Development of an Ice Crystal Generator and an Ice Crystal Impact Analysis Tool

Alexandre Miguel Correia Cuco
alexandrecuco@tecnico.ulisboa.pt
Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

July 2020

Abstract

The aviation industry is an ever-developing branch of engineering. Constant demands for efficiency and fuel saving mandate improvements in the various systems comprising an airplane. Among them, aerodynamic efficient surfaces play a significant role for achieving this goal. To better develop these structures, phenomena such as ice accretion must be firstly studied in order to understand the mechanisms which impact these systems' efficiency.

At the Aerodynamic Efficient Surfaces Laboratory at Airbus Central R&T in Ottobrunn, Germany, the science boundaries are constantly redefined to ultimately build safe and sustainable flying transportation. This thesis is inserted in the study of the physics behind ice accretion done in these facilities, more specifically at the iCORE icing wind tunnel.

As to better study ice accretion and the impact of these particles on metal surfaces, a new device for ice crystal generation was built to deliver a constant ice crystal mass flow in the wind tunnel. Two prototypes were built, based on two different techniques: 1) freezing of ice crystal on a nitrogen cooled drum and 2) ice scraping of an ice block. The required mass rates and size requirements of the ice crystals were achieved.

Moreover, a software was developed using computer vision to facilitate the study of ice crystal impacts at a microscopic level. The small scale of the tested system and high levels of noise posed great challenges on the automatization and processing of the recorded videos. The resulting software has a user-friendly GUI, having been developed with a compromise between automatization and user input to validate the results obtained.

1 Introduction

Safety and fuel energy efficiency have always been two of the most relevant pillars for commercial aircraft design. Consequently, it is the interest of aerospace manufactures to develop and research systems that comply with safety and certification standards and allow the construction of even safer and more efficient aircraft. These machines are subject to extreme weather conditions, such as atmospheric icing conditions. In these operational environment, collision of the aircraft with microscopic supercooled water droplets[1] and/or ice crystals may occur and lead to in-flight icing, which drops aircraft performance.

The aim of this thesis is to develop a system capable of delivering a constant flow of ice crystals inside an icing wind tunnel and the subsequent analysis of the ice crystal impacts using computer vision. The icing wind tunnel is located in the iCORE test facilities at Airbus in Ottobrunn and aims to support the understanding of icing phenomena on various surfaces and structures of aircraft.

Figure 1.1 - Icing in aircraft turbofan engine. Picture from Yeong et Al. [1].

For the generation of ice crystals, it is necessary to design and build an inexpensive device that can deliver the required ice crystal size distribution and mass flow in wind tunnels.

The analysis of the impacts requires a vast amount of variables to be identified from the recordings of a high-speed camera. In order to identify and characterize the crystals and their impacts on a solid surface, an automated method was developed to collect the data from all relevant impacts.

[1] Supercooled water droplets are liquid water droplets whose temperature can range from 0 to -40°C. These droplets remain in a liquid state due to the lack of suspended nucleation particles in the atmosphere or the lack of impacts between droplets (heterogeneous nucleation). Below -40°C, water freezes spontaneously (homogeneous nucleation).

Table 1.1 - Different ice crystal production techniques and qualitative description of the output crystals relative to their applicability in study of microscale ice crystal impacts.

Ice production techniques	Ice crystals' shape	Cost	Implementa-tion	Ice production rate	Size of ice crystals	Example of usage
Ice Cloud Chamber	Identical to natural ones	High	Hard	Very Low	Identical to natural ones	Manchester Ice Cloud Chamber [2]
Ice Grinding	Different to natural ones	Low	Easy	Very high	Too large	GKN Icing Research Wind Tunnel
Snow Cannon	Round with high LWC	Average	Easy	High	Good size distribution	Cox Icing Wind Tunnel [3]
Scraping a freezing Liquid Water film	Different to natural ones	Average	Hard	Average	Too large	Food industry
Accretion of supercooled droplets	Different to natural ones	Low	Average	High	Good size distribution	Technische Universität Braunschweig

This process will consist of a computer vision based Python program, which will take the footage from the tests and output the relevant information for each impact.

Overall, this thesis seeks to broaden the knowledge of high-speed microscopic ice crystal impacting. Not only individual ice crystal impacts are characterized, but also their influence on the long-term development of the ice accretion on aircraft surfaces. With this, aircraft icing research will be advanced and the development of a numerical model and prediction tool for ice crystal icing will be enabled.

2 Fundamentals and State of the Art

This chapter outlines the state of the art of ice crystal generation techniques. Different working principles and physical behaviors are used to obtain specific objectives.

2.1 In-flight Atmospheric Conditions

Aircraft are subject to a great spectrum of atmospheric weather conditions. Commercial flights have a cruising altitude of typically between 31,000 and 38,000 feet (roughly 9,5 and 11,5 km) where Outside Air Temperatures (OAT) can reach -60°C. At these OAT, water can only be present in the form of ice crystals, as supercooled water cannot be present at temperatures below -40°C [4]. Clouds with high concentration of Ice Water Content (IWC) and low Liquid Water Content (LWC) were thought to pose few threats to an aircraft, as ice crystals would simply bounce off the cold fairings. In most cases, this holds true. However, a different process occurs inside jet engines or on heated aircraft surfaces, which will be described in Chapter 2.2.

