

Computer vision frameworks for physics-based simulation of liquids and  
solids

by

Chendi Cao

B.S., Kansas State University, 2013

M.S., Kansas State University, 2016

---

AN ABSTRACT OF A DISSERTATION

submitted in partial fulfillment of the  
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science  
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2021

# Abstract

Simulating and visualizing fluid and solid materials in agricultural domains is an important and challenging problem in scientific computing and computer vision. Modern seed breeding programs require the ability to analyze seeds efficiently to be useful. Even simple measures such as volume and density can be challenging to compute efficiently with modest equipment. The dynamics of liquid and soil materials involve significant deformation during storm flows and require sophisticated numerical algorithms to achieve sufficient accuracy and visual realism. This dissertation focuses on extending volume carving techniques to measure seed volume and to create a new Material Point Method (MPM) models and finite volume models to simulate solids and fluids for dam safety analysis and visualization.

This dissertation makes the following major contributions: The first is to create a novel framework for the design and analysis of computer experiments. The framework is applied to perform efficient dam breach and internal erosion analysis on a large number of structures. Given historical dam breach or design data input, the modeling framework can also be used to conduct sensitivity analysis to determine which parameters make the most impact on the resulting dam erosion. The second contribution is to develop new models for numerical simulation of dam erosion by combining fluid flow models developed using Computational Fluid Dynamics (CFD) with new dam erosion models using the Finite Element Method (FEM). A new model that combines fluid flow and erosion simulation into a single model is also developed using the Material Point Method (MPM). The third contribution is to build a comprehensive image capture and processing framework for seed property analysis. Rather than having a human manually measure seed

properties such as length, width, thickness, and volume, the framework can automatically analyze a set of images from multiple angles and calculate the physical measurements for single seed samples. Finally, image analysis is extended using deep learning to increase the accuracy of rice image classification.

The proposed frameworks are suitable for larger scale and more dynamic data in both dam safety and agricultural domains. They are also useful for computer animation in developing physics-based special effects for the animation of dam erosion. Previous work on MPM has resulted in models used in animation for Disney Studios, and the new models proposed could be used for accurate animation of fluid flows and dam erosion. Finally, the combination of image analysis algorithms and deep learning has many applications in the biomedical domain as well as the agricultural domain.

Computer vision frameworks for physics-based simulation of liquids and  
solids

by

Chendi Cao

B.S., Kansas State University, 2013

M.S., Kansas State University, 2016

---

A DISSERTATION

submitted in partial fulfillment of the  
requirements for the degree

DOCTOR OF PHILOSOPHY

Department of Computer Science  
Carl R. Ice College of Engineering

KANSAS STATE UNIVERSITY  
Manhattan, Kansas

2021

Approved by:

Major Professor  
Dr. Mitchell L. Neilsen

# Copyright

© Chendi Cao 2021.

# Abstract

Simulating and visualizing fluid and solid materials in agricultural domains is an important and challenging problem in scientific computing and computer vision. Modern seed breeding programs require the ability to analyze seeds efficiently to be useful. Even simple measures such as volume and density can be challenging to compute efficiently with modest equipment. The dynamics of liquid and soil materials involve significant deformation during storm flows and require sophisticated numerical algorithms to achieve sufficient accuracy and visual realism. This dissertation focuses on extending volume carving techniques to measure seed volume and to create a new Material Point Method (MPM) models and finite volume models to simulate solids and fluids for dam safety analysis and visualization.

This dissertation makes the following major contributions: The first is to create a novel framework for the design and analysis of computer experiments. The framework is applied to perform efficient dam breach and internal erosion analysis on a large number of structures. Given historical dam breach or design data input, the modeling framework can also be used to conduct sensitivity analysis to determine which parameters make the most impact on the resulting dam erosion. The second contribution is to develop new models for numerical simulation of dam erosion by combining fluid flow models developed using Computational Fluid Dynamics (CFD) with new dam erosion models using the Finite Element Method (FEM). A new model that combines fluid flow and erosion simulation into a single model is also developed using the Material Point Method (MPM). The third contribution is to build a comprehensive image capture and processing framework for seed property analysis. Rather than having a human manually measure seed

properties such as length, width, thickness, and volume, the framework can automatically analyze a set of images from multiple angles and calculate the physical measurements for single seed samples. Finally, image analysis is extended using deep learning to increase the accuracy of rice image classification.

The proposed frameworks are suitable for larger scale and more dynamic data in both dam safety and agricultural domains. They are also useful for computer animation in developing physics-based special effects for the animation of dam erosion. Previous work on MPM has resulted in models used in animation for Disney Studios, and the new models proposed could be used for accurate animation of fluid flows and dam erosion. Finally, the combination of image analysis algorithms and deep learning has many applications in the biomedical domain as well as the agricultural domain.

# Table of Contents

List of Figures . . . . .	x
List of Tables . . . . .	xiii
Acknowledgements . . . . .	xiv
1 Introduction . . . . .	1
1.1 Background . . . . .	2
1.1.1 Physics-based simulation of fluids . . . . .	2
1.1.2 Physics-based simulation of solids . . . . .	4
1.2 Problem Statement . . . . .	5
1.2.1 Computer vision frameworks in dam safety . . . . .	5
1.2.2 Computer vision frameworks in agriculture . . . . .	6
1.3 Contribution and Dissertation Overview . . . . .	8
2 Computer Vision in Dam Safety - Risk Assessment Tool . . . . .	10
2.1 Parameterized Simulation Models (WDT) . . . . .	11
2.2 Parameter Studies . . . . .	11
2.3 Uncertainty and Sensitivity Analysis . . . . .	12
2.4 Risk Assessment Tool . . . . .	12
2.5 Conclusion . . . . .	19
3 Computer Vision in Dam Safety - Computer Simulation Models . . . . .	21
3.1 Computational Fluid Dynamic Method (CFD) . . . . .	22
3.2 Discrete Element Method (DEM) . . . . .	29

3.3	Smoothed Particle Hydrodynamic (SPH)	29
3.4	Coupled Model CFD-DEM, DEM-SPH	30
3.5	Material Point Method (MPM)	30
3.6	Deformation Gradient	32
3.7	Elastoplasticity	33
3.8	Constitutive Model: Soil Elastoplasticity	34
3.9	Momentum Exchange	36
3.10	Cohesion and Saturation	36
3.11	Discretization	37
3.12	Conclusion	47
4	Computer Vision in Agriculture	49
4.1	Computer Vision using Two Images	50
4.1.1	Single seed volume measurement	50
4.1.2	Slicing model for volume measurement	52
4.2	Computer Vision using Multiple Images	67
4.2.1	Volume carving framework for volume measurement	69
4.2.2	Turntable framework setup	70
4.3	Computer Vision using Deep Learning	90
4.3.1	Image classification model	90
4.3.2	Rice image classification using ResNet	92
4.4	Conclusion	96
5	Conclusions and future work	98
	Bibliography	102

# List of Figures

2.1	Iterative analysis with Dakota and models. . . . .	13
2.2	WinDAM/BREACH+Dakota GUI . . . . .	17
2.3	Surface Plot. . . . .	18
2.4	Scatter Plot. . . . .	19
3.1	Dam Region Simulation Result, Top(Left), Bottom(Right). . . . .	25
3.2	Hydrograph Comparison Plot. . . . .	27
3.3	Performance comparison plot for different number of cores. . . . .	29
3.4	Multi-phase deformation combining water and soil. . . . .	33
3.5	The deformation of a material. . . . .	33
3.6	Compression and tension for elastic and granular material. . . . .	35
3.7	Overtopping dam breach simulation with different cohesion at timestamp $t_0$ and $t_0 + \Delta t$ . . . . .	38
3.8	MPM algorithm with two grid multi-species. 1.(a)-(b) Transfer to grids: The mass and momentum of each species are transferred to its corresponding grid. 2.(b)-(c) Update grids momentum: The coupled water and soil grid velocities is updated. 3.(c)-(d) Update particles: All particle states, including the momentum, velocity, and cohesion based on saturation are updated. . . . .	39
3.9	Overtopping dam breach simulation with empty(left) and full(right) reservoir. . . . .	42
3.10	WinDAM C 05-HR-OvertopBreach(A) and our simulation(B) method comparison. . . . .	45

4.1	Slice segmentation method for seed model . . . . .	53
4.2	Main-view captured image and side-view captured image . . . . .	53
4.3	Two camera setup. Top: main-view, Left: side-view. Set at equal distance $d$ from seed . . . . .	53
4.4	The framework setup including a turntable, a light source, one top color camera, one side monochrome camera, a seed feeder, and an electric con- troller . . . . .	56
4.5	Wheat seed (top-view, color camera). A: Source image, B: Binary image after thresholding, C: Bounding box with detected contour, D: Sliced image	59
4.6	Wheat seed (side-view monochrome camera). A: Source image, B: Binary image after thresholding, C: Bounding box with detected contour, D: Sliced image . . . . .	60
4.7	Wheat sample with volume measurement results . . . . .	61
4.8	Milo sample with different numbers of slices $N = 5$ (A), $N = 10$ (B), $N = 20$ (C) . . . . .	63
4.9	Volumetric error for different slice numbers of two ceramic spheres. Sphere 1 diameter = 3.18 mm; Sphere 2 diameter = 6.35 mm. . . . .	63
4.10	Selected camera mount 3D models . . . . .	72
4.11	Turntable setup with 3D printed camera mounts in STL model . . . . .	74
4.12	2D view for volume carving method using voxel grid . . . . .	77
4.13	3D sphere reconstruction using parallel projection (left) or central projec- tion (right) . . . . .	78
4.14	Single seed 3D reconstruction framework . . . . .	79
4.15	3D reconstruction profile pictures for soybean . . . . .	80
4.16	3D reconstruction model for soybean . . . . .	80
4.17	Comparing number of images $N$ for 3D volume reconstruction using refer- ence ceramic ball . . . . .	83

4.18 Error rate when using different number of images (N) . . . . .	84
4.19 3D reconstruction and volume measurement for wheat, corn and soybean seeds . . . . .	84
4.20 Large-scale voxel reconstruction using reference ceramic ball . . . . .	85
4.21 Box plot, A: Glass beads method, B: Two images method (TT2Cam), C: Proposed method (N=36) . . . . .	87
4.22 Tukey comparison test: A: Glass beads method, B: Two images method (TT2Cam), C: Proposed method (N=36) . . . . .	88
4.23 Residual learning: a building block . . . . .	92
4.24 ResNet34 architecture . . . . .	93
4.25 Alkali spreading value from 2 to 7 . . . . .	94

# List of Tables

2.1	Silty sand (SM) and Clay (CL) bare material properties. . . . .	15
3.1	Input hydrograph data . . . . .	24
3.2	Run Time table for OpenFOAM simulation. . . . .	28
3.3	table of notation. . . . .	34
3.4	Condition for the cohesive dam's overtopping failure cases and simulation cases. . . . .	44
4.1	Camera calibration settings . . . . .	55
4.2	Volume measurement of wheat samples . . . . .	65
4.3	Camera calibration setting . . . . .	75
4.4	Volume data computed for reference spheres . . . . .	81
4.5	Summary data for reference spheres . . . . .	81
4.6	Volume measurement for reference ceramic ball using different voxel sizes	85
4.7	Summary data for reference spheres . . . . .	89
4.8	Model training top one accuracy using different ResNet architectures . . .	96

# Acknowledgments

I would like to express my sincere gratitude to my major advisor Dr. Mitchell L. Neilsen for the continuous support for my Ph.D. study and research. His insight and expertise greatly helped me grow as a researcher. Without his patience and encouragement, this thesis would never have materialized.

I also thank Professor Dr. Daniel Andresen, Dr. Torben Amtoft, and Dr. Weixing Song for serving on my thesis committee and providing suggestions on improving and extending my work.

My appreciation goes to all graduate students, faculties, friends from the K-State Computer Science Department for their help and insights. I would like to express my gratitude to Chaney Courtney, Dan Wagner, Friday James, Kai Zhao, Siddharth Amaravadi, Shanshan Wu, and Venkat Margapuri.

I would like to thank our collaborators from USDA. In particular, I would like to thank Dr. Daniel Brabec, Dr. Paul Armstrong, and Sophia Grothe for sharing their knowledge and facilitating my work on single seed image analysis.

I would especially like to thank my family for the love, support, and constant encouragement I have gotten over the years. I would deeply thank my parents and my wife. Their trust, patience, and encouragement have enabled me to reach this far.

# Chapter 1

## Introduction

Starting from its roots with the first neural network project in the 1950s, computer vision applications embedded with artificial intelligence have seen explosive growth. As the internet matured in the 1990s, large sets of images became more readily available online for analysis. This enabled rapid advances in computer vision algorithms that make it possible for machines to identify specific objects in photos and videos.

Computational technologies are widely expanding in diverse scientific domains. There is growing interest in integrating computer vision algorithms into other scientific domains. In the field of hydrology and dam safety, earthen embankment dams are a crucial part of the world's infrastructure. Dam risk assessment helps mitigate the catastrophic impact of dam failures and flooding. Earthen dam failure analysis has been conducted by many facilities using an empirical approach. In the past, computational resources were so limited, researchers built physical dam models to simulate the dam break process and analyzed the results to derive computational models. Physical models represent an important component in model design, but they are very expensive and time consuming to build. Typically, only a few dam structures can be built and analyzed each year. On the other hand, hundreds or thousands of abstract models can be simulated using computational models. This increases the velocity and accuracy of model creation and evolution for different scenarios. In Chapter 2, a novel risk assessment tool is described to facilitate

the analysis of hundreds of dams.

In the field of seed breeding, researchers also use physical measurement. They typically use calipers to measure single seeds and conducted property analysis based on this information. They might also use bead displacement to estimate seed volume without destroying the seed. There are some limitations to accuracy using physical property analysis. Developing efficient computer vision frameworks to advance productivity becomes an important and challenging problem in both of these domains.

Therefore, this thesis addresses the challenges of improving the productivity for various scientific domains in two ways: The first is through establishing modeling frameworks for the dam breach and internal erosion analysis. Given a historical dam breach or experimental data input, our modeling framework can conduct parameter studies and sensitivity analysis to see which parameter(s) make the most impact on the erosion of a dam structure. Also, our modeling frameworks can detect when and where erosion is occurring. The second is through building the image frameworks for seed property analysis. Rather than having a human manually measure the seed properties including length, width, thickness, and volume, we can automatically derive a set of measurements from our image analysis framework. The goal of this thesis is to study the power of computational modeling and computer vision, and their application in dam safety and agricultural domains.

## **1.1 Background**

### **1.1.1 Physics-based simulation of fluids**

For the physics-based simulation of fluids in dam safety, there is growing interest in analyzing the impact of fluid flow on different types of erosion including overtopping of the dam, internal erosion through the dam, and erosion of the auxiliary (also called emergency) spillway. A hydrograph is just a function describing the amount of discharge of water that results as a function of time. The analysis should provide information about

the integrity (ability to resist dam failure) and stability (ability to resist erosion) of the dam during the passage of different types of hydrographs. Passage of a stability-design hydrograph (SDH) should not cause any erosion on the downstream face of the dam or spillway; whereas, the passage of a freeboard hydrograph (FBH) should not cause the dam to fail. For example, a stability-design hydrograph might be the result of a 2-year storm; that is, a storm that we expect to occur once every two years, but a freeboard hydrograph might be the result of a 100-year storm.

Fluid simulations generally use one of two approaches: the Eulerian approach and the Lagrangian approach. The Lagrangian approach models individual particles and calculates the trajectory of each particle separately, but the Eulerian approach considers a collection of particles and calculates the overall diffusion and convection of the set of particles. The most popular fluid simulations using the Eulerian approach is called Computational Fluid Dynamics (CFD). Computational Fluid Dynamics(CFD) uses numerical analysis and data structures to analyze and solve problems involving fluid flow. The CFD modelling approach for dam-break simulation study has been extensively investigated<sup>1,2,3</sup>. In our previous study<sup>4,5</sup>, we explored using the Finite Volume Method (FVM) for dam breach simulation using the open-source CFD toolbox OpenFOAM. Cao and Neilsen simulated the flooding due to overtopping without considering the deformation of the solid body dam structure<sup>5</sup>.

Unlike the Eulerian approach, there is no numerical dispersion for the Lagrangian approach. Therefore, Lagrangian techniques are useful for simulations when large topological changes occur in the fluid interface. Smoothed Particle Hydrodynamics (SPH) is a well-known method in the computer simulation field. Instead of considering the dam embankment structure as a solid body material in<sup>4,5</sup>, the granular material like soil can be represented as either continuum or a set of individual particles using SPH. SPH modeling approach is used to simulate large surface flow<sup>6,7</sup>, and dam break flows<sup>8,9,10</sup>. In addition, the Discrete Element Method (DEM) is a very popular particle-based system used for fluid simulation. Often, CFD has been coupled with DEM in many studies and engineering problems including rockslides<sup>11,12</sup>, granular flow in water reservoirs<sup>13,14</sup>, and

fluid-particle interaction in dam break<sup>15;16</sup>. Other than CFD-DEM, the coupled model SPH-DEM is encountered in many multi-species simulation studies as well. Rungjiratananon et al. simulate sand-water interaction in real-time using a hybrid SPH-DEM model<sup>17</sup>. Wu<sup>18</sup> and Canelas<sup>19</sup> used the SPH-DEM method to simulate multi-phase free surface flow. Lenaerts<sup>20</sup> also proposed a dynamics framework for simulation of both fluid and porous granular material using SPH-DEM.

A hybrid approach, called the Material Point Method (MPM), was introduced to handle topological change by multi-species interaction like water and porous soil<sup>21</sup>. Bandara et al.<sup>22,23</sup> introduced the soil deformation and pore fluid flow using MPM and also the landslides in unsaturated soil slopes. Tampubolon et al.<sup>24</sup> simulate the interaction of sand and water mixtures using MPM and obtained encouraging results. For the porous material property, Klár<sup>25</sup> used the improved Drucker-Prager plastic flow model with volume correction. For the MPM implementation, Arduino<sup>26</sup> and Jassim<sup>27</sup> examined that momentum exchange using the two-grid MPM for the multi-species interaction. In this thesis, a novel, new dam breach simulation framework has been developed using the Material Point Method (MPM) to simultaneously simulate fluid flow and dam erosion for both analysis and computer animation.

### 1.1.2 Physics-based simulation of solids

For the physics-based simulation of solids, the focus is on the physical property analysis of seed samples. The principle goal for seed property analysis is to identify seed quality by measuring physical and physiological factors that regulate the performance of seeds. Among those physical factors, seed volume and density are among the most crucial aspects to measure since they are highly correlated with seed functionality and quality. There are several existing seed physical property analysis models using image processing including 2D affordable scanning<sup>28-30</sup> and commercial software designed for different seed types<sup>31;32</sup>. These existing models calculate typical parameters including width, length, and area from the 2D images. They can be used to analyze different seed

types such as rice, soybeans, and barley.

Other than these specific seed image processing frameworks which focus on measuring physical factors, general image processing models can be introduced for volume measurement. From a geometric perspective, there are stereo-based techniques, space-carving methods, and a newly developed Volumetric and Multi-View Convolutional Neural Network (CNN) methods. Although the CNN methods may use fewer images, effective solutions typically require multiple images.

To design an efficient framework for seed volume measurement, two approaches are further investigated; the first one is through building a volume measurement framework using only two images. The second one is by setting up a multiple-image capture framework to reconstruct the 3D model of the seed and compute the volume measurement using multiple images.

## **1.2 Problem Statement**

### **1.2.1 Computer vision frameworks in dam safety**

The goal of developing a computer vision framework for dam safety is to simulate the dam erosion processes and conduct both parameter studies and sensitivity analysis. Input files already generated by design engineers are automatically parameterized to enable efficient analysis of a range of material properties, hydrologic properties such as peak and duration of the hydrograph, and dam dimensions such as auxiliary spillway width, height, etc. This allows designers to consider a range of design options quickly. The tool has also been used by other research groups at USDA and the University of Illinois - Chicago.

Typically, in the field of dam safety simulation, two approaches for discretization have typically been used: Eulerian or Lagrangian. For the Eulerian based methods, quantities of interest are in fixed locations or fixed grids like the application using CFD and CFD-DEM mesh modeling. For the Lagrangian based method, quantities of interest are

attached to the materials including SPH, SPH-DEM, and MPM particle-based method. Many researchers have experimented with hybrid grid and particle methods. Among all these modeling methods, the Material Point Method (MPM) is tested and selected because MPM combines aspects of both types of discretization.

In this thesis, a novel, multi-species material point method simulates the interaction between soil and water. The wet soil transitions are considered from cohesive grains to flowing as water saturation increases. The overtopping dam erosion simulation model is created using multi-species particles and the two-grid material point method. For the purpose of validation, the MPM simulation model of overtopping dam erosion is compared with available experimental data and results from other physical-based models including WinDAM<sup>33</sup>.

### **1.2.2 Computer vision frameworks in agriculture**

The goal of developing a computer vision framework in agriculture is to accurately measure single seed volume and density. The physical properties of seed include length, width, thickness, and volume. Seed volume is among the most important agronomic traits because seed quality and seed traits are heavily dependent upon seed volume. The water displacement method is the most common way to measure the volume of the object. However, due to the tiny size of the single seed, these testing procedures can be time-consuming, labor-intensive, and subject to human error. In addition, the seed may be destroyed by absorbing water. Another approach is to measure seed displacement in small beads, but this approach is not as accurate and subject to overestimate the actual volume. To solve the seed volume measurement problem, many image analysis methods have been developed.

Neilsen, et. al<sup>34</sup> introduced a dynamic image processing method to identify seed contours and calculate individual seed length, width, and total seed count by analyzing dynamic video footage. Tanabata, et. al<sup>35</sup> developed a software package, called SmartGain, to provide high-throughput phenotyping from a single image and achieve

seed size information. Roussel<sup>36</sup> conducted a volume curving model and performed 3D shape reconstruction and achieved a great result for the volume measurement of a single seed. Fıratlıgil-Durmuş<sup>37</sup> applied the oblate spheroid model to get the size properties of legume seeds by image analysis. Razavi<sup>38;39</sup> developed a computer vision system to calculate the physical properties for basil seed and wild sage seed. Sabliov<sup>40</sup> tackles the image processing method to determine the volume of agricultural products like lemon, peach, and egg using a single camera. Cervantes<sup>41</sup> focuses on the parameters used to describe seed shape. Pedersen<sup>42</sup>'s work compares the proposed SKWCS system with traditional laboratory measurements using 16 single sorghum kernels.

However, there are several limitations in the previous studies, Neilsen<sup>34</sup>, SmartGain<sup>35</sup>, Sabliov<sup>40</sup>, Cervantes<sup>41</sup>, and Pedersen<sup>42</sup>'s work did not focus on seed property analysis like seed volume measurement. Roussel<sup>36</sup>'s work requires a high-cost professional camera, and it is not capable of dealing with multiple seeds at the same time. Amiryousefi, Fıratlıgil-Durmuş, and Razavi's work only design the computer vision system specific for a single type of seed sample, the pomegranate seed<sup>43</sup>, lentil seed<sup>44</sup>, legume seed<sup>37</sup>, wild sage seed Razavi<sup>38</sup> and basil seed<sup>39</sup>. These computer vision systems cannot be used to calculate volume measurements for other seed samples.

To the best of our knowledge, no rapid, accurate, and universal seed imaging framework has been designed for general-purpose seed volume measurement. To solve this problem, a seed property analysis framework is developed. The design goal of this framework is to accurately measure each seed volume on a low-cost turntable, with modest cameras and modest computing requirements.

After the implementation of the framework, some research questions arise. Can we find a better reference method to determine the true measurement of the seed volume? Can we also achieve more precise volume measurement by taking multiple images? To answer these research questions, we further investigate several approaches for volume measurement using multiple images. Among them, in order to achieve reasonable quality 3D reconstructions and volume measurement, we investigate volume carving, shape from the silhouette method<sup>45-47</sup> for 3D seed shape reconstruction.

A novel multiple image capture frameworks for the single seed volume measurement is designed and developed using the 3D reconstruction volume carving method. Although existing software and models have been developed in the literature and in practice, we propose a complete, end-to-end framework for volume measurement of a single seed. Our proposed work achieves high accuracy from multiple images automatically acquired and analyzed.

### 1.3 Contribution and Dissertation Overview

To summarize, the dissertation focuses on improving productivity in various scientific domains. It achieves this goal through two main approaches, the first is to create a novel framework for the design and analysis of computer experiments. The second is to build a comprehensive image capture and processing framework for seed property analysis.

The overview of the dissertation is as follows:

1. In Chapter 2, we consider the risk assessment problem and design an integrated framework by coupling traditional dam safety software, including WinDAM and BREACH, with DAKOTA. Our framework is able to provide visualization results and conduct sensitivity analysis for dam safety. This work was published previously<sup>34</sup>.
2. In Chapter 3, we develop a dam breach model using OpenFOAM to generate fluid flows and couple the flows step-wise with a Finite Element Model (FEM) used to model the erosion processes on the dam. Step-wise, as the dam profile changes, updates are fed back to update the CFD model to change the flow. This work has been published<sup>48</sup>. Due to the lack of detailed information about the topological change during the process, we present a Material Point Method (MPM) to simulate the overtopping dam breach using multi-species particles.
3. In Chapter 4.1, we present a turntable setup for measuring the single seed volume using a two image setup and our framework achieves great accuracy on the refer-

ence sphere ball. This work and paper are ready to be submitted to the related journal.

4. In Chapter 4.2, we propose a new 3D reconstruction framework for the efficient single seed volume measurement by taking multiple images and 3D reconstruction using the volume carving method. The proposed framework is an end-to-end product which simplifies the all process and easy to use. The related work has been accepted in CSCI 2020<sup>49</sup>.
5. In Chapter 4.3, we consider the classification problem for the rice image data. The rice images are labeled from 2 to 7 based on the dissolved level in the alkali solution. For our trained classifier, the proposed work can gain great results for classifying the different stages of rice images.
6. We conclude with future directions in Chapter 5.

## Chapter 2

# Computer Vision in Dam Safety - Risk Assessment Tool

Earthen embankment dams are a crucial part of the world's infrastructure and provide electricity, irrigation, flood control, and water supply to millions of people. Embankment dam risk assessment helps mitigate the catastrophic impact of dam-break flooding. Extreme weather, construction deficiencies, and other circumstances have led to many major dam failures in the last two centuries. These have resulted in significant loss of life, as well as economic, social, and environmental damage.

In the past, earthen dam failure experiments have been conducted by many facilities using an empirical approach. Computational resources were so limited during that time, researchers had to build a physical dam model to simulate the dam break process. With the explosive growth of computer vision frameworks, scientific computing and computer graphics can be integrated with dam engineering principles, and mathematical statistics models to obtain quantitative estimates of the embankment dam risk of erosion or failure. These conclusions can be obtained for a dam in its existing condition and for a range of risk reduction alternatives to explore equitable, economical, and rational resource allocations for structural safety, and practical and emergency planning to improve public safety management.

## 2.1 Parameterized Simulation Models (WDT)

The goal of the laboratory embankment breach testing program is to develop and improve the ability to model potential embankment failures numerically for risk assessment and emergency planning purposes. Parameterized simulation models (\*.WDT = WinDAM Template files) are automatically derived from user-input WinDAM C (\*.WDC files). They can be used to aid in the prediction of hydraulic and erosion processes that are associated with the flow over an earthen dam or through an internal erosion channel. Parametric simulation models become more feasible, while further developments reduce the reliance on physical models and improve their accuracy and usability. Other existing parametric models can be generated from models used to analyze overtopping-induced embankment failures, in addition to those used in this study: WinDAM C<sup>50</sup>, BREACH<sup>51</sup>, DL Breach<sup>52</sup> and EMBREA.

## 2.2 Parameter Studies

Parameter studies can specify a range of input parameters and compute the corresponding output which can be displayed in text or graphical format. DAKOTA<sup>53</sup> is a software toolkit developed by engineers at Sandia National Laboratories to provide a flexible, extensible interface between analysis codes and iterative systems analysis methods. DAKOTA contains optimization algorithms to help with parameter study capabilities. It provides an ideal platform for providing iterative analysis of the inputs selected by a user. DAKOTA currently supports four types of parameter studies:

1. Vector parameter studies
2. List parameter studies
3. Centered parameter studies
4. Multi-dimensional parameter studies

This framework supports the most general parameter study, the multi-dimensional parameter study. Any number of parameters can be specified over a range or fixed.

## 2.3 Uncertainty and Sensitivity Analysis

The goal of uncertainty analysis is to obtain a better understanding of the probable range of outputs given that there is a certain amount of uncertainty in the input. In particular, based on uncertain inputs, determine the distribution function (uncertainty) of the outputs and probabilities of failure (reliability metrics); identify the statistical measures (mean, variance, etc.) of the outputs; and identify the inputs whose variance contributes most to variance in the outputs (global sensitivity analysis)<sup>54</sup>. Sensitivity analysis is the study of how the uncertainty in the output of a mathematical model can be divided and allocated to different sources of uncertainty in its inputs. Sensitivity analysis can determine which inputs have the most influence on the output. Based on the uncertain input, sensitivity analysis can identify the statistical measures of the outputs and investigate the inputs whose variance contributes most to the variance in the outputs.

## 2.4 Risk Assessment Tool

Windows<sup>TM</sup> Dam Analysis Modules (WinDAM) is a set of modular software components that can be used to analyze overtopped earthen embankments and internal dam erosion<sup>50</sup>. The National Weather Service BREACH model is a physically-based model to predict the discharge hydrograph resulting from a breached earthen dam<sup>51</sup>. Dakota is an extensive software framework for design exploration and simulation<sup>53</sup>.

This section describes a new risk assessment tool developed using Dakota and WinDAM/BREACH. The integrated water resource risk assessment tool Dakota + WinDAM/BREACH is designed to fully integrate the simulation models in WinDAM/BREACH with the uncertainty quantification, sensitivity analysis, and parameter studies capabilities in DAKOTA. This tool interactively guides user input, parses the results, and invokes the sampling and

simulation models in the background. This tool also allows a user to conduct parameter studies and specify the inputs to be analyzed based on a list of probability distribution functions. The new tool has been designed to conduct coupled analysis over a wide range of input parameters including both structural properties and flow properties.

