Politecnico di Torino

Master's Degree in Aerospace Engineering

Politecnico
di Torino

von KARMAN INSTITUTE
FOR FLUID DYNAMICS

Master's Degree Thesis

Analysis of Slush-State Crystal Structures through Shape-From-Silhouettes 3D Reconstruction

Supervisors:

Prof. Sandra Pieraccini

Prof. Delphine Laboureur

Technical advisor:

Maria Teresa Scelzo

Candidate:

Lorenzo Galante

s290636

December 2023

To my family and friends
always ready to support me.
To science, relentless force,
propelling me forward
in my endeavours.

Abstract

This work proposes a novel approach for reconstructing the three-dimensional (3D) shapes of fluid crystal formations in their slush state, characterised by the coexistence of liquid and solid phases at the triple point temperature.

Slush mixtures present higher density and lower specific enthalpy compared to their liquid counterpart, thus representing attractive candidates for different applications, such as next-generation propellants in space and ground transportation. However, the presence of solid particles in the liquid phase poses notable obstacles in accurately capturing their complex morphologies, and innovative strategies are required to efficiently characterise and employ slush propellants. The present study leverages the Visual Hull technique, which is a well-known method in computer vision and graphics, to address this challenge.

The objectives of this research are threefold: to evaluate the efficacy of advanced computer vision methodologies in biphasic fluid characterisation; to provide a better understanding of crystal structure formation within the slush state; and lastly, to produce an extensive database, encompassing different types of fluid, which will serve as foundation for validating predictive models. Each of these activities constitutes a benchmark in developing groundbreaking storage methodologies for new propellants.

The first part of the work consists of a comprehensive study of several fluids to identify samples that exhibit similar features to real next-generation propellants (in particular hydrogen) and define a variety of test cases. The Visual Hull is then presented and explained, along with the related algorithm for developing the 3D model. Subsequently, a feasibility study is conducted to determine the minimum number of cameras that could be used while maintaining an acceptable error in the reconstruction. The following part describes the experiment implementation. A dedicated facility was designed and built, consisting of 8 paired cameras positioned on a circle around a transparent tank, filled with the chosen test fluid in its liquid phase. From above, a frozen body is suspended in the tank. The experimental section showcases the application of the Visual Hull technique to multiple scenarios involving different samples under controlled conditions. The results obtained from the 3D reconstructions are presented and the performance of the algorithm

is evaluated. In the final section, the broader implications of this research are discussed and future research directions are delineated.

Overall, this thesis provides a solid contribution to the advancement of computational imaging of slush-state fluid crystals, offering a holistic framework for precise 3D shape reconstruction. Moreover, the integration of Visual Hull-based methodologies in biphasic fluids observations represents a noteworthy leap for gaining insights into the qualities of this fascinating state.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Context

Given the ongoing rise in energy demand and the climatic issues humanity is currently experiencing, it is crucial to explore new methods to produce low-carbon energy. In this framework, hydrogen is steadily emerging as one of the most effective energy sources, as it does not emit greenhouse gases when it is produced using renewable energies. Therefore, the academic partnership project **BE-HyFE** (*Belgian Hydrogen Fundamental Expertise*, Figure 1.1) was launched to explore hydrogen's enormous potential. This project aims to support the Belgian industry in the development of a new approach to producing energy, that fulfils European environmental objectives.

Figure 1.1: BE-HyFE (Belgian Hydrogen Fundamental Expertise) logo

The present work is part of a PhD project funded by **BE-HyFE**, which focuses on "Advanced characterization of densified cryogenic hydrogen". Specifically, the research topic seeks to better understand the thermodynamic qualities of hydrogen in its slush state and to ultimately improve storage efficiency. The slush state is a multiphase mixture of vapour, liquid, and solid crystals close to the hydrogen triple point temperature, and storing hydrogen in this state can be a way to optimise its energy storage capacity.

The major objective of the research is to develop a multi-phase solver that can accurately simulate the thermal and dynamical interaction between particles and fluids, as well as take into account the crystals melting process, given their complex geometry. When subjected to specific heat regimes or mechanical stimulation, hydrogen crystals may experience a phase change; as a result, their physical boundaries may assume arbitrary and complex shapes. To validate the numerical solver, an experimental campaign is required, as detailed discussed in [1].

1.2 Project goals and methodology

Numerous researchers, among which Reynier et al [2] and Jin T. et al. [3], pointed out a notable lack of comprehensive knowledge on the slush-state characterisation. This is mainly due to the technical challenges associated with the research, often requiring specialized equipment and techniques, which implies considerable costs as opposed to very limited applications, until very recent years. With the rise of hydrogen propellants as next-generation fuels and energy sources, however, the interest in studying the slush state is constantly growing, as it allows considerable improvements in energy efficiency and thermal management [4].

The present work steps towards this framework, aiming to precisely characterise the slush particles by employing a well-established computer vision methodology, the Visual Hull technique. The advantage lies in establishing a straightforward, cost-effective, and easily repeatable observation strategy.

Understanding when and how the frozen crystals begin to melt due to heat exchange with the liquid phase constitutes a critical aspect of the research. Hence, the study focuses on tracing the evolving shape of the crystal over time, to precisely quantify heat transfer and acquire important data for validating predictive models.

Thus, the goal is to provide an extensive multi-fluid database that accurately

tracks the evolution of melting particle boundaries, which will be useful for the validation of a numerical solver, able to simulate the dynamics and thermal interactions of complex-shaped particles, such as hydrogen crystals in a biphasic flow.

This thesis is organised as follows: the next chapter introduces the slush state, describing its features and properties; in Chapter 3, the principal concept of the Visual Hull technique is explained in detail, along with the related codes used for developing the 3D model. Chapter 4 presents the experimental setup and all its components, while Chapter 5 discusses the calibration process of the setup. Chapter 6 contains the methodology validation, as well as several improvements to the technique. Finally, the resulting 3D reconstructions are shown in Chapter 7, and the conclusive remarks are summarised in Chapter 8.

Chapter 2

Slush state mixtures

2.1 Phenomenon overview

Slurries are dual-phase mixtures where solid particles are dispersed in a liquid carrier [5]; when the liquid and solid phases are formed from the same substance that is partially solidified coexisting in thermal equilibrium, it is referred to as **slush** [2]. Typically, slush formation occurs at the thermodynamic triple point, as shown below:

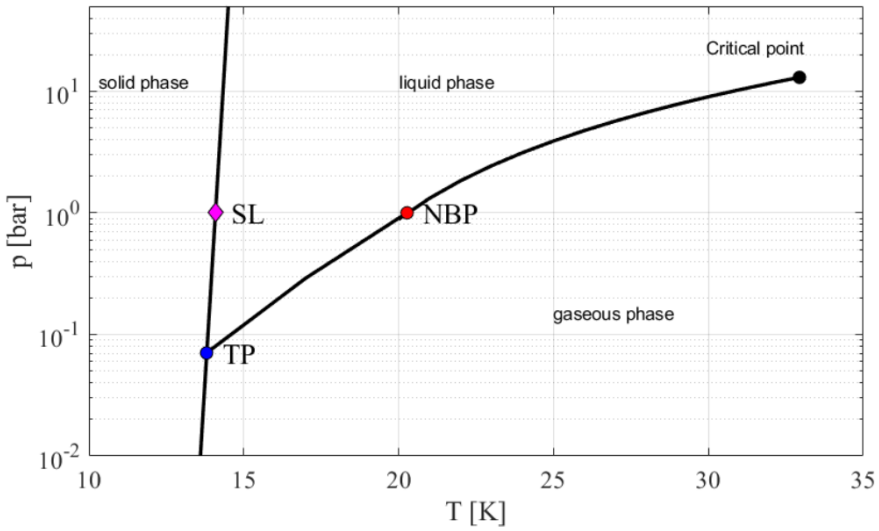


Figure 2.1: Hydrogen phase diagram, with its triple point denoted as TP, the normal boiling point reported as NBP, and an operational slush point indicated as SL [6]

As can be seen in the graph (Figure 2.1), it is possible to easily control the pressure of a slush-state substance by slightly adjusting the temperature. Furthermore, slush mixtures exhibit higher density compared to their liquid counterpart, and the presence of solid crystals reduces their specific enthalpy. These properties are displayed in Figure 2.2, which represents an enthalpy-temperature curve. External heat sources induce the melting of crystals, altering the initial solid fraction in the mixture. Then, the solid phase needs to completely melt before reaching the saturation curve, thus it does not transition to a vapour state as a significant portion of the heat gets absorbed from the solid crystals. This latent heat absorbed during the phase change enhances the slush's capacity to store energy, enabling the slush to endure higher temperatures for an extended period before vaporization [6].

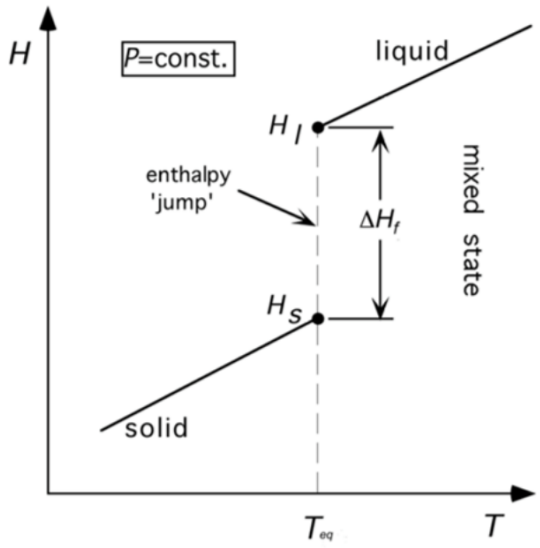


Figure 2.2: Generic enthalpy-temperature diagram [6]

The advantageous combination of lower temperature and specific enthalpy as the density increases makes the slush state an attractive choice for various industrial applications: for example, cryogenic slushes show potential as refrigerants and propellants for future space launchers.

2.2 Slush characterisation

The rheological properties of slush are directly influenced by the size and shape of its crystals, which typically exhibit an irregular shape. Thus, accurate measurements

of these characteristics are essential to efficiently design suitable infrastructures that enable the use, storage and transportation of slush mixtures [6].

The parameters involved in the evaluation of the qualities and properties of a slush mixture are:

- *Solid Volume Fraction* α, which represents the ratio of the solid phase's volume to the overall mixture volume. It is obtained as a weighted average of the densities of the two distinct phases, determined at the slush temperature;
- *Density Ratio* β, defined as the ratio between the substance's solid density and its liquid state density:
$$\beta = \frac{\rho_s}{\rho_l} \quad ,$$
where ρ_s is the solid density and ρ_l is the liquid density;
- *Solid particles mean diameter*, D_p;
- *Liquid Dynamic Viscosity*, μ_l. In literature, the slurry is often regarded as a Newtonian fluid, where viscosity rises as the solid volume fraction increases;
- *Settling velocity* V_{sl}, determined by the equilibrium between gravitational forces and drag force on a particle in a quiescent medium. Hence, it represents the maximum speed that the particle can reach in a liquid when its acceleration becomes zero. As reported in [6], it can be estimated analytically as:
$$V_{sl} = \frac{(\rho_l - \rho_s) g D_p^2}{18 \mu_l} \quad ,$$
where ρ_l, ρ_s are the liquid and solid densities, g is the gravity acceleration, D_p the particle diameter and μ_l the liquid dynamic viscosity of the substance. The settling velocity aligns with the direction of g, and it can be positive or negative depending on the density ratio between the two phases.

2.3 Experimental requirements

As documented in [6], particle size analysis of cryogenic slush has been conducted using various methods, such as photography or microscope imaging, thereby obtaining two-dimensional data about the slush crystals. However, these techniques

are limited in providing information on the three-dimensional shape of particles. Additionally, slush often exhibits a high solid concentration, making it challenging to detect particle borders with flow visualization methods. For these reasons, the application of the **Visual Hull** technique (introduced in Chapter 3) to slush particle observation can be considered extremely attractive, as it provides three-dimensional information from a set of 2D images. Nevertheless, it is necessary to outline definite requirements, due to the limitations of the observation technique and the significance to slush research. Considering the demanding thermodynamic conditions required to achieve the slush state for highly promising substances like hydrogen (Triple point at $T_{TP} = 14$ K) or nitrogen ($T_{TP} = 64$ K), it is worth to explore feasible alternatives, more accessible yet presenting some similarities with the target. This study aims to use hydrogen as a reference and identify other substances sharing one or more features similar to it, for comparative analysis. Accordingly, the guidelines for the implementation of the experiment are the following:

- The selected material must have identical compositions in both its liquid and solid states; in other words, the solid phase must be formed by solidifying the same substance as the liquid phase. In addition, there must not be any third (gaseous) phase involved;
- The solid phase must melt in the liquid phase. In addition, the substance must have a freezing point above $-30°C$, which is the lowest target point of the freezer present in the *Von Karman Institute* laboratory;
- The substance must be transparent, ensuring that the solid phase is distinctly visible within its liquid phase, and in clear contrast with the background. This is essential to enable accurate optical measurements and correctly extract the target's shape from the image;
- The risk hazards in the laboratory must be minimal (i.e. the mixture must not be toxic nor flammable);
- The chosen substitute mixture must have similar characteristics to slush hydrogen (for example, similar β);
- Information on temperature measurements is needed, as they will constitute the boundary condition for predictive model validation. Therefore, thermocouple installation must be considered.

Taking all these requirements into account, the experiment was performed with three different solid-liquid substances, in which the two phases are of the same species and melting occurs. The substances are in their liquid state at room temperature and are contained in a tank to imitate the storage condition.

2.4 Test substance selection

To identify the most suitable species to simulate some characteristics of hydrogen, an extensive literature review was carried out, which highlighted the viability of **Phase Change Materials (PCM)** as attractive candidates for the experiment.

Phase Change Materials (PCMs) are substances able to absorb and release heat during their phase change, thus making them latent heat storage materials. A phase change material (PCM) can efficiently absorb a considerable amount of energy during the melting process and subsequently release stored latent energy when solidifying, and these processes are usually almost isothermal [7], [8]. Several types of PCMs are available, each with distinct transition temperatures and applications. Paraffins, organic compounds derived from petroleum, are a common example. Moreover, they generally possess a relatively low melting point, usually around room temperature. Their most attractive properties for experimental purposes are reported in detail in [7], and summarised below:

- Favorable phase-transition temperature at ambient pressure;
- Several advantageous chemical properties:
  - Long-term chemical stability;
  - Construction materials compatibility;
  - Non-toxic and non-flammable;
- Economics properties: Affordable and widely available.

As a result, a table was compiled containing the most promising species that have been identified for experimental use. The first row of Table 2.1 displays the properties of hydrogen, for comparison with the features of the alternative substances. The settling velocity was calculated for two distinct values of the particle diameter D_p found in literature, to quantify V_{sl} either for small or larger solid particles.

Fluid	T_F [°C]	β	μ_l $[Pa \cdot s]$	V_{sl} $[m/s]$, $D_P = 0.5\ mm$	V_{sl} $[m/s]$, $D_P = 3\ mm$
Hydrogen	**−259.19**	**1.12**	$\mathbf{2.50 \cdot 10^{-5}}$	**0.04**	**1.5971**
Lauric Acid	43.50	1.06	$5.93 \cdot 10^{-3}$	0.0013	0.046
Paraffin Wax	$-50 \sim 60$	$0.92 \sim 1.18$	$\sim 20 \cdot 10^{-3}$	$8.16 \cdot 10^{-4}$	0.0294
Octadecane	27.50	1.04	$3.80 \cdot 10^{-3}$	0.001	0.0361
Eicosane	36.70	1.11	$4 \cdot 10^{-3}$	0.0028	0.0993
Polyethylene Glycol	34	1.02	$16.50 \cdot 10^{-3}$	$1.65 \cdot 10^{-4}$	0.0059
Water	0	0.92	0.89	$-1.25 \cdot 10^{-2}$	$-4.48 \cdot 10^{-1}$

Table 2.1: Substance selection: the relevant properties for each substance are summarised

The first substance listed is lauric acid, a saturated fatty acid of medium-chain length. It can be found in various vegetable fats as well as in coconut and palm kernel oils. It is mentioned in [9], where a certain amount of lauric acid is melted in a vertically inclined rectangular enclosure. However, it exhibits a rather high freezing temperature, thus requiring a heating system for phase change.

The next option is paraffins, which are white or colourless solid waxes. It is composed of saturated hydrocarbons, typically a combination of straight-chain n-alkanes CH3–(CH2)–CH; the crystallisation of the chains causes the release of a significant amount of latent heat. They are highly versatile substances, as the melting point and the latent heat of fusion depends on chain length. Consequently, they can manifest numerous qualities, including a wide range of freezing temperatures, reliability, non-corrosiveness and affordability [7]. Furthermore, a large number of studies employ several paraffin types, thereby making it a well-documented material. A few examples of relevant studies are [10], [11] and [12]. Among the different paraffin varieties, Octadecane is particularly interesting, as showcased in [12], where a solid volume of octadecane is melted in its liquid phase within a spherical tank. It can be seen that the solid phase is clearly recognisable within the transparent liquid phase. Nevertheless, it features a moderately high freezing temperature.

n-Eicosane is an aliphatic hydrocarbon used for candle production and also presents several applications for cosmetics, lubricants, plasticizers, and the petrochemical industry. It is a paraffin-based compound, employed by Kozak, Rozenfeld & Ziskind [13] for a study on close-contact melting effects.

Another appealing alternative is Polyethylene glycol (PEG), whose use is reported in [14]. It is a synthetic polyether compound known for its biocompatibility and hydrophilic nature. The most common applications are in the medical field as well as in the industrial and chemical sectors.

After a first assessment of the properties listed in the table, it was decided to discard all substances that require a heating system for melting, thus focusing on the identification of suitable paraffin for the considered experimental conditions. In particular, the company `Rubitherm GmbH` [15] produces a wide variety of kinds of paraffin for heat-storage applications; of these, two types were selected, whose properties are provided in the summary Table 2.2. Moreover, slush water is also included in both Table 2.1 and Table 2.2: despite its low-density ratio and high viscosity, it is easily available and manageable, thus serving as an initial simple test case for demonstrating the efficacy of the Visual Hull Technique.

Substance	T_F [°C]	β	ρ_l $[g/cm^3]$	ρ_s $[g/cm^3]$	μ_l $[mPa \cdot s]$	V_{sl} $[m/s]$, $D_P = 2\ mm$
Hydrogen	**−259.19**	**1.12**	**0.069**	**0.077**	**0.025**	**0.71**
Water	0	0.92	1	0.92	0.89	0.2052
RT-9HC	-9	1.17	0.75	0.88	1.1	0.2576
RT5HC	5	1.14	0.77	0.88	1.6	0.1499

Table 2.2: Selected substances properties

As can be seen, RT-9HC and RT5HC paraffins feature a density ratio fairly similar to that of hydrogen (especially RT5HC). The former freezes at $-9°C$, whereas the latter at $5°C$. In addition, both types are in liquid form at room temperature and do not require any specific storage methods.

Upon completion of this investigation, the structure of the experiment emerges: for each substance, a small part will be frozen, and then immersed in its liquid phase within a transparent tank. The alterations in both volume and temperature will be meticulously tracked over time.

Chapter 3

The Visual Hull Technique

3.1 Theory

The chosen technique for this work is called **Visual Hull construction**, and represents the starting point for a large majority of 3D shape construction methodologies, hence constituting an important challenge in the field of computer vision.

Visual Hull's fundamental idea is to use an object's *silhouettes* from multiple angles to reconstruct it in three dimensions. A silhouette can be defined as a two-dimensional representation of the outline of an object; every silhouette created by a camera view confines the subject to a *visual cone* (Figure 3.1). Specifically, each visual cone represents the projection of a silhouette obtained from a different view, and the intersection of all these cones forms a reconstruction of the real object's shape [16]. For this reason, this technique is part of a larger family of techniques that use silhouettes for the purpose of 3D reconstruction of the observed object, known as *Shape From Silhouette* (SfS), first introduced by Baumgart in his PhD thesis (1974) [17].

The final result is therefore an approximate reconstruction of the real shape of the observed object.

Using the notation provided by Laurentini in 1994 [19], it is possible to define the set of silhouette images obtained from N cameras at time t_i as:

$$\{\{S_i^n\}; n = 1, ..., N\} \quad .$$

Being the perspective projection function $\{\Pi^n\}$ employed to transform the 2D

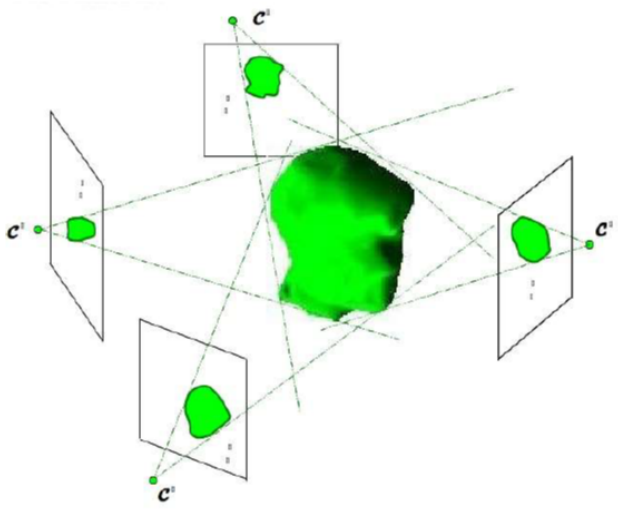


Figure 3.1: Visual Hull shape reconstruction, via the intersection of the visual cones [18]

image coordinates of a 3D point P in the n^{th} image, a volume Ω explains exactly $\{S_i^n\}$ if its projection onto the n^{th} image plane coincides with the silhouette image $\{S_i^n\}$ for all $n \in \{1, ..., N\}$ so that:

$$\Pi_n(\Omega) = S_i^n \quad .$$

The Visual Hull is defined to be the largest volume which exactly explains $\{S_i^n\}$ for all $n \in \{1, ..., N\}$ [18]. Therefore, the reconstruction gets better as the number of cones increases. Another important issue is the position of the cameras: the angle between each view should be reasonably wide, in order to capture all the features of the target. This can be seen in Figure 3.2, where every C_i^n represents a camera viewpoint.

3.2 Features of the technique

The main advantage of the *Shape-From-Silhouette* approaches is that the computation of the silhouettes is easy to implement, assuming an indoor environment with static lights and cameras. Where these conditions are not fulfilled, shadows or moving backgrounds can make it challenging to extract an accurate silhouette from the images. Moreover, the SfS construction provides an upper bound for object shape without knowing the reflectance and texture properties of the object.

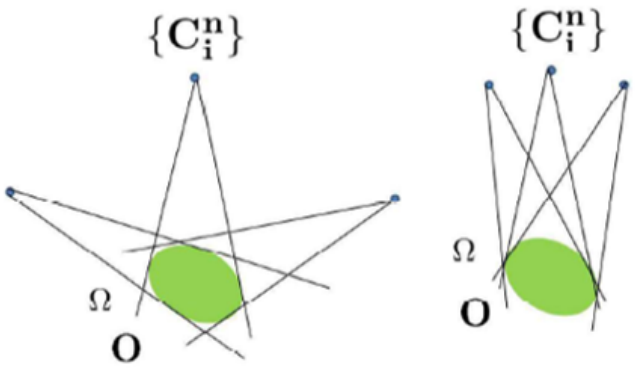


Figure 3.2: Importance of cameras positioning [18]

However, these algorithms also present some drawbacks to be addressed. First of all, the silhouette calculation is very sensitive to errors in camera calibrations and noise, which may cause problems for the intersection of the visual cones, therefore resulting in poor 3D final reconstructions. The results are also not accurate for concave objects, and the method requires a precise segmentation of the object silhouette from its background. Furthermore, the result of each SfS algorithm is just an approximation of the actual object's shape, especially if there are only a limited number of cameras; as a result, accurate reconstructions depend on the fulfilment of various conditions (such as proper light, object segmentation from the background, etc.) that cannot always be achieved, thus limiting the applications of the method.

3.3 Voxel-based approximation

The simplest way to obtain an actual construction of the Visual Hull would be by intersecting the visual cones (as showed in Figure 3.3); however, as the Visual Hull reconstruction consists of a curved and irregular surface, it requires a sophisticated geometrical representation for its cones. This leads to higher complexity and numerical instability, which can be avoided through the use of approximate representations, such as polyhedral shape or volume-based approach. In practice, the voxel-based SfS computation (which is a form of volume-based approximation) relies on the same assumptions of the visual cone intersection, but the final shape representation is done by 3D volume elements (called **voxels**). A voxel is a 3D space into a regular grid of elements, or in other words, is the

smallest unit of volume when dividing 3D space into discrete regions. Depending on the location of each voxel in the visual cones (inside or outside the silhouette), the region of space is divided into sections: all voxels forming a silhouette are intersected to reconstruct the 3D shape of the object, while voxels belonging to the background are discarded.

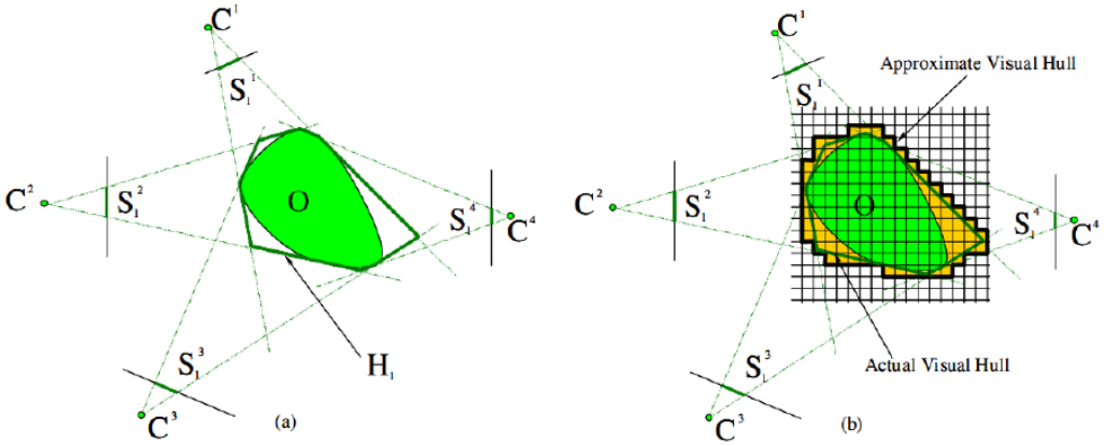


Figure 3.3: Left **(a)**: 2D example of visual cones intersection. Different views of the object O lead to different silhouettes, and the intersection of their projection forms the Visual Hull. Right **(b)**: 2D example of a voxel-based approximation of an object by a certain algorithm [16]

Although this technique is straightforward and fast, it has one major disadvantage: the resulting shape is significantly larger than the actual shape of the object (as illustrated in Figure 3.3), which makes it feasible only for applications where an approximation is sufficient. Nevertheless, by providing a sufficient number of visual cones (i.e. viewpoints) and a strategic camera placement, it is possible to overcome these limitations.

3.4 The Deterministic Visual Hull

The algorithm adopted in the current work is an implementation of the voxel-based Visual Hull reconstruction method developed by Egri & Han [20]. The method is called **Deterministic Visual Hull** and it makes use of *Foreground Segmentation* algorithms to determine if the voxel is inside or outside the silhouette. Basically, these algorithms replace the image with a binary mask (silhouette) that indicates which pixels are part of the object projection and which are part of the foreground: this means that if a pixel is inside the silhouette, its value is 1 (white), otherwise,

its value is 0 (black) and is part of the background (therefore, it is discarded). Then, they perform the conversion of the 3D pixels (X,Y,Z) from voxels to 2D values (x,y); hence, control is implemented to check whether the pixel belongs or falls under the silhouette.

Based on the Visual Hull algorithm, the desired dataset needs to contain images taken from a variety of views, together with the camera's intrinsic and extrinsic parameters. The code created by Egri & Han (that can be found in their GitHub [20]) uses the *Middlebury Multi-view Stereo dataset* [21].

3.4.1 Algorithm implementation

The Voxel-Based Visual Hull is computed with the following algorithm:

1. Read the files in input (images, camera parameters);
2. Rebuild the camera parameters matrices and vectors;
3. Pre-process the input images (if needed) and compute the silhouettes from them;
4. Create a Voxel grid, which constitutes the 3D space in which the shape of the object is computed. The dimensions are $M \times M \times M$, in which each voxel stores a value in x, y, z coordinates in the interval $[0, N]$;
5. For each voxel (M^3 in total), initialize with count 0;
6. For each silhouette, from $i = 1$ to N:
   For each voxel $v_j = (x; y; z)$, from $j = 1$ to $M \times M \times M$:
   - Perform the voxel projection onto the *i-th* image plane using $P_i(x; y; z; 1)T$, where P_i represents the projection matrix, and $(x; y; z; 1)T$ denotes the voxel's homogeneous coordinates. This process is divided into two steps:
     - project voxels centers to image;
     - project voxels to silhouette;
   - If the projection falls within the object mask, increment the voxel's total count by 1;
7. Perform operations on the voxel grid:
   - Store the computed voxel grid V;

- Compute a threshold T to determine whether a particular voxel gets filled (binary 1) or not (binary 0) in the final reconstruction. Hence, all voxels with a count $\geq$ T are regarded as filled and part of the object, whereas the others are considered in the background and remain empty;

8. Obtain mesh representation by converting the voxel grid by applying the Marching Cubes algorithm (generally used to extract iso-surfaces from volumetric data).

3.5 Input data optimisation

The existing Deterministic Visual Hull code, as presented by its authors, demonstrates excellent results when provided with 16 camera views as inputs, as presented by the authors [20]. Additionally, the dataset was obtained through the use of a mechanism able to precisely rotate a single camera around a stationary object, thus providing very accurate data. However, this sequential approach is impractical when applied to objects with dynamic shapes that evolve over time; therefore, a new methodology needs to be developed, involving multiple cameras to simultaneously capture images from different viewpoints.

Another important aspect is the significant complexity of the experimental setup, which also implies a substantial manufacturing cost. Hence, it becomes essential to explore the feasibility of reducing the number of camera views while maintaining an acceptable level of error in the final reconstruction (relative to the optimal scenario with 16 images). This leads to determining the optimal number of cameras that ensures an acceptable level of reconstruction.

The devised strategy for error estimation and viewpoint optimisation is divided into the following steps:

1. Identify different reliable error metrics, to evaluate the error between the reconstructions;

2. Quantification of the errors in reconstructions with a suboptimal number of images using the original dataset (specifically, the dinosaur sculpture from the *Middlebury dataset* [21]. The method consists of increasing the angle of observation between the views, thus eliminating some input images (since every view is defined by a set of angular coordinates);

3. Compare the errors and the reconstructions to identify the best trade-off result.

3.5.1 Error evaluation metrics

Given the voxel grid of the original reconstruction (obtained from the *Middlebury dataset* of a dinosaur sculpture [21]), two distinct metrics were employed to quantitatively assess the impact of reducing camera views on the reconstruction.

The first metric is the **voxel average error** and it involves voxel-wise comparison between two reconstructions. It is determined as the weighted average of the absolute difference between the studied reconstruction and the reference, hence denoting an absolute measurement expressed in data units (voxels). It is calculated as:

$$Err_{vox,avg} = \frac{\sum |voxel_{new} - voxel_{ref}|}{voxel\ number} \quad ,$$

where $voxel_{new}$ is the voxel grid of the considered reconstruction, while $voxel_{ref}$ is the reference. This metric provides insight into the deviation of the reconstruction at a voxel level: by calculating the difference between each pair of values, the obtained values indicate how much the values differ from each other.

The second metric, on the other hand, introduces the concept of **occupied volume difference**: this involves calculating the volume of the original reconstruction inside the bounding box (in terms of voxels) and then subtracting it from the volume occupied by the studied reconstruction within the same bounding box. Therefore, this metric assesses the similarities between the volumes of the two reconstructions. If the difference is small, the two reconstructions probably share a similar distribution of occupied voxels; conversely, a high value indicates that there is a significant discrepancy between volumes. It is calculated as:

$$\Delta Vol_{occ}\ [\%] = \frac{Volume_{new} - Volume_{ref}}{Volume_{ref}} \cdot 100 \quad ,$$

where the new volume is determined as the product between the number of occupied voxels in the grid and a single voxel size. As the number of viewpoints decreases, a greater volume is expected, since there will be less detail retrieved from the edges of the silhouettes.

3.5.2 Error Analysis and Results

The results are summarised below in Table 3.1:

Number of images	Average Voxel Error	ΔVol_{occ}
16 images	-	-
15 images	0.6211	0.0022 %
14 images	1.6899	0.6370 %
10 images	3.508	1.3635 %
8 images	4.3743	2.7868 %
6 images	5.5066	8.5527 %

Table 3.1: Error comparison when decreasing the number of images

As the input images decrease, there is a reduction in the accuracy of the reconstruction and a consequent increment of the object volume, which follows a non-linear trend. Specifically, it can be seen that for up to 10 input images the differences are minimal: the values in the considered voxel grid have a discrepancy up to about 3.5 units from the corresponding points in the reference, resulting in a very similar reconstruction. The volume further displays a slight increase with 8 viewpoints (less than 3% deviation from the original), but then the error considerably increases when using only 6 images, to almost 9%.

By evaluating the quality of the reconstructions and taking into account the metrics for quantifying the error, the error obtained with 8 images, less than 3%, was considered acceptable, with the advantage of a significant reduction in the viewpoints required for reconstruction (half of the initial inputs).

After evaluating the reconstruction quality and taking into account the error through the presented metrics, it was decided to employ a total of 8 viewpoints, as it differs by less than 3% from the reference with the advantage of a significant reduction in the required viewpoints (half of the initial inputs). Therefore, an experimental setup equipped with 8 cameras was designed, allowing the acquisition of 8 different views around the target.

Chapter 4

Experimental setup

In this chapter, the main components of the experimental setup are presented, as well as the complete setup views and its geometric model.

