

UNIVERSITY OF NEVADA, RENO

Automated detection of dune crest-lines in planetary satellite images

A thesis submitted in partial fulfillment of the requirements for the degree of Master of Science in Computer
Science and Engineering

by
David Leblanc

Dr. George Bebis/Thesis Advisor
Dr. Mircea Nicolescu/Thesis Co-Advisor

May 2017

© by David Leblanc 2017
All Rights Reserved



THE GRADUATE SCHOOL

We recommend that the thesis
prepared under our supervision by

DAVID LEBLANC

Entitled

Automated detection of dune crest-lines in planetary satellite images

be accepted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE

George Bebis, Ph.D., Committee Chair/Advisor

Mircea Nicolescu, Ph.D., Committee Co-Chair

Nicholas Lancaster, Ph.D., Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

May, 2017

Abstract

Dune-field patterns are believed to behave as self-organizing systems, but what causes the patterns to form is still poorly understood. The most obvious (and in many cases the most significant) aspect of a dune system is the pattern of dune crest lines. Extracting meaningful features such as crest length, orientation, spacing, bifurcations, and merging of crests from image data can reveal important information about the specific dune-field morphological properties, development, and response to changes in boundary conditions. However manual methods are labor-intensive and time-consuming.

In this research, we are developing the capability automatically detect crest-lines and computing the geomorphological properties of dune fields, using satellite images on planetary surfaces. Our goal is to develop a robust methodology and the necessary algorithms for automated or semi-automated extraction of dune morphometric information from image data.

We investigated multiple different approaches to solve the problem, ranging from appearance-based, gradient-based, and machine learning methods. We strive to solve the problem caused by the inherent bimodal distribution of the histogram of the gradients by computing the overall dominant orientation of the dune field gradients. The concept of the gradient direction map is also introduced as a method to segment areas of the image which agree with the dominant orientation of the dune fields. Crest-line candidates can be extracted from the segmented areas which are in turn filtered using the responses of a train machine learning model. Once candidate crest-lines have been detected, the morphological properties of the dune field can be computed.

Our approach has been tested on two distinct dataset including satellite images of six terrestrial regions and from Mars. The results reported perform well on both linear and complex dune structures, and have relatively high precision and recall rates and low error in computation of the morphological properties, indicating that our approach generally robust.

Contents

1	Introduction	1
1.1	Challenges	1
1.2	Contributions	3
2	Background and Previous Work	4
2.1	Dune Pattern Studies	4
2.2	Image Processing and Computer Vision	5
2.2.1	Preprocessing	5
2.2.2	Feature Detection	6
2.3	Dune Detection	8
2.3.1	Appearance-Based Methods	8
2.3.2	Machine Learning Methods	8
3	Methodology	10
3.1	Challenges	10
3.2	Overall Process Flow	11
3.3	Appearance-Based Approach	14
3.4	Edge-based Approach	16
3.4.1	Dominant Orientation From K-Means Clustering	18
3.4.2	Dominant Orientation From Histogram of Gradients	19
3.4.3	Dune Segment Linking	21
3.5	Watershed Transform Segmentation Approach	22
3.6	Frequency Domain Approach	24
3.7	Gradient Orientation-Based Approach	26
3.8	Machine Learning Approach	29
3.8.1	Dataset Creation	30
3.8.2	Feature Descriptor Extraction	30
3.8.3	Supervised Learning of the Crest-line Classifier Model	31
3.8.4	Response Maps	31
3.8.5	Crest-line Extraction: Thinning And Skeletonization	35
3.9	Machine Learning and Gradient-Based Mixed Approach	37
3.9.1	Gradient Orientation Crest-line Extraction	39
3.9.2	Localization Improvement of Crest-line Segments	39
3.10	Dune Field Morphology Metrics	40
3.10.1	Dune Field Orientation Computation	42
3.10.2	Inter-Dune Distance Computation	43
4	Experimental Evaluation	44
4.1	Study Areas	44
4.1.1	Terrestrial Dataset	44
4.1.2	Mars Dataset	47
4.2	Evaluation	47
4.3	Results	49
5	Conclusion	68
5.1	Discussion	68
5.2	Future Work	70

List of Figures

1	Examples of dune pattern analysis including metrics such as orientation, crest length, spacing, and termination points.	2
2	Sample satellite images of two distinct dune field regions: Kalahari and Winnemucca Dune Complex	2
3	Preliminary results of the Line Segment Detector (LSD [118]) applied to the linear dunes of the Kalahari (left) and transverse dunes in White Sands (right). The results of the LSD on linear dunes are good, while complex transverse dune results are poor.	6
4	The results of applying tensor voting schemes to find salient linear segments in an image ([65])	7
5	Cross-section of a dune slope with the sun's orientation effect on shaded and sunlit portions of the dune.	10
6	The variation in dune patterns such as (a) linear, or (b) complex, and the sun orientation and illumination effects on the resulting satellite images.	11
8	The overall process flow for general dune detection. The five steps include image processing, dominant orientation computation, crest-line candidate detection, calculations of dune metrics, validation of crest-line candidates.	11
7	Illustration of the intensity, gradient magnitudes, and gradient orientations, plotted along a path crossing a dune crest-line: (a), (d),(g) a sample image patch from Kalahari, Skeleton Coast, and White Sands regions, (b), (e), (h) the direction of the pixel samples plotted in (c), (f), (i) containing the intensity, gradient magnitude and orientations values along the sampled direction.	12
9	The process flow of the appearance-based approach. The images are first preprocessed, then an adaptive threshold is applied to binarize the image. Contours are extracted from the resulting binary regions. The contour segments are smoothed and filtered by gradient magnitude, resulting in the detected crest-lines.	14
10	Adaptive Threshold applied to the Kalahari dune field satellite image. In (a) the Kalahari input image, (b) the adaptive threshold process applied to produce the binary image.	15
11	The process flow of the gradient-based approach. The images are first preprocessed, then the Canny edge detection algorithm is applied to the image. The dominant orientation computation (3.4), which enables many of the detected Canny edges to be filtered out based on orientation. The remaining edges are linked and post filtered based on length and shape.	16
12	Canny edge detector after median and Gaussian filtering of the Kalahari dune image.	17
13	Results of K-Means clustering algorithm with $K = 2$ on the edge gradients detected using Canny, from Kalahari. The blue cluster represents the dune crest-line gradient vectors, and the red points represents all other non-crest-line edges. The cluster centers of blue is $(0.8958, -0.5496)$, and the cluster center of the red is $(-0.4111, 0.2705)$. The cluster centers are used to determine the dominant orientation.	18
14	Results of filtering out edges which are not part of the dominant orientation (in red), while preserving edges which agree (in green) with the dominant orientation computed by the K-Means clustering method.	19
15	Results of unnormalized (red) and normalized (green) histograms of gradients orientations ($N = 18$), where each represents a 20° slice. The histogram contains two major peaks, but the crest-line is the weaker of the two peaks. (a) Edges which belong to the dominant orientation which are invalid crest-lines. (b) With normalization, the true crest-lines now have the higher overall magnitude. (c) The 18 bin histogram of gradients with and without normalization.	20
16	Results of filtering based on the line orientation fitting, and choosing the peak in the histogram. In (a) the fitted lines from the dune segments, (b) segments which have orientation matching the peak in the histogram, (c) the histogram of line orientations weighed by segment length.	21
18	Simple illumination normalization algorithm applied to the Skeleton Coast image.	22
17	The process flow of the Watershed Transform dune detection approach. The images are first preprocessed, then some illumination normalization is applied to the image in order to improve the results of the Watershed transform. Contours are then extracted from the resulting regions which are filtered using the gradients, based on the dominant orientation.	22
19	Watershed Transform results example applied to the Skeleton Coast. In (a), the pixels labeled as <i>shaded</i> , (b) the pixels labeled as <i>sunlit</i> , (c) the output of the Watershed Transform, and (d) crest-line detection results from the Watershed Transform.	24
20	The process flow of the frequency domain approach. The images are first preprocessed, then translated into the frequency domain using the standard 2D discrete Fourier Transform. A high-pass filter is applied to the frequency domain to extract highest frequencies (which are assumed to contain dune crest-lines), and the inverse DFT is applied to return to the spacial domain. Crest-line detection can then be applied to the resulting image.	24

21	Frequency spectrum processing results from the Skeleton-Coast image. (a) Log-normalized shifted frequency domain, (b) Result of thresholding frequency domain magnitude based on average magnitude and standard deviation of the magnitudes, (c) Median filtering of the binary image of the frequency domain, (d) Result of ellipse fitting of the dominant frequencies region, (e) Mask used for high-pass filtering of the frequency domain, (f) Result inverse DFT of the high pass filter applied on the frequency domain.	25
22	The process flow of the gradient orientation-based approach. The image is first preprocessed for optimal gradient computation, and the dominant orientation is computed. The gradient direction map is then computed and thresholded to preserve all gradients which <i>agree</i> with the dominant orientation. The resulting binary image is a good candidate for contour extraction, which are subsequently filtered based on gradient orientation and magnitude. The resulting output is the detected crest-lines segments.	26
23	The gradient orientation-based approach illustrated: (a) The input image WDC, (b) The gradient direction map D computed by taking the dot product of the gradients at each pixel with the dominant orientation, (c) thresholding the gradient direction map for all dot products greater than 0, (d) the results of crest-line detection using the gradient-orientation based method, where green lines are true positives, and red lines are false positives.	27
24	The cause of the shift in localization when computing the gradient direction map. The transition from gradients which <i>agree</i> ($D_{ij} > 0$) with the dominant orientation almost always happens after the true crest-line due to the size of the kernel K	28
25	Effects of K on the orientation-based approach. In (a) a K value of 9 is used, which results in better localization at the cost of noisier results, (b) a K value of 21 produces much smoother results but poor localization.	29
26	The process flow of the machine learning approach. The image is first preprocessed, and some training data (samples of crest-lines and non-crest-lines) is reserved to construct the machine learning model. SIFT features are extracted from the training data to train the model to recognize crest-lines. With the model trained, SIFT feature descriptors are computed at each pixel of the remaining test images and are inputted into the trained model to generate the response map. Finally, the local maxima of the response map are detected in order to extract the crest-lines.	29
27	The SIFT descriptor explained. The image patch is split into a 4 by 4 grid, where each cell contains a histogram of gradients of eight directions. The result is a single vector of 128 floating point values. Source of the image: http://www.vlfeat.org/api/sift.html	30
28	Results of the Support Vector Machine classifier training for crest-line identification for the Terrestrial dataset shown in Figure 39. On the left is the response map (the output response of the SVM classifier using the SIFT features at each pixel). On the right, the thresholded the crest-line responses ($R_{i,j} > 0$) of the response map are overlapped on the original image.	33
30	A recursive algorithm for ridge following: On the first iteration, the algorithm is initialized using the detected anchor pixels, and begin search for each neighbor. Neighbors which are peaks along the ridge are used as anchors for the iteration. This algorithm recursively follows the ridge of the response images.	35
29	Results of the Gradient Boosted Tree classifier training for crest-line identification for the Terrestrial dataset shown in Figure 39. On the left is the response map (the output response of the GBT classifier using the SIFT features at each pixel). On the right, the thresholded the crest-line responses ($R_{i,j} > 0$) of the response map are overlapped on the original image.	35
31	The results of the recursive ridge following algorithm applied on the response map of the Kalahari image trained with the SIFT-GBT classifier. The results produced appear noisy and disjoint even with smoothing applied.	35
33	The process flow of the machine learning and gradient-based mixed approach. The image is first preprocessed, and the machine learning model training is identical to the approach presented in 3.8. Simultaneously, the dominant orientation is computed on the images, the gradient direction map is extracted, which is thresholded based on the dot product, and contours are extracted (in similar fashion to the approach explained in 3.7). Finally, the contours are filtered based on the response map, resulting in the detected crest-lines.	37
32	The results of Zhang/Suen Thinning [111] algorithm applied on the response map of the Kalahari image trained with the SIFT-GBT classifier. The output is much smoother and contiguous segments than the recursive ridge following method shown in Figure 30.	38
34	Computing the Gradient Orientation Map (b) ($D_{i,j}$) for the Kalahari image (a). The map is thresholded for all gradients which agree with the dominant orientation ($D_{ij} > 0$) in (c). The result produces smooth contiguous regions which contain candidate crest-lines.	39

35	Illustration of the issue of poor localization of the extracted contours. In (a), the binary image shown in Figure 34(c) is overlapped on top of the Kalahari image 34(a). In (b), a zoomed in image patch reveals the localization issue with the contours (shown in red) from the actual crest-line (shown in green). In order to improve the localization, the contour segments must be shifted towards crest-line. The shift is applied in the direction opposite to the computed dominant orientation.	40
36	The process of extracting candidate crest-lines using the gradient orientation map, and the machine learning approach. The Kalahari input image shown in (a), is processed to extract contours shown in (b). A shift towards the nearest peak in response values is applied (c), and all segments which have low responses are filtered out in (d). The color scheme shown in (b-d) is true positive crest-lines detections (green), the positive identified ground truth (blue), false negatives (yellow), and false positives (red).	41
37	The connected component algorithm applied a sample test image (a). The results (b) show that segments can be extracted correctly.	42
38	Computing the inter-dune distance using the average orientation. Once the average orientation has been computed, the inter-dune distance can be computed by measuring the distance between segments along an orthogonal axis to the average orientation.	43
39	Terrestrial dataset of six regions with respective labeled ground truth: (39(a)) Kalahari (Namibia), (39(b)) Namib (Namibia), (39(c)) Simpson (Australia), (39(d)) Skeleton Coast (Namibia), (39(e)) Winnemucca Dune Complex (USA), and (39(f)) White Sands (USA)	46
40	Ganges Chasms Mars dataset from [17] which includes (40(a)) 16 regions extracted, (40(b)) sample image from region 1 with corresponding labeled ground truth	48
41	Precision and Recall curves of the Gradient Orientation based approach on the Kalahari image for various values of K (the size of the gradient kernel).	51
42	Results of the gradient based approach projected on the image for the Skeleton Coast and WDC images. The true positives are shown in green and the false positives are shown in red. The overall precision and recall results are shown in Table 5.	52
43	Terrestrial test dataset of six regions with respective labeled ground truth: (43(a)) Kalahari (Namibia), (43(b)) Namib (Namibia), (43(c)) Simpson (Australia), (43(d)) Skeleton Coast (Namibia), (43(e)) Winnemucca Dune Complex (USA), and (43(f)) White Sands (USA), used for evaluation of machine learning approaches. Test	54
44	The results of the crest-line detection for the machine learning approach on terrestrial dataset for both the training images (a), (c), (e), (g), (i), (k), and the test images (b), (d), (f), (h), (j), (l) using the GBT classifier and SIFT features. Legend: <i>Green</i> - True Positives, <i>Red</i> - False Positives, <i>Yellow</i> - False Negatives, <i>Blue</i> - Ground truth correctly detected.	58
45	The results of the crest-line detection for the mixed machine learning and gradient approach on terrestrial dataset for both the training images (a), (c), (e), (g), (i), (k), and the test images (b), (d), (f), (h), (j), (l) using the GBT classifier and SIFT features. Legend: <i>Green</i> - True Positives, <i>Red</i> - False Positives, <i>Yellow</i> - False Negatives, <i>Blue</i> - Ground truth correctly detected.	61
46	Crest-line detection results plotted for all regions used in the Mars dataset. The Vaz [17] results (on the left) compared to our results (on the right). Legend: <i>Green</i> - True Positives, <i>Red</i> - False Positives, <i>Yellow</i> - False Negatives, <i>Blue</i> - Ground truth correctly detected.	65
47	Plots of the Precision and Recall comparison between our method and Vaz [17] for Area 1 (a) and Area (b) as the allowable distance threshold value ϵ increases.	67

List of Tables

1	Precision and recall results of the appearance-based approach described in section 3.3 for the Terrestrial dataset.	49
2	Precision and recall results of the edge-based line detection approach described in section 3.4 for the Terrestrial dataset. The results are shown for both the K-Means (section 3.4.1) and the histogram of gradients (section 3.4.2) approaches to computing the dominant orientation.	50
3	Precision and recall results of the watershed transform segmentation approach described in section 3.5 for the Terrestrial dataset.	50
4	Precision and recall results of the frequency domain based approach described in section 3.5 for the Terrestrial dataset.	50
5	Precision and recall results of the gradient orientation approach described in section 3.7 for the Terrestrial dataset.	51
6	True positive and false positive rates results of the training process to build the crest-line classifier models as described in section 3.8.3, using the SIFT descriptors for various classifiers (a) Support Vector Machines, (b) Normal Bayes, (c) Random Trees, (d) Gradient Boosted Trees	55
7	Results of the machine learning method presented in section 3.8 using a single Gradient Boosted Tree classifier and the SIFT features trained across all the regions, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.	56
8	Results of the machine learning method presented in section 3.8 using the Gradient Boosted Tree classifier and the SIFT features, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.	56
10	Results of the mixed machine learning and gradient-based method presented in section 3.9 using the Gradient Boosted Tree classifier and the SIFT features, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.	59
11	Results of the mixed machine learning and gradient-based method presented in section 3.9 using the Gradient Boosted Tree classifier and the SIFT features trained across all the regions, includes the Precision and Recall a and the Dune Metrics error results b for both the training and test image sets.	59
12	Precision and Recall results of the mixed machine learning and gradient-based method presented in section 3.9 (GBT-SIFT) trained on the Mars dataset presented in 4.1.2, compared to the results in [17]. Shown are the results of both the training set (odd areas) in (a), and test set (even areas) in (b).	62
13	Dune metrics results of the mixed machine learning and gradient-based method on the Mars dataset presented in 4.1.2. The dune metrics were computed on our crest-line detection results, the ground truth, and the crest-lines detected in [17]. Shown are the results of both the training set (odd areas) in (a), and test set (even areas) in (b).	66
14	Summary of the precision and recall results for each approach applied to the Kalahari, Namibia, Skeleton Coast, and WDC terrestrial images.	69

1 Introduction

Computer vision, image understanding, and machine learning are fields that have been thriving in the last few decades with advances in computing. The purpose of these are to extract information and understanding from images and various types of data. Many disciplines use the techniques to solve specific and challenging problems. Humans are typically good at recognizing and understanding these problems [50], but the work required to apply the solution can be either complex or tedious.

Along with the challenges and complexity associated with the problems is the data collection process. Technological advancements in the ability to capture, record, and store data from many different sources has provided increasingly large data sets which need processing. The amount of data is overwhelming for humans to manually process, therefore the need for automated solutions becomes even more significant.

In this research, the goal is to create an automated system able to analyze and extract geomorphological properties from satellite images of dunes by extracting dune crest-lines. Dune fields and crest-lines have various morphological properties which includes dune type, size, height, width, length, orientation, spacing or wavelength. In past research, the properties were measured manually by experts in the field using *Geographic Information Systems* (GIS) applications [23, 24, 25, 26, 55]. *Digital Elevation Maps* (DEM) have grown in popularity in recent years. DEM planetary data sets for Earth (e.g. SRTM, ASTER, GDEM, and other high resolution data sets), Mars (HiRISE, MOC), and Titan (Cassini radar) are now available and provides information on many of the aeolian bedform morphology, orientations, and dune types of each region. Therefore, a need for an automated, or semi-automated processing method and statistical analysis of the dune patterns has arisen.

Our research will focus on developing a automated system to extract, recognize, and characterize dune patterns on images of planetary surfaces. The requirements are that the application robustly and automatically (or perhaps semi-automatically) detects crest-lines from various dune morphological types, such as linear, complex, or other types. Additionally, the ability to automatically compute the dune properties such as crest length, orientation, and spacing could greatly benefit and provide a valuable tool for the research community.

The research done in [23, 24] shows that dune patterns can be characterized using the following metrics:

1. Trend or Orientation of dunes
2. Crest lengths of dunes
3. Dune spacing or wavelength
4. Defect density (breaks in the pattern, terminations, forks, bifurcations, etc) per unit length of the dune crest-line, used as a measure of the degree of organization of the field.

The metrics above are illustrated in Figure 1. These metrics are important properties which characterize the dune field, and enable researchers to deduct meaningful information about the particular regions. Ideally, any algorithm which processes the data should correctly and clearly identify crest-lines, characterize the dune types, and extract the metrics described above.

1.1 Challenges

In order to automatically compute the metrics, the dune crest-lines must first be correctly detected and identified. Therefore, the bulk of the research presented will focus on crest-line detection, along with some general metrics calculations. Crest-lines can be defined as series of linear or curved segments, where all points along the segments represent the dune ridge line, and the ends of the segments represent the termination or defects in the pattern.

There are many ways to represent crest-lines. The most basic representation could be a vector which holds the trend or orientation (azimuth) and length. This representation is simplistic and can provide a general overview of the dune field. Sets of linear segments also facilitate the computation of some of the metrics desired. However, there are issues associated with the linear segments representation.

1. Linear segments are only an approximate representation of the dune field and therefore there will be localization errors. In addition, dune crests are not always straight lines.
2. The process of determining which linear segments are part of the actual crest-line can be difficult since there may be many false positives present in a given image.
3. The data itself is quite noisy which will lead to many false positives in the detections.