Hydraulic engineers are comfortable with the tools of the trade to develop Computational Models which are typically used to evaluate a single set of model input parameters defining both structural and hydraulic properties. If a user wants to determine which input parameters are the most important or influential in determining peak outflow, flow duration, or the amount of dam erosion that may result, the user is faced with the daunting task of executing the model many times after simply tweaking the input parameters and recording the output which results. The goal of this new framework is to minimize this tedium.

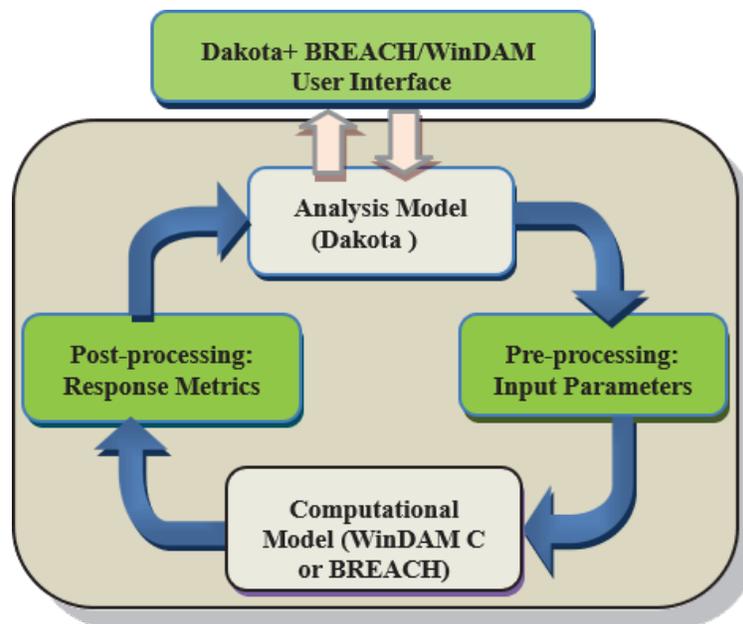


Figure 2.1: Iterative analysis with Dakota and models.

Dakota supports several different automated options for the Design and Analysis of Computer Experiments (DACE):

- Sensitivity Analysis (SA) - determine which inputs have the most influence on the output.

- Uncertainty Analysis (UA) -compare the relative importance of model input uncertainties on output.
- Uncertainty Quantification (UQ) - from a set of probability distributions on the inputs, propagate them through the model to obtain distributions on the outputs.
- Parameter Studies – specify a range of input parameters and compute the corresponding output to be displayed in text or graphical format.

The Analysis Model, shown in figure 2.1, is generated by the framework using the Dakota modeling language. An intuitive user interface guides the user to specify the input required to construct the Analysis Models, and a collection of parsers automatically convert input parameters generated by Dakota into the fixed-column format expected by the Computational Models; that is, they pre-process the models to generate input files for the model simulators. After the simulation, they post-process the output generated by the simulators to extract the key parameters of interest<sup>34</sup>. These parameters are then passed to Dakota to complete a single iteration as shown in figure 2.1. Thus, the framework, consisting of the green boxes, provides the glue to enable efficient analysis.

WinDAM/BREACH + Dakota is a framework created by Neilsen and Cao to extend the capabilities of both. It is designed to integrate the simulation models in WinDAM and BREACH with the uncertainty quantification, sensitivity analysis, and parameter studies capabilities in Dakota. In a Dakota input file, there are six specification blocks that may appear in the Analysis Model, and the blocks are identified using the keywords: environment, variables, interface, responses, model, and method. A model contains a set of variables, an interface, and a set of responses, and an iterator operates on the model to map the variables into responses using the interface. WinDAM C is extended to automatically generate Dakota input files for simple parameter studies on material properties. Then, Dakota is invoked to iterate on the WinDAM simulation models, or vice versa, as needed to generate output. Instead of having WinDAM drive the analysis, we can also use Dakota to drive the analysis iteratively as shown in figure 2.1.

A synthetic data set is developed to evaluate differently computational models over a range of different material properties and different erosion models ranging from a sand material which has high erodibility, to a clay material with low erodibility as shown in table 2.1.

In addition to a range of material properties, a user may want to evaluate a range of different dam configurations and sizes. For this synthetic analysis, a typical range of dam sizes is used ranging from a dam height of 4 feet to a maximum of 128 feet, with appropriate dam crest, etc., and internal erosion originating at 1/4th of the dam height. For convenience, we introduce a scale variable ranging from 0 to 5 to represent dam heights of  $2^{(scale+2)}$  ft. Not surprisingly, stronger materials with less hydraulic flow results in much less erosion, in fact, the same inflow resulted in no failure even after 72 hours for the same physical experiment conducted at the USDA Hydraulic Engineering Research Unit just outside Stillwater, Oklahoma.

	Silty sand(SM)	Clay(CL)
Total Unit Weight (Tw) ( $lb/ft^3$ )	120	130
Undrained Shear Strength (Us) ( $lb/ft^2$ )	70	0.3
Erodibility (Kd) ( $ft/h)/(lb/ft^2$ )	300	1700
Critical Shear Stress ( $lb/ft^2$ )	0	0.2

Table 2.1: Silty sand (SM) and Clay (CL) bare material properties.

Then, the framework automatically generates the initial template of a Dakota input control file and template files with placeholders for the parameters to be varied. Finally, the user can simply click on the Run button to save the Dakota input analysis file, and fire up Dakota and WinDamSim (the Computational Model) to perform the analysis in this case using iterative analysis on  $6 \times 6 \times 6 = 216$  input data sets. The values for erodibility Kd and undrained shear strength Us are computed by Dakota sampling on fixed intervals at the end of each partition. Custom parsers are used to extract and format the data generated by Dakota, and a Dakota double-precision pre-processing tool, dprepro, is used to replace the tokens for Kd , Us, and Scale in the template file with the values in the parameter input file to generate a simulation input file to be processed by WinDamSim.

A portion of a WinDAM C input file used in the synthetic study is shown below.

```
WINDAM 1/1/2009
OPTION SIMPLE BARESOIL NOPS INTERNAL
IEMODEL 2 120 70 1700 0.2
HYD 0.0167 0 1C 10
0.02
..
ENDtable
CRESTPRFL
0 8
900 8
ENDtable
STRROUTE 6 0 ELEV
```

The corresponding template file is shown below. Note that the fields denoting the values for erodibility,  $K_d$ , and undrained shear strength,  $U_s$ , are replaced with tags  $K_d$  and  $U_s$ , respectively. Also,  $Scale$  is used to allow for the range of different dam geometries to be specified. The values for  $Scale$  range from 0 to 5; e.g., dam crests of 4 feet to  $4 \times 25 = 128$  feet.

```
WINDAM 1/1/2009
OPTION SIMPLE BARESOIL NOPS INTERNAL
IEMODEL 2 120 {Kd} {Us} 0.2
HYD 0.0167 0 1C 10
0.02
..
ENDtable
CRESTPRFL
0 {4*(2**Scale)}
900 {4*(2**Scale)}
```

```

ENDtable
STRUTE { 3*(2**Scale) } 0 ELEV
. .
CONDUIT 0.2 0.2 {(2**Scale)} 450

```

This processing can take place in parallel to reduce the overall computation time. Once the iterative analysis is complete, the output can be viewed in text or graphical format. In this example, tabular text data is stored in the file *dakota\_multidim.dat*. The peak outflow is shown below in figure 2.2. A single template file can be used to represent a wide range of different input parameters representing both material properties and geometric properties. Graphs are generated using Jzy3d. Each surface below represents a different Scale from 0 (on the bottom) to 5 (on the top).

The framework user interface shown in figure 2.2, allows users to specify a range of input parameters. The framework also allows Dakota to drive the computational model and change the model inputs and perform different analyses.

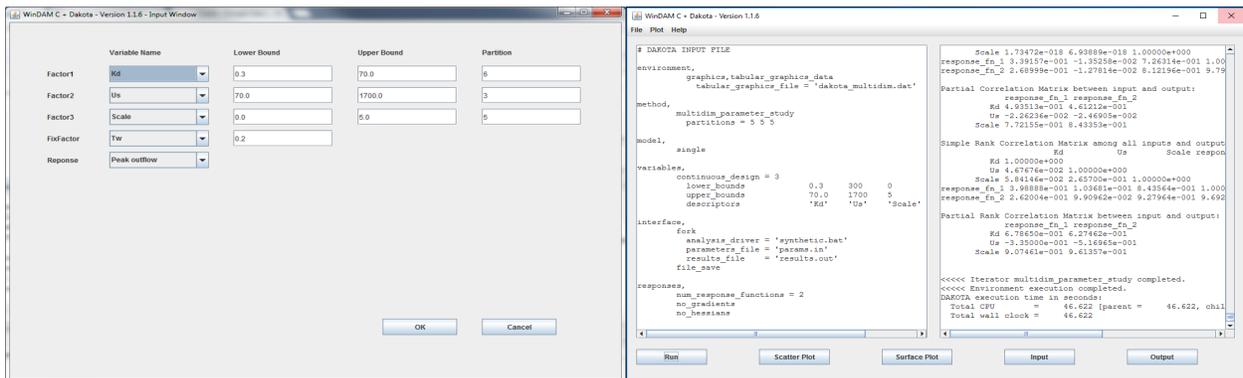


Figure 2.2: WinDAM/BREACH+Dakota GUI

The new interface shown in 2.2 allows users to specify a range of input parameters to allow Dakota to vary the model inputs and perform different types of analyses including uncertainty quantification, sensitivity analysis, and parameter studies. For example, users can analyze a range of varying input parameters. From the user interface, users just open an existing WinDAM C project, select the type of study, for example, a sim-

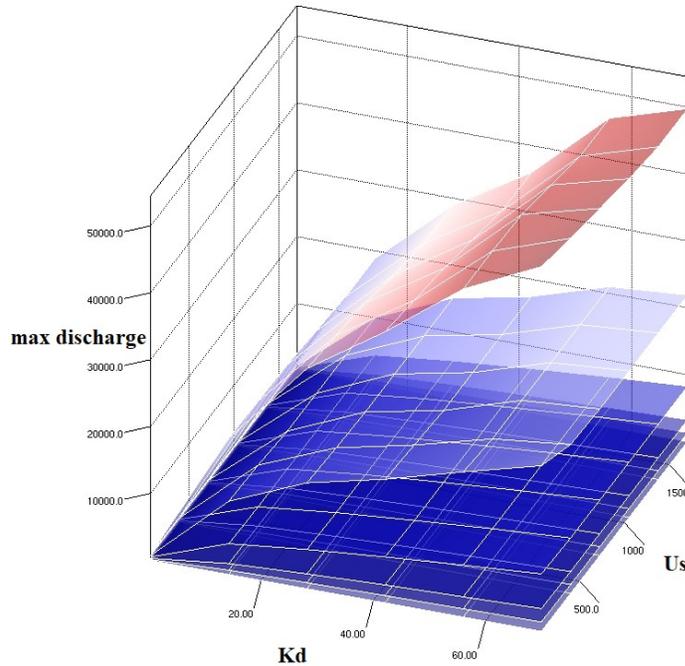


Figure 2.3: Surface Plot.

ple multidimensional parameter study. Then, the user is prompted to specify a range of parameters denoting different materials as shown in 2.2. The user can also specify the number of partitions in each dimension, or edit the generated Dakota input file directly.

For the given example, the resulting output shown in figure 2.3 and figure 2.4 confirms a strong positive correlation between erodibility  $Kd$  and peak discharge and a small negative correlation between Undrained Shear Strength  $Us$  and maximum discharge<sup>34</sup>.

The parameter study can be extended to perform uncertainty quantification by making a few small changes to the input. The resulting output confirms a strong positive correlation between  $Kd$  and peak discharge and a small negative correlation between  $Us$  and maximum discharge. The same framework can also be used to evaluate input files from an existing BREACH project. The corresponding BREACH variables, such as initial pool elevation  $HI$ , height of dam  $HU$  and Scale can be automatically parameterized, and response variable is set to `MaxOutflow`.

This processing can take place in parallel to reduce the overall computation time. Once the iterative analysis has been completed on all 75 input files. The resulting output

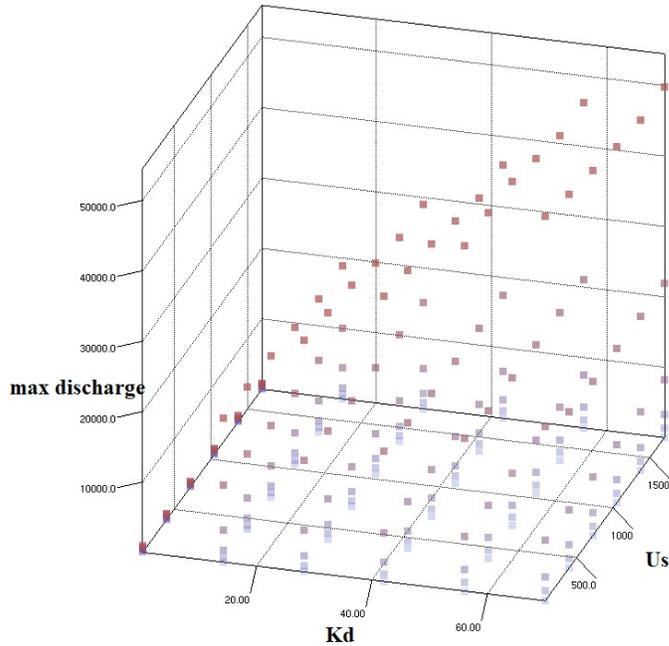


Figure 2.4: Scatter Plot.

confirms a strong positive correlation between HU and maximum discharge and a strong positive correlation between HI and maximum discharge.

Subsequent modules in WinDAM are being developed in stages to evaluate the performance of earth dams. Existing modules with well-defined interfaces enable efficient integration of existing legacy software with new model innovations. The system provides tools that can be used to better understand the structure, function, and dynamics of water control structures.

## 2.5 Conclusion

WinDAM is being developed in stages to evaluate the performance of earth dams. Existing modules with well-defined interfaces enable efficient integration of existing legacy software with new innovations. The system provides tools that can be used to better understand the structure, function, and dynamics of water control structures. This section describes how WinDAM and BREACH models can be analyzed efficiently using a novel

new Dakota interface.

Computational Fluid Dynamics(CFD) flow models and other computer graphics models are introduced in Chapter 3, to simulate the erosion process that results from the given flows. Efficient, high-level analysis can still be performed by combining these new models with Dakota. All of these simulation models are tested and the results are compared with the existing physics-based model and direct user input.

## Chapter 3

# Computer Vision in Dam Safety - Computer Simulation Models

For computer simulation modeling, interest in the potential flooding due to overtopping or internal erosion has existed for years. There are many simulation processes that can mimic and reproduce the visualization results. Fluid simulations are divided into the Eulerian and Lagrangian methods. The most popular fluid simulation software that uses the Eulerian method is called Computational Fluid Dynamics (CFD). Computational fluid dynamics uses numerical analysis and data structures to analyze and solve problems. CFD modelling approach for dam-break simulation study has been extensively investigated<sup>1 2 3</sup>. CFD uses three discretization methods: finite differences, finite volumes, and finite element methods. These three discretization methods use distinct mathematical models and governing equations for computation. In our previous study<sup>4;5</sup>, we explored the Finite Volume Method (FVM) for dam breach simulation using the open-source toolbox OpenFOAM. Cao<sup>5</sup> simulated the flooding due to overtopping without considering the deformation of the solid body dam structure.

Unlike Eulerian methods, there is no numerical dispersion for the Lagrangian method. Therefore, Lagrangian techniques are useful for simulations when large topological changes occur in the fluid interface. Smoothed Particle Hydrodynamics (SPH) is a well-known

method in the computer simulation field. Instead of considering the dam embankment structure as a solid body material in<sup>4,5</sup>, the granular material, such as soil, can be represented as either continuum or a set of individual particles using SPH. SPH modeling approach is used to simulate large surface flow<sup>6,7</sup> and dam break flows<sup>8,9,10</sup>. The Discrete Element Method (DEM) is also a very popular particle-based system to handle fluid simulation. Often, CFD is coupled with DEM in many studies and engineering fields including rockslide<sup>11,12</sup>, granular flow in water reservoir<sup>13,14</sup>, and fluid-particle interaction in dam break<sup>15,16</sup>. Other than CFD-DEM, the coupled model SPH-DEM is encountered in many multi-species simulation studies as well. Rungjiratananon et al. simulate sand-water interaction in real-time using a hybrid SPH-DEM model<sup>17</sup>. Wu<sup>18</sup> and Canelas<sup>19</sup> used the SPH-DEM method to simulate multi-phase free surface flow. Lenaerts<sup>20</sup> also proposed a dynamics framework for simulation of both fluid and porous granular material using SPH-DEM.

Another graphic approach Material Point Method(MPM) was introduced to handle topological change by multi-species interaction like water and porous soil<sup>21</sup>. Bandara et al.<sup>22,23</sup> introduced the soil deformation and pore fluid flow using MPM and also the landslides in unsaturated soil slopes. Tampubolon et al.<sup>24</sup> simulate the interaction of sand and water mixtures using MPM and get encouraging results. For the porous material property, Klár<sup>25</sup> used the improved Drucker-Prager plastic flow model with volume correction. For the MPM implementation, Arduino<sup>26</sup> and Jassim<sup>27</sup> examined that momentum exchange using the two-grid MPM for the multi-species interaction.

### **3.1 Computational Fluid Dynamic Method (CFD)**

Computational fluid dynamics (CFD) is a branch of fluid mechanics that uses numerical analysis and data structures to analyze and solve problems that involve fluid flows. OpenFOAM is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of computational fluid dynamics. Our work<sup>48</sup> uses OpenFOAM to design an architecture for analysis of water control structures. An

extensible computing framework is developed to compare with and on the existing WinDAM erosion models' use of OpenFOAM.

OpenFOAM<sup>55</sup> is a C++ toolbox for the development of customized numerical solvers, and pre-/post-processing utilities for the solution of computational fluid dynamics (CFD). Beocat<sup>56</sup> is a Kansas State University High-Performance Computing (HPC) cluster. Preliminary analysis can be conducted with a limited number of grid elements, and a more detailed analysis can be conducted on a high-performance computing cluster. The computing power for a single machine is limited, high-performance computing (HPC) plays a significant role in scientific computing for its capability of parallel computing for large-scale simulation jobs.

For this study, the new framework automates the process and can be used to evaluate the performance of OpenFOAM on a local laptop or on a cluster, such as Beocat. Computer simulation reproduces the behavior of a system using the mathematical model so that computer simulation becomes a useful tool to examine and compare with the experimental results. This extensible computing framework is used to compare and improve existing WinDAM<sup>33</sup> erosion models' use of OpenFOAM. The new framework allows computational fluid dynamics models for OpenFOAM to model the hydraulic flow over dams and through auxiliary spillways. OpenFOAM has extensive features and packages for solving complex fluid flows.

The first step is to construct the geometry of the dam and the water flow environment. The stereolithography (STL) file format is used to specify the dam cross-section. SALOME is an open-source software that provides a generic platform for pre-processing and post-processing numerical simulations. To define the geometry of the dam, the framework uses the input file to WinDAM to automatically generate an STL file. This STL file can be edited in SALOME. SALOME provides a variety of handy features, it can be used to generate a mesh from the STL file and view the geometric results using ParaView from within SALOME's generic platform.

For a simple trapezoidal dam cross-section, each dam STL file contains 18 triangular facets. Each triangular facet is defined by 3 vertexes and 1 direction. For this experimen-

tal design, a real empirical dam model is used to generate the physical parameters in a 2D geometry. Therefore, the length and mesh element in the Z direction has been set to one unit long. Though the model is constructed for 2D geometry, the default layout dimensions are 3D in OpenFOAM, so the non-zero (any length) is required for the Z dimension. Water velocity in X direction is recorded, and the Y-axis is the gravity direction.

For the construction of our water flow environment, SALOME has been widely used to generate the mesh and separate the water flow section and the airflow section in the geometry. The inlet height of water flow for our model is modified to match a real dam model. The background mesh is generated using the OpenFOAM ideasUnvToFoam solver. The ideasUnvToFoam solver is a format mesh conversion function that transfers the output file I-Deas unv format from SALOME to blockMesh format for OpenFOAM. Then, snappyHexMesh is the solver used to refine the mesh and generate more mesh elements on the border between the water and the solid dam. After running extrudeMesh by extrudeMeshDict, a 2D patch will remove the dam cross section and generate a dam shape in the water flow simulation model. In the future, we plan to apply 3D geometry in our simulation model as well.

time ( hr )	discharge ( cfs )	velocity (m/ s )
0	1.35	0.00095
0.063	4.54	0.00321
0.126	11.32	0.00801
0.189	23.72	0.01679
.	.	.
8.757	0.5	0.00035

Table 3.1: Input hydrograph data

After the geometry model is fixed, the water flow model must be specified. The reservoir capacity is specified as an initial condition in setFieldsDict. For the water flow, there are two possible types of flow in the pipe: laminar flow or turbulent flow. For this analysis, we specify turbulent flow, use the kEpsilon and RASmodel which stands for standard k-epsilon turbulence model for incompressible and compressible flows including rapid distortion theory (RDT) based compression term.

As shown in table 3.1, the left table is the WinDAM output for the hydrograph. The right table shows the ASCII text file for datatable.txt, the first column represents the time and the time interval; for this example input file, 02-SDHstability.WDC; the time interval is 0.063 hours. The second column has three values in one bracket, and it shows the different velocities in the X, Y, and Z directions for simulation purposes.

The conversion equation applied is shown below:

$$Velocity = \frac{Discharge}{(W \times D)} \quad (3.1)$$

Where  $W$  is Width and  $D$  is Depth.

In the analysis,  $W = 1\text{ m}$ ,  $D = 4\text{ m}$ . For units of measurement, all variables are converted from the US system (used by USDA) to the Imperial (MKS) system. In order to apply to the OpenFOAM model properly, a scale factor has been added. The initial velocity in the model is only in the X direction because the model is constructed using a 2D geometry and the water inlet is designed to flow in the X-direction. Thus, the initial velocity in the Y direction is 0, as shown in table 3.1.

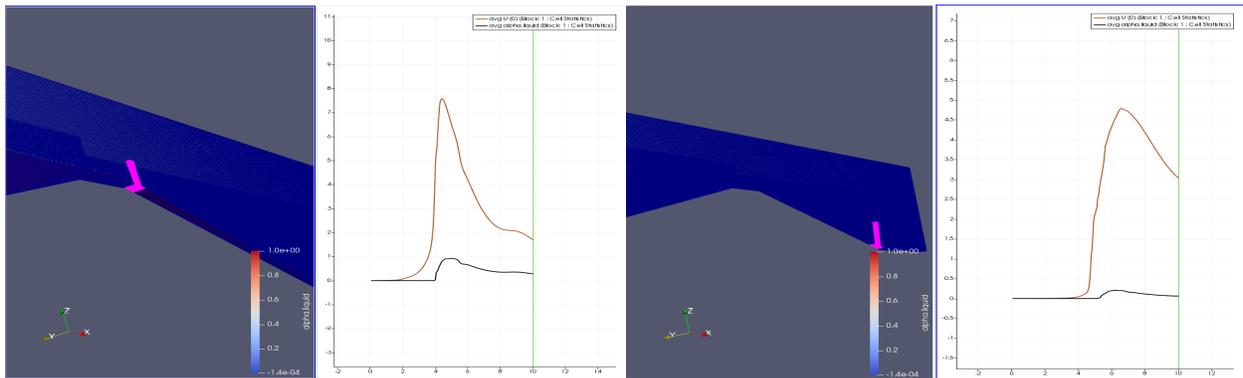


Figure 3.1: Dam Region Simulation Result, Top(Left), Bottom(Right).

To achieve the hydrograph data in a different aspect, the sample data from two different regions in our simulation model has been selected. One hydrograph data came from the top of the dam, and another set of hydrograph data came from the bottom of the dam as shown in figure 3.1. In figure 3.1, there are two different lines in the graph, the red line represents the measure of area ( $m^2$ ), and the brown line represents velocity ( $m/s$ ).

In figure 3.2, the scatter plot displays the results for the WinDAM and OpenFOAM simulation. The OpenFOAM simulation is based on a 10-second time frame and 0.05-time interval. The blue line represents the inflow, the orange line shows the total outflow for WinDAM and the gray line shows the total discharge for the top of the dam which is generated by the OpenFOAM simulation. For WinDAM, the total outflow is calculated near the top of the dam. Therefore, the result for the top of the dam generated in OpenFOAM is selected to compare with WinDAM. The hydrograph for these two different results should have the same amount of area in the graph because the amount of water that goes through the dam is expected to be similar. The hydrograph for WinDAM has a higher peak than OpenFOAM. For the OpenFOAM hydrograph, it is shifted slightly to the right compared to the WinDAM hydrograph, as expected. The turbulence that occurs at the end of the hydrograph shows that there is friction along the edge of the dam. Therefore, the total water volume is close to these two hydrographs. The reason why the OpenFOAM hydrograph has a two-second delay is that for the OpenFOAM simulation, the position of inlet water is built far away from the dam. It takes more time to reach the dam compared to WinDAM. WinDAM assumes that the water inlet is fairly close to the dam. For the overall comparison, both outflow hydrographs have similar patterns and shapes.

After completing a preliminary experiment, the model is executed on Beocat. Beocat is a High-Performance Computing (HPC) cluster at Kansas State University, it is running under CentOS and coordinated by a job scheduler system called Slurm.

The sample code:

```
#!/bin/bash
#SBATCH --job-name=opfile1_4C32G_0.0001_f2
#SBATCH --mem-per-cpu=32G
# Memory per core, use --mem= for memory per node
#SBATCH --time=02:00:00
# Use the form DD-HH:MM:SS
#SBATCH --nodes=1
#SBATCH --ntasks-per-node=4
```

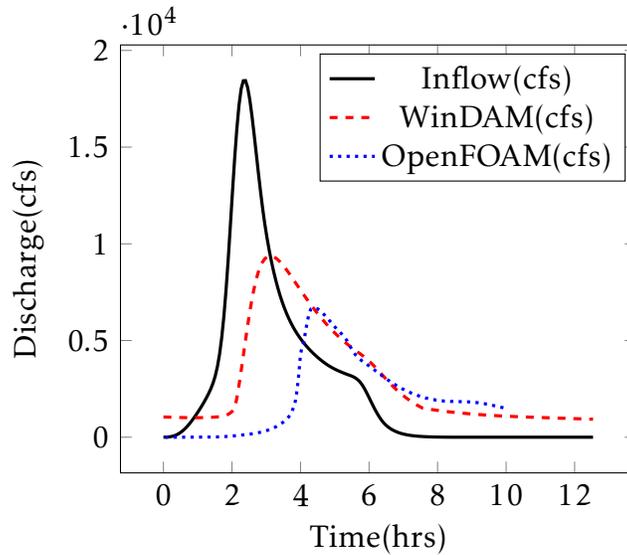


Figure 3.2: Hydrograph Comparison Plot.

```
#SBATCH --mail-user=caocd@ksu.edu
#SBATCH --mail-type=ALL
# same as =BEGIN, FAIL, END
singularity exec /opt/beocat/containers/
openfoam-v1712.img /bin/bash <<EOF
. /opt/OpenFOAM/setImage_v1712.sh
.....
OpenFOAM code
.....
EOF
```

Listing 3.1: Slurm job bash code.

As shown above, in Listing 3.1, starting with `/bin/bash` and OpenFOAM code is added inside the `SBATCH` code segment. One single job can request 32 GB of memory per CPU, request 4 core at a time and total running time of 2 hours. Since the OpenFOAM module is installed inside a container on Beocat, the number of nodes is limited to only one. For this running time simulation job, we select two flow functions:

$$f_1 = 0.5 + 0.5\sin(0.2\pi t), \quad (3.2)$$

$$f_2 = 4.$$

$f$  is the velocity of water flow in  $X$  direction.  $t$  is the time in seconds.

We run the simulation based on different cores and vary deltaT, where deltaT is the time step of the simulation. In OpenFOAM's controlDict file, we turn off the time stamp auto adjustment. The summary table as shown in table 3.2. The simulation is performed under the circumstance that the end time is 4 seconds, the write interval is 0.05 seconds, and the unit for runtime is in hours. For example, the first case uses the  $f_1$  flow function,  $\text{deltaT} = 0.0001$ , 1 Core represents that one 32GB memory CPU requested, therefore the simulation time case 1 is 4 hours 54 minutes and 59 seconds. By reviewing the result, roughly 1 core performs 2 times faster than 1 core, 4 core performs 2 times faster than 2 core, 8 core performs 2 times faster than 4 core. So, we achieve nearly linear speedup. As expected, the run time gets smaller when you increase the timestamp deltaT.

N	F	deltaT	1 Core	2 Core	4 Core	8 Core
1	$f_1$	0.0001	4:54:59	2:33:16	1:29:36	0:50:11
2	$f_1$	0.0002	2:38:29	1:17:46	0:52:56	0:28:19
3	$f_1$	0.0005	1:18:25	0:26:47	0:17:15	0:12:44
4	$f_1$	0.0010	0:46:08	0:21:36	0:09:26	0:07:25
5	$f_2$	0.0001	4:36:29	2:32:46	1:28:09	0:48:56
6	$f_2$	0.0002	2:43:03	1:12:17	0:52:29	0:27:23
7	$f_2$	0.0005	1:07:25	0:30:42	0:16:57	0:12:29
8	$f_2$	0.0010	0:34:40	0:22:44	0:09:16	0:07:09

Table 3.2: Run Time table for OpenFOAM simulation.

For different deltaT, figure 3.3 shows the speedup for the different numbers of cores; 8 cores perform approximately 5 times faster than 1 core, 4 cores perform almost 3 times faster than 1 core and 2 cores are about 2 times faster than 1 core. This figure also shows that with larger deltaT's, less speedup can be achieved.