2.2 Ice Particle Impacts and Ice Accretion

In-flight aircraft icing can be narrowed down to two main processes: collision with supercooled water droplets or collision with microscopic ice crystals. The latter phenomenon, which is the focus of this thesis, until recently was considered to not pose a threat to aircraft: ice particles were believed to bounce off engine and airframe surfaces. However, according to a database from Boeing, more than 160 engine icing events of commercial aircraft occurred since the 1990s could be traced back to ice crystals. As a jet engine aircraft flies through clouds with large IWC, High Altitude Ice Crystals (HAIC) hit the blades of the inlet fan and causes fragmentation of the crystals into even finer particles. Part of these particles are ingested into the compressor, whereas the remainder flow around the compressor (see Figure 2.1). Due to aerodynamic pressure the former particles are then pushed to the first stages of the compressor (Low Pressure Compressor) where temperatures are close to 0°C (ICI conditions).

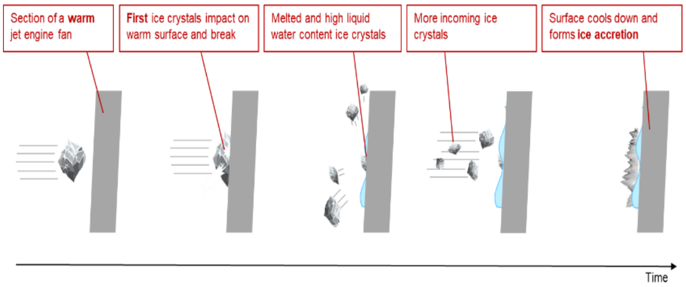

Section of a **warm** jet engine fan
First ice crystals impact on warm surface and break
Melted and high liquid water content ice crystals
More incoming ice crystals
Surface cools down and forms **ice accretion**
Time

Figure 2.1 – Schematics of ice crystal accretion on a warm jet engine fan.

As the first crystals impact onto the warm compressor blades, part of the crystals melt and leave back a trail of liquid water. The following crystals will accrete when colliding and sticking to the liquid water layer or the partially melted crystals. As this process reiterates in the various stages of the Low Pressure Compressor, large agglomerates of ice can form and disrupt the air flow inside the compressor, triggering efficiency losses. Later, due to aerodynamic pressure, these agglomerates detach from the compressor fan blades and follow the path of the compressed air into the High Pressure Compressor. Here, the ice can damage the blades and ultimately cause a *compressor surge*, i.e. breakdown of the whole compression which reverts the flow of air and may damage the compressor blades and vanes. Following these events, a complete flameout of the engine may occur and attempts to re-light the engine must be performed.

2.3 Ice Production Techniques

Ice crystals come up in different sizes and shapes depending on their growth process that is affected by the atmospheric conditions like temperature and supersaturation, which give the crystals different densities and shapes.

Initially, crystals have pristine shapes, however various aggregation mechanisms may also play a role and larger agglomerates of ice crystals may occur. Understanding the natural process for ice crystal formation is essential for designing artificial ice crystal generation. Shape, size and mass are the main features considered when choosing an ice crystal generation technique. Table 1.1 shows some methods for creating artificial ice crystals and their qualities for wind tunnel usage.

2.4 Object Tracking Software

While tracking an object in a video is a trivial task for the human eye, for a computer, a video is merely an array of numbers changing in time. A digital image is a numeric representation array of a two-dimensional image. Each image is divided in pixels, which carry an intensity value depending on the brightness of a given color. Object tracking is a process used to identify and track the position of an object over time using a camera. This process can be done manually but some applications such as road traffic control, surveillance or particle tracking can be too time consuming, hence the use of a software to aid this task. For this thesis, a contour detection method for identifying shapes followed by a custom algorithm to track the particles were used. For this, it is firstly necessary to process the acquired images.

2.4.1 Image processing

Each digital image composing the video must be processed or filtered in order to enhance specific features or to remove unwanted noise[2]. For particle detection, it is necessary to remove noise from raw images, such as the shadows produced by dust on the lenses or the windows of the test section, which could be interpreted as particles. A method called Bilateral Filtering was chosen [5]. With this method the edges of an object are less affected to blurring. This method is essential for ruling-out particles that are out of focus, which will be described later. A threshold method was used to binarize the images in order to have black and white images ready to be used by the contour detection algorithm. Due to noise and reflections of light on the crystals, a good threshold value must be found in order to eliminate unwanted noise while keeping the shape of the crystal mainly intact, as seen on Figure 2.2.

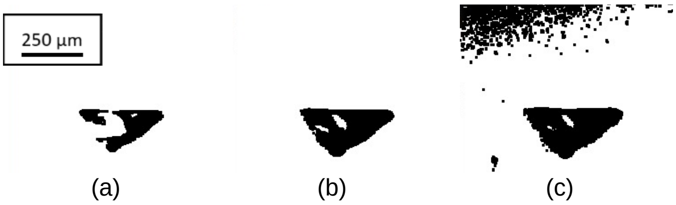


Figure 2.2 – Threshold values from high to low of an ice crystal.

After this method, morphological transformations are implemented in order to reduce the remaining noise [6].

3 Materials and Infrastructure

This chapter describes the tools and materials used in the thesis.

