladoje [LS15] operates with respect to each real-valued axis separately, therefore, boundary point sets were constructed by grouping them with respect to each direction. This was done by constructing the sets for each direction respective to the direction of interpolation, i.e.

$$\mathbf{x}_{bound,x}^k \in \mathbb{R} \times \mathbb{Z} \times \mathbb{Z}$$

$$\mathbf{x}_{bound,y}^k \in \mathbb{Z} \times \mathbb{R} \times \mathbb{Z}$$

$$\mathbf{x}_{bound,z}^k \in \mathbb{Z} \times \mathbb{Z} \times \mathbb{R}$$

Figure 4.2 displays, in 2D, how the same continuous shape is grouped into two sets for grid line sampling in each direction.

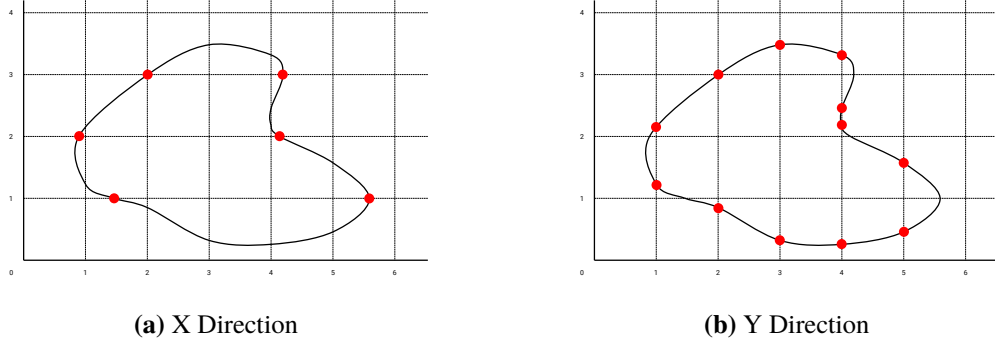


Figure 4.2: A continuous shape with Grid Line Sampling in each direction.

4.1.4 Distance Field

The algorithm used for the calculation of the distance field comes from the work of Lindblad and Sladoje [LS15]. The computation time of a distance transform is expensive, and the efficient implementation of Lindblad and Sladoje [LS15] is only of linear complexity. Additionally, their method is well-defined and is well-suited for parallelization. The use of distance fields is actually an idea taken from medial axis methods, where the distance transform is used to extract the skeleton of the pore structures [GDN+07]. Here, it is used to increase the accuracy of the Maximal Balls method.

Given three sets of boundary points, each corresponding to a real-valued index in one direction and an integer-valued index in the other two, it is possible to use the algorithm presented by [LS15]. However, the algorithm requires a specified structure of input, which will first be considered here as it is non-trivial to prepare the data for the algorithm and to be able to properly access it with the required indexing scheme. Further, the algorithm is presented by Lindblad and Sladoje [LS15] in two-dimensional space, with a suggestion as to how it can be used in three-dimensional space. The methods of this thesis will explicitly present the algorithm in 3D so that it is unambiguous how the methods were executed.

As mentioned above, to execute the algorithm for multi-dimensional data, the distance field algorithm must be executed separately for each dimension. Therefore, the input data sets must be arranged correctly for each dimension. Given three sets of points,

$$S_x = \{(x, y, z) \mid x \in \mathbb{R}, y \in \mathbb{Z}, z \in \mathbb{Z}\}$$

$$S_y = \{(x, y, z) \mid x \in \mathbb{Z}, y \in \mathbb{R}, z \in \mathbb{Z}\}$$

$$S_z = \{(x, y, z) \mid x \in \mathbb{Z}, y \in \mathbb{Z}, z \in \mathbb{R}\}$$

corresponding to the boundary points determined above, the input to the algorithm can be arranged.

The input defined by Lindblad and Sladoje [LS15] is defined here in 3D as a sorted array $s_x[y, z, i]$, where $y, z, i \in \mathbb{Z}$ correspond to the i -th real-valued x -coordinate in $S_x \cap C_{yz}$, where C_{yz} represents each column of x along the y and z axes of the grid. In other words, $s_x[y, z, i]$ contains the real-valued x -coordinates for each column where y and z are constant and i indicates the index of each point contained in that column. Swapping any coordinate allows the sets to be filled with

$s_x[1, 0] = 1.41$	$s_y[1, 0] = 1.20$
$s_x[1, 1] = 5.56$	$s_y[1, 1] = 2.15$
$s_x[2, 0] = 0.90$	$s_y[2, 0] = 0.84$
$s_x[2, 1] = 4.15$	$s_y[2, 1] = 3.00$
$s_x[3, 0] = 2.00$	$s_y[3, 0] = 0.32$
$s_x[3, 1] = 4.19$	$s_y[3, 1] = 3.55$
	$s_y[4, 0] = 0.26$
	$s_y[4, 1] = 2.16$
	$s_y[4, 2] = 2.43$
	$s_y[4, 3] = 3.37$
	$s_y[5, 0] = 0.45$
	$s_y[5, 1] = 1.59$
(a) $s_x[y, i]$	(b) $s_y[x, i]$

Figure 4.3: An example of the input sets created for the continuous shape with grid line sampling specified for each axis in *Figure 4.2*.

$e_x[0] = 0$	$e_y[0] = 0$
$e_x[1] = 2$	$e_y[1] = 2$
$e_x[2] = 2$	$e_y[2] = 2$
$e_x[3] = 2$	$e_y[3] = 2$
$e_x[4] = 0$	$e_y[4] = 4$
	$e_y[5] = 2$
	$e_y[6] = 0$
(a) $e_x[y]$	(b) $e_y[x]$

Figure 4.4: An example of the input set sizes based on the results from *Figure 4.2*.

real-valued indices along any column where the other two coordinates are integer and constant. The array is sorted in ascending order separately for each range within the array that corresponds to each column. An example of the 2D sets created for the continuous shapes in *Figure 4.2* is shown in *Figure 4.3*.

Furthermore, a 2D array $e_x[y, z]$ corresponds to the number of points in $S_x \cap C_{yz}$ for each y and z . This allows for accessing each i -th valued index in a column. An example of the size arrays created for the continuous shapes in *Figure 4.2* is shown in *Figure 4.4*.

By swapping the coordinates for each direction corresponding to the real-valued coordinate, the input can be represented the same way for each dimension, i.e. $s_y[x, z, i]$ with $e_y[x, z]$ for real-valued y -coordinates and $s_z[x, y, i]$ with $e_z[x, y]$ for real-valued z -coordinates.

Given the previously calculated boundary points organized into sets as defined above, the sets were then arranged into the desired structure using the developed *Algorithm 4.1*. The algorithm functions by first determining the number of boundary points on each x column for each y and z and linearly storing the index for each point. It then constructs the data into $s_x[y, x, i]$ using the stored indices and determined sizes, and finally sorts the values within the range of i for each y and z .

As accessing elements in the arranged input sets was non-trivial due to a variable size of i , *Algorithm 4.2* was developed to access the appropriate value in the sorted sets. It essentially functions by accumulating the number of all preceding sizes of the current position.

Algorithm 4.1 Distance Field Data Set Arrangement

```

procedure ARRANGEDATAX( $S_x = \{(x, y, z) \mid x \in \mathbb{R}, y \in \mathbb{Z}, z \in \mathbb{Z}\}$ )
   $temp_x := []$ 
   $count := 0$ 
  for all  $y \in \{0, \dots, y_{max} - 1\}$  do
    for all  $z \in \{0, \dots, z_{max} - 1\}$  do
       $e_x[y, z] := 0$ 
      for all  $j \in \{0, \dots, S_{x.size} - 1\}$  do
        if  $S_{x.y} == y \ \&\& \ S_{x.z} == z$  then
           $e_x[y, z] := e_x[y, z] + 1$ 
           $temp_x.push\_back(j)$ 
        end if
      end for
    end for
  end for
  for all  $y \in \{0, \dots, y_{max} - 1\}$  do
    for all  $z \in \{0, \dots, z_{max} - 1\}$  do
      for all  $i \in \{0, \dots, e_x[y, z] - 1\}$  do
         $s_x[y, z, i] := S_x[temp_x[count]].x$ 
         $count := count + 1$ 
      end for
    end for
  end for
  for all  $y \in \{0, \dots, y_{max} - 1\}$  do
    for all  $z \in \{0, \dots, z_{max} - 1\}$  do
      if  $e_x[y, z] \neq 0$  then
         $sort(s_x[y, z, 0], s_x[y, z, i])$ 
      end if
    end for
  end for
end procedure

```

In both *Algorithm 4.1* and *Algorithm 4.2*, the algorithms can be easily modified to prepare and access the data for any other dimension by swapping the appropriate coordinates. This is necessary in determining the remaining sets for all dimensions to be used as input to the algorithm.

Given the correctly prepared data and accessors, it was then possible to implement the EDT from Lindblad and Sladoje [LS15] to determine the shortest distance from each point on the integer grid to a boundary point, i.e. of each \mathbf{x}^i to \mathbf{x}_{bound}^k . The algorithm consists of two phases.

The first phase has been adjusted by Lindblad and Sladoje [LS15] from the original algorithm by Meijster et al. [MRH00] to allow for sub-voxel calculation of the distance field. For the x direction, it consists of scanning each column C_{yz} of the image and calculates the distance $G(x, y, z)$ of each point (x, y, z) to the nearest boundary point in $S_x \cap C_{yz}$. In other words, it scans a column where y and z are constant while the x direction is scanned once from top to bottom and once from bottom to top to find the minimum distance of each point in that column to a boundary point that lies within that column.

Algorithm 4.2 Distance Field Data Set Access

```

procedure ACCESSDATA $X(s_x[y, z, i])$ 
  if  $e_x[y, z] == 0$  then
    return(-1)
  end if
   $sum := 0$ 
  if  $z > 0$  then
    for all  $q \in \{0, \dots, z - 1\}$  do
       $sum := sum + e_x[y, q]$ 
    end for
  end if
  if  $y > 0$  then
    for all  $m \in \{0, \dots, y - 1\}$  do
      for all  $n \in \{z, \dots, z_{max}\}$  do
         $sum := sum + e_x[m, n]$ 
      end for
    end for
  end if
  for all  $j \in \{0, \dots, y - 1\}$  do
    for all  $k \in \{0, \dots, z - 1\}$  do
       $sum := sum + e_x[j, k]$ 
    end for
  end for
  return( $s_x[sum + i]$ )
end procedure

```

The second phase of the algorithm according to Lindblad and Sladoje [LS15] is identical to the original by Meijster et al. [MRH00]. It consists of scanning each row R_{xy} and R_{xz} and determines, for each point along the row, the minimum of the distance found in the first phase compared to the Euclidean distance of the current point to each point along the row plus each point's minimum distance found in the first phase [MRH00].

$Sep(i, u)$ is defined for the Euclidean distance as

$$Sep(i, u) = (u^2 - i^2 + g(u)^2 - g(i)^2) \mathbf{div}(2(u - i))$$

where \mathbf{div} is defined as integer division with rounding towards zero [MRH00]. $g(i)$ is used as a shorthand by Meijster et al. [MRH00] for all x with fixed y and z and is therefore equal to $g(i) = G(i, y, z)$.

Both algorithms are written here to explicitly define them again for the 3D usage of this thesis in *Algorithms 4.3* and *4.4* and are directly adapted from [LS15]. *Algorithm 4.3* shows the first phase of the algorithm, and *Algorithm 4.4* shows the second phase for row R_{xy} . In practice, it is sufficient to use $x_{max} + y_{max} + z_{max}$ to replace ∞ since all distances in the data are less than $x_{max} + y_{max} + z_{max}$ [MRH00].

Algorithm 4.3 Distance Field Phase 1

```

procedure PHASE1X( $s_x[y, z, i], e_x[y, z]$ )
  for all  $y \in \{0, \dots, y_{max} - 1\}$  do
    for all  $z \in \{0, \dots, y_{max} - 1\}$  do
       $i := 0$ 
       $p := -\infty$ 
      for all  $x \in \{1, \dots, x_{max} - 1\}$  do // Scan1
        while  $i < e_x[y, z] \wedge s_x[y, z, i] \leq x$  do
           $p := s_x[y, z, i]$ 
           $i := i + 1$ 
        end while
         $g_x[x, y, z] := x - p$ 
      end for
       $i := e_x[y, z] - 1$ 
       $p := \infty$ 
      for all  $x \in \{x_{max} - 1, \dots, 0\}$  do // Scan2
        while  $i \geq 0 \wedge s_x[y, z, i] \geq x$  do
           $p := s_x[y, z, i]$ 
           $i := i - 1$ 
        end while
         $g_x[x, y, z] := \min(g_x[x, y, z], p - y)$ 
      end for
    end for
  end for
end procedure

```

The coordinates are swapped with the algorithm as written for the second phase of the algorithm to determine the scan for R_{xz} and the minimum of the scans for R_{xy} and R_{xz} is used. When swapping the coordinates, it is important to note that $dt_x[x, y, u]$ is changed to $dt_x[x, u, z]$ in the second phase for z . Finally, the two phases are executed for each coordinate, and the minimum of the three results for each voxel determines the overall distance field for the data set. This is theoretically clear because each result of the algorithm in a certain direction gives the minimum distance for a single point with respect to that direction. The minimum distance of all directions for a single point is the overall minimal distance for that point [LS15].