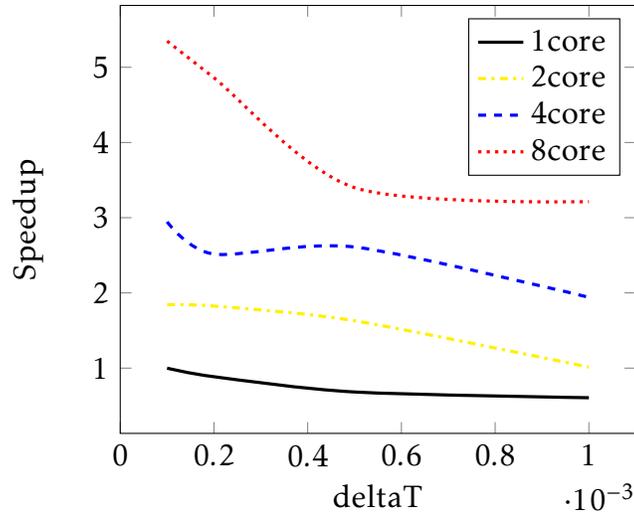


Figure 3.3: Performance comparison plot for different number of cores.

## 3.2 Discrete Element Method (DEM)

A discrete element method (DEM), also called a distinct element method, is a numerical method for computing the motion and effect of a large number of small particles. DEM is a particle-based simulation method, which was originally used for rock mechanics problems. Now, DEM is becoming widely accepted as an effective method of addressing engineering problems in granular and discontinuous materials.

## 3.3 Smoothed Particle Hydrodynamic (SPH)

Smoothed-particle hydrodynamics (SPH) is a computational method used for simulating the mechanics of continuum media, such as solid mechanics and fluid flows. It was developed by Gingold and Monaghan and Lucy in 1977, initially for astrophysical problems. It is a mesh-free Lagrangian method, and it can easily be adjusted with respect to variables such as density.

### 3.4 Coupled Model CFD-DEM, DEM-SPH

We know that computational fluid dynamics (CFD), discrete element method (DEM) and Smoothed Particle Hydrodynamic (SPH) techniques can be used to study the mechanical and hydraulic behavior of particles and fluid flow. The coupled DEM-CFD method has been widely used to study complex continuum-discrete problems, with a variety of engineering and industry applications, such as the modeling of granular flows<sup>57</sup>, granular impacting on a water reservoir<sup>14</sup>. Also, the coupling method of SPH and DEM can deal with application interactions between free surface flow and strong fluid particles.

### 3.5 Material Point Method (MPM)

Material Point Method (MPM) was initially introduced by Sulsky<sup>58</sup> and used to tackle more complex problems. Material Point Method (MPM) is a method to simulate fluids and solid materials undergoing large deformation. MPM discretize the model since it provides a natural and efficient way of treating contact, topological change, and history-dependent behavior.

The Material Point Method computes forces using a fixed Eulerian grid, while it also stores information on Lagrangian particles. The fixed grid handles the topology changes like collisions between the material. MPM stores the information using two distinct representations, and it must be transferred between them. MPM also avoids re-meshing and storing connectivity between particles.

The large deformation landslide problem using MPM was widely investigated. Soga<sup>59</sup> introduce MPM to analyze large-deformation of landslide mass movements and post-failure behavior. Troncone<sup>60;61</sup> use MPM to simulate landslide triggered by an increase in the groundwater level or water pressure. Yerro<sup>62;63</sup> apply MPM to analyze landslides for brittle soils or unsaturated soils. Also, there are lots of potentiality of MPM for analyzing geotechnical problems including the landslide that occurred at Oso in USA<sup>64</sup>, the landslides of Senise and Maierato in Italy<sup>65;66</sup>, the Sainte-Monique landslide<sup>67</sup>. Dong

applied MPM to solve submarine landslide problems<sup>68;69</sup>.

To handle multi-species interaction such as water and porous soil<sup>21</sup>, Bandara et al.<sup>22;23</sup> introduced models for soil deformation and porous fluid flow using MPM. Tampubolon et al.<sup>24</sup> simulate the interaction of sand and water mixtures using MPM and get encouraging results. For the porous material property, Klár<sup>25</sup> used the improved Drucker-Prager plastic flow model with volume correction. For the MPM implementation, Arduino<sup>26</sup> and Jassim<sup>27</sup> examined that momentum exchange using the two-grid MPM for the multi-species interaction.

For dam safety simulation, the two most common approaches for the discretization of solids or liquids are Eulerian or Lagrangian. For Eulerian-based methods, quantities of interest are in fixed locations or fixed grids such as in CFD or CFD-DEM mesh modeling. For the Lagrangian-based methods, quantities of interest are attached to the materials including SPH, SPH-DEM, and MPM particle-based methods. Among these modeling methods, in order to simulate dam erosion with multi-species interactions between soil and water, the proposed new dam simulation models use the Material Point Method which combines aspects of both types of discretization.

In particular, we construct a new model for the multi-species material point method to allow interaction between soil and water. We allow wet soil transitions from cohesive grains to flowing sediment as water saturation increases. We create an overtopping dam erosion simulation model using multi-species particles and a two-grid Material Point Method. For validation, the MPM simulation model of overtopping dam erosion is compared with available experimental data and results of other physical-based models like WinDAM<sup>70</sup>.

## Mathematical background

Tampubolon<sup>24</sup>, Atktn<sup>71</sup> and Borja<sup>72</sup> all considered multi-species using mixture theory. Therefore, the soil and water have been model as a multi-species continuum using mixture theory.

## Conservation laws

The conservation of mass and conservation of momentum has been calculated individually, and each species obeys the following conservation laws with respect to its own motion.

### Conservation of mass

$$\frac{D\rho}{Dt} + \rho \nabla \cdot v = 0 \quad (3.3)$$

### Conservation of momentum

$$\rho \frac{Dv}{Dt} = \nabla \cdot \sigma + \rho g \quad (3.4)$$

Here, the superscript  $s$  is used to represent soil,  $w$  is used to represent water, total mass density is  $\rho = \rho^s + \rho^w$  and total momentum is the sum  $\rho v = \rho^s v^s + \rho^w v^w$ . The velocity  $v = \frac{1}{\rho}(\rho^s v^s + \rho^w v^w)$ . After summing the two species, the standard conservation of mass is obtained in (??) and also note that the conservation of linear momentum for the individual species implies conservation of linear momentum for the mixture (??).

## 3.6 Deformation Gradient

The deformation gradient represents how deformed material is local. It is used to measure how the material has locally rotated and deformed due to its motion<sup>24</sup>. Plasticity is represented by factoring the deformation gradient into elastic and plastic parts as  $F = F^E F^P$ <sup>25</sup>.

$$\frac{DF}{Dt} = (\nabla v)F \quad (3.5)$$

By factoring the deformation gradient in this way, we divide this deformation history into two pieces, plastic deformation  $F^P$  and elastic deformation  $F^E$ . figure 3.4 shows

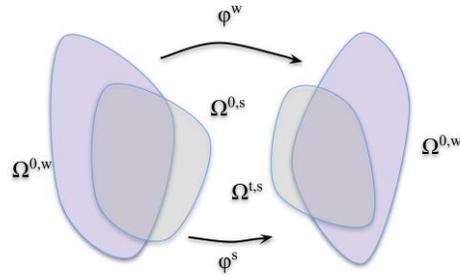


Figure 3.4: Multi-phase deformation combining water and soil.

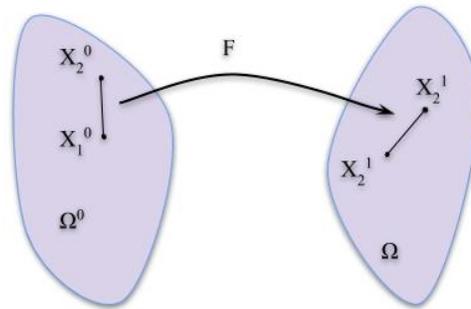


Figure 3.5: The deformation of a material.

single-phase deformation gradient  $F$  and material properties. figure 3.5 displays material property for multi-phase deformation between water and soil, the overlap region is captured through a momentum exchange.

### 3.7 Elastoplasticity

The elastoplastic theory defines when elastic and inelastic deformation occurs in the material. Elastic deformation is when the soil is forming piles, and inelastic deformation is when soil particle is freely rolling down a slope and won't try to return to its original shape. Compression and tension for elastic and inelastic deformation is shown in figure 3.6

Variable	Meaning
$I$	identity matrix
$a$	$a = s, w$ soil and water
$\Delta t$	time step size
$\frac{D}{Dt}$	material derivative
$\rho_\alpha$	density
$g$	gravitational constant
$c_E$	drag coefficient
$w_{ip}^{a,n}$	weight
$\nabla w_{ip}^{a,n}$	weight gradient
<b>Particle</b>	
$V_p^\alpha$	h initial particle volume
$m_p^\alpha$	particle mas
$x_p^{a,n}$	particle position
$v_p^{a,n}$	particle velocity
$F_p^{sE,n}$	soil elastic deformation gradient
$F_p^{sP,n}$	soil plastic deformation gradient
$J_p^{w,n}$	water determinant deformation gradient
$\phi_p^{s,n}$	water saturation
$c_{Cp}^{s,n}$	cohesion
$v_{cp}^{s,n+1}$	volume correction scalar
<b>Grid</b>	
$v_i^\alpha$	grid velocity
$m_i^\alpha$	grid node mass
$x_i^{a,n}$	grid node location
$\phi_i^{w,n}$	mixed water saturation on grid

Table 3.3: table of notation.

### 3.8 Constitutive Model: Soil Elastoplasticity

Tampubolon<sup>24</sup> modify the model in Klár<sup>25</sup> to include cohesive stresses for porous and water mixture. Therefore, the amount of cohesion varies with the saturation of water in the mixture. The soil partial stress  $\sigma^s$  is defined in terms of the hyperelastic potential energy density  $\psi^s$  as

$$\sigma^s = \frac{1}{\det(F^s)} \frac{\partial \psi^s}{\partial F} (F^{s,E}) F^{s,E^T} \quad (3.6)$$

The  $F^s$  term represents the deformation gradient of soil motion. The  $F^{s,E^T}$  and  $\frac{1}{\det(F^s)}$

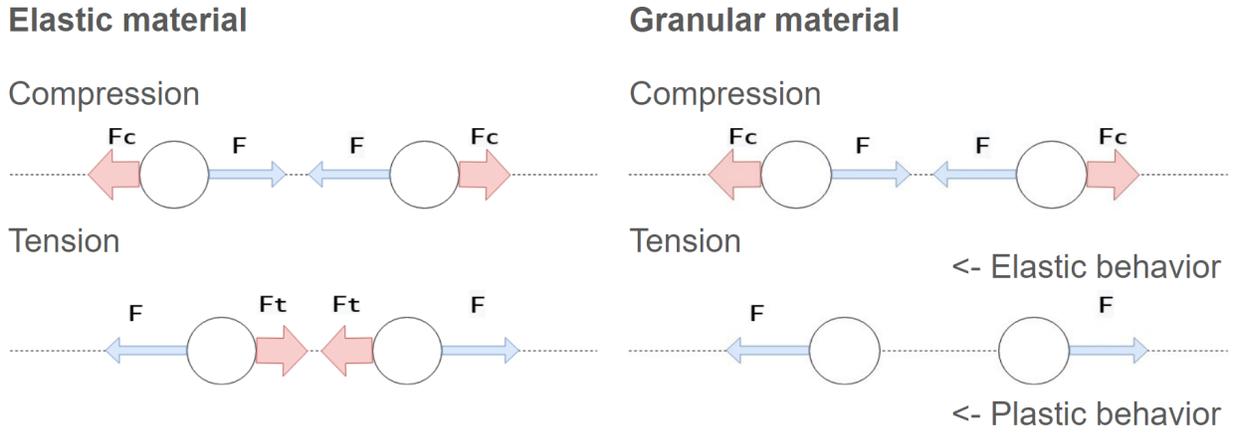


Figure 3.6: Compression and tension for elastic and granular material.

terms arise because of the potential energy density in terms of the deformation gradient. As in finite strain elastoplasticity<sup>73</sup>,  $F^S = F^{s,E} F^{s,P}$  defines plastic flow for porous soil.  $F^{s,E}$  represents the compression and shearing, while  $F^{s,P}$  represents the sliding and separation. We use the Drucker-Prager<sup>74</sup> plastic flow and yield condition to determine the elastic and plastic deformation gradient. Bonet and Wood<sup>73</sup> provides background for elastoplastic constitutive modeling. Dry porous material can be modeled effectively with the assumption made by the Drucker-Prager condition because the yield condition is defined from the constraint that shear stress should be no larger than the compressive normal stress in all directions<sup>24</sup>. For the elastic part of the constitutive behavior for the soil phase. The elastic potential energy density is defined in terms of the logarithmic strain  $\epsilon$  as

$$\psi^s(F^S) = \tilde{\psi}^s(\epsilon) = \mu \text{tr}(\epsilon^2) + \frac{\lambda}{2} \text{tr}(\epsilon) \quad (3.7)$$

where  $F^S = U \Sigma V$  is the singular decomposition of  $F^S$  and  $\epsilon = \log(\Sigma)$ .

## Water

We model the water incompressible stress<sup>75</sup> with partial stress

$$\sigma^w = -p^w I, p^w = k \left( \frac{1}{J^w \gamma} - 1 \right) \quad (3.8)$$

$p^w$  is the water pressure.  $J^w$  is the ratio of current to initial local volume of material in the water phase.

$$\frac{D^w}{Dt} J^w = \nabla v^w J^w \quad (3.9)$$

Where  $k$  is the bulk modulus of the water and  $\gamma$  is a term that more stiffly penalizes large deviations from incompressibility.

### 3.9 Momentum Exchange

The momentum exchange term  $P^s$ ,  $P^w$  for water and porous soil interactions can be considered as a combination of dissipative and reversible interactions<sup>72</sup>. Bandara<sup>23</sup> introduced the formulation, and they assume

$$P^s = c_E (v^w - v^s) + p^w \nabla \phi^w, P^w = -P^s \quad (3.10)$$

where  $c_E = \frac{n^2 \rho^w g}{\hat{k}}$  is drag coefficient and  $n$  is the soil porosity,  $\hat{k}$  is the soil permeability and  $g$  is the gravitational acceleration.  $\phi^w = \frac{\rho^w}{\rho^w + \rho^s}$  is the water volume fraction and  $p^w$  is the water pressure.

### 3.10 Cohesion and Saturation

The method used to estimate the saturation is the percentage of water in mixture to the total density, which is  $\phi^w = \frac{\rho^w}{\rho^w + \rho^s}$ . The soil cohesion varies as a function of water saturation. The cohesion of soil is zero when it is completely dry ( $\phi^w = 0$ ). Robert and Soga<sup>76</sup>

found that the increase to a maximal value  $c_C$  when the  $\phi^w$  is setting to 0.4. In our experiment, we create a model in which the water interaction with wet soil and also the wet soil can keep the shape. We set the soil cohesion to the maximal at the beginning, and the cohesion decreases linearly with increasing saturation beyond this point which means if we have full saturation for the mixture, the cohesion equals zero and ratio  $\phi^w = 1$ .

### 3.11 Discretization

Material Point Method (MPM) computes the forces using a fixed Eulerian grid but also keeps the information on Lagrangian particles. Information can be transferred through the grid and particles since MPM use two distinct representations. There are other approaches in the engineering literature<sup>77 27 26</sup> using MPM to solve the multi-species problem. We follow Tampubolon<sup>24</sup> to discretize the continuum equation using two sets of the grid. One is associated with soil and the other is associated with water. We use superscript  $\alpha = s, w$  indicates the corresponding species.

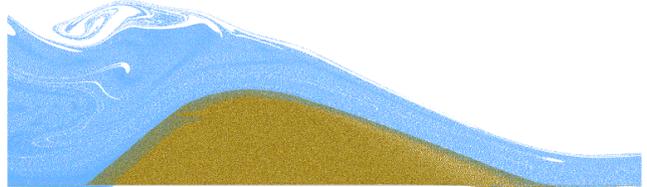
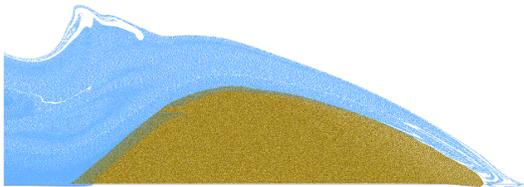
#### Transfer to grids

As the process demonstrated in figure 3.8, the primary representation of the state is stored on particles in MPM in figure 3.8(1). MPM transfers the mass and momentum of each species to the grid. We use the mass and momentum of each species to compute velocity on its corresponding grid in figure 3.8(2). We apply related velocity spatial derivatives: mass, position, velocity, and affine momentum.

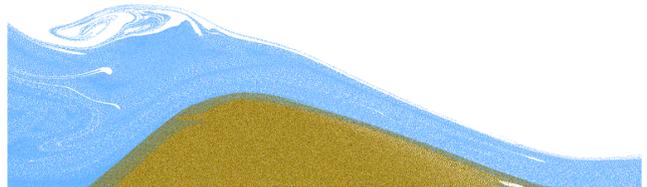
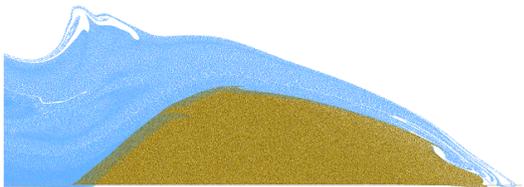
$$C_p^n = \sum w_{ip}^n (x_i^n - x_p^n)(x_i^n - x_p^n)^T = \begin{cases} \frac{h^2}{3} I & \text{cubic} \\ \frac{h^2}{4} I & \text{quadratic} \end{cases} \quad (3.11)$$

Where  $C_p^n$  is an inertia tensor matrix and  $h$  is the grid spacing. We use the APIC transfer as discussed in<sup>78</sup>.

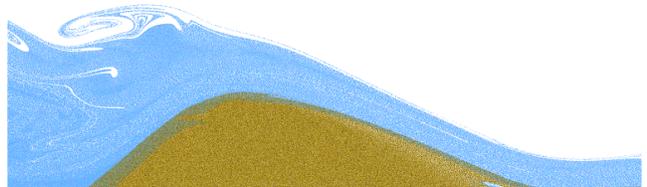
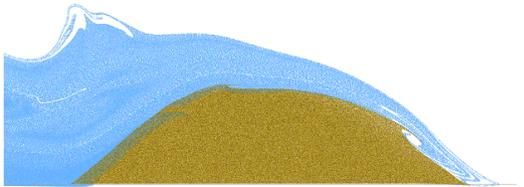
(1) Coh = 0.002



(2) Coh = 0.004



(3) Coh = 0.006



(4) Coh = 0.008

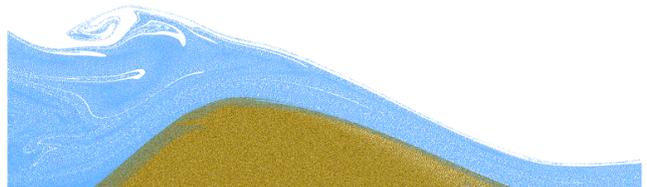
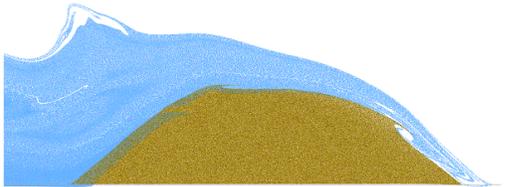


Figure 3.7: Overtopping dam breach simulation with different cohesion at timestamp  $t_0$  and  $t_0 + \Delta t$ .

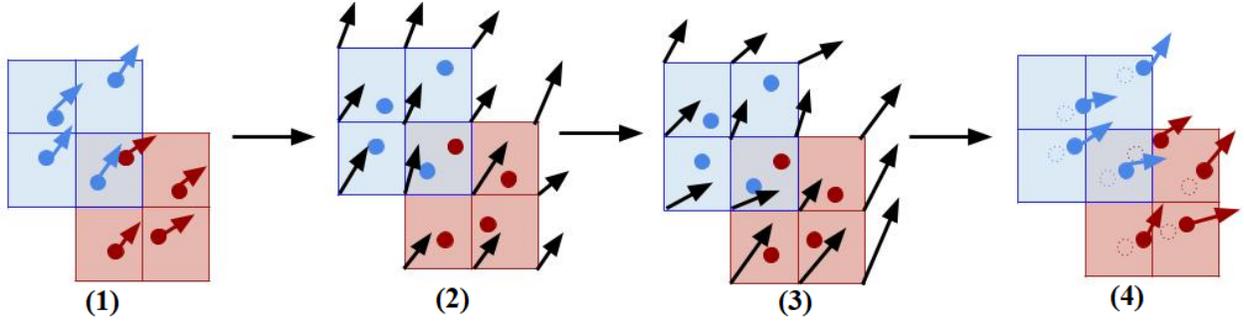


Figure 3.8: MPM algorithm with two grid multi-species.

- 1.(a)-(b) Transfer to grids: The mass and momentum of each species are transferred to its corresponding grid.
- 2.(b)-(c) Update grids momentum: The coupled water and soil grid velocities is updated.
- 3.(c)-(d) Update particles: All particle states, including the momentum, velocity, and cohesion based on saturation are updated.

The information needs to be transferred between grid and particle representations. We do this by assigning each particle and grid node to a weight that determines how firmly the particle and node interact. If the particle and grid node are close together, the weight should be large. Otherwise, the weight should be small. As demonstrated in figure 3.8, information like velocity and momentum exchange can be transferred between grid nodes and particles according to the weights. We initialize mass for grid nodes and particles, calculate weight, weight gradient, and kernel using cubic b-splines kernel for the particle. The first step is the transfer of state particles to the fixed Cartesian grid, and then distributing the mass of each particle to its surrounding grid nodes, summing up mass for grid nodes and multiplying the surrounding weight and particle mass, as addressed in the equation

$$m_i^{a,n} = \sum_p w_{i,p}^{a,n} m_p^a \quad (3.12)$$

We compute weights based on a kernel as  $w_{i,p}^{a,n} = N(x_p^{a,n} - x_i^{a,n})$ , where  $x_p^{a,n}$  and  $x_i^{a,n}$  are the locations of the particle and grid node locations based on each species  $a = s, w$ .

$$\hat{N}(x) = \begin{cases} \frac{1}{2}|x|^3 - |x| + \frac{2}{3} & 0 \leq |x| < 1 \\ \frac{1}{6}(2 - |x|)^3 & 1 \leq |x| < 2 \\ 0 & 2 \leq |x| < \infty \end{cases} \quad (3.13)$$

$$N(u) = \hat{N}\left(\frac{u_x}{h}\right)\hat{N}\left(\frac{u_y}{h}\right)\hat{N}\left(\frac{u_z}{h}\right) \quad (3.14)$$

Where  $h$  is the grid spacing. We use the cubic spline for all of our examples.

$$v_i^{a,n} = \frac{1}{m_i^{a,n}} \sum_p w_{ip}^{a,n} m_p^a (v_p^{a,n} + C_p^{a,n} (x_i^{a,n} - x_p^{a,n})) \quad (3.15)$$

For this step, we initialize the velocities for the particle and nodes, grid nodes velocity is calculated using velocity transfer using  $C_p^{a,n} = B_p^{a,n} (D_p^{a,n})^{-1}$  is where  $D_p^{a,n}$  is cubic b-splines kernel and  $B_p^{a,n}$  is affine momentum matrix followed the APIC transfer<sup>78</sup>.

## Update grids momentum

As shown in figure 3.8, MPM computes the forces using elastic and plastic deformation gradient and solve for coupled water and soil grid velocities. We determine the explicit grid node force using particle volume times energy differential, particle deformation gradient transpose, and weight gradient.

The forces in the soil and water phases are computed by

$$f_i^s(\hat{x}^s) = -\frac{\partial \psi^s}{\partial \hat{x}_i^s} = -\sum_p V_p^0 \left( \frac{\partial \psi^s}{\partial F_p^s} (F_p^{s,E}(\hat{x}^s)) \right) (F_p^{s,E,n})^T \nabla w_{ip}^{s,n} \quad (3.16)$$

$$f_i^w(\hat{x}^w) = -\frac{\partial \psi^w}{\partial \hat{x}_i^w} = -\sum_p V_p^0 \left( \frac{\partial \psi^w}{\partial J^w} (J^w(\hat{x}^w)) \right) J^{w,n} \nabla w_{ip}^{w,n} \quad (3.17)$$

Stomakhin<sup>79</sup> mentioned that  $\hat{x}_i^a$  can be considered as the position of the grid node  $i$  corresponding to species  $a$  that has been deformed from its original position  $x_i^a$  by the amount

of  $\Delta t v_i^{a,n+1}$ . The discrete momentum balance as

$$m_i^{s,n}(v_i^{s,n+1} - v_i^{s,n}) = \Delta t (f_i^s(\hat{x}^s) + m_i^{s,n}g + d_i^s(\hat{x})) \quad (3.18)$$

$$m_i^{w,n}(v_i^{w,n+1} - v_i^{w,n}) = \Delta t (f_i^w(\hat{x}^w) + m_i^{w,n}g + d_i^w(\hat{x})) \quad (3.19)$$

where the discrete interaction term given by

$$d_{ij}^s(\hat{x}) = -c_E m_i^s m_j^w (v_i^{s,n+1} - v_j^{w,n+1}) \quad (3.20)$$

$$d_{ji}^w(\hat{x}) = c_E m_i^s m_j^w (v_i^{s,n+1} - v_j^{w,n+1}) \quad (3.21)$$

for the drag coefficient  $c_E$ . Setting

$$M = \begin{pmatrix} M^{s,n} & \\ & M^{w,n} \end{pmatrix}, v = \begin{pmatrix} v^s \\ v^w \end{pmatrix} \quad (3.22)$$

$$f(\hat{x}(v^{n+1})) = \begin{pmatrix} f^s(\hat{x}^s) \\ f^w(\hat{x}^w) \end{pmatrix} \quad (3.23)$$

Then, we calculate the grid nodes velocity for both soil and water materials using drag coefficient  $c_E$  and updated grid nodes explicit forces are shown in figure 3.8(2) and figure 3.8(3).

$$(M + \Delta t D)v^{n+1} = Mv^n + \Delta t(Mg + f(\hat{x}(v^{n+1}))) \quad (3.24)$$

Where  $M$  is the mass matrix,  $v$  is the velocity matrix for both soil and water particles.  $g$  is gravity.

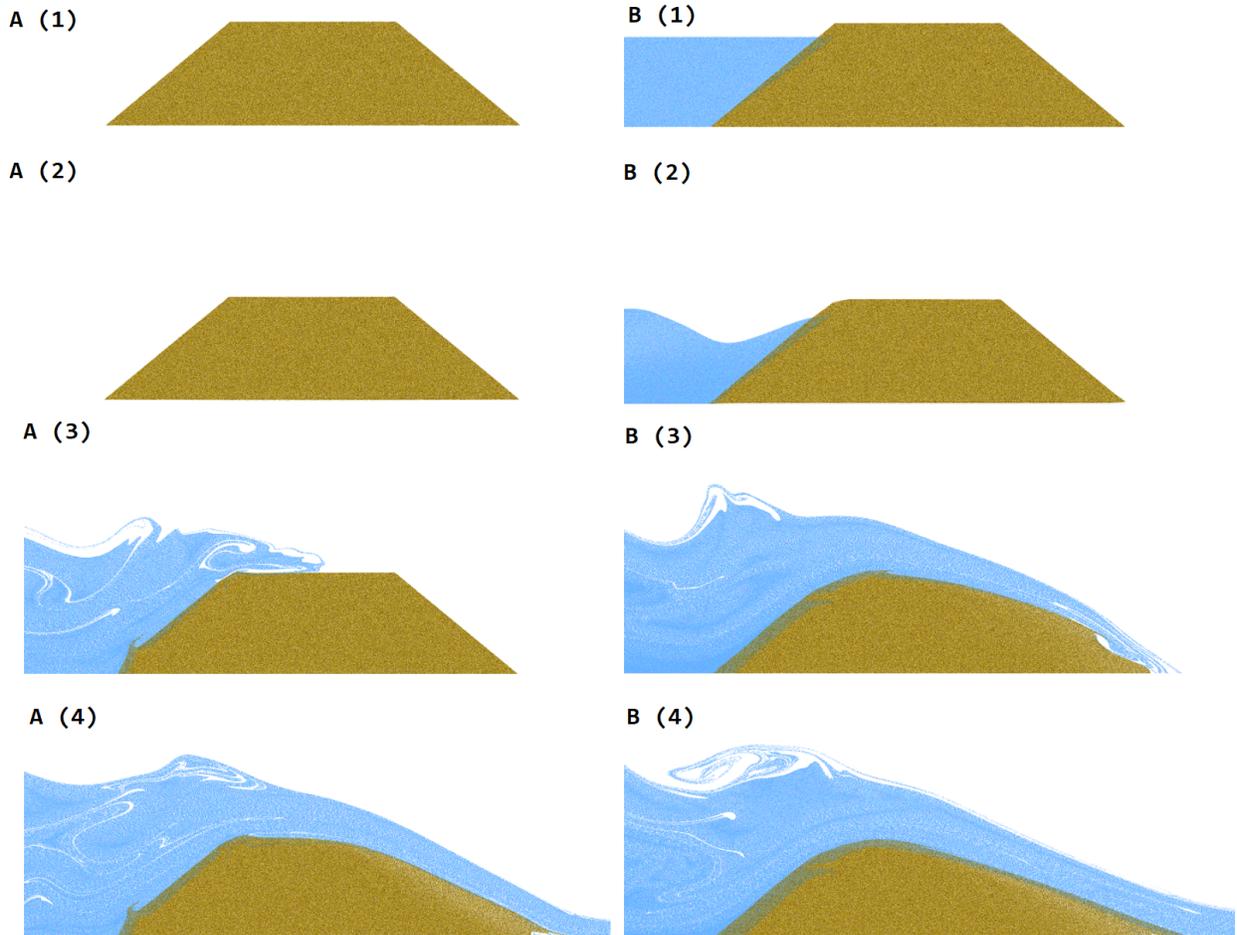


Figure 3.9: Overtopping dam breach simulation with empty(left) and full(right) reservoir.

