Characterisation of the Dust Cloud During Landing on Lunar and Martian Surfaces

Claudia Jimenez Cuesta[⋆†], *David Pickles*[⋆], *Kevin Worrall*[⋆], *Andrea Cammarano*[⋆] *and Hossein Zare-Behtash*[⋆]
[⋆]*James Watt School of Engineering, University of Glasgow*
University Avenue, Glasgow, G12 8QQ
[†]Corresponding author: Claudia.JimenezCuesta@glasgow.ac.uk

Abstract

Plume-Surface Interactions are complex phenomena present in extra-terrestrial landings which can damage spacecraft equipment. This work develops an edge detection algorithm to characterise the region of influence of a dust cloud formed upon plume impingement. High-speed imaging is used to test different image segmentation approaches and the results show that global thresholding provides a simple and efficient boundary detection method for regions of high particle concentration, while Gabor filtering in the frequency domain is an overall better yet more complex edge-detection approach. Lastly, a 3-D reconstruction is carried out by combining the detected edges from three perspectives.

1. Introduction

As spacecraft approach the surface of extra-terrestrial bodies, a full-deceleration before touch-down is required to ensure safe soft landings. This is extremely challenging in low density atmospheres - or the complete lack thereof - as the effectiveness of parachutes is very limited due to insufficient drag being achieved. Instead, retro-propulsive rockets are fired, decelerating the lander by providing thrust in the opposite direction of travel. However, this method comes at a cost: as the plumes from the exhaust nozzle impinge on the surface, the upper layers of regolith are eroded, causing particles and debris to be ejected at high speeds. Sandblasting can damage hardware,[1] flow redirection can induce de-stabilising loads on the lander[2] and lifted particles can reduce the visibility of pilots and the sensitivity of sensors.
During the Apollo and Viking eras, soil erosion experiments were carried out in the 1960s-1980s, applied to Lunar[3,4] and Martian[5,6] landing conditions. Studies were resumed in the 2000s for the Mars Exploration Rovers and Mars Science Laboratory Missions,[7,8] in which erosion mechanisms and rates were identified from visual data and CFD simulations. In recent years, Plume-Surface Interactions studies have been mainly computational, either assuming free-molecular flow via Direct Simulation Monte Carlo methods[9] or by using hybrid continuum-rarefied solvers[10] to predict particle ejecta. Confirming the validity of these results, however, has been difficult due to the lack of experimental data in relevant environments or at relevant physical scales.[11]
A novel approach in Plume-Surface Interactions study is presented here, in which the focus is shifted from erosion characterisation - and subsequent ejected particle predictions - to dust cloud characterisation. With the aim of establishing scalable relationships between nozzle exit flow conditions and the resulting cloud, three main variables of the cloud are aimed to be identified: its region of influence, particle concentration and particle speeds. The diagnostic techniques that will be explored include edge detection, light scattering models and Particle Image Velocimetry.
In this paper, the first step in cloud characterisation is presented, which consists on detecting the boundary of the cloud. A high pressure air nozzle is fired into a bed of Martian regolith simulant (crushed walnut shells[12]) and the edge detection capability of different image segmentation approaches is investigated. Having matured this technology and having explored methods of characterising the other two variables (particle concentration and speeds), future tests will be carried out in the vacuum chamber at the University of Glasgow, in which Lunar and Martian environments will be simulated.

2. Experimental Setup

Due to the slow turnaround of experiments carried out in the vacuum chamber (~ 4 hrs) and its high energy consumption, it was considered most appropriate to design a simplified, atmospheric, high-pressure air experiment to test the different edge detection approaches. The experimental setup consists of a jet simulator and a camera system.

2.1 Jet simulator

The components of the jet simulator are shown in Figure 1a, consisting of a SMC Pneumatic Solenoid valve (VDW series, G1/8 connection) that is actuated by a relay connected to a switch. The valve is used to provide a pulse of compressed air and the actuation time is measured by an oscilloscope (PicoScope 2405A).

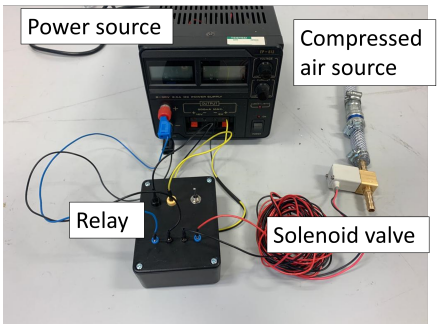
2.2 Camera system

To observe the dust cloud formation from different angles, up to 3 cameras are used:

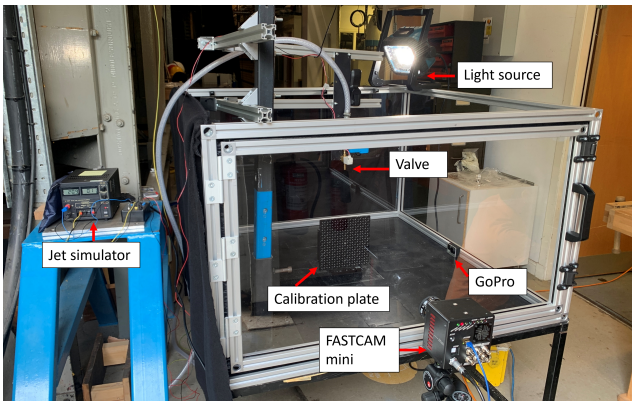
- A Photron FASTCAM Mini AX, with an image resolution of 1024x1024 pixels at a frequency of 2500 fps.
- Two GoPros, a Hero 4 and Hero 9, with a resolution of 1080x1920 pixels at 60 fps.

2.3 Experimental procedure

The first round of firings involved testing four different pressures (2 to 5 bar absolute pressure) and capturing the resulting cloud by a single camera (the FASTCAM Mini). The visual data recorded was used to develop and test the edge detection algorithms discussed in Section 3. After processing the data, a second round of experiments was carried out introducing two additional cameras (GoPros) and a light source. Calibration images were taken prior to firing to allow distortion correction, such that information from each perspective could be combined into a 3-D reconstruction, discussed in Section 4.