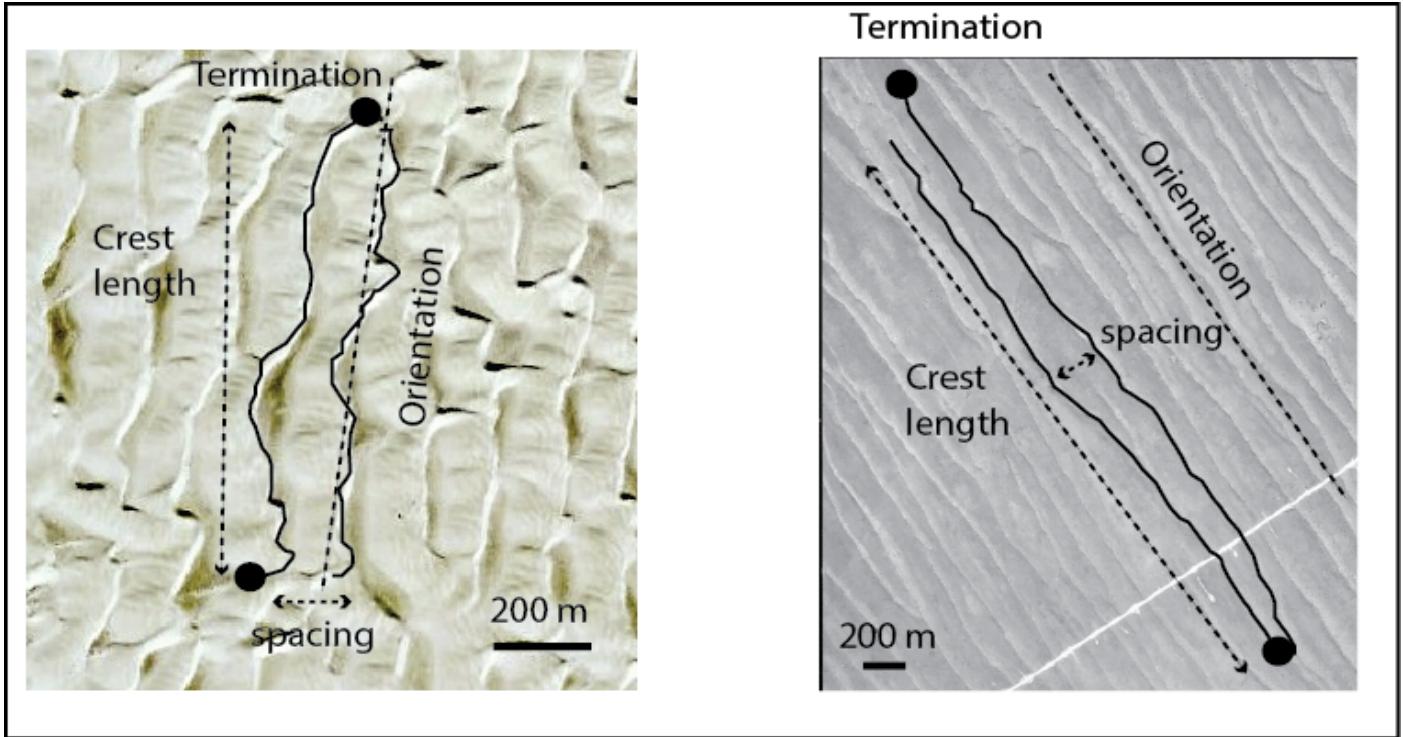


Figure 1: Examples of dune pattern analysis including metrics such as orientation, crest length, spacing, and termination points.

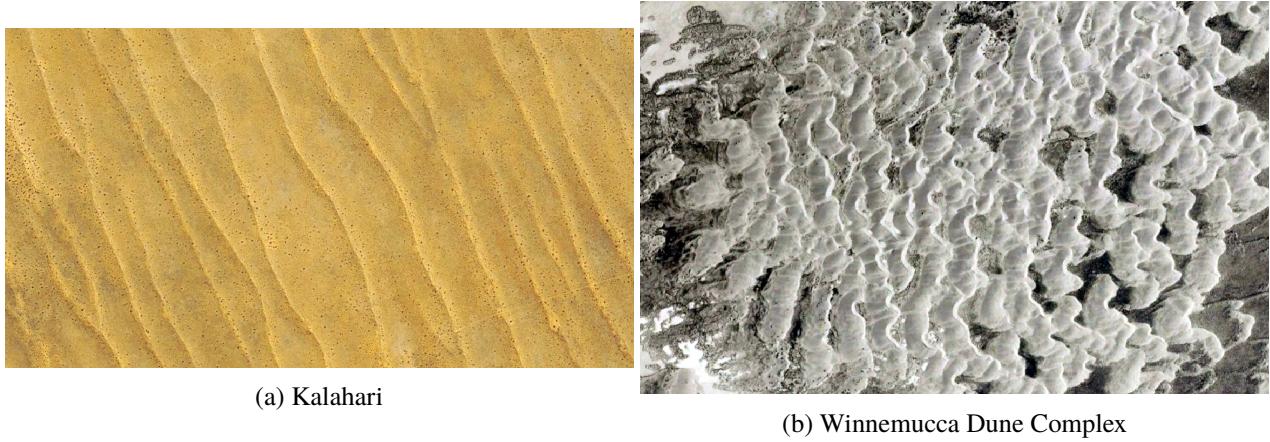


Figure 2: Sample satellite images of two distinct dune field regions: Kalahari and Winnemucca Dune Complex

4. Computing accurate geomorphological metrics becomes challenging.

A better representation would be a contiguous and smooth vector of two dimensional points which plot a curved segment, with the first and last points representing the location of the termination points. This is the representation which is favored and will be used as part of this research. The contiguous segment representation also provides a more accurate localization of dune crest-lines, and should produce more exact dune metric calculations.

Other challenges with automated crest-line detection are the algorithms and image processing tasks applied to the data sets. In satellite images of dune fields, the sun's orientation with respect to the image source greatly determines the appearance of the dunes. From an image representation perspective, a dune appears to have both a sunlit side, and a shaded side, as shown in Figure 2. This translates to bright regions (large pixel values) for sunlit areas, and darker regions (low pixel values) for shaded areas. Crest-lines, or dune ridges, can be defined to be the edge segments where the bright areas meet the darker areas in the image.

Issues arise when the sun's orientation is unknown, namely determining if an edge belongs to a crest-line ridge or is in fact a valley or the edge of a shadow. If the sun's orientation relative to the image is known, determining which edges are crest-lines or valleys becomes a much easier task. However, working under the assumption that the sun's relative orientation is unknown (which is the assumption made in this research), the ambiguity requires a resolution. In some cases, such as the Kalahari dune field shown in Figure 2(a), determining the sun's orientation can be inferred from the image itself, since the dune ridges are relatively linear and well defined. On the other hand, in areas like the Winnemucca

Dune Complex or WDC, shown in Figure 2(b), the sun's orientation cannot easily be determined. The factors which determine which edges are the crest-lines are not obvious, even for a human.

Another complexity which must be accounted for are the computer vision algorithms chosen to detect crest-lines. There are generally two approaches to solve these types of problems: appearance based and machine learning based approaches. Appearance-based approaches are typically fine tuned algorithms designed specifically to solve a problem set. The main drawback of these techniques are that they require significant manual tuning of parameters to achieve the desired accuracy to solve the problem. On the other hand, machine learning methods learn the problem set by example, and typically require many less parameters, which can be self optimized. Unfortunately, machine learning typically require a data set of training examples to construct the model to solve the problem. In this research, we explore both approaches using many different kinds of methods, specifically applied to the dune crest-line detection problem.

Once the crest-lines have been identified, the final challenge is computing the geomorphological properties of the dune fields. There are many levels of detail which can be computed for the metrics. At a coarse level, the global properties can be averaged for the entire dune field, while at a finer grain level, the metrics can be computed in local areas of the dune field around each crest-line. The latter is obviously more desirable, but requires more extensive and complex computations to accurately compute the desired metrics.

1.2 Contributions

The main contribution of this research is the development of a method to detect crest-lines from satellite images of planetary dune fields. The bulk of the research focuses on various regions on Earth, and the Ganges Chasma on Mars, also used in [17]. Each region contains a data set of manually labeled satellite images, which includes all ground truth crest-lines.

A variety of methods were applied on each image set, and evaluated using the ground truth. Both appearance-based methods and machine learning are used to extract the crest-lines. The appearance-based method uses a plethora of features such as intensity values, gradients, and frequency domain analysis, and other general image processing to attempt to find crest-lines. Machine learning methods are also used to construct models which can identify crest-line structures in the image. Each method is evaluated using the labeled ground truth, where the quality of each method is measured by precision and recall of the detected crest-lines, and the accuracy of the geomorphological metrics calculations. Each method has its own strengths and weaknesses, therefore the final proposed method combines both the edge-based and machine learning methods.

The gradients in the image play an important role in discerning crest-lines. A critical piece of the algorithm is determining the dominant orientation of the dune field which is the overall trend of the dunes. It can be computed from the gradients of the crest-lines, which are affected by the sun's orientation relative to the dune field.

Determining which gradients belong to the crest-lines and valleys is a challenging problem. A few solutions are proposed to solve this problem, including unsupervised methods such as K-Means, and normalized histograms of gradients. Once the dominant orientation gradients for crest-lines have been determined, the crest-lines can be more easily identified by keeping all gradients which *agree* with the overall dominant orientation gradient.

Another important contribution is the *gradient orientation map*. This maps the dot product of the gradients at each pixel with the computed dominant orientation gradient. In essence, pixels which *agree* with the dominant orientation are emphasized separating shaded and bright sides of the dunes, which allows crest-lines to be easily traced.

The final proposed solution combines the dominant orientation, gradient orientation map, and a machine learning response map to find the crest-lines. The gradient based approach is used to extract initial candidate crest-lines. The response map is then used to both improve the localization of the crest-lines and to filter out false positive candidates.

A method for computing the dune field morphology metrics is also proposed. The main metrics computed are the overall orientation of the crest-lines, and the spacing between the crest-lines. The orientation is computed after crest-lines have been detected, which is done by quantizing the detected crest-lines into linear segments and computing the orientation of the lines. The dune spacing is computed by measuring the orthogonal distance between the lines.

The research is split into the following sections: the previous work and background is presented in section 2, which presents an overview of the study of dune fields, and work done in the field of image processing and computer vision, and previous dune detection implementations. In section 3, the methodology is presented, including the various methods tried both for appearance-based, edge-based, and machine learning methods. It also presents the method for computing the geomorphological properties of a dune field. Section 4 presents the datasets used and the results of the detection and metrics for each approach. Finally, section 5 discusses the methods and their results, along with proposing some future work on improving the approaches.

2 Background and Previous Work

In this section, the previous work done in this application is reviewed. The need for dune crest-line detection has risen recently due to many research applications in this field. Therefore, understanding the concepts and properties of dune fields is important. Section 2.1 will present an overview of some of the research done in this field. Section 2.2 will discuss some of the related algorithms and methods for processing images and computer vision used in this research. Finally, section 2.3 will present the state of the art dune detection research which uses both appearance-based and machine learning approaches.

2.1 Dune Pattern Studies

Many studies have been done on the topic of dune fields, with the ultimate goal of understanding how dunes form, behave, and which factors contribute to the systems. In [31], an explanation of how dune-field patterns emerge is proposed and the degree of complexity from the standpoint of self-organizing system is discussed. Dune patterns are classified into two basic categories: simple or complex. According to this paper, a simple pattern is defined as having a single pattern type. Complex patterns are said to potentially have multiple spatially superimposed pattern types. Simple patterns trend towards a better ordering, so long as the wind regime remains constant. Changes in the trend of wind pattern will cause reorientation. Complex patterns can then be interpreted as the superposition of many generations of wind regimes.

Complexity can be measured using pattern analysis, measuring parameters such as the crest length, orientation, spacing, defect density, and other properties. These properties would be very useful information to extract for the dune detection. Finding crest-lines can help us identify the patterns, extracting meaningful data such as junctions, terminations, mergers, linking, and other interesting processes explained in [31].

In later work, [99] studied the dune field pattern formations on the north polar region of Mars. These dunes are thought to be mostly inactive, with relatively no movement over a period of 4 to 15 Martian years. The reason this region is interesting is because there are two nearly orthogonal crest-line orientations present in the region. This complex system of superposition of multiple patterns in this example showcases a set of *primary* and *secondary* crest-lines which would be valuable data to automatically segment. The *primary* dunes are the largest-scale dunes, extend over the entire length of the image set, and contain many *Y* junctions, an indication of well-organized linear dunes. In contrast, *secondary* dunes are rounded, have less defined features, and are perpendicular to the main primary crest-lines.

Other interesting features are the so-called *Slipfaces*. These typically appear along the *primary* crest-lines, in areas of intersection with the *primary* and *secondary* crest-lines. Another feature are the *Wind Ripples*. The ripples are present on the surface of most dunes with the exception of *slipfaces*. To detect these types of features, a higher resolution image is required, as those types of features are very small compared to the scale of the *primary* and *secondary* dunes. *Interdune Areas* are features specific to the area studied in [99], have a polygonal shape, and are indicative of ice.

The rest of the paper describes an in depth statistical analysis of the features to compute the flow fields and understand the geomorphic relationships which are present in the area. The paper [99] provides a lot of interesting and valuable information which can be used to understand the application of automatic dune detection. If these features can be extracted, it would be interesting to see if we can do a similar statistical analysis based on the results presented. The next step would be to retrieve the HiRISE dataset and their data or ground truth to compare.

In the most recent work, [97] study the aeolian dune-fields at different scales. Scale is an important factor to consider because aeolian dune-fields patterns can vary over a wide range of scales, both spatially and temporally. Being able to measure the change of scale over time is important in order to investigate the environmental conditions of the studied region. Aeolian dunes are developed typically in transitions from sand patches, to proto-dunes, to dunes, to dune-field patterns. Complex dune patterns are usually a juxtaposition of simple dune patterns at multiple scales. To summarize, being able to detect the crest-lines and other types of features at multiple scales is invaluable.

In other related work, [22] studied the migration of submarine sand dunes. The dataset used in this paper includes digital terrain models (DTMs) retrieved from high-density multibeam echosounders (MBES) taken of submarine sand dunes along the coast of New Brunswick. In order to measure the migration of these types of dunes, the motion was measured by simply subtracting the DTMs from sequences over time. The implementation uses a simple cross-correlation to find similarities from one set to the next. From the correlation matches found, the migration vector can be computed. The migration data processing can extract the flow fields of the dunes.

The research presented above is only a subset of the work done, but provides very valuable information on how, why, and what dunes are. The next section will describe some of the state-of-the-art technology in the field of image processing and computer vision.

2.2 Image Processing and Computer Vision

The main goal of image processing is to enhance and improve the quality of images in order to improve the performance of feature extraction. Computer vision is a field in which information is extracted from images for the purpose of understanding the image to solve a problem for a given application.

2.2.1 Preprocessing

For this application, the satellite images of the dune fields need to be processed using various known processing methods. Filtering out noise in images is critical to extracting good crest-line candidates. There are many common filtering approaches used in image processing and computer vision applications. In this application, satellite images of sand dunes may include various amounts of noise. In order to improve the dune detection algorithms, the goal of filtering is to filter out the noise while preserving or enhancing the necessary features such as the crest-line edges. In [11], the bilateral filter is proposed, which allows flat noisy regions to be filtered while preserving strong edges. Bilateral filtering is a simple non-iterative algorithm for filtering out noise.

Another key process is to normalize the illumination of the dune fields. Illumination normalization is a process in which the image illumination is made to be more evenly distributed across the image, to make the illumination invariant in conditions of varying lighting. This processing is most commonly used in face detection or recognition applications. In [124], a Gaussian illumination model is used to stretch the contrasts of darker areas. The process involves using a Quadtree method to divide the image into sub regions to find dark areas. The areas that meet these requirements are processed with the Gaussian illumination model to brighten them up, which evens out the illumination. Applying this normalization significantly improves performance of face recognition.

In a similar approach [34], the authors use a modified retinex algorithm to restore poorly illuminated areas of an image. According to Lambertian reflectance theory, images consist of two components, reflectance and illumination. If an image contains a set of small scale features, and large scale features, and illumination is only applied to the large-scale set, then the small scale features can be preserved. In this approach, the normalization is applied to small-scale and large-scale features independently. The Single Scale Retinex process is applied to the image to separate the reflectance (small scale) and illumination (large scale). After, images are thresholded and histogram equalization is applied, which spreads out intensity values of darker areas of the image.

In a remote sensing application, [38], illumination normalization is carried out in the gradient domain and using singular value equalization. According to the authors, the gradient domain image enhancement process can compress the dynamic range of images, increase contrast, and enhance fine details in darker regions. The goal of singular value equalization (SVE) is to make all images in a specific set to have similar mean intensity value. Some images may have a lower or higher mean intensity. By using SVE, the pixels can be processed to achieve a mean closer to the target intensity, therefore normalizing illumination across a set of images. The drawback of this method is that multiple images of the same or similar scenes are required. Many of these techniques can be applied directly to processing the satellite images of the dune fields in order to improve and normalize the illumination.

Another critical processing step is to extract edges from the images. Edges are extracted by taking the derivative of an image, and preserving the strongest gradients. There are many operators used to extract edges. The Canny edge detector, first introduced in [12], is a well known and used method to retrieve important edge features from an image. This famous detector is often used because of its qualities in detecting edges. The Canny edge detector typically produces a binary image of thin, contiguous, and well defined edge segments which makes it a good candidate solution for edge extraction.

In [83], enhancement to the method is proposed by analyzing the responses of detection at two scales. The benefits of this technique are better localization in images with larger amounts of noise, at the cost of a slightly lower detection rate. Overall the performance of the Canny operator is improved by using multiple scales.

The Canny detector is further improved in [122] for a runway detection application for unmanned aerial vehicles. The main contribution of this paper is the use the Canny operator combined with a mean filter, and using the Hough Transform to track runways. The advantages of the mean filtering is that it filters out noise, preserves edges (better than a traditional Gaussian filter), and makes the image less fuzzy. To solve the problem of the dual threshold of the Canny operator, the proposed method is to compute the thresholds dynamically based on averages.

In [119], an improved implementation of the popular Canny edge detector is proposed. According to the authors, the traditional Canny algorithm suffers from two main problems: the gradient calculation is sensitive to noise and the use of the fixed double threshold may not be suitable for images with high gradient variability. The main flaw in the gradient calculation has to do with the inequality of the edge detection in darker versus brighter regions. For the parameter selection of the double threshold, the paper propose choosing the high and low thresholds from the computed mean and standard deviation of the gradient magnitudes.

Another problem with edge detection methods can be the discontinuity of edges. The goal is often to link edges which

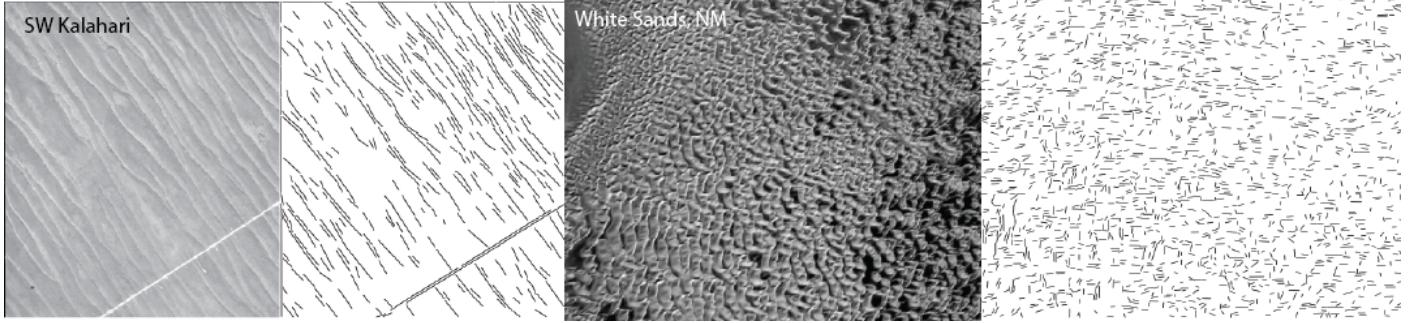


Figure 3: Preliminary results of the Line Segment Detector (LSD [118]) applied to the linear dunes of the Kalahari (left) and transverse dunes in White Sands (right). The results of the LSD on linear dunes are good, while complex transverse dune results are poor.

may suffer from noisy conditions in order to produce smoother contiguous segments. In [125], an edge linking approach is proposed which uses local neighborhood, using geodesic distance (as opposed to the common Euclidean distance measure) between edge candidates. When calculating the direction of edge end points, typically eight directions are used, which introduces error. To address this, [125] define a window size based on the maximum allowable edge gap, and fit a line to the edges, which allows a full range of directions to be accounted for. The geodesic distance measurement is not only based on Euclidean distance but also on the intensity values of the image, which reportedly gives better results.

2.2.2 Feature Detection

Once edges have been detected, the next logical step is to extract information from the edges. In previous work, in order to detect dune crest-lines, the edges needed to be organized into line segments so geomorphological properties can be computed. Computing the orientation and dune spacing of the dune field on the satellite images requires an algorithm that can accurately detect line segments present in an image.

Line detection is a well known and challenging problem in the field of computer vision. Traditional approaches such as the Hough Transform presented in [21], use edge detectors such as the Canny to extract line segments. Unfortunately, choosing the correct parameter and an appropriate threshold for preserving true positives while eliminating false positives for any task can be difficult. Inappropriate selection of the parameters causes inaccurate results in the computation of the geomorphological properties. Improvements on the technique proposed in [8] and [18] improve the results slightly but basically suffer from the same parameter selection problems.

An interesting alternative to the classical Hough transform is presented in [48] in an application inspection or maintenance of underwater cable components. The paper discusses image processing techniques to extract linear features in cluttered and noisy images. After applying some anisotropic filtering to remove high frequency noise, edges and lines are detected on the sonar based images. To detect potential linear features, they use a Phase Congruency detector. The Hough transform is then used to find linear features, and a criteria is determined to reject false positives and preserve true positives.

Another promising approach to line segment detection is presented in [118] which combines the features of the techniques presented above. In this approach, adjacent edge points are grouped by orientation, and lines are fitted to the local support region. Validation of line segments is applied using a threshold of the number of points which support the given line. The benefit of this approach is the ability to detect line segments in complex structures using minimal parameter selection.

Some preliminary work was done testing the Line Segment Detector (LSD) on some dune field images, shown in Figure 3. The method seemed to perform quite well on linear dune formations found in regions like the Kalahari, but performance was quite poor on regions with non-linear complex or transverse dune formations, such as the White Sands dune field.