## Update particles

As illustrated in figure 3.8(3) and figure 3.8(4), MPM updates all particle states, including the cohesion based on saturation as well as plasticity return mappings. The framework also keeps track of the water determinant  $J^w$ .

$$J_p^{w,n+1} = (I + \Delta t \cdot \text{tr}(\nabla v_p^{w,n+1}))J^{w,n} \quad (3.25)$$

Since we don't consider the effects of plasticity during simulation, the framework implicitly solves for momentum,  $\hat{F}^{s,E,n+1}$ , with the grid during the grid momentum update.

$$\hat{F}^{s,E,n+1} = (I + \Delta t \cdot \text{tr}(\nabla v_p^{s,n+1}))\hat{F}^{s,E,n} \quad (3.26)$$

For each grid, We set the indicator function of the overlap region between the soil and water constituents. Then, we compute the soil particle cohesion using the sum of surrounding grid nodes cohesion

$$\phi_p^{s,n+1} = \sum_i w_{s,ip}^n \phi^{w,n+1} \quad (3.27)$$

We use the cohesion linear function of water saturation.

$$c_{Cp}^{s,n+1} = c_{Cp}^{s,0}(1 - \phi_p^{w,n+1}) \quad (3.28)$$

Next, the framework applies the Drucker-Prager projection and introduces volume correction treatment. Plasticity is defined in terms of the singular value decomposition of the deformation gradient.  $F_p^{s,E,n+1} = U_p \Sigma_p V_p^T$ , and  $\epsilon_p = \ln \Sigma_p$ . For the artifact, we add an extra scalar attribute  $v_{cp}^{s,n}$  and at each time step<sup>24</sup>.

For the next step, the framework updates the grid node velocity for collisions and friction based on the explicit case. Then, velocities are updated for particles using the

surrounding grid node velocities. The velocity of particles updated according to

$$v_p^{a,n+1} = \sum_i w_{ip}^{a,n} v_i^{a,n+1} \quad (3.29)$$

The position of particles updated according to

$$x_p^{a,n+1} = x_p^{a,n} + \Delta t v_p^{a,n+1} \quad (3.30)$$

Finally, the algorithm updates the deformation gradient and position for both soil and water particles.

## Implementation and simulation results

Dam Failure Example	1	2	3	4	5	HansonE1S1	05-HR	05-HR
Cohesion $c_{Cp}$	0.01	0.008	0.006	0.004	0.002	0.008	0.008	0.008
Gravity ( $m/s^2$ )	3	3	3	3	3	3	9.8	9.8
Initial Reservoir Level (m)	No	3.25	3.25	3.25	3.25	4	2.4	2.4
Number of Water Particle	0	197542	197542	197542	197542	195811	95131	95131
Dam Height (m)	3.75	3.75	3.75	3.75	3.75	4.6	3.3	3.3
Crest Width (m)	6	6	6	6	6	3.68	1	1
Upstream Slope V/H	0.8333	0.8333	0.8333	0.8333	0.8333	0.3333	0.3882	0.3882
Downstream Slope V/H	0.8333	0.8333	0.8333	0.8333	0.8333	0.3333	0.3143	0.3143
Number of Soil Particle	277887	277887	277887	277887	277887	403225	173982	173982
Water Inlet Height (m)	7.25	7.25	7.25	7.25	7.25	7.25	2.5	2.5
Inlect Velocity (m/s)	2	2	2	2	2	2	5	2
Resolution	7000*2800	7000*2800	7000*2800	7000*2800	7000*2800	7000*2800	7000*2800	7000*2800
Number of Iteration	2399	2185	1503	2399	1979	2399	2087	2371
Simulation Time (hrs)	44	50	30	55	42	71	64	43

Table 3.4: Condition for the cohesive dam’s overtopping failure cases and simulation cases.

The project is implemented in C++ and visual studio 2017, and the implementation idea is inspired by Xia’s project<sup>80</sup>. Simulation times shown in table 3.4 have been measured on a PC with Intel(R) Core i5-3570K CPU, 16 GB RAM, and an NVIDIA GeForce GTX 470 GPU.

Parametric studies has been conducted and shown in figure 3.7 , 3.9 , 3.10 and table 3.4. For figure 3.7, it shows a different selection of the cohesion parameter  $c_{Cp}$  from the range 0.002 to 0.008, which will make an influence on the overtopping dam breach erosion simulation. The result shows that with the higher cohesion demonstrated in fig-

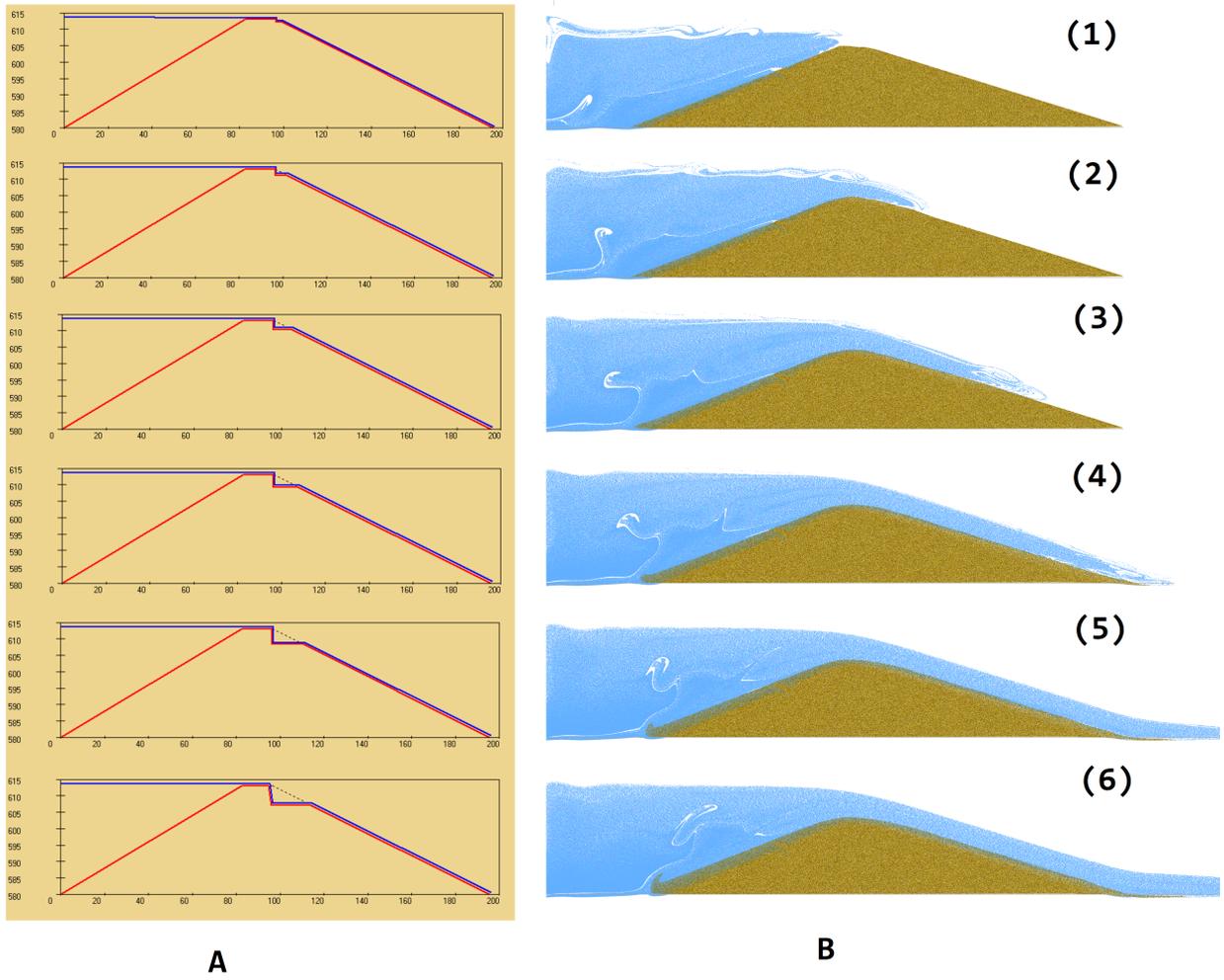


Figure 3.10: WinDAM C 05-HR-OvertopBreach(A) and our simulation(B) method comparison.

ure 3.7(4), the dam structure is more stable during the overtopping breach for the same timestamp. Especially on the right edge of the dam in figure 3.7(4), the dam structure contains less shape deformation than the low cohesive dam structure at the same timestamp compare with figure 3.7(3). The soil with a higher cohesion parameter for the earthen embankment homogeneous cohesive dam will maintain the dam structure from the overtopping dam breach.

For figure 3.9, it demonstrated two initial reservoir setups that will make an impact on the overtopping dam breach erosion along the downslope embankment. From figure 3.9 A(1) to A(4), it shows the empty initial reservoir overtopping breach process at each timestamp. For 3.9 A, the quantified parameters are shown in table 3.4 example 1. The initial reservoir level is 0 meters and dam dimensions including crest width which is 6 meters and dam height which is 3.75 meters. From figure 3.9 B(1) to B(4), it shows the full reservoir overtopping breach process at each timestamp. For 3.9 B, the quantified parameters are shown in table 3.4 example 2. For example 2, The initial reservoir level is 3.25 meters, and dam dimensions including crest width which is 6 meters, and dam height which is 3.75 meters. The result clearly shows that the 3.9 B experience more erosion at the downward levee during the breach process.

Also, WinDAM C<sup>70</sup> 05-HR-OBA example has been applied to proposed MPM simulation, and the comparison result is shown in figure 3.10. WinDAM C<sup>70</sup> is a module used to analyze overtopping earth embankments. figure 3.10 A is the output from WinDAM C. figure 3.10 B are the simulation results using the proposed MPM simulation model. The parameter setting and condition for cohesive dam's overtopping failure cases are shown in table 3.4. The dimension and dam structure specs are set up in the column 05-HR-OBA in table 3.4. WinDAM C simulates the result using the same 05-HR-OBA example. As shown in figure 3.10, from figure 3.10 A WinDAM output, the dam breach erosion process gradually deforms the right top edge of the dam crust. The physical based erosion model used by WinDAM is Hanson/Robinson Stress-driven Model<sup>81</sup>. Hanson/Robinson's model assumes the vertical cut in a cohesive soil was located at a distance of half the bank height back from the edge. Also, the head cut is made to advance through a series of

mass failures driven by erosion at the base. Therefore, the model will erode the cohesive soil on the base surface and form a wedge-shaped notch.

In our MPM simulation result in figure 3.10 B, the head cut during dam overtopping breach has occurred progressively. As the saturation of the water and soil mixture reaches the critical point, the right downward levee starts to erode. The right bottom has enlarged due to the erosion and gradual soil movement and the deformation shape of the dam head cut is not necessary a wedge shape. Compare with the output from WinDAM C, we noticed that the erosion deformation on the levee happened dramatically since for the WinDAM C since it used the Hanson/Robinson Stress-driven Model, and in our simulation result, the erosion process occurred smoothly due to the MPM method calculation based on the interaction between each particle and surrounding grids. table 3.4 also concludes other dam failure examples we simulated using our method. It contains our samples 1 to 5, which are the samples we simulate shown in figure 3.7,3.9. HansonE1S1 is another physical-based experimental dam failure example tested by Hanson<sup>82</sup>.

## 3.12 Conclusion

For the preliminary experiment, the CFD modeling has been introduced to simulate the overtopping dam break without the deformation of the downward levee. The simulation can be run locally and on a cluster to improve efficiency.

With further research, a novel dam breach simulation framework using the Material point method is developed. Based on the simulation results, our MPM method can provide extensive details during the erosion process on the downward levee. The simulation results have been compared with the existing physical-based model like WinDAM. For the current implementation, the model for the momentum exchange and energy density function is a simplified version and this can be improved. Also, the simulation time for our MPM method is 60 seconds, but the simulation time for WinDAM C is around 16 hours. For the current stage, our MPM simulates the process in a shortened time. In the future study, a large scale simulation will be designed, and we will implement the com-

plete version of momentum exchange and energy density function. By optimizing the experiment implementation, better experimental accuracy can be achieved and run time will be reduced at the same time.

Overall, our proposed MPM simulation framework provides rich details during the dam breach process. Unlike simply consider the dam break removal part as a triangle in a physics-based model like WinDAM, our framework can capture the water and soil interaction and gradually remove the soil during the breach process when simulating using the existing dam breach example.

# Chapter 4

## Computer Vision in Agriculture

Computer vision applications have been widely used in various agricultural fields including plant disease, phenotyping, grading and sorting of fruits and vegetables, and precision agriculture. The use of machine learning and computer vision has been increasing in agriculture and seed phenotyping. Making image analysis methods suitable for seed phenotyping applications is currently a driving force in the agriculture field. The principle goal for seed property analysis is to identify seed quality by measuring physical factors that regulate the performance of seeds. From computer vision fundamentals, pictures are constructed in a two-dimensional grid of pixels. Each pixel has a vector of three numbers to specify the red, green, and blue components (in the RGB color space). By applying a color threshold to an image, we can separate the seed from the picture background and measure the seed's physical properties. In some cases, other color spaces are better suited for effectively color thresholding the seeds.

Many image processing applications for seed phenotyping analysis have been investigated, including seed imbibition<sup>83</sup>, seed germination<sup>84–86</sup>, seed size property analysis<sup>37–39;44;87;88</sup>, and seed viability analysis<sup>89;90</sup>. Seed image processing methods such as segmentation, image clustering<sup>91</sup>, and feature extrusion<sup>43</sup> have been developed in the agriculture field. In addition, seed variety classification applications<sup>92–94</sup>, and combined machine learning classification methods using neural networks<sup>85;95;96</sup> have been investi-

gated.

## 4.1 Computer Vision using Two Images

### 4.1.1 Single seed volume measurement

The elliptical cylinder model performs well on a variety of seed samples including milo and wheat. The error rate is below 3% on selected reference objects when the number of slices used is at  $N=50$  or above. Due to the lack of the seed image processing framework for volume measurement in the agriculture field, we propose a framework that can calculate the seed volume by measuring its cross-section using two cameras to gather image information. First, we set up a turntable with two cameras: the top-view and the side-view. Next, we turn on the seed feeder and turntable, and our framework captures pictures for one single seed dynamically. Then, our model makes image analysis on two images of one seed and displays the result in real-time through the framework. Combining knowledge of seed image analysis, our proposed framework applies a state-of-the-art imaging processing model for seed image analysis and achieves rapid measurement of seed volume. This framework significantly improves the efficiency in the lab of seed property analysis for researchers.

The use of machine learning and computer vision has been increasing in agriculture and seed phenotyping. Making image analysis methods suitable for seed phenotyping applications is currently a driving force in the field of agriculture. The principle goal for seed property analysis is to identify seed quality by measuring physical factors that regulate the performance of seeds. From computer vision fundamentals, pictures are constructed by pixels. Each pixel may consist of a vector of three numbers to specify the red, green, and blue components (in the RGB color space). By applying a color threshold to the images, we can separate the seed from the picture background and measure the seed's physical properties.

The physical properties of the seed including length, width, thickness, and volume

are considered the most critical factors to be estimated. Seed volume is among the most important agronomic traits because seed quality and seed traits are heavily dependent upon seed volume. The water displacement method is a common way used to measure the volume of an object. Due to the tiny size of the single seed, these testing procedures can be time-consuming, labor-intensive, and subject to human error. In addition, seeds are porous, so they tend to absorb water making such measurements inaccurate. To solve the seed volume measurement problem, many image analysis methods have been developed. Neilsen, et. al<sup>97</sup> introduces a dynamic, real-time seed counter which is able to process dynamic video footage and generate an accurate seed count. SmartGrain is mobile software<sup>35</sup> that can provide high-throughput phenotyping from a single image and generate approximate seed size information. Roussel<sup>36</sup> uses a volume carving method to perform 3D shape reconstruction to estimate the volume of a single seed. Fıratlıgil-Durmuş<sup>37</sup> applied the oblate spheroid model to estimate the size of legume seeds by image analysis. Razavi<sup>38;39</sup> developed a computer vision system to calculate the physical properties for basil seed and wild sage seed. Sabliov<sup>40</sup> uses an image processing method to determine the volume of agriculture products like lemon, peach, and egg using a single camera. Cervantes<sup>41</sup> focuses on the parameters used to describe seed shape. Pedersen<sup>42</sup> compares the proposed SKWCS system with traditional laboratory measurements using 16 single sorghum kernels.

However, there are limitations in the previous studies, Neilsen<sup>97</sup>, SmartGrain<sup>35</sup>, Sabliov<sup>40</sup>, Cervantes<sup>41</sup>, and Pedersen<sup>42</sup>, all lack seed volume measurement. Roussel<sup>36</sup>'s work requires a high-cost professional camera, and it is not capable of dealing with multiple seeds. Amiryousefi, Fıratlıgil-Durmuş, and Razavi's work only design the computer vision system specific for one seed sample like pomegranate seed<sup>43</sup>, lentil seed<sup>44</sup>, legume seed<sup>37</sup>, wild sage seed Razavi<sup>38</sup> and basil seed<sup>39</sup>. These computer vision systems cannot be used to calculate volume measurements for other seed samples.

To the best of our knowledge, no rapid, accurate, and universal seed imaging framework has been designed for seed volume measurements. To solve this problem, Kansas State University cooperated with USDA to develop TT2Cam, a seed property analysis

framework. The design goal of this framework is to accurately measure seed volume using a turntable to capture multiple images from different angles. The seed is fed by a vibrating feeder while the turntable turns at the same time. We develop a novel and universal seed imaging processing framework to compute the seed volume by capturing two orthogonal images in real-time. The TT2Cam image processing framework is used for seed quality research. Subsequent research uses more than two orthogonal images.

### 4.1.2 Slicing model for volume measurement

Even seeds of the same variety have unique shapes. For empirical analysis, a variety of different types of seeds of different varieties are analyzed. The goal is to derive a universal model that can be used for distinct seed samples. Wheat seeds are generally considered as an ellipsoid or oblate spheroid. Subsequent research will also consider the crease which is typically found on the wheat seed. On the other hand, milo or soybean seeds generally have a spherical shape. In our framework, to calculate the volume of different seeds, we assume a single seed has an ellipsoid shape. The single seed can be more roundish like a sphere or more flat like an elongated ellipsoid. Then, we slice the whole seed, we estimate the single seed volume by adding up the individual slices. We apply an elliptic cylinder model for every single seed, each slice is considered as an elliptic cylinder as shown in figure 4.1. The estimated seed shape is calculated by the aggregation of the small slices.

After removing the background using an image threshold method and slicing the whole seed into several distinct pieces, we get the estimated seed volume by adding each slice together. The volume of each slice is given by the elliptic cylinder model:

$$v_i = \pi A_i B_i h_i$$

Where  $v_i$  is each slice volume,  $A_i$  is one radius of the ellipse,  $B_i$  is another radius of the ellipse, and  $h_i$  is the height of the elliptic cylinder. Therefore, the volume estimation

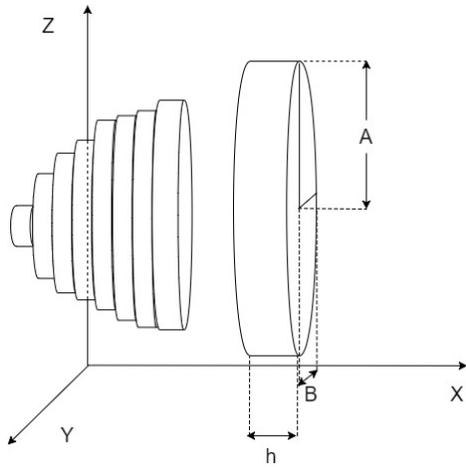


Figure 4.1: Slice segmentation method for seed model

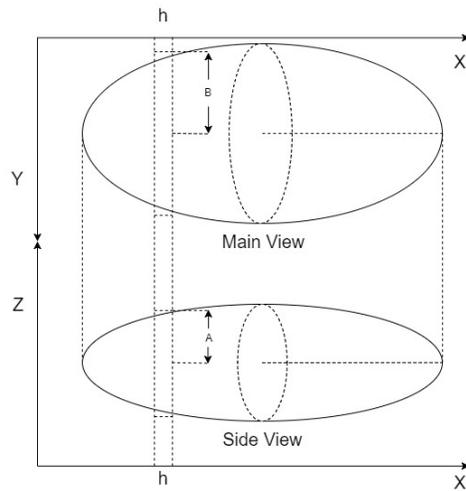


Figure 4.2: Main-view captured image and side-view captured image

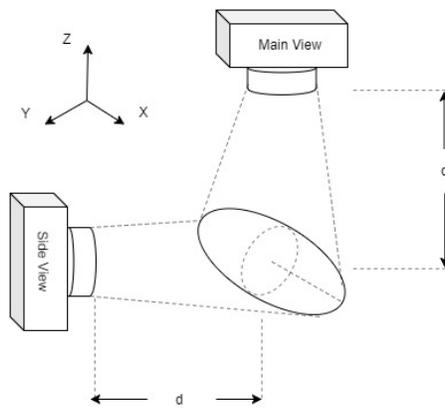


Figure 4.3: Two camera setup. Top: main-view, Left: side-view. Set at equal distance  $d$  from seed

model for a single seed:

$$V = \sum_{i=1}^n \pi A_i B_i h_i$$

Each slice contains  $A_i$ ,  $B_i$ ,  $h_i$  and we consider each slice as an elliptic cylinder model shown in 4.1. For each seed sample, we capture two images and we slice the whole image into many slices shown as figure 4.2.

## Framework setup

The framework consists of three main components: Dynamic Feeding, Image Capture, and Image Analysis.

1. **Dynamic Feeding:** A vibrating feeder and a turntable are used to feed the seed dynamically. The seed queue can be lined up perfectly by the feeder and pass through the camera section one by one on the turntable. The custom-built turntable has been connected to a Syntron Electric Controller with Syntron Vibrating Feeders F-TOC.
2. **Image Capture:** Two cameras are set up to capture a top-view and a side-view. When the seed passes by the front camera, the two cameras take pictures simultaneously. We are using a Techniquip Foi-150 Fiber Optic Illuminator as the light source, an Imaging Source DFK 37BUX287 color industrial camera is mounted as the top camera and an Imaging Source DMK 37BUX287 monochrome industrial camera is mounted as the side camera. Subsequent research uses two color cameras.
3. **Image Analysis:** The framework comes with a user interface for the Windows operating systems. The user interface and image capture is implemented using C# in Visual Studio. Image processing algorithms are implemented using Python and rely on a few libraries from OpenCV.

The two components of the framework: Dynamic Feeding and Image Capture are shown in figure 4.4.

Auto ROI	
Auto Functions ROI Preset	Enabled Center 50%
Exposure	
Brightness	0
Gain(Top color camera)	20.00 db
Gain(Side monochrome Camera)	8.00 db
Exposure	1/2000 sec
Auto Reference	0
Auto Max Value	1/2000 sec
Highlight Reduction	Disabled
Image	
Sharpness	0
Gamma	1
Denoise	0
Filp Horizontal	Disabled
Flip Vertical	Disabled
Partial scan	
Auto-center	Enabled
X Offset	0
Y Offset	0
Special	
Trigger	Disabled
GPIO IN	0
GPIO Out	0
Strobe	Disabled
WDR	
Tone Mapping	Enable, Auto Disabled
Intensity	0
Global Brightness Factor	0
Color(Top color camera only)	
Hue	0
Saturation	100%
White Balance	Auto Disabled
White Balance mode	Grey World
White Balance Red	1
White Balance Green	1
White Balance Blue	3.98

Table 4.1: Camera calibration settings

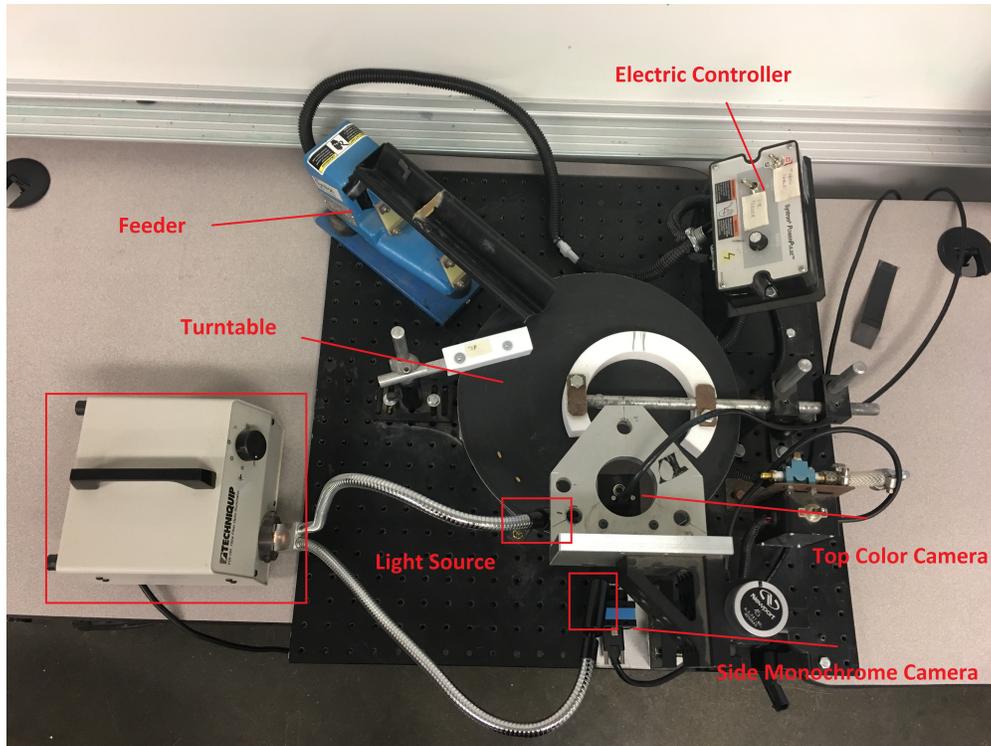


Figure 4.4: The framework setup including a turntable, a light source, one top color camera, one side monochrome camera, a seed feeder, and an electric controller

## Camera calibration

The acquired images without estimating camera pose and lighting source are known to be particularly sensitive to calibration errors. Therefore, requiring the precise position of the cameras and adjustment of light intensity is important for accurate image analysis. For this framework setup, the two cameras are mounted at the same distance as the object. From the top-view and side-view, the distance  $d$  from the object to the cameras are set to be fixed.

1. Light Source: Dual fiber optic lights are placed next to the two camera lenses and have the same distance  $d$  to the seed object.
2. Camera Setting: Our framework is developed using the IC Imaging Control .NET Component; the camera setting can be adjusted by setting device properties as shown in table 4.1.

3. Calibration Reference Object: To calibrate the camera distance and pixel ratio for each frame. We simply test a ceramic ball as a calibration object with  $r_0 = 3.8mm$  radius to calibrate the working distance  $d$  and pixel ratio for our framework.

The user interface is implemented using C#. Imaging Source cameras come with a C# package and libraries for low-level camera control. These libraries are integrated into the framework. Image processing for volume measurement using the slicing method is developed using Python and only relies on libraries from OpenCV. For each frame:

1. Convert the image to the HSV color space. HSV (Hue, Saturation, Value) is an alternative color space that can be used in place of the RGB color space for better color thresholding.
2. Adjust the threshold for the HSV image to filter out the background from the seed object and apply the color threshold to the HSV image.
3. Set the searching area of the region of interest-based on the seed sample.
4. Find the seed image contour using the result of the HSV image thresholding to generate a mask.
5. Construct a fixed orientation bounding box for the contour.
6. Draw the contour on the seed sample image.
7. Calculate the length of the individual slice based on the slice number.  $x_i = \frac{X}{N}$ .
8. Slice the whole seed region based on the contour and the bounding box.
  - First, construct a line that connected the left most and right most points for the contour.
  - Then, compute the middle point in that separating line and use it to separate high and low points on the contour and save them to the list.

- Next, calculate the average of the 5 adjacent high point list and low point list and make a rectangular box for each slice.
- If the adjacent high point cannot be found, in some cases there are no points in contour, we can use the most recent average point for the current slice from memory.

9. Calculate each slice values by applying elliptic cylinder model.  $v_i = \pi A_i B_i h_i$

10. The final step is to sum up all the single slice and get the total estimation of single seed volume  $V = \sum_{i=1}^n \pi A_i B_i h_i$ .

The seed volume measurement value and corresponding images are saved in a project folder. The user can also access the saved Excel spreadsheet which records all of the seed property data including sample name, width, length, thickness, and volume measurements. In the current phase, the framework works in a static mode, and it is not able to capture and process images in real-time. Modifications to make it run in real-time are underway. The seed property analysis results will be displayed when turntable and feeding machines are running simultaneously. In that scenario, our framework will be refined to deal with the heavy load of seed image processing jobs. For our framework, the velocity of the turntable is 53 mm/s, the radius is 118 mm, the perimeter is 741.4 mm and there is about 7 mm space between each adjacent seed, therefore at most 5 seeds pass through the camera per second. Also, our image processing algorithm can process 15 images per second. In future implementation, our framework can handle 5 seeds per second with the current turntable speed.