(a) Jet simulator components



(b) Test rig and camera setup

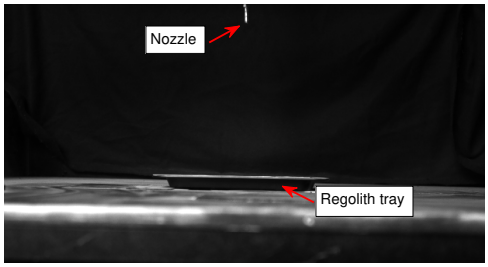
Figure 1: Experimental setup

3. Image Segmentation Approaches

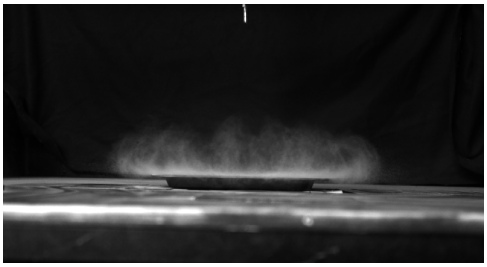
Image segmentation aims at dividing visual data into regions (or objects) of particular interest, such that these can be analysed in detail. In this research, the boundary of the dust cloud formed upon plume impingement is the segment of interest, as this corresponds to its region of influence. Three image segmentation approaches are presented in order of increased complexity: global thresholding, entropy segmentation, and Gabor filtering.

3.1 Global thresholding segmentation

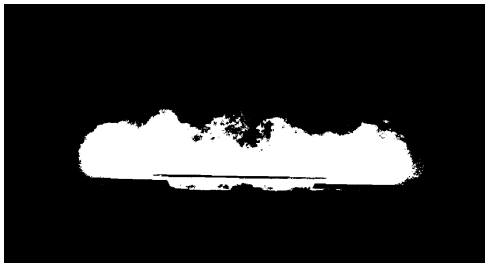
Image thresholding consists of converting a colour or grayscale image into a binary image, identifying objects based on colour or brightness and separating them from the background. If it is assumed that one unique class of objects exists (which is the case for this application, in which the dust cloud is the only object to be characterised), global thresholding can be carried out. This relies on the fact that an image has a bimodal histogram with two clusters of dominant pixel values and that a unique threshold value can be found to separate the two modes.[13]
Figure 2 shows the steps implemented to carry out global thresholding. Considering an initial frame prior to the firing of the nozzle F_b as the background (Figure 2a), each subsequent frame in a recording F_i (Figure 2b) can be binarized by first subtracting the background $I_i = |F_i - F_b|$ and then applying a global thresholding segmentation. Following Otsu's method,[14] MATLAB's built-in function `graythresh` calculates the global thresholding value that can be used in conjunction with the function `imbinarize` to process the frames such that the main object (the dust cloud) is highlighted (Figure 2c). To trace the edge of the cloud, a minimum pixel size is established via `bwareopen` and the boundary is traced via `bwboundaries` (Figure 2d). It should be noted that background subtraction is an essential part of the algorithm, as it allows the development of the cloud to be captured in isolation from other elements in frame, such as the surface which has a higher illumination than the black background and would instead be identified as an object.



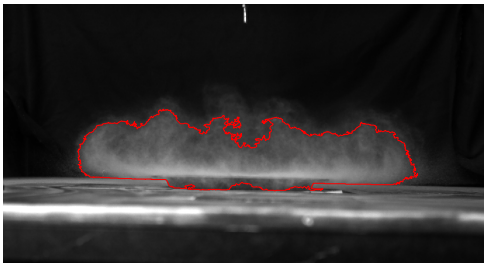
(a) Background frame F_b



(b) Investigated frame F_i



(c) Binarized frame



(d) Edge detected

Figure 2: Global thresholding algorithm

Threshold variation

It was observed that by using the global thresholding value automatically calculated, the boundary found resulted in an under-contoured cloud. By manually decreasing the threshold value, the contour detected increased significantly, as shown by the 50% threshold reduction plotted in blue in Figure 3. The choice for thresholding could be adjusted depending on the particle concentration to be captured, as lower concentration areas require a lower threshold to be included in the boundary. Methodologies to achieve this are discussed in Section 5.

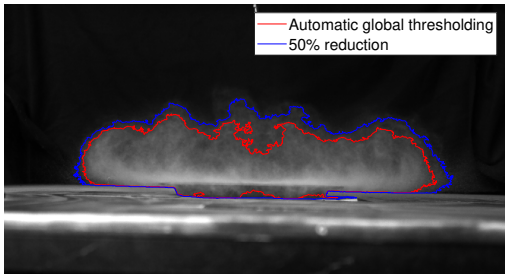


Figure 3: Comparison of thresholding values

3.2 Texture segmentation

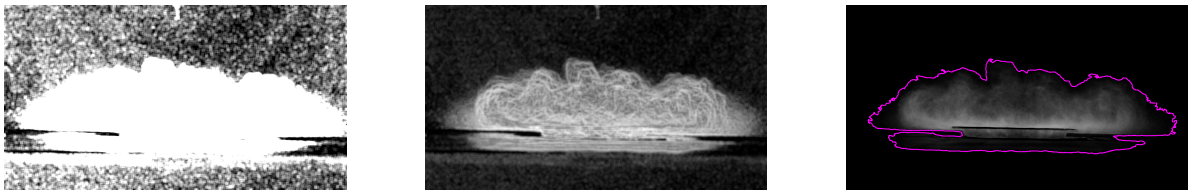
In the context of 2-D images, texture can be understood as the variation in spatial frequency across neighbouring pixels. Defining spatial frequency as the distribution of light or gray values in an image,[15] one can think of texture as a combination of irradiance patterns.[16] Along sharp edges and fine details, these regions will present high spatial frequencies while smoother regions will consist of lower frequencies.[17] In this section, this concept of texture is applied to carry out edge detection, through entropy analyses and Gabor filtering.