The complete experimental setup appears as follows:

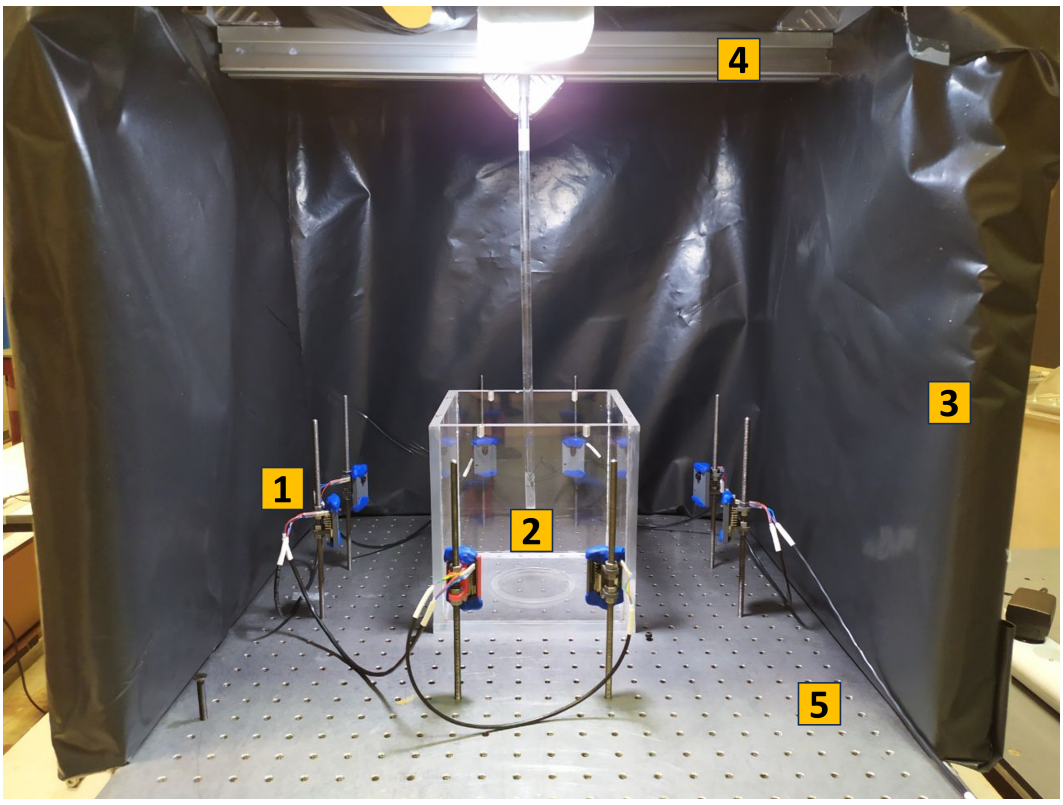


Figure 4.1: Complete Visual Hull setup structure, front view. Each component is marked by a number: 1) Cameras, 2) Tank, 3) Background box, 4) Support, 5) Optical plate

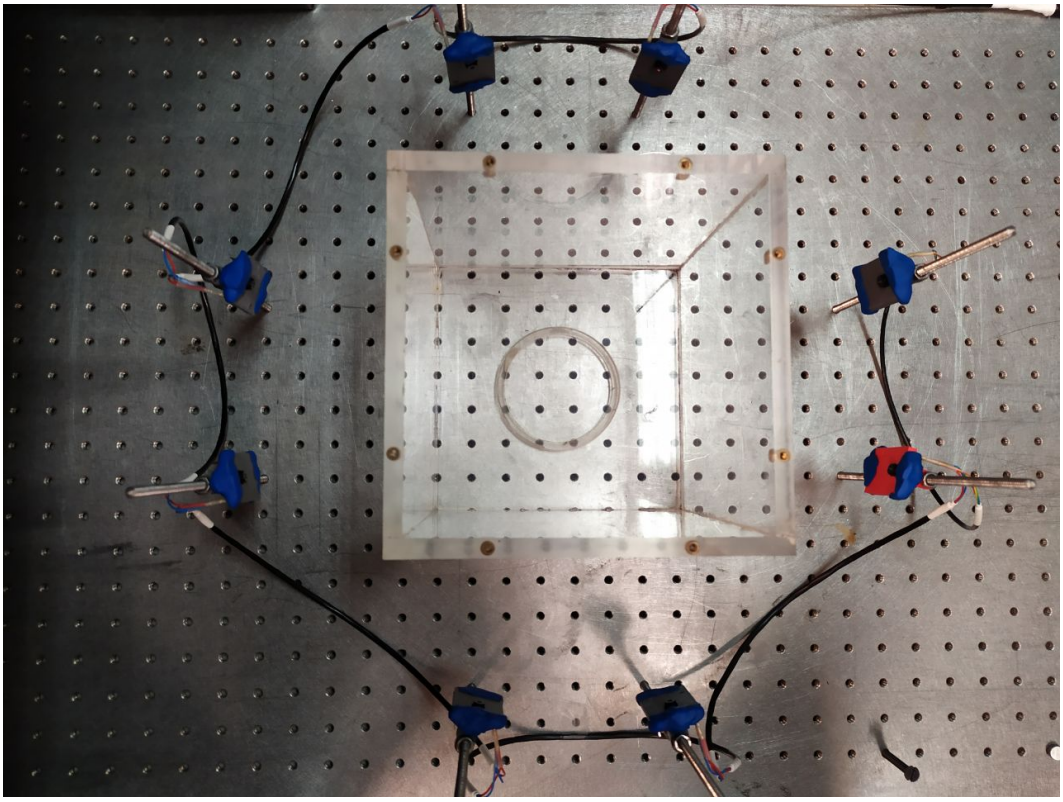


Figure 4.2: Experimental setup, top view

4.1 Cameras

The camera model employed in the experimental setup is the **ESP32 Cam**, developed by `Espressif Systems`. The ESP32 Cam is a compact camera module, which integrates the microcontroller with the camera sensor OV2640; it possesses remarkable technical specifications, providing substantial computing power for image processing and data handling.

The main features and characteristics are presented in the following Table 4.1. The camera is shown in Figure 4.3; additional information can be found at [22].

The necessary code for its operation is built-in in the `espressif` library on the `Arduino IDE` platform. To upload the script an FTDI programmer (shown in Figure 4.4) is needed, which serves as a USB-to-Serial converter, allowing to plug the camera to a USB port. Then, the camera establishes a web connection and can be controlled from the user interface on the terminal. For the present scenario, eight cameras were deployed, and connected to the same local wireless network.

To achieve precise control of the camera positions during experiments, a suitable

ESP32 Cam & Lens	
Microcontroller	32-bit dual-core processor
RAM	520KB SRAM + 4M PSRAM
Image Sensor	OV2640
Image resolution	up to 1600x1200
Field Of View	65°
Camera dimensions	40.5mm x 27mm x 4.5mm
Additional features	Onboard PCB antenna Wi-Fi and Bluetooth capabilities Arduino, MicroPython compatibility

Table 4.1: Camera specifications [22]

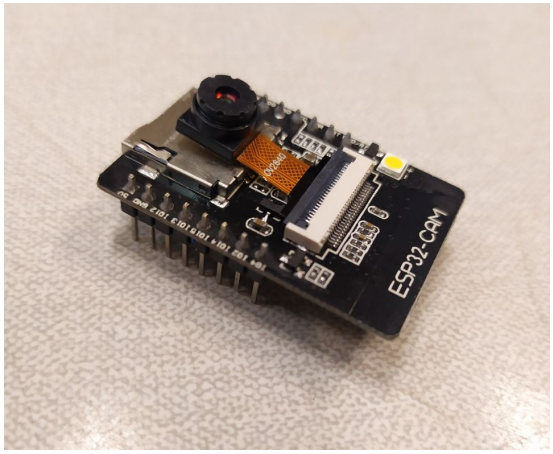


Figure 4.3: Esp32 Cam with OV2640 lens

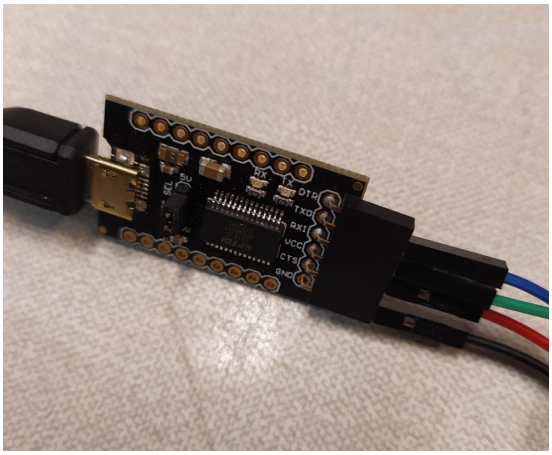


Figure 4.4: FTDI Programmer

cover and threaded support were employed. The cover, from [23], serves a dual function: firstly, it fixes the camera's position in the setup, blocking unnecessary degrees of freedom; secondly, it provides physical protection, preventing accidental damage to the delicate camera module. Additionally, the cover is fastened with multiple nuts to the threaded support, which is then inserted into the optical breadboard, ensuring the camera stability and allowing height adjustments. The cover and the threaded support are shown in Figure 4.5.

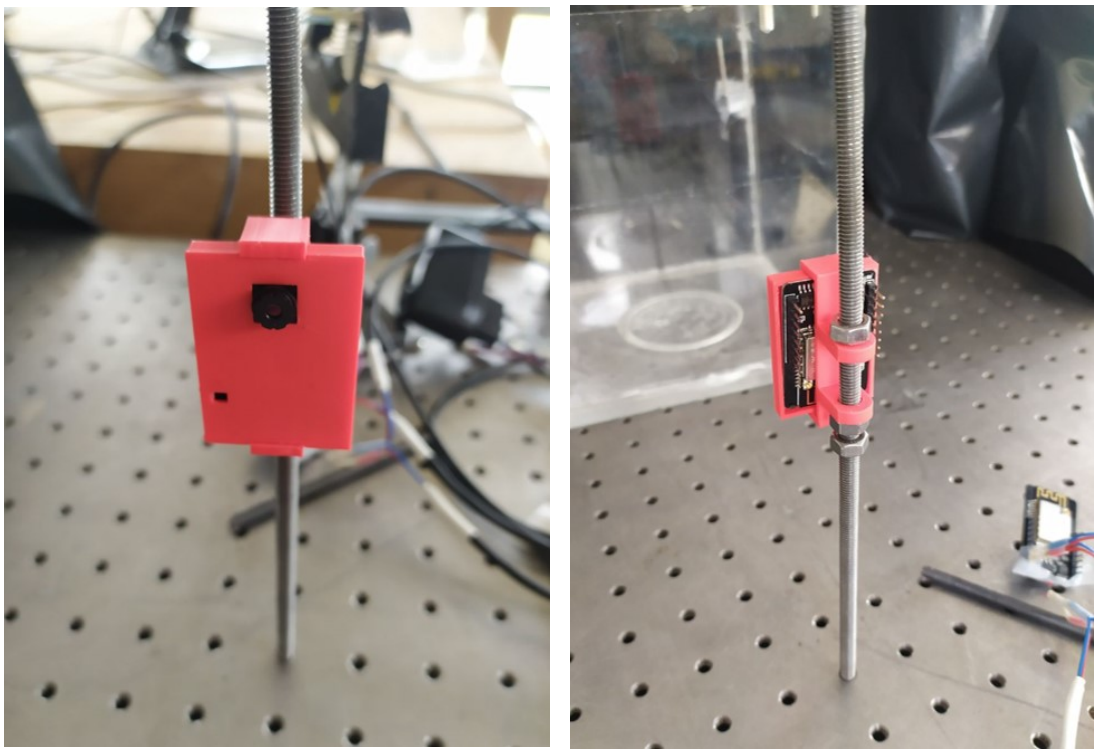


Figure 4.5: Camera cover fastened on the threaded support

4.2 Tank

The transparent tank is cubic-shaped with external dimensions $20 \times 20 \times 21$ cm (length $\times$ width $\times$ height). The walls thickness is 1 cm, thus providing an internal volume of $18 \times 18 \times 20$ $cm = 6480$ $cm^3 = 6.48$ liters. It is made of plexiglass to ensure transparency, allowing for clear observation of solid-liquid interactions. Furthermore, the cubic shape enables a uniform distribution of the fluid, maintaining homogenous conditions everywhere in the volume, and provides sufficient distance of the frozen object from the walls. The tank is depicted in Figure 4.6.

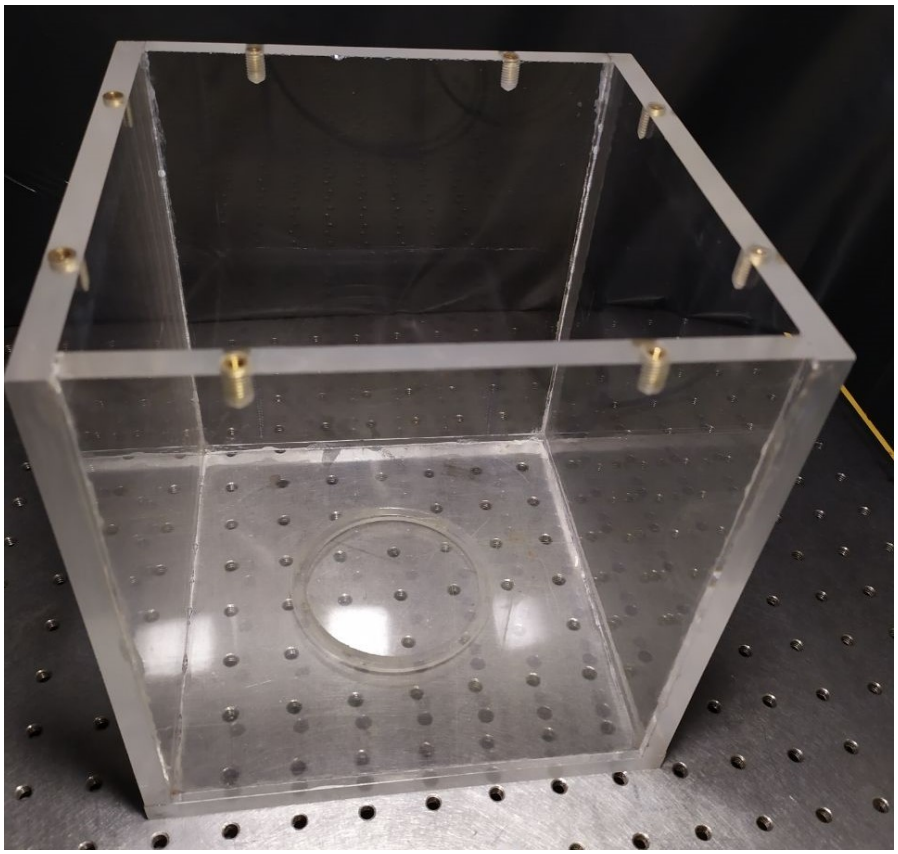


Figure 4.6: Plexiglass cubic tank

4.3 Background box, support and optical breadboard

For the deterministic visual hull algorithm to operate smoothly, a dark background is required: therefore, a cubic metal framework with one side open was built, with dimensions $60 \times 60 \times 60$ *cm*. The lateral walls were then wrapped with black plastic. A metal support with a plexiglass rod was then placed on the upper part of the lateral sides, to allow the thermocouples to be positioned in the tank, as shown in Figure 4.7.

Lastly, these components are placed on the optical breadboard, a flat metal surface provided with threaded holes at fixed distances, allowing standardised attachment points to secure the elements of the setup.

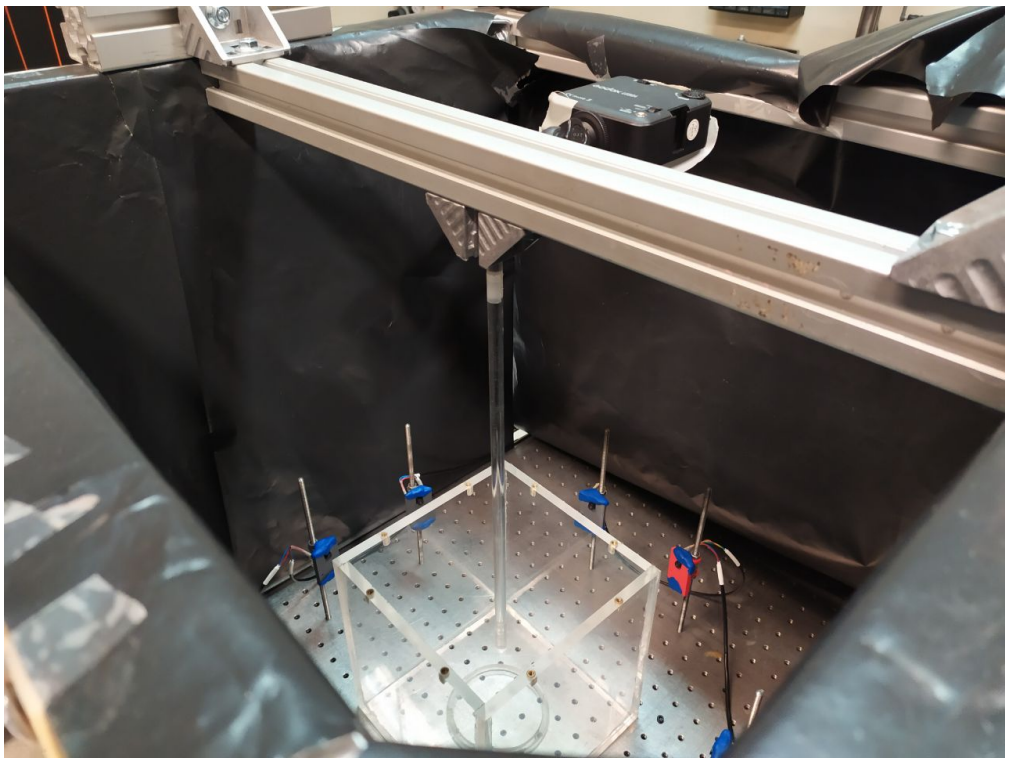


Figure 4.7: Background box and metal support of the transparent rod (upper view)

4.4 Thermocouples

A thermocouple is a sensor composed of two different metal wires joined at one end, called junction. When a temperature difference occurs between the junction and the other end (known as cold junction or reference junction), a voltage is generated across the wires. This voltage is directly proportional to the temperature variation, thus measuring the temperature at the hot junction [24].

The fundamental principle behind a thermocouple is the Seebeck effect: different metals generate distinct voltages when subjected to the same temperature gradient, thereby enabling the temperature to be determined by correlating it with the voltage difference. This is displayed in Figure 4.8:

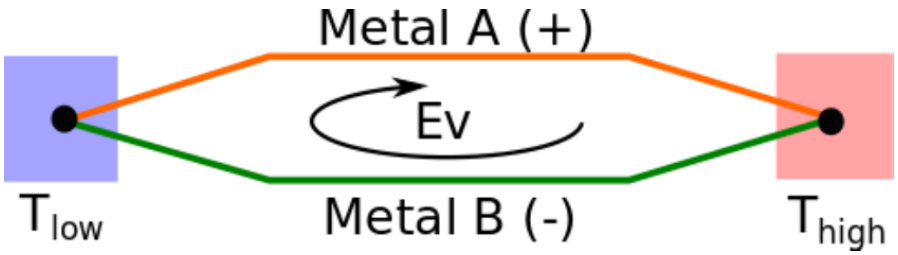


Figure 4.8: Basic principle of a thermocouple [24]

Thermocouples employ different combinations of metals or metal alloys. For the considered Visual Hull experimental setup, the type K thermocouples were used, made of chromel and alumel; the principal advantage of this sensor type is its low cost and wide temperature range. Specifically, a total of five thermocouples were deemed necessary, to monitor the heat transfer in the tank; the configuration is displayed in Figure 4.9. Two thermocouples measure the vertical temperature gradient, one at the bottom of the tank and one on the fluid surface at the top; a thermocouple is located on a lateral wall, while the last one is placed inside the frozen object. In this way, it is possible to assess the boundary conditions of the experiment. Temperature acquisitions are then obtained using the `Labview` software.

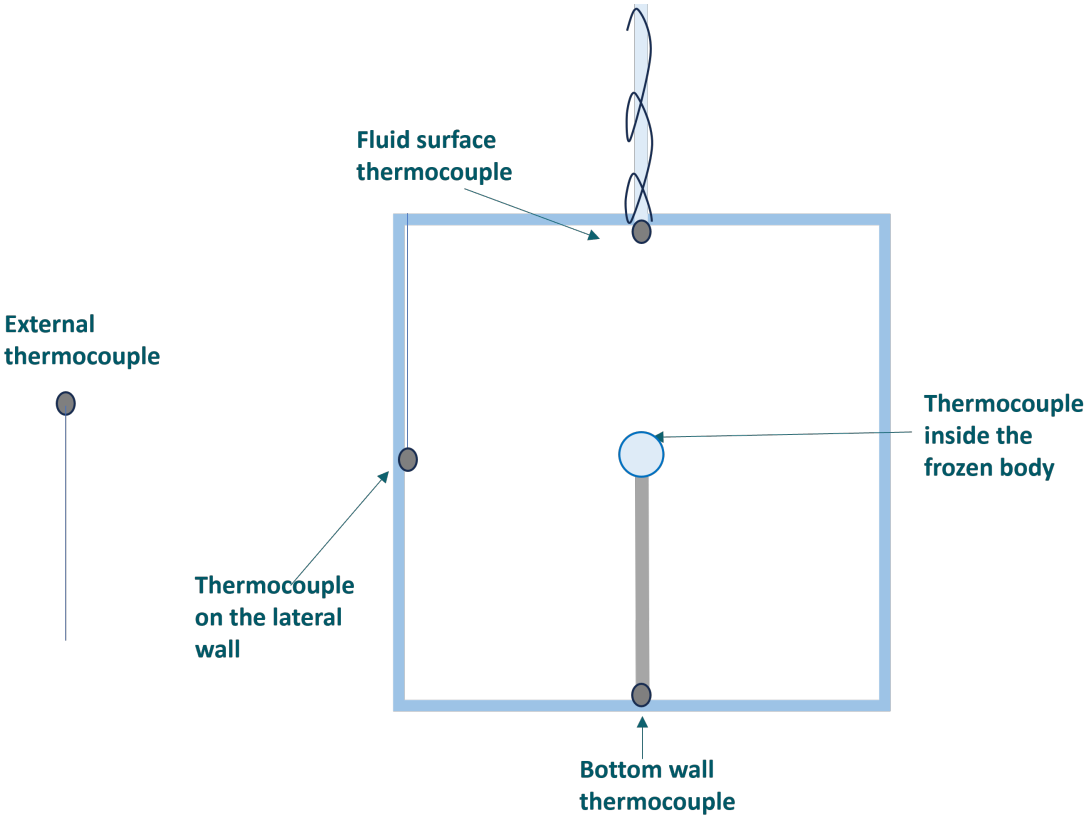


Figure 4.9: Thermocouples placement scheme

4.5 Modelling of the experimental setup

The establishment of a robust experimental framework is fundamental for accurate data acquisition, especially when it comes to three-dimensional reconstruction from images. To precisely track the dimensions and camera positions within the physical setup, a `MATLAB` model has been implemented. This model serves as a virtual representation of the real-world configuration, a "*digital twin*" that includes all the geometric information such as camera heights, orientation, and distances from the target. The inclusion of geometric information facilitates a comprehensive understanding of the experimental setup, while also providing a basis for the subsequent calibration process.

Assuming the center of the tank as the origin of the geometric reference system denoted as (X, Y, Z), and the real-world reference system indicated as (x_r, y_r, z_r), the cameras positions in (X, Y, Z) are summarised in the following table:

Camera	X $[cm]$	Y $[cm]$	Z $[cm]$
1	25.5	1.5	6.5
2	25.5	1.5	-6.5
3	6.5	1.5	-25.5
4	-6.5	1.5	-25.5
5	-25.5	1.5	-6.5
6	-25.5	1.5	6.5
7	-6.5	1.5	25.5
8	6.5	1.5	25.5

Table 4.2: Camera coordinates in the geometric reference system

The cameras are indeed positioned at a height of 1.5 cm from the origin, which corresponds to 11.5 cm from the ground in (x_r, y_r, z_r), around a circumference of 52.63 cm diameter. Moreover, the cameras are paired, and the separation distance between two cameras within a pair measures 13 cm. The angle between a camera pair and the setup reference system's origin (the tank's center) was denoted as θ, and can be calculated through the following process:

1. Computation of the circumference radius on which the cameras are placed. This parameter can be easily derived by knowing that the rectangle formed

by two opposing pairs has dimensions 51 cm × 13 cm since the diameter will coincide with the rectangle's diagonal:

$$D = \sqrt{51^2 + 13^2} = 52.63\ cm \quad \rightarrow \quad r = D/2 = 26.32\ cm \quad ;$$

2. Once the radius r is known, it is possible to determine the angle between two points and the center of a circumference Θ, with a trigonometric equation:

$$L = \frac{\Theta}{360°} \cdot 2\pi r \quad \rightarrow \quad \Theta = \frac{L}{2\pi r} \cdot 360° \approx 29° \quad ,$$

where $L = 13\ cm$ is the horizontal distance between Camera 1 and Camera 2.

Below is a visualisation of the three-dimensional geometric model, obtained using `MATLAB`:

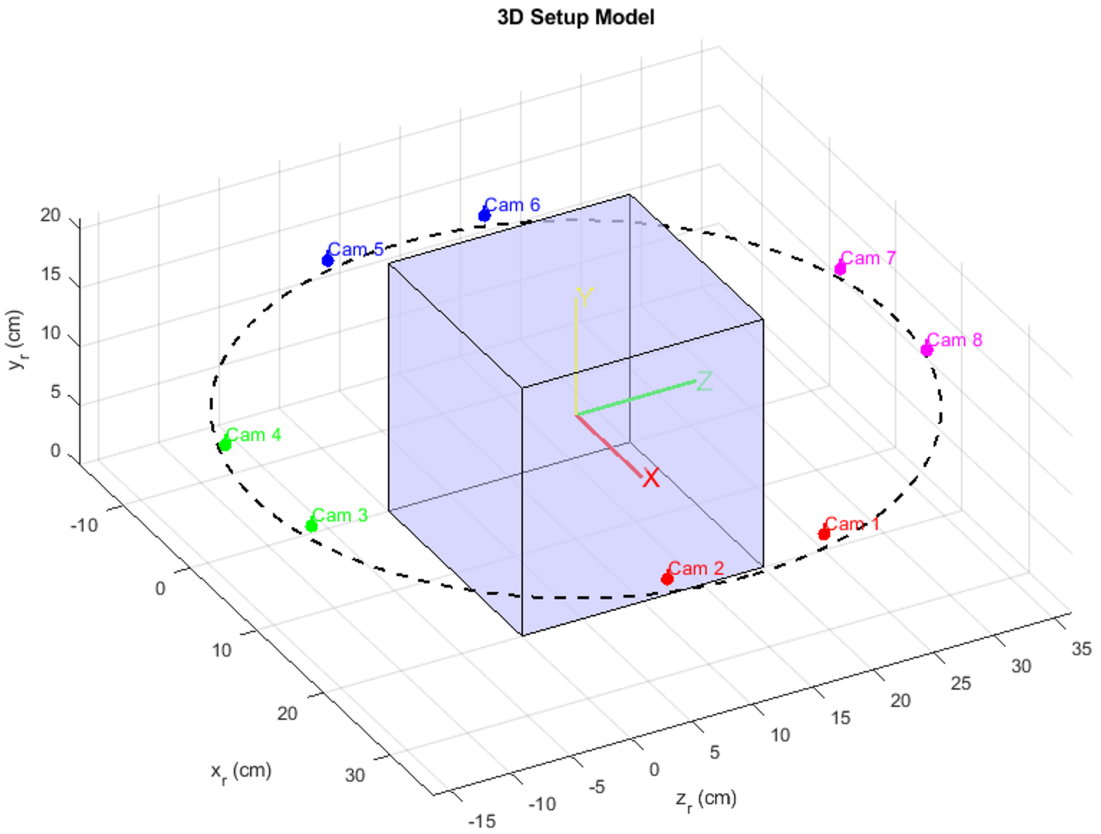


Figure 4.10: Digital twin of the experimental setup; the origin of the setup reference system (X, Y, Z) is in the center of the tank

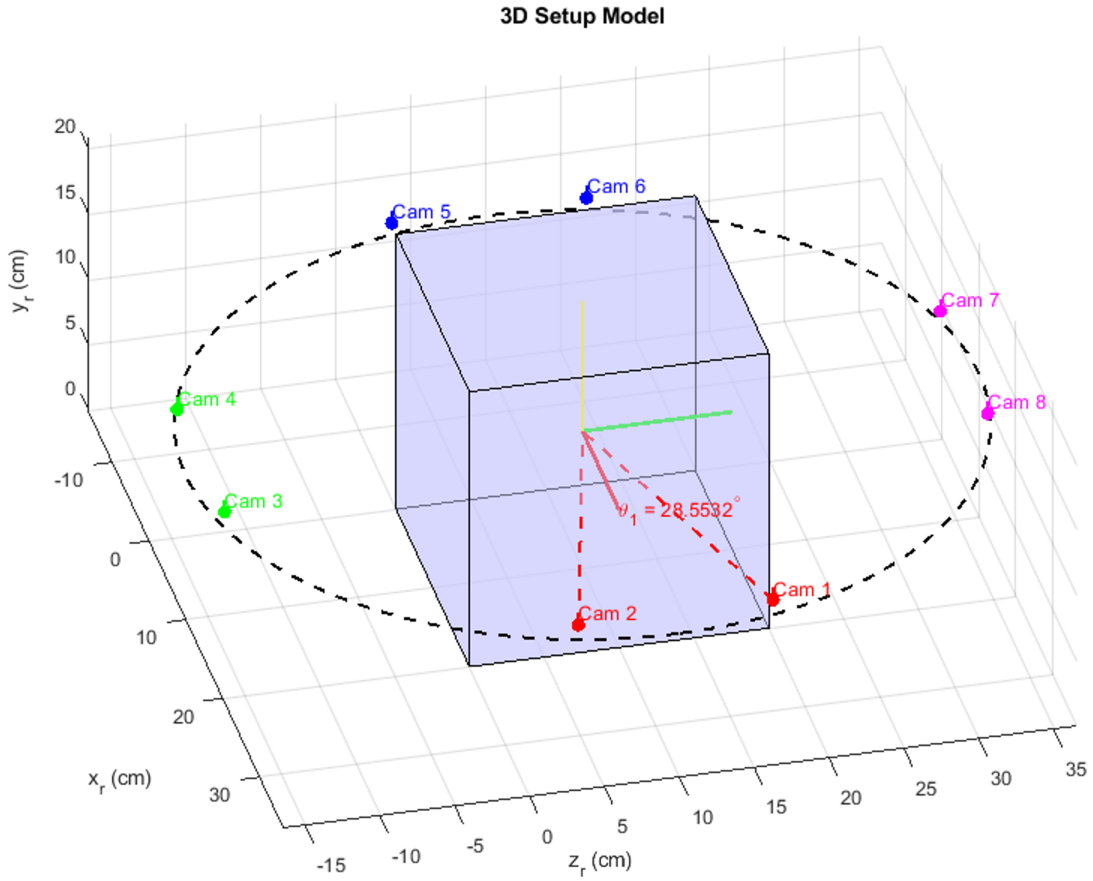


Figure 4.11: Digital twin of the experimental setup: the Θ is displayed in this image

Chapter 5

System Calibration

This chapter first introduces the theoretical model adopted for the calibration process; then, the calibration methods used on the setup are presented and explained. Afterwards, the calibration results are displayed and commented on.

5.1 Geometric camera calibration

The procedure of determining the parameters of a camera is called camera calibration. This means assessing all the information about the camera, hence establishing a precise correlation between a 3D point in the real world and the corresponding 2D projection (pixel) captured in the camera's image [25].

Typically, there are two kinds of parameters, each of which models a transformation:

- **Internal (or intrinsic) parameters:** refer to the camera or lens system. They represent the transformation that projects the world points viewed by the camera on its image plane. E.g. focal length, optical center (the center of projection), and radial distortion coefficients of the lens;

- **External (or extrinsic) parameters:** they determine the camera's orientation and position (rotation and translation) in relation to a world coordinate system. They model the transformation from the world coordinate system to the camera coordinate system. Essentially, they let the user view the world from the camera's perspective.

The theoretical model of the camera employed to perform the calibration is called *pinhole camera*.

5.1.1 The Pinhole Camera Model

The simplest model for an optical camera is the *pinhole camera model*. This camera system can be designed as a lightproof box with a small aperture between the object to observe and a photographic film or sensor [26]. As illustrated in Figure 5.1, every point of a 3D object reflects several rays of light outwards; however, only one (or a few) of these light rays of light enter the aperture and reach the film reaching and exposing the film. This establishes a direct correspondence between points on the 3D object and the film, effectively producing an image of the object, which appears rotated by 180 degrees.

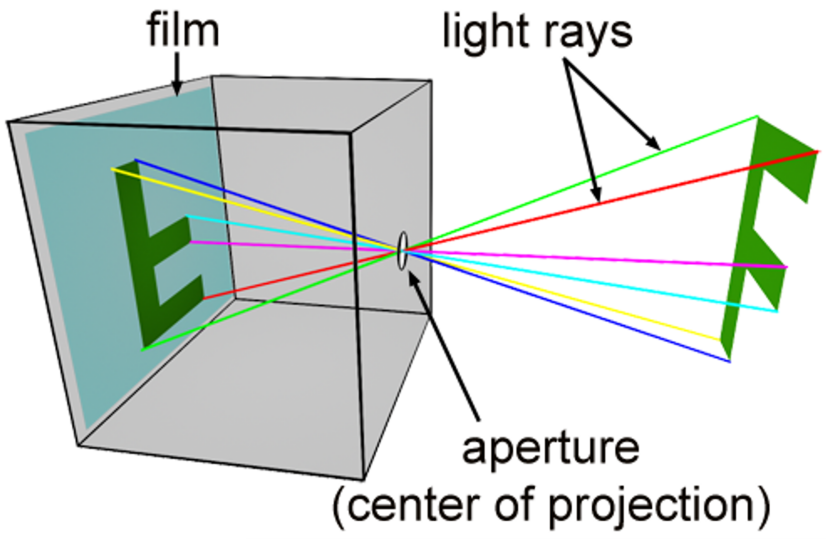


Figure 5.1: Pinhole camera model [27]

The pinhole camera model describes the mathematical relationship between the coordinates of a point in three-dimensional space and its projection onto the image plane of an ideal pinhole camera, where the camera aperture is described as a point and no lenses are used to focus light.