A line detection and image enhancement has also been proposed in [33], for a digital image restoration of heritage artwork. The images worked with have been deteriorated over time, and contain many cracks, faint lines, and broken strokes. The goal of this is to remove unwanted edges, and enhance the desired edges. The approach is implemented in three basic steps: Initial line detection with non-maxima suppression, true line detection using anisotropic refinement, and noise reduction. The initial step is to perform correlation convolving the image with some sort of edge detecting mask, rotated at different orientations, and retrieving the maximum response for the orientation for each pixel. The paper claims this produces a sharper and more reliable edge map, which is then filtered using non-maxima suppression. This process preserves strong lines while removing texture lines. The image is then binarized to remove all remaining weak

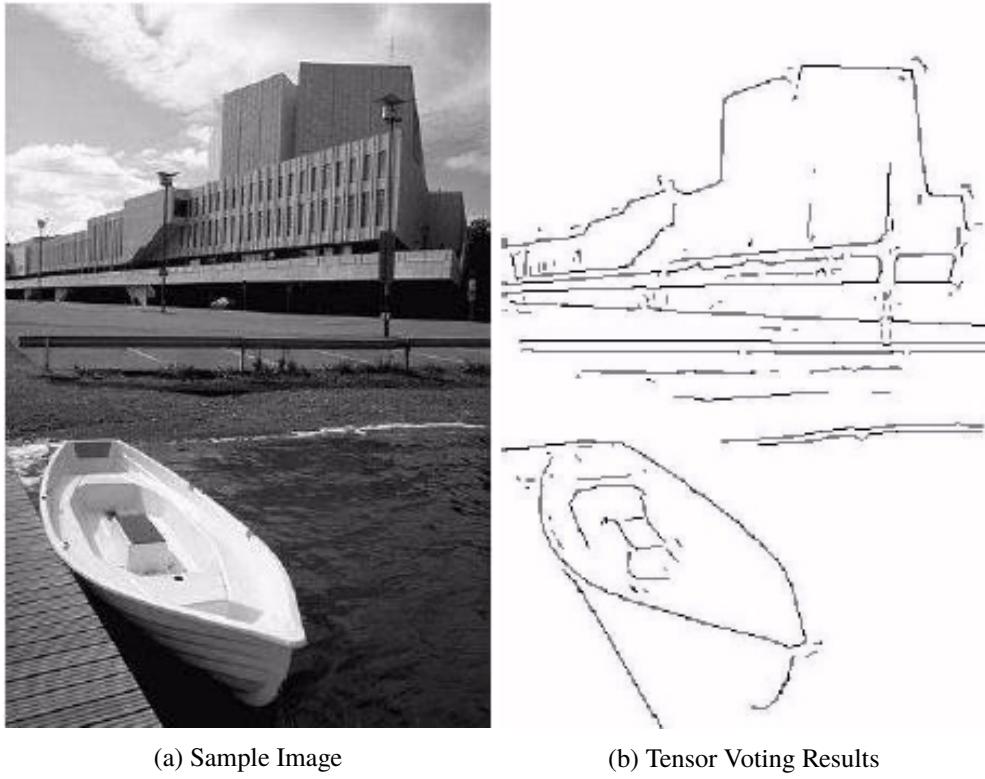


Figure 4: The results of applying tensor voting schemes to find salient linear segments in an image ([65])

edges, and only allowing strong edges to remain. The smoothing is applied to the original based on the processing done, smoothing out weak edges, while not smoothing strong true edges.

In a similar approach, [104] use rotating edge detection kernels and fuse the responses from the operations. An edge is defined as a location and a direction compared to a non-edge point such as a flat area or noise, which has no specific direction. Following this principle, the response in one direction should be higher than in the other directions. For a non-edge, the responses in each of the directions are similar. Therefore computing the mean and standard deviation of each of the kernel responses can help resolve edge versus non edge pixels. The results obtained from this process shows promising results when applied to noisy images. The approach proposed in [104] has definite potential for improved edge detection applied to dune crest-line detection. It also has potential for improvement of the method itself.

Tensor voting frameworks are also a potential way to extract linear segments from images, and could be useful in a dune detection application. Tensor voting has been used in various applications in computer vision such as in [52, 53, 75, 78, 112]. One of the interesting benefits of tensor voting approaches, as explained [65], is that they seem to show very good results in grouping oriented segments in highly cluttered images. The Gestalt principles of visual perception are used as the voting scheme where second order tensors are used to represent the segments. The goal is to filter out background segments while preserving the more salient segments. Figure 4 shows the result of applying tensor voting to a sample image.

In [35], a tensor voting approach is used to perform perceptual grouping on noisy data. The approach extends the standard tensor voting using Bayesian probabilities. The Bayesian framework has added benefits of reducing the learning variance of the perceptual grouping task. Another contribution of this paper is the use of a 2nd order tensor and two types of polarity vectors to handle both outliers and inlier noise. The approach also is able to process all types of geometric structures. For probabilistic tensor voting, the voting procedure uses random vector with a given standard deviation. It uses three layers of voting: sparse ball vote, sparse stick vote, and revised stick vote, where the goal is to change the positions of each token based on a random vector.

Another interesting approach for extracting features is segmentation. One particular method used in segmentation processes is the Watershed Transform, an approach which is used in this research and explained in section 3.5. The Watershed Transform is a well known algorithm for segmenting meaningful regions from an image. The main benefit of the watershed technique as opposed to methods, such as edge-based detection, is that the watershed produces closed boundaries, which allows contours to be detected. An overview of the Watershed algorithm is presented in [94].

In [84], segmentation is achieved using the watershed transform from texture gradients. The texture gradients are retrieved using various wavelet transforms. Similarly, in [105], an approach to texture segmentation is proposed. When applied to textures, the watershed algorithm tends to over-segment a texture into small homogeneous sub-texture regions. Therefore, a texture can be determined to be set of texture segments. The texture sub-regions can be merged using a

clustering algorithm. A wavelet transform is used to characterize the irregular shaped segment regions, which are used to cluster similar groups of sub-textures.

Another approach used in the Watershed algorithm is based on markers. The desired regions to segment can be extracted using some known feature which is marked on the image. In [100], this concept is applied to fingerprint pore extraction. Marker based segmentation has been shown to be robust for segmentations problems where the objects are closed contours with boundaries being ridges. In this application, the watershed transform is applied to the gradient image of the fingerprints, and the regional minimas are computed to obtain the markers, then compute the watershed transform based on the markers. In [62], the watershed segmentation is applied to a breast tumor detection application. All these various researches have the potential to be useful in our dune crest-line detection application.

Ultimately, the goal is to perform automated feature detection from satellite images. In [113], a semi-automated method for extracting roads from aerial or satellite images is proposed. The purpose of this paper is to localize roads from higher resolution images, using some form of segmentation. The proposed method is semi-automated because a user must provide an initial seed point, onto which a region growing algorithm is applied to extract the road. A level set method is applied to evolve the boundary of the road region, which can better handle noise and multiple road mergers. The boundary for the desired region is based on a speed function, which is designed based on uniformity, texture, and contrast properties. Once the regions have been extracted, the centerlines of the roads are estimated using a skeletonization procedure, with junction analysis to ensure that intersections are properly handled.

In [14], linear features of internal waves are extracted using a technique called *Multiscale Retinex*(MSR) feature extraction. Although the problem set in [14] is different, there appears to be significant overlap and comparison between dune crest-line detection and oceanic internal wave detection. The oceanic internal waves are typically generated from many sources such as tidal currents, ocean frontal boundaries, and other atmospheric conditions. The MSR is an image processing technique that provides consistency and dynamic range compression across an image with poor contrast. The paper discusses the use both the Wavelet Transform Modulus Maxima (WTMM) and the Canny edge detector, which turned out to have superior performance.

2.3 Dune Detection

The specific application of this research applies to automated detection of dune crest-lines. Some work has been done in automated dune detection. There are two main methods used to detect dune landforms and crest-lines: appearance-based and machine learning methods.

2.3.1 Appearance-Based Methods

In [68], a Geographic Information System (GIS, [27]) based model is proposed and developed to study sand dune migration patterns, with the ultimate goal of predicting threats to roads, irrigation networks, water resources, urban areas, agriculture and infrastructures. Multi-temporal images of a region were collected, enhanced, and a simple image subtraction technique is used to compute the migration shift of dunes.

An automated extraction of dune features approach is proposed in [76], which uses the Sobel gradient operator to detect crest-lines. In this approach, the authors claim that the Canny edge detector provided poor results, and the Sobel operator generated more consistent results. The gradient magnitude of the Sobel operator is used to threshold out the weaker edges. The use of a histogram to compute the orientations of the gradients was found to have a bimodal distribution, with one of the modes being the dune edges. Dune candidates are determined based on their gradient magnitude which filter out weaker candidates unless they are near strong candidates. This research addresses the issue of determining crest-lines versus valleys or shadows, explained in section 3.1, by using the histogram of gradients. Examination of the histogram reveals that they had a strongly bimodal distribution for which one peak represents crest-lines and the other represents the valleys or shadows. The assumption made in their research is that the stronger peak represents the crest-lines, and therefore all gradients which belong to the weaker peak (within 90 degrees of the peak) are filtered out. This assumption may not always be valid, and is addressed in section 3.4.

In [80], an overview of various line detection algorithms are reviewed in the context of a marine dune crest-line extraction method. The research evaluates four main appearance-based approaches ([13, 54, 123]) to detecting ridges from MultiBeam Echo-sounder System (MBES) data maps. The data for these approaches is essentially 3D depth maps of underwater dune forms. The challenge therefore is to trace the ridges to extract the dune crest-lines.

2.3.2 Machine Learning Methods

Many of the research techniques to detect and identify various landforms use commonly known machine learning techniques. In [20], object-orientated image analysis is used to classify the type of landform elements found in Digital Terrain

Models (DTM). Features such as elevation, curvature, and slope gradient are used to construct a classification model, which categorizes landform elements into nine distinct base shapes. The shapes are somewhat abstract and generalized, but the concepts presented in this paper could be applied to crest-line detection.

The research done in [110] developed a machine learning based approach to map and categorize various landforms on Mars. In this research, it is shown that regions can be clustered using unsupervised learning methods, and labeled using supervised learning, given a dataset which has been manually labeled. The popular Support Vector Machines classifier is used to predict the label of the clustered regions. Similar methods will be addressed in sections 3.4, 3.8, and 3.9.

The research done in [10] uses a simple approach in which the correlation is computed between two images taken at different times within a year. Various window sizes are used to account for dunes of different scale. An interesting observation is made; as the window size grows larger, the correlation values tend to increase in areas where dunes are present, while remaining relatively constant in areas without dunes. A supervised learning method is used to classify areas which dunes are present.

In [66], a supervised learning approach is used, training classifiers such as Support Vector Machine and Random Forests to detect dune structures on Mars. The method proposed in this paper is to classify small (40 by 40 pixels) cells in a quantized image grid. In each cell, features are computed based on the image gradients, using both phase and magnitude. In order to classify a cell as either a dune or not a dune, the features of both the cell and the cell's neighboring cells are used. The features extracted are then used to train the machine learning method, which is used to then predict if a cell is a dune.

Although this type of method has typically shown very good results, there are a few drawbacks with using supervised learning approaches. These types of methods usually require a fairly large labeled dataset which may not always be available. Anytime a dataset is constructed for this purpose, it is important to provide a large number of examples of different types of dunes, in order to get a robust representation of the problem set. In [66], 230 labeled images were used to train and test the method, and have a decent representation of various dune types. Another drawback of this approach is the use of cells, for which fixed-sized cells may not be scale invariant. Also, quantizing an image into larger cells will affect localization accuracy of the dunes. If the application requires higher localization accuracy, this type of supervised learning approach may not be suitable.

In [90], texture features are extracted from images to train a multilayer perception or neural network to generate landform maps. The texture features used are the Grey Level Co-occurrence Matrix (GLCM [88]), are fed into the neural network with goal of characterizing various landforms. A critical problem to solve is the selection of the window size for computing the texture features. Reportedly, smaller window sizes, relative to the spatial resolution, negatively affect the classification accuracy. The window size parameter is also addressed in section 3.8. A large dataset was used to train the neural network, and shows promising results.

The most current and direct research application to dune crest-line detection is presented in [17]. An object-based image analysis method is used to extract crest-lines from dune field datasets on mars. The main approach extracts features from the dune field images, and trains an artificial neural network classifier. Once the classifier is trained, the crest-lines can be extracted and the dune field can be analyzed. The object-based image analysis methodology is very similar to our research, and the datasets used in this paper were acquired. Section 4.3 shows the comparison of the results reported in [17] and our results, using our computed dune metrics.

The remainder of this paper discusses the various methodologies (section 3) implemented as part of this research. The approaches include both appearance-based and machine learning methods to crest-line detection. The results of each approach are then presented in section 4.

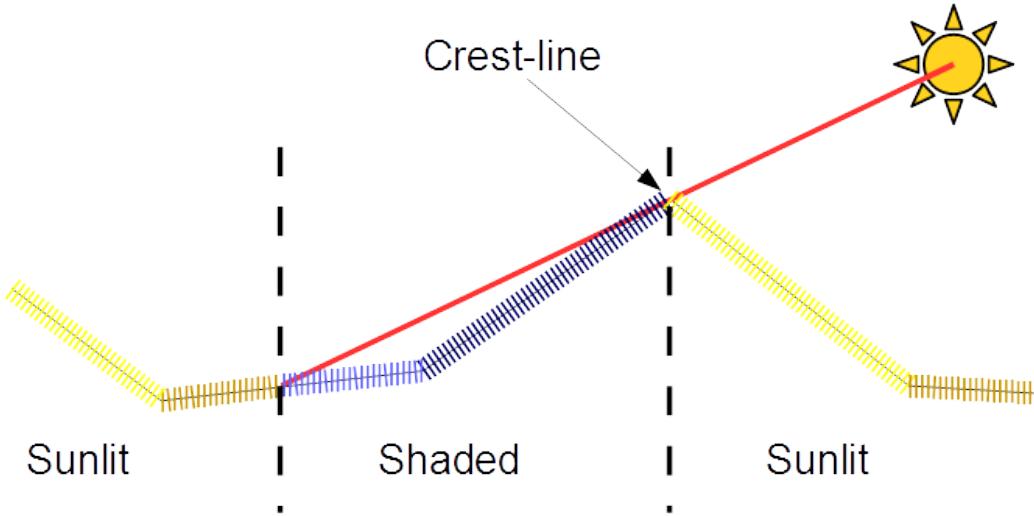


Figure 5: Cross-section of a dune slope with the sun's orientation effect on shaded and sunlit portions of the dune.

3 Methodology

The primary goal to reach when analyzing dune field images is to compute and extract meaningful features, as discussed in [23, 24]. The two main features extracted in this research are the following:

Orientation The orientation (both local and global) of a dune field can provide researchers valuable information about climate and wind patterns in the area.

Inter dune spacing The distance between crest-lines also provides valuable information about structure of the dune field.

There are many potential methods to extract such features, but in this research, a crucial first step is to detect crest-lines in the dune field. Crest-lines are the main prominent features in any dune fields. They are characterized by a distinguishable peaks along a ridge of dunes. Extracting the dune crest-lines is a challenging problem in itself, which is explained in 3.1. In this research, many different approaches have been attempted, which include an appearance-based approach in 3.3, an edge-based approach in 3.4, using the Watershed Transform in 3.5, a gradient orientation based method in 3.7, a machine learning approach in 3.8, and ultimately the mixed gradient and machine learning approach which is explained in 3.9. Finally, the computation of the dune field morphology metrics is explained in section 3.10.

3.1 Challenges

Automated detection of crest-lines from satellite images poses many problems. Finding and labeling crest-lines in images in most cases may be a straight forward task for an expert researcher in this field, but quickly becomes a tedious task as the amount of data increases. The need for automated detection of such features is very apparent. However, there are regions where dune fields are complex and dense, making automated dune detection a much more difficult task.

Generally, a crest-line is visible by the fact that it has a sunlit side, and a shaded side. Areas in the image where a bright region meets with a darker region are typically good candidates for potential crest-lines. In Figure 5, a cross-section of a sample dune ridge illustrates the effects of the sun's orientation on the illumination of the dune field.

This important criteria assumes that the sun's orientation is roughly perpendicular to the dune ridges. When ridges approach a parallel orientation to the sun's orientation, the crest-lines become less pronounced. The problem is evident as shown in Figure 6, from linear dunes in the Kalarahi (6(a)) to the complex structure of the White Sands dune field (6(b)).

Another important feature present dune crest-lines are the edges between the sunlit side and the shaded side. One assumption is that crest-lines produce relatively stronger edges than other areas of the image. This assumption was the principle guiding criteria in [76]. However, the strength of the edge depends on many factors:

1. The sharpness of the actual dune ridge: dune which have sharper crest-lines will undoubtedly produce stronger, more defined edges on the satellite images.
2. The sun's orientation and time of day: As the sun approaches the horizon, the sunlit and shaded sides of the dunes become more defined. Inversely, when the sun is higher in the sky, shadows become less pronounced. Also, as discussed previously, the orientation relative to the dune field itself affects the strength of the edges.
3. Scale, noise and resolution of the image can all affect the strength of edges.

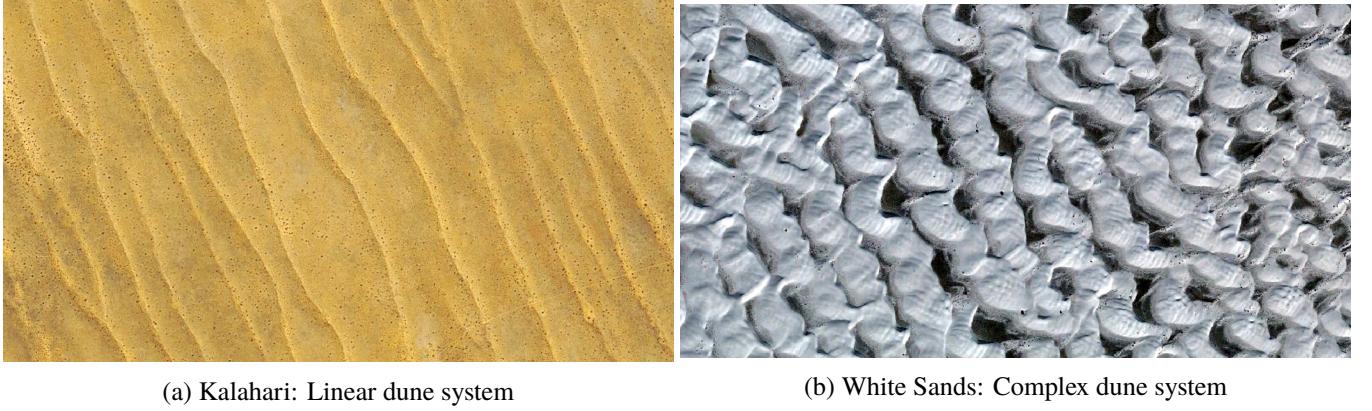


Figure 6: The variation in dune patterns such as (a) linear, or (b) complex, and the sun orientation and illumination effects on the resulting satellite images.

Understanding the issues with the gradients of the key problem of this research. As shown in Figure 7, the pixel values and gradient magnitudes (edge strength) can vary greatly in each region. One inclination is to select strong gradients in an image as candidates for crest-lines (as done in [76]). However, the gradients of crest-lines are not always larger than the gradients of valleys or the shadows.

In Figure 7, a plot of the intensity, gradient magnitudes, and gradient orientations along a cross section of a crest-line is shown. In the sample image patches 7(a), 7(d), 7(g), the intensity values of the pixels along the direction shown are plotted in blue in Figures 7(c), 7(f), 7(i), the gradients magnitudes are shown in red, and the gradient orientations are shown in a dotted black line.

Typically, a peak in intensity values of the pixels is present, representing the sunlit side of a dune. Inevitably, this will produce two peaks in the gradient magnitude on both sides of the intensity peak. One of them represents the true crest-line of a dune, while the other can either be the shadow cast by the dune, or the bottom valley of the dune. The main difference between the two peaks, other than their gradient magnitude, is the fact they are almost always in opposite orientations. One peak is going from dark to bright, the other from bright to dark. If the sun's orientation relative to the dune field is known, then choosing the correct peak is trivial. When the orientation is unknown, choosing the correct peak becomes a challenging problem.

In [76], the assumption was made that the weaker of the two edges can be filtered out, claiming that the larger peak is the crest-line, which is valid in the Kalahari image patch in Figure 7(c). However, the gradients produced by crest-lines may or may not be larger than their cast shadows, or the valleys. A clear example is shown in Figure 7(f). In this scenario, the true crest-line is actually represented by the first gradient peak. The second peak is in fact produced by the shadow of a different dune.

Another interesting challenge is found in the images themselves. The correct crest-line cannot easily be determined by simple observation. A person presented with the image patch shown in 7(d) would most likely not be able to determine with certainty which one of the edges represents the true crest-line. The obvious solution is to know the orientation of the sun relative to the dune field, but that may not always be possible.

All the challenges listed above need to be addressed in order to accurately and reliably detect dune crest-lines. In the next section, the overall process flow of the crest-line detection application is presented. Following sections will discuss the various approaches to solving the problems are presented as part of the research.

3.2 Overall Process Flow

In this research, many different approaches were attempted and compared to extract meaningful features from the images and compute the appropriate morphological properties of the dune fields. The overall process flow of detecting crest-lines can be represented as a five stage process:

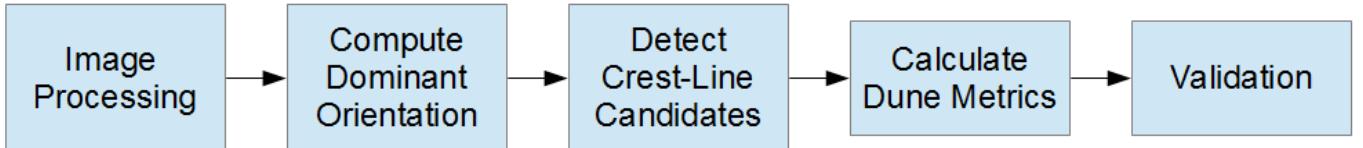


Figure 8: The overall process flow for general dune detection. The five steps include image processing, dominant orientation computation, crest-line candidate detection, calculations of dune metrics, validation of crest-line candidates.