Entropy

Entropy analysis is a statistical texture segmentation approach that measures the variability in an image.[18] This means that areas in which neighbouring pixels have a high variation in pixel intensity value will have high entropy, while areas of constant texture will have an entropy value of 0. Entropy is calculated using histogram data for an image, where $h(r_k)$ contains the histogram counts, i.e. the number of pixels present in an image for each intensity value r_k. In a grayscale image, r ranges from 0 (black) to 255 (white). Dividing $h(r_k)$ by the total number of pixels in an image N_t, the probability of occurrence of a particular intensity value is obtained:[19] $p(r_k) = h(r_k)/N_t$. Lastly, using the probability values, entropy is defined as:

$$E = -\sum_{k=0}^{255} p(r_k)\log_2(p(r_k)) \quad \forall \quad p(r_k) \neq 0 \tag{1}$$

Local entropy is calculated using subsets (or neighbourhoods) of an image and reflects the texture in small regions. This concept is applied to dust cloud identification, as the regions with scattered lifted particles will have high values of local entropy, mathematically captured by the fact that a wider range of intensity values with lower probability will result in a higher value of entropy. Hence, areas of constant intensity such as the background are expected to have entropy values close to 0 while areas of fast changing pixel values like the cloud will be highlighted. This is carried out via MATLAB's `entropyfilt`, which calculates local entropy considering 9 by 9 pixel neighbourhoods. The output (shown in Figure 4a) is then re-scaled to be in the range [0,1] as shown in Figure 4b. Then, global thresholding and edge detection are carried out, displayed in Figure 4c. This method overestimated the boundary of the cloud and it was observed that varying the threshold did not have a significant effect. Moreover, decreasing the neighbourhood size to the minimum of 3 by 3 did not improve the region detected either. Hence this segmentation approach is considered inappropriate for this application and further work is required to determine the cause of this behaviour.



(a) Local entropy (b) Local entropy re-scaled (c) Edge detected

Figure 4: Entropy algorithm

Gabor filters

Gabor filtering is an image segmentation technique that aims to mimic how the human visual system recognises textures through spatial and orientational sensitivities.[20] It is thought that mammals process visual information by dividing retinal images into "channels", which are filtered images that contain information within a range of spatial frequencies and orientations.[15] In other words, the retina applies band-pass filters, which are used to process the different components in an image. In a similar manner, Gabor banks are a collection of filters (or kernels) that act as the "channels", identifying specific textures in localised image regions, allowing features to be extracted.

A Gabor filter h in the 2-D spatial domain (x, y) is a wavelet-like transform in which the multipliers within the kernel h_i are defined as:

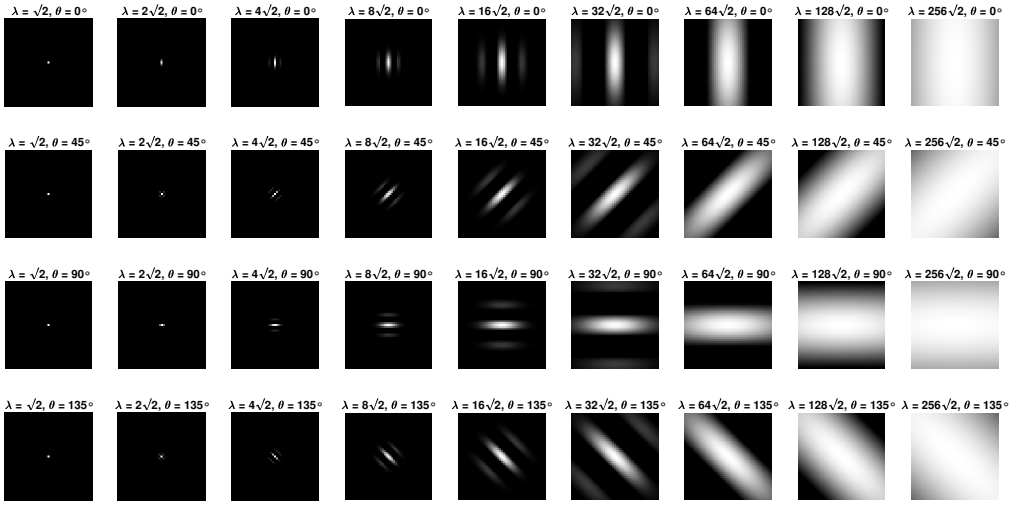
$$h_i(x, y, f, \theta) = \exp\left\{-\frac{1}{2}\left[\frac{x'^2}{\sigma_x^2} + \frac{y'^2}{\sigma_y^2}\right]\right\} \Upsilon(x, y, f, \theta) \tag{2}$$

Where f_i is the desired spatial frequency to be filtered for, inversely proportional to the wavelength ($\lambda_i \propto 1/f_i$) and θ is the desired orientation, clock-wise and normal to the $x-$axis. The positions within the kernel x', y' are defined as

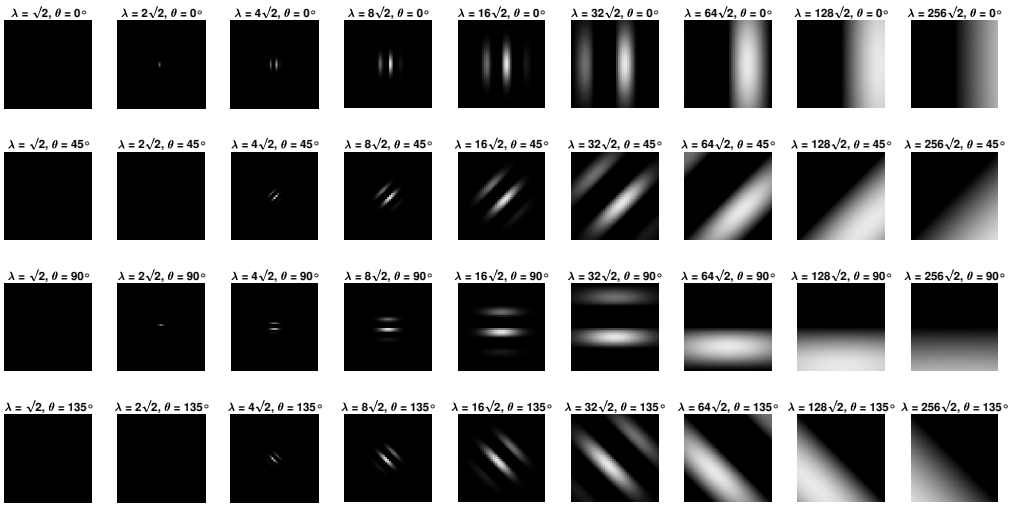
$x' = x\cos\theta + y\sin\theta$ and $y' = -x\sin\theta + y\cos\theta$.[21] The standard deviations of the Gaussian envelope σ_x and σ_y are calculated as:

$$\sigma_x = \frac{\lambda}{\pi}\sqrt{\frac{\log_2}{2}}\frac{(2^{BW}+1)}{(2^{BW}-1)} \text{ and } \sigma_y = \frac{\sigma_x}{\gamma} \tag{3}$$

Where BW is the bandwidth (set to an octave, following the idea that simple cells in the visual cortex have a bandwidth of an octave[22]) and γ is the aspect ratio, which defines the ellipticity of the Gabor function.[21] The sinusoidal carrier Υ present in Equation 2 can be defined as a complex exponential[20] or a cosine function, if even-symmetric Gabor filters are to be used.[15] The choice of kernel size, wavelengths and orientations will have a great effect on the texture identification capabilities of the Gabor filters. An initial set of parameters are tested, following the guidelines by Jain and Pektov.[15,23] First, minimum and maximum wavelengths are defined as $\lambda_{min} = \sqrt{2}$ and $\lambda_{max} = (N_c^2 + N_h^2)^{1/2}$ for an image N_c pixels wide by N_h pixels high. To ensure that frequencies (and therefore wavelengths) are one octave apart, a variable n is defined as $n = \log_2(\lambda_{min}/\lambda_{max})$ and used to calculate the radial wavelengths $\lambda_i = 2^i \times \lambda_{min}$ where i runs from 0 to $(n-2)$. This is equivalent to defining $\lambda_i = 1\sqrt{2}, 2\sqrt{2}, 4\sqrt{2}, ..., (N_c/4)\sqrt{2}$. These wavelengths are tested in combination with four orientations: 0 deg, 45 deg, 90deg and 135 deg. For an image with a resolution of 1024x1024 pixels (as provided by the high-speed Photron camera), the Gabor kernels are obtained following Equation 2 and shown in Figure 5.



(a) Real part



(b) Imaginary part

Figure 5: Gabor bank illustrating the different kernels used

Having calculated the Gabor bank, spatial filtering can be performed by correlating each kernel h with an image I

through MATLAB's `imfilter` to obtain the filtered image[19] $A = h \star I$. After a binarization using global thresholding, the images in Figure 6 were obtained and used to discard the smallest and largest wavelengths tested. On one hand, it was observed that the smallest wavelength tested had an insignificant filtering effect, as the kernels contained a central multiplier of 1, while others had multipliers with values close to 0. On the other hand, increasing wavelength values over $32\sqrt{2}$ had no effect on the filtered image obtained, as the detailed variations in pixel values were not accurately captured. Hence these larger wavelength filtered images have been omitted from Figure 6. Lastly, it was noted that the filtered images between wavelengths of $4\sqrt{2}$ and $8\sqrt{2}$ varied significantly and hence a middle value was tested. Figure 7 shows the next step in the algorithm, in which edge detection was performed. A Gabor kernel size of 3×3 was used, as larger sizes worsened the edge detection capability while increasing the computational time required. A wavelength of $6\sqrt{2}$ and orientation of 45 deg were selected as they provided the most accurate boundary tracing, not just for the frame investigated, but for the overall recording of cloud development, as shown in Figure 9.