An example of the intermediate result after the first phase of the Distance Field Algorithm applied to the continuous shape from *Figure 4.1* can be found in *Figures 4.5* and *4.6*, which shows the results of the calculation of the initial distance of all grid points to the nearest corresponding boundary point. It is important to note that for all columns that do not contain any boundary points, the initial distance for all points in that column is ∞ .

An example of the intermediate result after the second phase of the Distance Field Algorithm applied to the continuous shape from *Figure 4.1* can be found in *Figure 4.7*. It shows the results of the calculation of each set of distance transforms for each dimension and the overall total minimum distance. Here, only the points contained within the boundary are calculated and shown, as these are considered the void voxels and all voxels outside the boundary are considered as material voxels. It is important to note, however, that in the actual algorithm, the distance field is calculated for

Algorithm 4.4 Distance Field Phase 2

```

procedure PHASE2X( $g_x[x, y, z]$ )
  for all  $x \in \{0, \dots, x_{max} - 1\}$  do
    for all  $y \in \{0, \dots, y_{max} - 1\}$  do
       $q := 0$ 
       $s[0] := 0$ 
       $t[0] := 0$ 
      for all  $u \in \{1, \dots, z_{max} - 1\}$  do // Scan3
        while  $q \geq 0 \wedge f(t[q], s[q]) > f(t[q], u)$  do
           $q := q - 1$ 
        end while
        if  $q < 0$  then
           $q := 0$ 
           $s[0] := u$ 
        else
           $w := 1 + Sep(s[q], u)$ 
          if  $w < z_{max}$  then
             $q := q + 1$ 
             $s[q] := u$ 
             $t[q] := w$ 
          end if
        end if
      end for
      for all  $u \in \{z_{max} - 1, \dots, 0\}$  do // Scan4
         $dt_x[x, y, u] := f(u, s[q])$ 
        if  $u == t[q]$  then
           $q := q - 1$ 
        end if
      end for
    end for
  end for
end procedure

```

$g[0] = 1.41$
 $g[1] = 0.41$
 $g[2] = 0.59$
 $g[3] = 1.59$
 $g[4] = 1.56$
 $g[5] = 0.56$
 $g[6] = 0.44$

(a) $G[x, 1]$

$g[0] = 0.90$
 $g[1] = 0.10$
 $g[2] = 1.10$
 $g[3] = 1.15$
 $g[4] = 0.15$
 $g[5] = 0.85$
 $g[6] = 1.85$

(b) $G[x, 2]$

$g[0] = 2.00$
 $g[1] = 1.00$
 $g[2] = 0.00$
 $g[3] = 1.00$
 $g[4] = 0.19$
 $g[5] = 0.81$
 $g[6] = 1.81$

(c) $G[x, 3]$ **Figure 4.5:** An example of the initial distances calculated for the continuous shape in *Figure 4.2a*.

$g[0] = 1.20$	$g[0] = 0.84$	$g[0] = 0.32$	$g[0] = 0.26$	$g[0] = 0.45$
$g[1] = 0.20$	$g[1] = 0.16$	$g[1] = 0.68$	$g[1] = 0.74$	$g[1] = 0.55$
$g[2] = 0.15$	$g[2] = 1.00$	$g[2] = 1.55$	$g[2] = 0.16$	$g[2] = 0.41$
$g[3] = 0.85$	$g[3] = 0.00$	$g[3] = 0.55$	$g[3] = 0.37$	$g[3] = 1.41$
$g[4] = 1.85$	$g[4] = 1.00$	$g[4] = 0.45$	$g[4] = 0.63$	$g[4] = 2.41$
(a) $G[1, y]$	(b) $G[2, y]$	(c) $G[3, y]$	(d) $G[4, y]$	(e) $G[5, y]$

Figure 4.6: An example of the initial distances calculated for the continuous shape in *Figure 4.2b*.

$dt_x[1, 2] = 0.010$	$dt_y[1, 2] = 0.023$	$dt[1, 2] = 0.010$
$dt_x[2, 1] = 0.348$	$dt_y[2, 1] = 0.026$	$dt[2, 1] = 0.026$
$dt_x[2, 2] = 1.000$	$dt_y[2, 2] = 1.000$	$dt[2, 2] = 1.000$
$dt_x[3, 1] = 2.323$	$dt_y[3, 1] = 0.462$	$dt[3, 1] = 0.462$
$dt_x[3, 2] = 1.323$	$dt_y[3, 2] = 1.026$	$dt[3, 2] = 1.026$
$dt_x[3, 3] = 1.000$	$dt_y[3, 3] = 0.303$	$dt[3, 3] = 0.303$
$dt_x[4, 1] = 1.023$	$dt_y[4, 1] = 0.548$	$dt[4, 1] = 0.548$
$dt_x[4, 2] = 0.023$	$dt_y[4, 2] = 0.026$	$dt[4, 2] = 0.023$
$dt_x[4, 3] = 0.036$	$dt_y[4, 3] = 0.137$	$dt[4, 3] = 0.036$
$dt_x[5, 1] = 0.314$	$dt_y[5, 1] = 0.303$	$dt[5, 1] = 0.303$
(a) The distance transform calculated for the continuous shape in <i>Figure 4.2a</i> .	(b) The distance transform calculated for the continuous shape in <i>Figure 4.2a</i> .	(c) The minimum of the two distance transforms to determine the overall distance transform.

Figure 4.7: An example of the intermediate results of the overall distance transform calculated for the continuous shape in *Figure 4.1*.

all points including material voxels. It is not necessary to distinguish them, however, for example with a signed distance transform, because the void voxels have already been identified using the segmentation threshold.

Finally, to determine the radius, each point is calculated using the square root of the distance transform. The square root of the distance transform is used because the result of the algorithm gives the squared values, and, therefore, the actual radius of each point to the boundary is the square root of its corresponding distance transform as the EDT was used as the distance transform. *Figure 4.8* shows an example of the 2D continuous shape that shows the radius of each point within the boundary.

4.1.5 Maximal Balls Algorithm

With the calculation of the distance field complete, it was then possible to directly assign a maximal ball for each voxel according to its corresponding distance to a boundary, i.e. for each MB^i , $r^i = \sqrt{d^i}$, where d^i is the distance of \mathbf{x}_{void}^i calculated in *Subsection 4.1.4* [AH17].

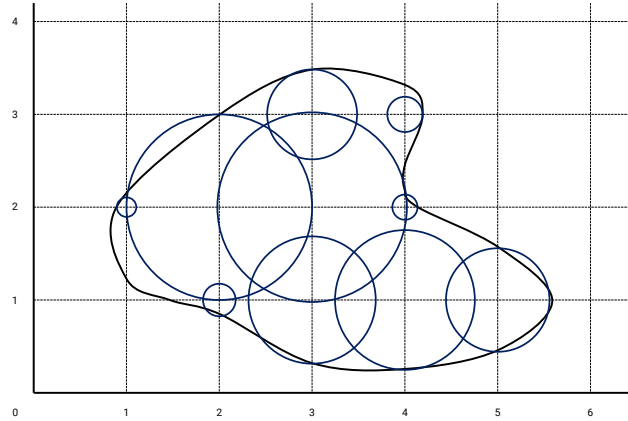


Figure 4.8: A continuous boundary, where all void voxels within the boundary show the radius calculated using the distance transform.

Here, a new continuous shape will be introduced as an example to highlight the features of the modified Maximal Balls algorithm. *Figure 4.9* shows a continuous shape where all void voxels have already been processed and assigned a radius according to their distance to the boundary. In addition, it shows the results of intermediate steps of the Maximal Balls algorithm as will be discussed in the following.

The modified version of the Maximal Balls algorithm by Arand and Hesser [AH17] is built on the work of Silin and Patzek [SP06]. Silin and Patzek [SP06] define relations between maximal balls using the terms master and slave, where „if the maximal balls of two voxels overlap, the one with the larger radius is called a master and the other one is called a slave"[SP06]. This hierarchy is then simplified to determine pore centers and throats of the pore network.

In the work by Arand and Hesser [AH17], they referred to this hierarchy using parent and children labels. Further, where by calculating the distance field and taking advantage of voxel coordinate indexing, the fact that most maximal balls have only one parent, and that pore centers can be identified by parent indices, they were able to eliminate voxel objects, lists, and a reference table in the work by Silin and Patzek [SP06] and do not need to store maximal balls explicitly in a data structure, but rather implicitly according to their index and radius [AH17].

To accomplish this, Arand and Hesser [AH17] introduce a 2-tuple morphology volume m_i , where

$$m_i = m(x^i) = [m_{state}^i \mid m_{parent}^i].$$

The first value indicates the state of voxel i , $m_{state}^i \in \{initialized, included, throat, material\}$. The second value indicates the index of the parent of the voxel. Using only the morphology volume and the distance field, Arand and Hesser [AH17] were able to complete the hierarchy to determine the center and size of all pores and throats.

Using the calculated distance field, *Algorithm 4.5* can be implemented to determine the pores and throats of the data. It first initializes the morphology volume to set all m^i with $x^i \in \Omega_{void}$ to $[initialized \mid i]$ and all m^i with $x^i \in \Omega_{material}$ to $[material \mid N + 1]$ [AH17]. In addition, during initialization, all void indices are stored in $I_{void}^{\downarrow d^i}$ and are sorted descending according to the radius, r^i , found using the distance transform [AH17].

Algorithm 4.5 Maximal Balls Algorithm

```

procedure MBA( $\{d^i\}$ )
  for all  $i \in \{0, \dots, N\}$  do // Initialization
    if  $\mathbf{x}^i \in \Omega_{void}$  then
       $m^i \leftarrow [initialized \mid i]$ 
       $I_{void}^{d^i}.push\_back(i)$ 
    else
       $m^i \leftarrow [material \mid N + 1]$ 
    end if
  end for
   $I_{void}^{\downarrow d^i} \leftarrow sort_{descending}(\{d^i\})$  // Processing
  for all  $i \in I_{void}^{\downarrow d^i}$  do
    if  $m_{state}^i == initialized$  then
      if  $m_{parent}^i == i$  then
         $m_{state}^i := included$ 
        ADDPORE( $\mathbf{x}^i$ )
      end if
      for all  $j \in \{i + 1, \dots, I_{void, size}^{\downarrow d^i}\}$  do
        if  $\|\mathbf{x}^j - \mathbf{x}^i\| \leq r^i$  then
          if  $m_{parent}^j \neq m_{parent}^i$  then
            if  $m^j == [initialized \mid j]$  then
               $m_{parent}^j := m_{parent}^i$ 
              if  $r^j + \|\mathbf{x}^j - \mathbf{x}^i\| \leq \epsilon r^j + r^i$  then
                 $m_{state}^j := included$ 
                Porei.Add( $\mathbf{x}^j$ )
              end if
            else if  $m_{parent}^j \neq j \ \&\& \ m_{parent}^j \neq m_{parent}^i$  then
               $m^j \leftarrow [throat \mid m_{parent}^j]$ 
            end if
          end if
        end if
      end for
    end if
  end for
   $\{\mathbf{x}_{throat}^i\}^{\downarrow d^i} \leftarrow sort_{descending}(\{\mathbf{x}_{throat}^i\}^{d^i})$  // Network Creation
  REDUCE THROATS( $\{\mathbf{x}_{throat}^i\}^{\downarrow d^i}$ )
  MAP( $pores \rightarrow throats$ )
  MAP( $throats \rightarrow pores$ )
  MERGE PORES( $\frac{r_{throat}^j}{r_{pore}^i} > \lambda$ )
end procedure

```

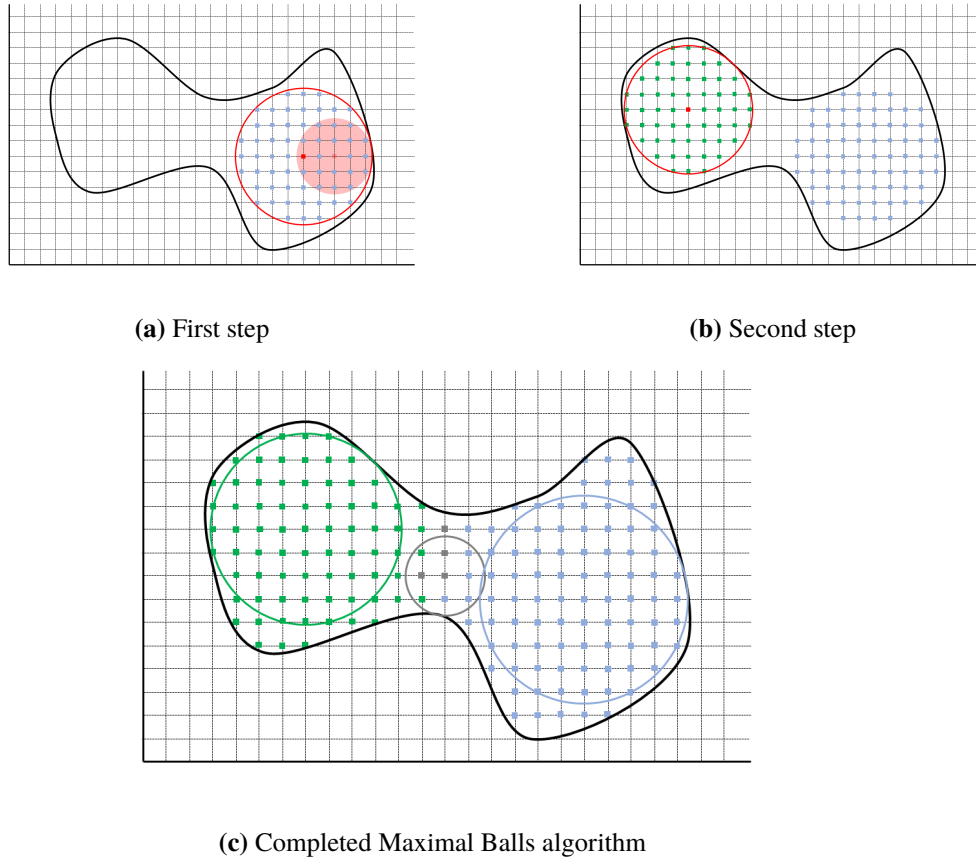


Figure 4.9: A continuous shape in 2D that has been processed according to the Maximal Balls algorithm. 4.9a shows the first step, where the voxel in red has the largest radius and is processed first. All voxels within its radius are assigned as children, and one voxel is shown to highlight when a voxel is completely included in its parent pore and will no longer be processed. 4.9b shows the next largest radius being processed and the voxels that are correspondingly assigned as children. 4.9c shows the final result, including the pore centers and voxels that have been assigned as throats with the corresponding throat center.