To mathematically model the pinhole camera, the following assumptions are made:

- The image plane, which corresponds to the sensor that captures the light

beams, is located in front of the pinhole, contrary to its actual placement i.e. behind the pinhole;

- All the incoming rays converge at the pinhole, which in geometric terms is called the *center of projection* or *camera center*, and diverge from it on the opposite side.

The idea is that the image of a point corresponds to its projection on the image plane, and this occurs where the line extending from the camera center to the point intersects the image plane [28].

Even if this simple model presents a certain degree of approximation, it represents a reasonable description of how a camera depicts a 3D scene, especially if a high-quality camera is used.

5.1.2 Intrinsics and extrinsics determination

Projecting a 3D point in the world coordinate system to camera pixel coordinates requires two steps: the 3D point is first transformed from world coordinates to camera coordinates with the use of the *Extrinsic Matrix*, which contains the information of rotation and translation between the two coordinate systems. Subsequently, the new 3D point in the camera coordinate system is projected onto the image plane through the *Intrinsic Matrix*, which contains the internal camera parameters. The Figure 5.2 illustrates the process.

The equations that relate 3D point (X_w, Y_w, Z_w) in world coordinates to its projection (u, v) in the image coordinates are:

$$\begin{bmatrix} u' \\ v' \\ z' \end{bmatrix} = \mathbf{P} \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad , \quad \textit{where:} \quad u = \frac{u'}{w'} \quad , \quad v = \frac{v'}{w'} \quad ,$$

$\mathbf{P}$ is the 3×4 Projection matrix and is composed of two parts: the intrinsic matrix $\mathbf{K}$ that contains the intrinsic parameters, and the extrinsic matrix $[\mathbf{R} \mid \mathbf{t}]$ which is a combination of a 3×3 rotation matrix $\mathbf{R}$ and a 3×1 translation $\mathbf{t}$ vector.

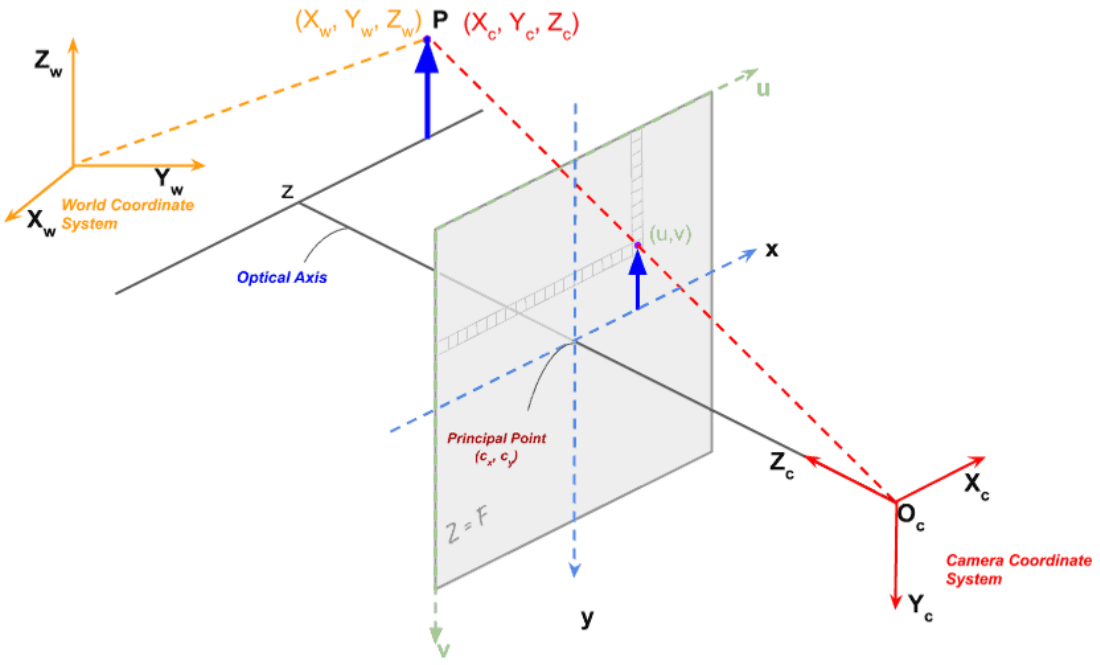


Figure 5.2: Projection of the point P onto the image plane [25]

The previous equation can be more clearly formulated as:

$$\mathbf{P} = \overbrace{\mathbf{K}}^{\text{Intrinsic Matrix}} \times \overbrace{[\mathbf{R} \mid \mathbf{t}]}^{\text{Extrinsic Matrix}}$$

Moreover, the intrinsic matrix $\mathbf{K}$ is upper triangular:

$$\mathbf{K} = \begin{bmatrix} f_x & \gamma & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \quad ,$$

where the x and y focal lengths are indicated as f_x, f_y (they are generally the same); while c_x, c_y represents the x and y coordinates of the optical center in the image plane. A common approximation is to use the center of the image. Lastly, γ is the skew between the axes, usually set to 0.

The calibration process aims to determine the 3×3 matrix $\mathbf{K}$, the 3×3 rotation matrix $\mathbf{R}$, and the 3×1 translation vector $\mathbf{t}$ using a known set of 3D points

(X_w, Y_w, Z_w) and their corresponding image coordinates (u, v) [25]. Once these values are acquired, the camera is considered calibrated.

A more detailed explanation can be found at [25], [26], [27], [29].

5.1.3 Analytical extrinsics calculation

Given the known geometry of the setup that was previously introduced and defined, it becomes feasible to analytically calculate the theoretical extrinsic parameters associated with the camera positions. These parameters serve as ideal baselines for evaluating the effectiveness of calibration. The goal is to position the cameras in the actual setup as accurately as possible through this comparative assessment.

The extrinsic matrix corresponds to a rotation matrix, which is a type of transformation matrix that describes the rotation of an object or a vector in a fixed coordinate system. In a 3D space is a 3x3 matrix: each column represents one of the axes of the reference system. The rotation matrix is also orthogonal, signifying that its transpose is equal to its inverse, and this implies that rotation is a linear transformation without deformation.

Considering a generic object P positioned in the (X, Y) plane, with coordinates p_x and p_y, the rotation matrix is able to describe its rotation in the counter-clockwise direction by an angle θ as shown in Figure 5.3:

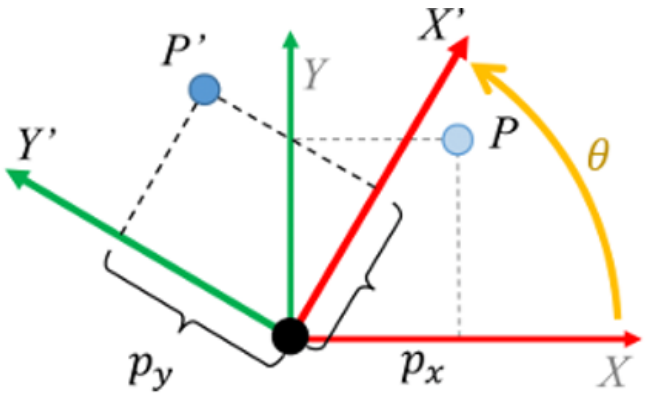


Figure 5.3: Rotation defined as a geometric transformation [r1]

where (X', Y') represents the new coordinates after the rotation. The rotation matrix operates on a vector, which maintains fixed coordinate axes after the transformation. Moreover, it is always a square matrix with real entities, meaning that it will always have an equal number of rows and columns [30].

Therefore, given a known angle θ, it is possible to rotate a vector around the three axes with the following matrices:

$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta) & -\sin(\theta) \\ 0 & \sin(\theta) & \cos(\theta) \end{bmatrix} \quad , \quad R_y(\theta) = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_z(\theta) = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad .$$

R_x is known as *Roll*, and it defines the counterclockwise rotation of θ around the x-axis; R_y is also referred as *Pitch*, and it represents the rotation of θ around the y-axis; R_z is called Yaw, and it is the rotation of θ around the z-axis [30]. Thus, within the circular camera array, assuming θ represents the angle between the x-axis and the first camera, it is possible to identify each camera as rotated around the y-axis with the angles shown in the following Figure 5.4:

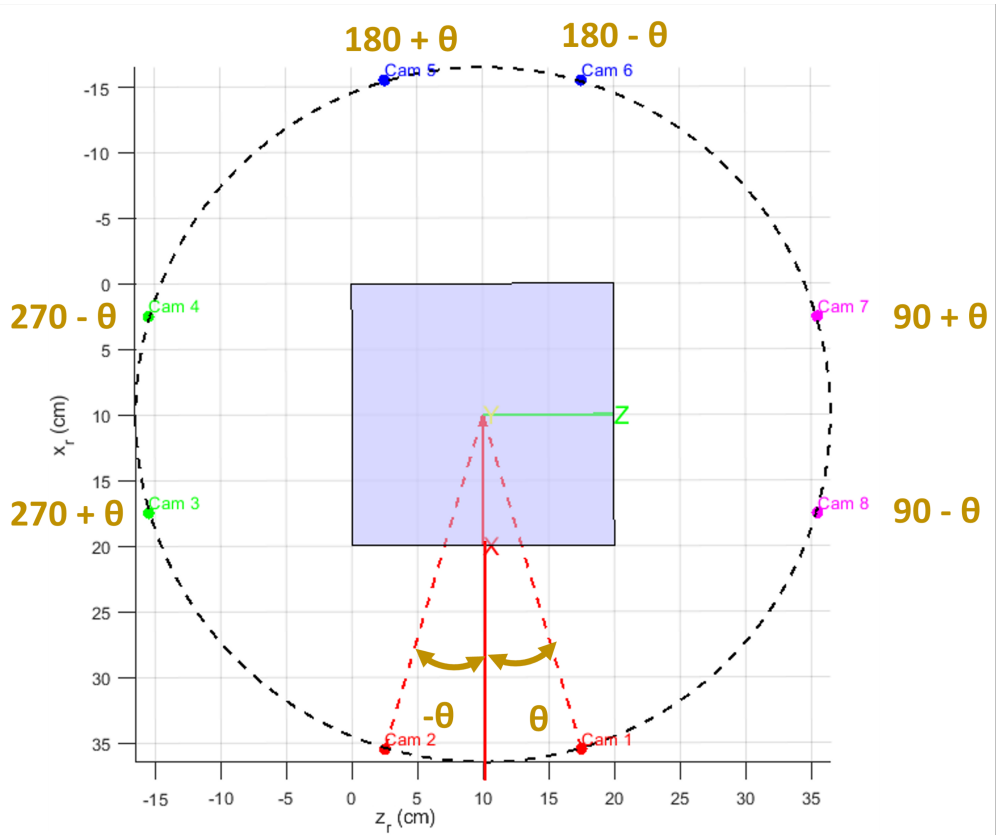


Figure 5.4: Camera angles in degrees (counterclockwise notation) [31]

It is now possible to determine the rotation matrices of each camera.

5.2 Multi-camera calibration technique

The calibration procedure is a critical step for the reconstruction of a three-dimensional object from two-dimensional information (such as images), which defines both measurement accuracy and scale [32]. State-of-the-art calibration techniques usually employ a checkerboard pattern with known geometric dimensions as the calibration target, due to its simplicity. This pattern's regularity and high contrast enable precise computation of corner features with subpixel accuracy and straightforward estimation of the checkerboard's 3D pose. There are several consolidated methods for automated checkerboard detection in images [33], even suited for multi-camera systems, which generally rely on a fundamental assumption: that there is spatial overlap between the cameras' fields of view so that specific points are visible for all cameras simultaneously. Then, the overlapping image portions are used to link the cameras' internal poses together through epipolar geometry. This is shown in Figure 5.5:

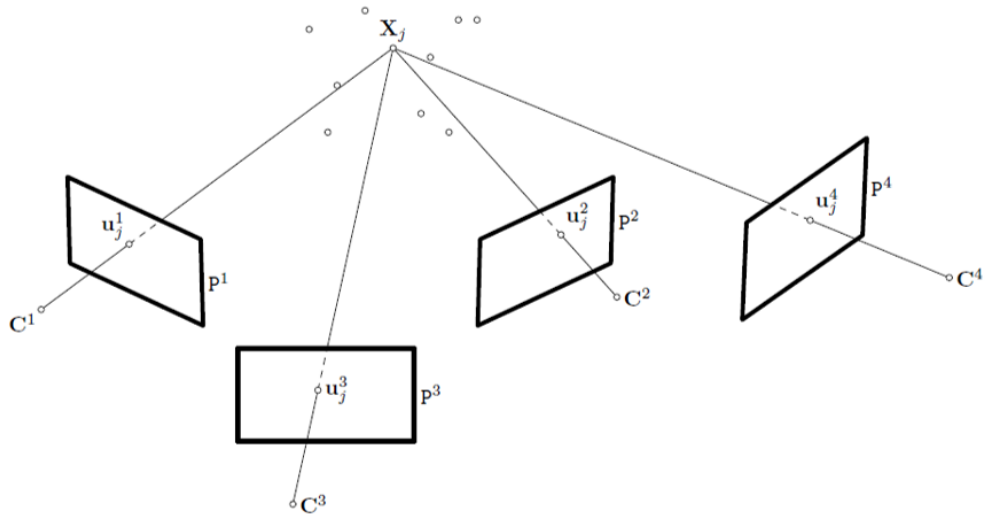


Figure 5.5: A four camera camera system able to observe a common point [34]

However, this approach proves to be inadequate for rigidly coupled cameras in circular or spherical imaging systems, due to the nonlinear orientation of cameras toward the checkerboard. In such setups, a feature point detected by one camera may not be visible to its opposing or orthogonal counterpart. Consequently, shared

feature points between opposing viewpoints are lacking due to limited viewing orientations, especially if there exists an angle gap of more than 90° between two cameras [35]. As a result, opposite cameras observing an identical planar pattern view distinct objects due to their non-overlapping fields of view, hindering a unified calibration approach. Moreover, establishing correspondences across multiple cameras with non-overlapping views becomes challenging, as it requires the detection of common features to determine the relative transformations between rig cameras, alongside each camera's intrinsic parameters [36].

This situation necessitates partial calibration structures to be connected through a Part-by-Part or Multi-Step Calibration method, which has the disadvantage of being time-consuming and prone to errors.

5.2.1 Multi-step calibration process

The most efficient procedure is to calibrate all cameras against one primary camera, with a calibration object concurrently visible to all. However, in the case of a circular camera array, multiple calibration steps become necessary, by repositioning the checkerboard pattern to ensure visibility from all cameras [37]. The process then requires the computation of a transformation matrix from one camera pair to the next within the sequence. Assuming multiple cameras designated from 1, 2... 8, the multi-camera calibration procedure unfolds across/involves the following three stages:

1. **Image acquisition:** the checkerboard pattern is positioned in front of a camera pair, inside of a tank filled with the tested fluid. A certain number of images is then taken (in the present case, 20 images per camera) before rotating the checkerboard by 90° and repeating the procedure with the next camera pair. Therefore, this sequence is carried out four times, as can be seen in Figure 5.6;

2. **Intrinsic and extrinsic parameters computation:** A dedicated code has been developed to compute the parameters of all 8 cameras, using the `Matlab Camera Calibration Toolbox` by *J.V. Bouguet* [38] as starting core and implementing appropriate geometric transformations on it. The algorithm, as illustrated in Figure 5.7, first computes an average intrinsic matrix alongside individual extrinsic matrices for each camera. The corners on the

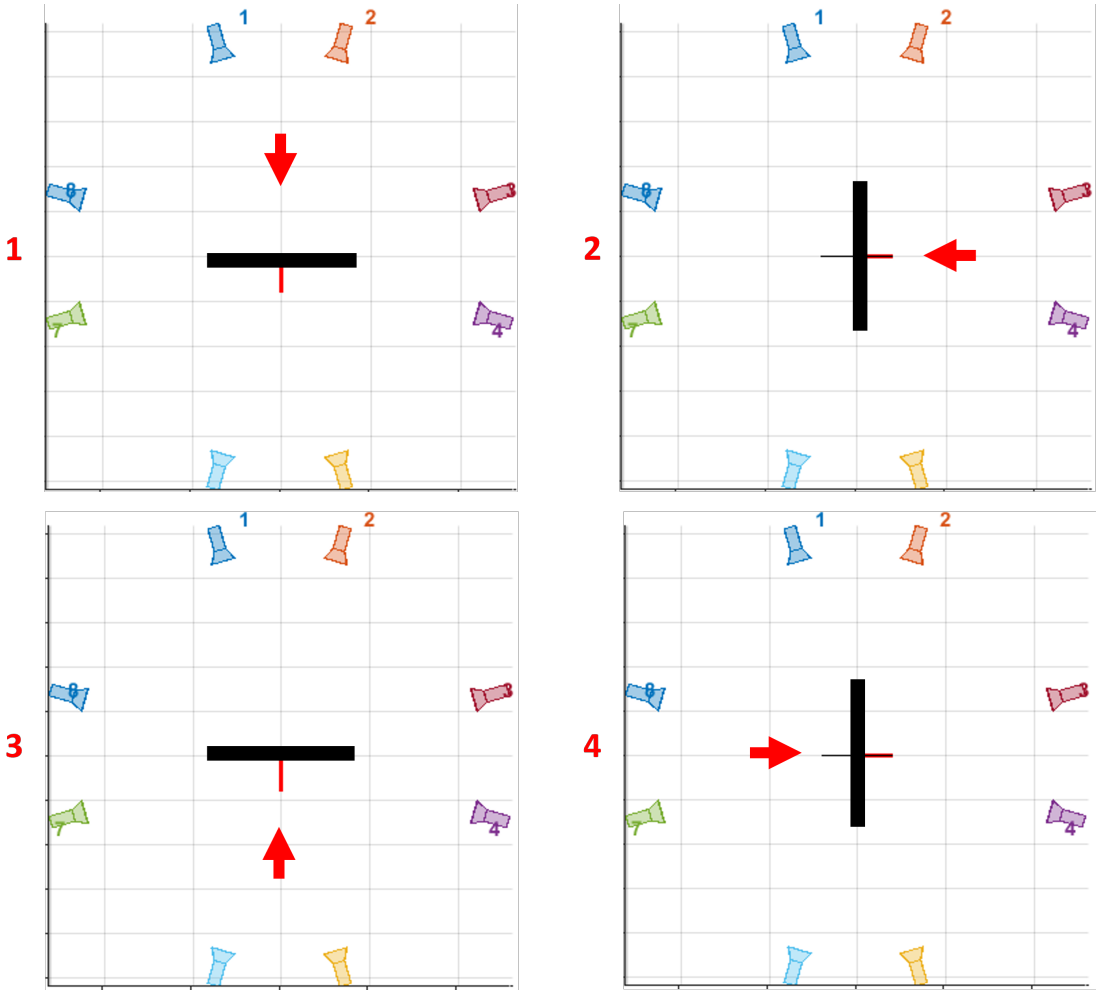


Figure 5.6: Multi-step image acquisition: the checkerboard (black object in the image) is initially positioned in front of the first camera pair and then rotated by 90° each time, for a total of four sequential steps

calibration target are detected using the `detectPatternPoints()` function, executed for every image in the calibration process. This produces a set of points for every camera, each point corresponding to a corner on the chessboard pattern. Using this data and an array containing distance values in millimetres corresponding to the pattern's square size, the cameras can be calibrated. The intrinsic and extrinsic parameters are calculated using the function `estimateCameraParameters()`; this routine provides the rotation matrix and translation vector for every pattern view, and a reprojection error [39]. For this process it is assumed that the cameras share identical intrinsic parameters, which is a reasonable hypothesis since they are all the same model;

hence, even if they present small differences between each other, they can be neglected;

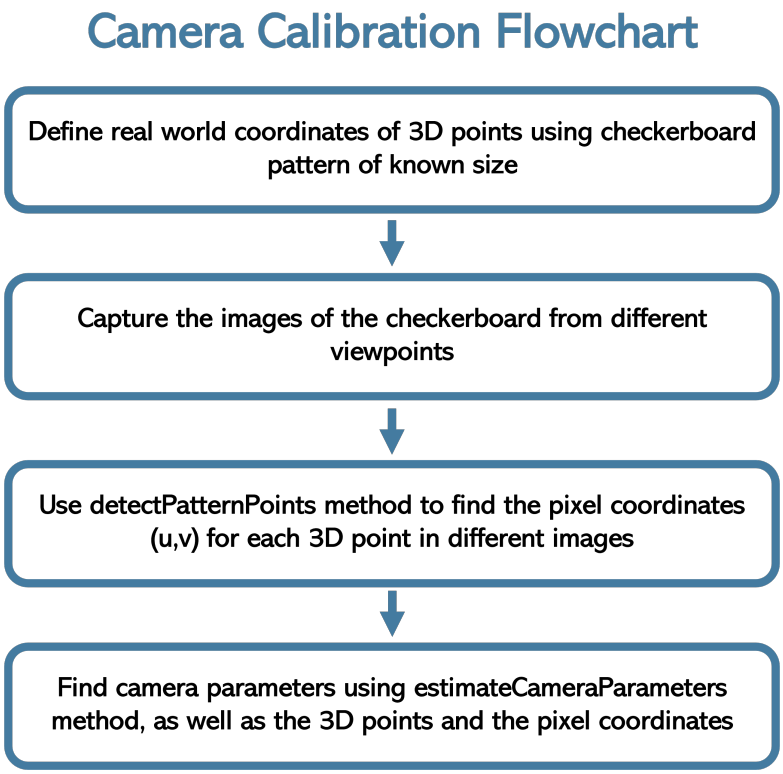


Figure 5.7: Matlab Camera Calibration Flowchart

3. **Geometric transformation:** Two cameras are selected as calibration reference; in the considered case, Camera 1 and Camera 2 were chosen. Subsequently, adjustments are needed for the viewpoints of the remaining three pairs. As the calibration computation assumes the center of the checkerboard as the origin of the reference system, if the pattern is positioned in the center of the setup, the geometric reference system will coincide with the camera's one. Hence, to rotate the viewpoints is sufficient to add a transformation matrix to the extrinsic matrix of each camera pair, representing a rotation around the y-axis from the first pair. The extrinsic matrices, represented in its compact notation as rotation vectors, appear as follows:

$$R_{cam1} = \begin{bmatrix} r_{x,1} \\ r_{y,1} \\ r_{z,1} \end{bmatrix} \quad , \quad R_{cam2} = \begin{bmatrix} r_{x,2} \\ r_{y,2} \\ r_{z,2} \end{bmatrix} \quad ,$$

$$R_{cam3} = \begin{bmatrix} r_{x,3} \\ r_{y,3} \\ r_{z,3} \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \quad , \qquad R_{cam4} = \begin{bmatrix} r_{x,4} \\ r_{y,4} \\ r_{z,4} \end{bmatrix} - \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \quad ,$$

$$R_{cam5} = \begin{bmatrix} r_{x,5} \\ r_{y,5} \\ r_{z,5} \end{bmatrix} + \begin{bmatrix} 0 \\ \pi \\ 0 \end{bmatrix} \quad , \qquad R_{cam6} = \begin{bmatrix} r_{x,6} \\ r_{y,6} \\ r_{z,6} \end{bmatrix} + \begin{bmatrix} 0 \\ \pi \\ 0 \end{bmatrix} \quad ,$$

$$R_{cam7} = \begin{bmatrix} r_{x,7} \\ r_{y,7} \\ r_{z,7} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \quad , \qquad R_{cam8} = \begin{bmatrix} r_{x,8} \\ r_{y,8} \\ r_{z,8} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{\pi}{2} \\ 0 \end{bmatrix} \quad ,$$

The obtained calibration parameters stay true only if the position, orientation, and focal settings of the cameras remain unchanged. This means that for static setups, where cameras are rigidly fixed in their positions, the calibration needs to be conducted only once, before all the acquisitions.

Chapter 6

Technique Validation

Before delving into the examination of the slush substances, it is imperative to validate the methodology and the code, ensuring any necessary adjustments for optimal performance in the specific scenario.

The process began with the calibration of the multi-camera system, enabling the acquisition of camera parameters. Afterwards, observation of an object with known dimensions was conducted to have a first assessment of the technique's effectiveness. Only after completing these steps, it was possible to proceed with the actual experiment.

6.1 Setup Calibration Results

A calibration checkerboard was used, fixed onto a metal plate with a height of 10 *cm*; thus, when submerged in the tank, the origin of the geometric reference system (the checkerboard center) would have a height of exactly 11 *cm*. The checkerboard features are listed in Table 6.1.

The cameras were manually positioned to align with the checkerboard center; afterwards, image acquisition was performed, following the procedure described in Chapter 5, which involves sequential rotations of the checkerboard. The obtained images were then fed into the algorithm, which extracted the intrinsic and extrinsic parameters of the cameras. The first results are shown on the following page, and visualised in Figures 6.1:

Checkerboard	
Checker Size	10 mm
Length	100 mm
Height	50 mm
Squares dimensions	10x5
Internal verticies	9x4

Table 6.1: Checkerboard features

Intrinsic Matrix:
$$\begin{bmatrix} 928.78 & 0 & 805.08 \\ 0 & 940.12 & 634.16 \\ 0 & 0 & 1 \end{bmatrix}$$

Radial Distorsion: [−0.17 , 1.13]

Tangential Distorsion: [0 , 0]

Camera	**Translation vector** $[mm]$	**Rotation vectors** $[rad]$
Cam 1	[-9.6256 , 2.9298 , 136.5509]	[-0.0486 , 0.1630 , 0.0016]
Cam 2	[7.7378 , 4.0619 , 137.2887]	[-0.0382 , -0.2107 , -0.0233]
Cam 3	[-10.3999 , -3.1700 , 135.1300]	[-0.0186 , 0.1714 , 0.0047]
Cam 4	[4.0274 , 1.7596 , 135.5294]	[-0.0177 , -0.1906 , -0.0271]
Cam 5	[-8.7401 , 4.0238 , 136.5777]	[-0.0480 , 0.1716 , -0.0044]
Cam 6	[6.1530 , -4.4746 , 135.1361]	[-0.0008 , -0.1999 , -0.0253]
Cam 7	[-10.3520 , -0.5580 , 132.2312]	[-0.0242 , 0.1232 , 0.0056]
Cam 8	[3.4595 , 3.0591 , 134.6773]	[-0.0222 , -0.2002 , 0.0158]

Table 6.2: First calibration results

The translation and rotation vectors presented in Table 6.2 represent the outcome of the calibration process. These vectors describe the position and orientation of each camera target's plane, from the camera point of view. The first two values in the translation vector represent the target's plane displacement on the x and y axis

from the origin of the reference system; for example, Camera 1 is observing a target positioned at (−9.6,2.9) *mm* from the origin. Then, the third value indicates the camera distance from its observed plane.

As can be seen, the results are rather far from expectations; specifically, there should ideally be no displacement on both the x and y axes and a distance of about 260 *mm* on the z-axis. Small deviations can be accepted, as long as the target planes intersect each other at the same point (which should approximately be the center of the reference system). However, it can be noted from Figure 6.1 that the planes do not intersect in the reference system, thereby leading to a need for process improvement.

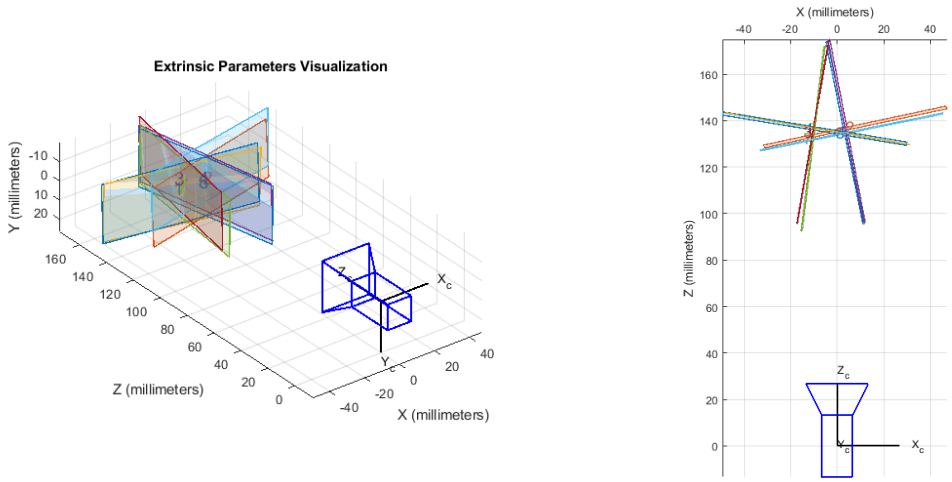


Figure 6.1: Extrinsic parameters visualisation. On the right, the top view of the camera target planes: the intersection does not occur in the center of the system

To reduce the displacement errors along the x- and y-axes, adjustments can be made by pre-processing the input images: the images were cropped, and the checkerboard's center was aligned with the image's center. The procedure is demonstrated in Figure 6.2, while the results of the second iteration are reported below and on the following page:

Intrinsic Matrix:

$$\begin{bmatrix} 1112.30 & 0 & 749.42 \\ 0 & 1130.10 & 516.34 \\ 0 & 0 & 1 \end{bmatrix}$$

Radial Distorsion: $[\, 0.038\ ,\ -0.64\,]$

Tangential Distorsion: [0 , 0]

Camera	**Translation vector** $[mm]$	**Rotation vector** $[rad]$
Cam 1	[−3.3200, 1.5931, 164.7334]	[−0.0524 , 0.1941 , 0.0041]
Cam 2	[−3.7512, 1.8464, 166.1583]	[−0.0345 , −0.2635 , −0.0270]
Cam 3	[−3.7200, 1.5565, 163.1124]	[−0.0370 , 0.2025 , 0.0013]
Cam 4	[−3.8020, 1.7323, 163.5824]	[−0.0208 , −0.2337 , −0.0278]
Cam 5	[−3.7523, 1.8119, 164.5766]	[−0.0470 , 0.2026 , −0.0012]
Cam 6	[−3.8435, 1.9813, 163.4270]	[0.0260 , −0.2483 , −0.0196]
Cam 7	[−3.3869, 1.9536, 159.7042]	[−0.0365 , 0.1424 , 0.0046]
Cam 8	[−3.8837, 1.3114, 162.4297]	[−0.0168 , −0.2448 , 0.0136]

Table 6.3: Improved calibration results

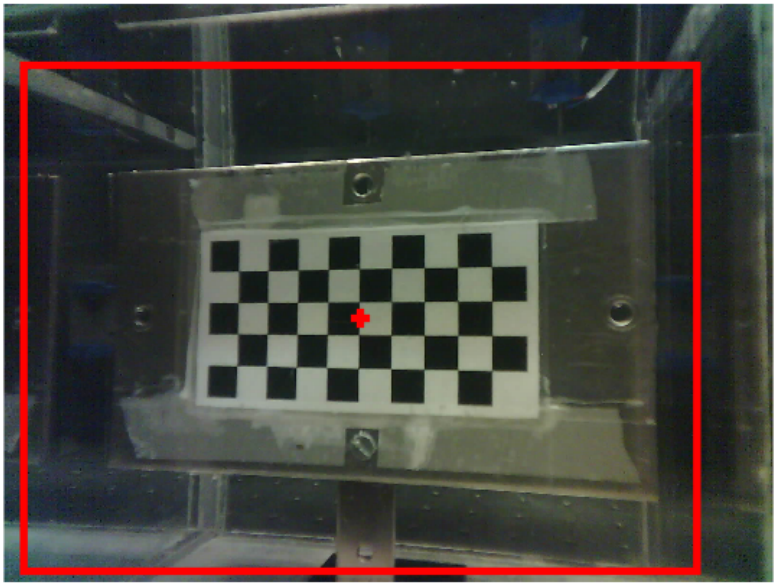


Figure 6.2: Input image pre-processing: the images were cropped, so that the checkerboard is perfectly centered in the image

The translation vectors now demonstrate notable improvement: the displacement between viewpoints measures less than 1 millimeter (approximately 0.8 mm) for both x- and y- axes. In addition, the camera's target plane is positioned almost perfectly in the reference system center, with a general distance of less than 4 mm

on the x-axis and less than 2 *mm* on the y-axis. There is also an improvement on the z-axis, although the calibration results are still far from the real positions. Nevertheless, the problem of accurately estimating the longitudinal distance from a pattern is a well-known issue in camera calibration; fortunately, it does not impact the efficacy of the Deterministic Visual Hull algorithm. Hence, it was decided to still employ the obtained results for the reconstruction purpose.