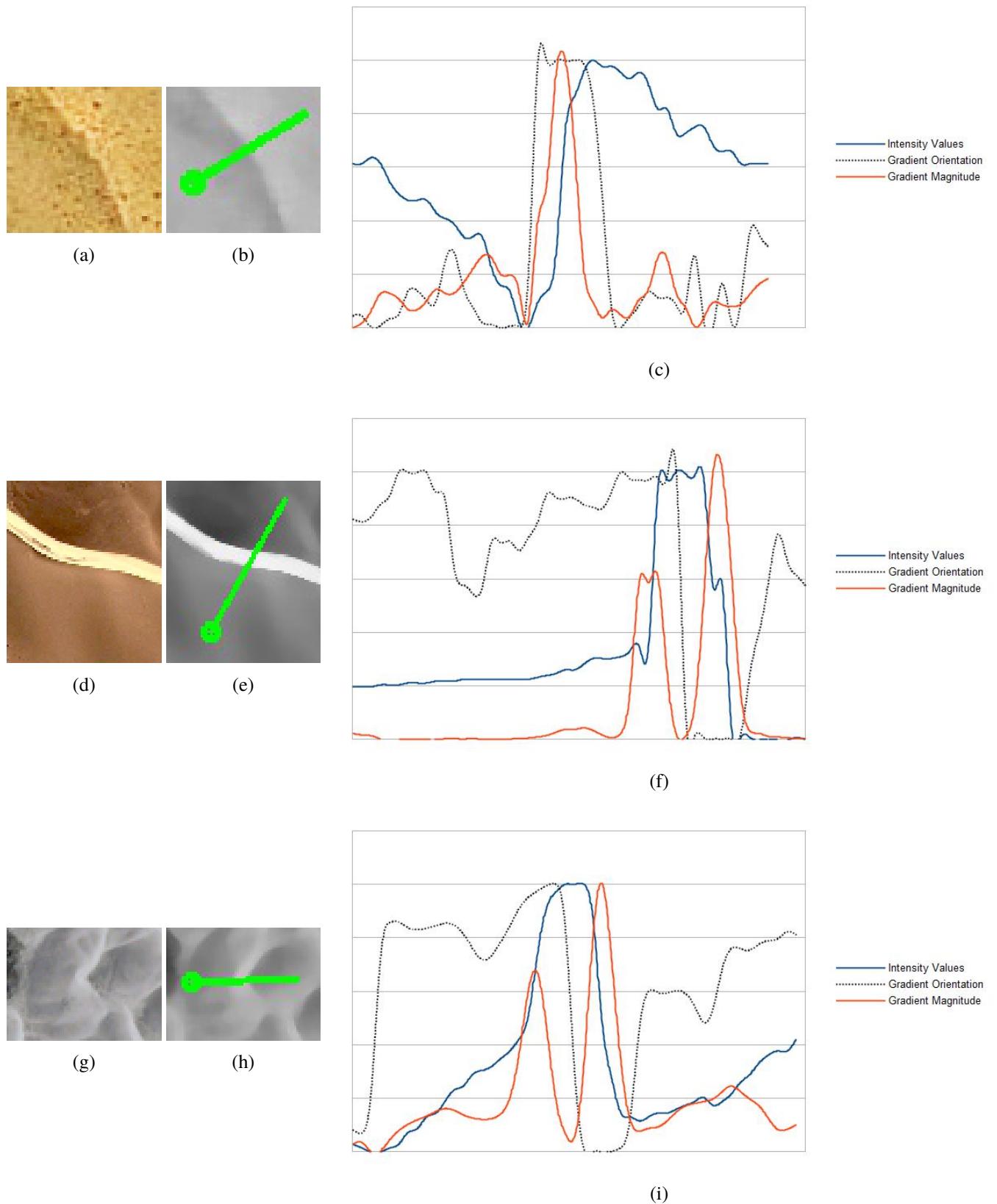


Figure 7: Illustration of the intensity, gradient magnitudes, and gradient orientations, plotted along a path crossing a dune crest-line: (a), (d),(g) a sample image patch from Kalahari, Skeleton Coast, and White Sands regions, (b), (e), (h) the direction of the pixel samples plotted in (c), (f), (i) containing the intensity, gradient magnitude and orientations values along the sampled direction.

Image Processing

The pre-processing stage is an important step to normalize and de-noise the input data. Image processing is typically a low level process which attempts to improve image quality, normalize illumination, or remove unnecessary noise from the image. This type of processing will in turn improve higher level feature extraction.

The first step is to insure that the images have roughly the same scale. To enforce scale normalization, the datasets scales were manually selected for appropriate crest-line detection application, and the image resolutions were resized to roughly 1000 pixel in width. The appropriate resolution is chosen based on a few factors:

- Source resolution: The selected resolution is limited by the original image's resolution. No extra information can be gained from increasing the source image's resolution.
- Processing time: Higher resolution images require much more processing time overall which could be a limitation in a certain application.
- Scale selection: If the application's domain is found in a higher scale space, a higher resolution may not necessarily be required.

In this application, the goal is to detect the major crest-line, which is a higher scale domain, which means a higher resolution image is not a necessity. However, the parameter selection in these processes are dependent on the resolution and scale of the images. In future work, the dominant scale of the crest-lines could be determined automatically to improve the robustness of the algorithms.

The next step in the process is to convert the images to gray scale. The images in the terrestrial, and Ganges datasets (described in section 4.1.1 and 4.1.2) are color images by default. Although color processing of these images has not extensively been tested as part of this study, it is arguable whether using color information would improve the detection results. By converting the images to gray scale, an added benefit of normalizing the images across many region is implicitly enforced. Finally, the conversion process turns the three channel image into a single channel image which improves overall processing performance.

The final preprocessing step is to improve the image quality with low-pass filters. Many of the algorithms in this method rely on gradient information. Therefore, to extract the meaningful gradients, removing the highest frequencies (which in typically are responsible for noise), while preserving the lower scale or more desirable gradients. Two main popular filters are utilized. The median filter, introduced in [3], is applied first to remove grain and salt and pepper noise while preserving important edges. In some cases, median filtering also helps remove some smaller scale objects such as bushes, trees, rocks, and other features which are indistinguishable from satellite images. Following this, the standard Gaussian filter is applied which removes the remaining high frequency noisy signals from the image.

An important aspect to note is the size of the filter masks, which is heavily dependent on the resolution of the input image. We found that in most cases, for images around 1000 pixels in width, a *sigma* value of approximately 1.5 and mask size of 7 by 7 was sufficient for this application. Once the image has been preprocessed, the gradients can be computed and saved for future use. Gradients can be computed by taking the first derivative of the image, using the popular Sobel operator.

An optional additional step is to apply histogram equalization to the image [87]. Poorly illuminated or low contrast images greatly benefit from histogram equalization, because it stretches the contrast of the image in a consistent way. This in turns also helps in the extraction of edges, and improves determination of the dominant gradient orientation.

For the most part, the image processing step is common to most approaches proposed in this section. Some approaches require more or less pre-processing, but typically the processing is applied to each image before any detection is attempted.

Computing the Dominant Orientation

The dominant orientation is defined as the orientation of the gradients of crest-lines for the entire dune field. Computing the dominant orientation is an essential process in order to determine which gradients correspond to the actual crest-lines. This determination greatly improves the ability to filter out false positive candidates. In datasets where the sun's azimuth is known, the dominant orientation can be directly calculated using the sun's orientation. However, in cases where this information is not provided, the dominant orientation must be computed automatically. Sections 3.4.1 and 3.4.2 offer two main approaches to computing the dominant orientation.

Detecting Crest-Line Candidates

The general crest-line candidate detection step extracts candidate segments which are most likely candidates for crest-lines. There are many potential methods to extract segments from the satellite images. The approaches explained in this

section include appearance-based methods, which use the pixel values to extract crest-lines (3.3 and 3.5), gradient-based methods, which compute the derivatives of the images to extract the gradients (3.4 and 3.7), frequency domain methods (3.6), and finally machine learning methods (3.8 and 3.9).

Each of these methods have slightly different processing flows but ultimately provide a way to extract candidate crest-lines. Once candidates have been extracted, some post-processing is often applied to reject some segments which do not fit the criteria of dune crest-lines. Rejected segments are typically filtered by length (too short), shape (too much curvature), weakness of gradients, or other factors. The remaining segments can also be smoothed in order to provide less noisy segments.

Dune Metrics Calculations

Computing the dune metrics is the process of extracting the various geomorphological properties of the dune fields from the detected crest-lines. The two main metrics computed in this research are the orientation and inter-dune spacing. The reason these are the main important features is because they actually provide a large amount of information for researchers on the behavior of the dune fields. Additionally, they are fairly trivial to compute once dune crest-line segments have been extracted. Within the framework of this research, they can serve also as a proof of concept and validation of each crest-line detection method.

In future work, other metrics, features, and properties could be computed, depending the specific research needs. The process of computing these metrics is discussed in detail in section 3.10.

Validation

The validation process evaluates the quality of each crest-line detector. In order to perform the validation, a dataset of ground truth is provided with each image. The main result metrics of the validation process are the precision and recall. Precision is the percentage of correctly identified crest-lines within the set of candidates. Recall is defined to be the percentage of correctly detected crest-lines from the ground truth.

In addition, the validation process compares the results of the dune metrics calculations from the detected crest-lines and the ground truth. The same process of computing the dune metrics is also computed on the ground truth, and compared side-by-side with the detected crest-lines. The process is described in detail in section 4.2.

The next section presents the first approach which is an appearance-based method.

3.3 Appearance-Based Approach

The appearance-based approach uses the pixel intensity values in the image to extract object regions. The process flow for this approach is shown in Figure 9. The goal of this approach is to extract the contours of bright regions in the images which are assumed to be the sunlit side of the dunes. The contours are then smoothed and filtered using the gradient magnitudes.

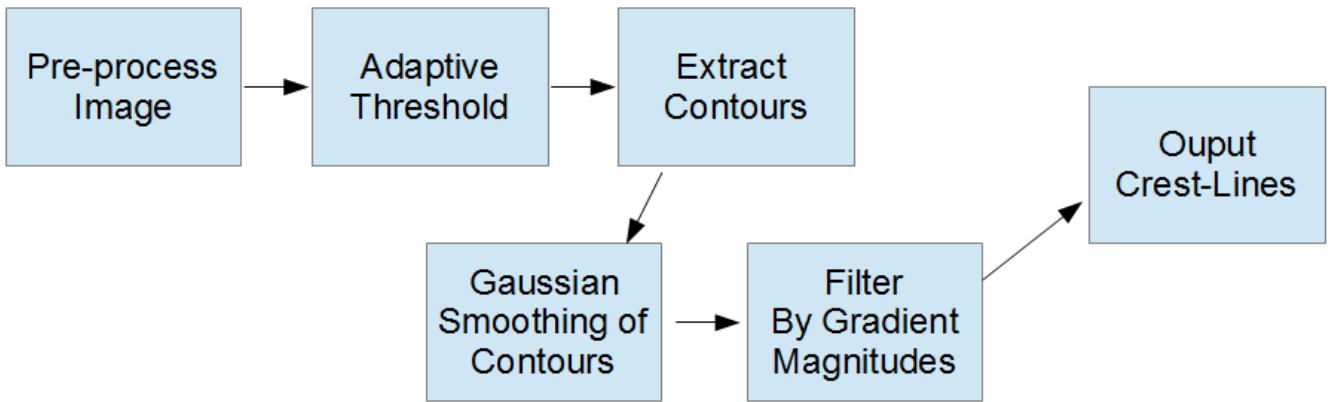


Figure 9: The process flow of the appearance-based approach. The images are first preprocessed, then an adaptive threshold is applied to binarize the image. Contours are extracted from the resulting binary regions. The contour segments are smoothed and filtered by gradient magnitude, resulting in the detected crest-lines.

As stated previously, dunes typically have a shaded and a sunlit side, and depending on the conditions and image quality, the actual edge or dune crest line may not be clearly defined. Image processing typically improves the quality

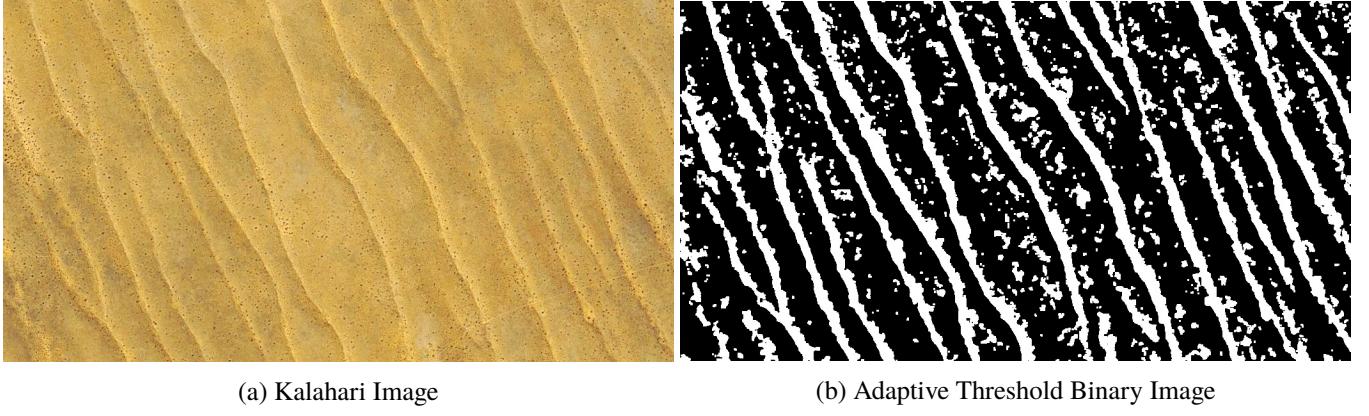


Figure 10: Adaptive Threshold applied to the Kalahari dune field satellite image. In (a) the Kalahari input image, (b) the adaptive threshold process applied to produce the binary image.

and allow the extraction of the dune crest lines. The preprocessing step uses a simple low-pass filter to remove noise and median filter to remove additional noise and insignificant features.

Under the assumption that the sunlit side of the dunes are the relatively brightest regions in the image, a thresholding method can be applied to extract the light side of a dune. The process of thresholding ([87]) is a common practice in image processing and computer vision. The process transforms the image into a binary (black and white) image for which regions can be extracted. The simplest thresholding method is to choose a fixed threshold value, and all pixels with intensity values greater than the threshold are set to 1 (white), and all others are set to 0 (black).

Using a fixed threshold may not be sufficient to extract the dunes since the illumination across an image is usually not uniform, and there may be poor contrast in intensity values. An adaptive threshold can be a useful tool to threshold an image with non-uniform illumination ([61, 82, 103]). An example of this type of thresholding is shown in Figure 10.

Once the image has been preprocessed, the bright regions are extracted from the dune field image using an adaptive thresholding process, resulting in a binary image. As shown in Figure 10, the bright regions shown consist of the sun illuminated side of the dunes along with some noise.

A region is defined as a contiguous group of pixels. From the regions, contours can be extracted, using a contour extraction algorithm described in [7]. Each contour can then be filtered by size of the region. Smaller regions are considered as noise, and can be rejected from the dune candidates. The remaining region contours are used as potential candidates for dune crest lines.

The contours extracted should contain both the crest-lines themselves, and the base or valley of each dune, and they should be on opposite sides of the contours. Because the adaptive thresholding process tends to be noisy, the resulting contours will also be noisy. The contours are therefore smoothed using a Gaussian kernel defined as:

$$G(i) = \frac{1}{\sqrt{2\pi}\sigma} e^{\frac{-i^2}{2\sigma^2}} \quad (1)$$

The Gaussian kernel is convolved on the contour, which a collection of two dimensional points (x, y) . The result of this convolution is a smoother contour which can be more reliably used as a dune crest line candidate. In order to filter out crest-line segments from non-crest-lines segments in the contours, we use the gradient magnitudes as a guide.

Typically, the assumption of this approach is that the dune crest lines are located on an area in the image where the derivatives are larger, or consistent along the edge of the dune. For each contour, the average magnitude μ of the derivative $\delta(i)$ of the image in the x and y directions is computed. Then, for each point within a contour compares its derivative with K neighbor's gradient magnitudes. Dune crest line candidate points will tend to have neighbors with similarly and consistently high gradients. To determine if a point $P(i)$ on a contour belongs on the dune crest line, we use the following criteria

$$P(i) = \begin{cases} 1 & \phi(i) \geq r \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where

$$\phi(i) = \frac{\sum_{k=0}^K \begin{cases} 1 & \delta(k) \geq \mu \\ 0 & \delta(k) < \mu \end{cases}}{K} \quad (3)$$

and $0 < r \leq 1$, which is the ratio of how many strong consistent edges to neighbors around a point i on the contour. Typically, r would be larger than 0.5, which means most of the neighbors of a point most have strong edges for a point to be considered a dune crest line candidate. The parameter r can be fine tune to allow more or less points to fit the criteria. This method will group common contour points based on gradient magnitude.

Once the dune crest line candidates on the contour have been extracted, the contour can be split into contiguous crest line segments. Some further processing can be applied to the segments, such as filtering out smaller or less significant segments (potential false positives). Overall, the approach does provide some valuable information, even though the results shown in Table 1 of section 4.3 are quite poor. The main problem with the approach is that there is too much variability in the input images causing the method to work well on some images, and quite poorly on others. The amount of parameters to tune for each specific dune region makes this method not a desirable approach for automated dune detection.

The next section presents an edge-based approach which relies on the gradients in the image to extract candidate crest-lines.

3.4 Edge-based Approach

Unlike the appearance-based method previously presented, the edge-based methods used gradient information exclusively to detect crest-lines. The approach relies strongly on the assumption that crest-lines typically have relatively large gradients. The overall process flow is shown in Figure 11, which uses Canny edges and dominant orientation computation to determine which gradients belong to crest-lines.

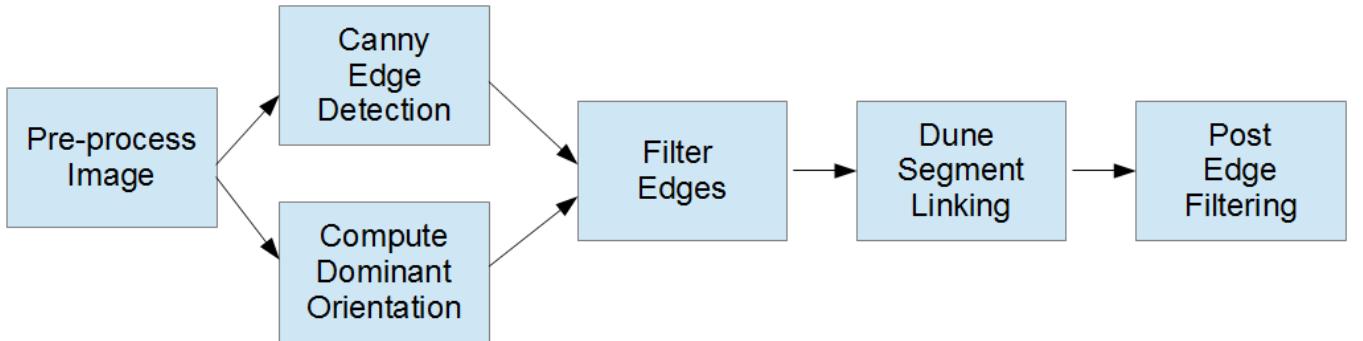


Figure 11: The process flow of the gradient-based approach. The images are first preprocessed, then the Canny edge detection algorithm is applied to the image. The dominant orientation computation (3.4), which enables many of the detected Canny edges to be filtered out based on orientation. The remaining edges are linked and post filtered based on length and shape.

As stated previously, there are a few challenges specifically associated with extracting edges from the dune satellite images:

1. Images tend to be noisy and highly textured, which results in many edges of varying orientations and magnitudes.
2. Many of the stronger edges may not necessarily be the crest-lines themselves. Generally, dunes cast relatively large shadows which can themselves produce strong edges.
3. Determining the appropriate values for the parameters of the edge detector can be difficult and greatly affect detection rates.

To address the concerns stated above, the popular Canny edge detector ([12]) is used along side some of the techniques inspired from [119] and [122]. First, in order to reduce the amount of noise in the images, both median and Gaussian filtering are applied to the image. The median filter removes small speckles (which for dune images may be small bushes or other contrasting features found in the images) while preserving major edge gradients. The Gaussian filter will then smooth out the gradient magnitudes and even out the image.

Once the image has been preprocessed, the Canny edge detector is used to recover the potential crest-line candidates. To overcome the parameter selection of the high and low threshold, the mean gradient μ and standard deviation σ are computed over the entire image. The high and low thresholds are determined as follows:

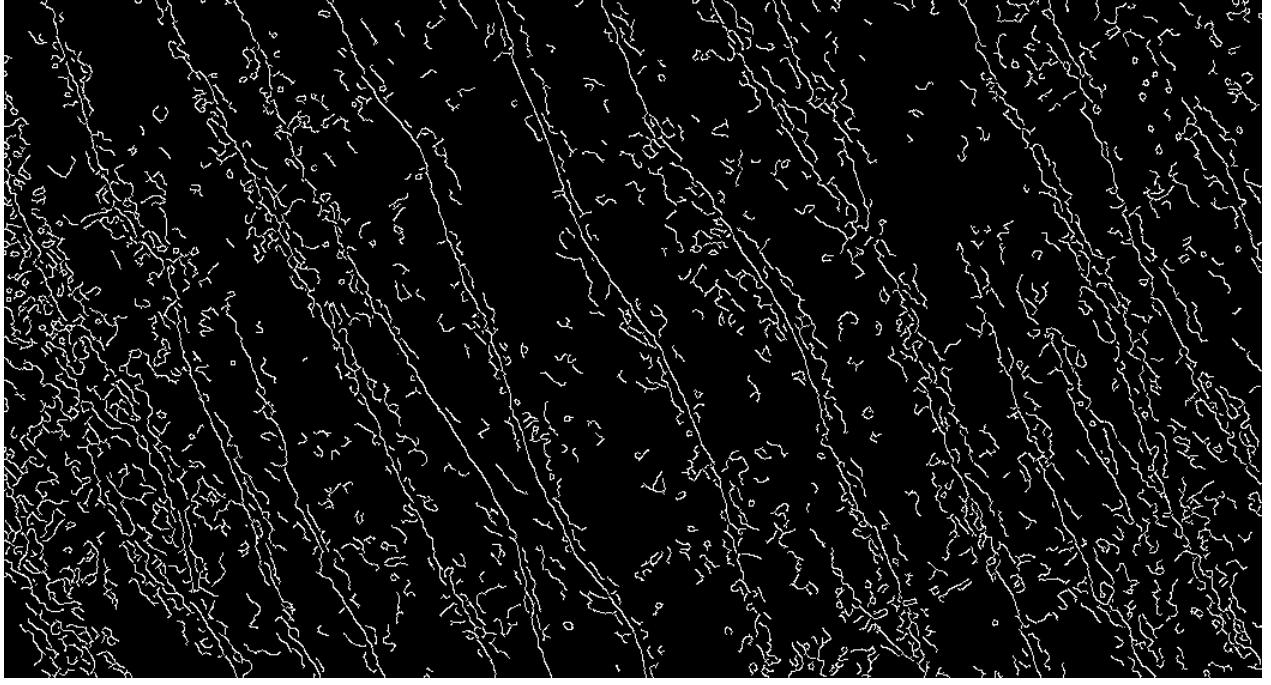


Figure 12: Canny edge detector after median and Gaussian filtering of the Kalahari dune image.

$$T_{high} = \mu + q \cdot \sigma \quad (4)$$

$$T_{low} = \frac{T_{high}}{2} \quad (5)$$

where q is a factor which can be determined based on the problem set. This makes the threshold parameter less sensitive to fixed value threshold problems. The result of the Canny edge detector after filtering are displayed in Figure 12. Clearly, the result of the Canny edge detector by itself is insufficient as many of the edges found are false positives. The next step in the process is to filter out the edge segments which do not belong to the dominant crest-lines.