The processing then continues in order of $I_{void}^{\downarrow d^i}$. First, each voxel is only processed if $m_{state}^i = initialized$. Otherwise, m_{state}^i is *material*, *included*, or *throat* and does not contribute to further processing and can be ignored [AH17]. If $m_{state}^i = initialized$ and $m_{parent}^i = i$, then the current voxel is a pore center and $m^i \leftarrow [included \mid i]$ [AH17]. The implementation for this thesis differs from Arand and Hesser [AH17] by storing pores separately, and for each new pore center discovered, a new pore is created using that voxel as the pore center.

Regardless if the voxel is a pore center or not, when $m_{state}^i = initialized$, the algorithm will proceed to explore all voxels within the radius of \mathbf{x}^i and where $r^j \leq r^i$, i.e. where the radius of each voxel within the radius of \mathbf{x}^i is less than the radius of \mathbf{x}^j [AH17]. Therefore, all voxels within the radius of \mathbf{x}^i are found using a flood filling algorithm, which fulfills the requirement of $\|\mathbf{x}^j - \mathbf{x}^i\| \leq r^i$, and they are processed if $r^j \leq r^i$. The flood fill algorithm was developed for this thesis, and increments

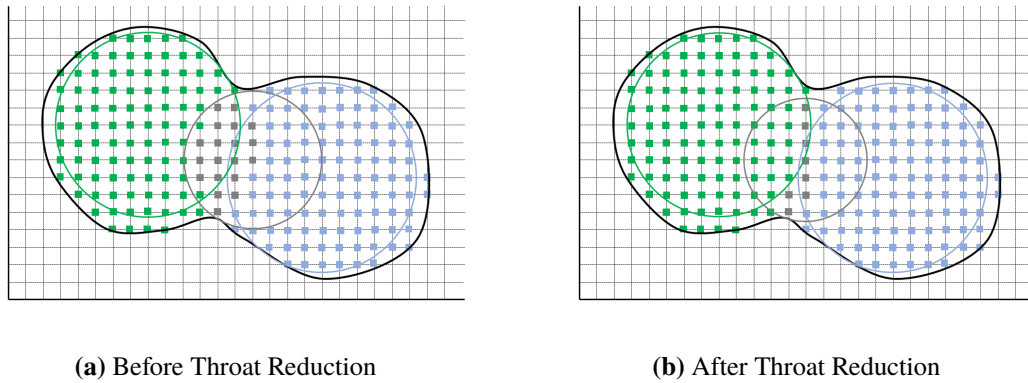


Figure 4.10: A continuous shape in 2D that has been processed according to the Maximal Balls algorithm. 4.10a shows the pore centers and voxels and throat center and voxels before the throat watershed algorithm. 4.10b shows the results after processing.

from \mathbf{x}^i in each direction positively and negatively. It proceeds in each direction until the radius r^i is passed and then continues in the next direction. Once the algorithm is complete, all voxels contained within the radius are found.

At this point, voxel \mathbf{x}^j will only be further processed if it does not already have the same parent as \mathbf{x}^i . In this case, \mathbf{x}^j falls into two scenarios, the first being that \mathbf{x}^j has not been processed yet, in which case the parent of \mathbf{x}^i is assigned as the parent of \mathbf{x}^j [AH17]. Further, if the radius of \mathbf{x}^j is completely included in the radius of \mathbf{x}^i , then it can be marked as included in the morphology volume, will no longer be processed, and \mathbf{x}^j is added to the pore which contains \mathbf{x}^i as the center [AH17].

This increases performance as included voxels will no longer be processed in further iterations of the algorithm. Analysis speed can be further increased by adding an inclusion parameter ϵ that allows including \mathbf{x}^j if its radius extends slightly beyond the boundary of the radius of \mathbf{x}^i [AH17]. This allows for performance increases without altering the morphological structure [AH17].

The second scenario for \mathbf{x}^j that does not have the same parent as \mathbf{x}^i is that \mathbf{x}^j has been previously processed and is not a pore center. In this case, \mathbf{x}^j is assigned as a throat [AH17].

At this point, it is necessary to reduce throat voxels using a watershed algorithm in order to restore hierarchy information from throats so that the pores connected to each throat can be determined [AH17]. To do so, throat voxels are sorted descending according to their radius and processed in that order. The 26-neighborhood of a throat voxel is examined, and for all neighboring voxels, if there is one distinct pore, the throat particle is added to the distinct pore and removed from the throat voxels. In this way, throat voxels are reduced until they only have two or more distinct pore neighbors [AH17]. Throat centers are identified as the largest voxel in a throat according to the radius values determined from the distance field [AH17]. *Figure 4.10* shows the results of the Maximal Balls algorithm before and after reducing throat voxels.

With the throats reduced, it is then possible to map all throats to each pore, and inversely all pores to each throat. This was implemented by creating a map for each, where all indices of one are stored with the indices of all its connected neighbors. For example, the map for pores includes all indices of pores that contain throats, and each index is associated with the indices for all throats connected to that pore.

The final phase of *Algorithm 4.5* is to merge pores. This is done in accordance with Homberg et al. [HBW+13] in order to account for unstable pores “due to the irregular shape and arrangement of particles.” For example, irregularly shaped pores using Maximal Balls will initially be classified as two or more mostly spherical shaped pores connected by large throats. Merging pores with large throats compared to the radius of the connected pore results in a singular pore that more accurately represents the material and its properties. This is also useful for long or flat pores for the same reasoning [HBW+13].

To merge pores, they are first sorted ascending according to their radius and processed in that order. If the ratio of the largest throat connected to the current pore compared to the pore’s own radius is above a threshold λ defined by the user, then all the voxels of the current pore are added to the largest pore connected to the throat [AH17]. Further, all throats of the merged pore are also added to the largest pore. If the throat between the current pore and the largest connected pore connected more than the two pores, then the pore is merged into the larger pore and the throat is ignored. However, if the throat only connected the two pores, then it is merged into the larger pore as well and removed as a throat.

4.2 Pore Tracking

A factor that led to interesting developments in this work and that did not need to be considered by the Maximal Balls method according to Arand and Hesser [AH17] is the implication of time. As the data being processed was for discretized time steps of a simulation, more information could be extracted from the pores as they related to the previous and following time step. This is not considered in previous works because most are only concerned with unchanging material samples. Therefore, this thesis implements a pore tracking scheme to identify the same pore between time frames and track its development.

Initially, a naive approach to pore tracking was implemented. This approach simply mapped pores from the current time frame to pores of the previous time frame based on the distance to the nearest pore center. In other words, all pores of the current time frame calculated the distance between their own pore center to the nearest pore center of the previous frame and were assigned to that pore.

The results of this approach were inaccurate and unstable. As pores developed, changes in pore centers and factors such as a pore splitting into two pores were tracked poorly. This led to the development of a more robust solution.

The improved solution proposed that by calculating the ellipsoid containing all of the points in a pore, then pore tracking could more accurately consider factors such as the size and orientation of a pore during processing. This allows for improved accuracy of tracking between time frames. It also provides the added benefit of being able to process the ellipsoids to extract information about the shapes of the pores themselves. This is in line with the three-dimensional pore shape classification developed by Claes et al. [CSCS16].

To calculate the ellipsoids containing the pores, each pore was processed using the coordinates of each voxel contained within that pore. These coordinates were used to calculate the covariance matrix of each pore, where the covariance matrix for three-dimensional data is a 9x9 matrix with each element found using

$$C = \frac{1}{N-1} \sum_N (X_i - \bar{x})(Y_i - \bar{y})$$

where N is the total number of voxels contained in the pore and \bar{x} and \bar{y} are the means of coordinate X and coordinate Y [Wei].

Along the diagonal elements of the matrix, the elements simplify to

$$C = \frac{1}{N-1} \sum_N (X_i - \bar{x})^2$$

which corresponds to the variance of the points over X .

If all other elements in the matrix are equal to zero, then the variances specified by the diagonal of the matrix correspond to the lengths of the ellipsoid in each direction along the axes. If the other elements are not equal to zero, then these covariances specify the orientation of the ellipsoid, which is found by using the eigendecomposition of the matrix to determine the unit vectors of the directions of the ellipsoid and the lengths in each direction. The Eigen library was used to implement this calculation of the eigenvalues and eigenvectors of the covariance matrix [GJ+10].

With this calculation complete, the ellipsoids containing each pore are fully defined and can be used to further process the pore tracking. To reduce computation time, an initial overlap condition is performed to determine if the sphere defined by the largest eigenvalue of a pore overlaps with the similarly defined sphere of any pore from the previous time frame. If only one overlap is detected, then those two pores are determined to be the same between time frames. If there are more than one overlapping spheres detected, then an algorithm is implemented to determine the amount of overlap between the current ellipsoid and each of the overlapping ellipsoids from the previous time frame. The ellipsoid with the highest number of overlapping points is assigned as the same pore from the previous time frame.

The algorithm to determine the amount of overlap is implemented in three dimensions by considering each voxel as a vector defined by its coordinates. For each voxel contained in the pore of the previous time frame,

$$\mathbf{x}_i = \langle x, y, z \rangle,$$

the center of the current pore's ellipsoid,

$$\mathbf{c} = \langle h, k, a \rangle,$$

is subtracted to determine a centered point,

$$\mathbf{p}_c = \mathbf{x}_i - \mathbf{c}.$$

Two matrices are defined for further processing:

$$A = \begin{bmatrix} r_x & 0 & 0 \\ 0 & r_y & 0 \\ 0 & 0 & r_z \end{bmatrix}$$

$$E = [e_{major} \ e_{sub} \ e_{minor}]$$

where r_x , r_y , and r_z are the lengths of each axis of the ellipsoid corresponding to the eigenvalues of the covariance matrix, and e_{major} , e_{sub} , and e_{minor} are the unit vectors of the directions of the maximum, medium, and minimum length axes of the ellipsoid corresponding to the eigenvectors of the covariance matrix [ht].

These matrices are used to determine the so-called whitening transform of the ellipsoid. This is done to statistically adapt the dimensions of the ellipsoid to be statistically independent, i.e. the covariance matrix of the transformed ellipse is diagonal. Essentially, this transforms the data into a unit sphere. It is then possible to apply this same transform to the centered point, \mathbf{p}_c , and if the point lies within or on the transformed sphere, then it lies within the original ellipsoid [ZL08].

The whitening transform is defined by the matrix

$$W = A^{-1/2} E^T.$$

Finally, the point is transformed using

$$\mathbf{p}_w = W\mathbf{p}.$$

If this point lies within the ellipsoid, then its value is less than or equal to one. If it lies outside of the ellipsoid, then its value is greater than one [ZL08].

This calculation is used to determine all points of the current pore to any and all ellipsoids from the previous time frame that overlap with the pores as determined by the sphere intersection test. The ellipsoid from the previous time frame that contains the highest number of points overlapping with the current pore is selected as the same pore and assigned to the current time frame.

This ability to use pore tracking is twofold. The first is to better visualize the entire data set as a whole. This means processing the entire data set and tracking all pores throughout all time frames. To accomplish this, a mapping was created to store pore centers and shapes (i.e. the eigenvectors and eigenvalues of each pore) between time frames. All pores that are matched with a previous pore are assigned the same color as the previous pore. Any pores that were not found in the current time frame are removed from the mapping and are considered to no longer exist.

The second use of pore tracking is to specifically track individual pores. This means selecting a pore and visualizing only that pore throughout all time frames that it exists. This was implemented as a parameter in the MegaMol interface that allows the selection of a time frame and an integer pore number. The pore corresponding to that number in that time frame would then be found and calculated for all time frames that it exists or if the end of a time range is reached.

4.3 Pore Statistics

Further processing using the calculated eigenvalues and eigenvectors could be performed in order to extract information about the shapes of the pores.