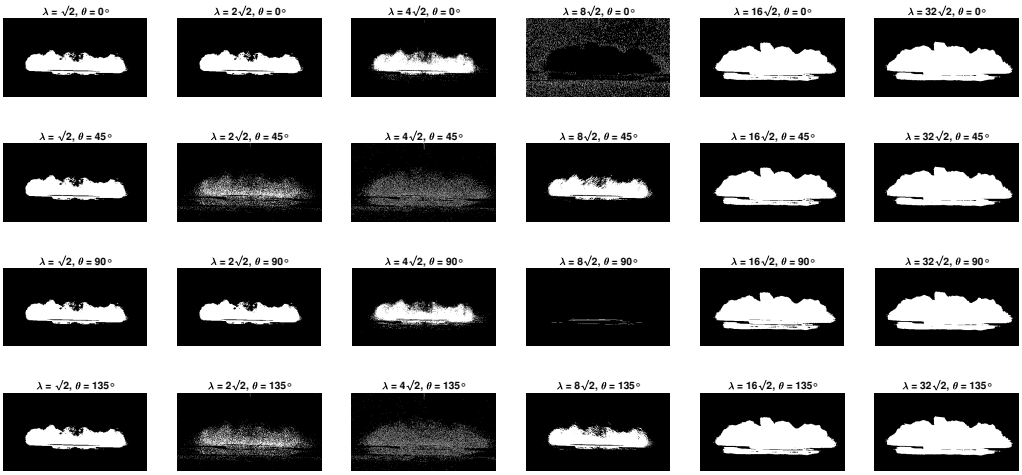


Figure 6: A filtered image using the Gabor bank

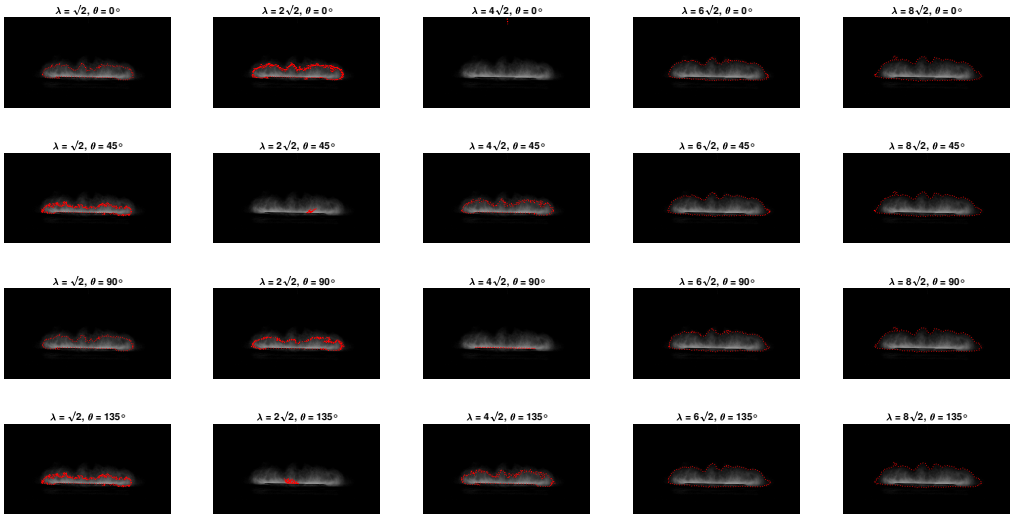


Figure 7: Edge detection using Gabor filters

Gabor filtering can also be carried out in the frequency domain, by using the Fourier domain representation of Equation 2:

$$\bar{H}(u, v, f, \theta) = 2\pi\sigma_x\sigma_y \left(\exp\left\{ -\frac{1}{2} \left[\frac{(u' - f)^2}{\sigma_u^2} + \frac{v'^2}{\sigma_v^2} \right] \right\} \right) \tag{4}$$

Where:

$$\sigma_u = \frac{1}{2\pi\sigma_x} \text{ and } \sigma_v = \frac{1}{2\pi\sigma_y} \tag{5}$$

In a similar manner to the spatial domain, $u' = u\cos\theta + v\sin\theta$ and $v' = -u\sin\theta + v\cos\theta$, however in this case, u, v span the frequency domain, representing the Discrete Fourier Transform samples, where $u = 0, 1, ...N_h - 1$ and $v = 0, 1, ..., N_c - 1$.
Exploiting the Fast Fourier Transform of the input image $I \rightarrow \bar{I}$, the filtered image $\bar{A}$ in the frequency domain is found through:

$$\bar{A} = \bar{H}\bar{I} \tag{6}$$

By carrying out filtering in the frequency domain for the same set of parameters tested in Figure 6, the following images are obtained (after an Inverse Fast Fourier Transform has been performed):

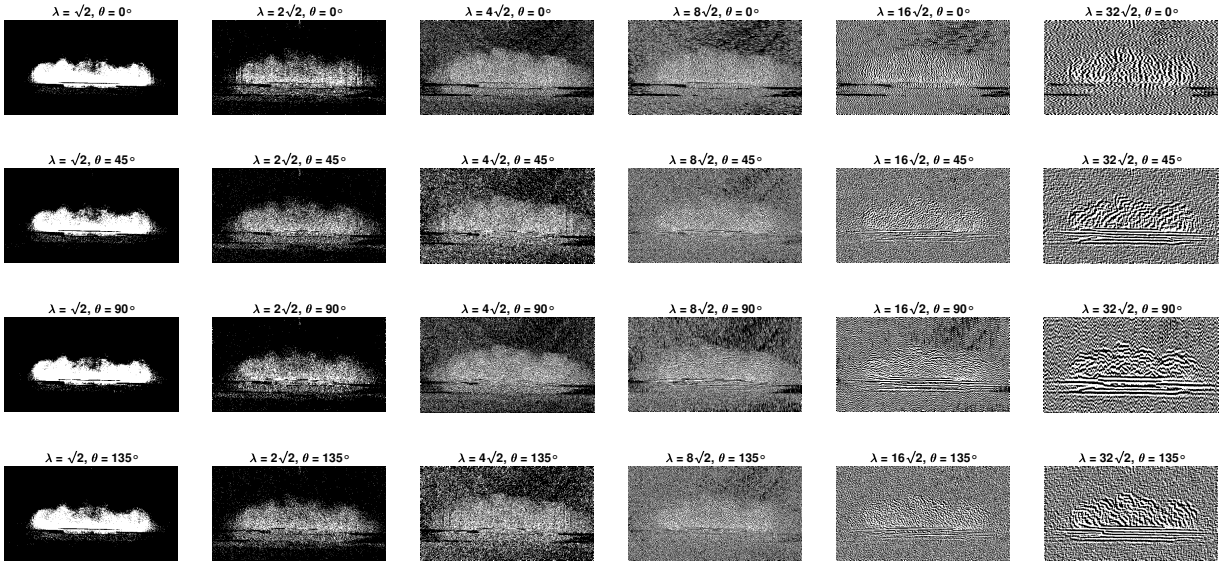


Figure 8: A filtered image using the Gabor bank in the frequency domain

It can be observed that increasing wavelength results in blurred filtered images and hence wavelengths smaller than $\sqrt{2}$ were investigated. Results did not improve beyond $\lambda < \sqrt{2}/5$ and hence this was chosen to carry out edge detection. The obtained boundary is presented in Figure 9 for a range of frames.

3.3 Comparison between approaches and discussion of accuracy

In order to establish which approach provides the most accurate edge detection, the dust cloud was identified by each method at different stages in its development, and the boundaries plotted for comparison. As expected, entropy segmentation was the most inaccurate method, resulting in a highly over-contoured boundary. The other three methods detected the edges with comparable accuracy, as presented in Figure 9. It should be noted that accuracy is measured by visual inspection, as a hard boundary of the dust cloud does not exist. In order to establish a more quantifiable measure of accuracy, particle concentration calculations must be introduced. It can be observed from Figure 9 that global thresholding and Gabor frequency filtering trace the edge in a region with higher particle concentration than that of Gabor spatial filtering. This leads to the question of which of the three methods is most appropiate for this application. In terms of complexity, Gabor filtering uses a range of parameters whose choice can have significant impact on the boundary traced. On the other hand, global thresholding is a simple approach that provides accurate results, however it may require the thresholding value to be adjusted if lower particle concentration areas of the dust cloud are to be included in the traced boundary.