## Experiment

Analysis experiments are conducted by feeding the seed sample using a vibrating feeder, the electric controller can adjust the feeding speed and the turntable speed while the turntable is running. Seeds are aligned perfectly by the white plastic obstacles placed on the side and middle of the turntable. When the seed dynamically goes through the

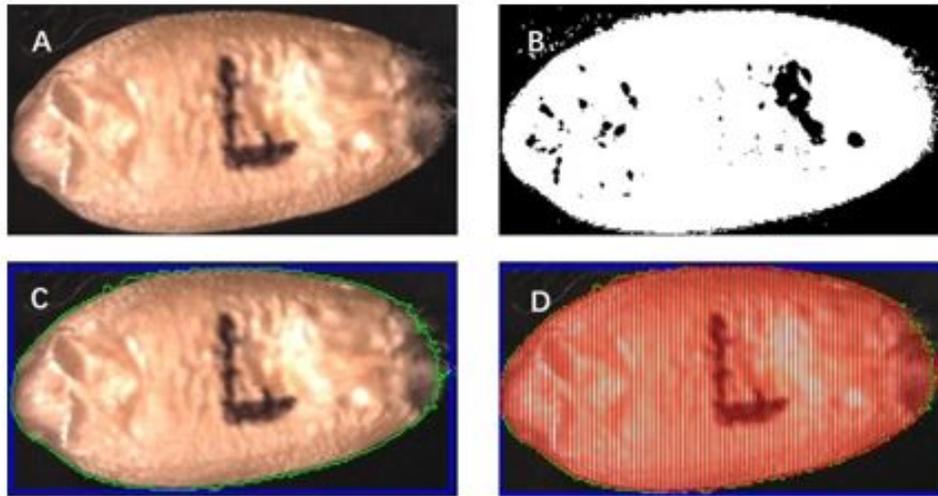


Figure 4.5: Wheat seed (top-view, color camera). A: Source image, B: Binary image after thresholding, C: Bounding box with detected contour, D: Sliced image

front region, the top color camera and the side monochrome camera capture the images simultaneously. The turntable, seed feeder, controller, and light source are shown in figure 4.4.

After images are captured, they are processed by image analysis code developed in Python. The image processing process for a wheat sample is shown in figure 4.5. After the camera calibration setup, the pixel ratio is set to  $p = 46.30$  pixels per millimeter. The main side view is captured using a color camera, which is shown in the A picture in figure 4.5. The first step is to apply the HSV image threshold to it and find the image contour using the binary threshold. For this wheat sample, the HSV threshold range is set to H [0,30], S [60,255], V [0,255]. In that HSV range, we filter out all colors except the roughly brown color which is the wheat color on the source image. The binary image is shown in the B picture in figure 4.5. After processing the source image, OpenCV library methods are used to detect the seed contour, and this contour is displayed on the source image. Due to the OpenCV contour detection algorithm, it finds all the possible contours inside the source image. Normally, the number of possible contours is more than one. Among them, only one contour has a similar size to the wheat seed in the source image, others are typically noise in the background. In our case, the searching area (or region of interest

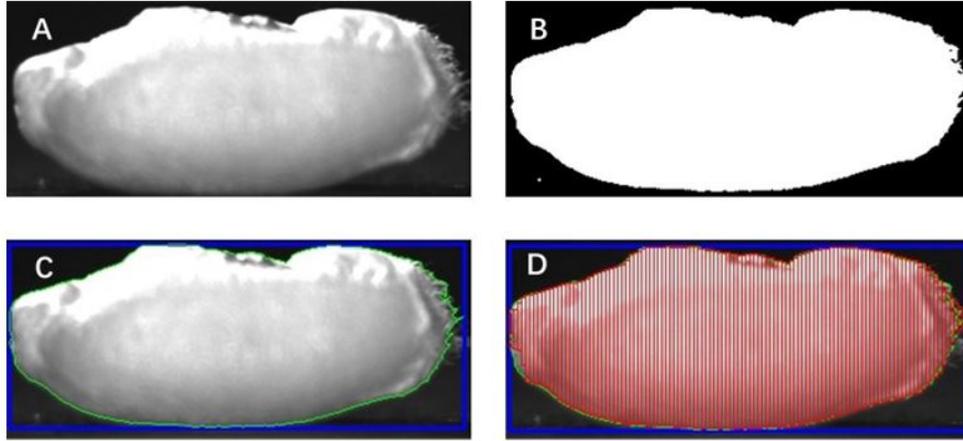


Figure 4.6: Wheat seed (side-view monochrome camera). A: Source image, B: Binary image after thresholding, C: Bounding box with detected contour, D: Sliced image

(ROI) of the contour region range in the interval [15000, 50000] in pixel value. Any contour size which falls outside of that range will be considered as noise and discarded. Once the target contour is found, our implemented algorithm will draw the contour in the picture, shown in green on the C picture in figure 4.5. The bounding box can be generated by minimizing the length and width of the contour constructed as shown as the blue line in the C picture in figure 4.5.

The same image processing algorithm is also applied to the side image captured by a monochrome camera shown in figure 4.6. Before applying our slicing method, we calibrate and adjust the measurement error for the two images. Although we mount both cameras in the same distance  $d$  from the lens to the seed, the side camera may still have some measurement error due to the turntable movement as shown in figure 4.7. The image is taken from closer seed sample distance  $= d - \Delta$  has a larger measurement of length and thickness,  $\Delta$  is measurement error. Likewise, the image taken from further position distance  $= d + \Delta$  has a smaller measurement compared to the distance  $d$ . Therefore, we applied a scale factor of  $s$  to the thickness and length measured in the side image to compromise the measurement error. Since we measure the length both in main and side camera, and the distance  $d$  between the top camera and object won't change during the experiment, we simply set the scale factor  $s = \frac{X_{top}}{X_{side}}$ . In that way, we can deal with radial

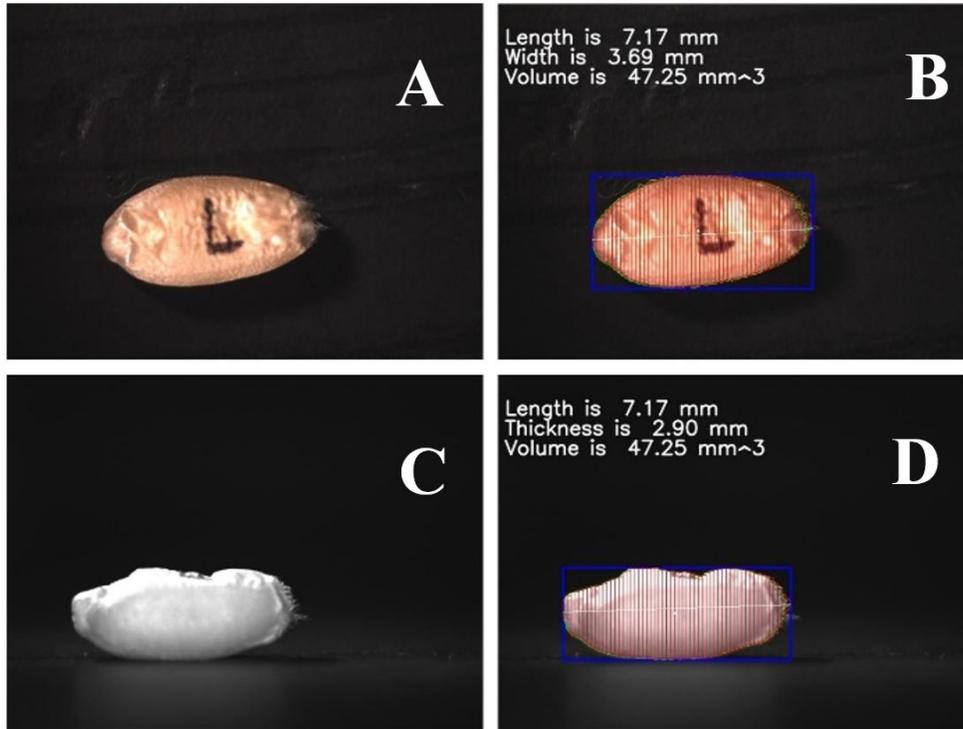


Figure 4.7: Wheat sample with volume measurement results

positioning errors. For the seed rotation problem, we handle it by adding a white plastic ring above the turntable. Due to the shape of the elongated seeds, when the seed is passing by the white plastic ring, the seed orientation will be adjusted to perpendicular to the radius. This system will minimize seed rotation errors when taking the side pictures. After setting the scale factor, the slicing method will be applied to the images of the wheat sample. The first step is to decide on the number of slices to use for each sample. We will further discuss the relationship between the number of slices selected and volume measurement in the next section. For now, we select  $N = 50$  slices for our wheat seed sample. The algorithm divides the length into 50 segments and calculates the average for the adjacent five upper and lower points, then draw and connect the area as a rectangle. The number of points can be smaller, five adjacent points are suitable for saving our most recent point list. A separation line is constructed and shown in figure 4.7, the middle point of each slice rectangle is calculated based on the separation line. We use the middle point from the separation line to distinguish the upper and lower points.

## Slice number and accuracy

We numerically investigate the impact of the number of slices  $N$  on the accuracy of the volume estimation for seed samples. To do so, we calculate the relative error  $E_i = \frac{V_i - V_0}{V_0}$ , where  $E_i$  is the error rate,  $V_i$  is the volume measurement calculated by our framework,  $V_0$  is the physical experiment volume measured by our collaborator at the USDA using a glass bead displacement method. figure 4.8 shows the different number of slice selections applied on the milo seed and gives different results on volume measurement. In this milo example,  $V_0 = 22.91\text{mm}^3$ . figure 4.8 picture A shows the milo sample divided into five rectangle slices  $V_1 = 24.49\text{mm}^3$ ,  $E_1 = 6.89\%$ . figure 4.8 picture B shows the milo sample divided into ten slices  $V_2 = 24.35\text{mm}^3$ ,  $E_2 = 6.28\%$ , and figure 4.8 picture C shows the milo sample divided into twenty slices  $V_3 = 23.18\text{mm}^3$ ,  $E_3 = 1.18\%$ . Comparing these three images, figure 4.8 picture A has the largest volume estimate due to the lowest number of slices, figure 4.8 picture C has the smallest volume estimate. The reason is that with the smallest number of slices, our framework will select the average of the adjacent points to calculate the width and length of an individual slice. These estimations of length will exceed the border of the actual contour of the seed, which will lead to a higher volume estimate in summation. When the number of slices increases in figure 4.8 C, the aggregation of the slices is closer to the contour shape of the milo seed, and it will achieve better results. figure 4.8 shown the experiment using the milo sample, we want to test our framework using other objects. To reduce human error when conducting the glass bead displacement method for volume measurement, we use a known volume reference. In the next step, we further discuss the impact of the different number of slices and error rates using reference objects. Our goal is to minimize the error rate of our volume measurement method.

Therefore, we tested the different number of slices and their corresponding error rate for two reference objects shown in figure 4.9. Sphere1 and sphere2 are two known volume reference ceramic bearing spheres of diameters 3.18 mm and 6.35 mm respectively. We conducted our experiment with the slice number selection from the range  $N = 5$  to 150.

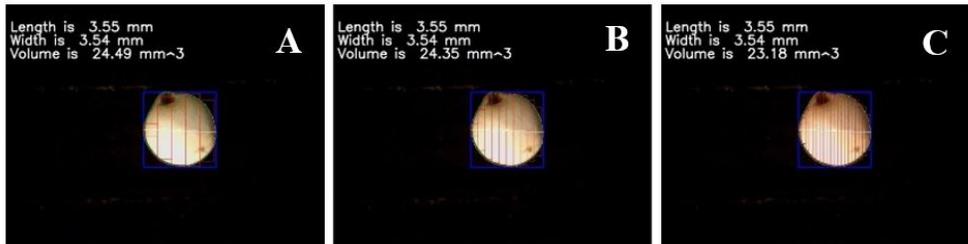


Figure 4.8: Milo sample with different numbers of slices N = 5 (A), N = 10 (B), N = 20 (C)

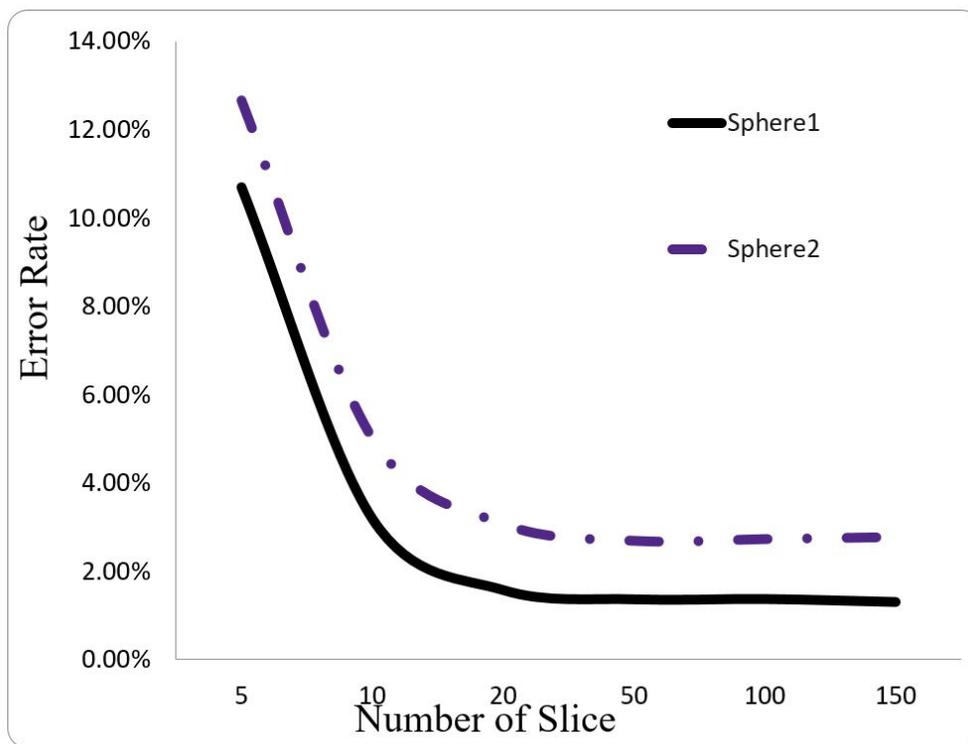


Figure 4.9: Volumetric error for different slice numbers of two ceramic spheres. Sphere 1 diameter = 3.18 mm; Sphere 2 diameter = 6.35 mm.

From figure 4.9, we can see that the error of volume measurement goes down as the number of slices goes up. As the beginning of the graph, the error rate is at a very high level above 10% when the number of slice  $N$  is 5. As we increase the number of slices, the error rate is significantly decreased, and the curve almost flat when the slice number  $N$  is greater than 50. The reason for this experiment result is that for the larger number of slices, the aggregation of the slices is close to the actual shape of the ceramic sphere. When we increase the number of slices, the error rate goes down. For figure 4.9, the graph shows that when the slice number  $N$  is greater than 50, for both reference object sphere1 and sphere2, the volumetric error rate is relatively low at less than 3%. According to the relationship between error rate and slice number  $N$ , we conclude that a good trade-off is to use  $N = 50$  as the number of slices for analysis. For the wheat and milo examples in this paper, the number of slices is set to  $N = 50$ .

We also investigate the camera calibration and lighting conditions to improve measurement accuracy. The light intensity and light color perform a crucial role in the image analysis procedure. We added an adaptive camera calibration algorithm to reduce the noise of the light source. By skipping sophisticated calibration steps, an adaptive threshold algorithm can fine-tune the camera setting and reduce the background noise of the single seed and extract the region of interest as a prior process. This added feat is very helpful for reducing noise and improving the performance of our framework.

## Volume measurement for different samples

For our experiment, we tested our algorithm on wheat seeds and milo seeds. As shown in table 2, there are two wheat seed samples named L and O, and twenty milo seed samples. The samples are provided by our collaborators at USDA in Manhattan, KS. The values  $WL_g$  and  $WO_g$  are the volumes of the L and O samples measured using a glass bead displacement method at USDA, as our reference for the wheat samples. For the wheat example,  $V_{WLg} = 51mm^3$ ,  $V_{WL1} = 49.25mm^3$ ,  $E_{WL1} = 3.43\%$ .  $V_{WL2} = 47.62mm^3$ ,  $E_{WL2} = 6.62\%$ .  $V_{WOg} = 21.1mm^3$ ,  $V_{WO1} = 18.86mm^3$ ,  $E_{WO1} = 10.62\%$ .  $V_{WO2} = 17.76mm^3$ ,  $E_{WO2} =$

Sample	Length (mm)	Width (mm)	Thickness (mm)	Volume (mm <sup>3</sup> )	Weight (mg)	Density (mg/mm <sup>3</sup> )
WLg	7.17	3.64	2.91	51.00	87.00	1.71
WL1	7.17	3.69	2.90	47.25	87.00	1.84
WL2	7.11	3.71	2.86	45.62	87.00	1.91
WOg	5.56	2.23	2.10	21.10	31.00	1.47
WO1	5.83	2.26	2.26	16.86	31.00	1.84
WO2	5.70	2.30	2.05	14.86	31.00	2.09
M1-1	3.87	3.76	2.46	19.55	31.80	1.63
M1-2	3.81	3.80	2.80	24.20	30.40	1.26
M1-3	4.04	3.72	2.98	28.18	37.30	1.32
M1-4	3.85	3.41	2.54	19.52	37.00	1.90
M1-5	3.75	3.68	2.73	25.08	29.50	1.18
M2-1	3.89	3.21	3.09	23.07	23.10	1.00
M2-2	3.91	2.70	3.00	21.91	18.70	0.85
M2-3	3.39	2.83	2.55	9.90	16.80	1.70
M2-4	3.88	3.27	2.68	23.95	20.60	0.86
M2-5	3.82	2.86	2.20	16.65	17.90	1.08
M3-1	3.61	3.41	2.26	15.86	21.10	1.33
M3-2	3.75	3.62	2.55	25.40	23.90	0.94
M3-3	3.42	3.42	2.10	13.26	20.90	1.58
M3-4	3.35	3.28	2.53	13.66	18.90	1.38
M3-5	3.55	3.54	2.86	25.04	26.40	1.05
M4-1	3.79	3.70	2.32	21.67	26.30	1.21
M4-2	4.41	3.85	2.47	21.60	33.30	1.54
M4-3	3.80	3.71	2.31	21.91	24.60	1.12
M4-4	4.16	3.96	2.74	26.41	36.80	1.39
M4-5	4.50	4.23	2.91	30.44	37.60	1.24

Table 4.2: Volume measurement of wheat samples

15.82%. From the experimental results, we can conclude that our volume measurement method performs very well on the wheat sample overall. The large error rate for the O sample is because of the small size of the wheat seed. Due to the small size of the wheat seed, the glass bead displacement method from USDA may introduce human error for calculating the  $V_{WOG}$ .

M1-1 to M4-5 denote the twenty different milo seed samples. All of the volume measurements use  $N = 50$  slices when computing volume. Accurate weight data can be easily obtained to compute the density as mass/volume. To evaluate the performance of our algorithm, we compare our milo density with previous works. Pedersen's work<sup>42</sup> uses their SKWCS method to calculate the single kernel characteristics. Pedersen compares his SKWCS method with traditional laboratory measurement data using single kernel data for 16 milo (sorghum) seeds. The average milo density from the Pedersen SKWCS system is 3.70 g/cm<sup>3</sup>. The average density from Traditional Laboratory Measurements is 1.39 g/cm<sup>3</sup>. The average density from our imaging processing method is 1.25g/cm<sup>3</sup>. For the single milo seed, the Pedersen SKWCS system has 1.31 g/cm<sup>3</sup> difference with the laboratory results, our method outperformed his system and achieve 0.14 g/cm<sup>3</sup> difference with the laboratory results. Roussel<sup>36</sup>'s model achieves great volume measurement results using 36 images for a single seed. They use a ball-bearing ball and measured the diameter use a digital sliding caliper and calculated the volume 14.137 mm<sup>3</sup> ±0.007 mm<sup>3</sup> with 0.05% volume tolerance. Their algorithm measures the volume of 14.11 mm<sup>3</sup> and mean diameter of 2.998 mm for the bearing ball. This is a relative error of -0.19%. The absolute error rate of their method can be lower than 2.5%, but it requires lots of images and a high-end camera. Our image processing framework only requires two images for one single seed, our model is more time-efficient. Other than wheat and milo seed, we continue testing different seed samples. Unlike Amiryousefi, Firatlıgil-Durmuş, and Razavi's work only targeting pomegranate seed<sup>43</sup>, lentil seed<sup>44</sup>, legume seed<sup>37</sup>, wild sage seed Razavi<sup>38</sup> and basil seed<sup>39</sup>, our image processing framework are capable to measure different kinds of single seed samples and achieve great measurement results simultaneously.

## Conclusions and future work

Our framework provides a user-friendly interface, a Windows application TT2Cam for agriculture researchers to use, and conduct image analysis. The volume measurement algorithm designed in Python is integrated into the framework. Researchers can easily capture two pictures and get the measurement result of a single seed from image analysis conveniently. The algorithm provides fast, novel, and universal image processing solutions for seed volume measurement. Based on the main view image captured by a color camera, side view image captured by a monochrome camera, the algorithm can detect the contour of the single seed, automatically slice the whole seed, apply the scale factor for calibration purpose, and finally estimate the volume of a single seed. For image processing, only two cameras are used to gather pixel information. There is a possibility to get more information by adding more cameras from different angles. For the camera setting and orientation. Our framework has one color camera and one monochrome camera. The reason we use one monochrome camera is that the image sourcing monochrome camera has NIR wavelength data collection. In the future, we can also investigate more advanced color analysis by extracting NIR wavelength data from the monochrome camera. We plan to add more features to our TT2Cam framework. Seed volume measurement is our first step, trigger function in real-time, color information extraction, and automated seed density measurement will be our future steps.

## 4.2 Computer Vision using Multiple Images

Modern seed breeding programs require the ability to analyze seeds efficiently to be useful. Even simple measures such as volume and density can be challenging to compute efficiently with modest equipment. Accurately measuring seed volume becomes a highly under-constrained problem. Multiple images from different perspectives are required.

We present an efficient and affordable 3D single seed volume measurement system to extract image contours and compute volumes using a modified volume carving method in

a controlled lab environment. The framework is constructed with a turntable, a stepper motor controlled by an Arduino microcontroller, three orthogonal cameras, and camera control via a modest computer used for data acquisition and processing. For testing, images are captured using only a side camera from different angles by rotating the turntable. Then, the framework processes the multiple images in parallel and reconstructs 3D seed objects to calculate the volume based on the voxel numbers. The proposed framework: (1) generates single seed 3D geometries for visualization, (2) calculates precise seed volumes within seconds, and (3) achieves less than a 3% error rate on a reference ceramic sphere.

The principle goal for seed property analysis is to identify seed quality by measuring physical and physiological factors that regulate the performance of seeds. Among those physical factors, seed volume and density are among the most crucial aspects to measure since they are highly correlated with seed functionality and quality. There are several existing seed physical property analysis models using image processing including 2D affordable scanning<sup>28-30</sup> and commercial software designed for different seed types<sup>31;32</sup>. These models calculate typical parameters including width, length, and area from the 2D images. They can be used to analyze different seed types such as rice, soybeans, and barley. Other than these specific seed image processing frameworks to measure physical seed factors, we can also apply general image processing models for volume measurement. From a geometric perspective, there are stereo-based techniques, space-carving methods, and newly developed methods based on convolutional neural networks (CNNs). Although CNN methods may use fewer images, effective and accurate solutions typically require multiple images.

For seed property analysis, 3D reconstruction modeling can be applied to solve the volume measurement problem and conduct property analysis. Reconstructing and recovering the 3D shape of a single object is a long standing ill-posed problem, which has been explored by computer vision, computer graphics, and machine learning communities. The single object 3D reconstruction model has made enormous progress and rapid revolution since 2015<sup>98</sup>. The goal of image-based 3D reconstruction is to infer the 3D

geometry and structure of objects from one or multiple 2D images. The recovery of a 3D shape using multiple 2D images has been extensively investigated.

In order to achieve high quality 3D shape reconstruction, volume carving with shapes generated from the silhouette method<sup>45–47</sup> is used for reconstruction. A silhouette of an object in an image refers to a contour separating the object from the background. Shape-from-silhouette methods require multiple views of the scene taken by cameras from different viewpoints. For each image, the silhouette of the target objects is segmented using background subtraction. The retrieved silhouettes are back-projected to a common 3D space with projection centers equal to the camera locations<sup>99</sup>.

In this section, we present a novel measurement framework for single seed volume measurement using a modified 3D reconstruction volume carving method. Although existing software and models have been well-known or developed in the literature and in practice, we propose a complete, end-to-end system for the volume measurement of a single seed. Our proposed work achieves high accuracy from multiple images automatically acquired and processed.

To demonstrate the accuracy of the proposed framework, a reference ceramic ball is reconstructed with less than a 3% error rate. In cooperation with collaborators from USDA, the framework is used to measure a variety of seeds, including wheat seeds. The computed results are compared with the results obtained directly using beads and volume displacement measurement. The results indicate that the proposed framework is highly accurate and usable in practice. It is also much faster than physical measurement using volume displacement.

#### **4.2.1 Volume carving framework for volume measurement**

##### **Shape from silhouettes**

The shape-from-silhouette method for single seed volume measurement requires multiple images from distinct views. For each image, the silhouette of the target objects is segmented using background subtraction. The retrieved silhouettes are back-projected

to a common 3D space with projection centers equal to the camera locations.

We use the intrinsic camera matrix  $K$  and the distance between the origin of our working volume and the camera center from the calibration. The origin of the working volume is selected to be the intersection point (IP) of the seed bottom horizontal line and the centerline in the image<sup>36</sup>. We acquire  $N$  images, showing a seed under equidistantly spaced rotation angles  $a_i$  where  $i \in \{1, \dots, N\}$ . The rotation is around the vertical axis through the IP and is parallel to the  $y$ -axis of the camera. We apply a grayscale threshold method on each image and segment each image into a binary mask  $M_i$  for  $i \in \{1, \dots, N\}$ . For each image, we calculated the camera projection matrix  $P_i$  from the rotation angle  $a_i$  by

$$P_i = K(R_i) \quad (4.1)$$

Where  $R_i$  is the rotation matrix corresponding to the given angle  $a_i$ ,  $T_i$  is the translation matrix for the optical center. After that, we define an equidistantly spaced cubic voxel grid around the world origin, and set each voxel size to  $1mm^3$ . Each voxel center with homogeneous world coordinates  $\vec{X}$  is projected to a point  $\vec{x}_i$  in each mask  $M_i$  by

$$\vec{x}_i = P_i \vec{X} \quad (4.2)$$

If a voxel belongs to the foreground object, its value  $V(\vec{X})$  is set to 1. If the voxel does not belong to the foreground object, its value  $V(\vec{X})$  is set to 0.

$$V(\vec{X}) = \prod_{i=1}^N M_i(\vec{x}_i) \quad (4.3)$$

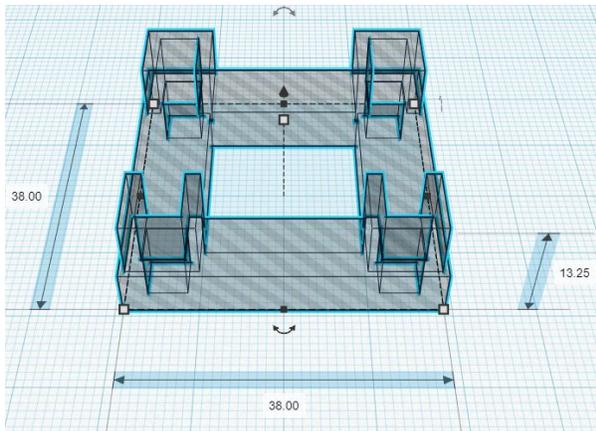
The mask  $M_i$  in this volume carving method is sensitive to misalignment of the object volume. Camera calibration to minimize misalignment is discussed in the next section.

## 4.2.2 Turntable framework setup

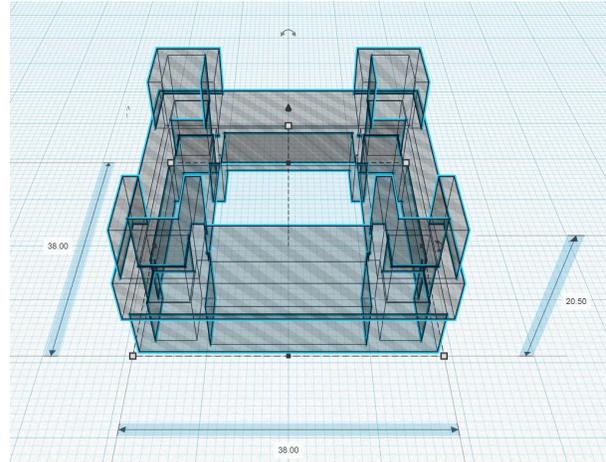
The hardware design is a relatively simple, convenient turntable design. The system includes a NEMA 23 stepper motor to control the rotation of a 3D printed turntable

with a minimum step size of 1.8 degrees (a maximum of 200 steps per rotation), three orthogonal cameras, and an LED light as shown in figure 4.14.