A 2D histogram was calculated to represent the distribution of the shapes of the pores. The range of the histogram is defined by the ratio of the largest eigenvalue of a pore to its intermediate eigenvalue. This represents a range of pores that represent in one extreme long and thin pores, with

little variance as the smallest eigenvalue must also be smaller than the intermediate eigenvalue. In the other extreme case, the ratio of the largest to intermediate eigenvalue approaches one with the smallest eigenvalue ranging anywhere between zero to the intermediate eigenvalue. This means the shapes in this extreme can represent anything between disk-shaped pores to spherical shaped pores. Referring to *Figure 2.3*, where the ratios are flipped compared to the definition here, shows how the smallest eigenvalue is not impactful when rod-like. Therefore, this method distinguishes between finger-like pores from the combination of more sponge-like shaped pores.

To determine the bin sizes and distribution within, the MegaMol interface of the Maximal Balls module allows for the input of the total number of bins as a parameter. Therefore, the bin sizes are calculated dynamically based on the user input.

First, the histogram was calculated linearly based on the ratio value. However, it became clear due to an largely uneven distribution that it would be better to plot the histogram using a logarithmic scaling. Therefore, to calculate the mapping of a ratio value i to bin indices, the following formula was developed:

$$index_i = round \left(10 \times \frac{\log(value_i)}{\log(\sqrt[x]{value_{max}})} \right)$$

This scales the values logarithmically while preserving the linear distribution of bins. The choice of x affects the spread of the histogram distribution.

This equation was found based on the desire to scale the values to be binned logarithmically, or more specifically where

$$x^{power_i} = value_i,$$

where $value_i$ is the ratio value to be binned. To map this function automatically with a desired spread using an unspecified range is difficult. However, since the values of concern were ratios where the numerator is always larger than the denominator, it is clear that the minimum possible value is one.

To consider the effect of the maximum value, which remains unknown, this particular case corresponds to

$$x^{power_{max}} = value_{max},$$

Since only the magnitudes of the radii are considered and are therefore positive, an equation has been determined for $x = \sqrt[power_{max}]{value_{max}}$. For a bin count of 50, it is convenient to select an $power_{max} = 5$ so that every ten bins increases the power by one starting with $index_{min} = 0$ corresponding to the minimum ratio of one. Therefore, the bin index corresponds to $10 \times power_i$.

Using these equations, it is possible to reformulate them to calculate the corresponding index value, while rounding the result to determine the proper integer bin value.

An additional step was performed when the distribution of values was still too disproportionate. This was determined to be because some smaller eigenvalues could approach zero, and would therefore skew the maximum value to be notably larger than the median value. In order to prevent skewing the distribution due to these ratios approaching infinity, a limit was used for the maximum value. Any ratios exceeding that value were simply mapped to the largest ratio bin.

Further, it was made possible to display all timeframes in a single histogram by arranging the bins along the x-axis, time frames along the y-axis, and mapping the size of each bin to color. This allows to intuitively see the development of the distribution of pores over time.

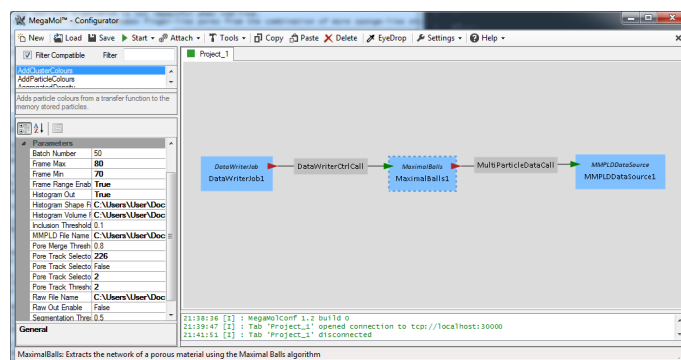


Figure 4.11: The MegaMol Configurator interface showing the Maximal Balls module within the data writing pipeline in addition to the parameters of the module [GKM+15].

4.4 Visualization

MegaMol was the visualization tool used for testing and results. As the Maximal Balls method was implemented within MegaMol, it was possible to create a visualization pipeline consisting of the input data, the Maximal Balls module, and a DataWriter module, as shown in *Figure 4.11*.

The data from the simulations was a series of .vtk files that needed to be modified to be used in MegaMol. This was done by first formatting them into the MegaMol standard MMPLD format file using a converter within MegaMol.

The MMPLD format is essentially a binary dump of data in a specified format including a header. The header specifies the version, number of frames of data, bounding and clipping boxes, and a seek table of byte offsets [GKM+15]. The data is then formatted according to the time frames and lists, which allow the file to represent multiple types of data.

With the input data properly formatted, it was first visualized using a particle filter module in MegaMol to segment the material and void voxels to create simple first visualizations of the material. In these visualizations, the voxels were located at each position according to their coordinate data and were represented as gray spheres with a fixed radius if their value corresponded to material or removed if void space.

Once the Maximal Balls method was implemented, it was then possible to create visualizations using the processed data. Initially, the data was only outputted as three lists that corresponded to the pores, throats, and material. This allowed visualizing all processed data together or each independently using a particle list selector module within MegaMol.

Pore voxels were visualized using their coordinate locations in the integer grid. They were rendered as spheres, where the radius of each sphere in the pore corresponded to its radius calculated using the distance transform. Further, each sphere contained in a pore used the same color to identify pores. Each pore's color was first chosen randomly, and pores that were tracked between time frames were assigned the same color as the previous pore.

Material voxels were visualized in accordance with the initial visualizations, where all were gray. However, after processing, material voxels also included a radius from the distance transform. Therefore, material voxels are also visualized using their radius, which allows for a more accurate

visualization of the border between material and void space. However, many material voxels early in the process contain a large radius due to their distance from the initial development of pores. Therefore, a limit was set for the size of the material radii to ensure a consistent visualization while still retaining a higher level of accuracy at the borders.

Lastly, throat voxels were visualized similarly to pore and material voxels using their location and radius. In order to make them easily visible and maintain a high contrast with the existing pore and material voxels, they were visualized as black spheres.

The histograms for each data set were output from the Maximal Balls module in MegaMol as a text file. They were then processed in Python using matplotlib.

5 Results

5.1 Performance

To properly assess the performance of the implementation, the different timings for each sub-process were recorded. These were recorded during processing using the chrono library in C++.

The distance field calculation algorithm was well-suited for parallelization, and was therefore implemented as such. The different loops were parallelized using the OpenMP library. No other methods were parallelized.

The results of performance metrics taken over the course of processing for three different data sets are shown in *Figure 5.1*, *Figure 5.2*, and *Figure 5.3*. The performance metrics show for each time step how long the corresponding method within the processing method performed. Each data point indicates the time required to execute for that time step. Therefore, the total time for each method is the sum of all time steps. The pore information displays the total number of pore voxels found in the corresponding time step. It is important to note that this is not the number of pores found, but the total number of voxels that were calculated as pores.

The total time required for each method for each data set is shown in *Table 5.2*. The average time per frame is also calculated and shown.



Figure 5.1: Performance of implementation for Data Set 1 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.

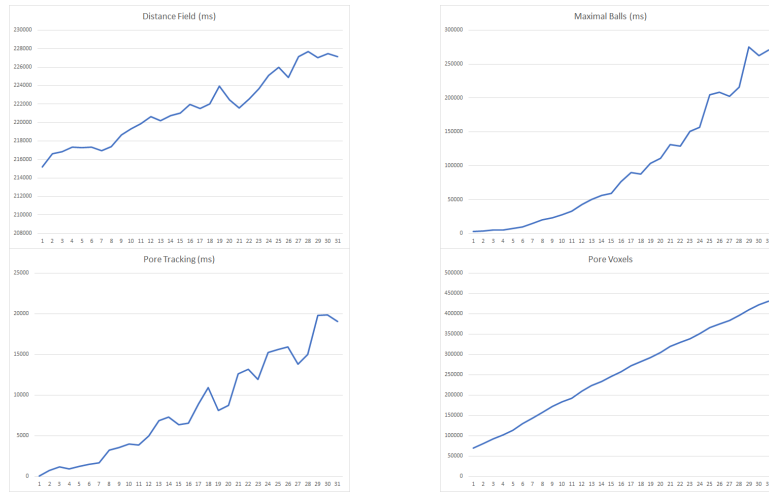


Figure 5.2: Performance of implementation for Data Set 2 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.

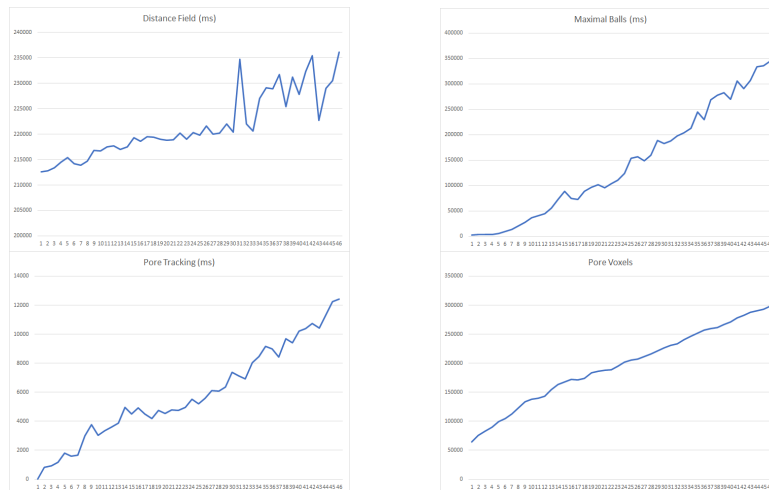


Figure 5.3: Performance of implementation for Data Set 3 over time. The x-axis corresponds to the time frame and the y-axis to time in milliseconds of processing.

Figure 5.4 shows the visualization pipeline setup that was used to create all visualization results from the Maximal Balls output.

5.2 Data Set Visualization

5.2.1 Full Data Set Visualizations

The first visualization result comes from visualizing the data set before processing. The visualization shown uses a constant radius for each particle and filters out particles with values above the segmentation threshold of 0.5. Therefore, it is a visualization of the material particles.

	Distance Field	Maximal Balls	Pore Tracking
Data Set 1	117.62	46.78	2.75
Data Set 2	118.07	50.51	4.38
Data Set 3	173.51	109.66	4.53

Table 5.1: Performance results in minutes per submethod of Maximal Balls implementation.

	Total	Average
Data Set 1 (32 frames)	167.16	5.22
Data Set 2 (32 frames)	172.96	5.40
Data Set 3 (47 frames)	287.70	6.12

Table 5.2: Average results in minutes per time frame of Maximal Balls implementation.

An example of the data set at selected time frames is shown in *Figure 5.5*, *Figure 5.6*, and *Figure 5.7*. These are the three sets of data used for the processing of this thesis, and will be the basis for the discussion. All results over time will be consistently oriented so that the development of the pores begins on the left and propagates to the right over time.

The processing for the Maximal Balls method as described in *Chapter 4* was applied to each time frame for each of these data sets. The final result of processing is shown in *Figure 5.8*. This shows the final time frame of processing for each set of data. It is a visualization of the full data output, which is the radius values for all voxels within the volume in addition to the mapping of each voxel to its determined type. The types material, throat, and pore are mapped to gray, black, and a random color respectively. As these results result in the inability to see most of the pores, visualizations are provided of the same time frames for each data set with the material removed. These images are shown in *Figure 5.9*.

Further visualizations are shown to convey the results over time. Corresponding to *Figure 5.5*, *Figure 5.6*, and *Figure 5.7*, the same time frames for each data set are visualized with only the pore output for each. These visualizations show the development of the pores in the material over time while disregarding all other data. They are found in *Figure 5.10*, *Figure 5.11*, and *Figure 5.12*.

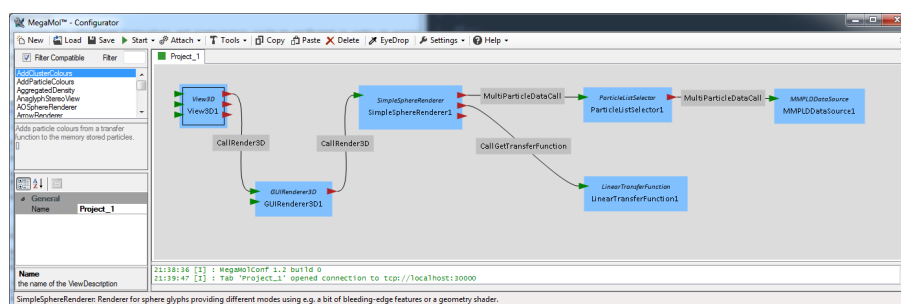


Figure 5.4: The MegaMol Configurator interface showing the visualization pipeline of the Maximal Balls output [GKM+15].

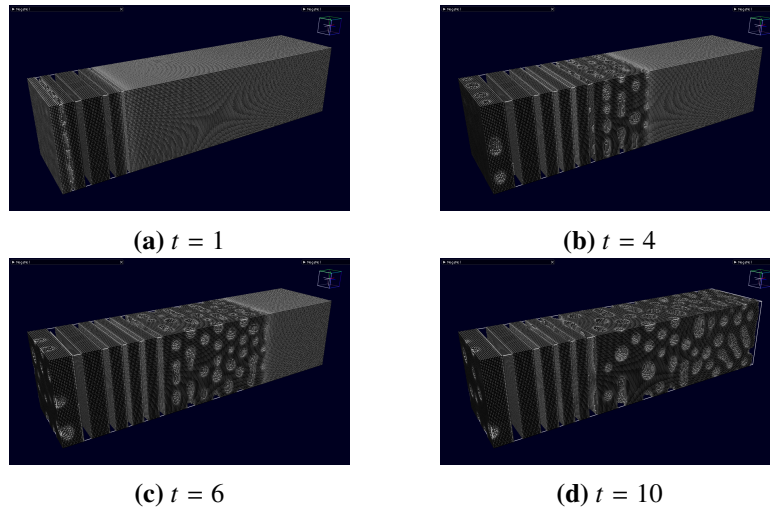


Figure 5.5: Initial visualizations of Data Set 1 over time. t corresponds to the time frame of the image.