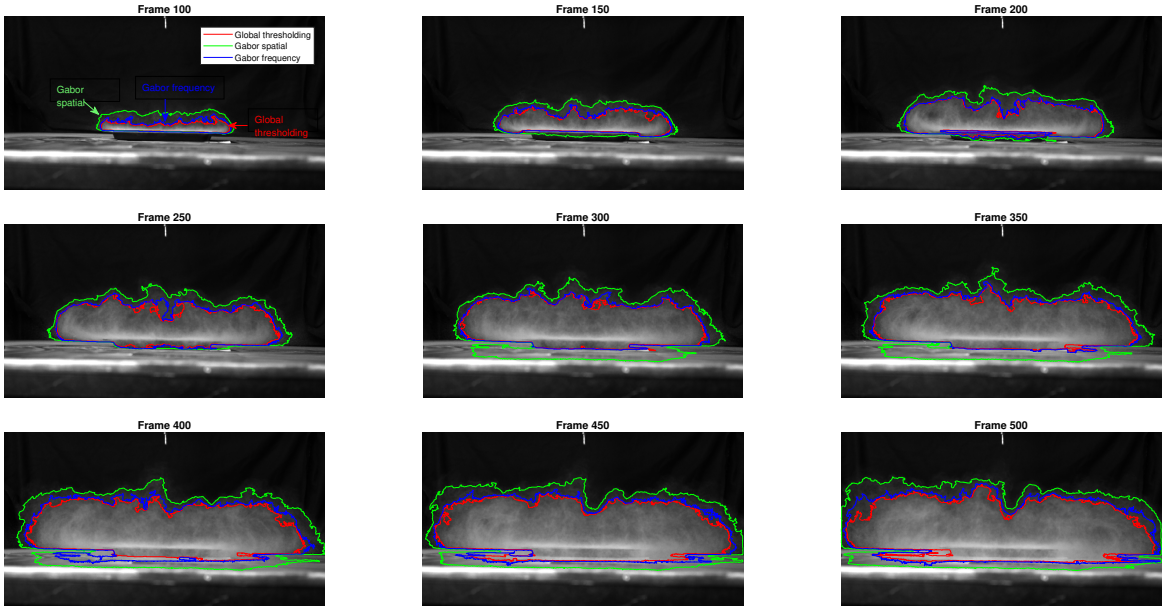


Figure 9: Edge detection comparison between approaches

4. Three-Dimensional Reconstruction

Having developed and tested algorithms to carry out boundary detection from one perspective, additional cameras were introduced for the second round of experiments to better observe cloud growth.

4.1 Checkerboard calibration and distortion correction

All camera lenses introduce distortion in the images they capture, which can be more or less noticeable to the human eye depending on the field of view of the camera. Although the distortion caused by the fish eye effect of the GoPros is most significant, distortion of the 28 mm lenses of the FASTCAM camera must also be corrected, in order for the boundary of the dust cloud to be accurately determined from the different captured images and combined into a unique reconstruction. Distortion correction can be achieved through checkerboard calibration, in which a pattern of known dimensions is used to find the intrinsic parameters of a camera. Checkerboard calibration images covering the field of view are taken (as seen in Figure 10) and input to MATLAB's `detectCheckerboardPoints` and `cameraIntrinsics` functions, which estimate the radial and tangential distortion parameters, allowing the images to be corrected via `undistortImage`.

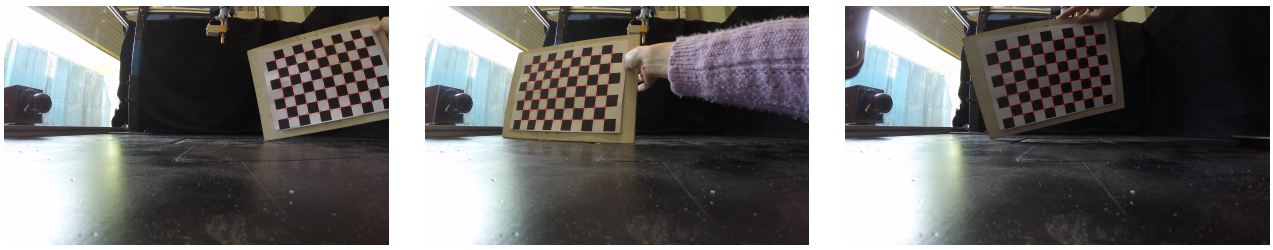


Figure 10: Checkerboard calibration images with detected corners (plotted in red)

4.2 Mapping to a global coordinate system

In order to combine the information obtained by the different cameras, the procedure presented in Figure 11 is followed. First, the distortion in the videos at each perspective is corrected and an image segmentation algorithm is used to

find the cloud edge in intrinsic coordinates (i.e. pixel indices c, r). Then, a mapping between the intrinsic spatial coordinate system (in pixels) and a local world coordinate system u, v (in milimeters) is found using `imref2d` and `intrinsicToWorld` functions. These require a relationship between pixel size and distance to be specified, which is obtained through an image of a calibration plate of known dimensions and position in the physical space. Lastly, the edges are translated into a global coordinate system x, y, z according to the camera's position β, in which $x = u\cos\beta$, $y = u\sin\beta$ and $z = v$. This requires that for each camera, the calibration plate is aligned with the global origin (under the valve nozzle and at the surface of the regolith tray).