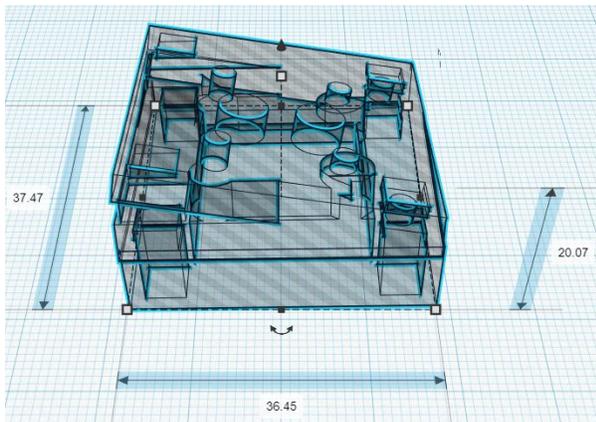
- Turntable control: A NEMA 23 stepper motor controls the base and a 3D printed custom-sized turntable is designed to handle a range of seeds at the required camera distance. The seed samples can be placed in the center of the turntable perfectly. The stepper motor is connected to a power supply and controlled using a low-cost Arduino microcontroller using a Synthetic gShield stepper driver.
- Physical framework: For printing the 3D parts, a MakerGear M2 3D printer is used<sup>100</sup>. For each part, a 3D stereolithography (STL) model is designed using Coreform Cubit<sup>101</sup>. The STL 3D camera mounts are shown in figure 4.10. All parts are designed in millimeters. After the STL model is designed, they are sliced into gcode using slic3r<sup>102</sup>. Finally, 3D parts are printed on the Makergear M2 3D printer using the Printron<sup>103</sup> software or the Makergear M2 software.
- Image capture: The image capture component is built using three ImagingSource color cameras DFK 37BUX287<sup>104</sup>. These cameras are mounted orthogonally at three different angles: front, left, and top, each 90 degrees apart. The images are taken using the ImagingSource IC measurement software.
- Controller and power supply: The stepper motor is controlled by a DFRduino UNO R3 board<sup>105</sup> attached with a Sythetos gShield v5 board<sup>106</sup>. A Universal Gcode Sender (GCS)<sup>107</sup> is used to test the whole system. Then, simple control software is developed to control the system by sending gcode strings. The turning degree and capture rate can be controlled by the user based on their needs. For example, a rotation of 360 degrees requires a step size of 6.4. If  $N = 36$  equidistant images are desired, the turntable should rotate 10 degrees on each step; therefore, the rotation is set to a  $6.4/36 = 0.178$  step size for 10 degrees.
- Mounts and stands: All camera mounts are 3D printed, the base is built using simple Legos to stabilize the whole system. The mounts and parts are designed



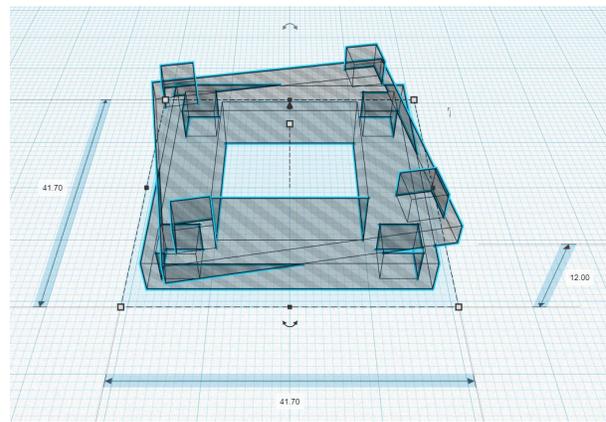
(a) Mount bottom



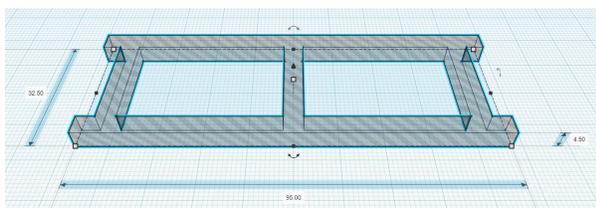
(b) Mount join



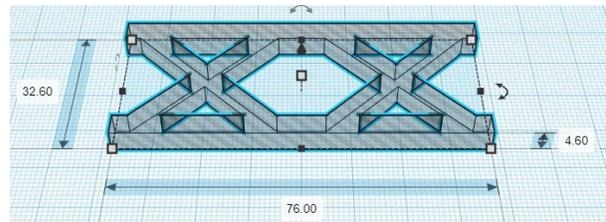
(c) Rotate 10 degrees in X and Y direction



(d) Rotate 10 degrees to the left



(e) Pillar



(f) Pillar with crossbars

Figure 4.10: Selected camera mount 3D models

to be assembled together. For example in figure 4.10f, the length of the pillar is 76.00 mm, the width is 32.50 mm, and the height is 4.60 mm, and in in figure 4.10a,4.10b,4.10c,4.10d, the length of the square is 5.00 mm to fit the pillar leg which length is 4.60 mm.

## Camera calibration

The acquired images without appropriate parameters, adapting camera pose, and adjusting lighting sources are known to be particularly sensitive to calibration errors. Therefore, controlling the precise positioning of the cameras, determining the accurate photo parameters, and adjusting light intensity are crucially important for accurate image analysis. For the framework setup, three cameras are mounted at the same distance as the object. From the top view, side view, and front view. The distance  $d$  from object to cameras are set to be fixed. To reduce glare and focus on the turntable, the camera angle is rotated by 10 degrees as shown in figure 4.10c,4.10d. For the testing phase, only the front camera is used to capture multiple images when rotating the turntable. Future testing will involve all three cameras.

- Camera Position: As shown in figure 4.11, the cameras are mounted at different angles, but they are all orthogonal to each other. To better capture images without glare from overhead lighting, the front camera is rotated 10 degrees along the x-axis towards the turntable, the side camera is rotated 10 degrees along the y-axis towards the turntable, and the top camera is rotated 10 degrees along both the x and y-axes away from the turntable.
- Camera Parameters: The image properties and camera properties of the framework are shown in table 4.3. ImagingSoure IC measurement software is used to capture raw image data. The camera settings for the software is shown in table 3.3.
- Light Source: One goose-neck LED light is placed next to the left side camera, the light source is set on top at a distance  $d$  from the object.
- Calibration Reference Object: To calibrate the camera distance and pixel ratio for each frame, we simply test a ceramic ball of a known fixed size as a calibration object to calibrate the working distance  $d$  and pixel ratio for the framework.

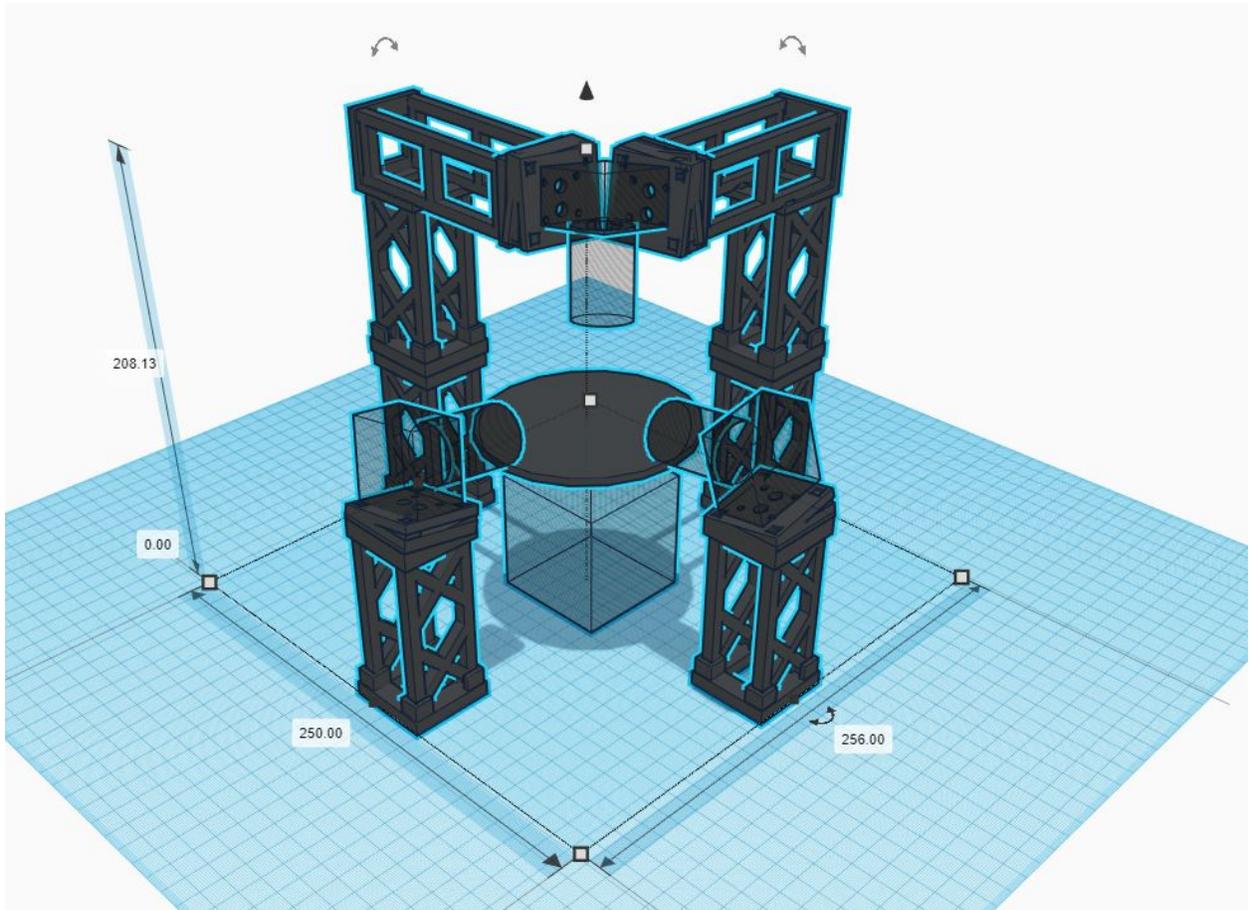


Figure 4.11: Turntable setup with 3D printed camera mounts in STL model

Auto ROI	
Auto Functions ROI	Enabled
Preset	Center 50%
Exposure	
Brightness	0
Gain(Top color camera)	0 db
Exposure	1/30 sec
Auto Reference	128
Auto Max Value	1/30 sec
Highlight Reduction	Enabled
Image	
Sharpness	0
Gamma	100
Denoise	0
Image size	720 x 540
Pixel per mm at working depth	145/5.74
Pixel size	1/0.0062
Focal Length	12.5
Color	
Hue	0
Saturation	64
White Balance	Auto Disabled
White Balance mode	Temperature
Auto-Preset	Auto Daylight
Temperature Preset	Cool White LED
Temperature	7500

Table 4.3: Camera calibration setting

## Software implementation

The framework software is implemented using Python. When capturing the raw data, the Imaging Source camera is controlled by the IC Measurement Software. For each experiment:

1. Determine the number of images  $N$  taken by the front camera. Therefore, the degree between adjacent images is  $D = \frac{360}{N}$ .

2. Crop and trim images in the input sequence, and save the offset for each image for image matching using normalized cross-correlation.
3. Convert each image to the HSV color space. HSV (hue, saturation, value) is a better alternative representation used in place of the RGB (red, green, blue) color space for better thresholding.
4. Adjust the threshold of the HSV image to filter out the background from the seed object and apply a threshold to the HSV image to obtain a binary mask image.
5. Get the total number of images  $N$ , and set the turntable center using pixel coordinates.
6. Apply volume carving method to reconstruct seed shape.
  - First, the software finds the center of the reconstruction cuboid by finding the intersection point (IP) of the seed bottom horizontal line and the centerline in the image. A cuboid is constructed outside of the object defined by voxels. Each voxel size is  $1 * 1 * 1$ , and initially, we fill the cuboid with the voxels.
  - Then, for each image of a different angle, the software trims, crops, and generates the mask image from raw input. The projection matrix<sup>108</sup> for each image is computed and the background is carved out.
  - The reconstructed 3D shape is the object that remains after carving out all  $N = 36$  images.
7. For each voxel, the volume is  $v_i = 1*1*1$ , the total voxel volume of the 3D reconstruction seed is  $V = \sum_{i=1}^n v_i$ . The total volume of the seed is calculated by multiplying by a factor, which is the cubic volume of a pixel per mm in camera properties.

## Experiment

Each camera view captured one image from its visual hull and carve out the background of the images. The camera is using the central projection method shown in figure 4.12.

Therefore, we will notice the surplus volume as a measurement error when creating 3D objects. We further investigate the 3D reconstruction object using a different number of camera views. From figure 4.13 by Roussel<sup>36</sup>, we can see parallel projection and central projection for the 3D reconstruction on the reference sphere with the different number of camera views.

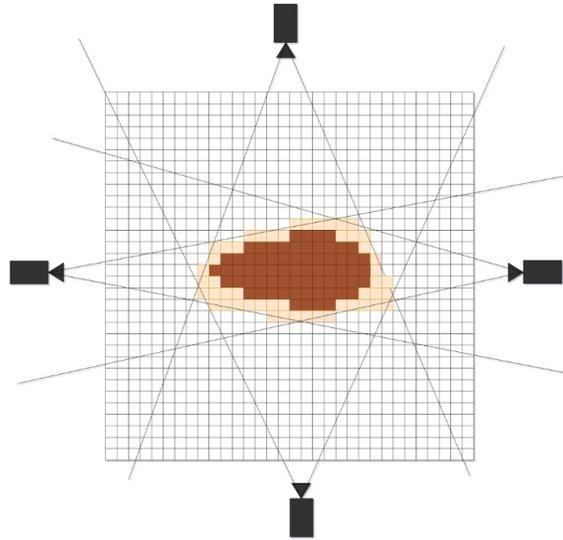


Figure 4.12: 2D view for volume carving method using voxel grid

For demonstration purpose, we conduct our preliminary experiment on the soybean. The images are captured, processed, and analyzed show in the folder in figure 4.15. The 3D reconstruction model of soybean is shown in figure 4.16.

As shown in table 4.4, we applied our volume carving framework on two reference ceramic spheres with known diameters of 3.95 mm for computed sphere volumes of 32.27 mm<sup>3</sup>. We did 4 repetitions on each sphere and computed the average sphere volume and all statistical analysis in table 4.5. For the reference spheres, our method achieves accurate results, the coefficient of variance is around 1%. Our method gets slightly larger estimates than the actual volume of the sphere.

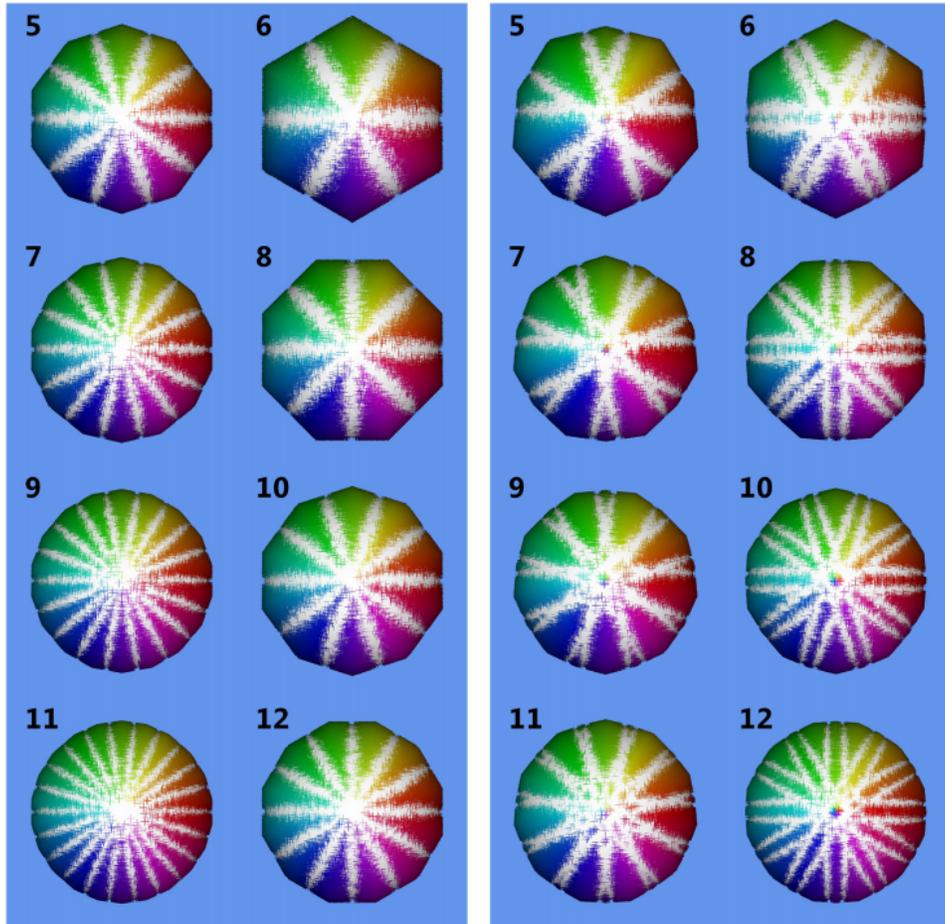


Figure 4.13: 3D sphere reconstruction using parallel projection (left) or central projection (right)

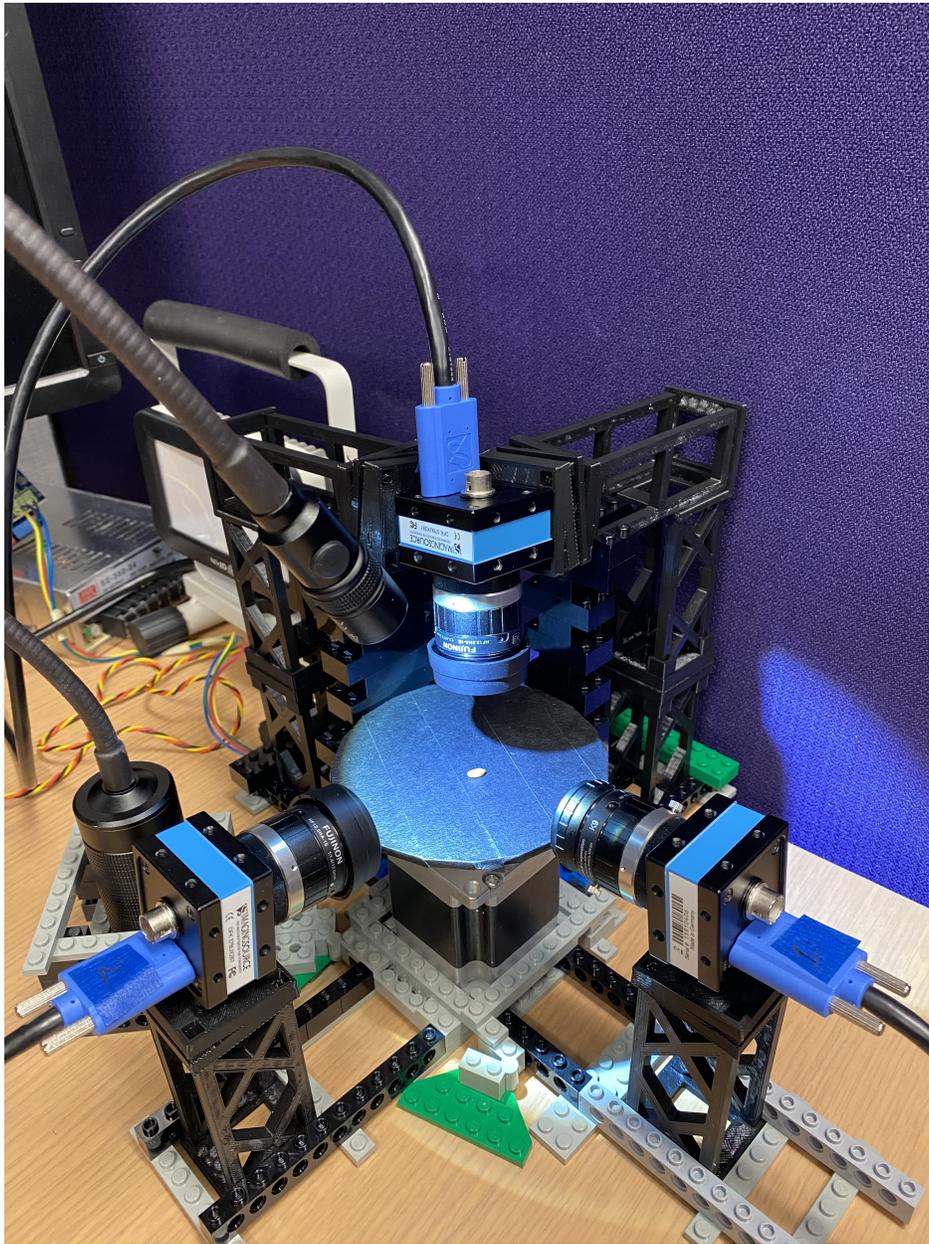


Figure 4.14: Single seed 3D reconstruction framework

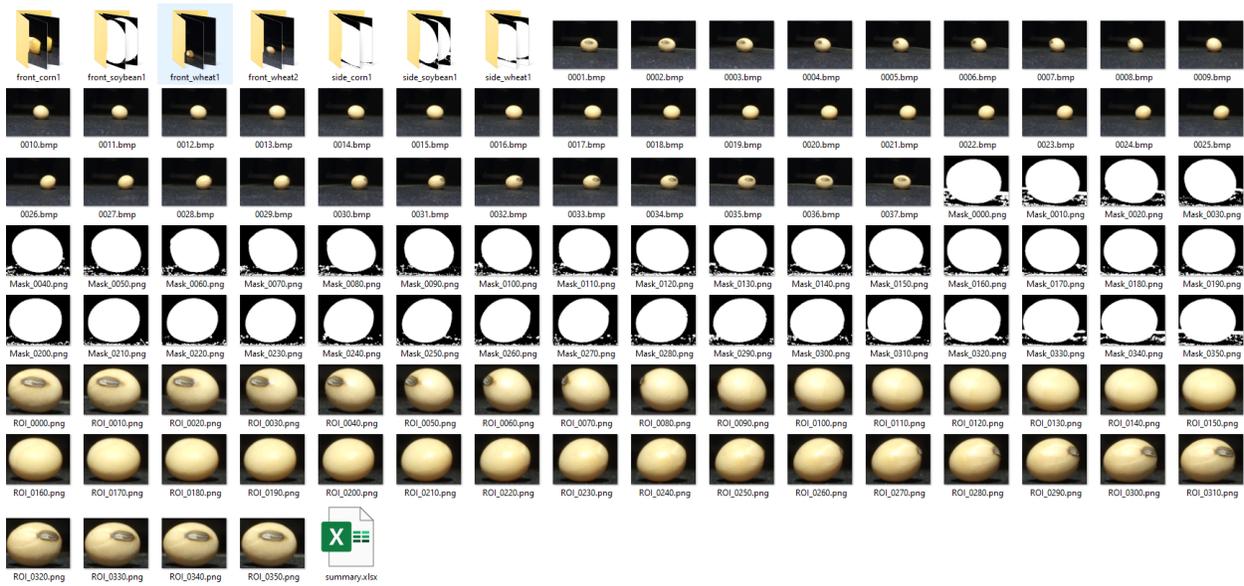


Figure 4.15: 3D reconstruction profile pictures for soybean

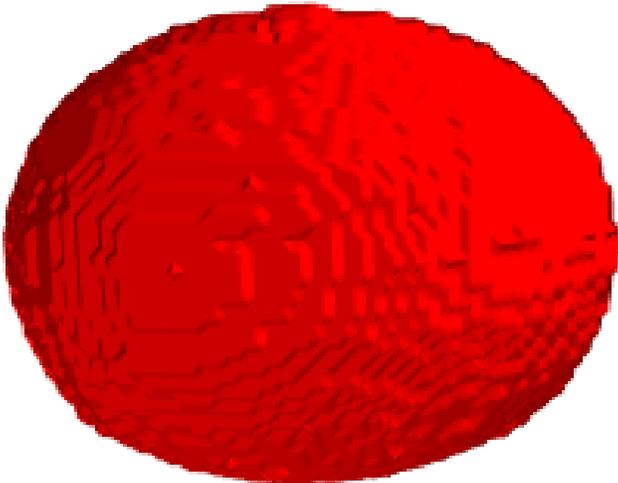


Figure 4.16: 3D reconstruction model for soybean

Sphere Sample	Volume (mm <sup>3</sup> )	Difference	Absolute Relative Error(ARE)
1_1	33.41	1.14	3.53%
1_2	33.08	0.81	2.51%
1_3	33.12	0.85	2.63%
1_4	32.65	0.38	1.18%
2_1	33.59	1.32	4.09%
2_2	33.68	1.41	4.37%
2_3	32.92	0.65	2.01%
2_4	33.89	1.62	5.02%

Table 4.4: Volume data computed for reference spheres

	Average volume(mm <sup>3</sup> )	Sphere volume(mm <sup>3</sup> )	Standard Deviation	Coefficient of Variation(CV)
Sphere1	33.07	32.27	0.31	0.95%
Sphere2	33.52	32.27	0.42	1.25%

Table 4.5: Summary data for reference spheres

## Results and analysis

### Number of images and accuracy

After implementing the code and setting up the turntable framework, results are computed from many wheat samples. But, before applying our framework to small seeds, reference objects are used to test the accuracy of shape reconstruction from the silhouette volume carving method. Shapes reconstructed from the silhouette 3D reconstruction using a finite set of objects images results in an approximated visual hull of the object. The object obtained by the visual hull is guaranteed to enclose the object in 3D space. However, some surplus volume is typically produced in the visual hull by the finite set of objects images shown in figure 4.12,4.13 because the seeds are not totally concave, they also have convex regions. The four cameras project the orthogonal visual hull in 2D space. The brown region represents the wheat seed object, and the light brown region represents the 3D reconstruction volume from the visual hull. It is challenging to engrave the surplus region using only four images.

To further investigate the relationship between the number of images captured and

volume measurement accuracy, an experiment is conducted using a reference ceramic sphere with radius  $r = 1.975\text{mm}$ , the theoretical volume is  $32.75\text{mm}^3$ . For the experimental design, different numbers of images  $N$  are acquired from rotating the turntable at a  $D = \frac{360}{N}$  degrees in each step. For example, if  $N = 5$ , the turntable is rotated  $\frac{360}{5} = 72$  degrees for each step and a total of 5 images are captured.

Before the experiment, we calculated the arithmetic volume of the reconstructed object when using a central projection. If  $N = 4$ , the reconstruction object is exactly a Steinmetz solid<sup>109</sup>, and the volume is

$$V_{\text{steinmetz}} = \frac{16r^3}{3} \quad (4.4)$$

and the actual volume of a sphere is

$$V_{\text{sphere}} = \frac{4\pi r^3}{3} \quad (4.5)$$

Therefore, we calculate the theoretical reconstruction volume using our framework.

$$V = \frac{4N}{3} \left( \tan\left(\frac{180}{N}\right) \right) r^3 \quad (4.6)$$

Different numbers of images  $N$  have been selected and to evaluate the performance of the volume measurement using a ceramic ball. The 3D reconstruction object is shown in figure 4.17 and the line chart is shown in figure 4.18.

To test the reconstruction shape and volume measurement using the different numbers of images  $N$ , we set the number of images from 5 to 12. figure 4.17 shows for  $N = 12$  number of images yields much better reconstruction than for  $N = 6$ . This is in full agreement with the measurement error in figure 4.18. From the experimental data, we observe that for the  $N = 12$ , the theoretical error rate is below 3%. As the number of images goes up, the error rate goes down. When we select  $N = 36$ , the error rate is also below 3%. In practice, other error sources may dominate when we select a higher number of  $N$ . For the

number of image value selection, we want to achieve higher accuracy in the controlled environment. Therefore, based on the  $N$  parameter analysis of the reference ceramic sphere, we achieve an experimental error rate of less than 3%, and we apply  $N = 36$  as the number of images value to the experiment samples including wheat, soybean, and corn seeds.

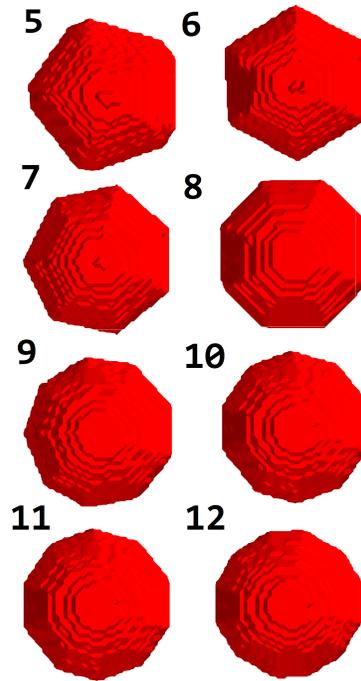


Figure 4.17: Comparing number of images  $N$  for 3D volume reconstruction using reference ceramic ball

The framework is tested on different seeds, the 3D reconstruction and volume measurement data for wheat, corn, and soybean seed is shown in figure 4.19. Four pictures of each seed from 0, 90, 180, and 270 degrees are selected to show the comparison between original seed pictures and reconstruction 3D models. For wheat seed, the framework is able to accurately calculate the volume and also capture the brush end and germ end details from the seed shape property. Also, for the corn seed and soybean seed, the overall performance of 3D reconstruction is great except for the imperfection for the concave and dark region as shown in the 270-degree picture for corn.