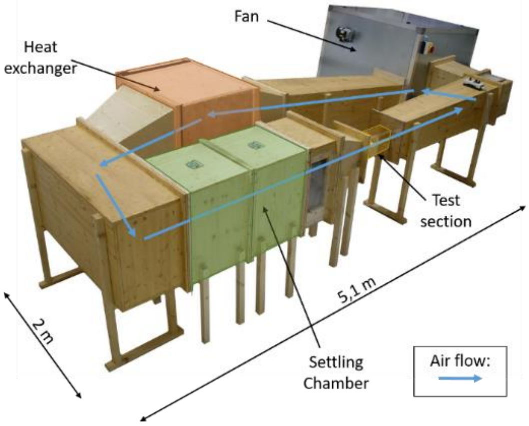


Figure 3.1 - iCORE icing wind tunnel.

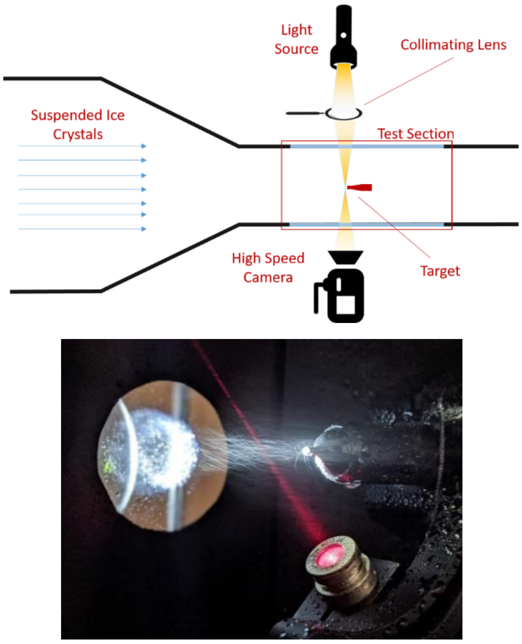


Figure 3.2 – Simplified top view of the iCORE wind tunnel test section (top) and detail of the target (bottom).

[2] Noise is a random variation of brightness or color information in an image. In this thesis, small particles on the windows of the wind tunnel or the lenses of the camera will also be encompassed in this category.

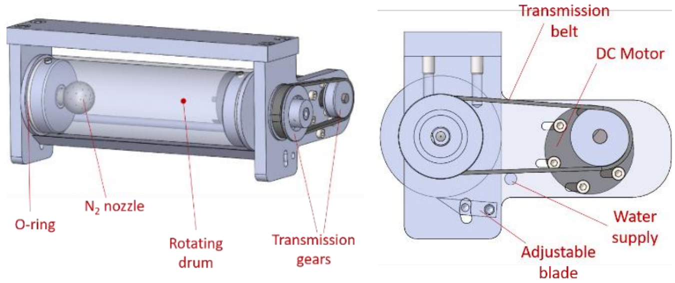
All the experiments took part in the iCORE Laboratory test facilities at Airbus Central R&T in Ottobrunn. Here, an icing wind tunnel was used to perform the necessary experiments. The experimental setup consists of four independent systems: (1) the icing wind tunnel (Figure 3.1), (2) the high-speed recording setup, i.e. the high-speed camera, the lighting and the triggering unit (Figure 3.2), (3) the ice crystal generator (which was developed in this thesis) and (4) a heated target (Figure 3.2). Various airspeeds and temperatures, target temperatures, and ice crystal flow rates can be set, while maintaining the same recording setup.

4 Hardware and Algorithm Development

4.1 Ice Crystal Generator

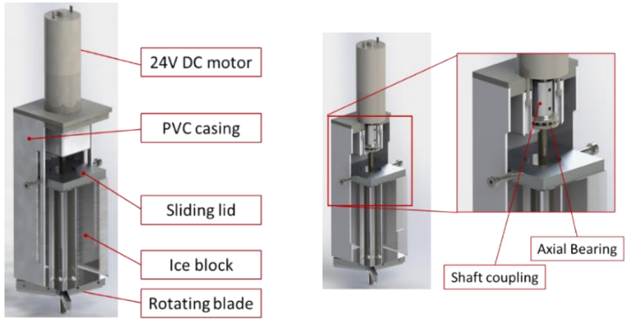
This chapter describes both the hardware and software developed aiming to create a constant ice crystal flow inside the iCORE wind tunnel and the analysis of these crystals to complement the previously described materials and infrastructure.

A study of water atomization with liquid nitrogen using lump system analysis was made and it was concluded that this method could not be used for creating a constant flow of ice crystals. This was followed by a study of ice formation on a cold surface similar to the one used in Cryopelletization [7]. This study showed that it is possible to generate ice in the quantities necessary for the application at the iCORE wind tunel. A first prototype was designed, show on Figure 4.1. The system works by inserting a flow of nitrogen through the nitrogen nozzle and the drum is powered by an igus 24VDC motor which is connected by a belt between two gears. A liquid water spray is positioned just after the adjustable blade in order to maximise the time of contact between the aluminium drum and the sprayed water. A second prototype was designed based on a simpler method where a solid block of ice is shaved by a rotating blade, a similar technique used in other icing wind tunnels. This device is comprised of a PVC tube with a square section that keeps the block of ice from rotating due to the torque of rotating shaving blades below. An axle links the motor on top with the water jet blade holder at the bottom, where titanium blades with teeth shave the ice, as shown on Figure 4.2.



(a) (b)

Figure 4.1 – CAD of the final iteration of the first prototype.



(a) (b)

Figure 4.2 - Render of the second prototype. (a) cut view of the outside PVC casing and (b) cut view of the inner casing holding the axial ball bearing and the shaft coupling for the rotating axle.