Many of the edges provided in the edge detection step are noise, shadows, and non crest-line textures. However, the crest-lines all have one distinctive feature in common: all of their gradient have relatively similar orientations. Similarly, because of the sun's orientation relative to the dune field, edges generated from the shadows of the dunes will also have similar orientations within themselves. Given these properties, there should be a clear delineation between crest-line edges, and non-crest-line edges. The next task then clearly becomes determining the orientation of the gradients which belong to crest-lines. This is accomplished by computing the dominant orientation of the image.

Dominant Orientation Determination

Computing the dominant orientation of the dune field is an important step in the process. The dominant orientation is the overall gradient direction of the crest-lines to be detected. This process can help determine the average orientation of the dune field as well.

An important point to make is that this process could be optional if the dominant orientation is known *a priori*. Working with a dataset where the values are known improves the overall results of detection and enables skipping of this step. In cases where this information is unknown, or unavailable, automated dominant orientation determination is a potential solution.

Typically, there exists a gradient peak at the crest-line peak of the dunes (although this may not always be the case for less pronounced dunes). The direction of the gradient should agree with the overall dominant orientation of the dune field. Computing the dominant orientation of the field can be challenging in many cases because dunes typically have two major orientations, which are opposite directions. This phenomenon is clearly illustrated in Figure 15. The true crest-line peak, where the sunny side meets the shaded side is one gradient (Figure 15(b)). At the foot of the dune, where the shaded returns to light, a second gradient will be present, whom has an opposite direction (Figure 15(a)).

Two main methods have been implemented to determine the dominant orientation of the dune field.

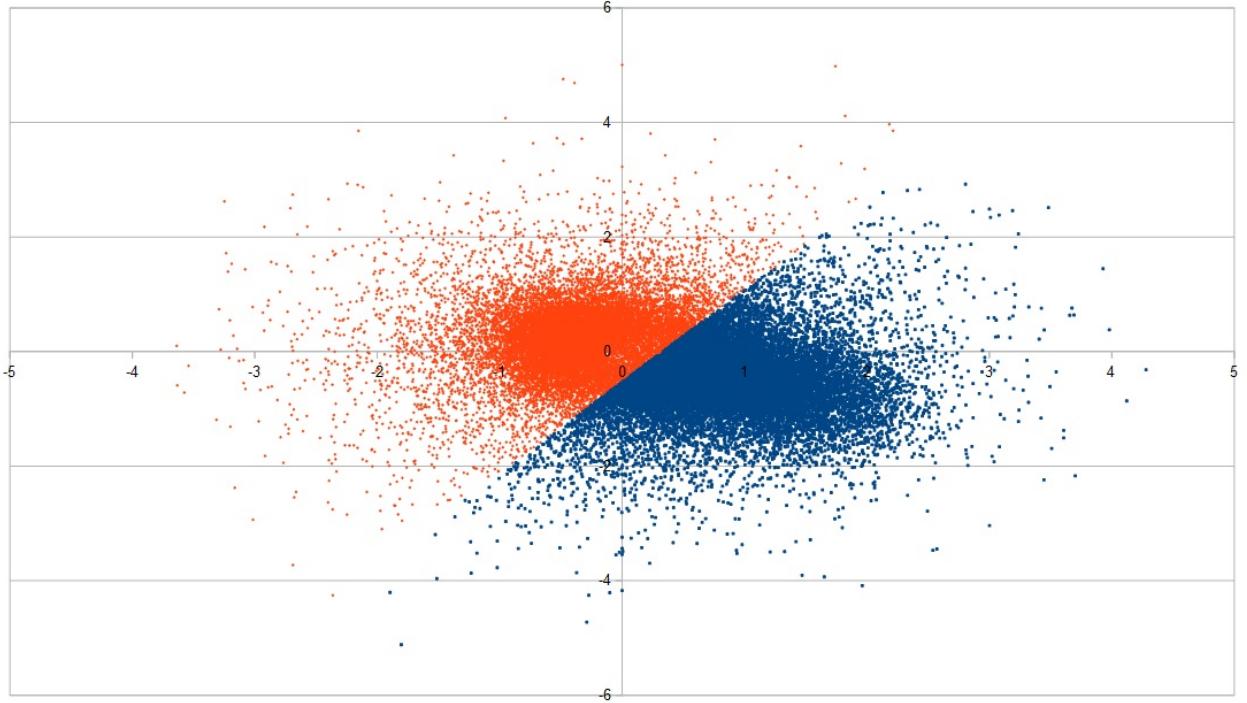


Figure 13: Results of K-Means clustering algorithm with $K = 2$ on the edge gradients detected using Canny, from Kalahari. The blue cluster represents the dune crest-line gradient vectors, and the red points represent all other non-crest-line edges. The cluster centers of blue is $(0.8958, -0.5496)$, and the cluster center of the red is $(-0.4111, 0.2705)$. The cluster centers are used to determine the dominant orientation.

3.4.1 Dominant Orientation From K-Means Clustering

Resolving the dominant orientation can be achieved by grouping similar edges together by clustering the gradients. The input data for clustering is the gradient vectors themselves. The gradient vectors are simply the first derivatives in the x and y directions: $\langle \delta_{x_i}, \delta_{y_i} \rangle$. Derivatives are typically computed by convolution of filters such as the Sobel operator ([107]).

There have been many approaches to clustering, but one of the simplest and most common is the K-Means algorithm ([28, 64, 69]). The primary benefit of using a clustering algorithm is that it is unsupervised, therefore the dominant orientation can be inferred from the gradient vectors themselves.

Typically, the main parameter that is required for K-Means is the value K , which is simply the number of clusters. In this problem set, the goal is to separate the crest-line group from all other non-crest-line edges. Therefore, clustering is computed using two clusters, by setting $K = 2$. There may be some value in trying different number of clusters, but this was not investigated as part of this research. Some of the benefits include better separability of gradients.

In order to get normalized results for the clustering algorithm, the gradients themselves are normalized. To normalize the gradients, the average gradient magnitude of the detected edges is computed as:

$$\bar{\delta}_\mu = \frac{\sum_{i=0}^P \sqrt{\delta_{x_i}^2 + \delta_{y_i}^2}}{P} \quad (6)$$

where P is the total number of detected edges from the Canny edge detector, δ_{x_i} and δ_{y_i} are the x and y gradient component of the i^{th} point. Each gradient is then normalized by dividing by the average magnitude $\bar{\delta}_\mu$. This process is commonly called *L2 normalization*.

$$\dot{\delta}(x_i, y_i) = \left(\frac{\delta_{x_i}}{\bar{\delta}_\mu}, \frac{\delta_{y_i}}{\bar{\delta}_\mu} \right) \quad (7)$$

The set of normalized gradient vectors are then clustered using the K-Means algorithm with $K = 2$. In Figure 13, the results of the clustering method are shown. The blue points are gradient vectors which potentially belong to the dune crest-line. Studying the points, we notice that there is a skew towards the stronger edges in the blue cluster, which we will call the dominant cluster. This is determined by computing the centroids of each cluster. The cluster with the larger overall gradient magnitude is therefore assumed to be the dominant orientation and the set which contains the crest-line points.

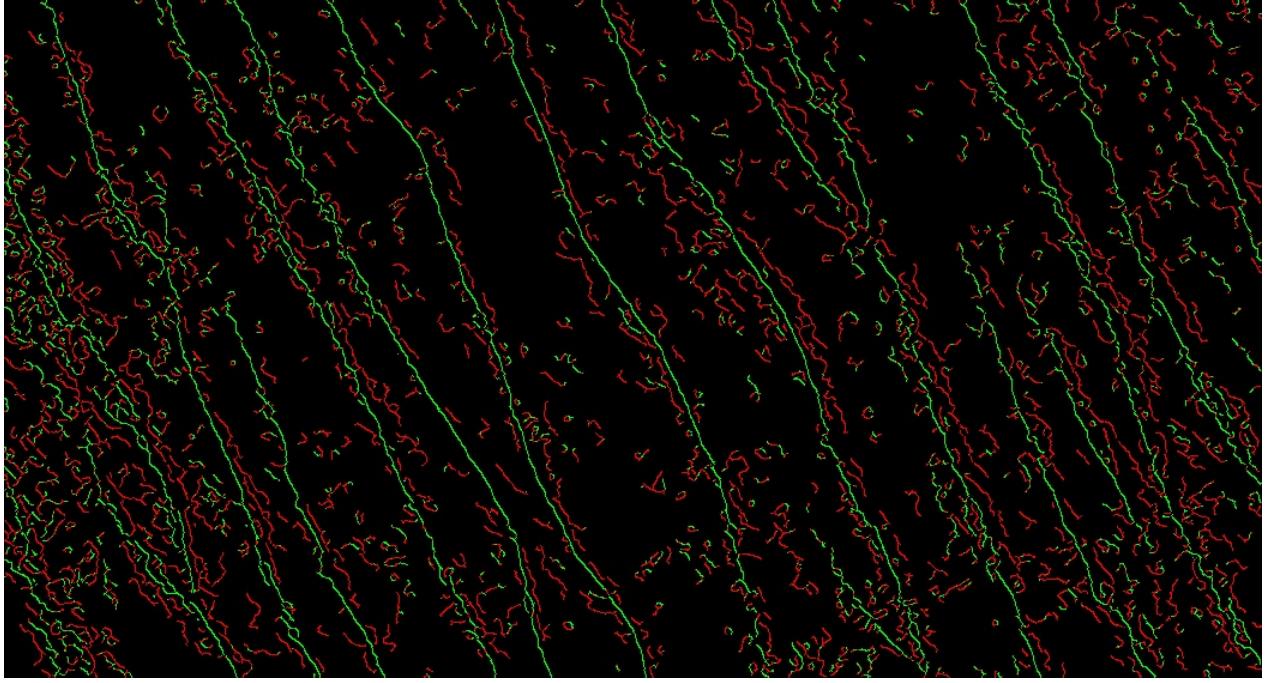


Figure 14: Results of filtering out edges which are not part of the dominant orientation (in red), while preserving edges which agree (in green) with the dominant orientation computed by the K-Means clustering method.

Once the clusters have been determined, all edges that agree (below to the dominant orientation cluster) are then assumed to be crest-line candidates. All other edges are filtered out. A sample of the filtering method is shown in Figure 14.

Once the weak edges have been filtered out, the remaining edges are used as crest-line candidates. The K-Means cluster removes a majority of false positives, but does preserve some small edge segments which are most likely noise. By using a connected components technique, these smaller less significant edge segments can be removed, keeping only longer contiguous segments.

3.4.2 Dominant Orientation From Histogram of Gradients

An alternative to the clustering method is to compute a histogram of the gradients based on orientation. Histograms of gradient directions can be used to determine the true orientation. Histograms have been used for many processes and applications including [67, 77, 95]. Quantizing the space allows us to split edges according to their orientation, where each bin in the histogram represents an edge orientation. Each bin then covers an angle of $\frac{2\pi}{N}$ radians, where N is the number of bins. Each bin of the histogram is weighed by the magnitude of edges. Once the histogram is constructed, peaks in the magnitude will help determine the dominant orientation (Φ) of the dune crest-lines. The approach poses three main problems:

1. Determining N : In practice, a larger value for N has shown to provide finer grain resolution which improves dominant orientation determination. In Figure 15, a value of $N = 18$ is used (resulting in bins of 20° width).
2. Multiple Peaks: Often, images may have multiple peaks in the histogram, but only one of the peaks should represent the true dominant orientation of the crest-lines. As noted in [76], the histograms typically have a strong bimodal distribution, which is very apparent in Figure 15(c).
3. Global Solution: The computed dominant orientation is global to the image, which may not be optimal in cases of complex dune structures. A better solution would be to compute the dominant orientation in a local area of the image, to account for local shifts in orientation.

One assumption made is that the bin with the largest value represents the orientation of the crest-line edges. Although this assumption holds for many cases, it is not always the case. The Skeleton Coast test image provides a good example of this case in Figure 15. In this particular case, there are two major peaks in the histogram, where the stronger is not the part of the crest-line edge group, and the weaker one is. Choosing the higher peak will cause invalid crest-lines to be chosen. In order to determine which peak best represents the crest-line edge group, some normalization can be applied. The normalization process begins by computing the mean vector of the gradients from the edge image as follows:

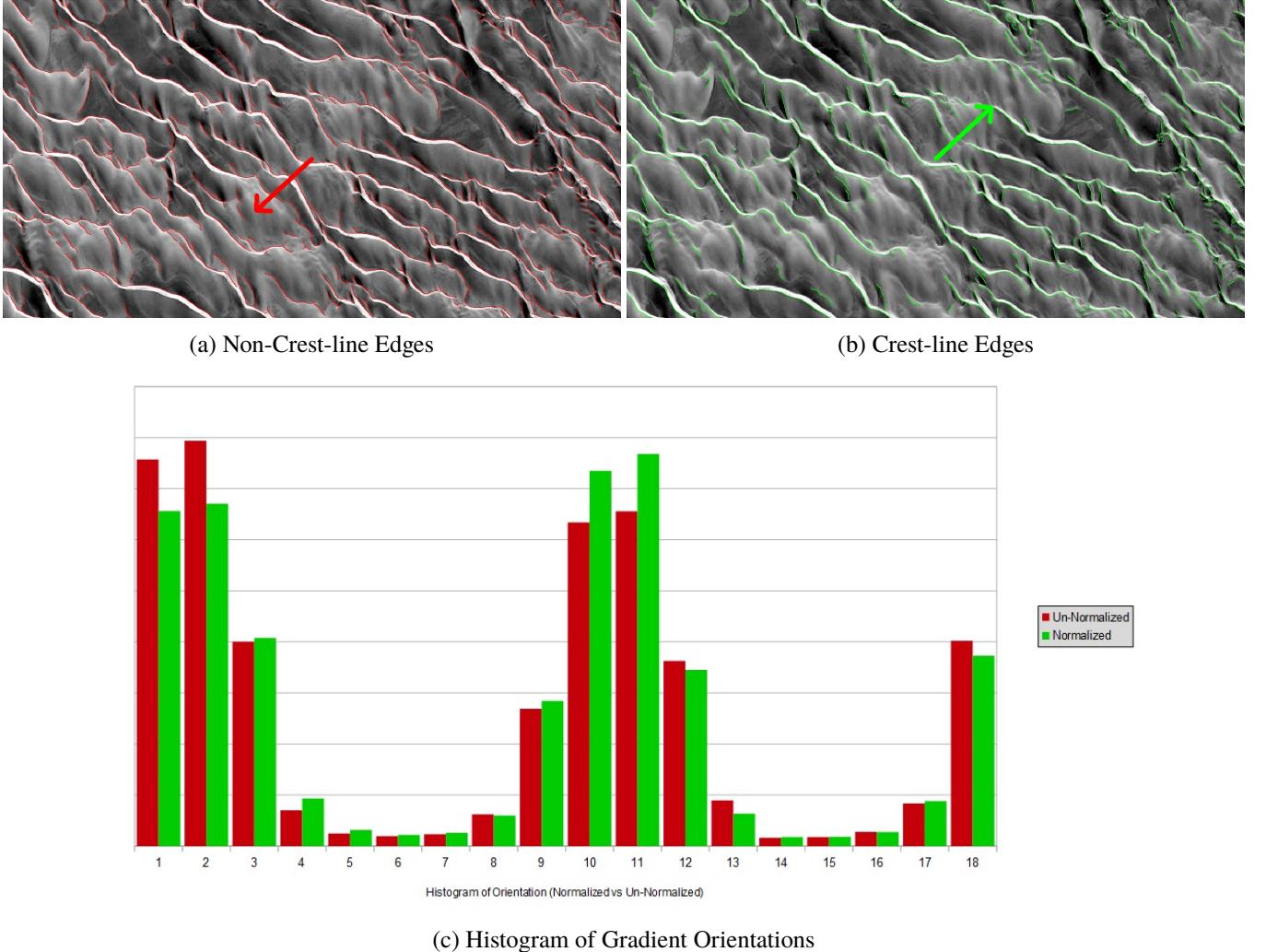


Figure 15: Results of unnormalized (red) and normalized (green) histograms of gradients orientations ($N = 18$), where each represents a 20° slice. The histogram contains two major peaks, but the crest-line is the weaker of the two peaks. (a) Edges which belong to the dominant orientation which are invalid crest-lines. (b) With normalization, the true crest-lines now have the higher overall magnitude. (c) The 18 bin histogram of gradients with and without normalization.

$$\hat{\mu} \langle \bar{x}, \bar{y} \rangle = \left\langle \frac{\sum_{i=0}^P \delta_{x_i}}{P}, \frac{\sum_{i=0}^P \delta_{y_i}}{P} \right\rangle \quad (8)$$

where P is the total number of detected edges from the Canny edge detector, δ_{x_i} and δ_{y_i} are the x and y gradient component of the i^{th} point. The mean orientation is computed from the mean vector as:

$$\bar{\theta}_\mu = \arctan \left(\frac{\mu_y}{\mu_x} \right) \quad (9)$$

The gradients are then normalized by simply subtracting the mean vector from each gradient:

$$\dot{\delta}(x_i, y_i) = (\delta_{x_i} - \mu_x, \delta_{y_i} - \mu_y) \quad (10)$$

The normalized orientation $\dot{\theta}_i$ can be computed:

$$\dot{\theta}_i = \arctan \left(\frac{\dot{\delta}_{y_i}}{\dot{\delta}_{x_i}} \right) \quad (11)$$

To determine which bin the i^{th} edge point belongs to, we simply calculate $\left\lceil \frac{\dot{\theta}_i \cdot N}{2\pi} - 0.5 \right\rceil$, and increment that bin by the magnitude of the normalized gradient by $\sqrt{\dot{\delta}_{x_i}^2 + \dot{\delta}_{y_i}^2}$. In essence, this normalization process removes the uneven skew of the gradients in the overall image. Removing this skew allows true crest-line edges to be fairly compared with other stronger edges. As shown in Figure 15, the normalization process softens the stronger dominant edge and enhances the

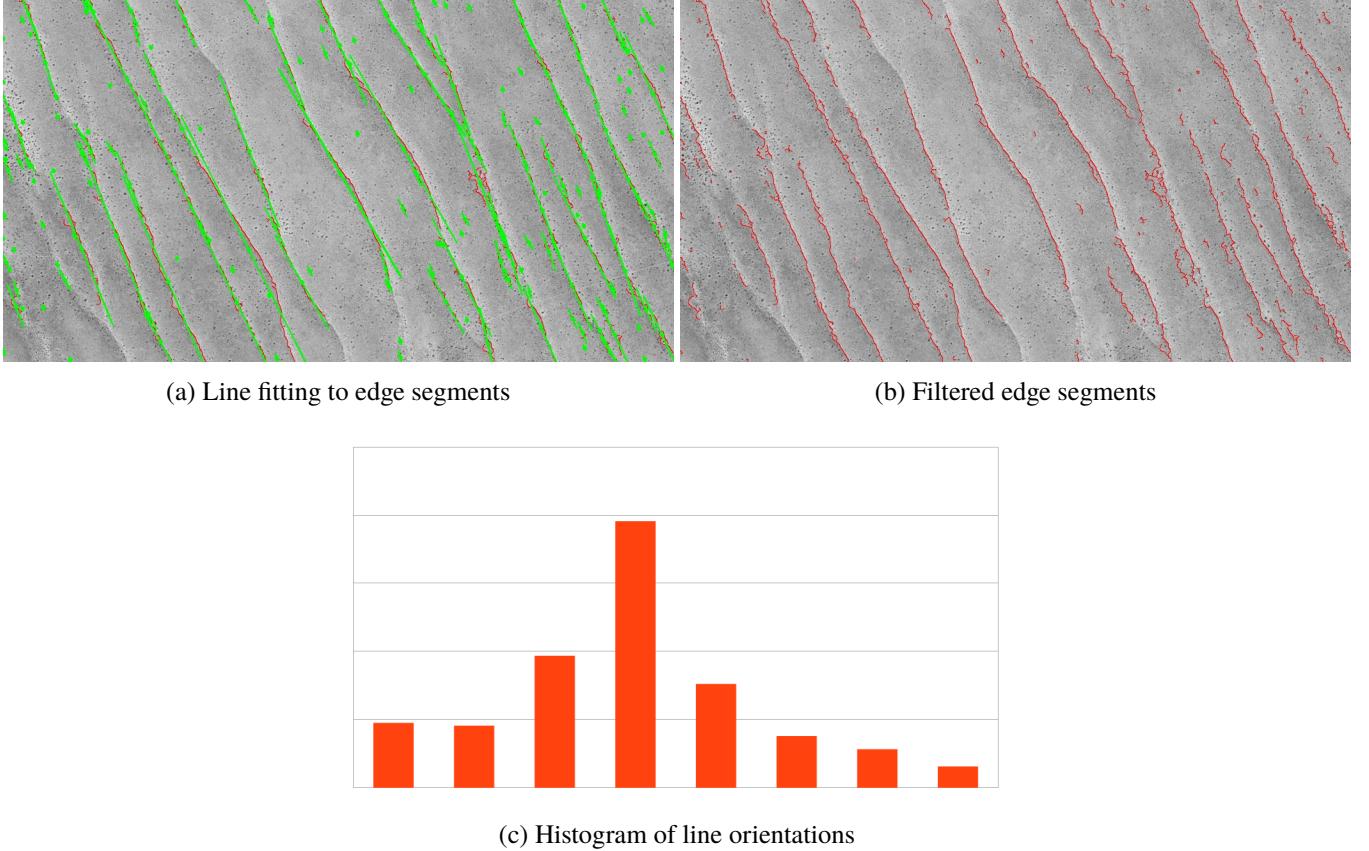


Figure 16: Results of filtering based on the line orientation fitting, and choosing the peak in the histogram. In (a) the fitted lines from the dune segments, (b) segments which have orientation matching the peak in the histogram, (c) the histogram of line orientations weighed by segment length.

true crest-line edges. This process enables true crest-lines to be accurately determined in images where the valleys of dunes are sharp and contain strong edges.