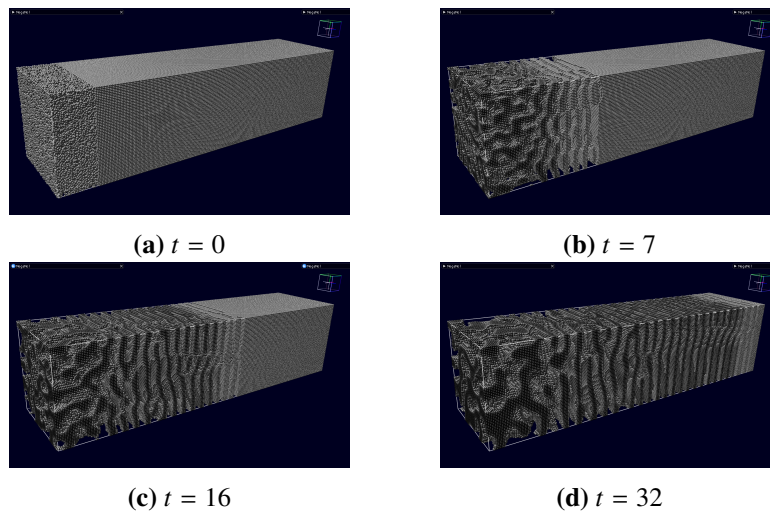


Figure 5.6: Initial visualizations of Data Set 2 over time. t corresponds to the time frame of the image.

5.3 Pore Tracking Visualization

5.3.1 Complete Pore Tracking

To display the results of tracking all pores between time frames, *Figure 5.13* shows a flat view of the short side of the porous volume. In the images, it is possible to see pores that remain the same color between time frames as being tracked as the same pore.

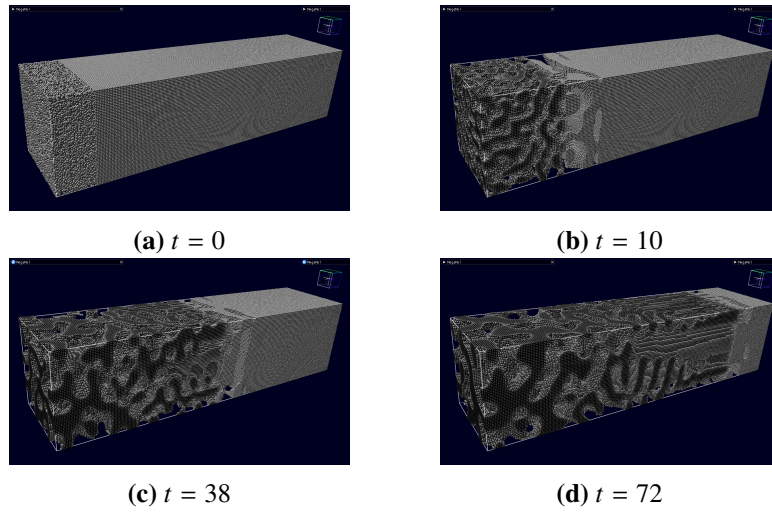


Figure 5.7: Initial visualizations of Data Set 3 set over time. t corresponds to the time frame of the image.

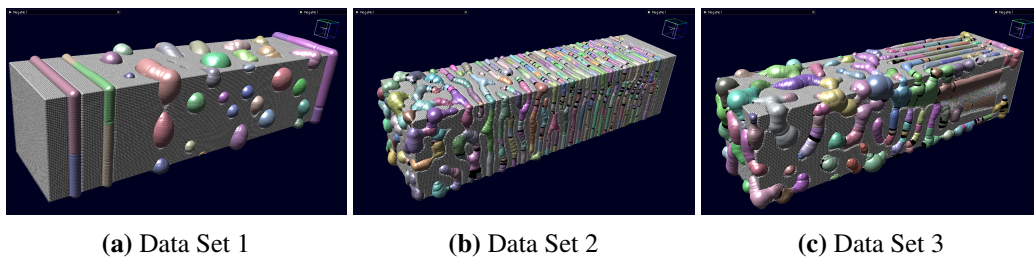


Figure 5.8: Full outputs of the Maximal Balls method for the final time frame of each data set.

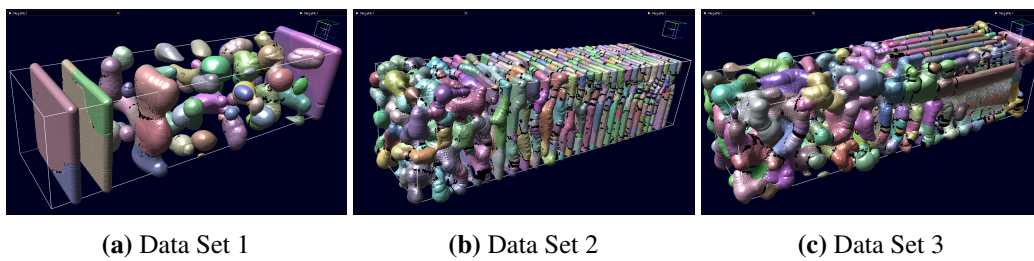


Figure 5.9: Output of only pores and throats from the Maximal Balls method for the final time frame of each data set.

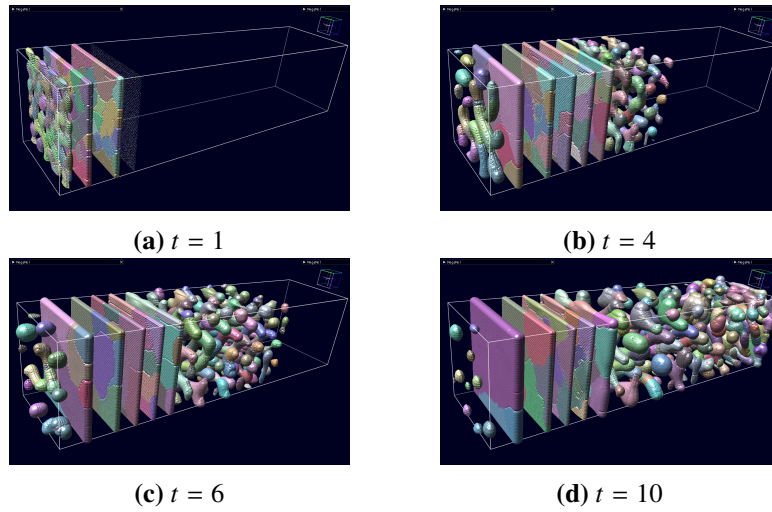


Figure 5.10: Pore visualizations of Data Set 1 over time. t corresponds to the time frame of the image.

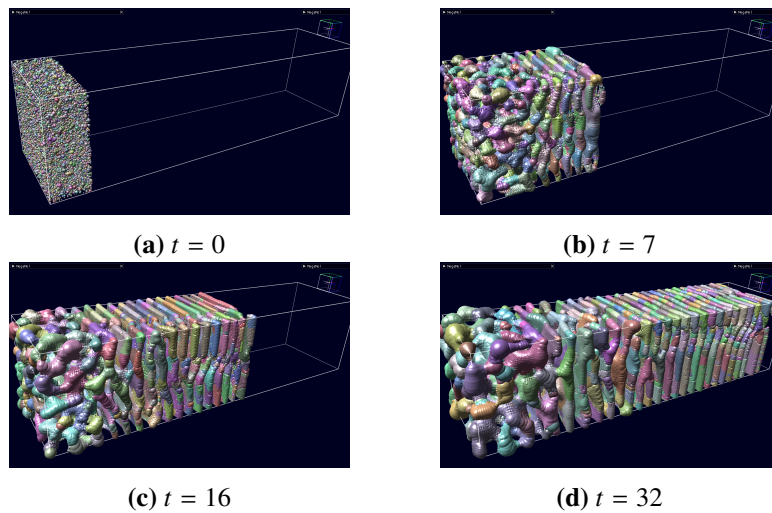


Figure 5.11: Pore visualizations of Data Set 2 over time. t corresponds to the time frame of the image.

5.3.2 Individual Pore Visualizations

In addition to visualizations of entire data sets, images are provided corresponding to the outputs of the individual pore tracking. The first visualization shows a single pore tracked over a series of time frames. It is found in *Figure 5.14*.

A second visualization using the ability of pore tracking was created as an attempt to compare the calculated pores to the direct visualization of the input data. This was done by creating a mesh of the input data around the selected pore and comparing the two. The process included calculating the extents of a selected pore and creating a raw binary output of the input voxel values. This output was then processed using the visualization software Inviwo [JSS+18]. Inviwo was used to create

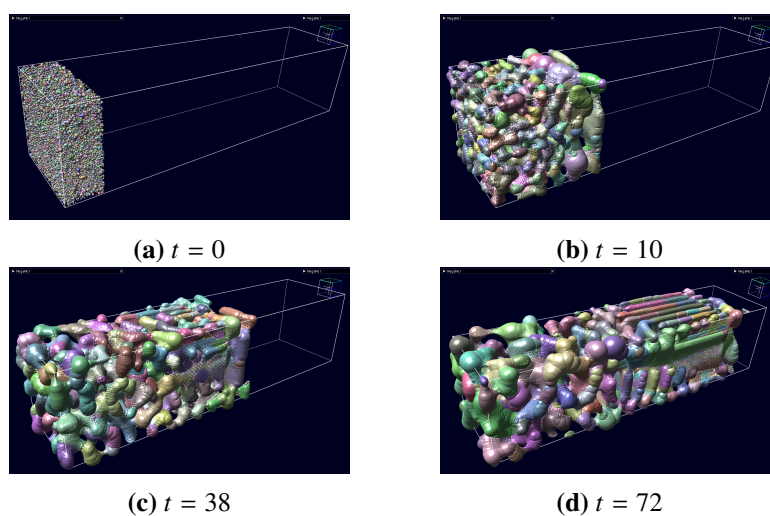


Figure 5.12: Pore visualizations of Data Set 3 set over time. t corresponds to the time frame of the image.

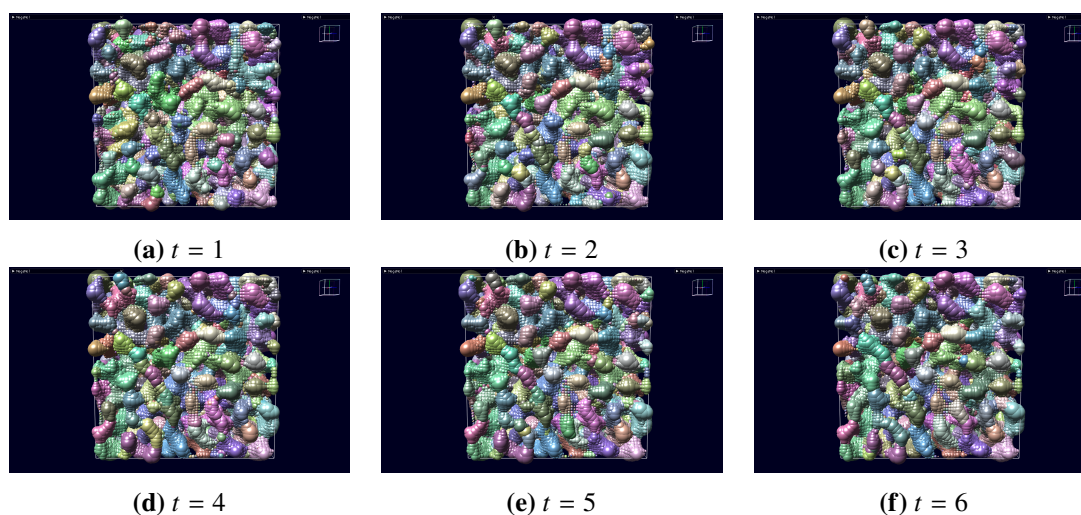


Figure 5.13: Visualization of pore tracking as pores develop between time frames.

a surface mesh from the binary data and output an object file. The object file was then able to be processed in MegaMol to visualize the volume directly. The result of a single pore from the Maximal Balls method is compared to a volume rendering of the input data in the corresponding bounds and shown in *Figure 5.15* at two different orientations.