Thin titanium blades were used to vibrate while shaving the ice in order to eject the ice as it was formed on the surface, thus preventing it from melting on the surface. Two steel blocks are fixed to the sliding lid providing constant pressure on the ice as it is being shaved. The whole device is place on the top of the settling chamber of the wind tunnel.

4.2 Ice Crystal Detection and Tracking

The development of the ice crystal analysis Python program is now described. Two of the main libraries used were OpenCV [8] and Matplotlib [9]. Particle detection was performed by using the OpenCV function findContours(), which takes a binary image and recognizes the borders between black and white pixels, outputting an array of point coordinates which correspond to a detected objected [10]. Image transformations such as blurring, opening and contrast and brightness were performed in order to achieve binary images from the raw recorded grayscale images.

The testing takes place in a three-dimensional space, hence opening the possibility of having objects which are not inside the narrow depth of field provided by the Long Distance Microscope. Due to focal depth being so narrow – roughly the width of the target, 1 mm - most of the particles which fly outside of it are not recorded.

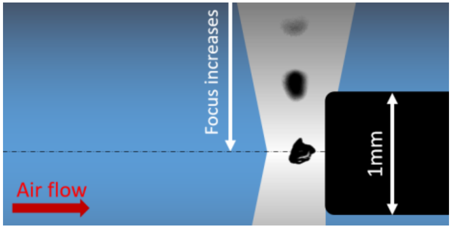


Figure 4.3 – Top view of a representation of the focal distance with examples from recorded crystals. The grey band represents the influence of the light source, while the black shape on the right represents the target. The figure is not to scale.

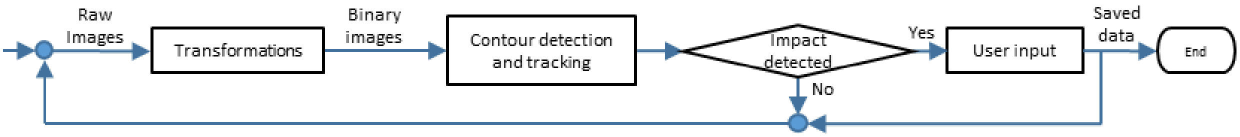


Figure 4.4 - High level flowchart of the ice crystal detection, tracking and characterization. The software keeps running until the whole video is analysed or the user ends the characterization.

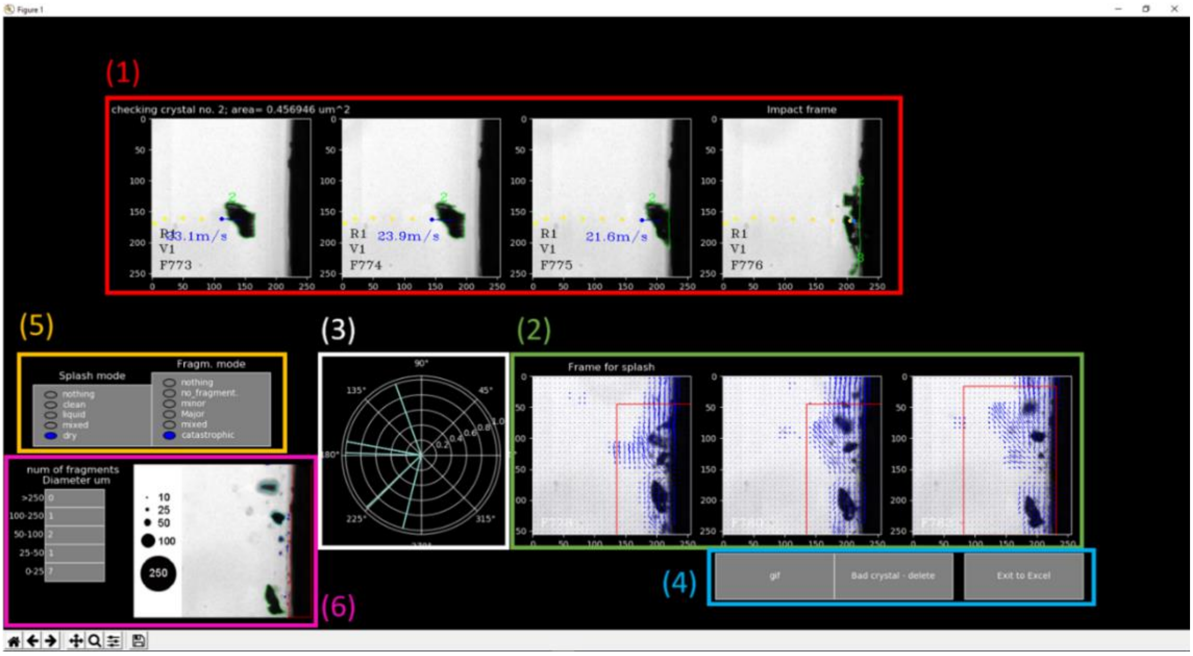


Figure 4.5 – Screenshot of the GUI. (1) frames until impact; (2) frames after impact with optical flow overlaid; (3) optical flow directions; (4) buttons; (5) radio buttons; (6) Post impact size distribution.

Nonetheless, some particles may fly close the focal plane, but miss the target, which is perceived as a blurry shadow flying across the frame, as seen on Figure 4.3. Therefore, it is of great interest to quantify how much in focus a picture is. This was done by using the Laplacian operator.