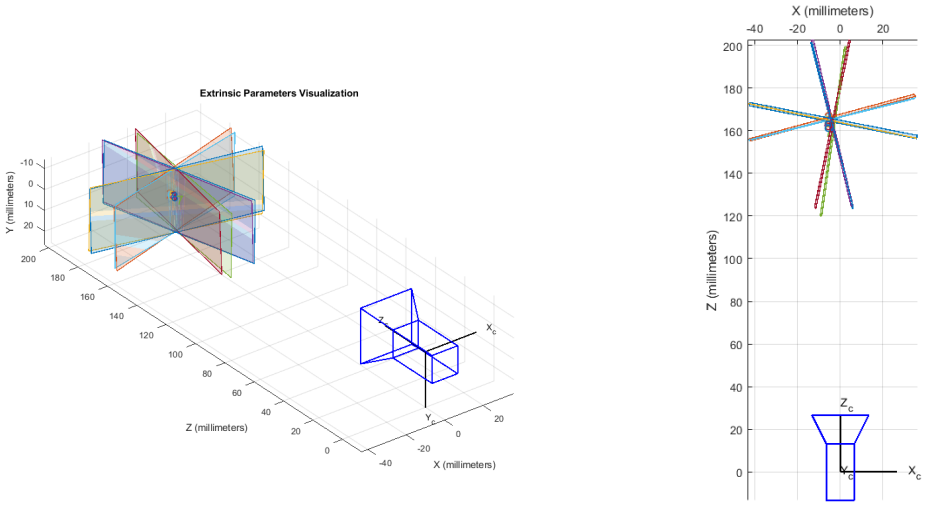


Figure 6.3: Extrinsic parameters visualisation, second iteration: all the planes are now intersecting each other almost perfectly. Minor inaccuracies still persist, but given their small magnitude, they were deemed negligible

The final computed camera positions are depicted in Figure 6.4, referred to as the geometric reference system (with the origin in the center of the circumference on which the cameras are positioned).

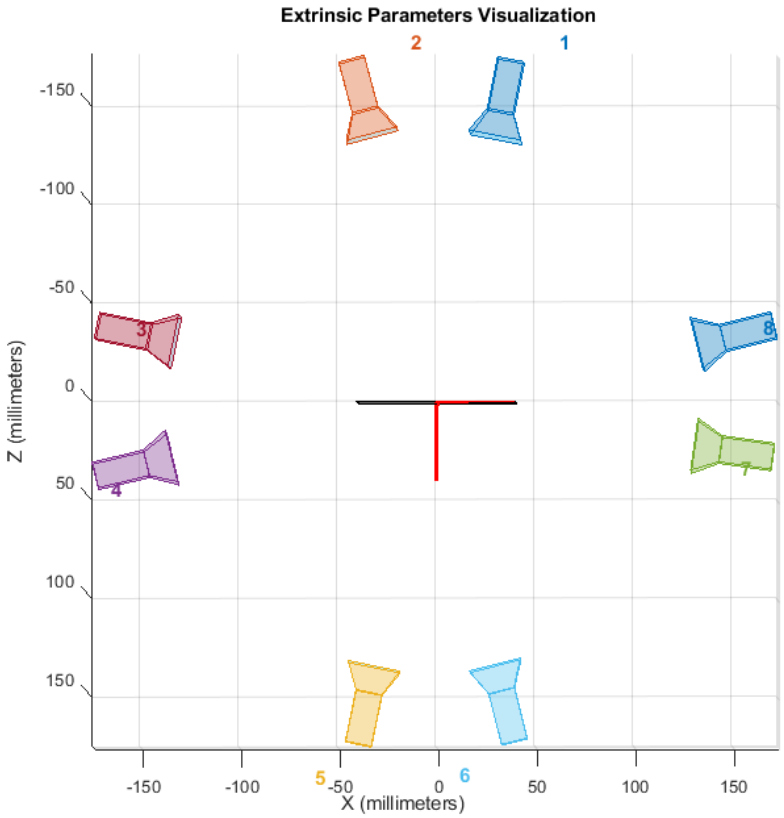


Figure 6.4: Reconstructed camera positions, with the use of the intrinsic and extrinsic parameters

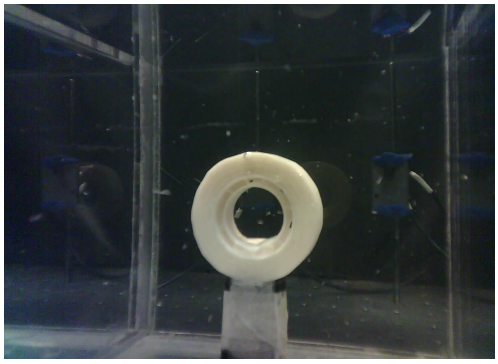
In summary, calibration holds a significant role in the process of 3D object computation from two-dimensional information. The accuracy of the provided data directly influences the output quality. Achieving satisfactory results is a challenging task, demanding precise control over the cameras' position and orientation, which might not always be feasible in all scenarios. In such cases, mitigating errors becomes imperative, to prevent error propagation in the later stages.

6.2 Algorithm Performance Verification

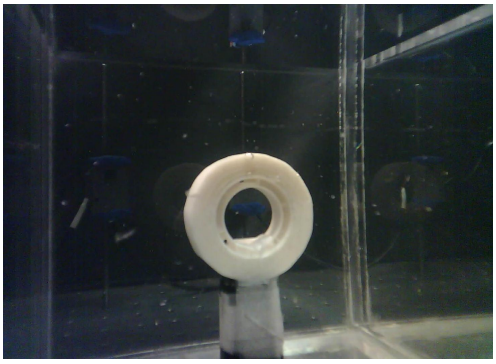
A first assessment of the algorithm's effectiveness is crucial for the successful execution of the project. To accomplish this, an object with known dimensions was employed as a target for observation. The chosen target, a scotch tape with a 5.5 *cm* diameter and 2.5 *cm* thickness (depicted in Figure 6.5), is readily available and its white colour provides a sharp contrast against the dark background, thus complying with the constraints of the Deterministic Visual Hull.

Figure 6.5: Target object for the algorithm validation

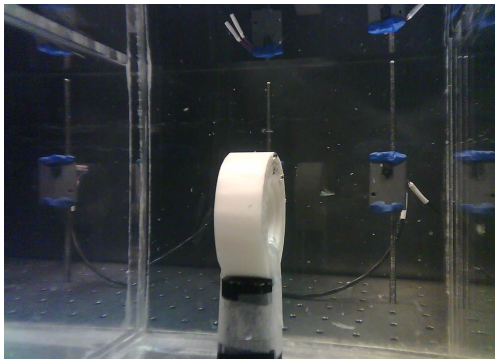
The object was thus secured on support, immersed in the plexiglass tank filled with water and positioned in its center. The acquired images are displayed in Figure 6.6.



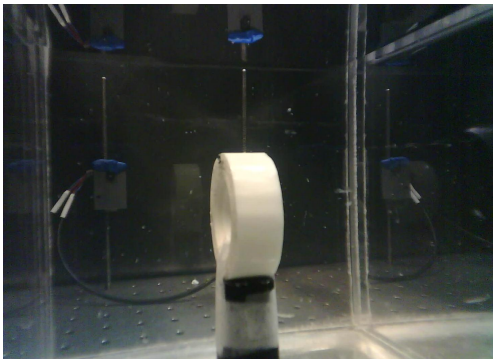
(a) Camera 1



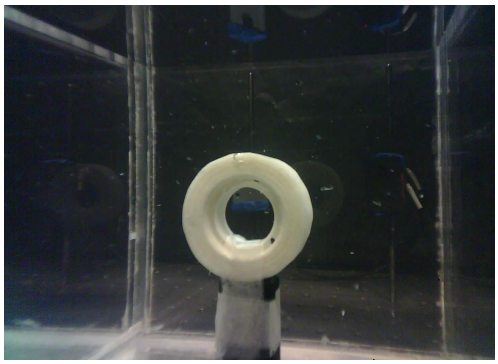
(b) Camera 2



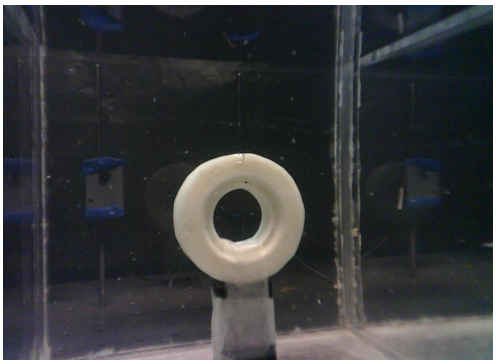
(c) Camera 3



(d) Camera 4

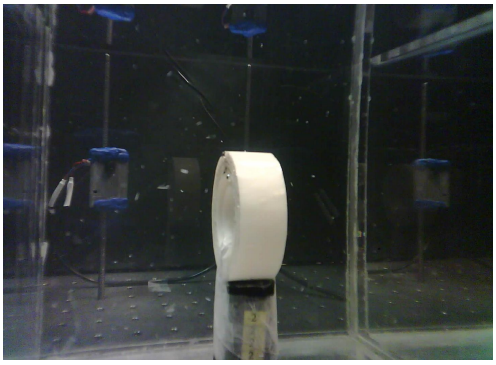


(e) Camera 5



(f) Camera 6

(g) Camera 7



(h) Camera 8

Figure 6.6: Original input images

6.2.1 Image Processing

Given the input images, the silhouettes are computed. This is the most important step for achieving a high-quality 3D reconstruction since the 3D shape information is extracted from the silhouettes. However, the initial silhouettes, as depicted in Figure 6.7, did not meet the desired quality standards: the target shape is not completely disconnected from the support; moreover, the presence of a high level of background noise may result in a faulty reconstruction. Conversely, an ideal silhouette must have precise segmentation of the observed target (i.e. the object shape must be separated from the other elements) with minimal background noise, particularly around the target's edges.

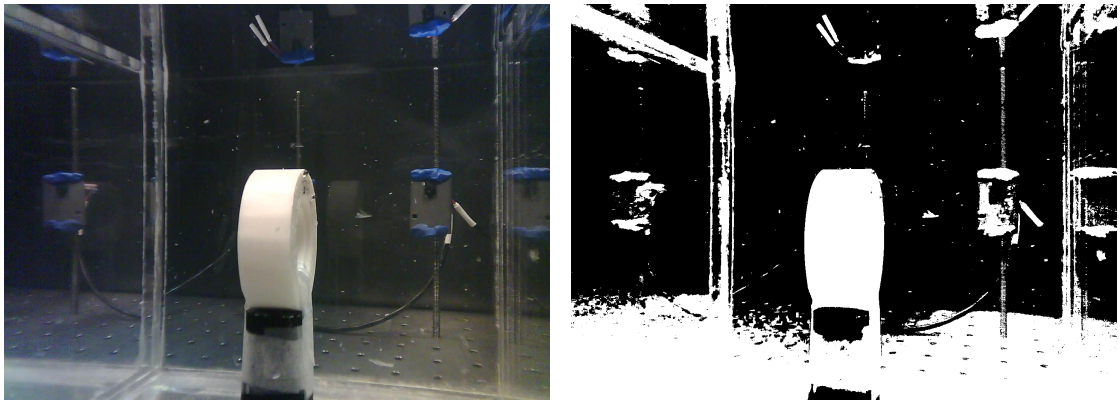


Figure 6.7: (Left) Original frame; (Right) An example of initial silhouette computation (right)

The production of higher-quality silhouettes was accomplished by integrating three distinct image processing methodologies within the code; these methods can be used sequentially or separately, as required from the test case, to ensure optimal results.

Contrast Enhancement

The first operation consists of contrast modification, through the `MATLAB` function `imadjust()`. This function manipulates the image intensity values, leading to an increased contrast in the output image. The operation enables a clearer distinction between the object and the background by intensifying the target silhouette's whiteness. Then, a slightly higher threshold is applied for silhouette computation, effectively reducing the background noise. This process aims to isolate the target

shape within the silhouette. However, the main drawback of this approach is the contrast intensification for sufficiently white background elements, making it insufficient to completely eliminate noise. Consequently, alternative methods must be incorporated to address this limitation.

Another useful operation is to convert a coloured image into a black-and-white format using the function `rgb2gray()`. This transformation facilitates the silhouette creation by converting all the colour pixel values to black and white; consequently, pixels with colour values are close to darker shades and thus excluded, as each pixel value is compared to the threshold.

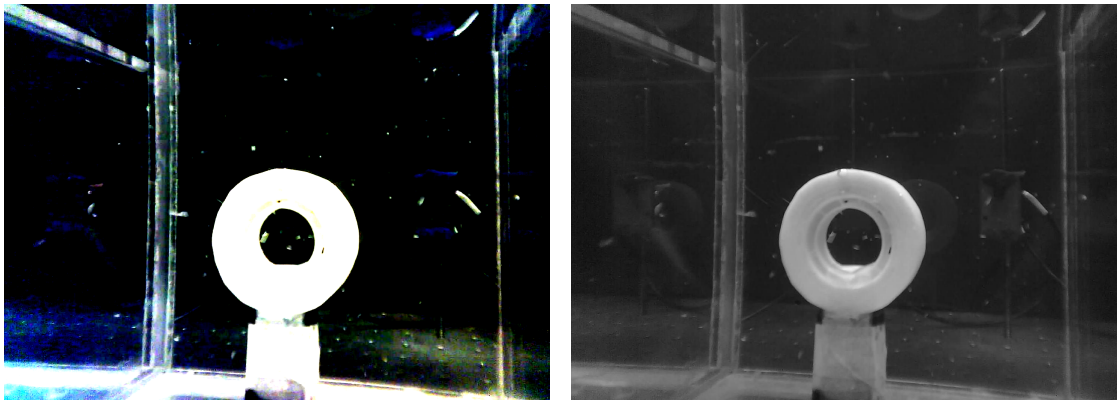


Figure 6.8: Examples of image processing operation: contrast enhancement (on the left), and greyscale conversion (on the right)

Background Removal

Another relevant operation is background removal, with the `MATLAB` function `imsubtract()`. Specifically, given two input arrays X, Y, this function operates by subtracting each element in Y from the corresponding element in X and provides the difference in the corresponding element of an output array Z. Assuming that the input arrays X and Y are numeric arrays of the same class and size, the resulting array Z maintains the same size and class of the inputs. Since an image is an array of pixels, each with a specific value, this function enables the subtraction of two images. The results are illustrated in Figure 6.9.

On the other hand, this procedure may not always consistently succeed. When dealing with objects immersed in a fluid, slight alterations in background details might occur throughout the acquisition process, which might compromise the operation's effectiveness.

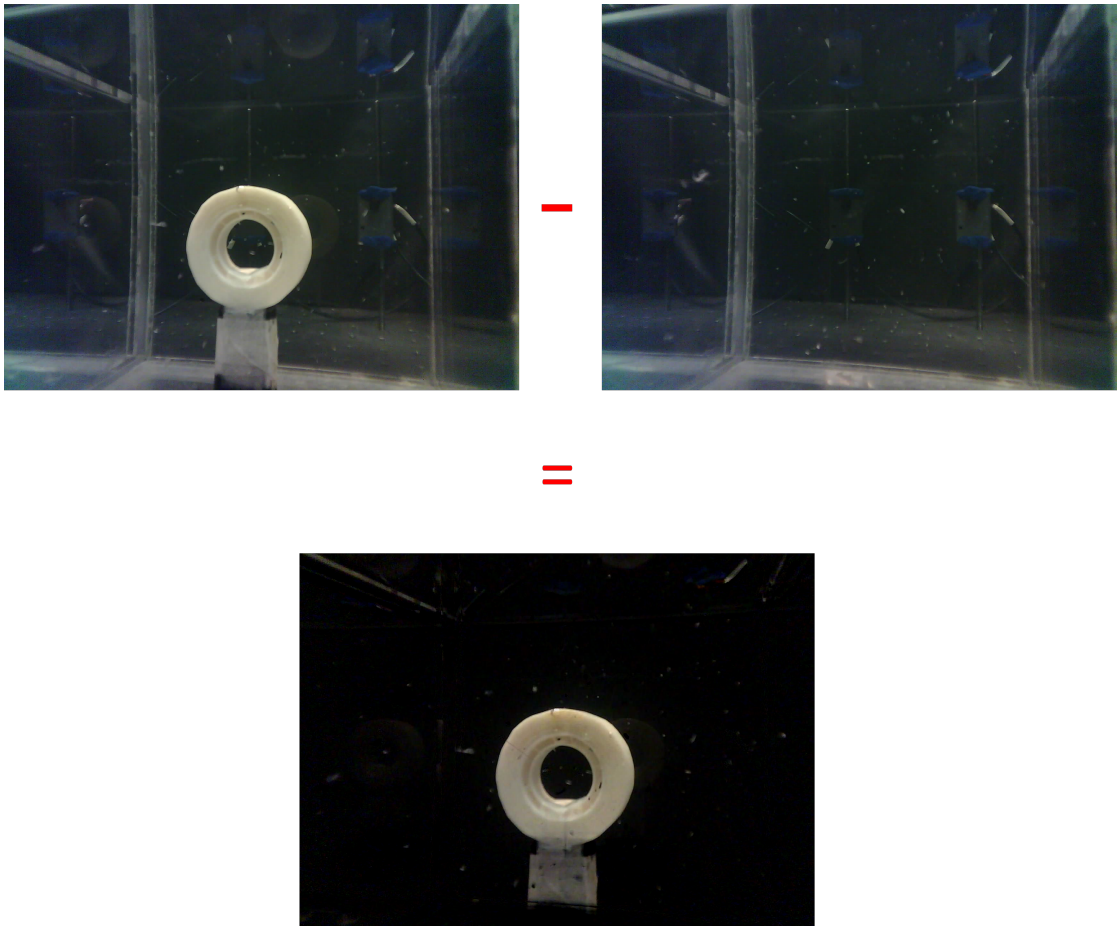


Figure 6.9: Background removal demonstration: the second image is subtracted from the first, and the resulting image only contains the target object

Largest Blob Extraction

The last operation is the most complex one, but also the most efficient. Specifically, given an input binary image (i.e. a silhouette), it is possible to extract the largest area composed of white pixels: the algorithm recognises a *'blob'* area, by counting the white pixels connected to each other. If a white pixel is surrounded by one or more white pixels, it is categorised as part of a blob area; otherwise, it is considered background noise and removed. In the next step, the blob's properties are determined, retaining only the greatest area while discarding the others. This strategical elaboration enables the isolation of the object's silhouette from the background noise.

Further refinement can be implemented when the object possesses a known geometry, such as rectangular, circular, or spherical shapes. In such cases, it is

possible to select the largest blob area that respects a certain defined aspect ratio and discard the others. This optimization allows the algorithm to consistently choose the circular blob that matches the input data, even if larger blobs (like the floor silhouette) exist within the silhouette.

the operation structure can be summarised as follows:

1. **Image reading and thresholding:** The image is read by the code and the silhouette is computed. This step also includes any additional pre-processing operations (if required);

2. **Connected component labelling:** using the function `bwlabel()`, all the connected regions in the binary image are identified and labelled;

3. **Region properties measurement:** properties calculation (e.g., area, centroid, major and minor axis lengths) of the labelled connected components, through the function `regionprops()`

4. **Shape filtering:** computes the aspect ratio of each labelled component and removes regions that do not match the expected shape (in this case, ellipsoidal);

5. **Minor blobs removal:** the `bwareaopen()` function removes small connected components based on their area. It depends on the threshold (higher thresholds allow for larger blobs to be retained);

6. **Silhouette update:** it creates a new silhouette with the resulting image.

An example of the process is displayed in Figure 6.10. Nonetheless, the major drawback of this approach is the heavy dependence on the user's intervention on the setting parameters, thus making it a considerably time-consuming procedure.

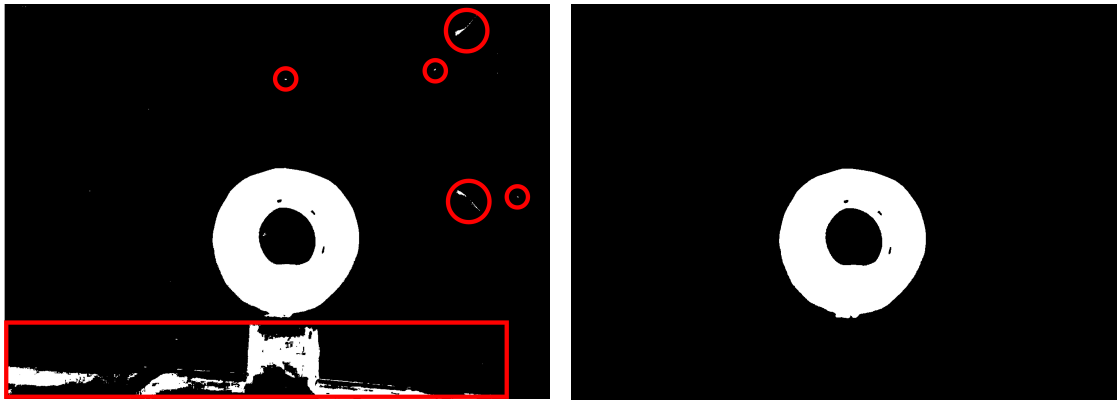
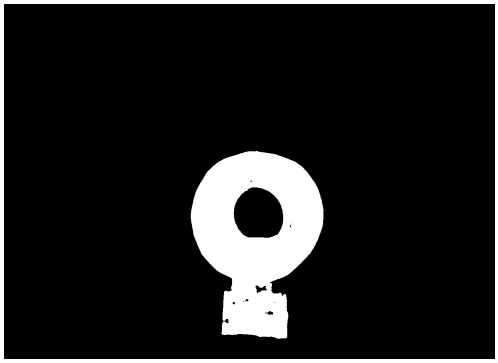


Figure 6.10: Blob extraction example: on the left, all the unwanted blobs are highlighted. The algorithm thus proceeds to remove those blobs from the silhouette; the result is shown on the right, where the target's silhouette is the only shape left in the image

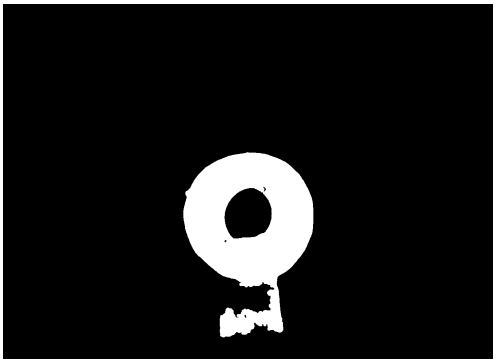
6.2.2 Silhouettes

The silhouettes are obtained by converting the colour images to binary images, which are 2D matrices with elements of 0 and 1 corresponding to black and white pixels, respectively. The input image is replaced by a binary mask (the silhouette), where pixels that belong to the object projection (white pixels, with value 1) are separated from those in the background (black pixels, with value 0). This technique is called *Foreground Segmentation*.

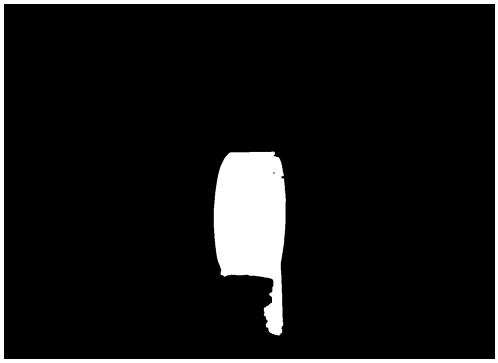
For the considered test case, a simple background removal was applied as pre-processing. It can be seen that in some images, a portion of the support is included in the silhouette; this occurrence does not pose any issue, as not all the images contain the same segment. Hence, during the silhouette intersection process, any parts not present in all eight images will be excluded from the final reconstruction. The silhouettes are displayed in the Figure 6.11.



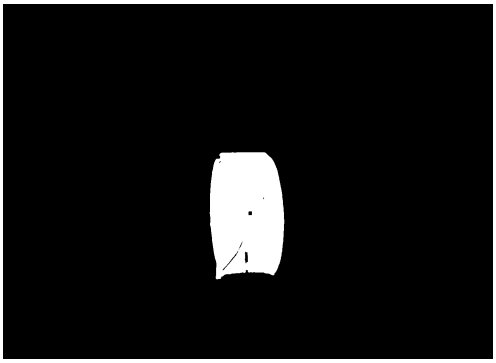
(a) Silhouette from Camera 1



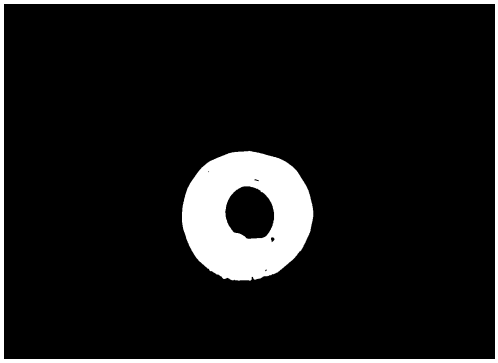
(b) Silhouette from Camera 2



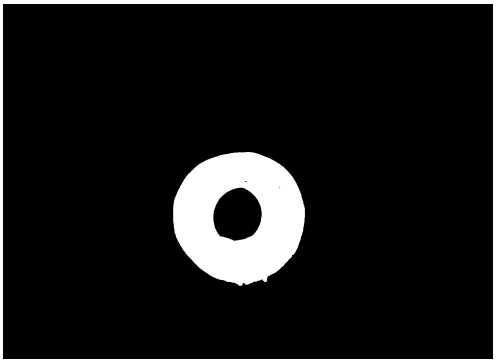
(c) Silhouette from Camera 3



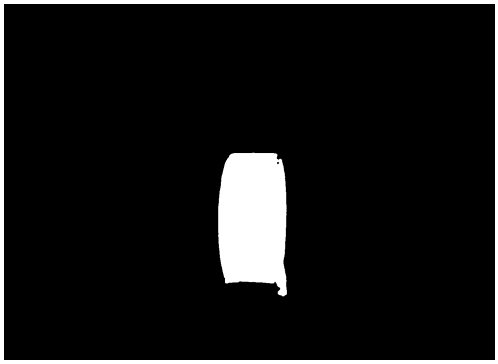
(d) Silhouette from Camera 4



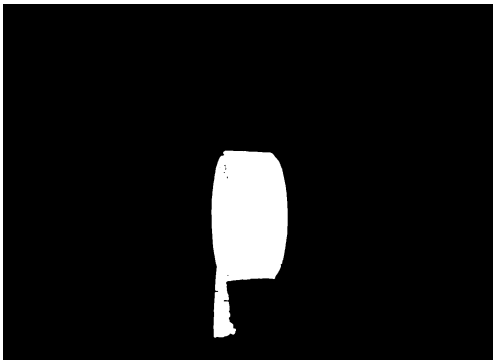
(e) Silhouette from Camera 5



(f) Silhouette from Camera 6



(g) Silhouette from Camera 7



(h) Silhouette from Camera 8

Figure 6.11: Silhouettes

6.2.3 3D Reconstruction

The final reconstruction is highly dependent not only on the silhouettes but also on the voxel grid definition. The bounding box dimensions are set prior to the reconstruction, thus determining the volume in which the object will be reconstructed; then, that space is divided into $100 \times 100 \times 100$ voxels, effectively creating the voxel grid. consequently, depending on the reconstruction dimensions, a large voxel grid may introduce a certain level of approximation when reconstructing small objects. A first reconstruction has been computed with a large voxel grid, to ensure that the object is contained in it. The result is displayed in Figure 6.12, while the voxel grid dimensions are listed below:

$$
\begin{aligned}
xlim &= [-200 \quad 200] \\
ylim &= [-200 \quad 100] \\
zlim &= [-200 \quad 80]
\end{aligned}
$$

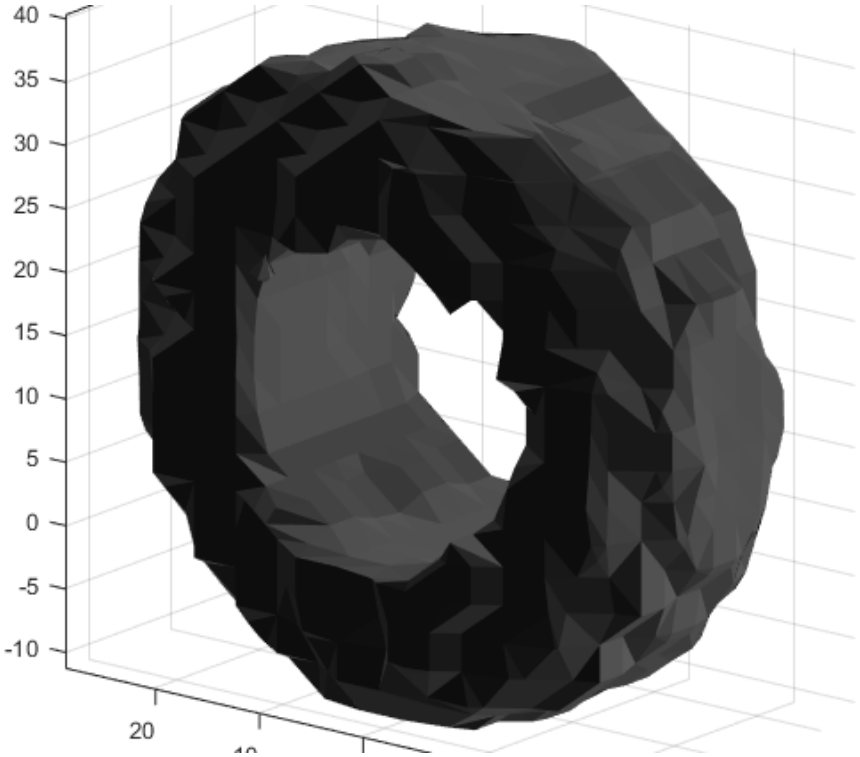


Figure 6.12: 3D reconstruction of the target in a large voxel grid. The level of shape approximation is still high; nevertheless, the cylindrical shape can be recognised

However, the reconstructed object dimensions are about $6 \times 6 \times 3.5$ *cm*; even if the shape was correctly created, it still presents a high level of approximation. Therefore, by reducing the voxel grid to the minimum, a more detailed reconstruction can be obtained, as shown in Figure 6.13. The bounding box is now defined as:

$$xlim = [-20 \quad 40]$$
$$ylim = [-40 \quad 20]$$
$$zlim = [-20 \quad 15]$$

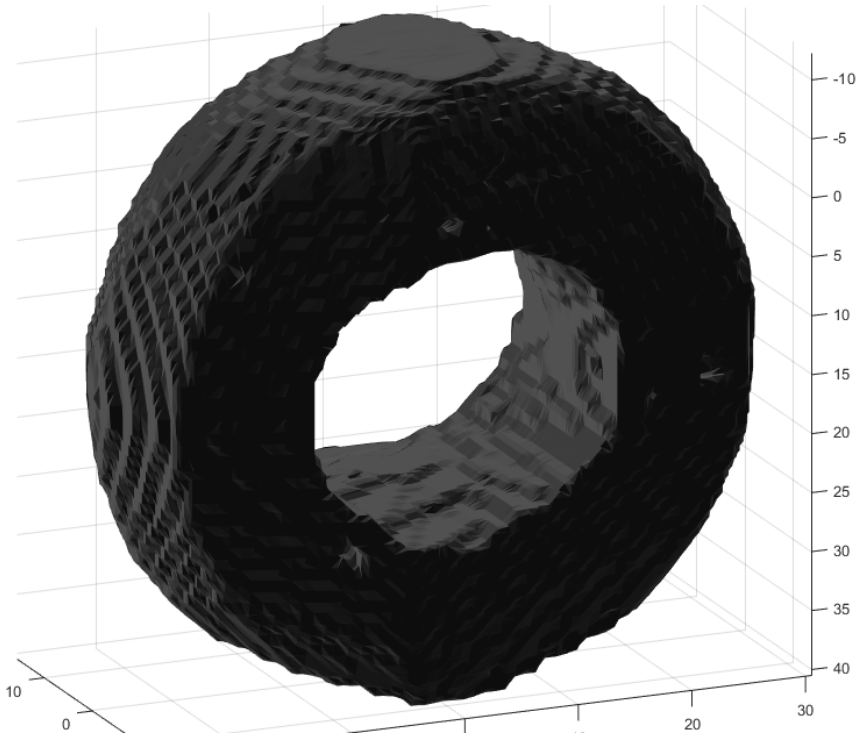


Figure 6.13: Final reconstruction

The reconstruction now features greater sharpness and details; its size is also quite similar to the real dimensions, except for the z-axis (which appears to be about 1.5 *cm* larger than the actual size). This deviation can be clearly seen in Figure 6.14, and is due to the Visual Hull limitation: the technique can only provide an upper bound approximated shape of the real object, thus largely depending on viewpoint number and position. For the present case, the constraining viewpoints are those provided by Cameras 3, 4, 7 and 8: their perspectives are not perfectly aligned with the object thickness, thus introducing an additional volume that does

not exist in reality.

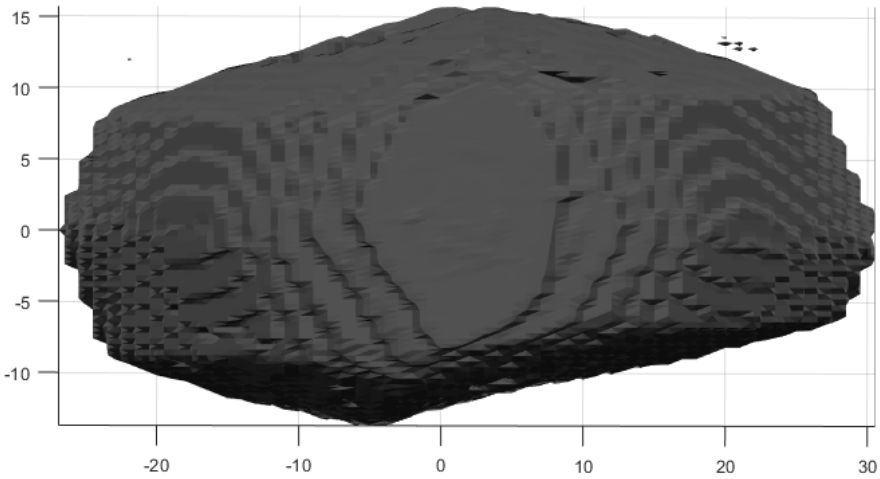


Figure 6.14: Final reconstruction detail: the thickness is highly dependent on the viewpoints

Nevertheless, the object was successfully reconstructed, (an upper bound approximation of it), and this constitutes a successful first step for the implementation of the technique on the considered matter. Further improvements can be certainly obtained by optimising the camera positions or increasing the camera number. However, these improvements will be left to future studies.