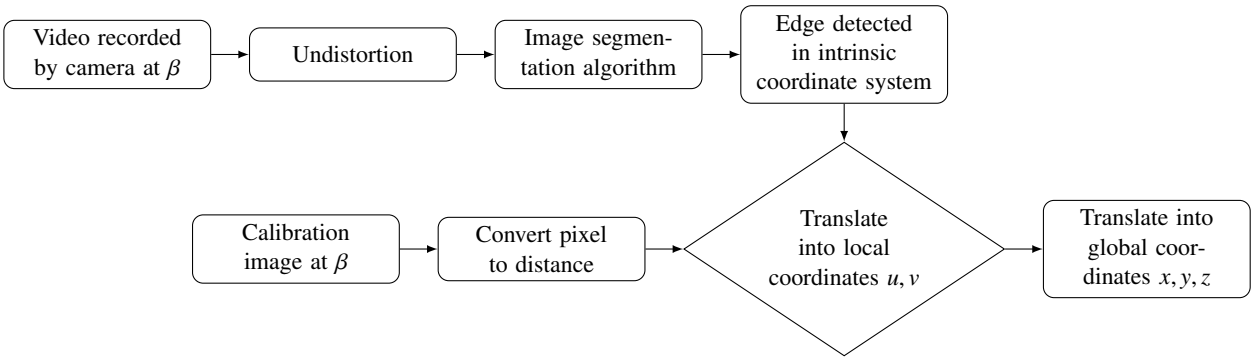


Figure 11: Mapping to global coordinate system flow chart

Figure 12 shows the edges detected for each distortion corrected view from the cameras. It can be observed that the GoPros capture a brighter cloud as opposed to the FASTCAM, which is due to their lower frame rate. The high-speed camera is able to capture instantaneous lifted particle positions, while the GoPro's longer exposure results in a blurred and brighter cloud. Each boundary is mapped into the global coordinate system, allowing the three-dimensional reconstruction to be carried out, shown in Figure 13b.

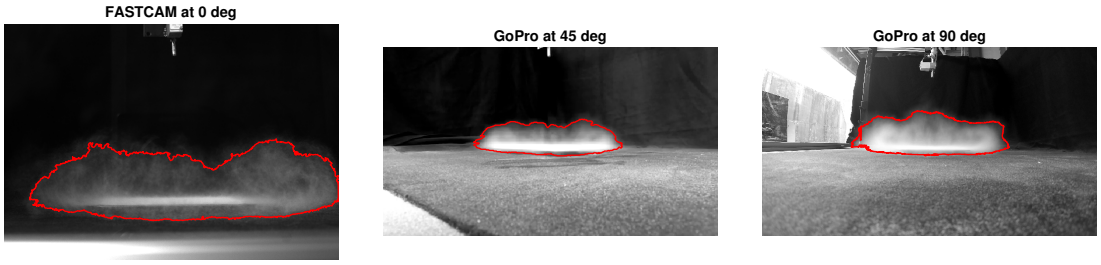
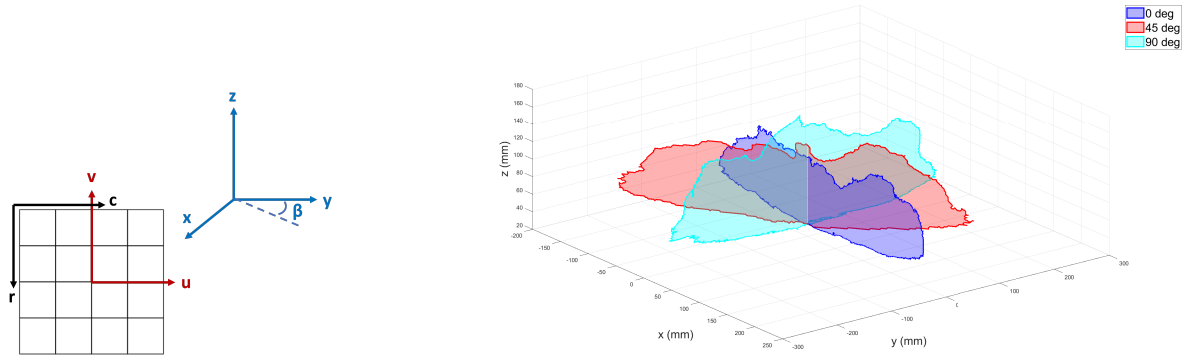


Figure 12: Camera perspectives

4.3 Sources of error

By carrying out the 3-D reconstruction with the methodology discussed, errors are incurred from different sources. Firstly, the distortion correction of the cameras is carried out by obtaining intrinsic parameters via checkerboard calibration. Although measures were taken to minimise the error, such as taking images covering the whole field of view, conditions such as small variations in lighting can decrease the accuracy of the correction. Secondly, by converting pixel to world distances, it is assumed that the camera and calibration plate are aligned perfectly perpendicular. Although they were positioned using a laser level, it is likely that a small deviation is present, resulting in a skewed view of the plate that will induce an error in the pixel-distance averaged across the plate. Moreover, the center of the calibration plate is also assumed to be aligned with the valve nozzle, as the origin in the three-dimensional global coordinate system was taken to be this point during local to global coordinate mapping. Thirdly, with regards to the edges detected, it should be noted that these correspond to the projected boundaries of the cloud from each view, as opposed to planar views of the cloud's cross-section as assumed in the 3-D reconstruction. This can lead to an inaccurate estimation of the region of influence, if either features of a plane are obstructed by the developing cloud ahead of the plane investigated or the region of the cloud behind the plane is captured by the camera and projected onto the detected edge. The former is present in the reconstruction displayed in Figure 13b: the 90 deg plane appears to have



(a) Intrinsic, local and global coordinate systems

(b) 3-D reconstruction from the individual planar views

Figure 13: Mapping and reconstruction of the dust cloud

a central growth when in reality, this corresponds to the elevated sides observed by the FASTCAM at 0 deg. Lastly, when combining the information from the three cameras, the difference in frame rates between the FASTCAM (2500 fps) and the GoPros (60 fps) meant that the exact frame could not be matched, resulting in the reconstruction to be composed of perspectives at slightly different instances. It was also observed during the post-processing of videos that the nozzle had a slight deviation in its vertical positioning, causing the cloud to develop un-evenly in the y, z plane, as it can be observed by the 0 deg camera perspective in Figure 12.