Identifying an object is a crucial step in computer vision, but in this case it brings little information if made without tracking. It was necessary to identify an object not only in space but also in time, i.e. a trajectory path must be recorded for each object. Recording high-speed micro ice crystals pose many challenges, such as limited frame rates and resolutions. Furthermore, the cheer speed at which the ice crystals move implies that before impact, an ice crystal can appear in just two or three frames upon impacting the target.

Before being run through the impact detection algorithm, whose position on the software is shown on Figure 4.4, each frame is run through the findContours() function. The area is computed for each contour, and the ones which are smaller than a threshold of 5 pixels are discarded as they are usually noise. This simple step is a very effective way to remove noise and a complement to the morphological transformations. Briefly, the developed tracking algorithm resembles a *greedy algorithm.* Each newly identified contour, j, is run through a sequence of functions, $f_{i,c}$, where each has a weight, a_i, associated to it. These functions evaluate the probability of the newly identified contour to be in fact a crystal, c, which was previously identified but in a following frame. After a contour is run through these functions, the values are summed and a final score, $S_{j,c}$, is reached and stored in an array. The value $S_{j,c}$ corresponds to the likeability of the contour to be in fact a crystal which was previously identified but in a following frame.

After the contour being compared with all the previous identified crystals, the lowest score of $S_{j,c}$ will dictate whether the contour j is a crystal c, hence the resemblance to a *greedy algorithm.*

One of the most import values to take from this analysis is the outcome of each impact on the target. Due to the limitation of resolution, frame rate of the camera and the high velocities of the resulting debris, it often difficult or impossible to track every individual out coming particle resulting from an impact. Due to the importance of these values, and in order to keep the characterization automation to the maximum, a different technique from the previously described particle tracking had to be applied. An Optical Flow [11] approach was chosen to tackle this problem. Due to the large amount of data which had to be recorded for each crystal, a Class named Crystal was created. From Python Documentation [12], a class is defined as “a means of bundling data and functionality together. Creating a new class creates a new *type* of object, allowing new *instances* of that type to be made”.

These sentences can be broken down into two main aspects: object and instances. Taking the context of this thesis, the object can be considered as a blueprint of a crystal, while an instance is each individual crystal which is created by an object. Every time a new crystal is identified, a new instance of the class Crystal is created and with it, attributes are created in order to preserve the necessary information for each crystal.

Due to the large amount of data which had to be recorded for each crystal, a Class named Crystal was created. From Python Documentation [12], a class is defined as "a means of bundling data and functionality together. Creating a new class creates a new *type* of object, allowing new *instances* of that type to be made." These sentences can be break down into two main aspects: object and instances. Taking the context of this thesis, the object can be considered as a blueprint of a crystal, while an instance is each individual crystal which is created by an object. Every time a new crystal is identified, a new instance of the class Crystal is created and with it, attributes are created in order to preserve the necessary information for each crystal.

A GUI was build in order to give the user a feedback of the output of the software, allowing modifications to be made, when necessary (see Figure 4.5). For each impact detected, the program automatically stops and prompts the user with a window generated with the Matplotlib library [9]. In this window there are six key elements: (1) frames before the impact with the path of the tracked crystal; (2) frames with the Optical Flow representing the direction of the particles after the impact; (3) polar graph where the directions of the particles after the impact are represented; (4) buttons for different purposes, (5) radio buttons for impact characterization; (6) Impact fragments interactive size distribution.

Accretion on the target is one of the most relevant aspects to be measured. The rate of accretion is variable, and it is important to find a correlation for number of impacts, types of crystal and rate of accretion. For this, the first 5 binary frames of a video are averaged and used as a reference for the following frames. Finally, the results are exported to excel using the library of functions XlsxWriter.

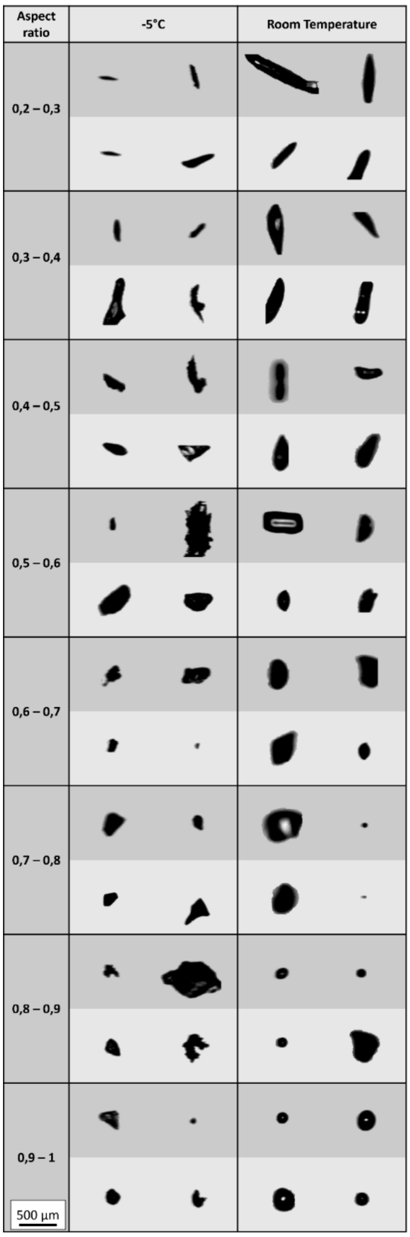
5 Results and Discussion

This thesis intends to find solutions for creating a flow of ice crystals in the iCORE wind tunnel and to provide with a tool for characterizing ice crystal impacts at a micro scale. The following data gives an insight of the capabilities of both systems and how they were developed and work together.