Chapter 7

Visual Hull Results

In this chapter, the experimental procedure is described and the results are presented and commented. The 3D solid object reconstruction with the Visual Hull technique was carried out for the three different cases, identified in Chapter 2: water, paraffin RT-5HC and paraffin RT-9HC. A Python code was specifically implemented to control all 8 cameras simultaneously, reported in Appendix A.1. Images of the melting mass were acquired about every 20 seconds (the cameras also take about 7 seconds for a complete acquisition). At the same time, temperature measurements were taken using `Labview`. The acquisitions continued until the frozen mass was completely melted; then, an additional photo was taken to capture the background (thus being able to perform the background removal on the input images). After that, the code was finally stopped and the measurements were interrupted.

Once the images have been acquired and the calibration completed, the reconstruction can be computed through the Deterministic Visual Hull. The code takes the intrinsic and extrinsic parameter values from a file '`.txt`', and generates the respective matrices and vectors. Subsequently, it proceeds to create the silhouette of the target for every image, extracting from the image the pixels with a value higher than a certain predefined threshold. After that, it reconstructs the 3D model by intersecting the voxelised silhouettes.

To sum up, the inputs provided to the code are:

- The intrinsic and extrinsic matrices for every image (which are determined through the calibration process, and have been saved in a '`.txt`' file);
- A different number of views of the object (for this study, 8 images).

Each experiment was repeated three times, in order to create a consistent database and verify repeatability of the experimental conditions and results. The details are summarised in the Table 7.1.

Substance	**Case name**	**Melting time** $[s]$	**Total images set**
Water	`W_1`	857	32
	`W_2`	789	28
	`W_3`	856	32
Paraffin RT5HC	`Par5_1`	1094	38
	`Par5_2`	1101	38
	`Par5_3`	1421	49
Paraffin RT-9HC	`Par-9_1`	704	22
	`Par-9_2`	663	22
	`Par-9_3`	750	22

Table 7.1: Test matrix of the performed experiments

It is worth noticing that only data from `W_1`, `Par5_1` and `Par-9_2` are presented in this thesis.

7.1 Slush State Water, Test W_1

The initial test case was conducted with water, which will serve as a preliminary scenario for predictive model verification. The tank was filled with water and spherical rubber moulds were used, producing ice spheres of about 5 cm in diameter. The initial shape of the ice is depicted in Figure 7.2: the shape appears mostly spherical, with minor imperfections on the upper hemisphere. Afterwards, the frozen volume (at -15 $°C$) was immersed in the water tank (initially at about 18 $°C$), thereby starting the experiment. The measured temperature evolution is illustrated in Figure 7.1:

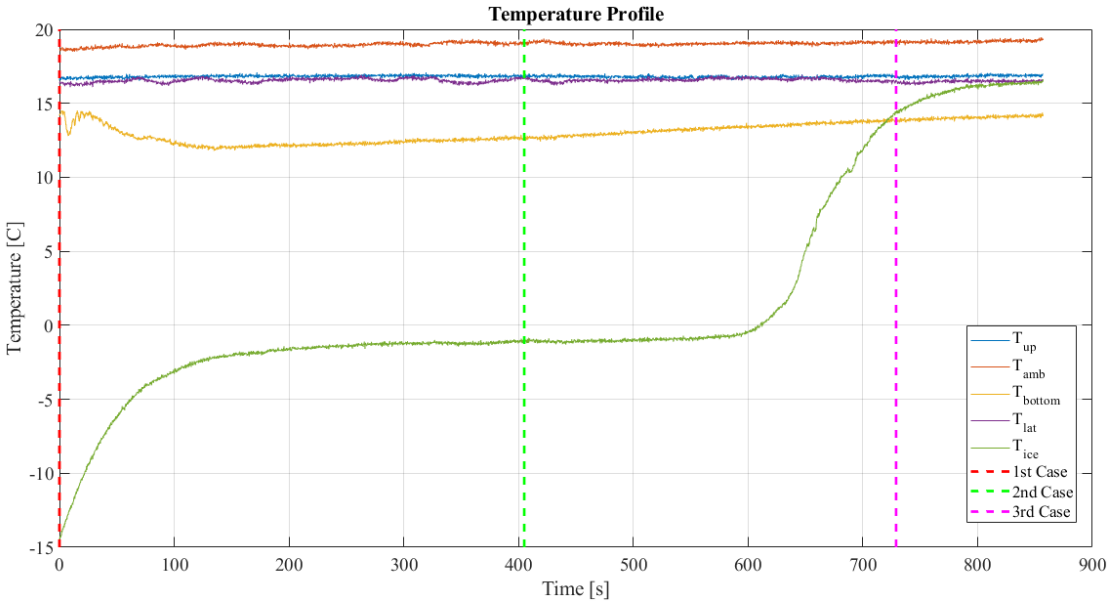


Figure 7.1: Temperature profile during the Test W_1

where T_{up} is the measured temperature on the water surface (thus on the upper part of the tank), T_{amb} the external environment temperature, T_{bottom} the temperature at the bottom of the tank, T_{lat} is the temperature of the tank lateral wall, and T_{ice} the temperature inside the ice. The ice temperature initially decreases, to reach a plateau after about 100 seconds: during this time interval, the ice core melts, until the ice volume becomes sufficiently small to expose the temperature sensor. From 600 seconds onward, the thermocouple is in direct contact with the water, thereby measuring an increasing temperature. The graph also shows a lower temperature at the bottom of the tank, if compared with the measurements taken at the top and side wall; this discordance is attributed to the downward movement

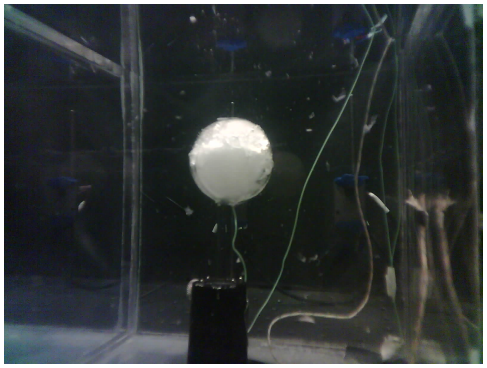
of the cold water mass.

The total duration of the test was 857 seconds (about 14 minutes), during which 30 sets of images were taken. Three sets of images are analysed in this paragraph, for the purpose of the volume reconstruction during melting: at t=0s (dashed red line in Figure 7.1), at t=405s (green dashed line) and at t=729s (magenta dashed line).

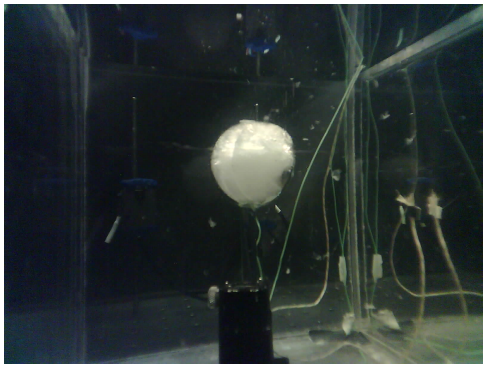
Figure 7.2: Test `W_1`: picture of the solid object at the beginning of the experiment.

7.1.1 Reconstruction at t = 0 *s*

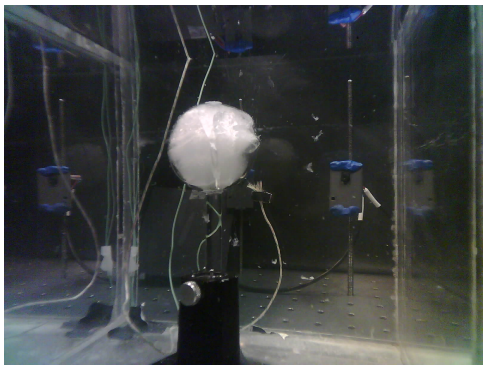
The initial batch of images captures the ice sphere in its original size, depicting its state before the melting process significantly alters its form. The input images and the silhouettes are reported in Figure 7.3 and in Figure 7.4 respectively. The silhouettes were obtained performing the following operations, in sequence: background removal; grey-scale image conversion; and largest blob extraction, as detailed in Chapter 6.



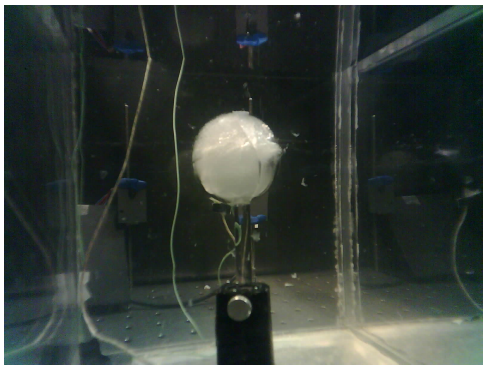
(a) Camera 1



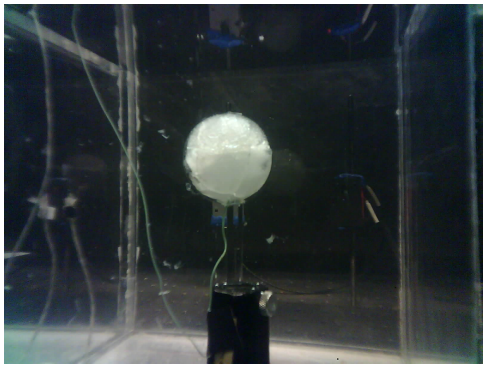
(b) Camera 2



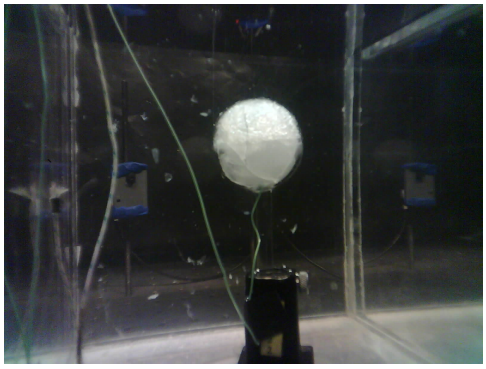
(c) Camera 3



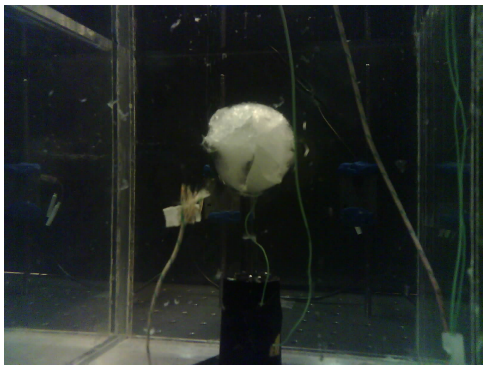
(d) Camera 4



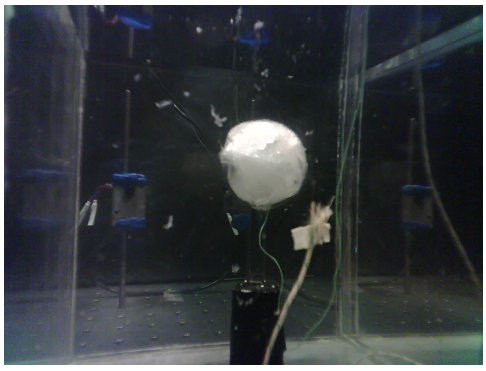
(e) Camera 5



(f) Camera 6

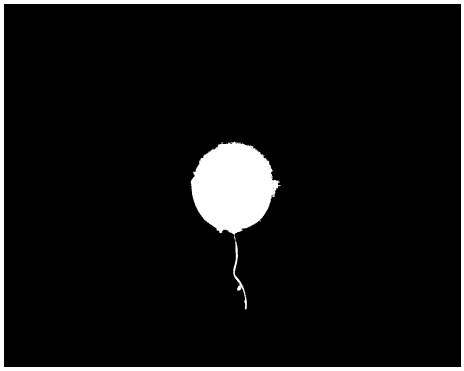


(g) Camera 7

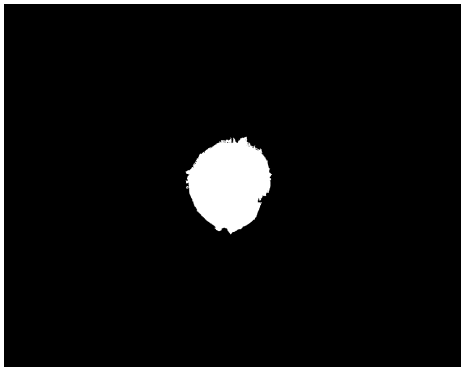


(h) Camera 8

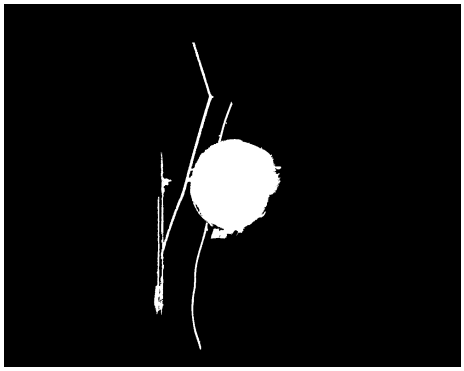
Figure 7.3: Input images for Test W_1 at t=0s



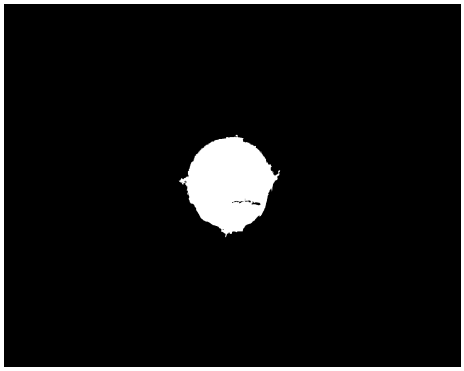
(a) Silhouette of ice from Camera 1



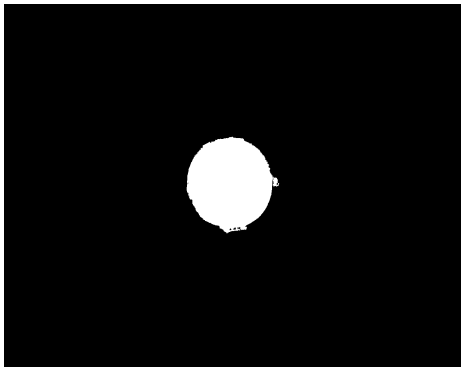
(b) Silhouette of ice from Camera 2



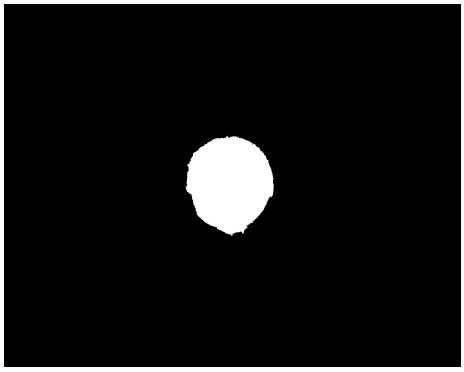
(c) Silhouette of ice from Camera 3



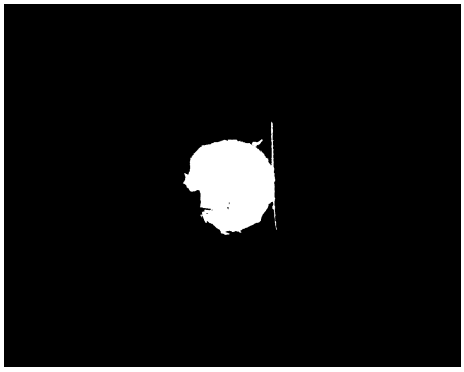
(d) Silhouette of ice from Camera 4



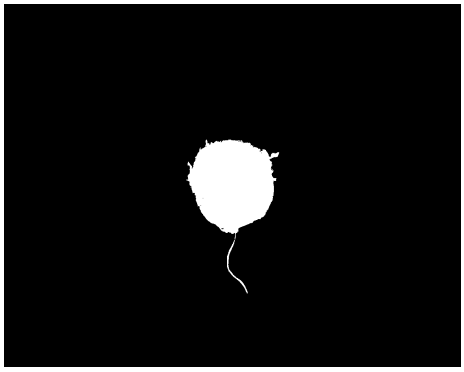
(e) Silhouette of ice from Camera 5



(f) Silhouette of ice from Camera 6



(g) Silhouette of ice from Camera 7



(h) Silhouette of ice from Camera 8

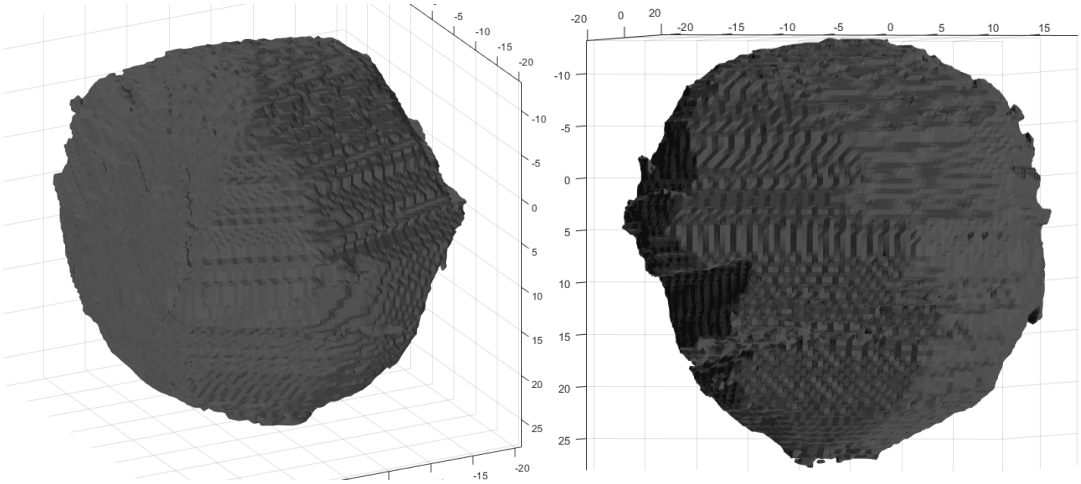
Figure 7.4: Silhouettes for Test `W_1` at t=0s

The silhouettes still contain some background details, but these do not affect the final reconstruction. The voxel grid, defined below, perfectly reflects the real dimensions of the ice sphere, which should have a diameter of approximately 5 cm:

$$xlim = [-25 \quad 25]$$
$$ylim = [-20 \quad 30]$$
$$zlim = [-25 \quad 25]$$

It is also possible to determine the volume occupied by the reconstruction within the considered voxel grid, which is about 89.18% of the total space.

The reconstruction is reported in Figure 7.5.



(a) (b)

Figure 7.5: Test W_1 at t=0s: lateral views of the reconstructed ice volume

The spherical shape is clearly recognisable, despite the upper bound approximation of the shape performed by Visual Hull (whose details are evident in Figure 7.6). A noteworthy observation is presented in Figure 7.7: the frozen sphere exhibits a concentrated white core and a relatively more translucent outer section (in contact with water). The ice shape could not be completely recognized in the transparent area; this led to incomplete silhouettes, hence resulting in a visible deviation of the final reconstruction from the original shape.

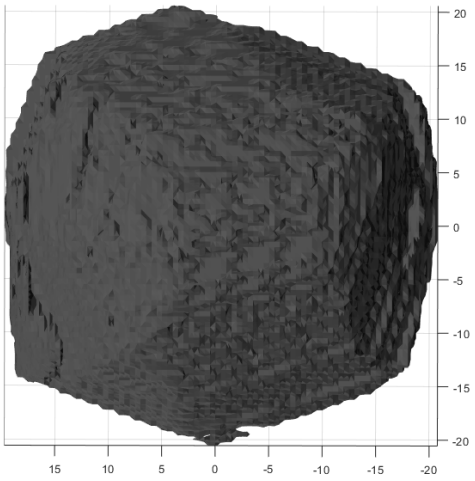


Figure 7.6: Test `W_1` at t=0s: top view of the reconstructed ice volume

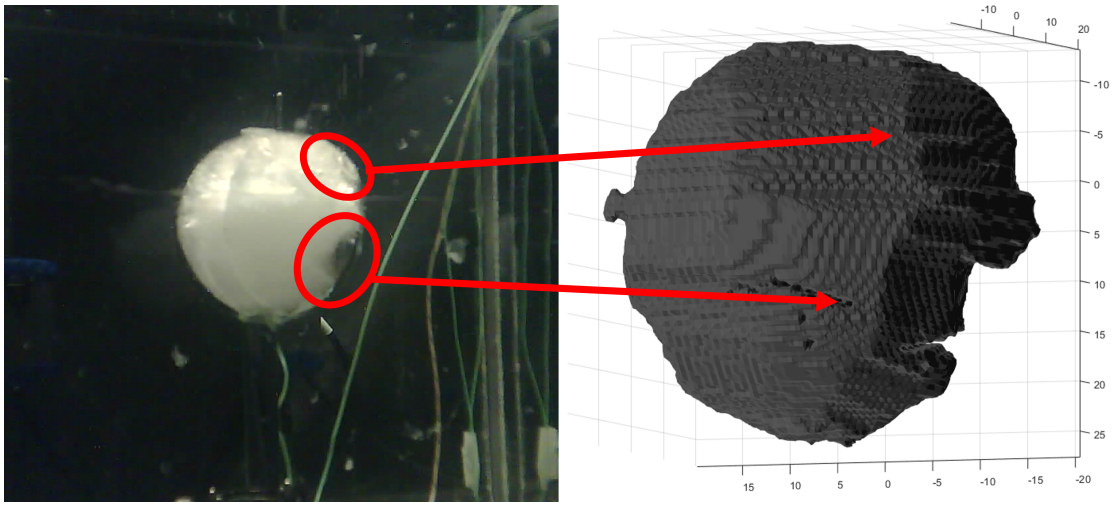


Figure 7.7: In the left image, a detail of the image acquired from Camera 3 is displayed: the transparent regions of the ice were not recognised, thus generating a reconstruction with a missing part

7.1.2 Reconstruction for t = 405 s

The second time step considered is after almost 7 minutes from the starting time. At this point, only the core of the ice is present; hence, it was not necessary to apply any image enhancement. The complete set of input images and the silhouettes are reported in Appendix A.1 and A.2. The reconstruction is depicted in Figures 7.8 and 7.9.

The current reconstruction reveals only a slightly smaller volume than the initial

one, occupying approximately 81.87% of the total voxel grid volume. This is because in the initial set of images, the reconstructed volume was lower than the real one, due to the imperfect detection of the ice boundaries.

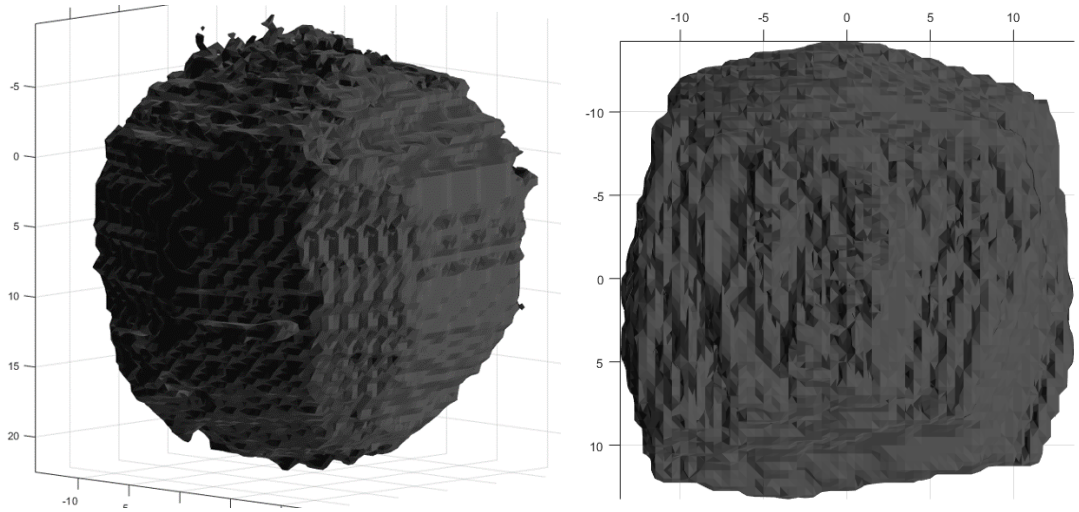


Figure 7.8: Test W_1 at t=405s: lateral view (left) and top view (right)

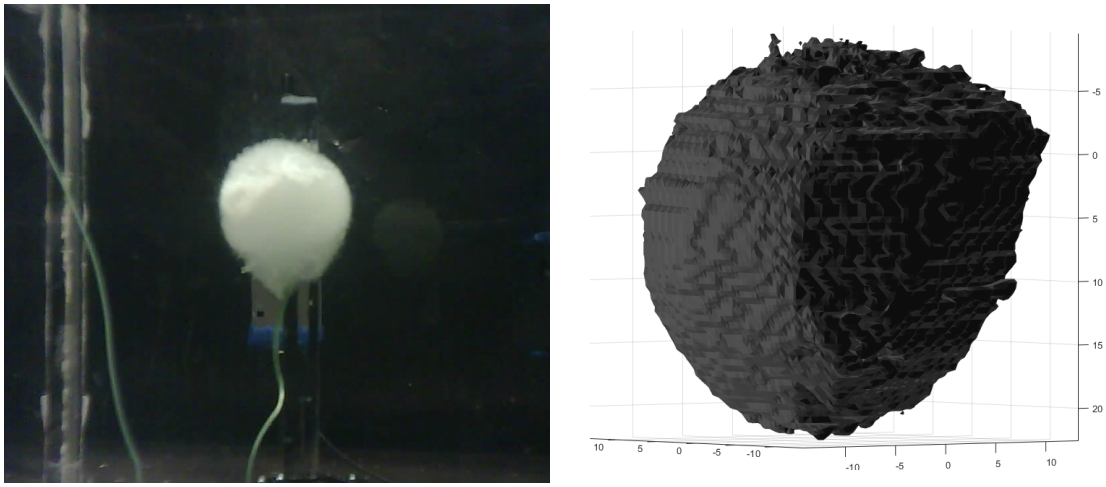


Figure 7.9: Test W_1 at t=405s: (left) a detail from Camera 5 image; (left) the output of the reconstruction

7.1.3 Reconstruction for t = 729 *s*

The last considered time step is after 729 seconds from the start (about 12 minutes). Afterwards, the reconstruction of the ice volume becomes extremely difficult or even impossible, due to its very size and insufficient contrast with respect to the background. To compute the silhouettes, the following operations were performed: background removal; grey-scale image conversion; and largest blob extraction. The results are illustrated in Figures 7.10 and 7.11.

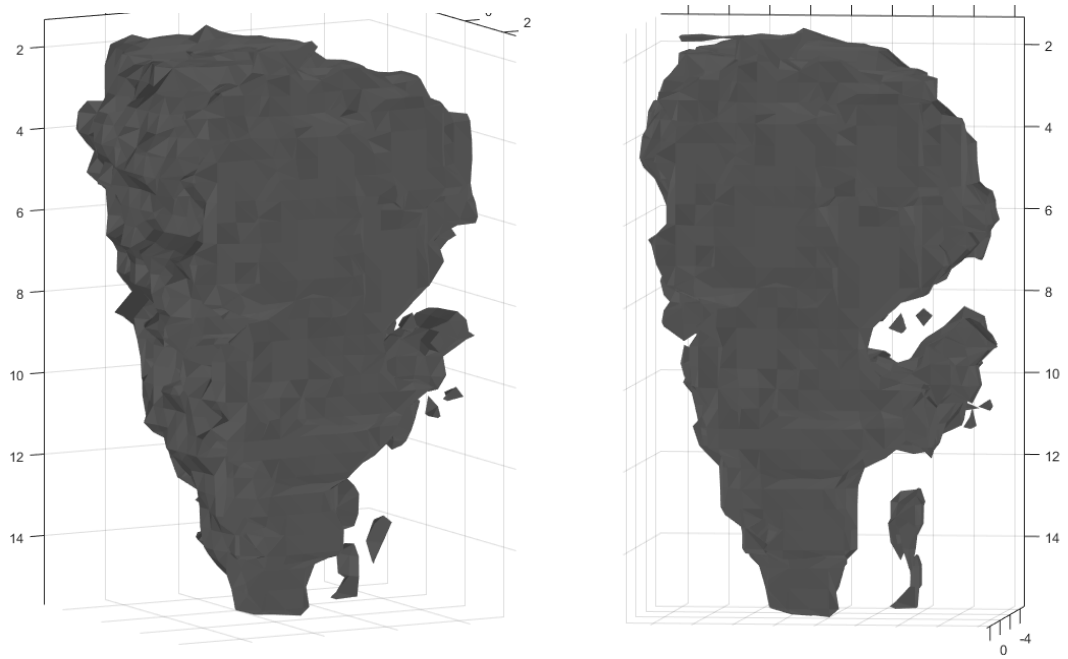


Figure 7.10: Test `W_1` at t=729*s*: lateral views.

Despite numerous operations to improve the quality of the silhouettes, the result was still rather poor, with a flawed reconstruction containing aberrations. The primary issue derives from the fact that the frozen volume is situated on one side of the supporting stick, thus invalidating some viewpoints and hindering a clear observation. In addition, the white section of the stick (where the thermocouple is fixed) poses problems for contrast enhancement due to its proximity to the iced volume, thereby significantly altering the silhouette.

Nevertheless, the reconstructed object dimensions are approximately 1 $mm \times$ 0.8 $mm \times 5$ mm, with an occupied volume in the voxel grid of about 21.76%. Therefore, the reconstructed volume steadily decreased during the different time steps.

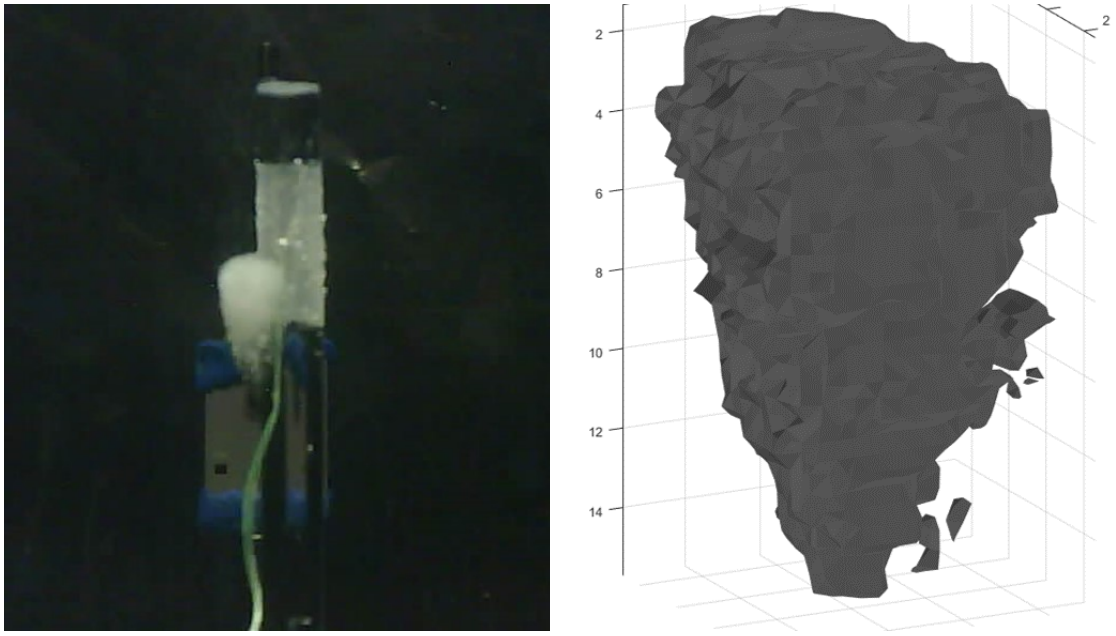


Figure 7.11: Visual comparison: detail from Camera 5 image (left); reconstructed ice volume (right)

7.2 Paraffin RT5HC, Test `Par5_1`

The second experiment series was conducted with the paraffin RT5HC, which freezes at 5 °C. Again, the tank was filled with the same paraffin type in its liquid state at ambient temperature, and the frozen body (initially at T = -21 °C) was immersed into it. The frozen paraffin body before melting is depicted in the following Figure 7.13. As opposed to the previous test case, the frozen body presents a granular appearance combined with a bright white colour. These features constitute an advantage for the Visual Hull technique, as higher contrast leads to better silhouette, thus enhancing the overall reconstruction quality.

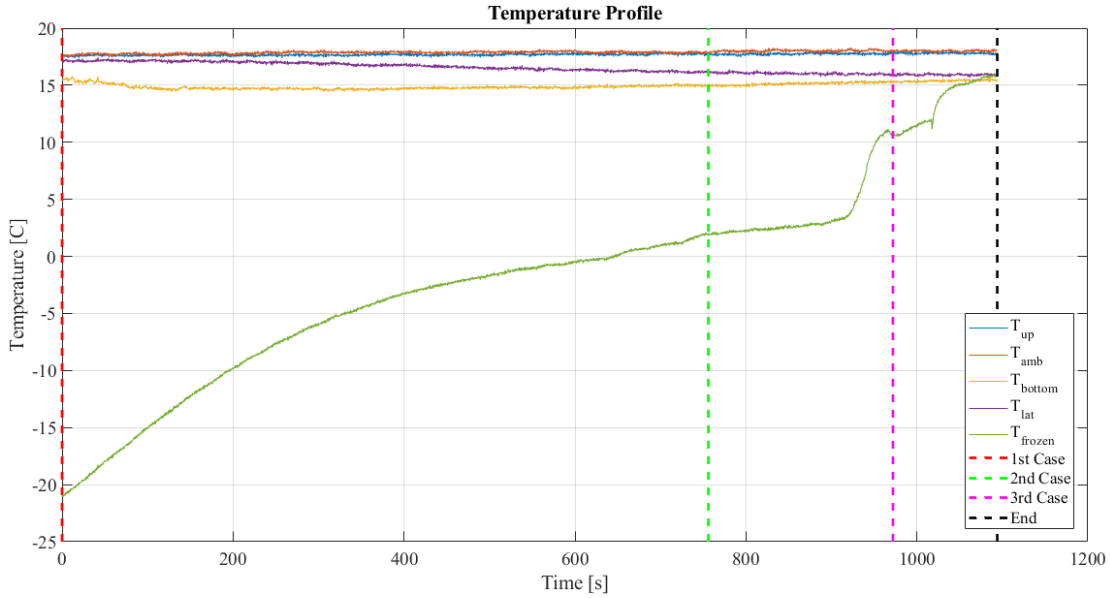


Figure 7.12: Temperature profile during the Test `Par5_1`

Photographs were taken at time intervals of 27 seconds; the total duration of the experiment was 1094 seconds (about 18 minutes). The temperature measurements evolution is displayed in Figure 7.12. As before, T_{up} is the measured temperature on the liquid paraffin surface (thus on the upper part of the tank), T_{amb} the external environment temperature, T_{bottom} the temperature at the bottom of the tank, T_{lat} is the temperature of the tank lateral wall, and T_{frozen} the temperature inside the frozen object. The temperature of the frozen paraffin steadily decreases until about 900 seconds, when the trend slope increases sharply. The frozen core was almost completely melted, and shortly after 1000 seconds, the thermocouple got exposed to the liquid state paraffin.