Once the histogram is computed and normalized, the dominant orientation vector $\hat{\theta}_{dom}$ is determined by averaging the gradients belonging to the histogram bin with the highest value. With the dominant orientation knowledge acquired, candidate crest-line dunes can be detected.

3.4.3 Dune Segment Linking

With the dominant orientation computed, many false positives have been rejected. The next step of the process is to group segments which link together. Since each crest-line candidate segments is a collections of two dimensional points, an intuitive step is to fit a line to the candidate segments. The line fitting algorithm used is from the OpenCV library [7], which aims to minimize:

$$\sum_i \rho(r_i) \quad (12)$$

where r_i is the distance between a point on the line, and $\rho(r_i)$ is the function used. We use the simplest least square method, $\rho(r) = \frac{r^2}{2}$. The result of the line fitting produces a vector of four values $\langle x, y, v_x, v_y \rangle$, where the (x, y) components are the position of the line, and (v_x, v_y) is the normalized vector which is collinear to the line, encoding the orientation of the line.

Once all segments are fit with a line, we can filter the segments based on the orientations of the lines. To get a general view of the distribution of the orientation of the lines, a histogram of the line orientations can help determine dominant lines. The histogram can be constructed based on the orientation of the lines, weighed by segment length. As shown in figure 16(c), the histogram appears to have a unimodal distribution. When choosing all lines which belong to the peak bin in the histogram, the resulting in the output image 16(a) The results show that many of the false positives have been filtered out.

Since the shown distribution appears to be unimodal, a better approach may be to simply compute the mean and standard deviation of the line orientations. Choosing the appropriate sigma value to filter out the noise and false positives

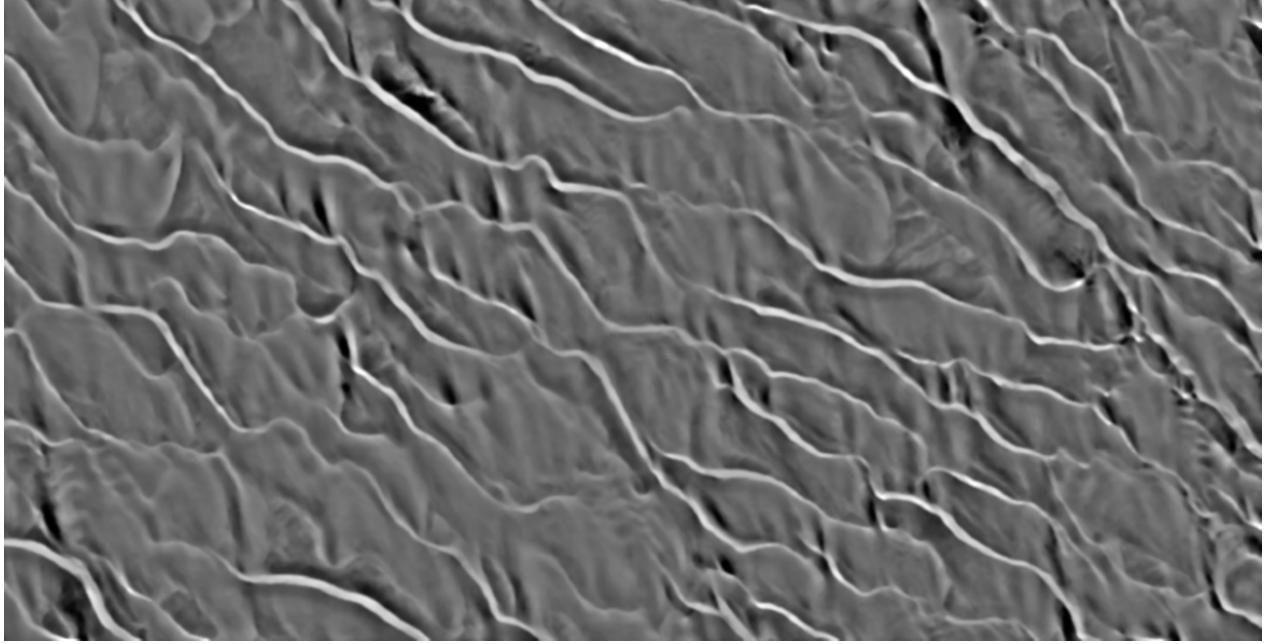


Figure 18: Simple illumination normalization algorithm applied to the Skeleton Coast image.

depends on the distribution. In some cases, such as images with crest-lines which are more curved, the variation orientations may be more significant. In order to avoid the filtering of true positives, a linking algorithm should be used to link segments which belong to the same crest-line.

The next section presents another appearance-based approach which utilizes the Watershed Transform to detect dune crest-lines.

3.5 Watershed Transform Segmentation Approach

Similarly to 3.3, the goal of this approach is to segment out the brighter region of the image, which represents the sunny side of the dune, using the watershed transform. The overall process of this approach is shown in Figure 17.

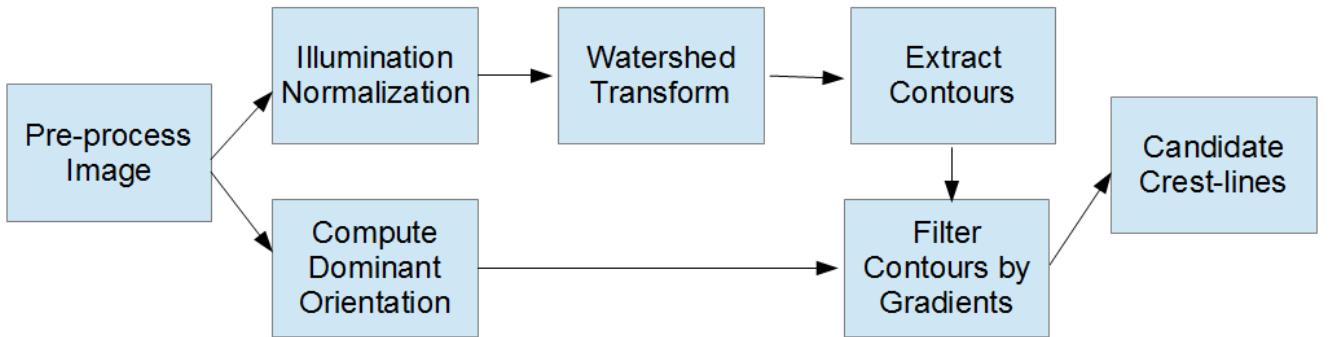


Figure 17: The process flow of the Watershed Transform detection approach. The images are first preprocessed, then some illumination normalization is applied to the image in order to improve the results of the Watershed transform. Contours are then extracted from the resulting regions which are filtered using the gradients, based on the dominant orientation.

The watershed transform ([5, 6, 57]) is a powerful segmentation tool, which in this approach is used to determine the boundary of the bright dune region. Typically, this approach can produce better results than a simple thresholding method. In order to extract reliable regions, image illumination normalization is applied, as shown in Figure 18, to make the illumination more evenly distributed across an image. After applying some preprocessing to the image (again, median and Gaussian blurring to remove as much noise as possible), the illumination normalization presented is very basic, and better normalization schemes should be tried.

The normalization process is done in a fairly simple and efficient way, using integral images. The integral image concept was originally proposed in [15], and was popularized for computer vision applications by [117]. The integral

image representation allows rectangular features to be computed very efficiently. Each pixel (x, y) of the integral image is the sum of the pixels above and to the left of it, as described in [117]:

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \quad (13)$$

To normalize using integral images, the mean μ_i of the image, which can be done efficiently once the integral images is computed, given an $N \times M$ image.

$$\mu_i = \frac{ii(M, N)}{M \cdot N} \quad (14)$$

The goal of the illumination normalization is to bring pixel intensities value closer to the mean intensity of the image for a more even distribution of intensities. To accomplish this, we simply compute a ratio multiplier of the mean intensity μ_i over the square local area ($w \times w$) mean intensity centered around a pixel (x, y) . The mean of a local area can be computed efficiently using the integral image theory:

$$\mu_l(x, y) = D - C - B + A \quad (15)$$

$$A = ii\left(x - \frac{w}{2}, y - \frac{w}{2}\right) \quad (16)$$

$$B = ii\left(x + \frac{w}{2}, y - \frac{w}{2}\right) \quad (17)$$

$$C = ii\left(x - \frac{w}{2}, y + \frac{w}{2}\right) \quad (18)$$

$$D = ii\left(x + \frac{w}{2}, y + \frac{w}{2}\right) \quad (19)$$

Then, the ratio of image intensity to local mean intensity simply is $r_{\mu_l(x, y)} = \frac{\mu_i}{\mu_l(x, y)}$. The pixel intensity value at that location is then multiplied by this ratio to get the normalized value:

$$\tilde{i}(x, y) = r_{\mu_l(x, y)} \cdot i(x, y) \quad (20)$$

With the image illumination normalized, the watershed transform can be used to segment out the brighter regions of the image which will be the dune sunny sides. The watershed implementation used is provided by the OpenCV library, and is a variant of the implementation proposed in [72]. The watershed uses an initial seed rough image of the desired locations to segment out, which in this application is the brighter regions of the image. The desired regions to extract are the bright and shaded areas. The threshold values for these can be determined from the image's mean and standard deviation intensity values:

$$T_{high} = \mu_i + \rho_{high} \cdot \sigma_i \quad (21)$$

$$T_{low} = \mu_i + \rho_{low} \cdot \sigma_i \quad (22)$$

Where the ρ_{high} and ρ_{low} are scalar values determined empirically. All pixels above the high threshold are labeled as the desired region to segment (in white) as shown in Figure 19(b), and all pixels below the lower threshold are set to the non-desired region (in gray), shown in 19(a). A morphological eroding operation is applied to better separate the regions for the watershed transform. Using the labeled image and normalized illumination image, the watershed algorithm is able to segment out the dune regions reliably, as shown in 19(c).

After computing the watershed segmentation image, the regions extracted from it can be retrieved using contour detection. The contours retrieved are closed loops where one segment will be along the dune crest line and the other not. To determine which segments of the contour belong to the dune crest-line, we can use what was learned from section 3.4.2: Building a histogram of the image gradients, choosing the dominant orientation, and filtering out all contour points which do not agree with the dominant orientation.

Results can be seen in Figure 19. Although the results appear promising, the Watershed transform method has a few drawbacks:

- An assumption that a dune contains a bright side. Not all images (especially the Simpson image), have crest-lines that have a bright edge.

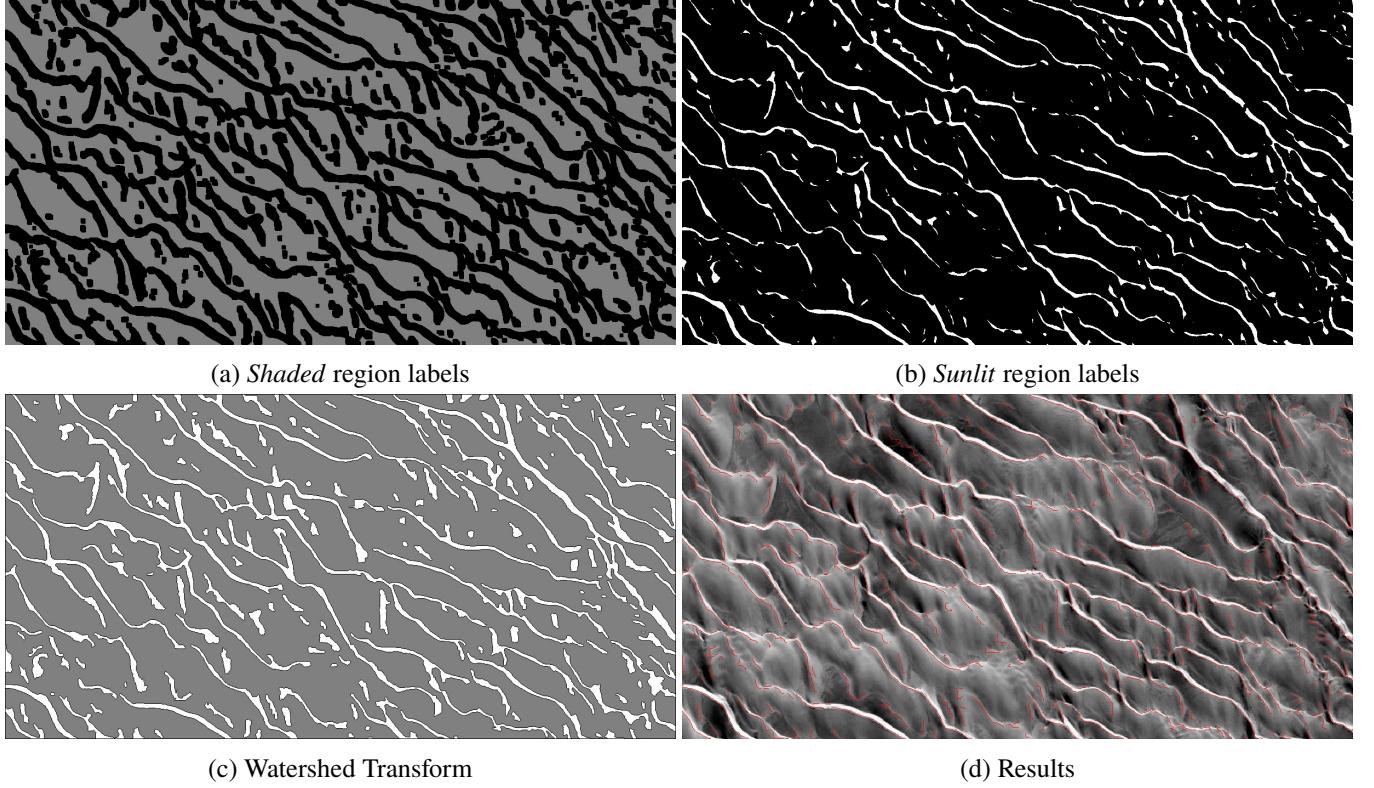


Figure 19: Watershed Transform results example applied to the Skeleton Coast. In (a), the pixels labeled as *shaded*, (b) the pixels labeled as *sunlit*, (c) the output of the Watershed Transform, and (d) crest-line detection results from the Watershed Transform.

- The edges of the contours appear to be slightly noisier than edge-based approaches.
- Localization of the segments is poor. Some segments do not lie on an actual edge, which may be reported as a false positive.

On the other hand, the watershed approach does appears to have some key benefits:

- Better connectivity of dune crest-line segments. When edges become weak between two connecting segments, the edge based approaches produce gaps, where the watershed is able to fill the gaps.
- Watershed method could be further expanded to include color image processing.

The next section explores another approach which utilizes the frequency domain to extract crest-lines.

3.6 Frequency Domain Approach

In this approach, we take advantage of the fact dune-field patterns seem to have some periodic pattern, which could possibly be extracted from the frequency domain. Converting the images into the frequency domain using the Discrete Fourier Transform (DFT) ([1, 2]) provides a method for applying some processing on individual frequencies of the image. The process flow for this approach is shown in Figure 20.

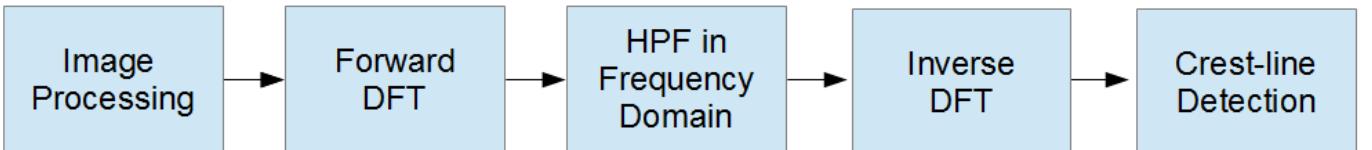


Figure 20: The process flow of the frequency domain approach. The images are first preprocessed, then translated into the frequency domain using the standard 2D discrete Fourier Transform. A high-pass filter is applied to the frequency domain to extract highest frequencies (which are assumed to contain dune crest-lines), and the inverse DFT is applied to return to the spacial domain. Crest-line detection can then be applied to the resulting image.

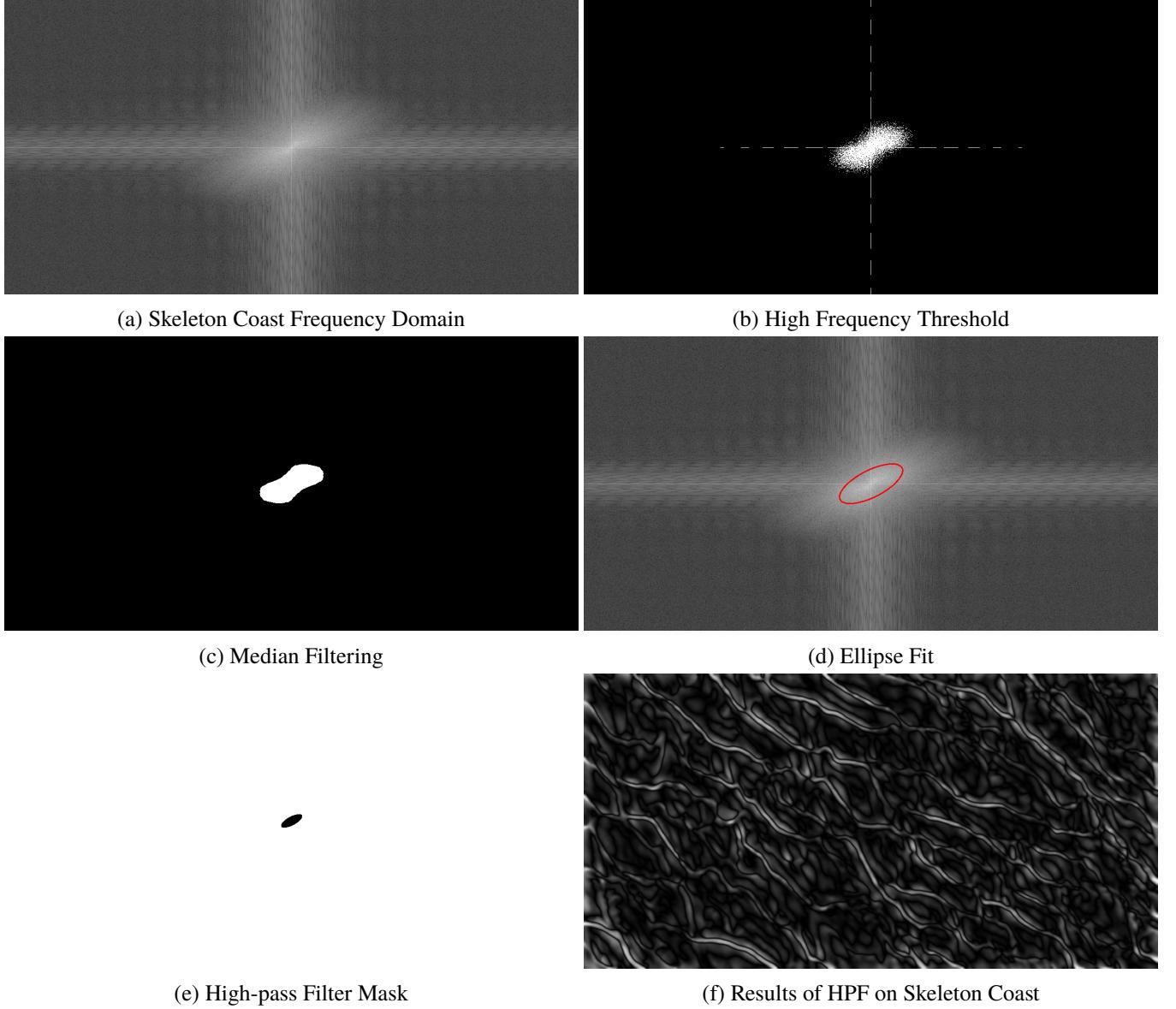


Figure 21: Frequency spectrum processing results from the Skeleton-Coast image. (a) Log-normalized shifted frequency domain, (b) Result of thresholding frequency domain magnitude based on average magnitude and standard deviation of the magnitudes, (c) Median filtering of the binary image of the frequency domain, (d) Result of ellipse fitting of the dominant frequencies region, (e) Mask used for high-pass filtering of the frequency domain, (f) Result inverse DFT of the high pass filter applied on the frequency domain.

In this application, the features we desire to extract are the dune crest-lines, which in the spacial domain are edges. In the frequency domain, the crest-lines may belong to some of the higher frequencies (or possibly a subset of the frequency spectrum). The shifted frequency domain output of the DFT is shown in Figure 21.

From the frequency domain, dominant frequencies can be extracted by searching for large magnitudes of frequencies. Determining which frequencies to extract will be key for this application. More research needs to be done on how to extract features from the frequency domain. Typically, from the dataset used, it can be shown that a dominant orientation in the frequency appears to be present in the dune-field images. From preliminary observations, the dominant orientation seems to correlate exactly with the orientation of the dune crests.

In order to determine this dominant orientation, the frequency domain magnitude image is thresholded based on the mean and standard deviation of the frequency magnitudes, as shown in Figure 21(b). Because output of this threshold is somewhat noisy, a median filter is applied to create a smooth region of for the dominant frequency (21(c)). The contour of the region is then extracted, which is used to fit an ellipse (21(d)).

The computed ellipse's orientation appears to match the orientation of the dune gradients. The information provided by the ellipse may prove useful. Fitting an ellipse is currently mainly used for determining the orientation of the gradients of the dune crests.

There may be a better approach to extract information from the frequency domain, or other features. Currently, we use this ellipse to generate the mask for high-pass filtering. The mask shown in Figure 21(e)) simply rejects all lower

frequencies (close to the center of the spectrum), and allows all higher frequencies to pass through.