5. Conclusions and Future Work

This paper presents the first stage in the development of a dust cloud detection algorithm in the context of Plume-Surface Interactions during extra-terrestrial landings. In order to characterise the region of influence of the growing cloud, the edge detection capability of different image segmentation approaches was evaluated on footage from an atmospheric conditions experiment, in which high-pressure air impinges on a bed of Martian regolith simulant. Three cameras were used to capture the development of the cloud from different perspectives, allowing a three-dimensional reconstruction to be carried out. Having successfully developed a methodology for region of influence characterisation, future work will aim at characterising the other two main variables of a dust cloud: particle concentration and velocities. Signal scattering models, such as light and radar dissipation across dust clouds will be explored to evaluate particle concentration, while techniques such as Particle Image Velocimetry (PIV) will be used for velocity estimations. In addition, scalable parameters including nozzle exit properties (pressure and diameter) and impingement height will be introduced, and relationships between dust cloud growth in atmospheric and extra-terrestrial conditions will be explored. This will require a re-design of the jet simulator, such that the nozzle geometry can be defined and used in the scaling of parameters, as well as ensuring a vertical alignment of the jet. Lastly, measures to reduce the errors discussed in Section 4.3 will be explored, such as introducing laser sheets to highlight planes for accurate planar edge detection, as well as extending the experiments to pulsed-jet impingement with synchronised high-speed cameras. Moreover, a new approach of global mapping will be explored using the cameras' extrinsic parameters, estimating their angle and rotation with respect to the origin and converting intrinsic coordinates to global coordinates without the use of the dotted calibration plate.

Acknowledgements

This research has been undertaken under the Mary Gibbs Dunlop scholarship, funded by the Engineering and Physical Sciences Research Council (EPSRC) of UK Research and Innovation. The authors would like to thank Joe Gibbs, Thomas Williams and Derek Gray for their input in this work, and the administrative and technical staff at the University of Glasgow, especially Mr Bernard Hoey, for their help and support.

References

[1] Philip T. Metzger, John E. Lane, Christopher D. Immer, and Sandra Clements. Cratering and blowing soil by rocket engines during lunar landings. *International Conference on Case Histories in Geotechnical Engineering*, 2008.

[2] A. Rahimi, O. Ejtehadi, K.H. Lee, and R.S. Myong. Near-field plume-surface interaction and regolith erosion and dispersal during the lunar landing. *Acta Astronautica*, 175:308–326, 2020.

[3] Norman S. Land and Harland F. Scholl. Scaled lunar module jet erosion experiments. *NASA Report*, 1966.

[4] Leonard V. Clark. Experimental investigation of close-range rocket-exhaust impingement on surfaces in a vacuum. *NASA Report*, 1970.

[5] Leonard V. Clark. Effect of retrorocket cant angle on ground erosion - a scaled viking study. *NASA Report*, 1970.

[6] G.L. Romine, T. D. Reisert, and J. Gliozzi. Site alteration effects from rocket exhaust impingment during a simulated viking mars landing. *NASA Report*, 1973.

[7] Anita Sengupta, James Kulleck, Steve Sell, John Van Norman, Manish Mehta, and Mark Pokora. Mars lander engine plume impingement environment of the mars science laboratory. In *2009 IEEE Aerospace conference*, pages 1–10, 2009.

[8] Manish Mehta, Anita Sengupta, Mark Pokora, Leslie Hall, and Nilton Renno. *Mars Landing Engine Plume Impingement Ground Interaction*, pages 143–157.

[9] Xiaoying He, Bijiao He, and Guobiao Cai. Simulation of rocket plume and lunar dust using dsmc method. *Acta Astronautica*, 70:100–111, 2012.

[10] Aaron B. Morris, David B. Goldstein, Philip L. Varghese, and Laurence M. Trafton. Approach for modeling rocket plume impingement and dust dispersal on the moon. *Journal of Spacecraft and Rockets*, 52(2):362 374, 2015.

[11] Ashley M. Korzun, Chad J. Eberhart, Jeffrey West, Peter Liever, Andrew Weaver, James Mantovani, Austin Langton, Beverly Kemmerer, and Austin Atkins. *Design of a Subscale, Inert Gas Test for Plume-Surface Interactions in a Reduced Pressure Environment*. 2022.

[12] C. White, H. Zare-Behtash, K. Kontis, T. Ukai, J. Merrifield, D. Evans, Coxhill I., T. Langener, and J. Van den Eynde. Test facility to investigate plume-regolith interactions., 2019.

[13] Jadwiga Rogowska. Chapter 5 - overview and fundamentals of medical image segmentation. pages 73–90, 2009.

[14] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979.

[15] Anil K. Jain and Farshid Farrokhnia. Unsupervised texture segmentation using gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[16] A.C. Bovik, M. Clark, and W.S. Geisler. Multichannel texture analysis using localized spatial filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(1):55–73, 1990.

[17] American Psychological Association. Perception of high and low spatial frequency information in pigeons and people. 2015.

[18] Rafael C. Gonzalez and Richard E. Woods. *Digital image processing*. Pearson, fourth, global edition, 2002.

[19] Rafael C. Gonzalez, Richard Eugene Woods, and Steven L. Eddins. *Digital image processing using MATLAB*. Pearson Education, 2004.

[20] David A. Clausi and M. Ed Jernigan. Designing gabor filters for optimal texture separability. *Pattern Recognition*, 33(11):1835–1849, 2000.

[21] Mohammad Haghighat, Saman Zonouz, and Mohamed Abdel-Mottaleb. Cloudid: Trustworthy cloud-based and cross-enterprise biometric identification. *Expert Systems with Applications*, 42(21):7905–7916, 2015.

[22] Daniel A. Pollen and Steven F. Ronner. Visual cortical neurons as localized spatial frequency filters. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-13(5):907–916, 1983.

[23] P. Kruizinga and N. Petkov. Nonlinear operator for oriented texture. *IEEE Transactions on Image Processing*, 8(10):1395–1407, 1999.