Three sets of images are analysed in this paragraph for the purpose of the volume reconstruction during melting: at t=0s (dashed red line in the figure), at t=756s (green dashed line) and at t=972s (magenta dashed line). The black dashed line marks the end of the experiment.

Figure 7.13: Test Par5_1: picture of the solid object at the beginning of the experiment.

The voxel grid reference dimensions are the same as in the previous case:

$$xlim = [-25 \quad 25]$$
$$ylim = [-20 \quad 30]$$
$$zlim = [-25 \quad 25]$$

7.2.1 Reconstruction for t = 0 s

As before, the first time step coincides with the initial acquisition. The input images and the silhouettes are reported in Figure 7.16 and Figure7.17 respectively.

The reconstruction is shown in Figures 7.14 and 7.15. The object reconstructions are fairly accurate, with a diameter of 5 cm and a height of 4.4 cm, due to its imperfect spherical shape. Those dimensions lead to an occupied volume of about 95.08% of the total voxel grid volume.

The high occupied volume percentage is mainly due to a superior reconstruction quality, especially if compared with the Test W_1. Indeed, the initial reconstruction of the ice sphere was obtained from points and surfaces within its body. Conversely, in the current Test Par5_1 the reconstruction was achieved solely from points

located on the body's surface.

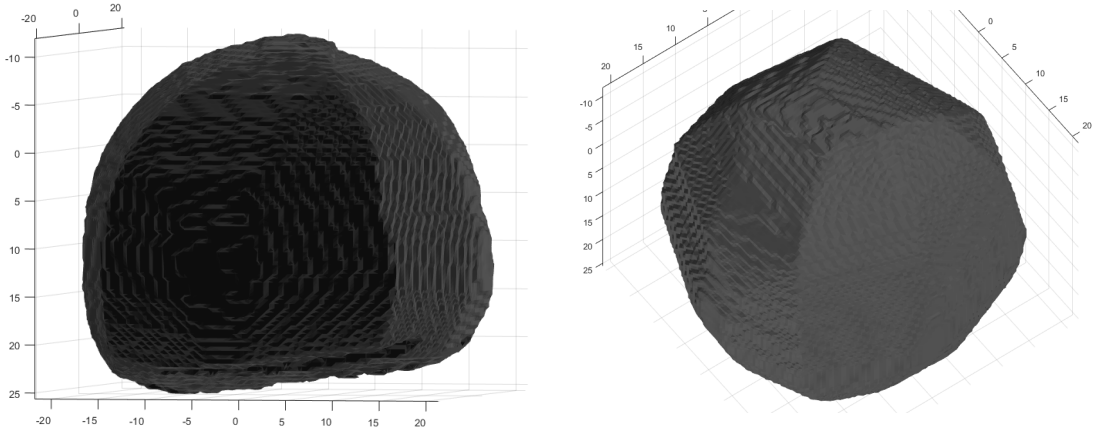


Figure 7.14: Test `Par5_1` reconstruction at t=0*s*, lateral view (left) and top view (right). The body presents a very similar geometry to the original frozen volume

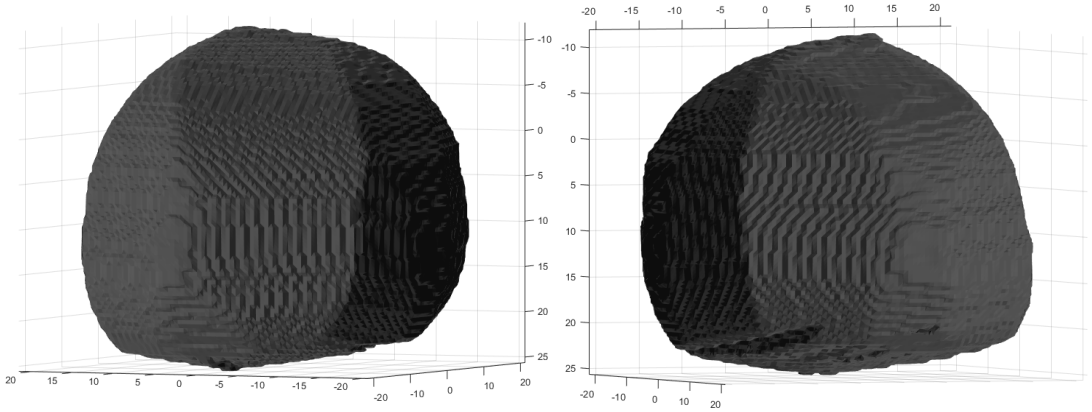
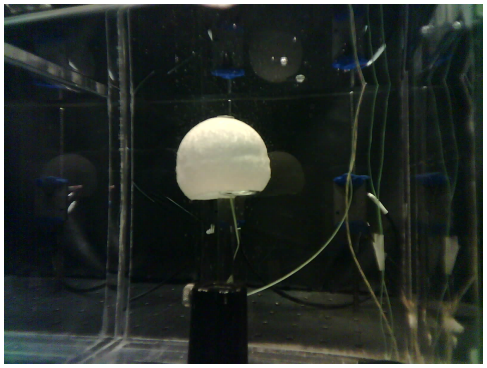
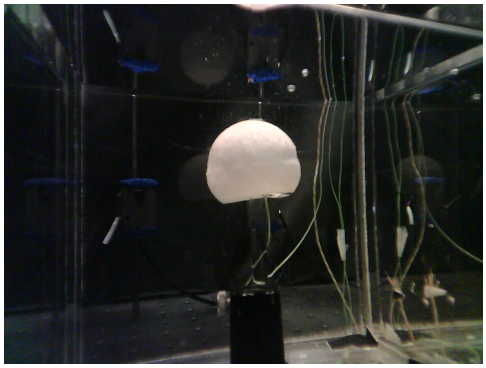


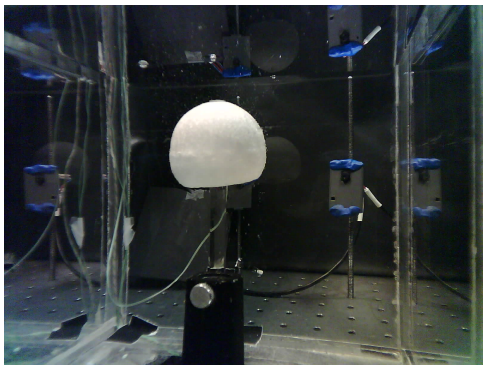
Figure 7.15: Test `Par5_1` reconstruction at t=0*s*, lateral views



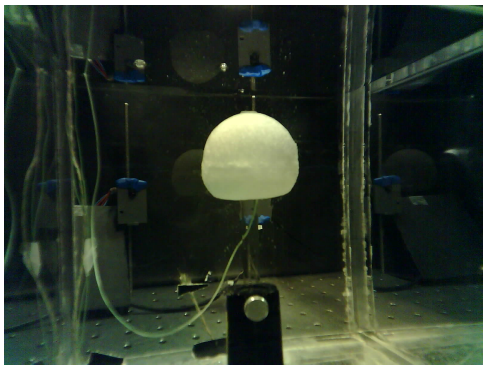
(a) Camera 1



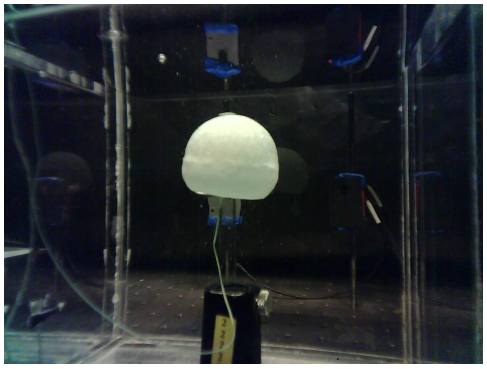
(b) Camera 2



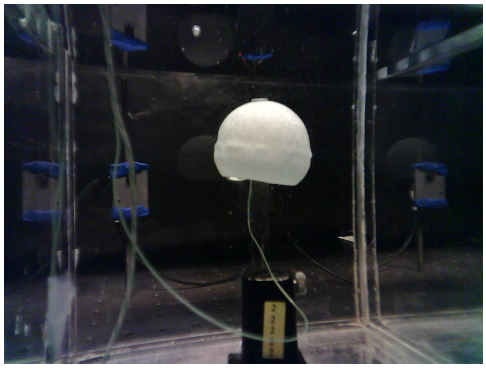
(c) Camera 3



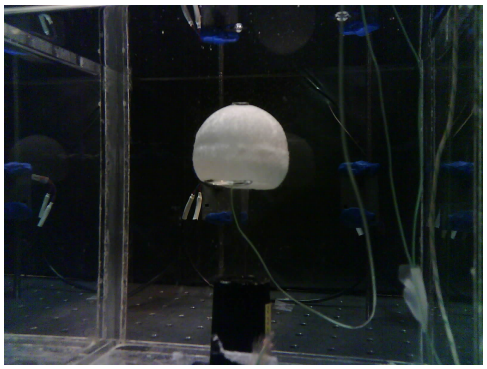
(d) Camera 4



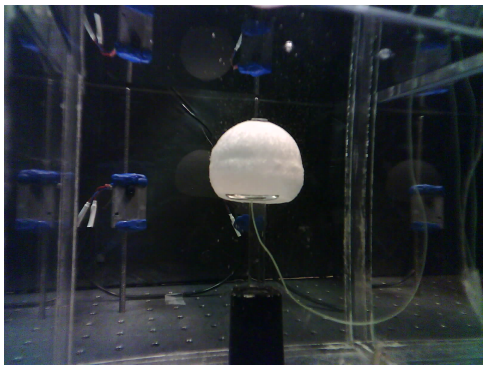
(e) Camera 5



(f) Camera 6

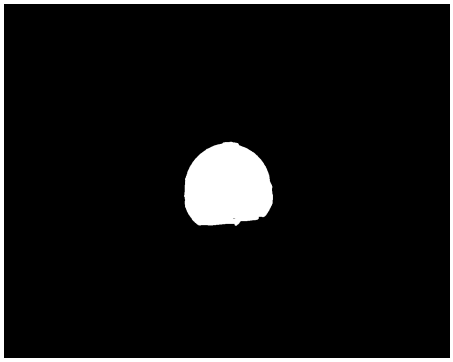


(g) Camera 7

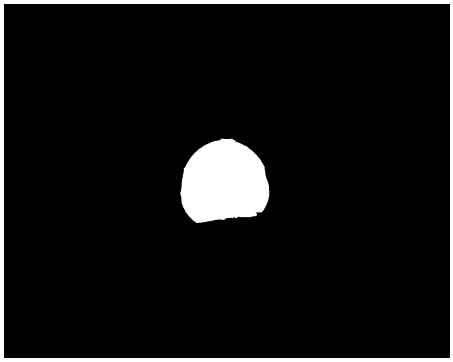


(h) Camera 8

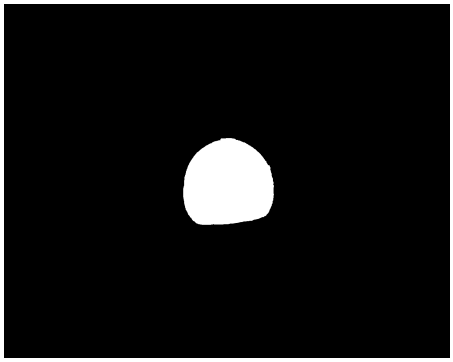
Figure 7.16: Original images for Test Par5_1 at t=0s



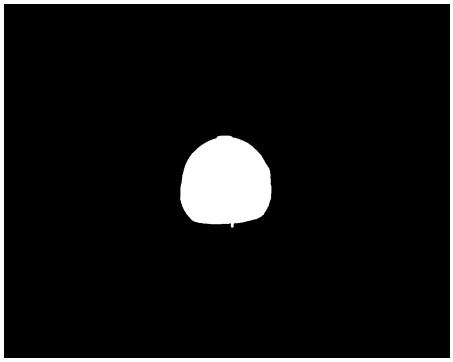
(a) Silhouette from Camera 1



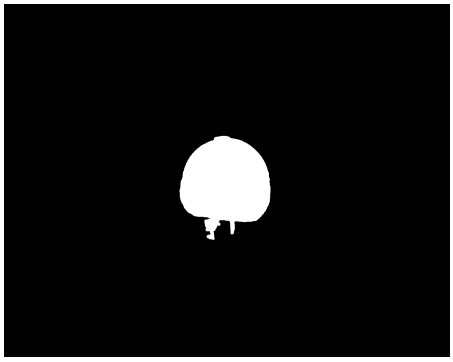
(b) Silhouette from Camera 2



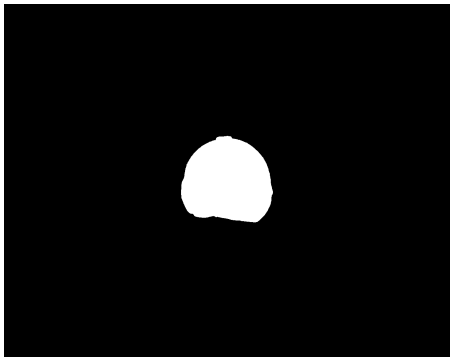
(c) Silhouette from Camera 3



(d) Silhouette from Camera 4

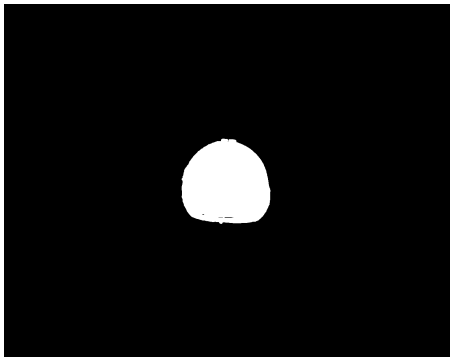


(e) Silhouette from Camera 5



(f) Silhouette from Camera 6

(g) Silhouette from Camera 7



(h) Silhouette from Camera 8

Figure 7.17: Silhouettes for Test `Par5_1` at t=0s

7.2.2 Reconstruction for t = 756 *s*

The second time step is at 756 seconds from the starting time. The paraffin frozen volumes appear to be almost halved from the beginning. The complete set of input images and silhouettes are provided in the Appendix A.6. For the silhouette computation, the grey-scale conversion and largest blob extraction were performed. The reconstruction is depicted in Figures 7.18 and 7.19. The object-occupied volume within the voxel grid is 47.70%. The remarkable aspect of this case study lies in the high reconstruction accuracy, despite a significant reduction of the target's volume; this is due to the sharp contrast of the frozen paraffin shape against the background.

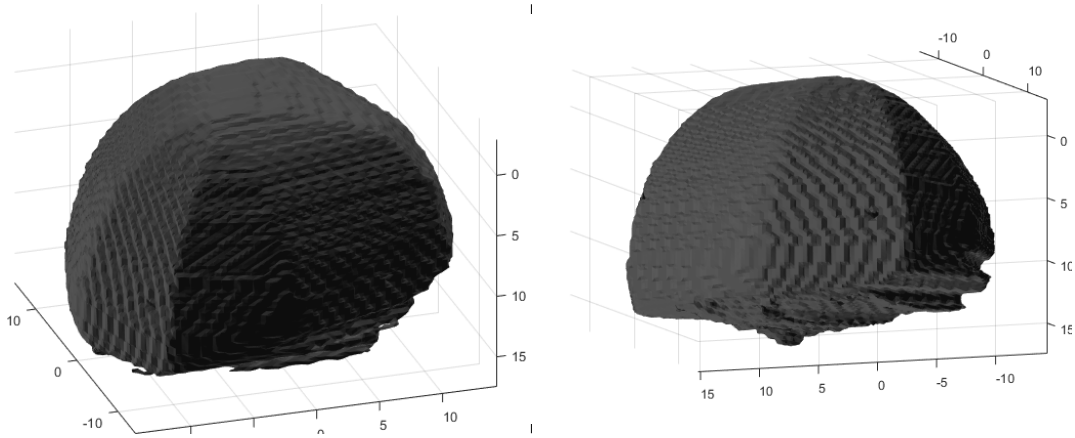


Figure 7.18: Test Par5_1 reconstruction at t=756*s*, lateral views (left) and top view (right)

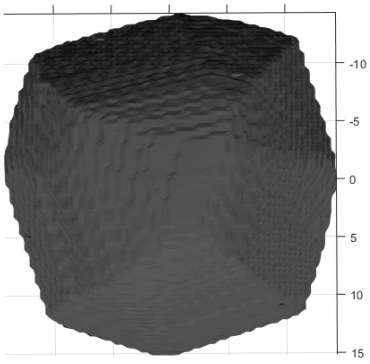


Figure 7.19: Test Par5_1 reconstruction at t=756*s*, top view

7.2.3 Reconstruction for t = 972 *s*

The last set of images analysed for this test were acquired after 972 seconds (about 16 minutes) from the starting time. The paraffin body, during its melting process, has assumed a very peculiar shape, thereby making interesting the evaluation of the algorithm's effectiveness for this scenario.

The input images and silhouettes, displayed in Figures 7.22 and 7.23 respectively, clearly illustrate an asymmetrical and randomly sized shape. Nevertheless, the algorithm succeeds in reconstructing the element fairly acceptably, albeit with some inevitable imperfections, as shown in Figures 7.20 and 7.21.

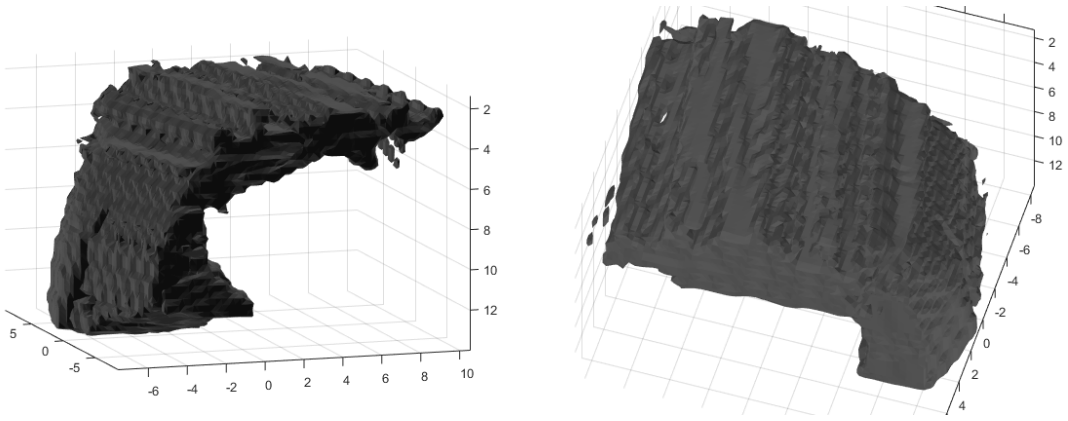


Figure 7.20: Test `Par5_1` reconstruction at t=972*s*, Lateral view (left); top view (right)

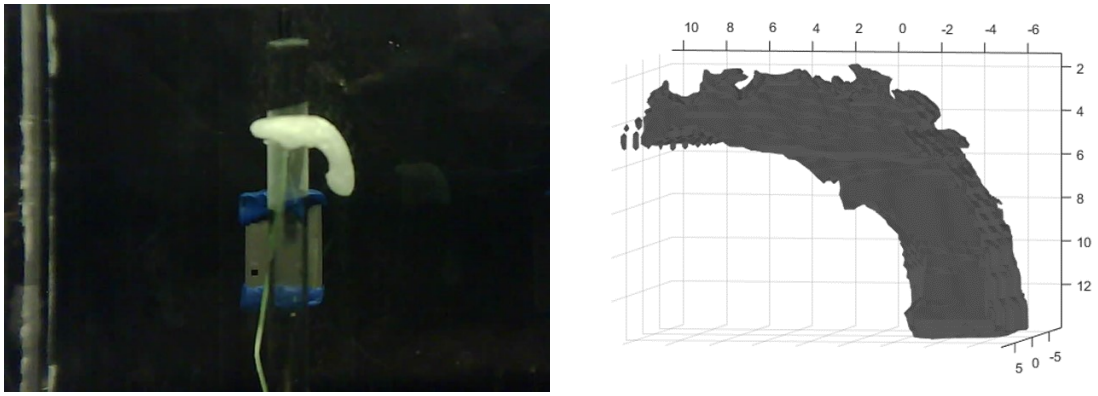
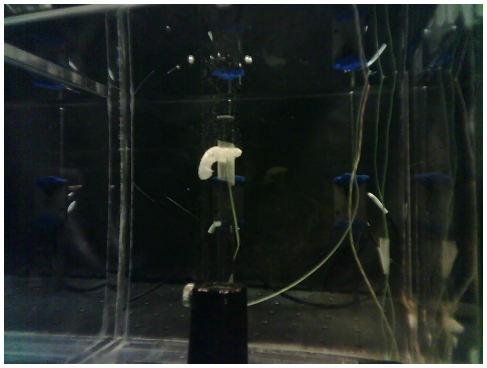
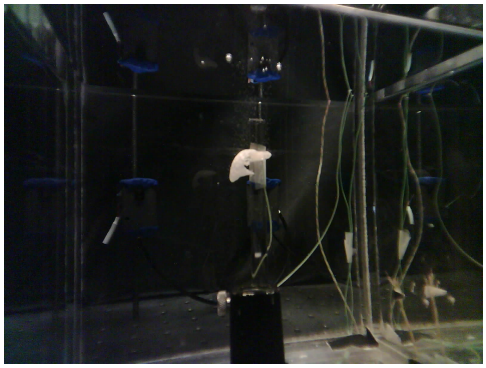


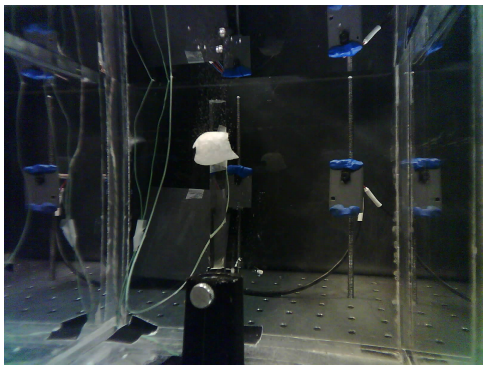
Figure 7.21: Test `Par5_1` reconstruction at t=972*s*: detail from Camera 5 image (left); reconstructed paraffin object (right)



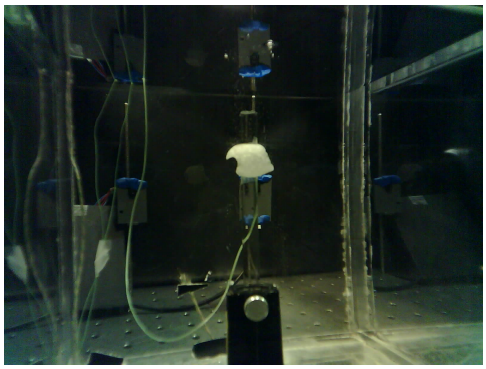
(a) Camera 1



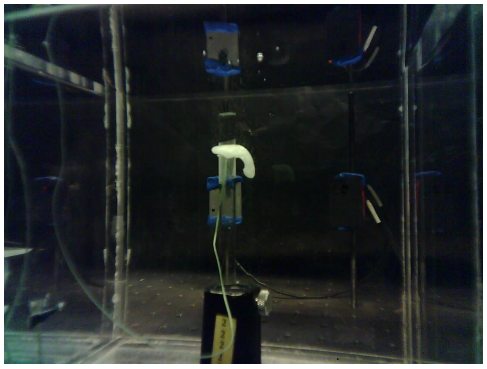
(b) Camera 2



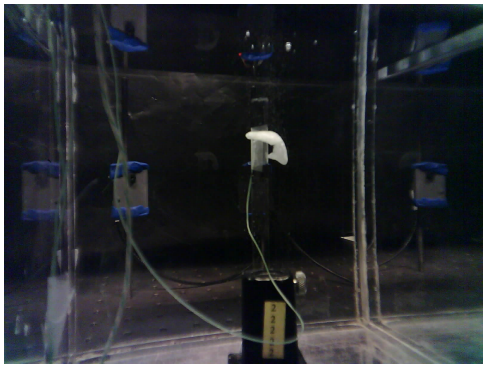
(c) Camera 3



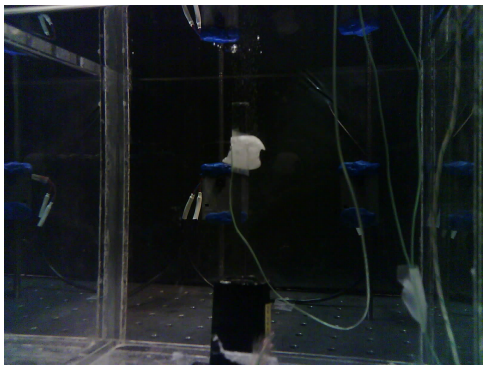
(d) Camera 4



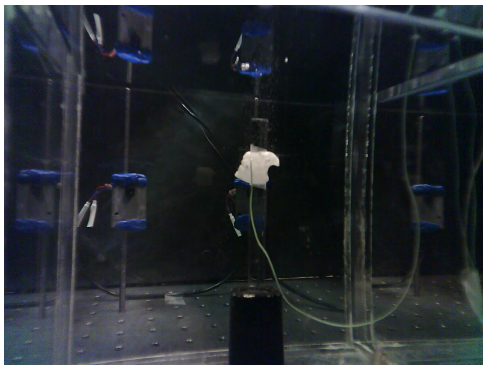
(e) Camera 5



(f) Camera 6

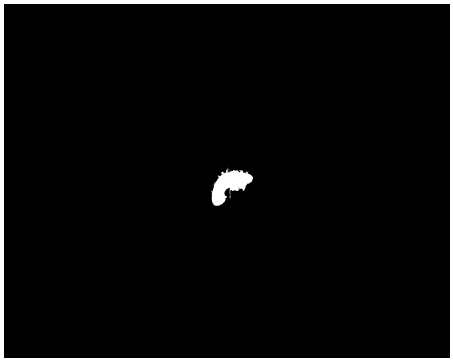


(g) Camera 7

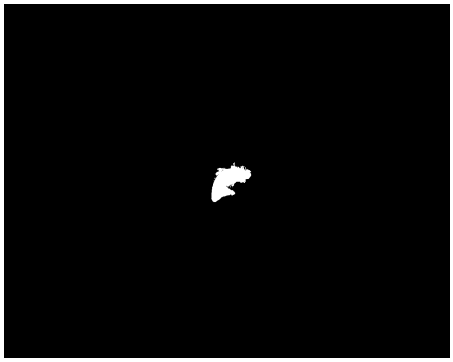


(h) Camera 8

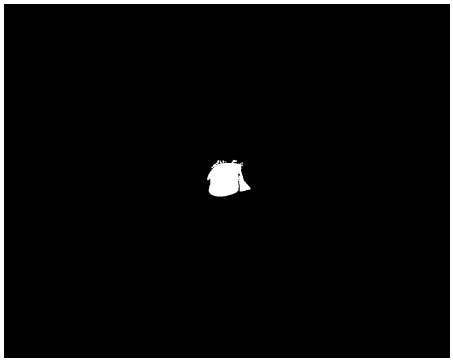
Figure 7.22: Input images for Test Par5_1 at t=972s



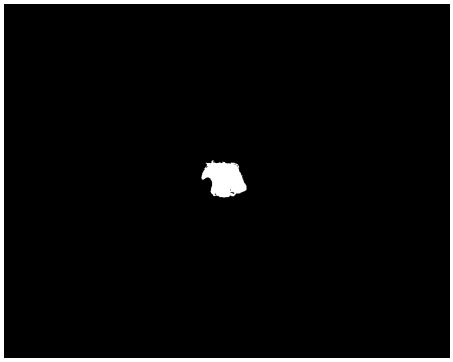
(a) Silhouette from Camera 1



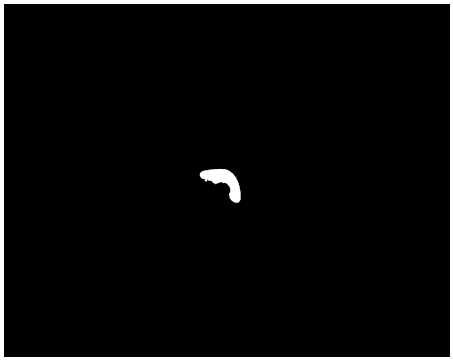
(b) Silhouette from Camera 2



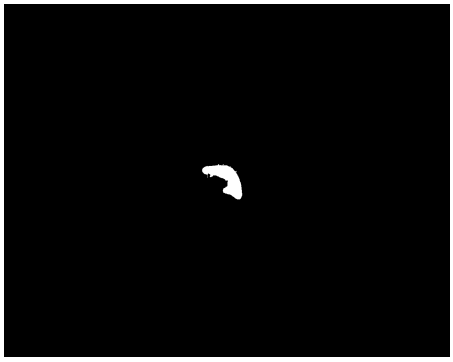
(c) Silhouette from Camera 3



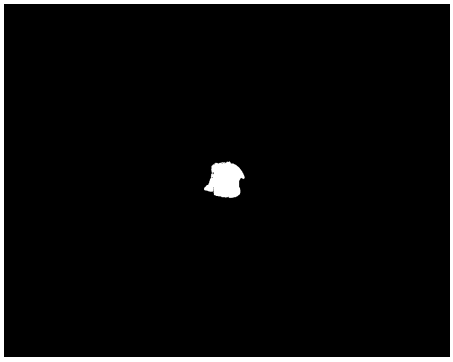
(d) Silhouette from Camera 4



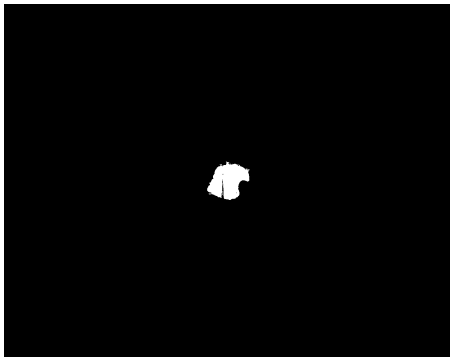
(e) Silhouette from Camera 5



(f) Silhouette from Camera 6



(g) Silhouette from Camera 7



(h) Silhouette from Camera 8

Figure 7.23: Silhouettes for Test `Par5_1` at t=972s

7.3 Paraffin RT-9HC, Test Par9_2

The third test was performed with the paraffin RT-9HC, with a freezing point at -9°C. As for the previous cases, the tank was filled with the same paraffin type in its liquid state (initially at T=18 °C), and the frozen body (initially at T=-20°C) was immersed in it. This time, due to the paraffin properties, the rubber mould was not able to fully contain it, thereby shaping the crystallised body with a half-spherical geometry, as depicted in Figure 7.32. Again, the images were taken at time intervals of 27 seconds; the total duration of the experiment was 704 seconds (about 11 minutes). The temperature profile is illustrated in the following Figure 7.24:

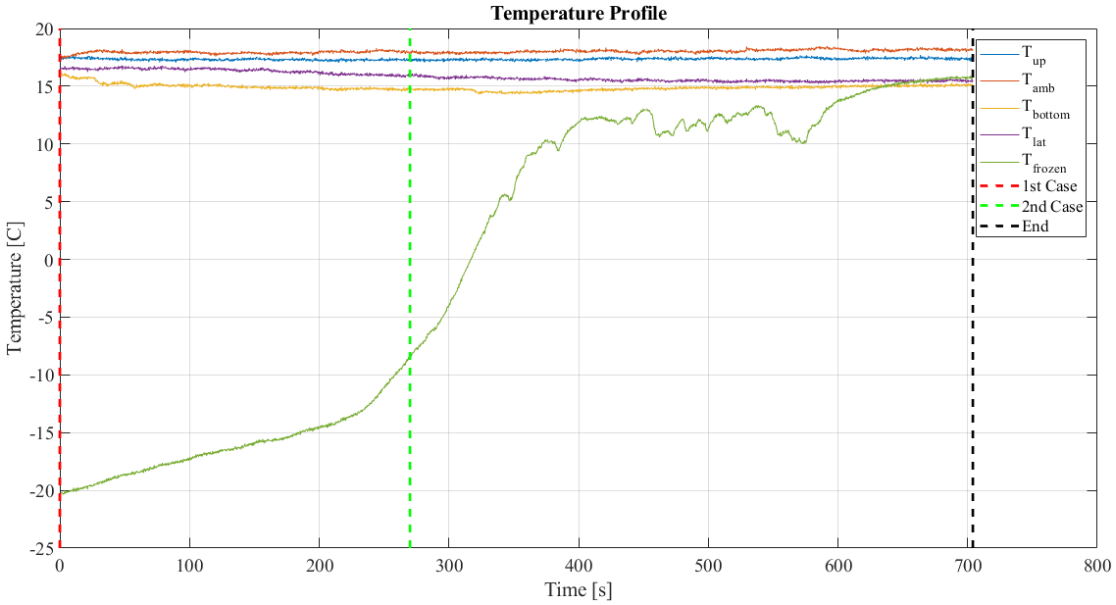


Figure 7.24: Temperature profile during the Test Par-9_1

where T_{up} is the measured temperature on the liquid paraffin surface (thus on the upper part of the tank), T_{amb} the external environment temperature, T_{bottom} the temperature at the bottom of the tank, T_{lat} is the temperature of the tank lateral wall, and T_{frozen} the temperature inside the frozen object. The temperature of the frozen paraffin decreases sharply until about 400 seconds when the trend slope increases sharply. It then follows an oscillating pattern between 10°C and 13°C: this interval corresponds to the paraffin core melting. After about 570 seconds, the thermocouple was exposed to the liquid state paraffin, thus the temperature started to align with the others within the tank.