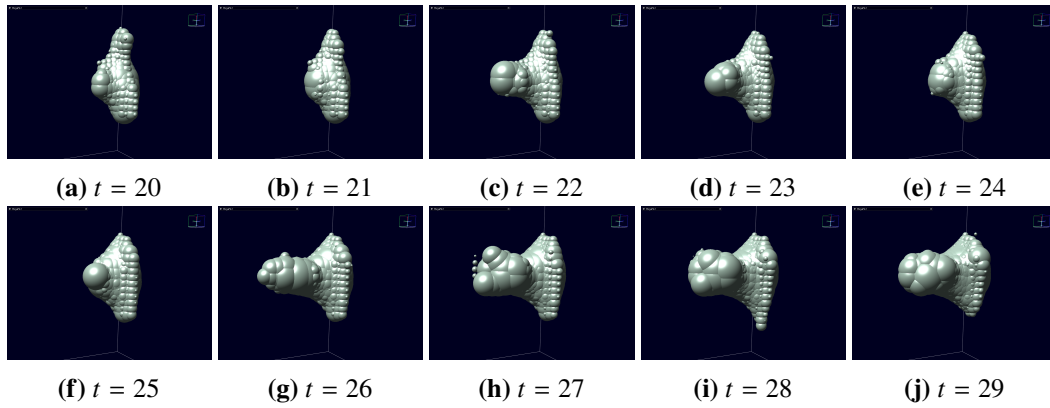


Figure 5.14: Pore tracking of a randomly selected pore over time. t corresponds to the time frame of the image.

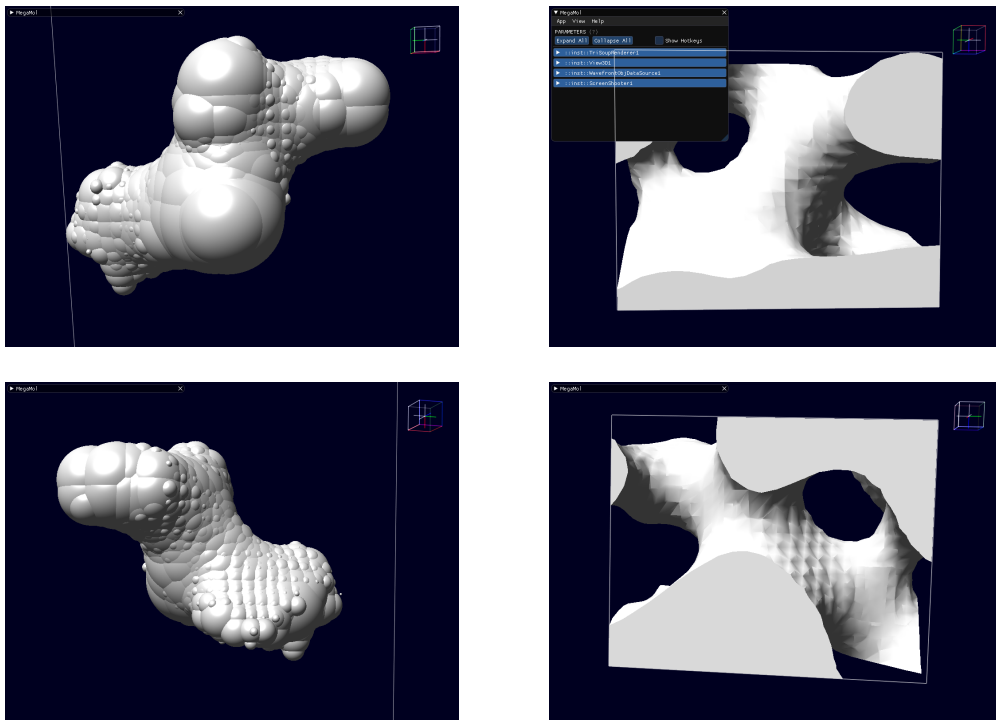


Figure 5.15: A comparison of a single pore result calculated according to the Maximal Balls method to a direct volume visualization of the input corresponding to the same area.

6 Discussion

6.1 Analysis

6.1.1 Performance Analysis

One of the first positive results for improving performance came from the development of the flood fill algorithm defined in *Section 4.1.5*. While Arand and Hesser [AH17] recommend the use of a flood fill algorithm, this was not implemented in the first iterations of the module.

While exact results were not implemented, it was recorded that the flood fill algorithm decreased the time to process a single frame from approximately eighteen minutes to one minute. Previously, the method worked by comparing the current particle being processed to all other particles with the voxel's radius calculated from the distance field. This is clearly a huge performance increase, as this previous implementation was of quadratic complexity.

This method of flood filling is essentially contributing more computing towards a found result in order to reduce the total amount of computation time due to avoiding future iterations. This is not only important for the improvement in efficiency of processing for this method, but also for considering other aspects of the methodology that could be improved, as will be discussed in *Section 6.3.1*.

Overall, performance was extremely dependent on the time. A higher time frame directly correlated with an increase in processing time. This makes sense, because the material initially has a very low porosity and increases as the porous structures develop. This can be seen in the visualization of the total number of pore voxels.

This issue with performance over time affected the ability to fully process large data sets with a high number of frames. In these cases, processing had to be split into batches. While performance was overall satisfactory, increasing efficiency to be able to calculate larger data sets for longer is desired. This may be accomplished by better considering the effect of time on the data structures and optimizing for longer scenarios.

6.1.2 Visual Analysis

The first visualization result of the material by itself provides a foundation for interpreting the rest of the results. For one, it gives a general idea of the distribution and shape of pores, which allows to make predictions regarding the distribution of shapes statistics.

It is possible to see the development of the material over time in the time frames shown. *Figure 5.6* and *Figure 5.7* show examples of highly porous material with many small pores. Over time, one can see the initial formation of many small sponge-like pores as it develops into longer, finger-like

pores, or possibly even lamella pores. It is expected not only to have visual results of pores that fit these empty spaces, but also the development of the histogram over time to progress from mostly sponge-like pores to a higher distribution of both.

The same analysis can be applied to *Figure 5.5*. This one clearly develops a very different pore morphology compared to *Figure 5.6* and *Figure 5.7*. *Figure 5.5* shows where some pores early form as lamella pores, but as time progresses, larger spherical pores with lots of space between the pores begin to form. Instead of an end result that has a high distribution of sponge or finger like pores, it is clear that this data set represents a strong mix of both. As such, we expect large singular pores to be mapped to each of these spherical and lamella pores and a distribution to be mostly mixed.

The results of pore visualizations are, from an intuitive perspective, in good agreement with the original data. An analysis of the three sets of data and combined results of material and pores shows good results. Further, the ability to track individual pore shapes is useful for a better understanding of their development.

An interesting note about the visualization of the material, is that alone it changes between the initial visualization and the results of the Maximal Balls method. This is because the initial visualization uses a constant radius for each particle. Therefore, the visualization is only an approximation of the pore space and the boundary with the material. However, after the calculation of the Maximal Balls, the material voxels also have a calculated radius to the border between void space and material. Therefore, they can also be visualized to have a more accurate visualization of the material on its own. This is useful as a more accurate representation for material may be required for more accurate analysis of the boundary.

While an intuitive visual analysis would lead to the conclusion that each slice that results from the formation of lamella pores in this data set is a single pore, the results of the Maximal Balls method maps several smaller lamella shaped pores to each expected individual pore. While this should still result in a similar distribution as would be originally expected, the difference is not ideal and the reasoning is due to a known limitation in the post processing of pore and throat merging. This will be discussed in more detail in *Section 6.2*.

However, there were also situations where it seems beneficial that these larger pores were split into multiple pores. For instance, *Figure 6.1* shows a lamella pore that is initially classified as multiple pores. In the following frames, the structure of the pore breaks down into several smaller pores that are consistent with the initial classification. This shows that although an intuitive analysis may show some faults, in certain scenarios the method may be very good at interpreting unobservable properties.

The histograms for each data set can be used as an analytical tool to determine the distribution of pore shapes. The visualization of all histograms are found in *Figure 6.2*, *Figure 6.3*, and *Figure 6.4*. In order to properly map the value of the number of values mapped to each bin, the viridis colormap from Hunter [Hun07] was selected to be perceptually uniform.

When a shape has a high ratio in the longest to intermediate then the smallest eigenvalue is not impactful and will mostly be finger-like. As it decreases, then pores will range from disk-like to spherical, but the main distinction is still preserved between finger-like pores and sponge-like pores.

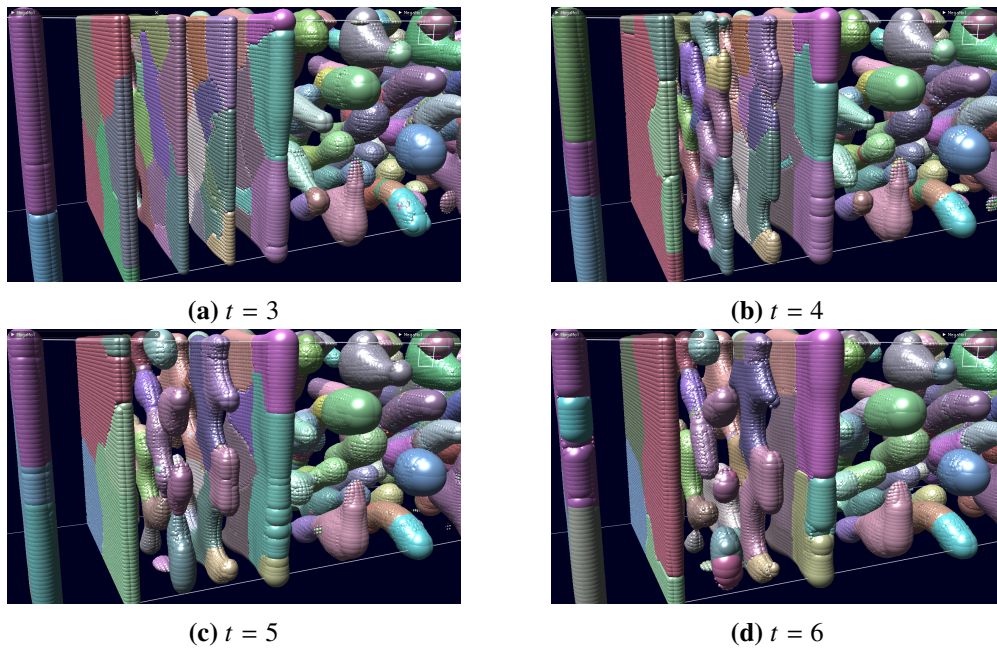


Figure 6.1: A visualization of a lamella pore being split over multiple time frames.

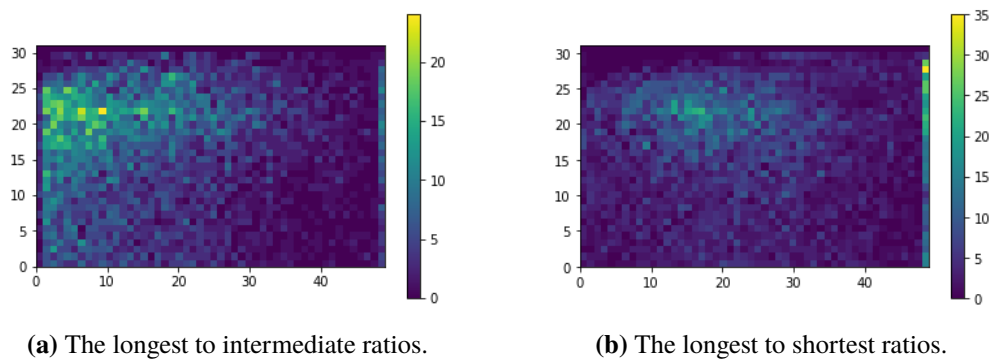


Figure 6.2: The histograms corresponding to Data Set 1. The x-axis corresponds to the bins and the y-axis corresponds to the time frame.

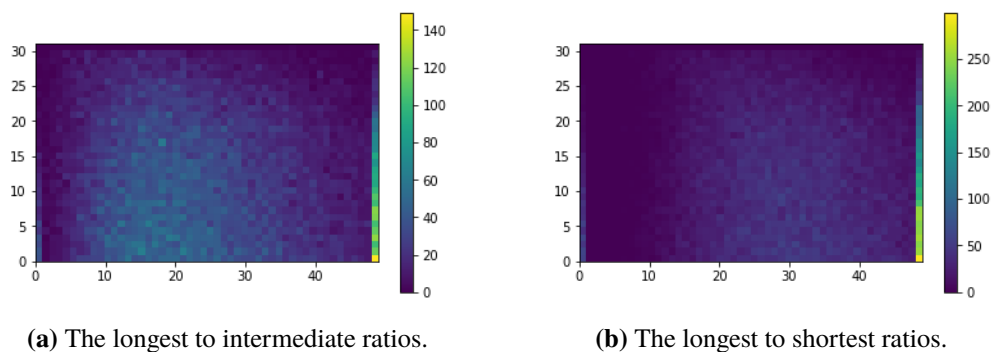


Figure 6.3: The histograms corresponding to Data Set 2. The x-axis corresponds to the bins and the y-axis corresponds to the time frame.

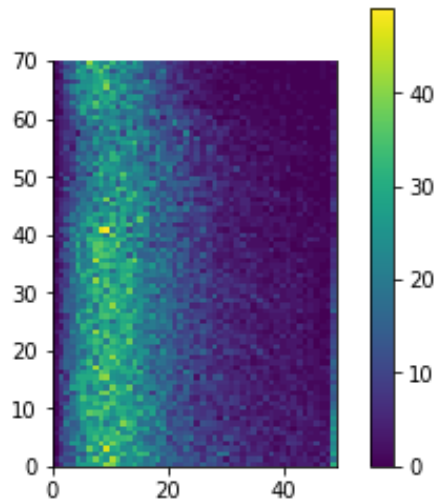


Figure 6.4: The histograms corresponding to Data Set 3. The x-axis corresponds to the bins and the y-axis corresponds to the time frame. Note: Due to an error in processing, only the longest to intermediate ratios are available.