Upon applying the mask to the frequency domain, and taking the inverse discrete Fourier Transform to return to the spacial domain, the result produces a kind of edge detection filter, where lower values could potentially be used to determine locate dune crests. The results can be seen in 21(f).

Unfortunately, no further research was done on using the frequency domain. There may be some benefit in exploring this approach further in the future. The next approach presents are gradient orientation-based approach.

3.7 Gradient Orientation-Based Approach

In this approach, we used knowledge learned from a previous gradient based approach, but focus on the orientation component of the gradients rather than the magnitude. The overall flow of the approach is shown in Figure 22.

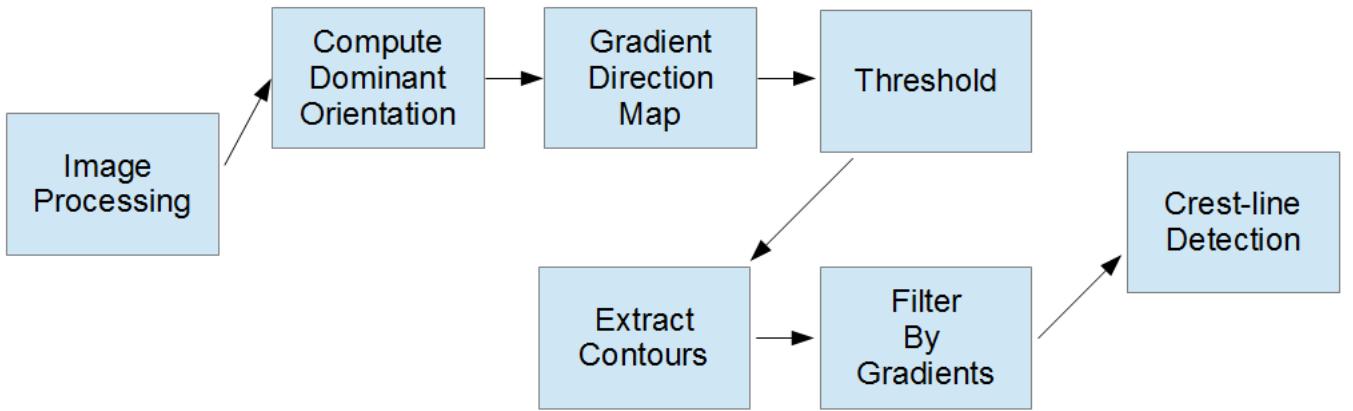


Figure 22: The process flow of the gradient orientation-based approach. The image is first preprocessed for optimal gradient computation, and the dominant orientation is computed. The gradient direction map is then computed and thresholded to preserve all gradients which *agree* with the dominant orientation. The resulting binary image is a good candidate for contour extraction, which are subsequently filtered based on gradient orientation and magnitude. The resulting output is the detected crest-lines segments.

In section 3.4, we learned that the dominant orientation of the dune field could be determined based on the derivatives of the image. In that approach, we used the gradient magnitudes to find major crest-line edges, and filtered out edges which do not *agree* with the dominant orientation of the gradients. Using edges provides good localization but is prone to discontinuity of edge segments.

In contrast, this approach uses the gradient orientations of each pixel to find crest-lines. As stated previously, dunes have a bright and shaded side. Gradients in the bright and shaded sides of the dune have a tendency to point towards the crest-line.

The dominant orientation explained in section 3.4 is an important feature to consider in this approach. An aspect of the dominant orientation that was not discussed previously is the fact that by definition, the dominant orientation will always point from the shaded area of the dune towards the sunlit area. This phenomenon is due to the gradient operator. Subtracting a dark pixel (lower value) from a bright pixel (larger value), results in a positive value, which in terms means that the gradients of darker pixels always point towards brighter pixels.

Given this fact, a case can be made for comparing the gradients of each pixel to the dominant orientation vector. All gradient vectors which *agree* with the dominant orientation are then considered to be the shaded sides of a dune, and all gradients which *disagree* with the dominant orientation are therefore part of the sunlit side of a dune. Finally the transition between areas that *agree* and those that *disagree* must be a good potential crest-line candidate.

The concept of *agreement* can be defined mathematically using a simple dot product. Given an dominant orientation defined as a unit length vector $\hat{\delta}_\mu$, and the local unit length gradient vector at (i, j) defined as

$$\hat{\delta}_{ij} = \left\langle \frac{\delta_{x_{ij}}}{\sqrt{\delta_{x_{ij}}^2 + \delta_{y_{ij}}^2}}, \frac{\delta_{y_{ij}}}{\sqrt{\delta_{x_{ij}}^2 + \delta_{y_{ij}}^2}} \right\rangle \quad (23)$$

The *agreement* measurement D_{ij} then is simply the dot product:

$$D_{ij} = \hat{\delta}_{ij} \bullet \hat{\delta}_\mu \quad (24)$$

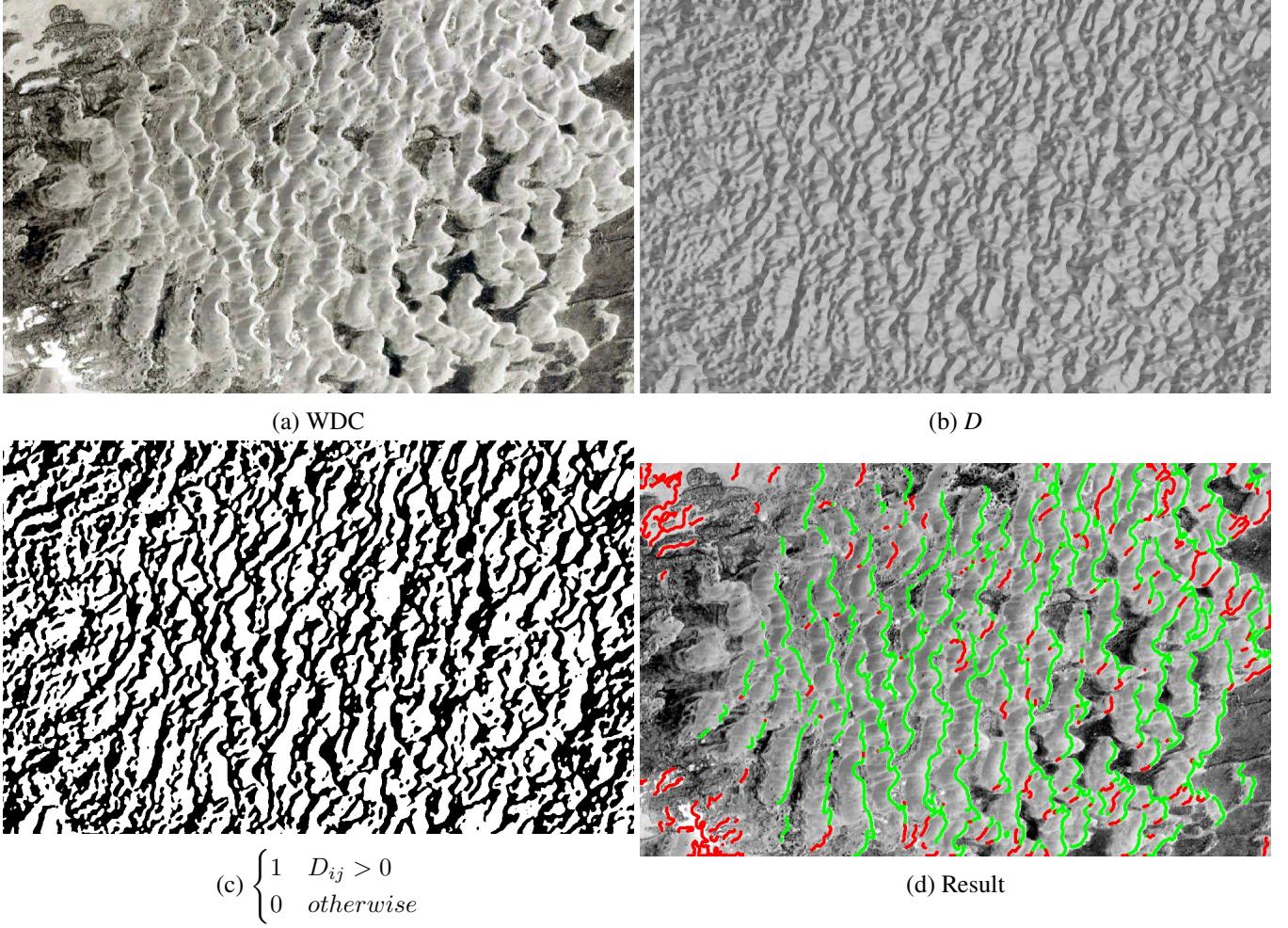


Figure 23: The gradient orientation-based approach illustrated: (a) The input image WDC, (b) The gradient direction map D computed by taking the dot product of the gradients at each pixel with the dominant orientation, (c) thresholding the gradient direction map for all dot products greater than 0, (d) the results of crest-line detection using the gradient-orientation based method, where green lines are true positives, and red lines are false positives.

Since $\hat{\delta}_{ij}$ and $\hat{\delta}_\mu$ are both unit-length vectors, the range possible values for D_{ij} is $[-1, 1]$, where positive values represent gradients which *agree* with the dominant orientation, 0 is perpendicular to the dominant orientation, and negative values *disagree* with the dominant orientation. The result of the dot product computed at each pixel is shown in Figure 23(b), which we call the gradient direction map.

The regions produced by the gradient direction map have been shown to be typically smooth and have good continuity in this application. There appears to be a clear separation between regions in *agreement* and other regions. The separation becomes even clearer when the gradient direction map is thresholded based on the dot product values. In Figure 23(c), a binary image of the gradient direction map is computed by thresholding all dot products above 0.

Once the gradient direction map has been thresholded, the binary image provides well defined regions which allows the use of contours to determine the crest-lines. Contours are extracted in a similar manner as presented in sections 3.3 and 3.4. The contours extracted provide good continuity of the crest-lines, but also contain many elements which are not part of the crest-line. To reject segments of the contour which are not part of the crest-lines, two techniques have been tried.

Intensity Based:

Region contours contain crest-lines segments and non-crest-line segments. In this approach, the algorithm uses intensity values of the pixels along the contours. Segments of the contour which are on brighter regions of the image are considered to be part of the crest-lines. The use of intensity values to determine which segments of a contour are on the actual crest-line may not be an optimal solution, but has provided adequate means of rejecting many false positives.

The main benefit of this approach is that the resulting regions have sharp and defined edges along the dune crest-line, regardless of how sharp an edge is in actuality. In many cases, dunes in the same image may have some sharp edges along the crest-line while others have less defined crest-lines. Using gradient magnitude in areas with poorly defined crest-lines yields poor results, while the gradient direction maps provides a sharper edge and a better estimate of where a crest-line

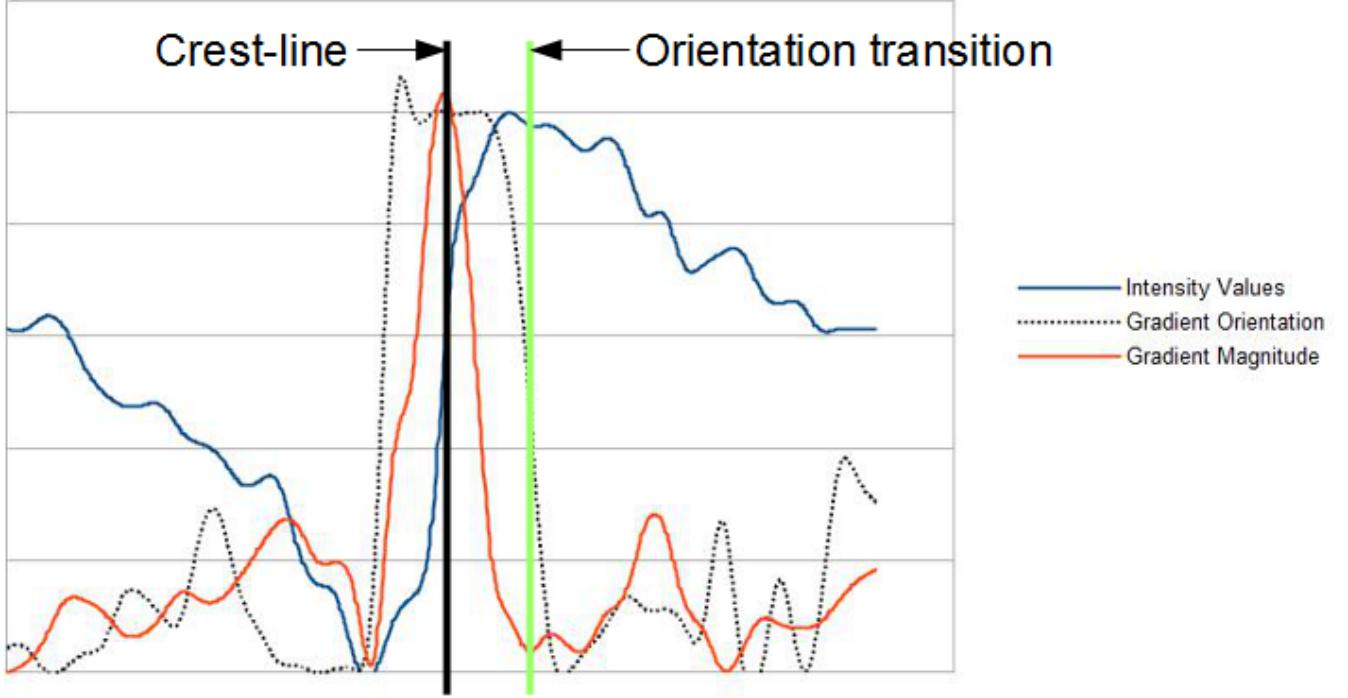


Figure 24: The cause of the shift in localization when computing the gradient direction map. The transition from gradients which *agree* ($D_{ij} > 0$) with the dominant orientation almost always happens after the true crest-line due to the size of the kernel K .

may be. The downside of this method is that not using the gradient magnitude results in poor localization of the actual crest-line.

Gradient Magnitude Based:

This algorithm uses the gradient values of the pixels along the contours. The assumption is that segments of the contour with larger gradients are most likely crest-line candidates.

Using the gradient magnitudes provide much more accuracy in localizing the desired crest-lines. The gradient orientation based approach has an inherent localization problem do the the computation of the gradient orientation and size of the kernel to compute the gradients. The contours will almost always be shifted slightly away from the true crest-line as shown in Figure 24.

Figure 24 is the exact same as the plot shown in Figure 7(c) with the crest-line plotted, and the shift shown. The shift shown is the transition from gradients which *agree* with the dominant orientation ($D_{ij} > 0$) to those which *disagree* with the dominant orientation $D_{ij} < 0$. The shift is heavily dependent on the size of the kernel K used to compute the gradients.

The contours detected will inevitably land on the orientation transition. As shown in Figure 25, the parameter K will have a tendency to shift the contour lines towards the dominant orientation. In order to improve localization, the contours must be shifted towards to nearest gradient magnitude peak.

Two methods can be used to shift contours towards the crest-lines. Since the shift is simply dependent on the parameter K , a fixed shift can be applied in the opposite direction of the dominant orientation. This approach may be adequate if localization is not critical, but results produced are generally less accurate.

A better solution is to search in the opposite direction of the dominant orientation for the nearest gradient magnitude peak. Each point on the contour can then be shifted by the computed distance to each gradient peak.

Parameter Effects

For this approach, there are relatively few parameters to optimize. The main parameters are:

1. The value of K , which determines the size of the kernel which computes the gradients. The effects of varying this parameter are shown in Figure 25. Choosing the appropriate value for K depends largely on the scale of the

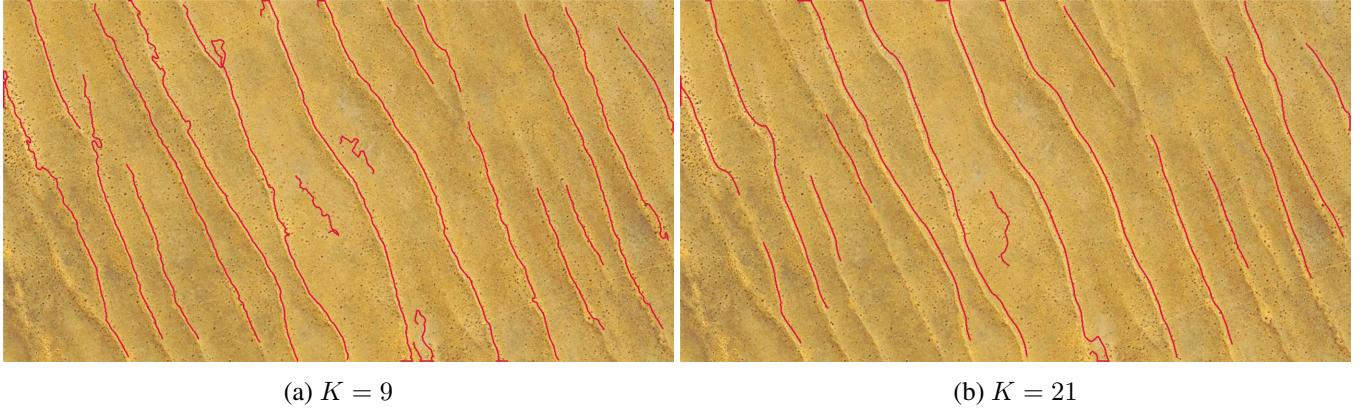


Figure 25: Effects of K on the orientation-based approach. In (a) a K value of 9 is used, which results in better localization at the cost of noisier results, (b) a K value of 21 produces much smoother results but poor localization.

provided image, size and spacing of the dune crests. Larger kernels typically produce smoother results as shown in 25(b), but result in poor localization.

2. The value of R , which determines the intensity or gradient magnitude value threshold for rejecting false positives. This is the main value used to determine the appropriate level of detection. Lower threshold values typically increase the detection rates and false positive rates.
3. The minimum segment length value, which determines the minimal allowable length for the detected segments. This parameter depends largely on the scale or resolution of the image, and the desired precision of the crest-line detection. Segments which are shorter than the minimum segment length are assumed to be noise, and are filtered out.
4. The threshold value for the dot product, which determines how closely edges must match the dominant orientation. A value of 0 appears to be adequate in most cases, but it could be adjusted to suit the needs of the application.

Overall, using the orientation information has merit and provides valuable information for crest-line detection. The approach is used and further improved in the approach presented in section 3.9. The next approach presents a traditional machine learning centric approach.

3.8 Machine Learning Approach

In the machine learning approach, the goal is to teach a classifier to recognize crest-lines vs non-crest-lines. A similar approach was implemented in [20, 110, 10, 66, 90, 17]. The goal of this approach is to create a machine learning model using a training dataset, then generate the response map which is the response of the model at each pixel in a given image. Ideally, the responses are high at pixels which contain crest-lines, and low everywhere else. The response map can be thresholded and local maximum can be traced in order to retrieve the crest-lines. The overall flow is illustrated in Figure 26.

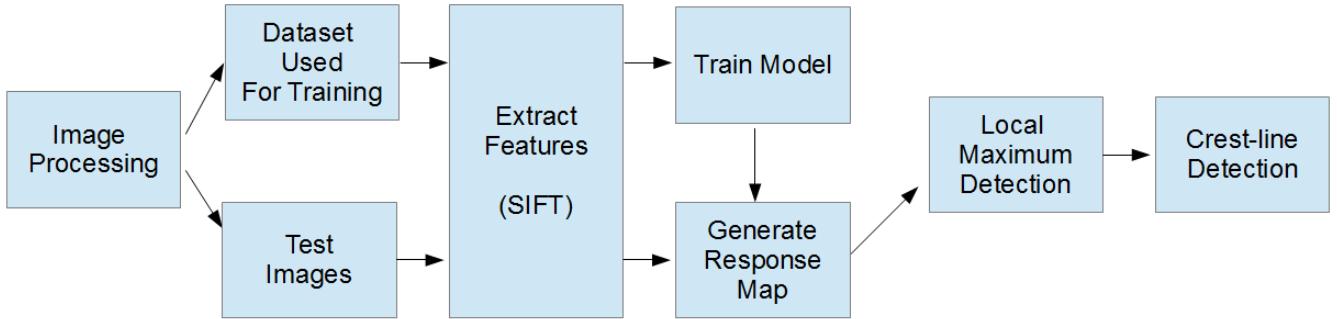


Figure 26: The process flow of the machine learning approach. The image is first preprocessed, and some training data (samples of crest-lines and non-crest-lines) is reserved to construct the machine learning model. SIFT features are extracted from the training data to train the model to recognize crest-lines. With the model trained, SIFT feature descriptors are computed at each pixel of the remaining test images and are inputted into the trained model to generate the response map. Finally, the local maxima of the response map are detected in order to extract the crest-lines.

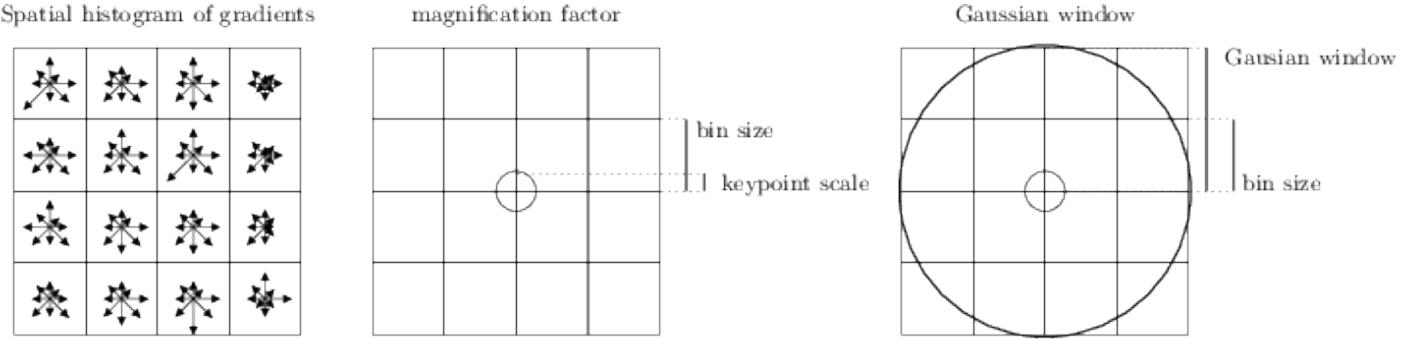


Figure 27: The SIFT descriptor explained. The image patch is split into a 4 by 4 grid, where each cell contains a histogram of gradients of eight directions. The result is a single vector of 128 floating point values. Source of the image: <http://www.vlfeat.org/api/sift.html>

The first step of the process is to appropriately separate the data into training, validation, and test sets in order accurate report the performance of the method.