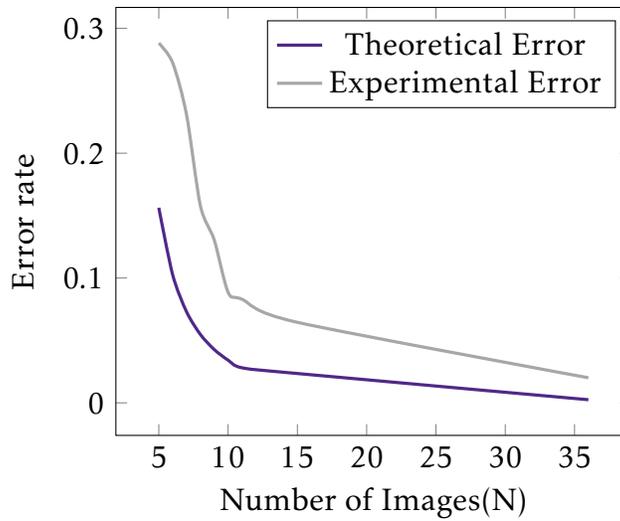


Figure 4.18: Error rate when using different number of images (N)

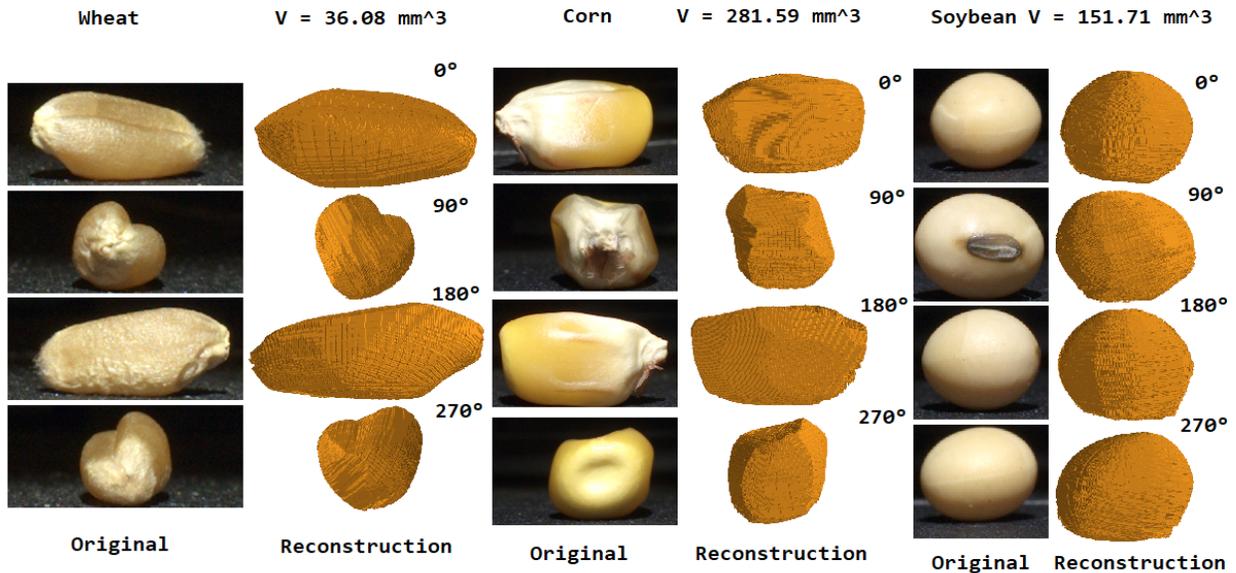


Figure 4.19: 3D reconstruction and volume measurement for wheat, corn and soybean seeds

### Voxel size

To understand the voxel size impact on the performance of the volume measurement, we conduct complexity and accuracy analysis for the volume carving method. The complexity of the volume carving algorithm depends on the number of voxels and the number of images acquired. Run time experimental data is shown in table 4.6. We select different

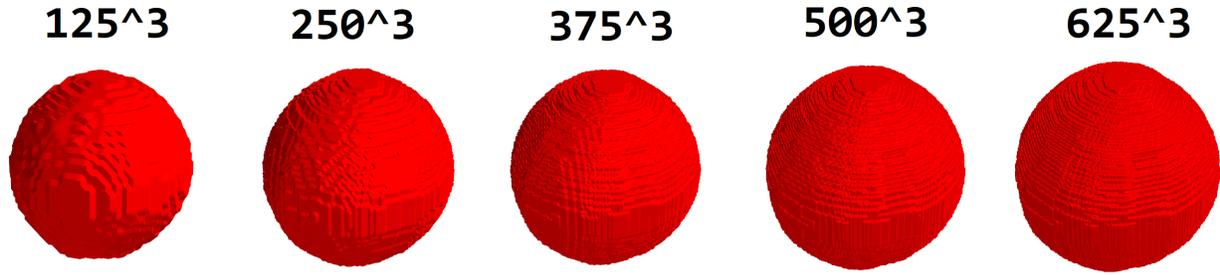


Figure 4.20: Large-scale voxel reconstruction using reference ceramic ball

numbers of voxel sizes from  $125 \times 125 \times 125$  to  $625 \times 625 \times 625$ , which increments of 125 voxels. The run time significantly increases from 10s to 681s. We plot the run time and voxel size relation plot in figure 4.20, the data are shown in table 4.6.

Based on figure 4.20, we clearly find out that the larger the voxel size we select, the more roundish and smooth the reconstruction ball we can achieve from our framework. From table 4.6, we observe that the voxel size and run time have a positive relationship with each other. When we increase the voxel size, the run time complexity significantly increases. For the relationship between voxel size and volume measurement, the situation is really different here. Although the volume slightly goes up, we didn't detect noticeable improvement for the volume measurement by increasing the voxel size. Considering the tradeoff between time-efficient and volume accuracy, we choose voxel size  $125^3$  in our experiment.

Voxel size(X*Y*Z)	Volume(mm <sup>3</sup> )	Run time(s)
125*125*125	33.44	10.08
250*250*250	33.46	42.15
375*375*375	33.46	138.23
500*500*500	33.46	328.07
625*625*625	33.47	681.65

Table 4.6: Volume measurement for reference ceramic ball using different voxel sizes

### Comparison with other methods

We performed volume measurement experiments on wheat samples using the proposed framework. Our collaborator from USDA provided these wheat samples and divided

them into five categories. They are HRW, HRS, SRW, S.WHITE, and DURUM. To compare with other methods, we also introduced the two image volume measurement method from our preliminary work and the glass bead displacement method. In the previous work TT2Cam used a turntable setup with two cameras, then the framework captures two pictures for single seed, and measure the volume using an elliptical cylinder model. We modified TT2Cam by applied a mirrored turntable. In that way, we can only use one camera to capture two orthogonal images at the same time. For the second method, the glass beads displacement experiments are performed by our collaborator research scientists from USDA. We know that extensive researches are using glass beads of 100-200  $\mu m$  as solid displaced media due to their uniform size, compactness, and fluidity properties.

A total of one hundred wheat samples are selected from five categories. For each category, the wheat samples are also divided by its size, small, medium, large, and x-large. We applied the three-volume measurement methods on these wheat samples including the glass beads method, two image methods using the TT2Cam framework, and our proposed framework using  $N = 36$  number of images.

For comparison purposes, the same small group of all the wheat samples has been selected to do statistical analysis. A total of 75 wheat samples are analyzed. For each group, 25 wheat samples have been tested. Before applying the Tukey posthoc comparison test, the normality assumption must be checked on the data. Other than using the visual normality checks like histogram plot or quantile-quantile plot, the D'Agostino's  $K^2$  Test has been introduced to test normality. Distinct group, glass beads method (A), two images method (B), and our proposed method (C) has been tested individually using D'Agostino's  $K^2$  Test to check normality.

For Group A, the test statistic is 3.964 and the p-value is 0.138; for group B, the test statistic is 0.817 and the p-value is 0.665; finally, for group C, the test statistic is 1.784 and the p-value is 0.410. All these results show that we failed to reject the null hypothesis, so we conclude that the data is drawn from a normal distribution. This means that all three groups of data meet the assumption of normality. Another ANOVA assumption

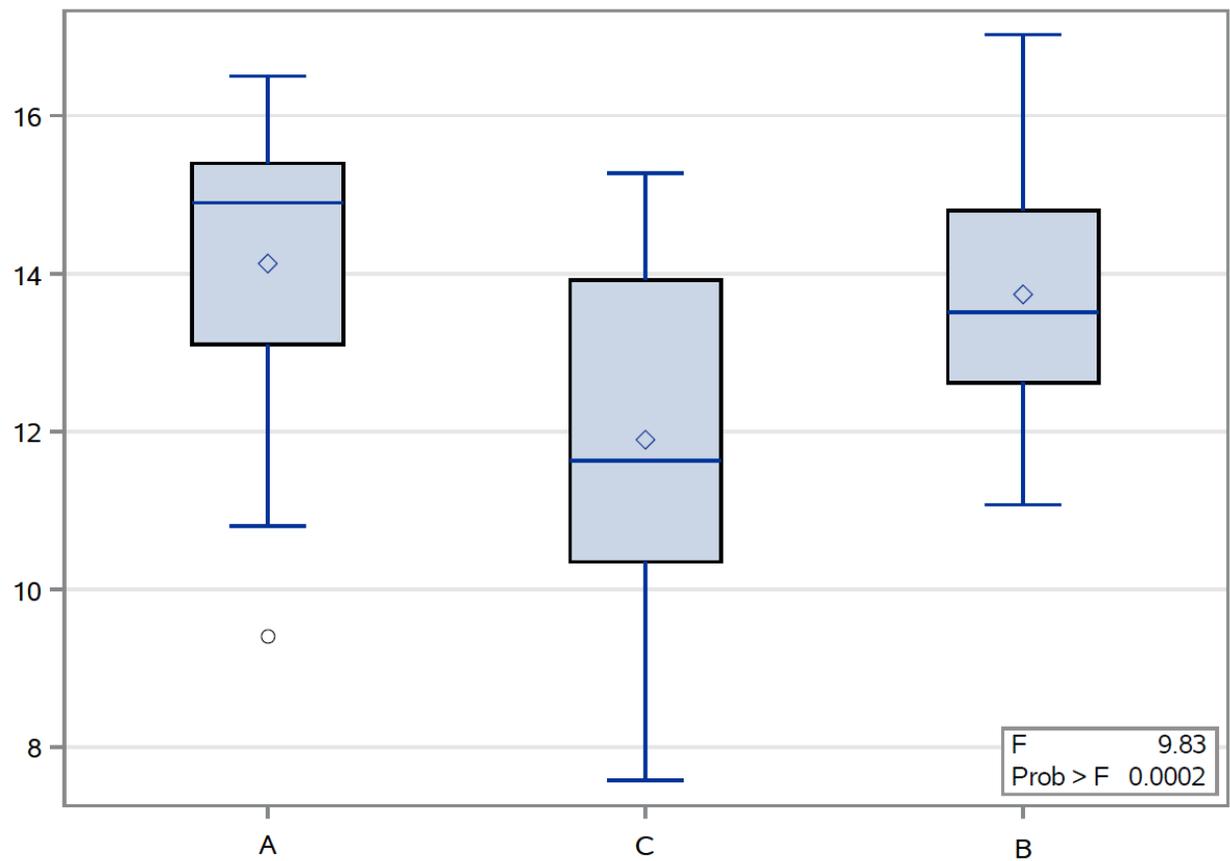


Figure 4.21: Box plot, A: Glass beads method, B: Two images method (TT2Cam), C: Proposed method (N=36)

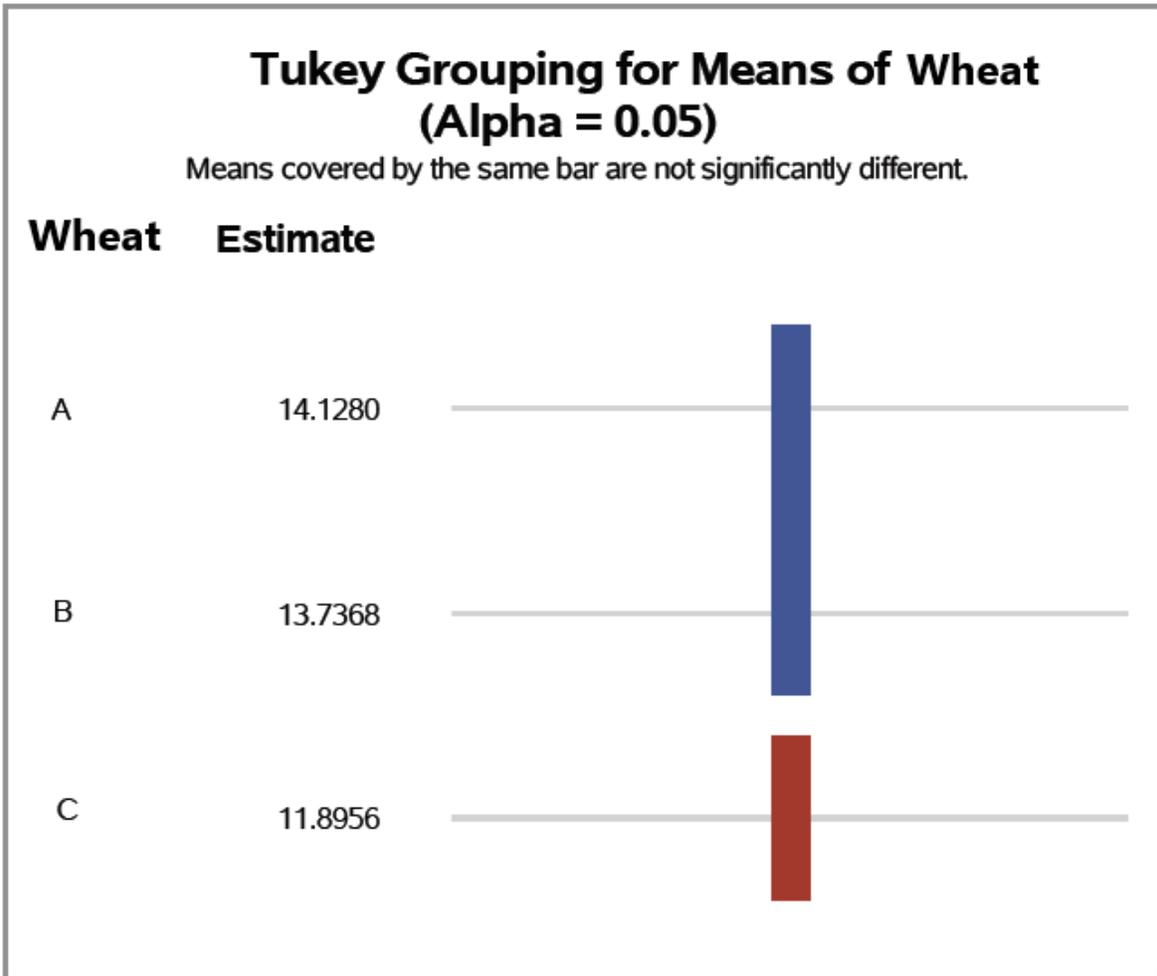


Figure 4.22: Tukey comparison test: A: Glass beads method, B: Two images method (TT2Cam), C: Proposed method (N=36)

that must be checked is the homogeneity of variance. For our case, Bartlett’s test has been used. As the p-value from Bartlett’s test is 0.57, we failed to reject the null hypothesis and conclude that treatments have equal variances. Therefore, both assumptions have been met and we can apply the Tukey comparison test on our data to evaluate which pairs have significantly different treatments.

The box plot and the data analysis from the Tukey comparison test are shown in figure 4.21, 4.22. From the box plot, we observed that the glass beads method(A) and two images method(B) both have a higher average value than our proposed method(C). Our proposed method(C) has a wider range between  $Q_2$  and  $Q_3$ , which means the measurement data is more sparse than the glass beads method(A) and two images method(B). The results from the Tukey comparison test also shows that for all methods A, B, and C, group C is significantly different from group A and group B. But group A and group B have no significant difference between them. As shown in table 4.7, it is clear that our proposed method can achieve a low error rate of less than 3% on the reference ceramic ball. The reference ball has radius  $r = 1.975mm$ , and volume  $32.75mm^3$ . Comparing with the glass beads method (A) and two images method (B), our proposed method (C) performs significantly better with respect to volume measurement accuracy.

We also compare our result with Roussel’s work<sup>36</sup>, they are able to achieve an error rate below 0.1% when using  $N = 36$  images. There are several reasons they are able to achieve better accuracy, they are using an expensive camera and lens setup, and they applied a different carving algorithm since they set a tiny stand to hold the seed and remove the surplus afterward. For our experiment environment, we haven’t put any stand to elevate the seed, it may potentially cause some information loss on the bottom of the seed.

	Average volume( $mm^3$ )	Absolute relative rate(ARE)	Coefficient of Variation(CV)
Sphere1	33.07	2.46%	0.95%
Sphere2	33.52	2.87%	1.25%

Table 4.7: Summary data for reference spheres

## Conclusions and future work

We presented an inexpensive, efficient computer vision system to simplify the image capture and analysis process for single seed volume measurement. The simple volume carving method combined with an inexpensive 3D printed, affordable turntable setup is sufficiently accurate for the volume measurement of most single seeds. To optimize the framework, we are currently testing different numbers of images  $N$  and the resulting performance with respect to efficiency, both in space and time, and accuracy. When  $N = 36$ , for a reference ceramic ball, the proposed framework achieves less than a 3% error rate. For this affordable setup, systematic errors are relatively low.

For future work, other factors that influence the accuracy will be further investigated. For example, the system needs to identify the concave and convex regions of the seed and adjust the volume measurement for concave regions. There is a need to add a more precise stand to hold a seed. Our volume carving algorithm assumes the bottom part of the 3D object is a flat surface due to the lack of information from the bottom. We will improve this by using the top camera and changing the orientation of the single seed with a more sophisticated seed holder. Finally, automating calibration to account for camera distances and orientation will allow small perturbations in camera position and orientation to be automatically accounted for. Overall, the framework provides an end-to-end, efficient computer vision system for accurate seed reconstruction for volume and density measurement.

## 4.3 Computer Vision using Deep Learning

### 4.3.1 Image classification model

The convolutional neural networks (CNNs) has been widely used in the late 1980s. The architecture contains convolutional layers, pooling layers, fully connected layers. In recent years, deep learning models exploit multiple layers of nonlinear information processing<sup>110</sup>. Among them, deep CNNs (DCNNs) were brought into the limelight as a

result of the deep learning renaissance.

The image classification problem is that the task of assigning an input image a label from a fixed set of categories. The image classification has a large variety of practical applications, and many other distinct computer vision tasks including object detection and segmentation can be reduced to image classification.

One fundamental image classifier is K Nearest Neighbor which did not introduce the neural network and conventional neural network. Although the Nearest Neighbor is very simple to implement and the classifier takes no time to train, it can rarely appropriate for use in practical image classification settings and performs poorly when the images are high-dimensional objects.

If we are now going to develop a more powerful approach to image classification, we eventually naturally extend to Neural Networks and Convolutional Neural Networks. The approach will have two major components: a score function that maps the raw data to class scores, and a loss function that quantifies the agreement between the predicted scores and the ground truth labels.

There are several architectures of Convolutional Networks we can choose from including LuNet, AlexNet, ZF Net, GoogleNet, VGGNet, ResNet. In practice, very few people train an entire Convolutional Network from scratch, because it is relatively rare to have a sufficient dataset. Instead, it is common to pretrain a ConvNet on a very large dataset, and then use the ConvNet as a pretrained model for the task of interest.

## **ResNet architectures**

A residual neural network (ResNet) is a framework to ease the training of networks that are substantially deeper than those used previously. The ResNet are easier to optimize and can gain accuracy from considerably increased depth.<sup>111</sup> Instead of setting each stacked layer directly fit an underlying mapping, ResNet lets these layers fit a residual mapping. The ResNet builds the shortcut connections and skips one or more layers see figure 4.23. The formulation of  $F(x) + x$  can be realized by feed-forward neural networks

with these shortcut connections. For the ResNet design, these shortcuts simply perform identity mapping, and their outputs are added to the outputs of the stacked layers. In this way, it will not add any computational complexity or extra parameter.

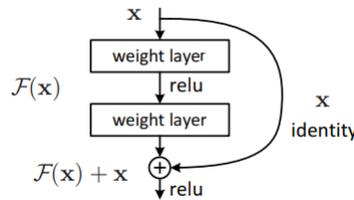


Figure 4.23: Residual learning: a building block

The ResNet also obtain excellent results on the ImageNet Classification dataset and CIFAR-10 dataset. The ResNet152 obtain 3.57% top-5 error rate on the ImageNet test set and won the 1st place in the ILSVRC 2015 classification competition. Therefore, we decide to applied ResNet as the backbone for our rice image classification framework.

The ResNet architectures have been tested using two models for ImageNet. One is plain network, it is inspired by the philosophy of VGG nets. The other is the residual network, it is based on the plain network. The plain network has two simple design rules: 1) for the same output feature map size, the layers have the same number of filters, 2) if the feature map size is halved, the number of filters is doubled so as to preserve the time complexity per layer. The network ends with a global average pooling layer and a 1000-way fully-connected layer with softmax. The total number of layers is 34. For the residual network, the shortcut connections have been inserted into the plain network. The shortcut can be directly used when the input and output are of the same dimensions.

### 4.3.2 Rice image classification using ResNet

Rice has been one of the staple foods that contribute significantly to human food supplies. Numerous rice varieties have been cultivated, imported, and exported worldwide. For our experiment, the rice image data were taken at a different state when the rice grains were in dilute alkali to measure the alkali spreading value. Alkali spreading value of the whole kernel milled rice is the measure of volume expansion in contact with dilute

### 34-layer residual

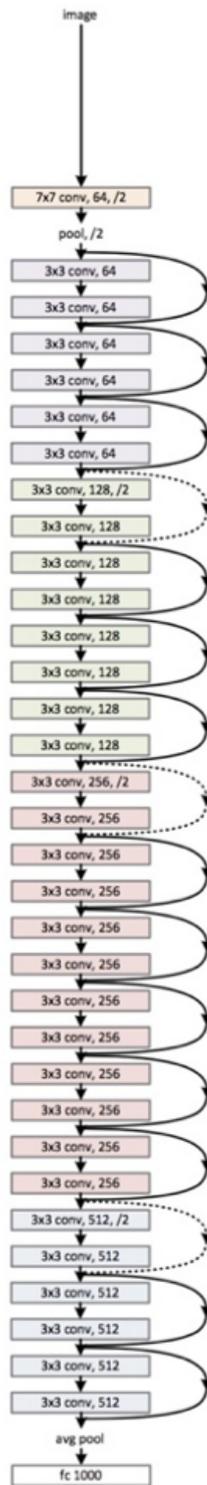


Figure 4.24: ResNet34 architecture

alkali and also a measure of gelatinization temperature. The disintegration of different rice grain in contact with dilute alkali may vary from no apparent effect to a completely dispersed state. The starch gel area of the images of alkali-gelatinized grains was measured after a given gelatinization time. Then, the images are manually labeled by experts from the range 2 to 7. No.2 stands for the initial state, and No.7 stands for the almost completely dispersed state as shown in figure 4.25.

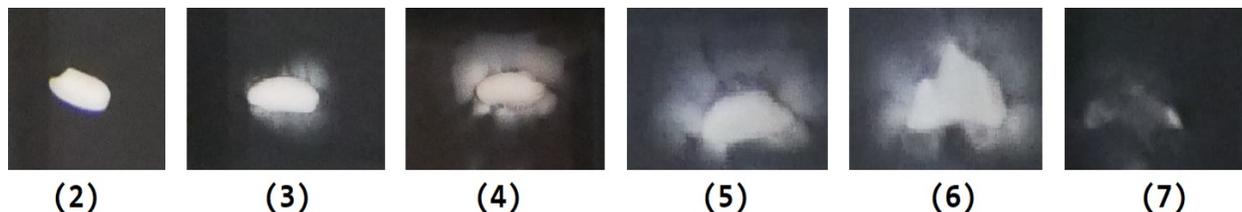


Figure 4.25: Alkali spreading value from 2 to 7

Kansas state university cooperates with USDA, the rice images are provided by the USDA. Each alkali spreading value sample folder contains 70 images, the total amount is 420 images. Therefore, the task is to use these images to train an image classifier to distinct the rice alkali spreading value based on the image input.

## Experiment

For the model training setup, the top 1 error rate is used to evaluate the performance of our classifier. That means the model is considered to have classified a given image correctly if the target label is the model's top prediction. The image data has been divided into three groups: training dataset, validation dataset, and test dataset. The training dataset is the sample of data used to fit the model. The validation dataset is the sample of data used to provide an unbiased evaluation of a model fit on the training dataset while turning the model hyperparameters. The test dataset is the sample of data used to provide an unbiased evaluation of a final model fit on the draining dataset. Therefore, for our dataset, we split 75% images in the training dataset, 12.5% images in the validation dataset, and 12.5 %images in the test dataset.

For the categories label, we divide the alkali spreading value from 2 to 7 into 3 groups (alkali spreading value small(2,3), medium(4,5), and high(6,7)). The training steps are as follow:

1. For each epoch, it has two-phase which are the training and validation phase.
2. Then, the training method literately processes the inputs and labels from the dataset.
3. Next, parameter gradients are set to zero initially.
4. During the training, the model tracks the training history and optimize the hyper-parameter if it is in the training phase.
5. For each epoch, the model calculates the loss and accuracy for the result.

If the image classifier is trained based on the limited amount of the dataset, the transfer learning is the practical initialization for the task. ResNet is used as a pretrained model. The pretrained model has been trained on ImageNet, which contains 1.2 million images with 1000 categories.

The pre-setup of transfer learning before the training step is as follow

1. First, the pretrained model ResNet34 is loaded for transfer learning.
2. Then, the fully connected layer is reset based on the output sample size.
3. Next, the cross-entropy loss is calculated for evaluating the performance of our classification model.
4. The model optimizes the parameters using stochastic gradient descent by updating weights after each epoch is calculated.
5. The model adjust the learning rate based on the number of epochs.

## Results

ResNet pretrained model is applied on ImageNet, by removing the last fully-connected layer, then the rest of the ResNet is treated as a fixed feature extractor for the new rice image dataset. For the ResNet architecture, the distinct ResNet architecture has been testing including ResNet18, ResNet34, ResNet101 and ResNet152. As shown in figure 4.8, training experiments using different ResNet architectures. ResNet34 architecture is shown in figure 4.24.

	ResNet18	ResNet34	ResNet152
Accuracy Overall :	85%	78%	78%
Accuracy of ASV_L :	95%	65%	100%
Accuracy of ASV_M :	80%	80%	95%
Accuracy of ASV_H :	80%	90%	40%

Table 4.8: Model training top one accuracy using different ResNet architectures

For experiment results, the alkali spreading value has been divided into three groups (alkali spreading value small(2,3), medium(4,5), and high(6,7)). For all these training experiments, a range of epochs has been conducted. Due to the performance result, 150 epochs are selected. For the ResNet18, the accuracy is around 85% overall. For the ResNet34, the accuracy is around 78% overall. For the ResNet152, the accuracy is around 78% overall.

## 4.4 Conclusion

For the two image turntable framework, the experiment is conducted by feeding the seed samples using the vibrating feeder, with adjustable speed controlled by the electric controller. While the turntable is running, the seed is aligned perfectly by the white plastic obstacles placed on the side and middle of the turntable. When the seed dynamically goes through the front region, the top color camera and the side monochrome camera will capture the images simultaneously. Our framework provides a user-friendly interface, a windows application TT2Cam for breeders or agriculture researchers to use and

conduct image analysis. The volume measurement algorithm designed in Python will be integrated into the original C# framework, researchers can easily capture two pictures and get the measurement result of a single seed from image analysis conveniently by the C# application.

For the multiple-image turntable framework, we propose an inexpensive, efficient computer vision system to simplify the image capture and analysis process for single seed volume measurement. The simple volume carving method combined with our 3D printed, affordable turntable setup is sufficiently accurate for the volume measurement of most single seeds. To optimize our framework, we are testing different numbers of images  $N$  and the resulting performance with respect to efficiency, both in space and time, and accuracy. When we select  $N = 36$  in our reference ceramic ball, the proposed framework achieves less than a 3% error rate. For this affordable setup, systematic errors are relatively lower than we expect. For future work, other factors that influence the accuracy will be further investigated. For example, the concave and convex region of the seed, improving the camera calibration method, adding a stand to hold a seed. Also, our volume carving algorithm assumes the bottom part of the 3D object is the flat surface due to the lack of information from the bottom. We will improve it by using the top camera or change the orientation of the single seed. Overall, we conclude that our proposed framework provides an end-to-end, efficient computer vision system for accurate seed reconstruction for volume and density measurement.

For the deep learning classification framework, we propose a rapid, convenient rice image classification application to classify the rice dispersion images using alkali spreading value. The neural network model has been trained using 420 rice dispersion images in the range of 2 to 7 of alkali spreading values. For the raw image data, the alkali spreading value is manually labeled by the agriculture experts. The whole dataset is divided by 75% images in the training dataset, 12.5% images in the validation dataset, and 12.5% images in the test dataset. For this proposed application, we are able to achieve 85% accuracy on the 3 categories (alkali spreading value small(2,3), medium(4,5), and high(6,7)).

# Chapter 5

## Conclusions and future work

WinDAM is being developed in stages to evaluate the performance of earth dams. Existing modules with well-defined interfaces enable efficient integration of existing legacy software with innovations. The system provides tools that can be used to better understand the structure, function, and dynamics of water control structures. Our work in Chapter 2 describes how WinDAM and BREACH models can be analyzed efficiently using a novel new risk assessment framework that uses Dakota to perform parameter studies, sensitivity analysis, and uncertainty quantification. The next step in Chapter 3 shows the coupling of CFD flow models with physical models to model the erosion that results from a given flow. Our CFD coupling framework shows promising results when modeling the internal and overtopping erosion process in accurately modeling the total and peak discharges that may result from dam erosion or failure.

After further investigation, to deal with the topological change of the dam structure, the material point method (MPM) is introduced for the dam breach simulation framework in Chapter 3. Due to the limitation of the mesh constructed by the CFD coupling framework, we simulate the erosion processes that cause dam deformation. A new model is developed and implemented using the material point method. It is shown to be a good solution to model both flow and dam structure deformation because it incorporates both Lagrangian and Eulerian methods. Lagrangian methods employ a framework in which

space is discretized into initial sub-volumes, whose flow paths are then charted over time. Eulerian methods, on the other hand, employ a framework in which the motion of material is described relative to a mesh that remains fixed in space throughout the calculation. MPM methods combine Lagrangian and Eulerian frames of reference allowing us to more accurately model the interaction at the liquid-solid interface.

Therefore, we implemented our dam erosion framework using the MPM method. Based on the simulation results, our MPM method can provide extensive details during the erosion process on the downstream face of a dam. We also compare our simulation results with an existing physical-based model such as WinDAM. For the current implementation, the model we use for the momentum exchange and energy density function is a simplified version and this can be improved upon. Also, the simulation time for the MPM method is limited to 60 seconds, but the simulation time for a typical WinDAM C run is around 16-24 hours. The obvious next step is to analyze a run over a full simulation interval. For the current stage, our MPM model only simulates the process in a shortened time. Overall, our MPM simulation framework provides rich details during the dam breach process. It can capture the water and soil interaction and gradually remove the soil during the breach process when simulating using the existing dam breach example.

For future study, we will further investigate the implementation method, we will design the simulation model on a larger scale, and also implement a complete version of the momentum exchange and the energy density function. Also, the algorithm can be optimized to reduce run-time complexity. In particular, we want to achieve better experimental accuracy and reduce the run-time as well.