As the longest to shortest ratio increases, pores in general will go from spherical shapes to flatter shapes. Again, when the longest to intermediate ratio is high, then this ratio is not a major factor. However, as the longest to intermediate ratio decreases, the longest to shortest ratio is more impactful on determining the overall shape.

As a further analysis of the accuracy of the pore shape extraction, the results from *Figure 5.15* are shown in *Figure 6.5* to confirm its results. The red line clarifies the alignment of the two pores and it is possible to see that the result of the Maximal Balls algorithm is in good agreement with the original data.

6.2 Limitations

A limitation that led to two issues in particular was the calculation of the pores at the boundary of the material. This issue comes from void space along the edges of the boundary that does not extend deep enough into the material, which causes miscalculations of pores. This results in the calculation of many small pores along the boundaries, because they do not have a large enough radius associated with them to contain enough other voxels to be merged together. Therefore, what visually appears to be one pore that extends into the boundary is represented as many pores located at each voxel along the border. Arand and Hesser [AH17] were also aware of this issue of the method, and resolved it by calculating pore statistics only by considering pore and throat centers that were far enough away from the boundary. This is clearly only a quick fix to treat the symptom and not the result, however, and remains a limitation of this thesis. An example of such a result can be seen in *Figure 6.6*.

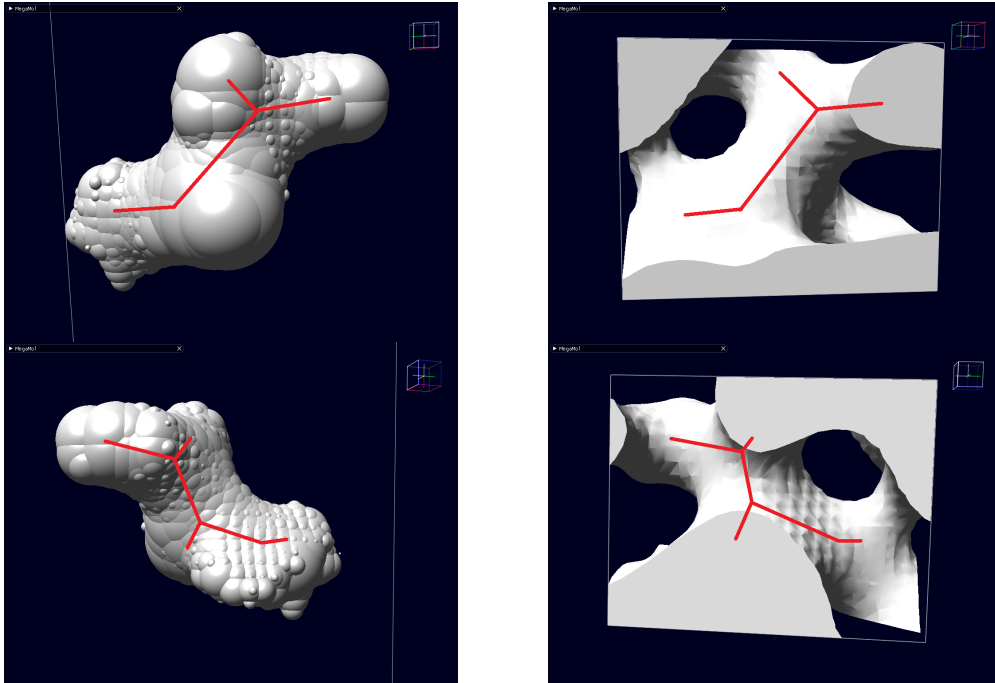


Figure 6.5: The results of single pore tracking and volume mesh analysis show good agreement with the calculation of the Maximal Balls method.

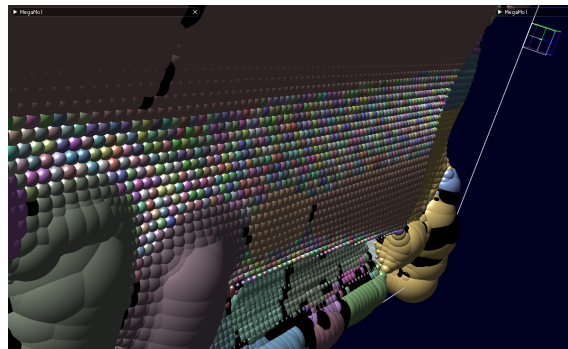


Figure 6.6: Results near the boundary condition that lead to many small pores in what appears to be one pore.

A possible solution for this issue could be to extend the boundary by artificially calculating the continuation of the border using the opposite surface as an entry point. This would be possible in this thesis because this data set loops in such a way.

The issue with this is firstly that the distance transform algorithm is very specific in its implementation. It includes a necessary order to properly calculate the distance transforms, and artificially increasing the size of the material would require restructuring the algorithm, specifically its start and end conditions. It would require even further restructuring to be able to conceptually separate the calculations for the material and for the artificial extension. This would be necessary to avoid unnecessary calculations of distances that are entirely outside of the actual volume. Lastly, by extending the volume by an extra half of the total distance in every direction except the length, this

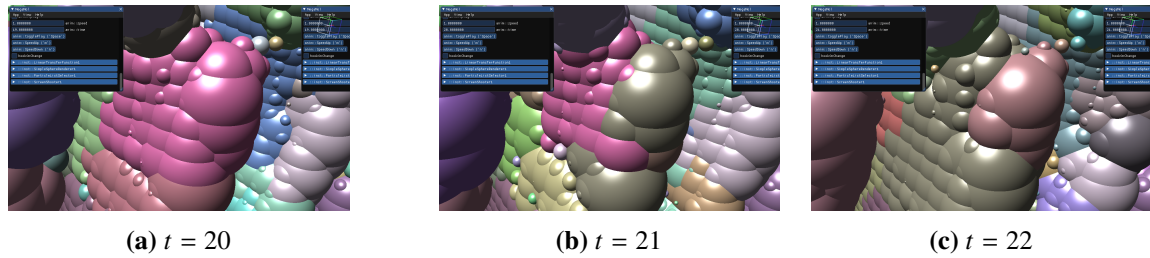


Figure 6.7: An example of a boundary pore that has a high variance in assignment.

results in five times the original volume. The distance transform is one of the most computationally demanding phases of this implementation, and that amount of increase would make computation times infeasible to implement efficiently.

An additional limitation that results from the pore boundary miscalculation is the poor tracking of pores along the edges of the material. Since pores are often miscalculated in these regions, visually observing pores that are tracked between frames in these areas are often erratic and not indicative of an intuitive analysis. The unstable nature of these results can be seen in *Figure 6.7*, where a pore along the edge is suddenly assigned a new color between time frames.

Another issue with pore tracking was the ability to individually track pores. While pore selection was possible using input parameters to the Maximal Balls module, determining a pore to track and visualize was only possible through a guess and check method using the desired starting frame and a randomly selected pore index. This made it difficult to select desired pores and led to many repeated calculations in search of quality pores to track. It would be beneficial to implement a feature to select pores in a stage after processing to allow for interactive visualizations. A potential method for doing so would be to implement a slider that iterates through all pores in the given time frame and tracks that pore over time.

A known limitation of the Maximal Balls implemented by Arand and Hesser [AH17], and therefore a limitation of this thesis, is the post processing of throats. More specifically, while pore centers are typically very accurate using this method, Arand and Hesser [AH17] found that about 70% of throat centers are in agreement with their reference [AH17]. The results of this are a higher number of throats found than expected and less merging of pores. This results in shapes such as the lamella pores being subdivided into multiple pores, when it is expected that they would be classified as a single pore.

6.3 Future Work

6.3.1 Pore Network Efficiency

In agreement with the discovery regarding the increase in performance of using a flood fill algorithm, the following two suggestions for further research are hypothesized as being potential bottlenecks of the current implementation of this thesis that can be made more efficient.

The first is the concern of throat reduction. While no exact timings were recorded, it was observed during the calculation of the pore network that a significant amount of the total time for the algorithm was spent on processing the throat reductions. This can be explained by the fact that the entire iteration of throats is restarted each time a throat that must be removed is found. In a worst case scenario, this iteration approaches quadratic complexity and many throat are revisited that may never be of concern.

It is proposed that there is potential for an algorithm similar to flood filling to be implemented to increase the efficiency of throat reduction. When a throat voxel is found that must be reduced, it is likely that its neighboring throat particles will also be reduced later iterations. Using a flood fill could lead to direct processing of the neighboring throats, however, the reason for restarting the iteration completely is because it may be a voxel on the other side of the throat that must be reduced first. Therefore, it is important that throat particles are always processed in the order of maximum radius when processing this way. The proposed change would therefore need to consider this fact in order to be effective.

Alternatively, when a throat voxel is found that must be reduced, it would likely be more efficient to consider the individual throat that it is connected to and process it entirely before continuing. This would likely also be more efficient because it is not necessary to iterate through every voxel multiple times, but reduce throats individually until consistent with the throat definition and then proceeding to the next largest throat voxel while ignoring processed throats.

In addition to the efficiency of calculating throats is the accuracy of calculating throats. As identified in *Section 6.2*, the Maximal Balls algorithm by Arand and Hesser [AH17] is limited in correctly determining the distinction between pores and throats. As this implementation is otherwise very efficient and accurate, further research in improving the post-processing of the pore network would be beneficial for pore network analysis as a whole.

The other proposal for efficiency improvements in the current method using a flood fill mindset is the calculation of pore tracking. The restriction of comparing every ellipsoid corresponding to each pore to every other ellipsoid corresponding to the pores of the previous time frame was already considered in the current implementation. This was done by using the outer sphere of each ellipsoid as an intersection test before then comparing overlap. However, it is theorized that further improvements could be made by using a more accurate intersection test. There exists methods to exactly test if two non-axis aligned ellipsoids intersect. While computationally more expensive than a sphere intersection test, it is proposed that this calculation would reduce the number of false positives found in the current method enough to be beneficial to the overall computation time, as the cost of computing the amount of overlap is high. Implementing this change is an interesting application for future research.

6.3.2 Shape Analysis

Pore shape analysis is an interesting aspect of the pore network that was considered in this work. Understanding shapes and how they are formed is a fundamental aspect of this thesis. Further research in shape analysis would be beneficial in determining more robust methods for ascertaining the distribution of pore shapes in a material.

One interesting possibility would be the application of machine learning to pore shape analysis. While this thesis determined the distribution of pores based on two different criteria that were directly compared, it would be interesting to see the results of a machine learning algorithm application to analyze any number of features corresponding to pore shapes. The correlations between pore volume, surface area, orientation, and even more exact representations of pore shapes rather than approximating ellipsoids could lead to determinations currently unknown. A machine learning algorithm may even also lead to more exact predictions of pore shapes and sizes based on other factors such as the initial composition and velocity of the precipitation front. The implications on further research in predicting pore properties could be significant.

6.3.3 Visualization

Several features could be implemented to extend the current work in providing visual tools to better understand the material. Firstly, a creation of meshes from the calculated pores would be more understandable and allow for a better visual analysis. In addition, they would allow for further determinations of properties by being able to calculate the surface areas of each pore.

One visualization that would be useful for analyzing the network is to be able to visualize ranges of pores based on different properties. For instance, one could select a range of volumes of interest and would then be able to see all pores in the material that meet that criteria. This level of interaction would be interesting and useful for researchers.

It would also be beneficial to implement more pore network visualization techniques as described by Grau et al. [GVTA10]. In addition to being able to select a pore using a slider in real-time as proposed in *Section 6.2*, it would be interesting to allow for the connections of the selected pore to be visualized. This would result in interesting visualizations that would vary greatly for each goal, such as visualizing only direct neighbors, visualizing all pores throughout the network that can be reached by the selected pore, or visualizing the shortest path from the selected pore to an open pore on the surface of the material. In any of these potential scenarios, similar complications observed by Grau et al. [GVTA10] would have to be considered, such as occlusion of important features and using color and opacity to visualize depth.

Further visualizations of interest would be creating abstract graphs of the different properties. Visualizing features of the pore network as in the bubble graphs produced by Grau et al. [GVTA10] or three dimensional abstractions such as the ball and stick visualizations by Silin and Patzek [SP06] would be interesting for an analysis of the network. It would be beneficial to even consider more complicated graphs that include more features, such as the different properties of the pores. This visualization tool as an interactive one would also be an interesting point of future works. The ability to select a pore and display different properties and connections within in the network could allow researchers to more quickly and effectively analyze simulation results.

7 Conclusion

In this thesis, the effects of applying a pore network analysis technique to mesoscopic simulations of pore formations was observed. The results led to both visual and statistical analyses of these developments. It was found that the Maximal Balls method was efficient enough to function as the pore network extraction method. This method was implemented and then applied to three sets of data sourced from simulations of phase inversion processes. Further analysis techniques were applied to determine the efficiency of the implementation. In addition, pore shapes and their development were analyzed and visualized.

The results consisted of a high amount of visual tools to better understand the structure of porous membrane development and point to promising work for the future. While some limits existed within the implementation, they could be accounted for here in hopes of better applied research for future work. Pore network morphology is a useful tool for phase inversion process analysis, and will likely be an important research topic as more works lead to its further understanding.