This time, only two sets of images are analysed: the first at t=0s (dashed red

line in the figure) and the second at t=270s (green dashed line).

Figure 7.25: Test `Par9_2`: picture of the solid object at the beginning of the experiment.

7.3.1 Reconstruction for t = 0 *s*

The input images and silhouettes are displayed in the next pages (Figures 7.28 and 7.29). As depicted in Figure 7.26 and Figure 7.27, the reconstruction of the solid shape appears to be very accurate, even displaying most of the surface imperfections of the initial body. The occupied volume within the bounding box is 94.71%, thus demonstrating the algorithm's excellent performance.

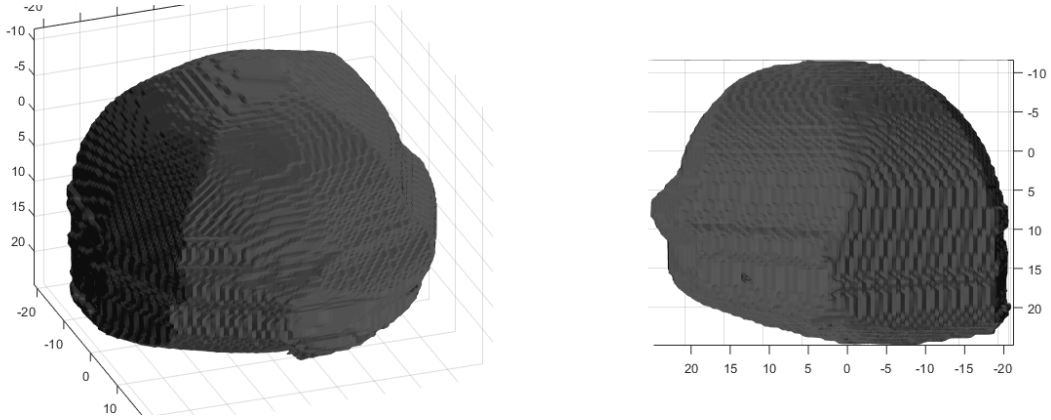


Figure 7.26: Test Par9_2 reconstruction at t=0*s*, lateral views

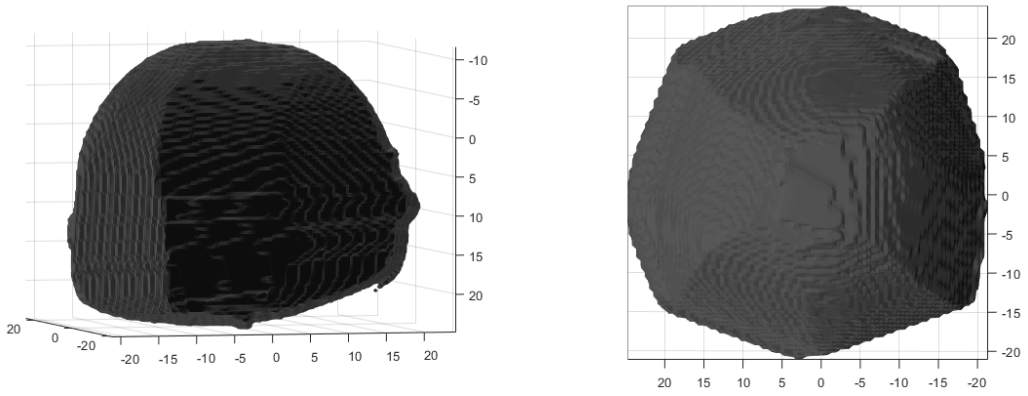
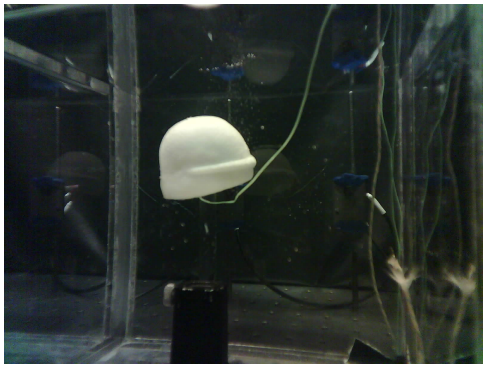
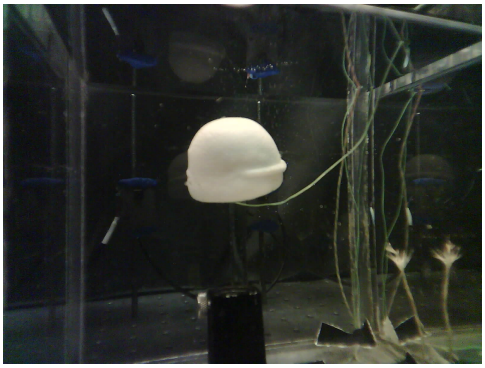


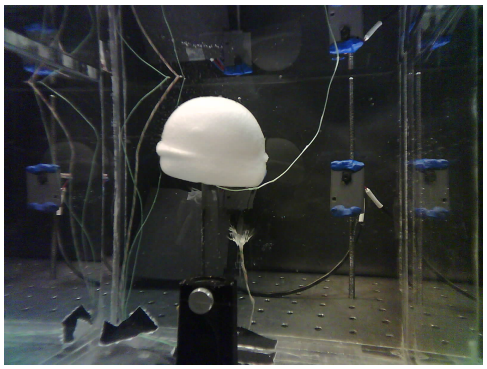
Figure 7.27: Test Par9_2 reconstruction at t=0*s*, more details (left); top view (right)



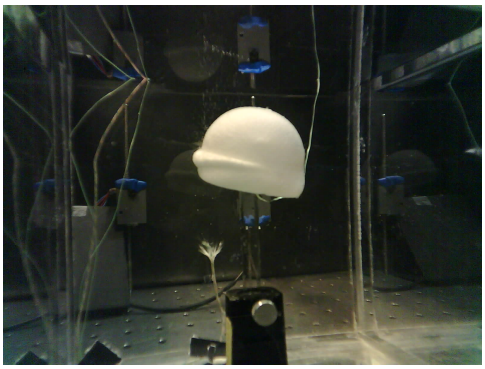
(a) Camera 1



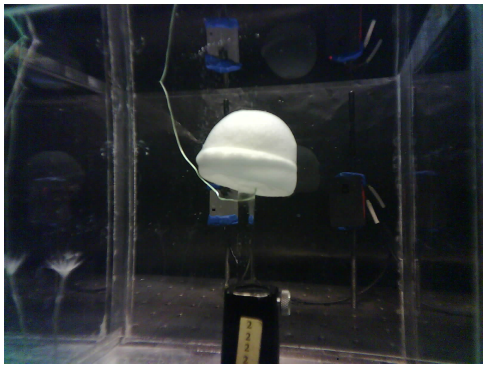
(b) Camera 2



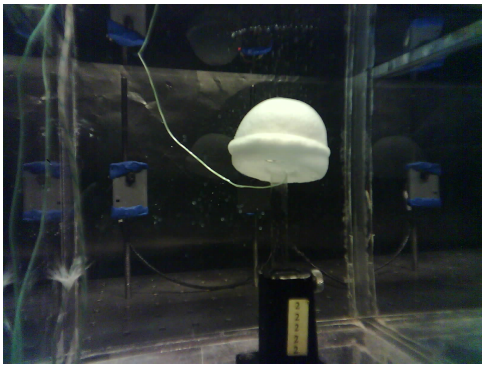
(c) Camera 3



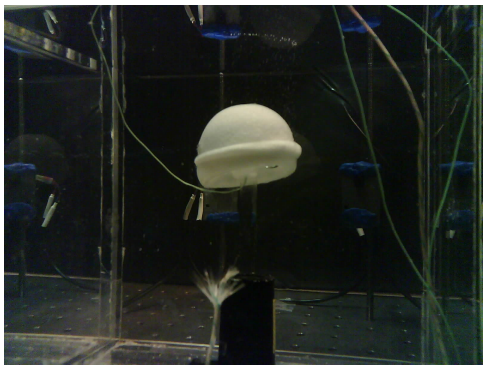
(d) Camera 4



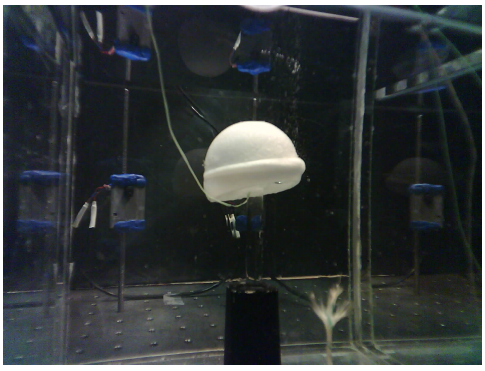
(e) Camera 5



(f) Camera 6



(g) Camera 7



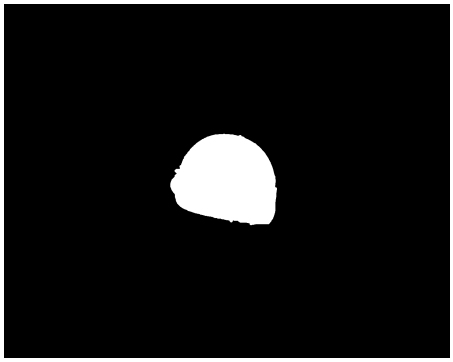
(h) Camera 8

Figure 7.28: Original images for Test `Par-9_1` at t=0s

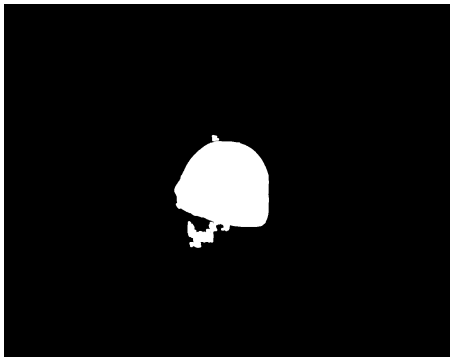
(a) Silhouette from Camera 1

(b) Silhouette from Camera 2

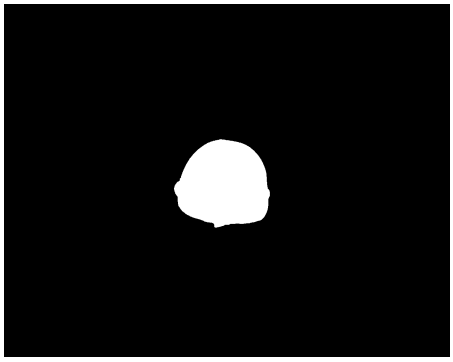
(c) Silhouette from Camera 3



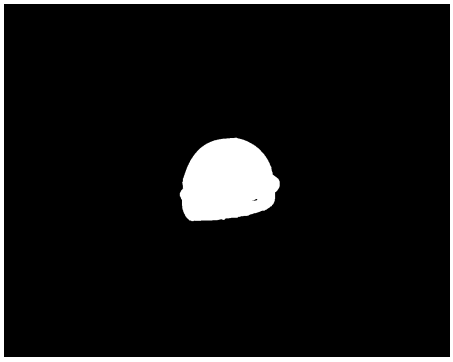
(d) Silhouette from Camera 4



(e) Silhouette from Camera 5



(f) Silhouette from Camera 6



(g) Silhouette from Camera 7

(h) Silhouette from Camera 8

Figure 7.29: Silhouettes for Test `Par-9_1` at t=0s

7.3.2 Reconstruction for t = 270 *s*

The input images and silhouettes are displayed in the Appendix A.7 and A.8. Once again, the reconstructed object appears to be rich in details, as shown in Figures 7.30 and 7.31. The occupied volume within the bounding box is 66.34%.

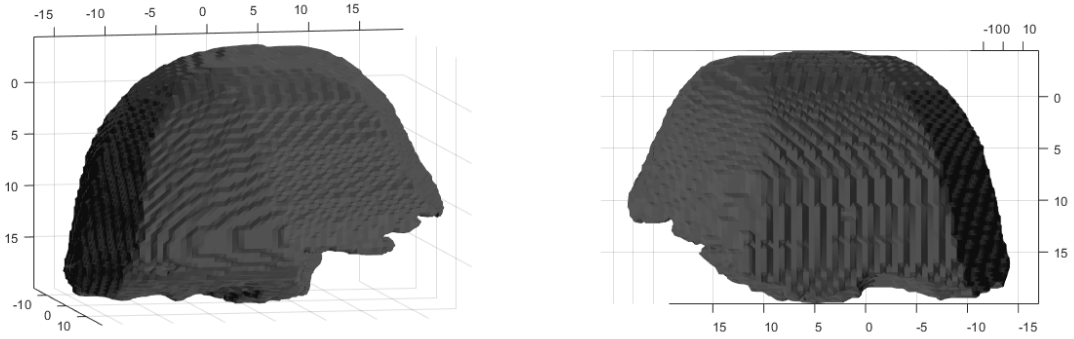


Figure 7.30: Test `Par9_2` reconstruction at t=270*s*, opposite lateral views

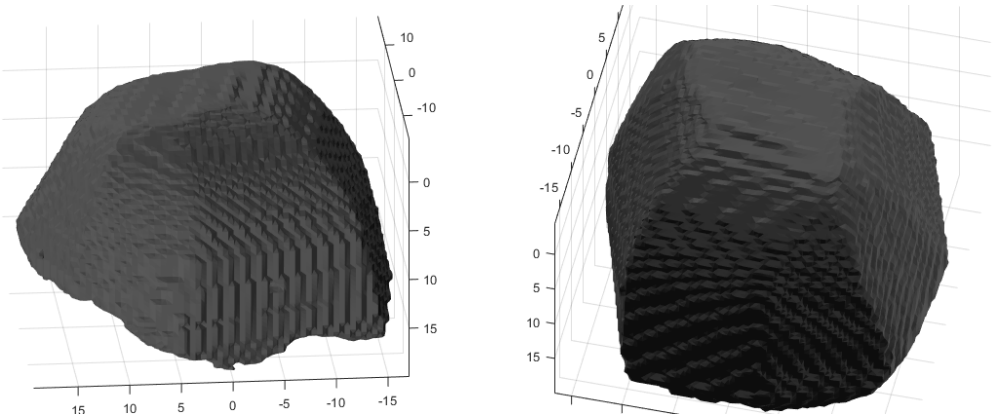


Figure 7.31: Test `Par9_2` reconstruction at t=270*s*, top views

7.4 Volumes Comparison

In order to evaluate the volumetric variation of the melting body, it is possible to compute an approximation of the reconstruction volume can be computed. The calculation is performed by counting all the voxels that are part of the reconstruction and then multiplying the result by a single voxel's dimensions. In `MATLAB` it can be done as:

$$Vol_r = sum(voxels(:, 4) > 0) \cdot prod(voxel_size)$$

Since the considered voxel grid was the same for all the reconstructions, a single voxel has a fixed dimension of $0.5mm \times 0.5mm \times 0.5mm$. The results are plotted for the considered test cases `W_1`, `Par5_1` and `Par9_2`.

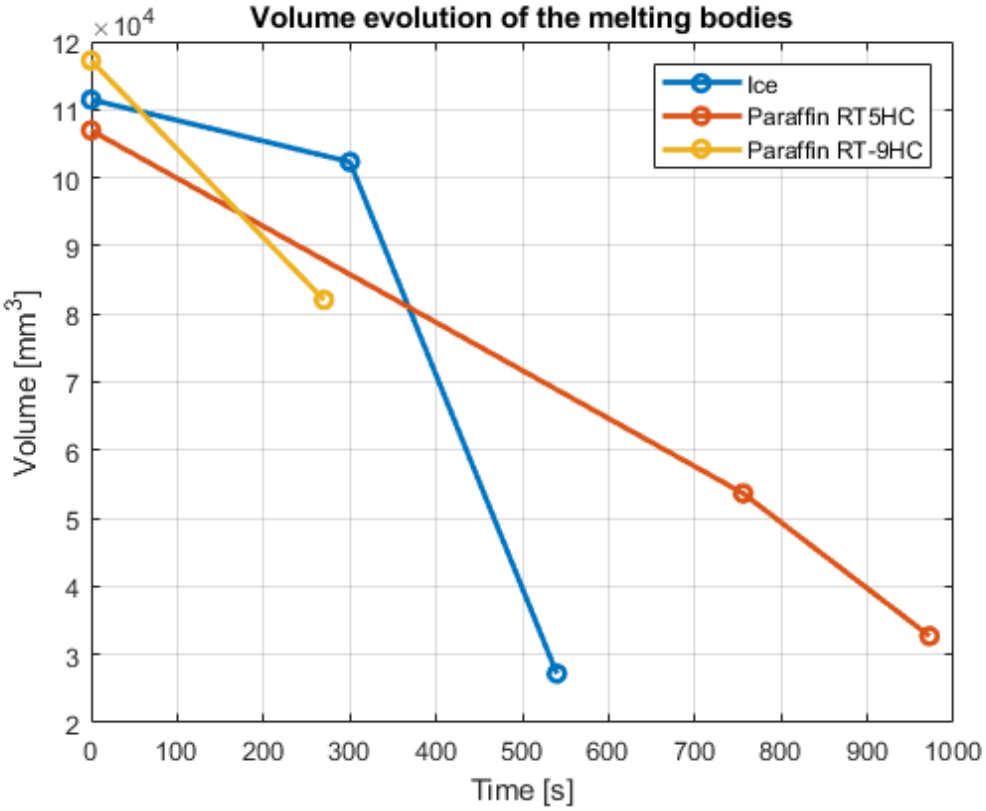


Figure 7.32: Volume evolution of the frozen bodies as time passes

The results are significantly influenced by the material properties. For instance, the RT5HC paraffin experiences a slower melting process compared to the ice, due to its higher freezing point. Conversely, the RT-9HC paraffin displays a faster melting curve as its freezing temperature is considerably lower than the conditions

to which the frozen body was subjected.

The comprehensive results obtained throughout the analysis are summarised in Table 7.2 below. An important remark about the approximate volume needs to be pointed out. As previously defined, this volume corresponds to the number of voxels that form the reconstruction, multiplied by the size of a voxel. However, these voxels are not always completely filled, and this occurrence -which can be quite frequent- leads to the computation of an overestimated volume, sometimes much larger than the actual volume. Such a situation can be found in 7.1.3 and 7.2.3, where a small and fragmented object is reconstructed, but the resulting volume still appears to be bigger than the actual shape, as showcased in Table 7.2.

Consequently, it can be concluded that this parameter can only be considered as a preliminary estimation, rather than an accurate reconstruction descriptor.

Substance	**Time step** $[s]$	V_{occ}	**Approximated volume** $[mm^3]$
Water	0	89.18%	$1.1147 \cdot 10^5$
	405	81.87%	$1.0233 \cdot 10^5$
	729	21.76%	27195
RT5HC	0	95.08%	$1.0696 \cdot 10^5$
	756	47.70%	$5.3659 \cdot 10^4$
	972	29.0781%	$3.2713 \cdot 10^4$
RT-9HC	0	94.7123%	$1.1721 \cdot 10^5$
	270	66.34%	$8.2076 \cdot 10^4$

Table 7.2: Volumes comparison

Chapter 8

Conclusion and future work

The main objectives of this research were to provide a novel valid approach for slush particle observation and to establish an extensive experimental database to validate predictive models for solid particle melting. These goals are essential for a better understanding of the slush state, especially for future energy storage applications.

To accomplish this, experiments were conducted using a circular multi-camera array and a plexiglass tank, where the analysed substances, solidified in spherical geometry, were immersed in their respective liquid phases and melted under controlled temperature conditions. The acquired data was then employed to obtain 3D models of the crystallised bodies for different time steps. Different types of fluids were investigated, providing a wide database of slush state simulations.

The first step was to identify a substance whose slush was in similarity to the target application of slush hydrogen. Several fluids were analysed in terms of density ratio and particle settling velocity. Paraffin was selected as a satisfying substitute candidate because of its transparent appearance, high solidification temperature and non-toxicity. A dedicated experimental setup was then designed and built, to allow optical access to the experimental specimen and temperature measurement to control the experiment boundary conditions.

To describe the melting bodies' visual features, the Visual Hull (VH) technique was adopted, a computer vision method that allows three-dimensional information to be retrieved from a two-dimensional representation of the object shape. The approach proved effective in accurately approximating static melting objects but

showed limitations in reconstructing targets with unclear shape segmentation from the background. Extensive Image Processing operations were also performed on the input images, to improve the shape segmentation from the background.

The experiment yielded overall success, demonstrating the effectiveness of the technique in reconstructing the 3D interface of a melting particle over time. Each fluid showed a different melting dynamic, which is consistent with the required experimental framework.

Further steps forward concern the input image pre-processing and the experimental setup. In particular, the employed image processing routines proved to be extremely time-consuming, as they depended on each image's features. Future work can therefore integrate specific image processing routines for each camera, thus automating the operation.

Moreover, this work focused solely on replicating the slush phenomenon on a macroscopic scale, whereas in reality, this phenomenon occurs on significantly smaller scales. Future research endeavours should therefore focus on developing the technique for microscopic measurements in this domain.

To tackle this challenge, many improvements can be made to the model. Firstly, the camera positions can be optimized, potentially introducing an asymmetric array. In addition, the number of viewpoints can be increased. To this extent, the integration of mirror systems can be an interesting option, but only if provided with an effective calibration method. Refinement of the calibration process and the introduction of a more accurate camera model represent further potential steps forward.

In conclusion, this research proved the effectiveness of the Visual Hull technique for melting particle observation, thereby contributing to the slush-state characterisation.

Appendix A

Visual Hull Results

In this section are reported the input images and silhouettes of the test cases not included in the main work.

A.1 Multi-Camera Control Code

The following code was written using `Python` as programming language; its purpose is to let the user adjust the camera settings and start a recurrent acquisition, with a predefined time step.

Listing A.1: Python Multi-Camera Control Code

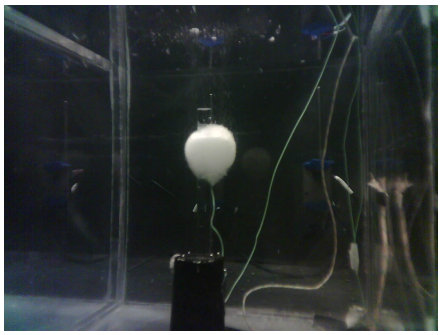
```
import requests
import os
import time

main_folder = "camera folder"
execution_count = 1 # Count the number of execution
# camera IP address complete list
camera_ips = ["192.168.0.100","192.168.0.101","192.168.0.102","
    192.168.0.105","192.168.0.106","192.168.0.107", "192.168.0.108","
    192.168.0.109"]

# Set the parameters needed for the acquisition
for camera_ip in camera_ips:
    try:
        params = {
            "framesize": "13",  # 1600x1200
```

```
15          #     "quality": "63",  # JPEG quality (from 4 to 63)
16          #     "auto_exposure":"false",
17          }
18          response = requests.get(f"http://{camera_ip}/control?var=
    framesize&val=13", params=params)
19          if response.status_code == 200:
20              print(f"Camera {camera_ip} parameters has been set.")
21          else:
22              print(f"Camera {camera_ip} did not respond correctly.")
23      except requests.exceptions.RequestException as e:
24          print(f"Error while communicating with camera {camera_ip}: {e
    }")
25
26  via = input("Press [Enter] to start the acquisition...")
27
28  # Main loop
29  while True:
30      folder_name = os.path.join(main_folder, str(execution_count))
        # Create folder for current execution
31      os.makedirs(folder_name, exist_ok=True)
32
33      for i, camera_ip in enumerate(camera_ips):      # Take the picture
34          try:
35              response = requests.get(f"http://{camera_ip}/capture")
36              if response.status_code == 200:
37                  image_data = response.content
38                  image_name = f"camera_{i + 1}.jpg"
39                  image_path = os.path.join(folder_name, image_name)
40                  with open(image_path, "wb") as image_file:
41                      image_file.write(image_data)
42                      print(f"Image correctly acquired by camera {
    camera_ip} and saved as camera_{i + 1}.png")
43              else:
44                  print(f"Camera {camera_ip} did not respond correctly.
    ")
45          except requests.exceptions.RequestException as e:
46              print(f"Error while communicating with camera {camera_ip
    }: {e}")
47
48      # Increase the execution counter ore
49      execution_count += 1
```

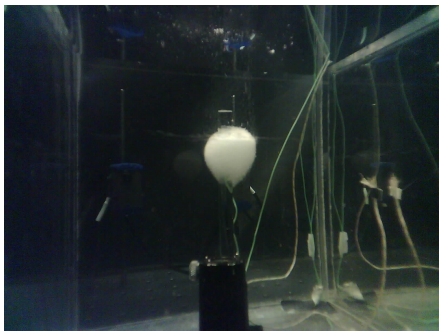
```
50      print(f"Numero di iterazione {execution_count - 1}")
51      time.sleep(20) # Time before the next execution
```

A.2 Slush State Water

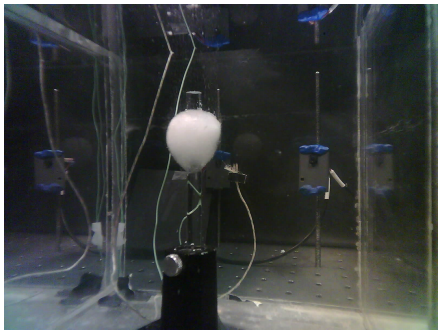
A.2.1 Reconstruction for t = 405 *s*



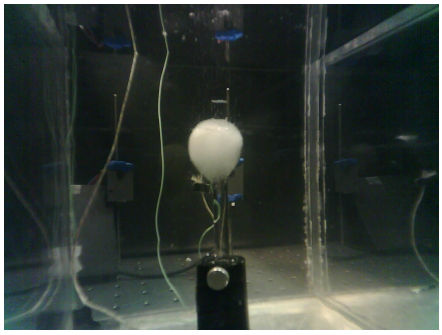
(a) Camera 1



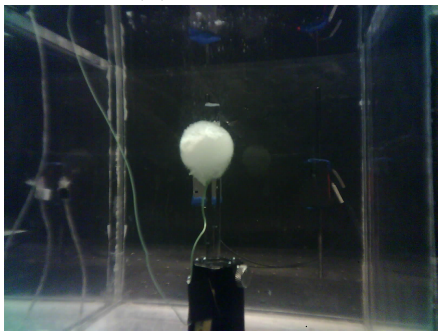
(b) Camera 2



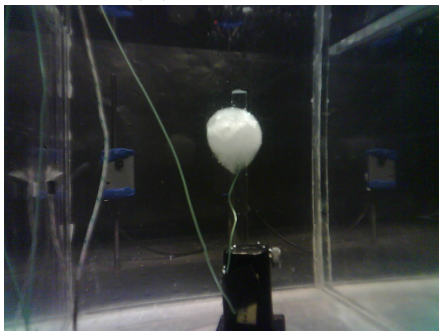
(c) Camera 3



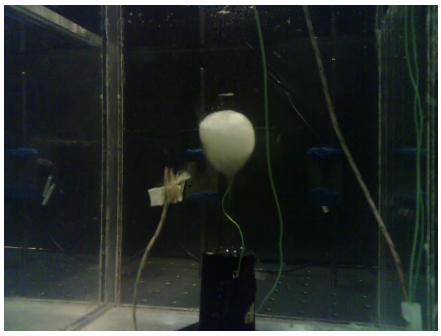
(d) Camera 4



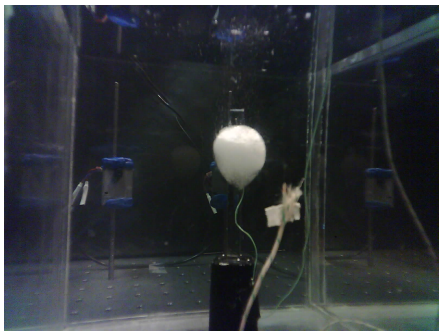
(e) Camera 5



(f) Camera 6



(g) Camera 7

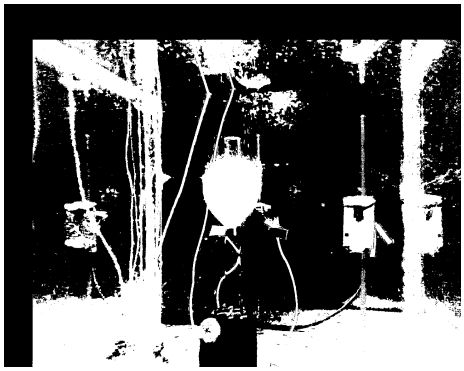


(h) Camera 8

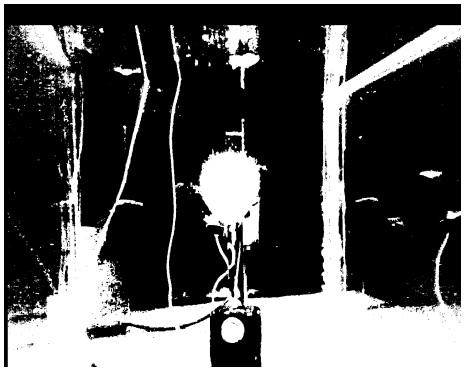
Figure A.1: Original ice images for t = 405 seconds

(a) Silhouette of ice from Camera 1

(b) Silhouette of ice from Camera 2



(c) Silhouette of ice from Camera 3



(d) Silhouette of ice from Camera 4

(e) Silhouette of ice from Camera 5

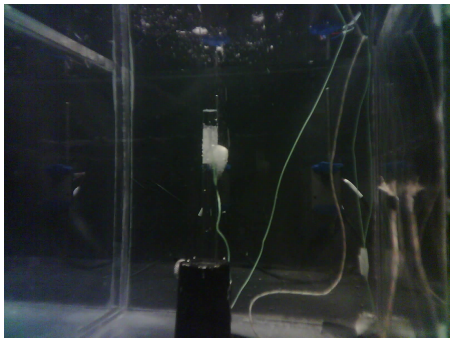
(f) Silhouette of ice from Camera 6

(g) Silhouette of ice from Camera 7

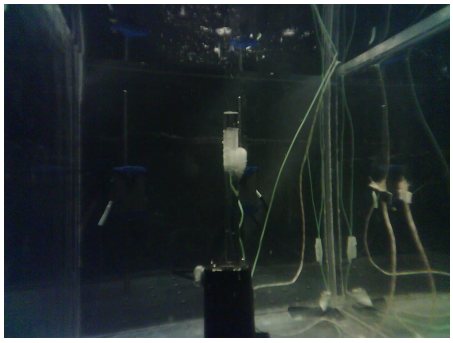
(h) Silhouette of ice from Camera 8

Figure A.2: Ice silhouettes for t = 405 seconds

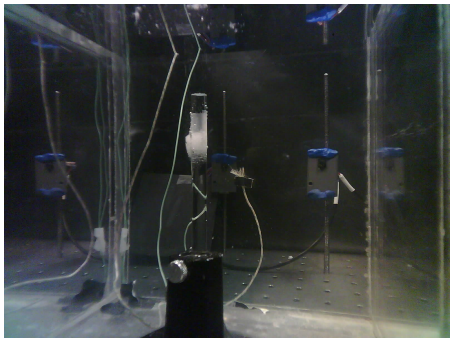
A.2.2 Reconstruction for t = 729 *s*



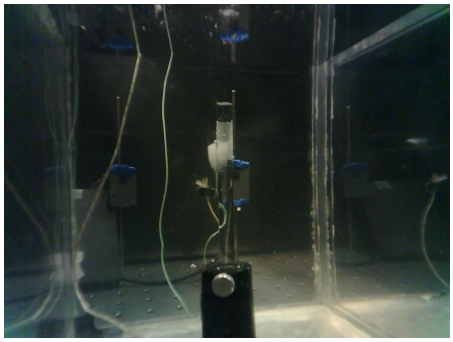
(a) Camera 1



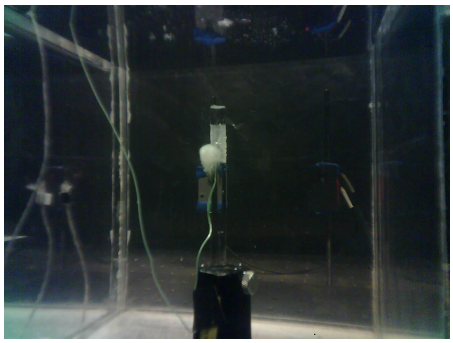
(b) Camera 2



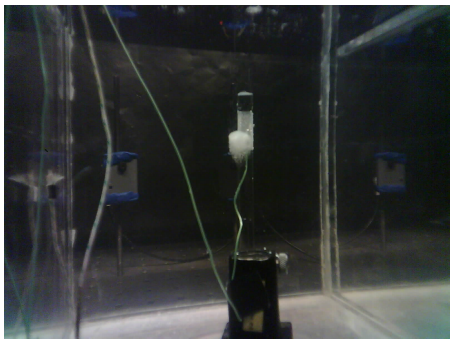
(c) Camera 3



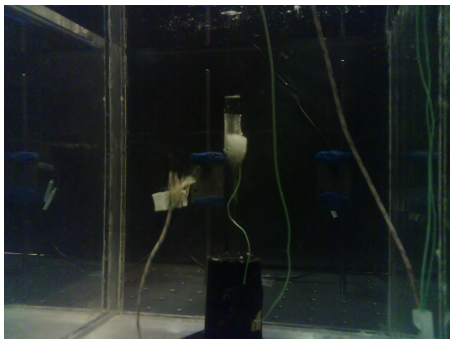
(d) Camera 4



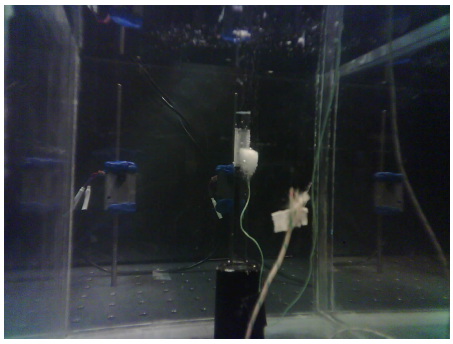
(e) Camera 5



(f) Camera 6

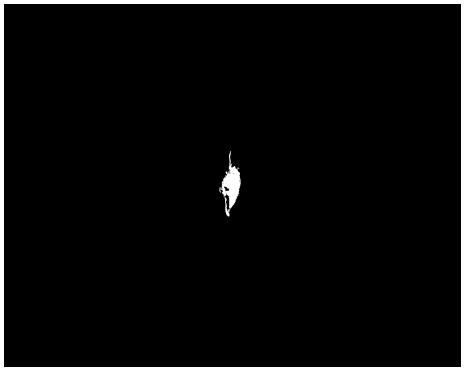


(g) Camera 7

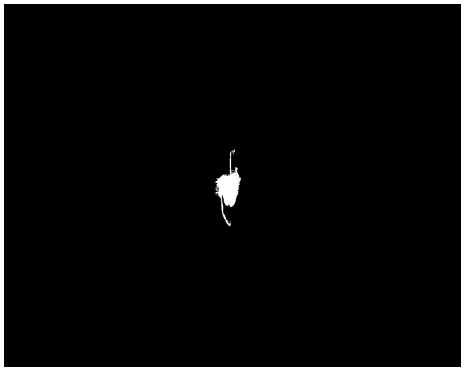


(h) Camera 8

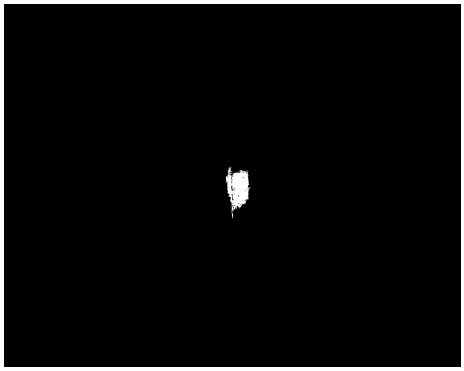
Figure A.3: Original ice images for t = 729 seconds