5.1 Ice crystals formation techniques and built prototypes

Due to some limitations, it was not possible to use liquid nitrogen on the first prototype. Cooled nitrogen gas was used as a coolant, impacting the freezing time and rate of the ice produced and unsatisfactory results were achieved. Several spraying techniques, various shaving methods for the ice and a more powerful motor was used, however the ice produced had a big LWC and was not produced in sufficient quantities. The impossibility of using liquid nitrogen as a coolant was a big drawback in this project.

Table 5.1 - Examples of crystals with different true aspect ratios at RT and at -5°C. The software automatically stored the images and the background was manually removed.


Aspect ratio
-5°C
Room Temperature
0,2 – 0,3
0,3 – 0,4
0,4 – 0,5
0,5 – 0,6
0,6 – 0,7
0,7 – 0,8
0,8 – 0,9
0,9 – 1
500 µm

The second prototype used premade ice, which eliminated the high LWC problem.

An adaptation of the computer vision program was used to characterize the ice produced by the second prototype. Ice crystal mass flow rates, shape, speed were evaluated at different outside air temperatures, wind speeds and blade rotation speeds. 19.766 crystals were automatically recorded and analyzed, and a database was created.

It was seen that the biggest mass ice flow was created when the motor was supplied with 18V and the average equivalent diameter of the ice crystals stood at approximately 139,5 µm (see Table 5.2), which is inside the required size range. From Table 5.3 and Table 5.4, it can be seen that OAT and airspeed have little effect on ice mass flow or the ice crystals average equivalent diameter. This dimension was computed by taking the area of each crystal and approximating it to a projected area of a sphere. The diameter of this sphere would be the equivalent diameter.

Table 5.2 – Ice mass flow and equivalent diameter general values for different input voltages.

Voltage input [V]	Ice mass flow range [mg/sec]	Average equivalent diameter [µm]
18	7,50 - 9,03	139,5
14	4,42 - 9,73	153,7
10	3, 09 - 4,12	123,1

Table 5.3 – Ice mass flow and equivalent diameter general values for different OAT.

Temperature	Ice mass flow range [mg/sec]	Average equivalent diameter [µm]
RT	3,09 - 9,03	150,1
-5°C	3,46 - 9,73	132,4

Table 5.4 – Ice mass flow and equivalent diameter general values for different airspeeds.

Airspeed [m/s]	Ice mass flow range [mg/sec]	Average equivalent diameter [µm]
50	4,12 - 9,03	125,7
75	4,42 - 9,73	153,8
100	3, 46 - 7,50	143,6

Later, the aspect ratio (Figure 5.1) and ISO circularity were calculated to evaluate the possibility of linking a liquid water content on the ice crystals with their shape. It was observed that crystals flying at warmer temperatures were rounder (Table 5.1), however there was a small linkage between these two parameters. This second prototype produced ice crystals with the required size and mass flow.

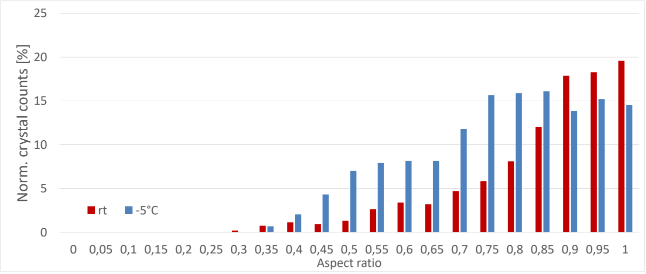


Figure 5.1 - True aspect ratio for the measured crystals at RT and -5°C OAT.

5.2 Python Program Development

The development of the software went through many iteration stages, going from a simple automatic characterization of one crystal impact to a complex program which tracks the development of the impacts through time while presenting a GUI for validation of the automatic analysis.

The successive transformations performed on the images were previously described on Chapter 2.4.1 and one result can be seen on Figure 5.2 where an ice crystal is correctly identified.

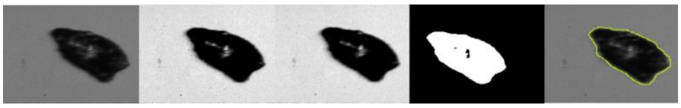
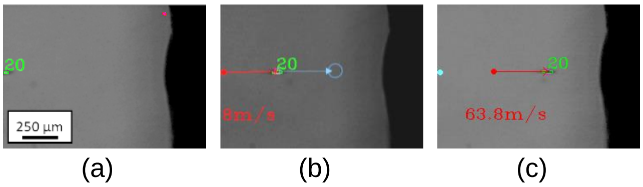


Figure 5.2 - Image transformations from the original frame (left) until the final object detection on the right.