Citations

- [ACCP70] P A. Choquette, L C. Pray. “Geologic Nomenclature and Classification of Porosity in Sedimentary Carbonates”. In: *American Association of Petroleum Geologists Bulletin* 54 (Feb. 1970), pp. 207–250 (cit. on p. 19).
- [AH17] F. Arand, J. Hesser. “Accurate and efficient maximal ball algorithm for pore network extraction”. In: *Computers & Geosciences* 101 (2017), pp. 28–37 (cit. on pp. 23, 26, 29, 30, 38, 39, 41–43, 57, 60, 62, 63).
- [AKB07] A. S. Al-Kharusi, M. J. Blunt. “Network extraction from sandstone and carbonate pore space images”. In: *Journal of Petroleum Science and Engineering* 56 (2007), pp. 219–231 (cit. on p. 23).
- [ARTW03] R. Al-Raoush, K. Thompson, C. S. Willson. “Comparison of Network Generation Techniques for Unconsolidated Porous Media”. In: *Soil Science Society of America* 67 (2003), pp. 1687–1700 (cit. on pp. 22, 25, 26).
- [ARW05] R. Al-Raoush, C. Willson. “Extraction of physically realistic pore network properties from three-dimensional synchrotron X-ray microtomography images of unconsolidated porous media systems”. In: *Journal of Hydrology* 300 (2005), pp. 44–64 (cit. on pp. 15, 22).
- [CSCS16] S. Claes, J. Soete, V. Cnudde, R. Swennen. “A three-dimensional classification for mathematical pore shape description in complex carbonate reservoir rocks”. In: *Mathematical Geosciences* 48 (2016), pp. 619–639 (cit. on pp. 19, 20, 43).
- [DB09] H. Dong, M. J. Blunt. “Pore-network extraction from micro-computerized-tomography images”. In: *Physical Review E* 80 (2009), pp. 036307/1–036307/11 (cit. on pp. 22, 23, 25, 30).
- [ER03] P. Español, M. Revenga. “Smoothed dissipative particle dynamics”. In: *Physical review. E, Statistical, nonlinear, and soft matter physics* 67 (Mar. 2003), p. 026705. DOI: [10.1103/PhysRevE.67.026705](https://doi.org/10.1103/PhysRevE.67.026705) (cit. on p. 27).
- [GDN+07] A. Gyulassy, M. Duchaineau, V. Natarajan, V. Pascucci, E. Bringa, A. Higginbo, B. Hamann. “Topologically Clean Distance Fields”. In: *IEEE Transactions on Visualization and Computer Graphics* 13 (2007), pp. 1432–1439.
- [GJ+10] G. Guennebaud, B. Jacob, et al. *Eigen v3*. <http://eigen.tuxfamily.org>. 2010 (cit. on p. 44).
- [GKM+15] S. Grottel, M. Krone, C. Muller, G. Reina, T. Ertl. “MegaMol – A Prototyping Framework for Particle-based Visualization”. In: *Visualization and Computer Graphics, IEEE Transactions on* 21.2 (2015), pp. 201–214. ISSN: 1077-2626. DOI: [10.1109/TVCG.2014.2350479](https://doi.org/10.1109/TVCG.2014.2350479) (cit. on pp. 25, 28, 47, 51).

- [GM77] R. A. Gingold, J. J. Monaghan. “Smoothed particle hydrodynamics - Theory and application to non-spherical stars”. In: *Monthly Notices of the Royal Astronomical Society* 181 (1977), pp. 375–389. doi: [10.1093/mnras/181.3.375](https://doi.org/10.1093/mnras/181.3.375) (cit. on p. 27).
- [GMH12] S. Gao, J. N. Meegoda, L. Hu. “Two methods for pore network of porous media”. In: *International Journal for Numerical and Analytical Methods in Geomechanics* 36 (2012), pp. 1954–1970 (cit. on pp. 17, 21).
- [GVTA10] S. Grau, E. Vergés, D. Tost, D. Ayala. “Exploration of porous structures with illustrative visualizations”. In: *Computers & Graphics* 34 (2010), pp. 398–408 (cit. on pp. 17, 21, 26, 27, 64).
- [HBW+13] U. Homberg, D. Baum, A. Wiebel, S. Prohaska, H.-C. Hege. *Definition, Extraction, and Validation of Pore Structures in Porous Materials*. Konrad-Zuse-Zentrum für Informationstechnik Berlin, 2013 (cit. on pp. 21, 26, 43).
- [HHN18] M. Hopp-Hirschler, U. Nieken. “Modeling of pore formation in phase inversion processes: Model and numerical results”. In: *Journal of Membrane Science* 564 (2018), pp. 820–831 (cit. on pp. 15, 20, 21, 24, 27).
- [Hun07] J. D. Hunter. “Matplotlib: A 2D graphics environment”. In: *Computing In Science & Engineering* 9.3 (2007), pp. 90–95. doi: [10.1109/MCSE.2007.55](https://doi.org/10.1109/MCSE.2007.55) (cit. on p. 58).
- [JHE+13] A. P. Jivkov, C. Hollis, F. Etiese, S. A. McDonald, P. J. Withers. “A novel architecture for pore network modelling with applications to permeability of porous media”. In: *Journal of Hydrology* 486 (2013), pp. 246–258 (cit. on pp. 22, 25).
- [JO12] A. P. Jivkov, J. E. Olele. “Novel Lattice Models for Porous Media”. In: *MRS Proceedings* 1475 (2012), imrc11–1475–nw35–o62 (cit. on p. 21).
- [JSS+18] D. Jönsson, P. Steneteg, E. Sundén, R. Englund, S. Kottraval, M. Falk, A. Ynnerman, I. Hotz, T. Ropinski. “Inviwo - A Visualization System with Usage Abstraction Levels”. In: (2018). URL: <http://arxiv.org/abs/1811.12517> (cit. on p. 54).
- [JWC+07] Z. Jiang, K. Wu, G. Couples, M. I. J. van Dijke, K. Sorbie, J. Ma. “Efficient extraction of networks from three-dimensional porous media”. In: *Water Resources Research* 43 (2007), pp. 1–17.
- [KHS+14] M. Krone, M. Huber, K. Scharnowski, M. Hirschler, D. Kauker, G. Reina, U. Nieken, D. Weiskopf, T. Ertl. *EuroVis - Short Paper: Evaluation of Visualizations for Interface Analysis of SPH*. The Eurographics Association, 2014. ISBN: 978-3-905674-69-9. doi: [10.2312/eurovisshort.20141166](https://doi.org/10.2312/eurovisshort.20141166).
- [LC87] W. E. Lorensen, H. E. Cline. “Marching Cubes: A High Resolution 3D Surface Construction Algorithm”. In: *SIGGRAPH Comput. Graph.* 21.4 (Aug. 1987), pp. 163–169. ISSN: 0097-8930. doi: [10.1145/37402.37422](https://doi.org/10.1145/37402.37422). URL: <http://doi.acm.org/10.1145/37402.37422> (cit. on p. 31).
- [LS15] J. Lindblad, N. Sladoje. “Exact Linear Time Euclidean Distance Transforms of Grid Line Sampled Shapes”. In: *Mathematical Morphology and Its Applications to Signal and Image Processing* 9082 (2015), pp. 645–656 (cit. on pp. 30–32, 34–36).
- [MRH00] A. Meijster, J. B. T. M. Roerdink, W. H. Hesselink. *Mathematical Morphology and Its Applications to Image Processing: A General Algorithm for Computing Distance Transforms in Linear Time*. Kluwer Acad., 2000. ISBN: 978-94-011-1040-2. doi: [10.1007/978-94-011-1040-2](https://doi.org/10.1007/978-94-011-1040-2) (cit. on pp. 34, 35).

- [Mon12] J. Monaghan. “Smoothed Particle Hydrodynamics and Its Diverse Applications”. In: *The Annual Review of Fluid Mechanics* 44 (2012), pp. 323–346.
- [Nim13] J. Nimmo. *Porosity and Pore Size Distribution*. Reference Module in Earth Systems and Environmental Sciences, Elsevier Inc., 2013. doi: [10.1016/B978-0-12-409548-9.05265-9](https://doi.org/10.1016/B978-0-12-409548-9.05265-9) (cit. on p. 21).
- [RAF+94] J. Rouquerol, D. Avnir, W. Fairbridge, D. H. Everett, J. H. Haynes, N. Pernicone, J. D. F. Ramsay, K. S. W. Sing, K. K. Unger. “Recommendations for the Characterization of Porous Solids”. In: *Pure and Applied Chemistry* 66 (1994), pp. 1739–1758 (cit. on p. 18).
- [RH10] A. Raoof, S. M. Hassanizadeh. “A New Method for Generating Pore-Network Models of Porous Media”. In: *Transport in Porous Media* 81 (2010), pp. 391–407.
- [RJK06] M. Rezaee, A. Jafari, E. Kazemzadeh. “Relationships between permeability, porosity and pore throat size in carbonate rocks using regression analysis and neural networks”. In: *Journal of Geophysics and Engineering* 3 (2006), pp. 370–376 (cit. on p. 21).
- [SGD06] H. Strathmann, L. Giorno, E. Drioli. *An Introduction to Membrane Science and Technology*. Consiglio Nazionale delle Ricerche, 2006 (cit. on pp. 15, 20, 21).
- [SHCZ18] T. Sweijen, S. M. Hassanizadeh, B. Chareyre, L. Zhuang. “Dynamic Pore-Scale Model of Drainage in Granular Porous Media: The Pore-Unit Assembly Method”. In: *Water Resources Research* 54 (2018), 4193–4213 (cit. on pp. 23, 29).
- [SP06] D. Silin, T. Patzek. “Pore space morphology analysis using maximal inscribed spheres”. In: *Physica A* 371 (2006), pp. 336–360 (cit. on pp. 23, 25, 26, 29, 39, 64).
- [SSA05] A. P. Sheppard, R. M. Sok, H. Averdunk. “Improved Pore Network Extraction Methods”. In: *Society of Core Analysts* (2005), pp. 1–11 (cit. on pp. 17, 22, 23, 25, 26).
- [UMW+12] D. M. Ushizima, D. Morozov, G. H. Weber, A. G. Bianchi, J. A. Sethian, E. W. Bethel. “Augmented Topological Descriptors of Pore Networks for Material Science”. In: *IEEE TRANSACTIONS ON VISUALIZATION AND COMPUTER GRAPHICS* 18 (2012), pp. 2041–2050 (cit. on p. 22).
- [WDBF96] P. van de Witte, P. Dijkstra, J. van den Berg, J. Feijen. “Phase separation processes in polymer solutions in relation to membrane formation”. In: *Journal of Membrane Science* 117 (1996), pp. 1–31 (cit. on p. 15).
- [Wei] E. W. Weisstein. *Covariance Matrix From MathWorld—A Wolfram Web Resource*. Last visited on 09/05/2019. URL: [\url{http://mathworld.wolfram.com/CovarianceMatrix.html}](http://mathworld.wolfram.com/CovarianceMatrix.html) (cit. on p. 44).
- [XBJ16] Q. Xiong, T. G. Baychev, A. P. Jivkov. “Review of pore network modelling of porous media: Experimental characterisations, network constructions and applications to reactive transport”. In: *Journal of Contaminant Hydrology* 192 (2016), pp. 101–117 (cit. on p. 21).
- [ZCSJ07] B. D. Zdravkov, J. J. Cermák, M. Sefara, J. Janku. “Pore classification in the characterization of porous materials: A perspective”. In: *Central European Journal of Chemistry* 5(2) (2007), pp. 385–395 (cit. on pp. 17, 19, 21).

- [ZFS+18] H. Zhang, S. Frey, H. Steeb, D. Uribe, T. Ertl, W. Wang. “Visualization of Bubble Formation in Porous Media”. In: *IEEE Transactions on Visualization and Computer Graphics* 25 (2018), pp. 1060–1069 (cit. on p. 22).
- [ZHHN18] C. Zander, M. Hopp-Hirschler, U. Nieken. “Mesoscopic simulation and characterization of the morphological evolution in phase separating fluid mixtures”. In: *Computational Materials Science* 149 (2018), pp. 267–281 (cit. on pp. 15, 20, 22, 27).
- [ZL08] S. Zafeiriou, N. Laskaris. “On the Improvement of Support Vector Techniques for Clustering by Means of Whitening Transform”. In: *IEEE Signal Processing Letters* 15 (2008), pp. 198–201. ISSN: 1070-9908. DOI: [10.1109/LSP.2007.914949](https://doi.org/10.1109/LSP.2007.914949) (cit. on p. 45).
- [ZZY00] C. Zhao, X. Zheo, Y. Yue. “Determination of pore size and pore size distribution on the surface of hollow-fiber filtration membranes: a review of methods”. In: *Desalination* 129 (2000), pp. 107–123.
- [(ht) user3731622 (<https://math.stackexchange.com/users/341117/user3731622>). *Check if a point is within an ellipse*. Mathematics Stack Exchange. URL:<https://math.stackexchange.com/q/1793612> (version: 2016-05-21). eprint: <https://math.stackexchange.com/q/1793612>. URL: <https://math.stackexchange.com/q/1793612> (cit. on p. 45).

All links were last followed on 09 May 2019.

Declaration

I hereby declare that the work presented in this thesis is entirely my own and that I did not use any other sources and references than the listed ones. I have marked all direct or indirect statements from other sources contained therein as quotations. Neither this work nor significant parts of it were part of another examination procedure. I have not published this work in whole or in part before. The electronic copy is consistent with all submitted copies.

place, date, signature