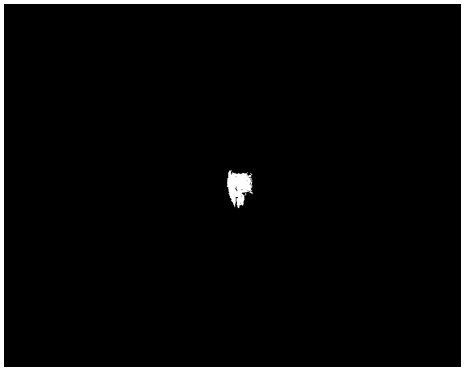
(a) Silhouette of ice from Camera 1



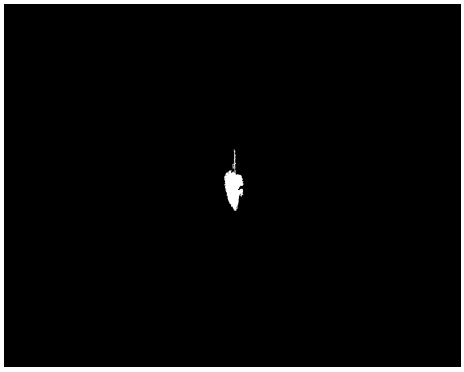
(b) Silhouette of ice from Camera 2



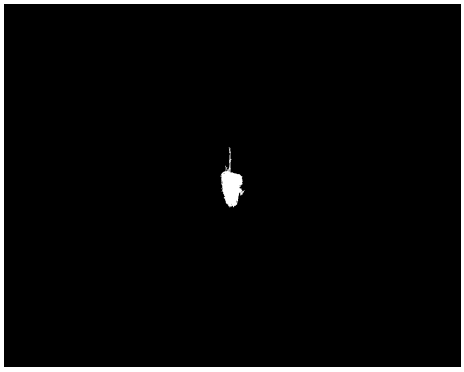
(c) Silhouette of ice from Camera 3



(d) Silhouette of ice from Camera 4

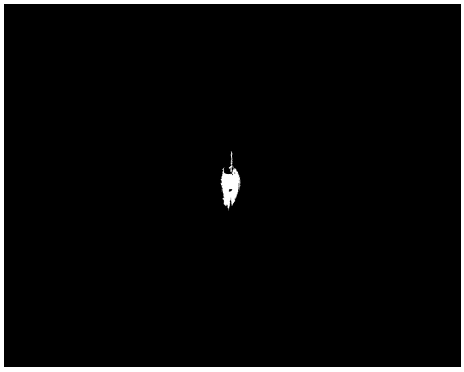


(e) Silhouette of ice from Camera 5



(f) Silhouette of ice from Camera 6

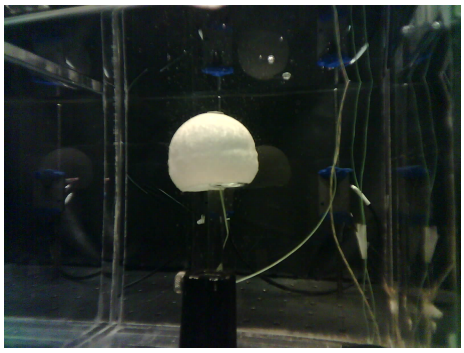
(g) Silhouette of ice from Camera 7



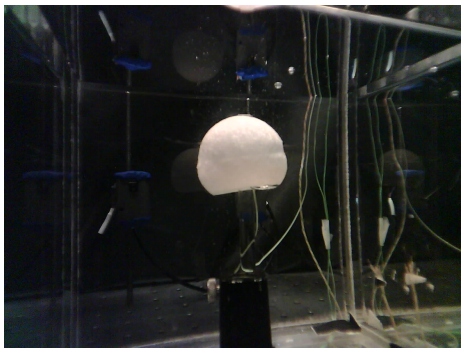
(h) Silhouette of ice from Camera 8

Figure A.4: Ice silhouettes for t = 729 seconds

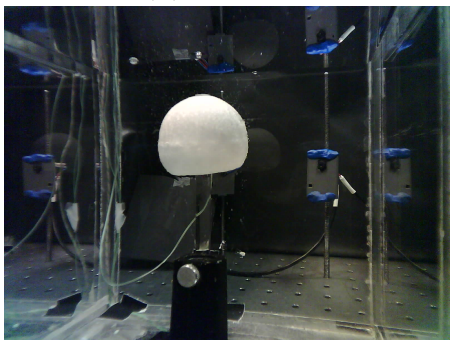
A.3 Test Par5_1; Reconstruction for t = 756 s



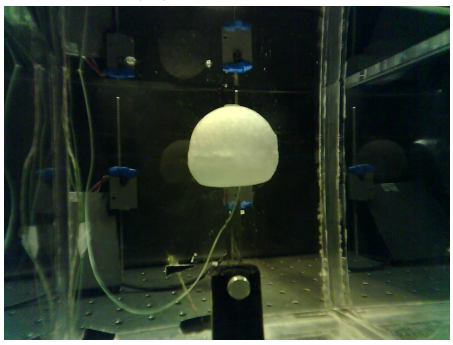
(a) Camera 1



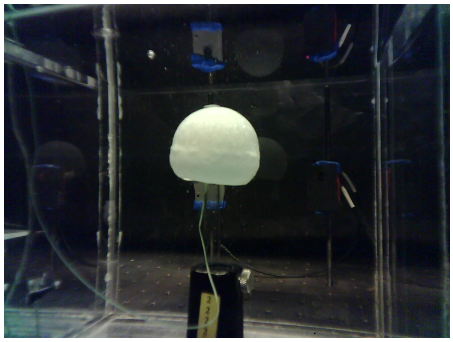
(b) Camera 2



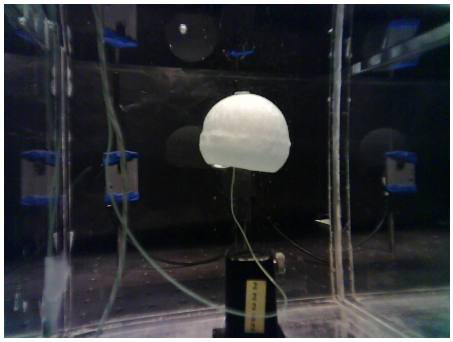
(c) Camera 3



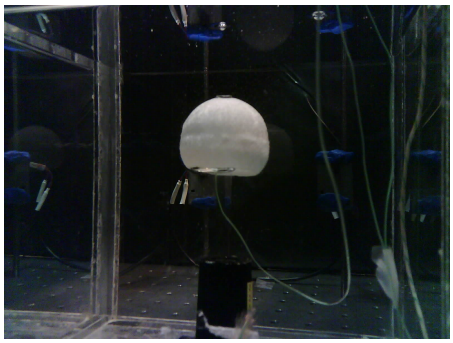
(d) Camera 4



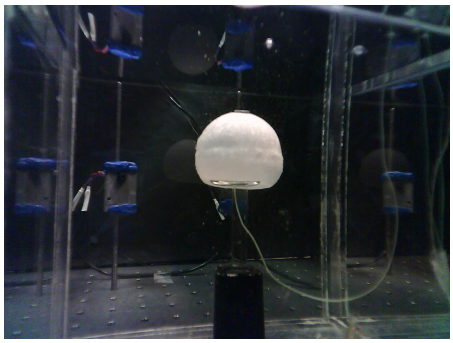
(e) Camera 5



(f) Camera 6

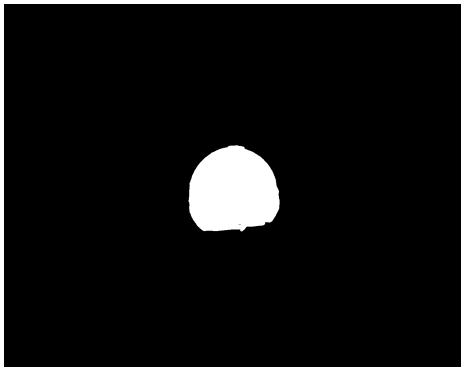


(g) Camera 7

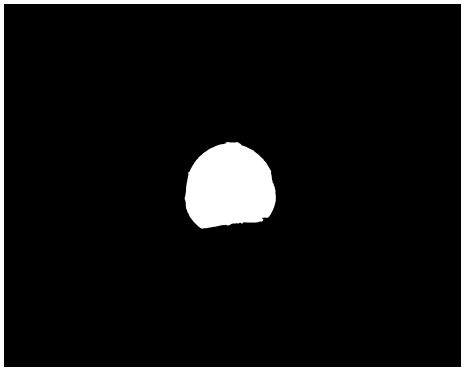


(h) Camera 8

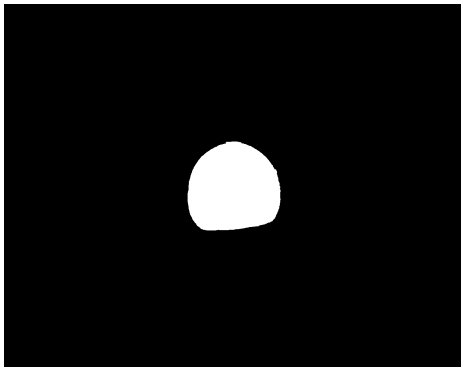
Figure A.5: Original images for t = 756 seconds



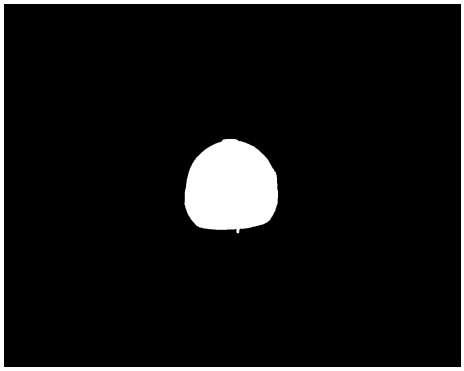
(a) Silhouette from Camera 1



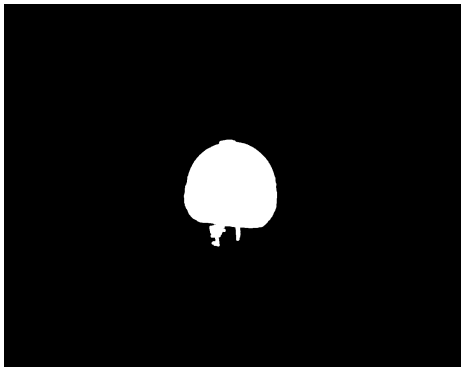
(b) Silhouette from Camera 2



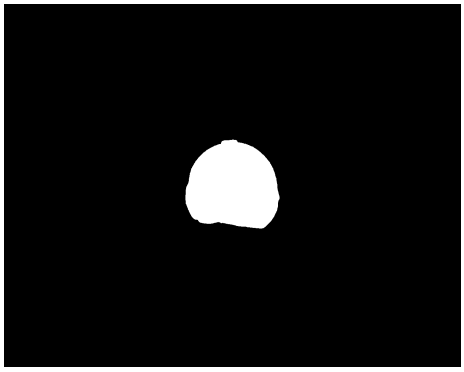
(c) Silhouette from Camera 3



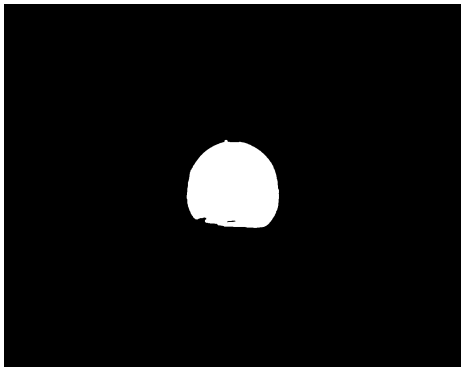
(d) Silhouette from Camera 4



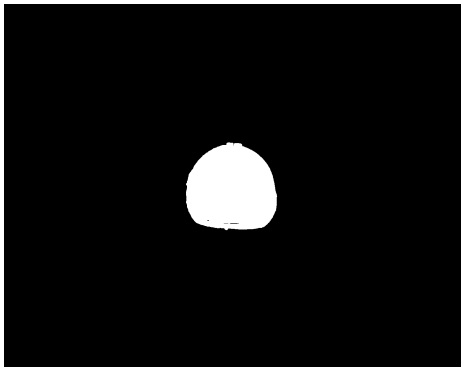
(e) Silhouette from Camera 5



(f) Silhouette from Camera 6



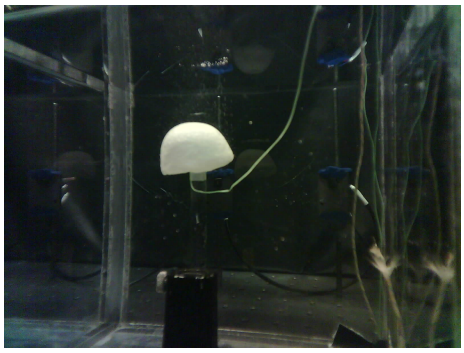
(g) Silhouette from Camera 7



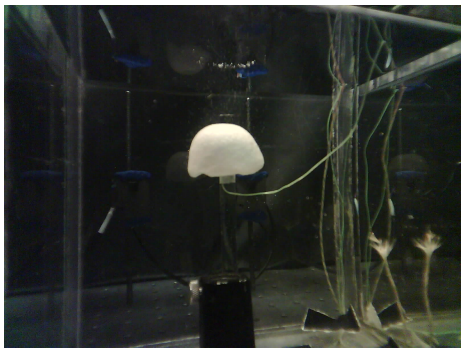
(h) Silhouette from Camera 8

Figure A.6: Silhouettes extraction for RT5HC at t = 756 s

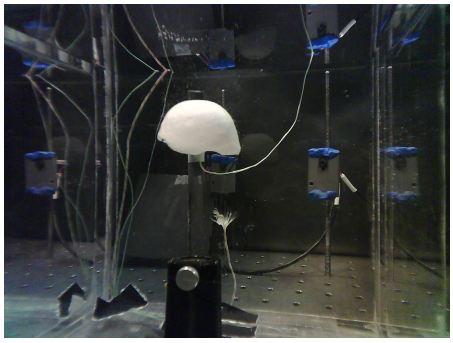
A.4 Test Par-9_1; Reconstruction for t = 270 s



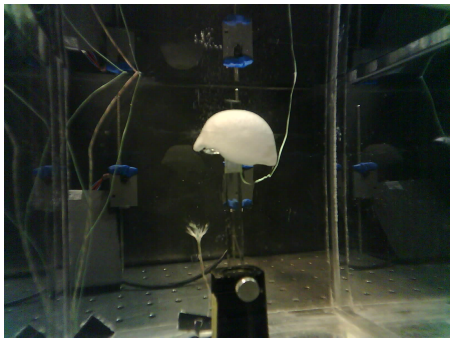
(a) Camera 1



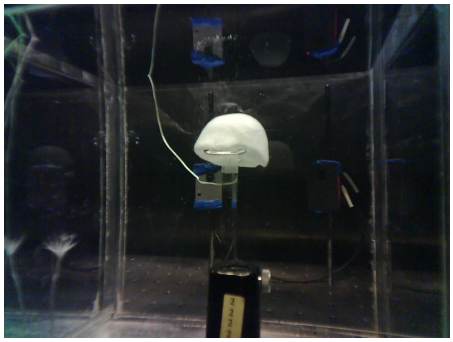
(b) Camera 2



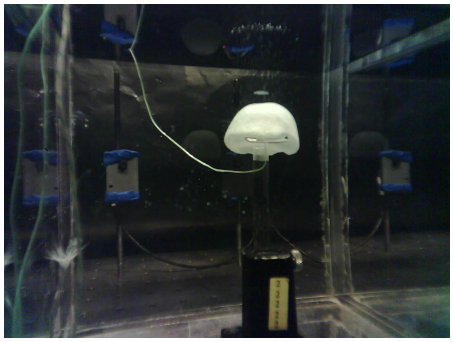
(c) Camera 3



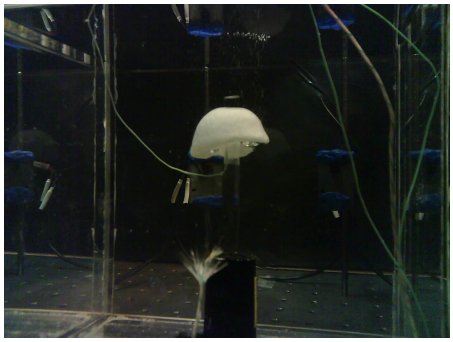
(d) Camera 4



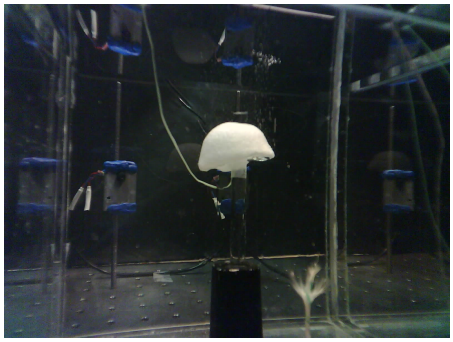
(e) Camera 5



(f) Camera 6

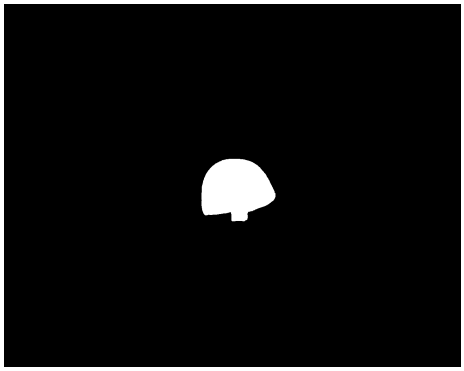


(g) Camera 7

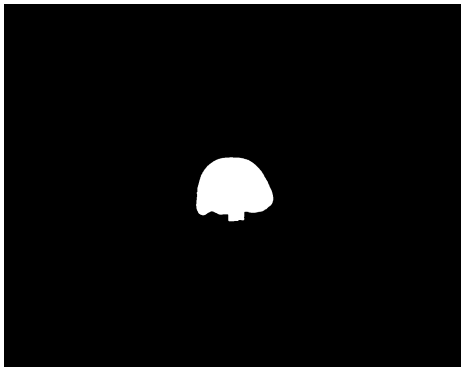


(h) Camera 8

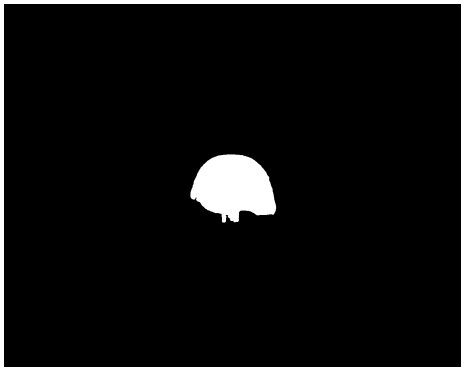
Figure A.7: Original images for t = 270 seconds



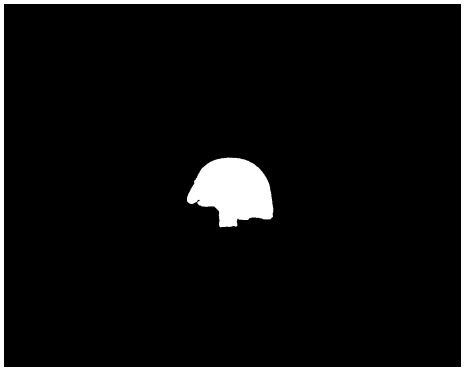
(a) Silhouette from Camera 1



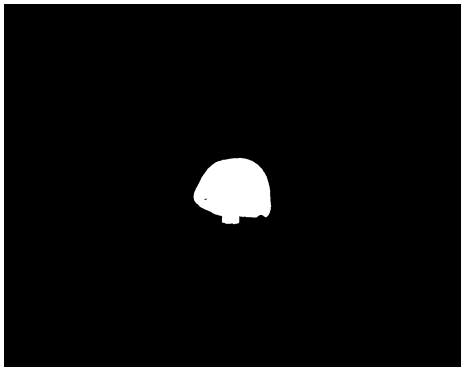
(b) Silhouette from Camera 2



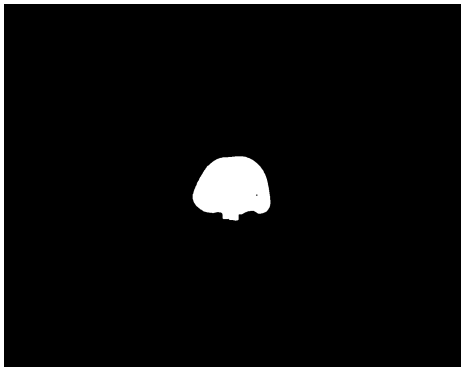
(c) Silhouette from Camera 3



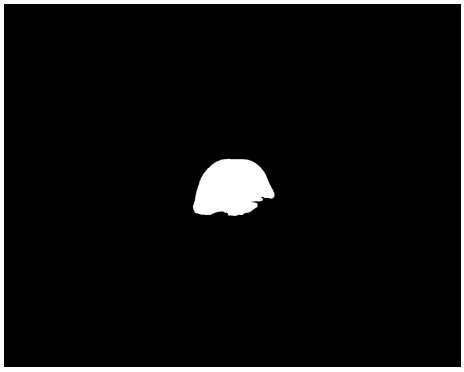
(d) Silhouette from Camera 4



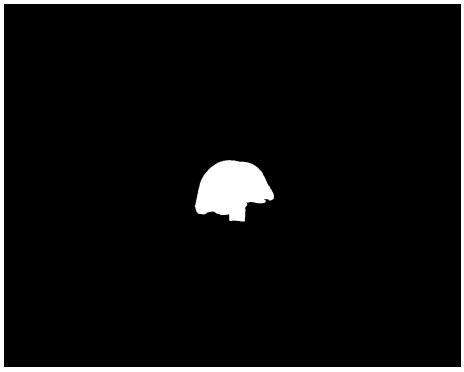
(e) Silhouette from Camera 5



(f) Silhouette from Camera 6



(g) Silhouette from Camera 7



(h) Silhouette from Camera 8

Figure A.8: Silhouettes extraction for RT-9HC at t = 270 s

Bibliography

[1] Vallisa L. *Measurement techniques for 3D shape reconstruction in advanced characterization of densified cryogenic hydrogen*. von Karman Institute for Fluid Dynamics, University of Ghent, 2021 (cit. on p. 2).

[2] Reynier Ph., M. Bugel, and J. Lecoîntre. «Modeling of Slush Flows». In: *40th International Conference on Environmental Systems* (Oct. 2016). DOI: 10.2514/6.2010-6140 (cit. on pp. 2, 5).

[3] Jin T. et al. «Numerical prediction of flow characteristics of slush hydrogen in a horizontal pipe». In: *International Journal of Hydrogen Energy* (Oct. 2016), pp. 1–12. DOI: http://dx.doi.org/10.1016/j.ijhydene.2016.09.054 (cit. on p. 2).

[4] Borri E. et al. «Phase Change Slurries for Cooling and Storage: An Overview of Research Trends and Gaps». In: *Energies* (2022). DOI: https://doi.org/10.3390/en15196873 (cit. on p. 2).

[5] Scelzo M. T. Eneren P. Sakamoto Y. Peveroni L. «Design and validation of a capacitance-based sensor for slurry density measurement». In: *Experimental Thermal and Fluid Science* 122 (2021), pp. 15, 6873. DOI: https://doi.org/10.1016/j.expthermflusci.2020.110299 (cit. on p. 5).

[6] Maria Teresa Scelzo. *Experimental and numerical investigation of slurry flows in pipelines: a contribution towards slush propellants for future rockets' engines*. Université Libre de Bruxelles, 2021 (cit. on pp. 5–7).

[7] et al. Atul Sharma V.V. Tyagi. «Review on thermal energy storage with phase change materials and applications». In: *Renewable and Sustainable Energy Reviews* 13 (2009), pp. 318–34. DOI: 10.1016/j.rser.2007.10.005 (cit. on pp. 9, 10).

[8] C.P. Tso M.Z.M. Rizan F.L. Tan. «An experimental study of n-octadecane melting inside a sphere subjected to constant heat rate at surface». In: *International Communications in Heat and Mass Transfer* 39 (2012), pp. 1624–1630. DOI: http://dx.doi.org/10.1016/j.icheatmasstransfer.2012.08.003 (cit. on p. 9).

[9] Hamid Jahedi Amlashib Babak Kamkaria. «Numerical simulation and experimental verification of constrained melting of phase change material in inclined rectangular enclosures». In: *International Communications in Heat and Mass Transfer* 88 (2017), pp. 211–219. DOI: http://dx.doi.org/10.1016/j.icheatmasstransfer.2017.07.023 (cit. on p. 10).

[10] R. K. Gupta and S. D. Senturia. «EXPERIMENTAL INVESTIGATION OF SOLID-LIQUID PHASE CHANGE IN CYLINDRICAL GEOMETRY». In: *2007 ASME-JSME Thermal Engineering Summer Heat Transfer Conference*. Vancouver, British Columbia, CANADA, July 2007 (cit. on p. 10).

[11] Hamid Jahedi Amlashib Babak Kamkaria. «Experimental and computational study of constrained melting of phase change materials (PCM) inside a spherical capsule». In: *International Journal of Heat and Mass Transfer* 52 (2009), pp. 3464–3472. DOI: 10.1016/j.ijheatmasstransfer.2009.02.043 (cit. on p. 10).

[12] Maria das Graças E. da Silva Kamal A.R. Ismail. «Melting of PCM around a horizontal cylinder with constant surface temperature». In: *International Journal of Thermal Sciences* 42 (2003), pp. 1145–1152. DOI: 10.1016/S1290-0729(03)00093-0 (cit. on p. 10).

[13] Gennady Ziskind Yoram Kozak Tomer Rozenfeld. «Close-contact melting in vertical annular enclosures with a non-isothermal base: Theoretical modeling and application to thermal storage». In: *International Journal of Heat and Mass Transfer* 72 (2014), pp. 114–127. DOI: http://dx.doi.org/10.1016/j.ijheatmasstransfer.2013.12.058 (cit. on p. 10).

[14] Gennady Ziskind Yoram Kozak Tomer Rozenfeld. «An experimental investigation of the melting process in a rectangular enclosure». In: *International Journal of Heat and Mass Transfer* 42 (1999) (cit. on p. 11).

[15] Rubitherm. *PCM RT-line product information*. URL: https://www.rubitherm.eu/en/productcategory/organische-pcm-rt (cit. on p. 11).

[16] Schneider D. *Visual Hull*. URL: https://homepages.inf.ed.ac.uk/rbf/CVonline/LOCAL_COPIES/AV0809/schneider.pdf (cit. on pp. 13, 16).

[17] Baumgart Bruce Guenther. *Geometric Modeling for Computer Vision*. tanford University, 1974 (cit. on p. 13).

[18] Mendez M. *Dynamics of bubble rising in quiescent viscous media*. Master thesis at von Karman Institute for Fluid dynamics, 2012 (cit. on pp. 14, 15).

[19] Aldo Laurentini. «The Visual Hull Concept for Silhouette-Based Image Understanding». In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 16 (Mar. 1994), pp. 150–162. DOI: 10.1109/34.273735 (cit. on p. 13).

[20] Han X. N. Egri G. *Shape-from-Silhouette using Visual Hull and Deep Image Prior*. URL: https://github.com/egrigokhan/deep-visual-hull-prior (cit. on pp. 16–18).

[21] Seitz S.M. Curless B. Diebel J. Scharstein D. Szeliski R. URL: https://vision.middlebury.edu/mview/data/ (cit. on pp. 17–19).

[22] Espressif Systems. URL: https://www.espressif.com/en/products/socs/esp32 (cit. on pp. 22, 23).

[23] *ESP32-Cam mounting for camera arm CR10s Pro*. URL: https://www.thingiverse.com/thing:4301549 (cit. on p. 23).

[24] Fontaneto F. «Temperature Measurements». In: *IMT lecture Series 2022*. von Karman Institute for Fluid Dynamics, Rhode Saint-Genèse, Brussels, Belgium, 2022 (cit. on p. 27).

[25] LearnOpenCV. *Camera Calibration using OpenCV*. URL: https://learnopencv.com/camera-calibration-using-opencv/ (cit. on pp. 33, 36, 37).

[26] Savarese S. Hata K. *Course Notes 1: Camera Models*. URL: https://web.stanford.edu/class/cs231a/course_notes/01-camera-models.pdf (cit. on pp. 34, 37).

[27] Scratchapixel. *3D Viewing: the Pinhole Camera Model*. URL: https://www.scratchapixel.com/lessons/3d-basic-rendering/3d-viewing-pinhole-camera/how-pinhole-camera-works-part-1.html (cit. on pp. 34, 37).

[28] Krishna N. *Image Formation and Pinhole Model of the Camera*. URL: https://towardsdatascience.com/image-formation-and-pinhole-model-of-the-camera-53872ee4ee92 (cit. on p. 35).

[29] van den Boomgaard R. *Lecture Notes Image Processing and Computer Vision*. URL: https://staff.fnwi.uva.nl/r.vandenboomgaard/IPCV20162017/LectureNotes/CV/PinholeCamera/index.html (cit. on p. 37).

[30] *Rotation Matrix*. URL: https://www.cuemath.com/algebra/rotation-matrix/ (cit. on pp. 37, 38).

[31] *Coordinates Transformations*. URL: http://motion.pratt.duke.edu/RoboticSystems/CoordinateTransformations.html (cit. on p. 38).

[32] Guido Straube et al. «Modelling and calibration of multicamera-systems for 3D industrial supervision applications». In: *Photonics and Education in Measurement Science* (2019). DOI: 10.1117/12.2532320 (cit. on p. 39).

[33] Yi Ma Ziran Xing Jingyi Yu. «A New Calibration Technique for Multi-Camera Systems of Limited Overlapping Field-of-Views». In: *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)* (2017). DOI: 10.1109/IROS.2017.8206482 (cit. on p. 39).

[34] T. Pajdla T. Svoboda D. Martinec. «A Convenient Multi-Camera Self-Calibration for Virtual Environments». In: *Teleoperators and Virtual Environments* (2005) (cit. on p. 39).

[35] You Yang Firas Abedi and Qiong Liu. «Group geometric calibration and rectification for circular multi-camera imaging system». In: *Opt. Express* 26 (2018), pp. 30596–30613 (cit. on p. 40).

[36] Cornelius Lilge Ferenc Kahlesz and Reinhard Klein. «Easy–to–Use Calibration of Multiple–Camera Setups». In: *Proceedings of the ICVS Workshop on Camera Calibration Methods for Computer Vision Systems* (2007) (cit. on p. 40).

[37] Zivid. *Multi Camera Calibration Theory*. URL: https://support.zivid.com/en/latest/academy/applications/multi-camera-calibration/multi-camera-calibration-theory.html (cit. on p. 40).

[38] Bouguet J. Y. *Camera Calibration Toolbox for Matlab, 2010*. URL: https://data.caltech.edu/records/jx9cx-fdh55 (cit. on p. 40).

[39] et al. Liu K. «Auto calibration of multi-camera system for human pose estimation». In: *IET Comput. Vis.* 16 (2022), pp. 607–618. DOI: https://doi.org/10.1049/cvi2.12130 (cit. on p. 41).