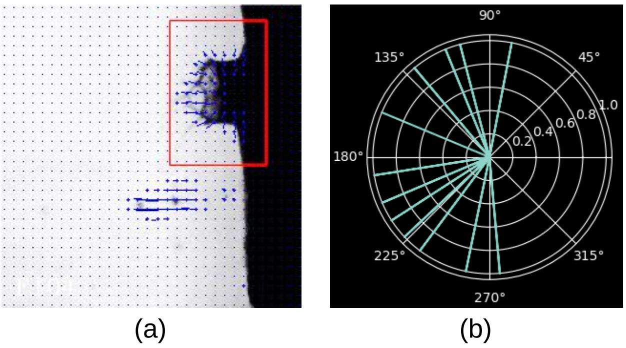
Particle tracking and impact detection proved to be the biggest challenge in this thesis. While contour detection was a great method for identifying the crystals and most of the resulting particles of the impacts, it was challenging to label these shapes and detect when an impact occurred.



(a) (b) (c)

Figure 5.3 – Three consecutive frames where the ice crystal with id. number 20 is recorded. The blue arrow on (b) was added to display the predicted position of the crystal in the following frame (c); Note: on the top right corner of (a), one can see one shape which was correctly discarded for falling below the threshold of the minimum area.

The algorithm for tracking incoming crystals (Section 4.2) provided good results (see Figure 5.3), but was not good for tracking the resulting debris from impacts. Here, the combination of object detection, optical flow (Figure 5.4) and the GUI provided the best results for the analysis.



(a) (b)

Figure 5.4 – (a) Optical flow representation on top of the original frame where a crystal impact created a large water splash and (b) representation of the directions on a polar graph of the interest area in red.

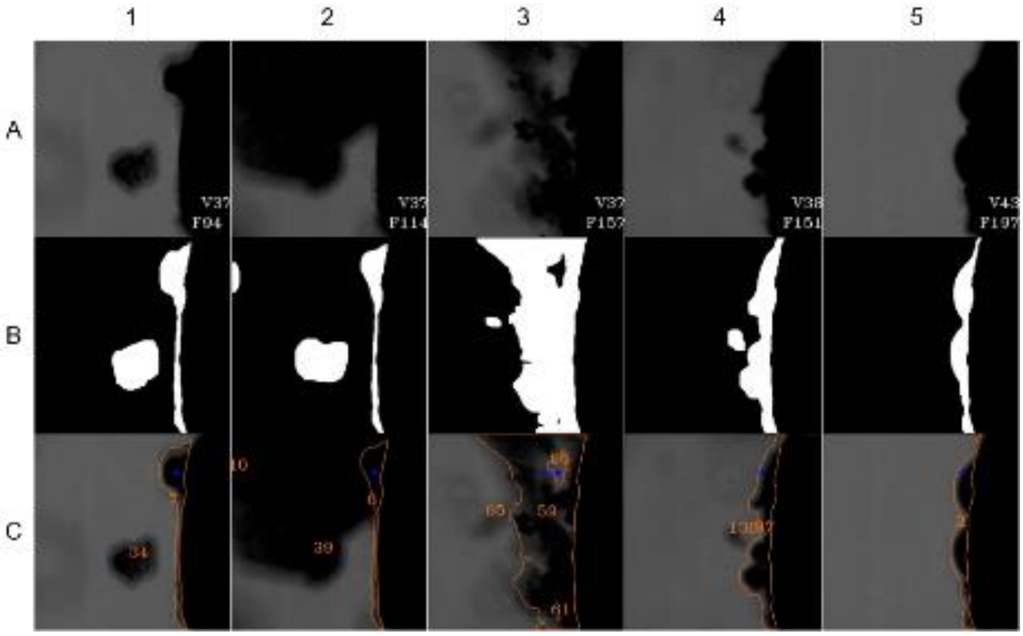
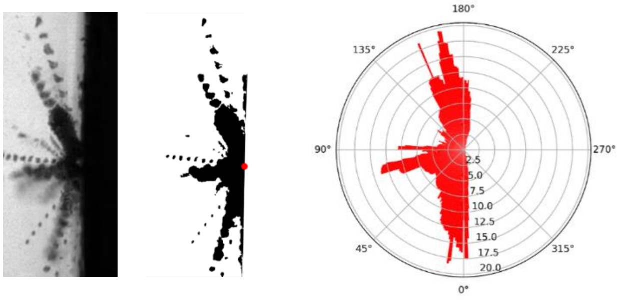


Figure 5.5 - Image Sequence of ice accretion projection measured by the automatized analysis. 5 images (1-5) show the original frame (A), the particle detection mask (B), and the original frame superposed with the determined ice accretion projection (C)

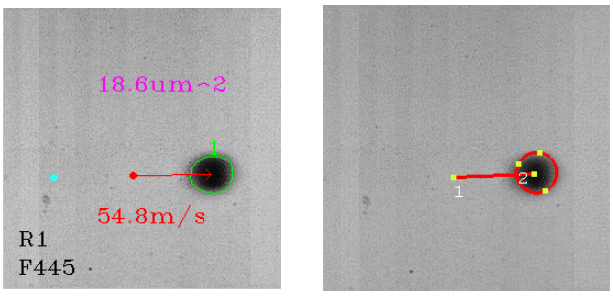


(a) (b) (c)

Figure 5.6 - Stacked image of post impact fragments of ice crystal impact (a), processed image with illustrated impact position (b) and final plot of mean velocities (c).

Table 5.5 – Average of the absolute difference between velocities and diameters measure both manually and using the software.

Airspeed [m/s]	Num. of crystals analysed	Δ velocity [m/s]	Δ diameter [µm]
25	37	0,077344734	0,603017052
50	39	0,093735157	0,858298976
100	18	0,08662385	0,495165198



(a) (b)

Figure 5.7 - Example of (a) automatic and (b) manual analysis, where the manal analysis outputed an area of 17,9 µm and a velocity of 55,6 m/s.