In Chapter 4.1, we provide a user-friendly Windows application TT2Cam for agriculture researchers to use and conduct image analysis. With the volume measurement algorithm designed in Python, researchers can easily capture two pictures and get the measurement result of a single seed from image analysis conveniently. The algorithm provides fast, novel, and universal image processing solutions for seed volume measurement. Based on the main view image captured by a color camera, side view image cap-

tured by a monochrome camera, the algorithm can detect the contour of the single seed, automatically slice the whole seed into equal height rectangle pieces, apply the scale factor for calibration purpose, and finally estimate the volume of a single seed.

For the image processing part, we only use two camera settings to gather pixel information. For the camera setting and orientation, our framework has one color camera and one monochrome camera. The reason we use one monochrome camera is that the image sourcing monochrome camera has NIR wavelength data collection. The number of cameras can be extended to three or more to gain more information from various angles.

In the future, the advanced color analyses including extracting NIR wavelength data will be further investigated. Other options to improve the accuracy are color adjustment and light adjustment, these methods can be added to our TT2Cam frameworks. Seed layout also plays a significant impact on the accuracy of seed volume measurement. Also, the next step is to add trigger function in real-time, color information extraction, and seed density measurement to our TT2Cam framework. The goal of this framework is to complete the volume measurement in real-time after two images are captured when a seed passes a fixed point on the turntable.

The completed framework in Chapter 4.2 provides an inexpensive, efficient computer vision system to simplify the image capture and analysis process for single seed volume measurement from any number of images. The simple volume carving method combined with our 3D printed, affordable turntable setup is sufficiently accurate for the volume measurement of most single seeds. To optimize our framework, we are testing different numbers of images  $N$  and the resulting performance with respect to efficiency, both in space and time, and accuracy. When we select  $N = 36$  in our reference ceramic ball, the proposed framework achieves less than a 3% error rate. For this affordable setup, systematic errors are relatively lower than we expect. Overall, we conclude that our proposed framework provides an end-to-end, efficient computer vision system for accurate seed reconstruction for volume and density measurement.

For future work, other factors like the concave and convex regions of the seed which influence the accuracy will be further explored. Also, we can change the camera cali-

bration method, or add a stand to hold a seed to improve the testing results. Since our volume carving algorithm assumes the bottom part of the 3D object is the flat surface due to the lack of information from the bottom, the algorithm can be improved by using the top camera or change the orientation of the single seed.

# Bibliography

- [1] Hatice Ozmen-Cagatay and Selahattin Kocaman. Dam-break flow in the presence of obstacle: Experiment and CFD simulation. *Engineering Applications of Computational Fluid Mechanics*, 5(4):541–552, 2011. ISSN 1997003X. doi: 10.1080/19942060.2011.11015393.
- [2] C. Biscarini, S. Di Francesco, and P. Manciola. CFD modelling approach for dam break flow studies. *Hydrology and Earth System Sciences*, 14(4):705–718, 2010. ISSN 16077938. doi: 10.5194/hess-14-705-2010.
- [3] S. Sina Hosseini Boosari. Predicting the Dynamic Parameters of Multiphase Flow in CFD (dam-break simulation) using artificial intelligence-(cascading deployment). *Fluids*, 4(1), 2019. ISSN 23115521. doi: 10.3390/fluids4010044.
- [4] M.L.Neilsen. Computational fluid dynamics models for WinDAM. In *Proceedings of the 30th International Conference on Computers and their Applications*, 2015.
- [5] Chendi Cao and Mitchell L Neilsen. Flexible Architecture for Analysis of Water Control Structures. *International Conference On Computer Applications In Industry And Engineering*, 2018.
- [6] Matthieu De Lefte, David Le Touzé, and Bertrand Alessandrini. SPH modeling of shallow-water coastal flows. *Journal of Hydraulic Research*, 48(SUPPL. 1):118–125, 2010. ISSN 00221686. doi: 10.1080/00221686.2010.9641252.
- [7] Kai Gong, Songdong Shao, Hua Liu, Benlong Wang, and Soon Keat Tan. Two-phase SPH simulation of fluid-structure interactions. *Journal of Fluids and Structures*, 65: 155–179, 2016. ISSN 10958622. doi: 10.1016/j.jfluidstructs.2016.05.012. URL <http://dx.doi.org/10.1016/j.jfluidstructs.2016.05.012>.

- [8] Mahmoud Reza Mollaie Nia, Rasool Ghobadian, Satar Ahmadyan, and Hamzeh Ebrahimnejadian. Simulation of Dam Break Using Modified Incompressible Smoothed Particles Hydrodynamics (MPM-I-SPH). 5(1):1051–1057, 2014.
- [9] Jaan Hui Pu, Songdong Shao, Yuefei Huang, and Khalid Hussain. Evaluations of swes and sph numerical modelling techniques for dam break flows. *Engineering Applications of Computational Fluid Mechanics*, 7(4):544–563, 2013. ISSN 1997003X. doi: 10.1080/19942060.2013.11015492.
- [10] Andreia Moreira, Agnès Leroy, Damien Violeau, and Francisco Taveira-Pinto. Dam spillways and the SPH method: two case studies in Portugal. *Journal of Applied Water Engineering and Research*, 9676(May), 2019. ISSN 23249676. doi: 10.1080/23249676.2019.1611496.
- [11] Jidong Zhao and Tong Shan. Coupled CFD-DEM simulation of fluid-particle interaction in geomechanics. *Powder Technology*, 239:248–258, 2013. ISSN 00325910. doi: 10.1016/j.powtec.2013.02.003. URL <http://dx.doi.org/10.1016/j.powtec.2013.02.003>.
- [12] Zhen Ming Shi, Hong Chao Zheng, Song Bo Yu, Ming Peng, and Tao Jiang. Application of CFD-DEM to investigate seepage characteristics of landslide dam materials. *Computers and Geotechnics*, 101(April):23–33, 2018. ISSN 18737633. doi: 10.1016/j.compgeo.2018.04.020. URL <https://doi.org/10.1016/j.compgeo.2018.04.020>.
- [13] T. Zhao, S. Utili, and G. B. Crosta. Rockslide and Impulse Wave Modelling in the Vajont Reservoir by DEM-CFD Analyses. *Rock Mechanics and Rock Engineering*, 49(6):2437–2456, 2016. ISSN 07232632. doi: 10.1007/s00603-015-0731-0. URL <http://dx.doi.org/10.1007/s00603-015-0731-0>.
- [14] Tong Shan and Jidong Zhao. A coupled CFD-DEM analysis of granular flow im-

- pacting on a water reservoir. *Acta Mechanica*, 225(8):2449–2470, 2014. ISSN 00015970. doi: 10.1007/s00707-014-1119-z.
- [15] Xingyue Li and Jidong Zhao. Numerical simulation of dam break by a coupled CFD-DEM approach. *15th Asian Regional Conference on Soil Mechanics and Geotechnical Engineering, ARC 2015: New Innovations and Sustainability*, (January):691–696, 2015. ISSN 2188-8027. doi: 10.3208/jgssp.TC103-03.
- [16] Kyung Min Park, Hyun Sik Yoon, and Min Il Kim. CFD-DEM based numerical simulation of liquid-gas-particle mixture flow in dam break. *Communications in Nonlinear Science and Numerical Simulation*, 59:105–121, 2018. ISSN 10075704. doi: 10.1016/j.cnsns.2017.11.010.
- [17] Witawat Rungjiratananon, Zoltan Szego, Yoshihiro Kanamori, and Tomoyuki Nishita. Real-time Animation of Sand-Water Interaction. 27(7), 2008.
- [18] Ke Wu, Dongmin Yang, and Nigel Wright. A coupled SPH-DEM model for fluid-structure interaction problems with free-surface flow and structural failure. *Computers and Structures*, 177:141–161, 2016. ISSN 00457949. doi: 10.1016/j.compstruc.2016.08.012.
- [19] Ricardo Canelas. A generalized SPH-DEM discretization for the modelling of complex multiphase free surface flows. *SPHERIC workshop*, (June):1–105, 2013. ISSN 0870-2551 (Print) 0870-2551. doi: 10.13140/RG.2.1.1283.2407.
- [20] Toon Lenaerts and Philip Dutré. Mixing Fluids and Granular Materials. 28(2), 2009.
- [21] Francesca Ceccato and Alba Yerro. Modelling soil-water interaction with the Material Point Method . Evaluation of single-point and double-point formulations. (June), 2018.

- [22] Samila Bandara, Alessio Ferrari, and Lyesse Laloui. Modelling landslides in unsaturated slopes subjected to rainfall infiltration using material point method. (January):1358–1380, 2016. doi: 10.1002/nag.
- [23] Samila Bandara and Kenichi Soga. Coupling of soil deformation and pore fluid flow using material point method. *COMPUTERS AND GEOTECHNICS*, 63(January 2015):199–214, 2016. ISSN 0266-352X. doi: 10.1016/j.compgeo.2014.09.009. URL <http://dx.doi.org/10.1016/j.compgeo.2014.09.009>.
- [24] Andre Pradhana Tampubolon and Los Angeles. Multi-species simulation of porous sand and water mixtures. 36(4), 2017.
- [25] Gergely Kl and Theodore Gast. Drucker-Prager Elastoplasticity for Sand Animation. 2016.
- [26] P Arduino, W Shin, J A Moore, and G R Miller. Modeling strategies for multiphase drag interactions using the material point method. (February):295–322, 2010. doi: 10.1002/nme.
- [27] Issam Jassim, Dieter Stolle, and Pieter Vermeer. Two-phase dynamic analysis by material point method. (October 2012):2502–2522, 2013. doi: 10.1002/nag.
- [28] Candace R. Moore, David S. Gronwall, Nathan D. Miller, and Edgar P. Spalding. Mapping quantitative trait loci affecting arabidopsis thaliana seed morphology features extracted computationally from images. *G3: Genes, Genomes, Genetics*, 3(1): 109–118, 2013. ISSN 21601836. doi: 10.1534/g3.112.003806.
- [29] Alex P. Whan, Alison B. Smith, Colin R. Cavanagh, Jean Philippe F. Ral, Lindsay M. Shaw, Crispin A. Howitt, and Leanne Bischof. GrainScan: A low cost, fast method for grain size and colour measurements. *Plant Methods*, 10(1):1–10, 2014. ISSN 17464811. doi: 10.1186/1746-4811-10-23.
- [30] Rowan P. Herridge, Robert C. Day, Samantha Baldwin, and Richard C. Macknight.

- Rapid analysis of seed size in Arabidopsis for mutant and QTL discovery. *Plant Methods*, 7(1):1–11, 2011. ISSN 17464811. doi: 10.1186/1746-4811-7-3.
- [31] Winseedle. URL [https://regentinstruments.com/assets/winseedle\\_about.html](https://regentinstruments.com/assets/winseedle_about.html).
- [32] Seedcount. URL <https://www.nextinstruments.net/products/seedcount>.
- [33] Mitchell L. Neilsen, Darrel M. Temple, and Gregory J. Hanson. WinDAM: Earthen embankment erosion analysis. In *Proceedings of the 24th International Conference on Computers and their Applications in Industry and Engineering*, 2011.
- [34] Mitchell L. Neilsen and Chendi Cao. Coupled dam safety analysis using BREACH and WinDAM. In *Proceedings of the 15th International Conference on Scientific Computing*, pages 82–85, 2017.
- [35] Takanari Tanabata, Taeko Shibaya, Kiyosumi Hori, Kaworu Ebana, and Masahiro Yano. SmartGrain: High-throughput phenotyping software for measuring seed shape through image analysis. *Plant Physiology*, 160(4):1871–1880, 2012. ISSN 00320889. doi: 10.1104/pp.112.205120.
- [36] Johanna Roussel, Felix Geiger, Andreas Fischbach, Siegfried Jahnke, and Hanno Scharr. 3D surface reconstruction of plant seeds by volume carving: Performance and accuracies. *Frontiers in Plant Science*, 7(June2016):1–13, 2016. ISSN 1664462X. doi: 10.3389/fpls.2016.00745.
- [37] Ebru Firatligil-Durmuş, Evžen Šárka, Zdeněk Bubník, Matyáš Schejbal, and Pavel Kadlec. Size properties of legume seeds of different varieties using image analysis. *Journal of Food Engineering*, 99(4):445–451, 2010. ISSN 02608774. doi: 10.1016/j.jfoodeng.2009.08.005.
- [38] Seyed M.A. Razavi, A. Bostan, and M. Rezaie. Image processing and physico-mechanical properties of basil seed (*ocimum basilicum*). *Journal of Food Process*

- Engineering*, 33(1):51–64, 2010. ISSN 01458876. doi: 10.1111/j.1745-4530.2008.00259.x.
- [39] Seyed M.A. Razavi, A. Bostan, and R. Rahbari. Computer image analysis and physico-mechanical properties of wild sage seed (*Salvia macrosiphon*). *International Journal of Food Properties*, 13(2):308–316, 2010. ISSN 10942912. doi: 10.1080/10942910802398453.
- [40] C. M. Sabliov, D. Boldor, K. M. Keener, and B. E. Farkas. Image processing method to determine surface area and volume of axi-symmetric agricultural products. *International Journal of Food Properties*, 5(3):641–653, 2002. ISSN 10942912. doi: 10.1081/JFP-120015498.
- [41] Emilio Cervantes, José Javier Martín, and Ezzeddine Saadaoui. Updated Methods for Seed Shape Analysis. *Scientifica*, 2016:1–10, 2016. doi: 10.1155/2016/5691825.
- [42] J. F. Pedersen, C. R. Martin, F. C. Felker, and J. L. Steele. Application of the single kernel wheat characterization technology to sorghum grain. *Cereal Chemistry*, 73(4):421–423, 1996. ISSN 00090352.
- [43] Mohammad Reza Amiryousefi, Mohebbat Mohebbi, and Ali Tehranifar. Pomegranate seed clustering by machine vision. *Food Science and Nutrition*, 6(1): 18–26, 2018. ISSN 20487177. doi: 10.1002/fsn3.475.
- [44] Ebru Firatligil-Durmuş, Evžen Šárka, and Zdeněk Bubník. Image vision technology for the characterisation of shape and geometrical properties of two varieties of lentil grown in Turkey. *Czech Journal of Food Sciences*, 26(2):109–116, 2008. ISSN 12121800. doi: 10.17221/1/2008-cjfs.
- [45] Worthy N. Martin, Worthy N. Martin, and J. K. Aggarwal. Volumetric Descriptions of Objects from Multiple Views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-5(2):150–158, 1983. ISSN 01628828. doi: 10.1109/TPAMI.1983.4767367.

- [46] Aldo Laurentini. The Visual Hull Concept for Silhouette-Based Image Understanding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 16(2):150–162, 1994. ISSN 01628828. doi: 10.1109/34.273735.
- [47] Michael Potmesil. Generating octree models of 3D objects from their silhouettes in a sequence of images. *Computer Vision, Graphics and Image Processing*, 40(1):1–29, 1987. ISSN 0734189X. doi: 10.1016/0734-189X(87)90053-3.
- [48] Chendi Cao and Mitchell L. Neilsen. Flexible architecture for analysis of water control structures. *31st International Conference on Computer Applications in Industry and Engineering, CAINE 2018*, pages 83–88, 2018.
- [49] Chendi Cao and Mitchell L. Neilsen. Efficient seed volume measurement framework. In *Proceedings of the 2020 International Conference on Computational Science and Computational Intelligence*, 2020.
- [50] D.M Temple, G.J. Hanson, and M.L. Neilsen. WinDAM – Analysis of overtopped earth embankment dams. In *Proceedings of the ASABE Annual Conference*, page 062105, 2006.
- [51] D L Fread. BREACH: An erosion model for earthen dam failures, July 1988, revision 1, August 1991. NOAA, 1991.
- [52] Weiming Wu. Simplified physically based model of earthen embankment breaching. *Journal of Hydraulic Engineering*, 139(8):837–851, 2013. ISSN 07339429. doi: 10.1061/(ASCE)HY.1943-7900.0000741.
- [53] Brian M Adams, Keith R Dalbey, Michael S Eldred, David M Gay, Laura P Swiler, William J Bohnhoff, John P Eddy, and Patricia D Hough. DAKOTA, A Multilevel Parallel Object Oriented Framework for Design Optimization, Parameter Estimation, Uncertainty Quantification, and Sensitivity Analysis. *Manual*, (December 2009):User Manual, 2010.

- [54] Mitchell L Neilsen. Global Sensitivity Analysis of Dam Erosion Models. (June), 2013.
- [55] H. G. Weller, G. Tabor, H. Jasak, and C. Fureby. A tensorial approach to computational continuum mechanics using object-oriented techniques. *Computers in Physics*, 12(6):620, 1998. ISSN 08941866. doi: 10.1063/1.168744.
- [56] Beocat: Research computing at kansas state university. URL <https://beocat.ksu.edu/>.
- [57] Tao Zhao, Feng Dai, and Nu wen Xu. Coupled DEM-CFD investigation on the formation of landslide dams in narrow rivers. *Landslides*, 14(1):189–201, 2017. ISSN 16125118. doi: 10.1007/s10346-015-0675-1.
- [58] D Sulsky. A particle method for history-dependent materials. *Computer Methods in Applied Mechanics and Engineering*, 118(1-2):179–196, 1994. ISSN 00457825. doi: 10.1016/0045-7825(94)00033-6.
- [59] K. Soga, E. Alonso, A. Yerro, K. Kumar, S. Bandara, J. S.H. Kwan, R. C.H. Koo, R. P.H. Law, J. Yiu, E. H.Y. Sze, and K. K.S. Ho. Trends in large-deformation analysis of landslide mass movements with particular emphasis on the material point method. *Geotechnique*, 68(5):457–458, 2018. ISSN 17517656. doi: 10.1680/jgeot.16.D.004.
- [60] Antonello Troncone, Enrico Conte, and Luigi Pugliese. Analysis of the slope response to an increase in pore water pressure using the material point method. *Water (Switzerland)*, 11(7):1–13, 2019. ISSN 20734441. doi: 10.3390/w11071446.
- [61] Antonello Troncone, Luigi Pugliese, and Enrico Conte. Run-out simulation of a landslide triggered by an increase in the groundwater level using the material point method. *Water (Switzerland)*, 12(10), 2020. ISSN 20734441. doi: 10.3390/w12102817.

- [62] A. Yerro, E. E. Alonso, and N. M. Pinyol. The material point method for unsaturated soils. *Geotechnique*, 65(3):201–217, 2015. ISSN 17517656. doi: 10.1680/geot.14.P.163.
- [63] A. Yerro, E. E. Alonso, and N. M. Pinyol. Run-out of landslides in brittle soils. *Computers and Geotechnics*, 80:427–439, 2016. ISSN 18737633. doi: 10.1016/j.compgeo.2016.03.001. URL <http://dx.doi.org/10.1016/j.compgeo.2016.03.001>.
- [64] Alba Yerro, Kenichi Soga, and Jonathan Bray. Runout evaluation of oso landslide with the material point method. *Canadian Geotechnical Journal*, 56(9):1304–1317, 2019. ISSN 12086010. doi: 10.1139/cgj-2017-0630.
- [65] Enrico Conte, Luigi Pugliese, and Antonello Troncone. Post-failure stage simulation of a landslide using the material point method. *Engineering Geology*, 253 (November 2018):149–159, 2019. ISSN 00137952. doi: 10.1016/j.enggeo.2019.03.006. URL <https://doi.org/10.1016/j.enggeo.2019.03.006>.
- [66] Enrico Conte, Luigi Pugliese, and Antonello Troncone. Post-failure analysis of the Maierato landslide using the material point method. *Engineering Geology*, 277 (July):105788, 2020. ISSN 00137952. doi: 10.1016/j.enggeo.2020.105788. URL <https://doi.org/10.1016/j.enggeo.2020.105788>.
- [67] Quoc Anh Tran and Wojciech Sołowski. Generalized Interpolation Material Point Method modelling of large deformation problems including strain-rate effects – Application to penetration and progressive failure problems. *Computers and Geotechnics*, 106(April 2018):249–265, 2019. ISSN 18737633. doi: 10.1016/j.compgeo.2018.10.020. URL <https://doi.org/10.1016/j.compgeo.2018.10.020>.
- [68] Youkou Dong, Dong Wang, and Mark F. Randolph. Investigation of impact forces on pipeline by submarine landslide using material point method. *Ocean Engineer-*

- ing, 146(October 2016):21–28, 2017. ISSN 00298018. doi: 10.1016/j.oceaneng.2017.09.008. URL <https://doi.org/10.1016/j.oceaneng.2017.09.008>.
- [69] Y. Dong, D. Wang, and M. F. Randolph. Quantification of impact forces on fixed mudmats from submarine landslides using the material point method. *Applied Ocean Research*, 102(December 2019):102227, 2020. ISSN 01411187. doi: 10.1016/j.apor.2020.102227. URL <https://doi.org/10.1016/j.apor.2020.102227>.
- [70] Mitchell L Neilsen and Nichols Hall. WinDAM : Earthen Embankment Erosion Analysis WinDAM : Earthen Embankment Erosion Analysis. (May), 2014.
- [71] By R J Atktn, Applied Mathematics, and Se Computing Science. CONTINUUM THEORIES OF MIXTURES : BASIC THEORY AND HISTORICAL. XXIX(2), 1975.
- [72] Ronaldo I Borja. On the mechanical energy and effective stress in saturated and unsaturated porous continua q. 43:1764–1786, 2006. doi: 10.1016/j.ijsolstr.2005.04.045.
- [73] Javier Bonet and Richard D. Wood. *Nonlinear Continuum Mechanics for Finite Element Analysis*. Cambridge University Press, 2 edition, 2008. doi: 10.1017/CBO9780511755446.
- [74] D C Drucker and W Prager. SOIL MECHANICS AND PLASTIC ANALYSIS OR LIMIT DESIGN \* c ' i = X df / daa ., pages 157–165, 1943.
- [75] Markus Becker. Weakly compressible SPH for free surface flows. pages 1–8, 2007.
- [76] K.Soga D.Robert. Soil-Pipeline Interaction in Unsaturated Soils.
- [77] Samila Bandara and Kenichi Soga. Computers and Geotechnics Coupling of soil deformation and pore fluid flow using material point method. *COMPUTERS AND GEOTECHNICS*, 63:199–214, 2015. ISSN 0266-352X. doi: 10.1016/j.compgeo.2014.09.009. URL <http://dx.doi.org/10.1016/j.compgeo.2014.09.009>.

- [78] Chenfanfu Jiang, Andrew Selle, and Joseph Teran. The Affine Particle-In-Cell Method. 2015.
- [79] Alexey Stomakhin, Craig Schroeder, Lawrence Chai, Joseph Teran, and Andrew Selle. A material point method for snow simulation. 2013.
- [80] Yiming Xia. Flame. *GitHub repository*, 2018.
- [81] G J Hanson, K M Robinson, and K R Cook. Prediction of Headcut Migration Using a Deterministic Approach. 44(3):525–531, 2001.
- [82] G J Hanson, K R Cook, and S L Hunt. Physical Modeling of Overtopping Erosion and Breach Formation of Cohesive Embankments. 48(5):1783–1794, 2005.
- [83] Etienne Belin, Clément Douarre, Nicolas Gillard, Florence Franconi, Julio Rojas-Varela, François Chapeau-Blondeau, Didier Demilly, Jérôme Adrien, Eric Maire, and David Rousseau. Evaluation of 3D/2D imaging and image processing techniques for the monitoring of seed imbibition. *Journal of Imaging*, 4(7):1–16, 2018. ISSN 2313433X. doi: 10.3390/jimaging4070083.
- [84] Paween Khoenkaw. An Image-processing based algorithm for rice seed germination rate evaluation. *20th International Computer Science and Engineering Conference: Smart Ubiquitous Computing and Knowledge, ICSEC 2016*, pages 2–6, 2017. doi: 10.1109/ICSEC.2016.7859890.
- [85] Uroš Škrubej, Črtomir Rozman, and Denis Stajnko. The accuracy of the germination rate of seeds based on image processing and artificial neural networks. *Agricultura*, 12(1-2):19–24, 2016. doi: 10.1515/agricultura-2016-0003.
- [86] Pratiksha Vyas, Suchita Yerkar, and Pallavi Harde. Germination of Rice Seed Using Image Processing and Matlab. *International Journal of Electronics, Communication & Soft Computing Science and Engineering*, pages 1–6, 2017.

- [87] R. Abalone, A. Cassinera, A. Gastón, and M. A. Lara. Some physical properties of amaranth seeds. *Biosystems Engineering*, 89(1):109–117, 2004. ISSN 15375110. doi: 10.1016/j.biosystemseng.2004.06.012.
- [88] Wiesnerová Dana and Wiesner Ivo. Computer image analysis of seed shape and seed color for flax cultivar description. *Computers and Electronics in Agriculture*, 61(2):126–135, 2008. ISSN 01681699. doi: 10.1016/j.compag.2007.10.001.
- [89] Ashabahebwa Ambrose, Lalit Mohan Kandpal, Moon S. Kim, Wang Hee Lee, and Byoung Kwan Cho. High speed measurement of corn seed viability using hyperspectral imaging. *Infrared Physics and Technology*, 75:173–179, 2016. ISSN 13504495. doi: 10.1016/j.infrared.2015.12.008. URL <http://dx.doi.org/10.1016/j.infrared.2015.12.008>.
- [90] Insuck Baek, Dewi Kusumaningrum, Lalit Mohan Kandpal, Santosh Lohumi, Changyeun Mo, Moon S. Kim, and Byoung Kwan Cho. Rapid measurement of soybean seed viability using Kernel-based multispectral image analysis. *Sensors (Switzerland)*, 19(2):1–16, 2019. ISSN 14248220. doi: 10.3390/s19020271.
- [91] Rashidah Ruslan, Muhammad Firdaus Ibrahim, Siti Khairunniza-Bejo, Aimi Athirah Aznan, Ibni Hajar Rukunudin, and Fathin Ayuni Azizan. Effect of Background Color on Rice Seed Image Segmentation Using Machine Vision. *2018 International Conference on Computational Approach in Smart Systems Design and Applications, ICASSDA 2018*, pages 1–4, 2018. doi: 10.1109/ICASSDA.2018.8477614.
- [92] S. Durai, C. Mahesh, T. Sujithra, and A. Suresh. Survey of rice seed quality analysis for varietal purity estimation by using image processing techniques. *International Journal of Engineering and Technology(UAE)*, 7(1.7 Special Issue 7):34–37, 2018. ISSN 2227524X. doi: 10.14419/ijet.v7i1.7.9383.
- [93] Min Huang, Chujie He, Qibing Zhu, and Jianwei Qin. Maize seed variety classification using the integration of spectral and image features combined with feature

- transformation based on hyperspectral imaging. *Applied Sciences (Switzerland)*, 6 (6), 2016. ISSN 20763417. doi: 10.3390/app6060183.
- [94] Xiaoling Yang, Hanmei Hong, Zhaohong You, and Fang Cheng. Spectral and image integrated analysis of hyperspectral data for waxy corn seed variety classification. *Sensors (Switzerland)*, 15(7):15578–15594, 2015. ISSN 14248220. doi: 10.3390/s150715578.
- [95] Ece Olcay Aygun, Sercan and Yalcin, Hulya and Gunes. Seed Texture Classification by Random Forest and Neural Networks. pages 0–3, 2017.
- [96] Zhao Yan Liu, Fang Cheng, Yi Bin Ying, and Xiu Qin Rao. Identification of rice seed varieties using neural network. *Journal of Zhejiang University: Science*, 6 B(11): 1095–1100, 2005. ISSN 10093095. doi: 10.1631/jzus.2005.B1095.
- [97] Mitchell L. Neilsen, Chaney Courtney, Siddharth Amaravadi, Zhiqiang Xiong, Jesse Poland, and Trevor Rife. A dynamic, real-time algorithm for seed counting. In *Proceedings of the 26th Intl. Conf. on Software Engineering and Data Engineering*, pages 7–12, 2017.
- [98] Xianfeng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3D Object Reconstruction: State-of-the-Art and Trends in the Deep Learning Era. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2019. ISSN 0162-8828. doi: 10.1109/tpami.2019.2954885.
- [99] M. Aharchi and M. Ait Kbir. A Review on 3D Reconstruction Techniques from 2D Images. pages 510–522, 2020. doi: 10.1007/978-3-030-37629-1\_37.
- [100] MakerGear MakerGear™. Makergear m2 3d printer, 2020. URL <https://www.makergear.com/products/m2>.
- [101] Coreform Cubit: Geometry and mesh generation toolkit. <https://coreform.com/products/coreform-cubit>.

- [102] Slic3r - open source 3d printing toolbox. <https://slic3r.org/>.
- [103] Printron: Pure python 3d printing host software. <https://www.pronterface.com/>.
- [104] The imaging source - industrial camera manufacturer. <https://www.theimagingsource.com/>.
- [105] Dfrduino uno r3 - arduino compatible - dfrobot. <https://www.dfrobot.com/product-838.html>.
- [106] gshield v5 - synthetos. <https://synthetos.myshopify.com/products/gshield-v5>.
- [107] Universal gcode sender. [https://winder.github.io/ugs\\_website/](https://winder.github.io/ugs_website/).
- [108] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000. ISBN 9780521623049. URL <https://books.google.com/books?id=MhrYnQEACAAJ>.
- [109] Steinmetz solid - wikipedia. [https://en.wikipedia.org/wiki/Steinmetz\\_solid](https://en.wikipedia.org/wiki/Steinmetz_solid). (Accessed on 10/20/2020).
- [110] Waseem Rawat and Zenghui Wang. Deep Convolutional Neural Networks for Image Classification: A Comprehensive Review. 2733:2709–2733, 2018. doi: 10.1162/NECO. URL [https://www.mitpressjournals.org/doi/pdf/10.1162/neco\[\\_\]a\[\\_\]00990](https://www.mitpressjournals.org/doi/pdf/10.1162/neco[_]a[_]00990).
- [111] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2016-Decem:770–778, 2016. ISSN 10636919. doi: 10.1109/CVPR.2016.90.