Finally, an overlapping of masks after impact was done to identify the resulting mass and speed of the particles. The impact points are taken from the analysis and corrected in the GUI when necessary and the coordinates of each impact and frame are exported to a Pickle file which is then read by an independent separate script after the analysis was done. This script averages the masks after an impact in order to obtain a map of the black pixels which represent a fragment. Then, vectors are computed to connect each black pixel to the impact point and the angles are measured. An example of resulting plot is shown on Figure 5.6. The velocities of each particle are measured by taking the distance of each black pixel and dividing by the number of frames after the impact. This process is done multiple times and the values are averaged

It is clear to see that this analysis provides a trustworthy and precise analysis of an impact by correctly plotting both directions and velocities on the polar chart on Figure 5.6 (c).

Accretion on the target was correctly identified and measured, as seen on Figure 5.5.

The code went through many stages of optimization where memory leaks, non-releasing of variables and other minor issues were solved to speed up the process.

5.3 Dimensions and results reliability

Having an automated program making measurements raises the question whether the results are trustworthy. To evaluate this issue, the Photron Fastcam Viewer (PFV4) Software was used to measure the sizes and speed of some selected crystals. The manual approach is a very time-consuming process for multiple ice crystals, hence the development of an in-house software. As shown on Figure 5.7, the software outputs measures which are similar to the ones being manually taken.

Similar tests to the one displayed Figure 5.7 were done for other 94 different crystals and the results are shown on Table 5.5. The comparisons of measures taken manually and by the software gave trustworthiness to the data outputted by the program. Both measures were almost identical, which is an indicator that the program is an accurate tool for analysis of ice crystal impacts.

6 Conclusions

This thesis had two main goals: (1) To build a system capable of delivering constant rates of ice crystals with a specific size range to best mimic the natural conditions encountered in-flight in the atmosphere and (2) to build a software tool to analyze ice crystals and their impacts on cold and warm surfaces.

It was an exciting project where both ends of the test setup were developed in order to provide a solid test environment to measure ice crystal impact on surfaces. This work helped pave the way for more efficient test setups in the iCORE wind tunnel. The experimental phase consisted of testing and characterizing the produced ice with two different approaches and testing the software for performance. Two machines were built in order to provide a constant flow of ice crystals in the wind tunnel. The first prototype was based failed due to the lack of cooling power available, while the second prototype, a simpler and less prone to fail machine, provided a constant ice flow within the requirements. Some delays were encountered, which pushed the testing campaign to the limit of conclusion of this thesis, however clear results were drawn from the test campaign and helped validating the capabilities of the software. The software will be further improved, and new features may be added for better data acquisition. Nonetheless, the software proved to be a helpful tool to reduce the time taken to evaluate these impacts, while giving accurate data.

Bibliography

[1] Y. H. Yeong, J. Sokhey, and E. Loth, “Ice Adhesion on Superhydrophobic Coatings in an Icing Wind Tunnel,” in *Biological Research*, vol. 5, no. 1, 2017, pp. 28–37.

[2] C. Emersic, “Manchester Ice Cloud Chamber (MICC) Facility overview.” [Online]. Available: http://data.cas.manchester.ac.uk/micc/micc.htm. [Accessed: 11-Oct-2019].

[3] K. Al-Khalil, E. Irani, and D. Miller, “Mixed Phase Icing Simulation and Testing at the Cox Icing Wind Tunnel,” in *41st Aerospace Sciences Meeting and Exhibit*, 2003, no. January.

[4] P. G. Debenedetti and E. H. Stanley, “Supercooled and glassy water,” *J. Phys. Condens. Matter*, vol. 15, no. June, p. 42, 2003.

[5] C. Tomasi and R. Manduchi, “Bilateral Filtering for Gray and Color Images,” *Int. Conf. Comput. Vision, Bombay, India*, 1998.

[6] G. X. Ritter and J. N. Wilson, *Handbook of computer vision algorithms in image algebra*, Second. CRC Press, 2001.

[7] D. Ratul and A. A. Baquee, “Pellets and Pelletization Techniques: a Critical Review,” *Int. Res. J. Pharm.*, vol. 4, no. 4, pp. 90–95, 2013.

[8] “Motion Analysis and Object Tracking — OpenCV 2.” [Online]. Available: file:///C:/Users/THO1M0/Documents/References/Motion Analysis and Object Tracking — OpenCV 2.4.13.7 documentation.html. [Accessed: 06-Nov-2019].

[9] Matplotlib Org., “Matplotlib: Python plotting,” *Matplotlib*, 2019. [Online]. Available: https://matplotlib.org/. [Accessed: 17-Nov-2019].

[10] S. Suzuki and K. Abe, “Topological structural analysis of digitized binary images by border following,” *Comput. Vision, Graph. Image Process.*, vol. 30, no. 1, pp. 32–46, 1985.

[11] B. D. Lucas and T. Kanade, “Iterative Image Registration Technique With an Application To Stereo Vision.,” *Proc. Imaging Underst. Work.*, vol. 2, pp. 121–130, 1981.

[12] “The Python tutorial - Classes.” [Online]. Available: https://docs.python.org/3/tutorial/classes.html. [Accessed: 14-Nov-2019].