3.8.1 Dataset Creation

Machine learning techniques typically require training data to construct the model in order to reliably solve classification problems. This method of training is called supervised learning ([71, 74]). For crest-line detection, we therefore need to split our data into two classes: crest-lines and non-crest-lines.

For each class, the data also needs to be separated into three sets for training, validation and testing.

Training The training set is used to construct the model to learn the task at hand.

Validation The validation set is used to evaluate the effectiveness of the trained model. Based on the results produced on the validation set, the model parameters can be tuned to produce better results on the validation data.

Testing The test set is an independent set used to report the true performance of the model.

In order to build the model for crest-line classification, the first step is to collect the data to build each set. In this application, our datasets included the ground truth for each image set. The ground truth is essentially a binary image with the labeled crest-lines.

To construct an appropriate dataset for the approach, the set of images is first split into the training/validation and test sets. Half of the images are reserved for testing, and the remaining half is used to training and validation. From the set of training images, the ground truth images are used to extract locations of crest-lines and non-crest-lines. For each image, pixel positions are chosen randomly, if the ground truth at the position contains a crest-line, the pixel location is added to the crest-line set, otherwise it is added to the non-crest-line set. The sampling is optimized so that N samples are chosen for each class.

The resulting output is that each training image now has N samples of crest-lines, and N samples of non-crest-lines. The next step in the process flow is to extract features from each of these samples, which will be used to feed into the model for training and validation.

3.8.2 Feature Descriptor Extraction

Once sample locations of crest-lines and non-crest-lines have been selected, descriptors need to be extracted from each sample location. A descriptor is typically a vector of values which describes the local neighborhood around a pixel location ([67, 49, 63, 114]). The range of possible descriptor types is almost limitless. Descriptors can range from simple pixel values in an image patch, to gradient information, to more sophisticated descriptors such as HOG (Histogram of Oriented Gradients [16]), LBP (Local Binary Pattern [108, 109]), Haar Features ([116]), SURF (Speeded Up Robust Features [42]), SIFT (Scale Invariant Feature Transform [67]), and many more.

As part of this research, many of those types of features have been investigated. However, the SIFT descriptor has been shown to provide the best overall results. One downside of the SIFT descriptor is the computational requirements of extracting and matching the descriptors. The main reason is the size of a standard SIFT descriptor, which is a vector of 128 floating point values, and becomes computational significant as the number of data samples increase. Despite this downside, SIFT is still an appropriate option since there are no real-time requirements in this application.

The SIFT descriptor, originally proposed in [67] has been a standard in computer vision applications for over a decade. SIFT features extract a scale and rotation (and possibly affine) invariant region around a point of interest, and applies the descriptor to it. The descriptor is computed by splitting the scale invariant region around the point of interest in a 4 by 4 cell grid. In each cell, the histogram of gradients of 8 directions as shown in Figure 27. The histogram in each cell is then concatenated into a single vector of 128 floating point values, and L_2 normalization is applied to normalize the vector to unit length.

In this application, the descriptor is applied to each pixel on the image, therefore there is no need for point of interest detection. To compute the SIFT descriptor, the scale and orientation of the local feature is needed. The scale in our research is ignored and is replaced by using a fixed-size window around every pixel. On the other hand, the orientation is calculated using the gradients at the local pixel location. In future work, there could be some improvements to results by computing the local scale and computing the orientation at the computed scale. Some work was done on this, but no significant improvement was noticed.

After computing the descriptor for each crest-line and non-crest-line training/validation and test samples, the classifier model can then be constructed, optimized, and validated.

3.8.3 Supervised Learning of the Crest-line Classifier Model

The supervised learning method to build a classifier requires a dataset of labeled samples. Typically, each training sample is an input vector of values, with an expected output response. In this application, the input vectors are the features extracted, such as SIFT. The output response would either be a crest-line (assigned a response value of 1), or a non-crest-line (assigned a response value of -1).

The dataset of training data and their expected responses are fed into the classifier which attempts to fit a model or estimate an optimal function which separates the classes (crest-line versus non-crest-line) for the given input.

There are many different kinds of classifiers in machine learning: Bayesian, decision trees, artificial neural networks, Support Vector Machines, and many more ([96, 92, 86, 70, 41, 120, 9, 4]). As part of this research, we studied the various classifiers, notably SVM and Gradient Boosted Trees ([29, 30]) which provided the best performance with the SIFT features for crest-line recognition.

To train the classifier, the dataset extracted (explained in section 3.8.1) of N crest-line samples and N non-crest-line samples are used. Many of the classifiers have free parameters to optimize, therefore a *K-Fold* validation is used to construct the optimal classifier.

K-Fold validation is the process of splitting the data into K folds, where one fold is used for validation and the remaining are used to train. The validation set is used to analyze results of samples not seen by the trained model, validating the reliability of the model. Typically, the process is repeated for each fold to ensure proper data representation. This process produces an optimal solution for the model while minimizing over-fitting to the training data. Over-fitting is a concept in machine learning which is defined as over learning the training data which results in weaker performance on validation and test examples ([40, 47, 32]).

For this application, four main classifiers were tested: SVM, Normal Bayes, Random Trees, and Gradient Boosted Trees, all provided within the OpenCV framework [7]. Once the models have been trained, they are tested using a test dataset. The results of the training of these classifiers are shown in section 4.3.

With the models recognizing crest-lines versus non-crest-lines, the next step is to construct a response map at each pixel of the images.

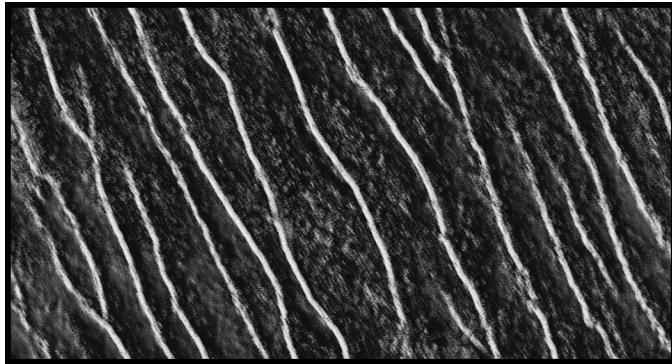
3.8.4 Response Maps

Response maps are images where each pixel values represent a response to a certain function. In the case of this method, the SIFT descriptor is first computed at each pixel and then used as an input for the trained classifier. Since the classifier was trained to output 1 (for crest-lines) or -1 (for non-crest-lines), the range of response values is then $[-1, 1]$. The resulting image is the response map of candidate crest-line.

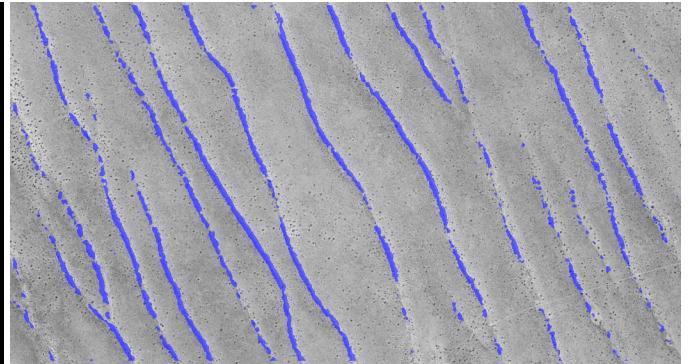
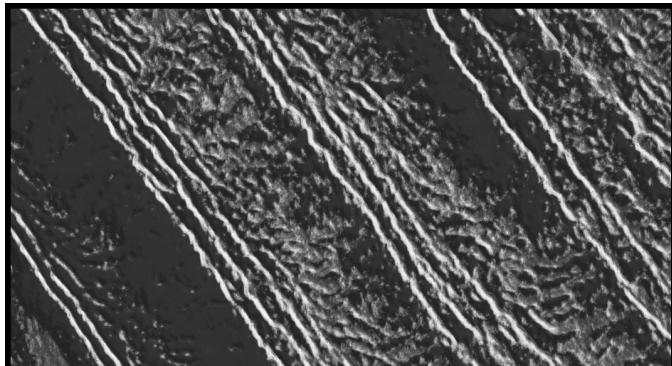
The response map therefore produces an image where bright regions are likely crest-line candidates, and darker regions can be filtered out using a simple threshold. The two main classifiers which produced generally better results are the Support Vector Machines and the Gradient Boosted Trees.

The results of the SVM classifier are shown in Figure 28. On the left are shown the response maps for the SVM classifier (defined as R), and on the right are the thresholded results overlapped on top of the image. This is accomplished by simply thresholding the response map R for all pixels (i, j) which are $R_{i,j} > 0$.

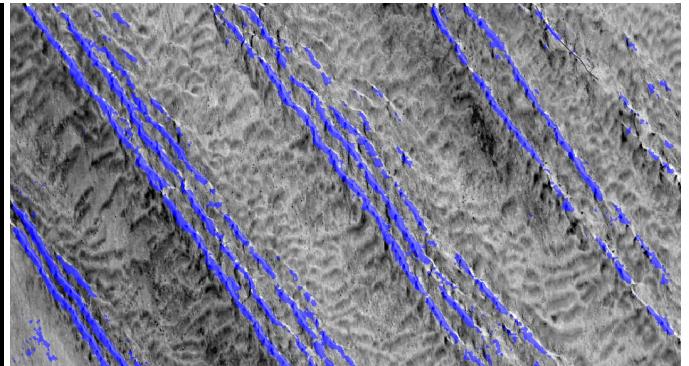
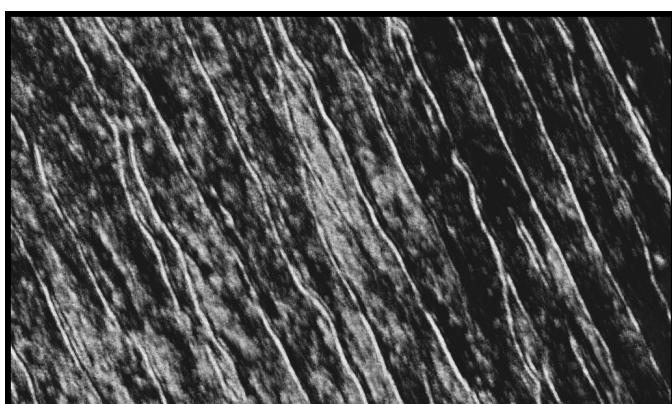
The results shown are acceptable but have a tendency to be a bit noisier. There are noticeable breaks in the results, which are especially visible in Figures 28(b) and 28(h). Additionally, as shown in the results Table 6a, the SVM classifier seems to over fit the training data slightly.



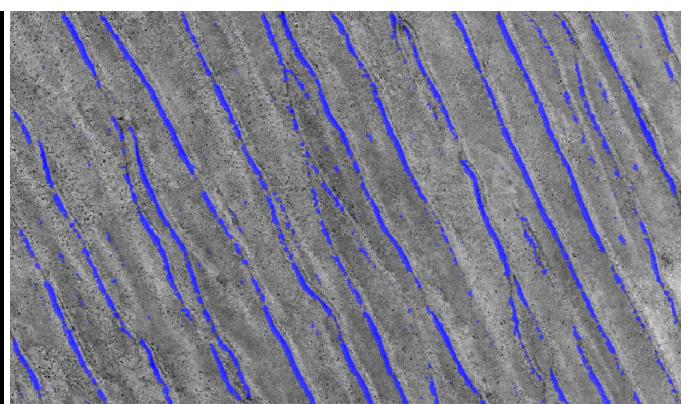
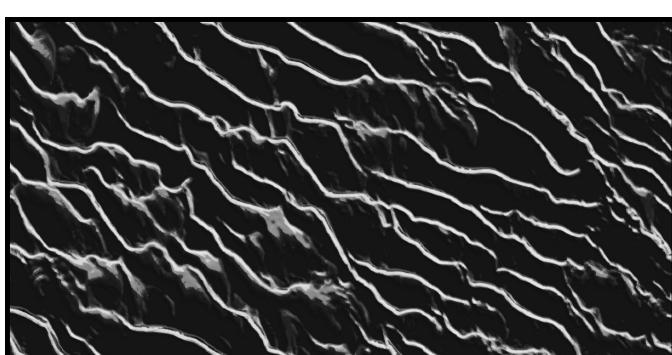
(a) Kalahari Response Map (SVM)

(b) Kalahari $R_{i,j} > 0$ 

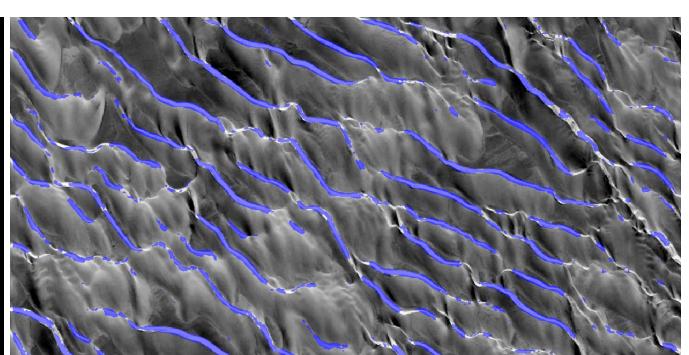
(c) Namib Response Map (SVM)

(d) Namib $R_{i,j} > 0$ 

(e) Simpson Response Map (SVM)

(f) Simpson $R_{i,j} > 0$ 

(g) Skeleton Coast Response Map (SVM)

(h) Skeleton Coast $R_{i,j} > 0$

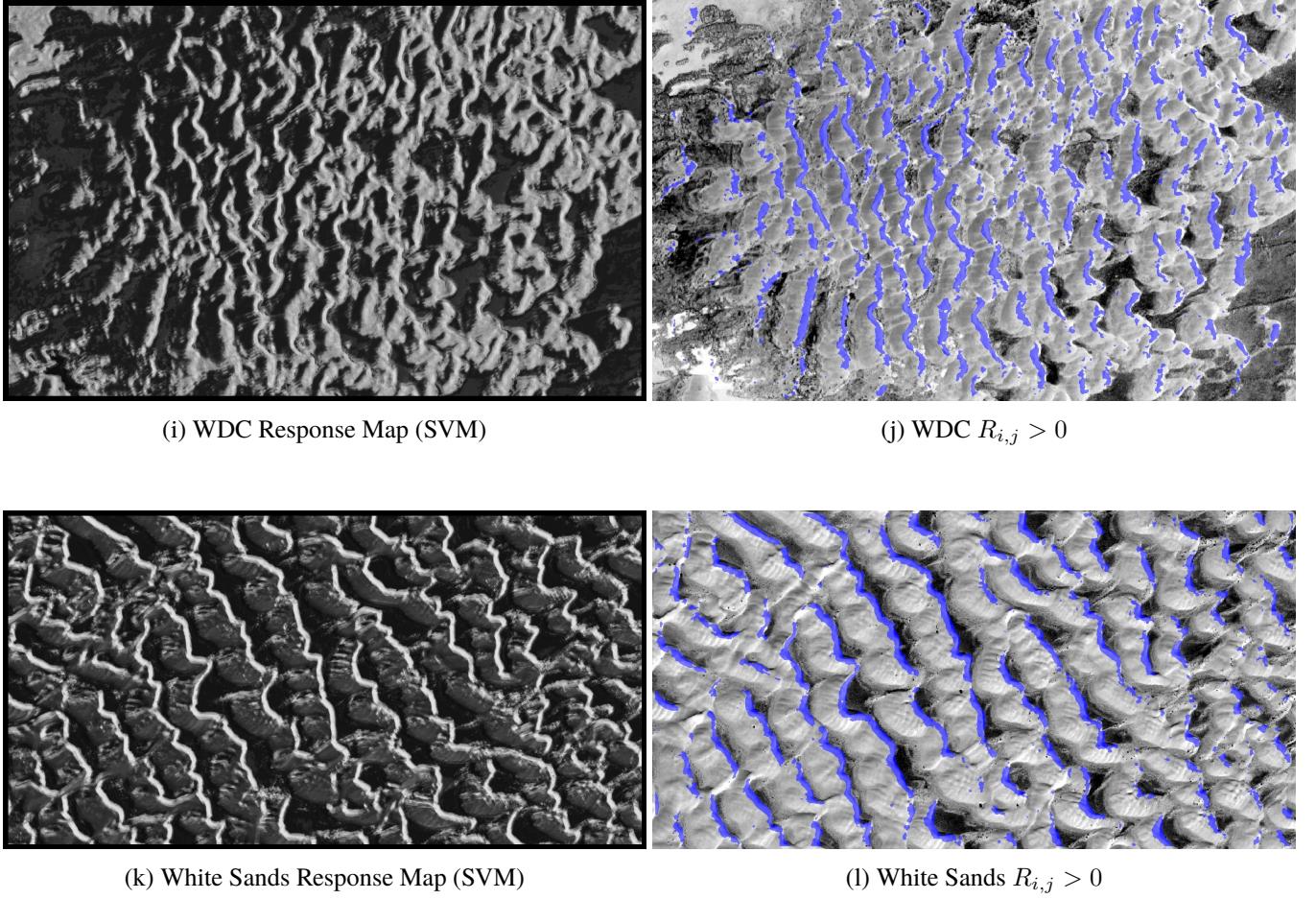
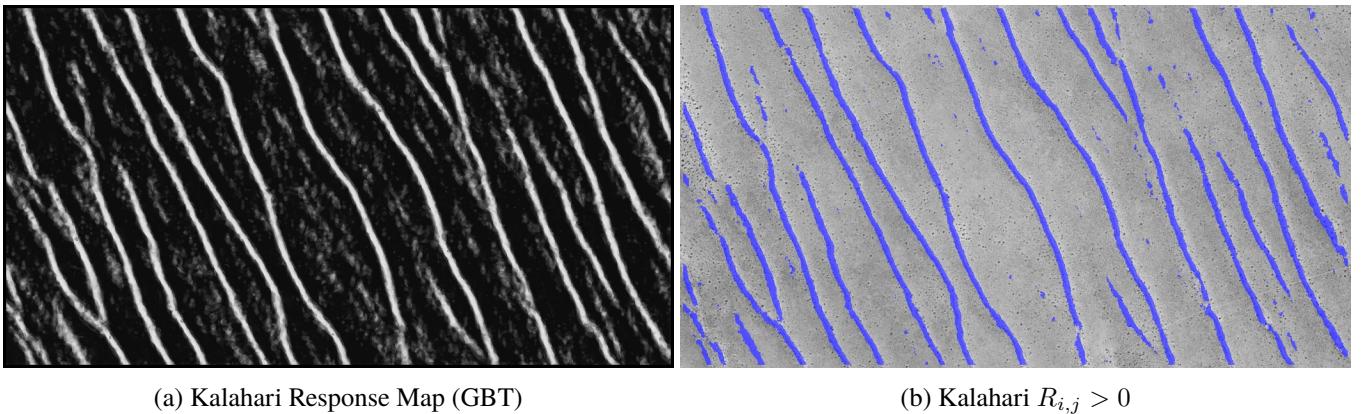
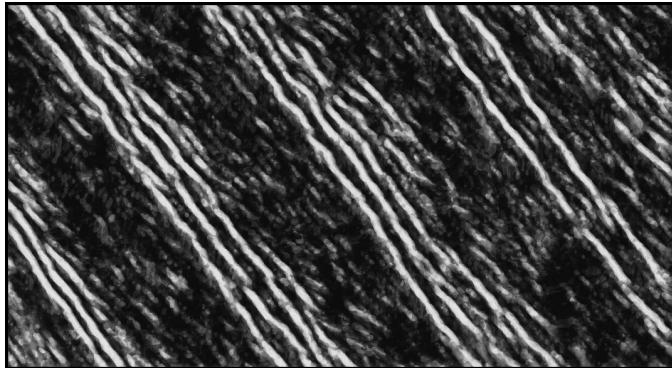


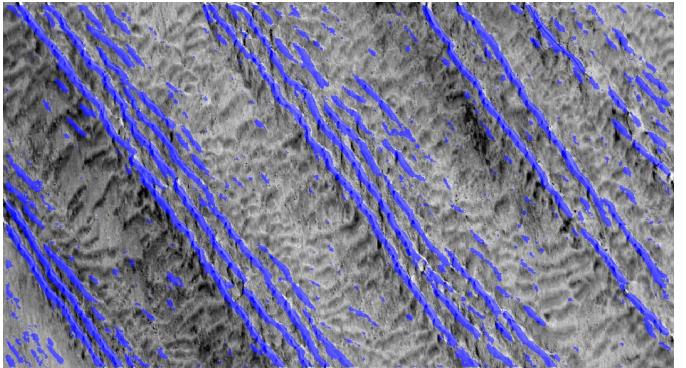
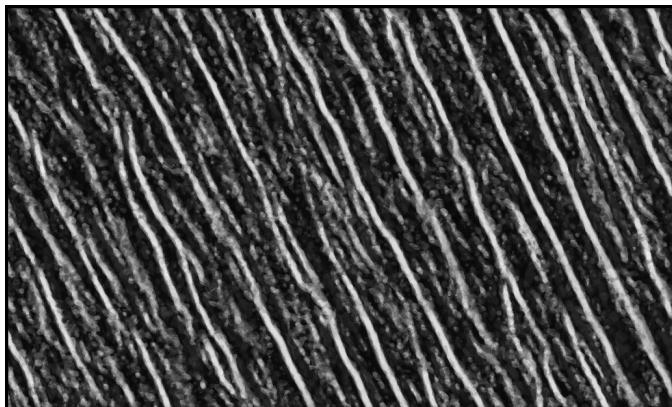
Figure 28: Results of the Support Vector Machine classifier training for crest-line identification for the Terrestrial dataset shown in Figure 39. On the left is the response map (the output response of the SVM classifier using the SIFT features at each pixel). On the right, the thresholded the crest-line responses ($R_{i,j} > 0$) of the response map are overlapped on the original image.

Alternatively, the Gradient Boosted Trees classifiers seems to produce slightly smoother results. The results of the Gradient Boosted Trees classifier training are shown in Table 6d. In Figure 29, the response maps produced by the GBT classifier are shown, similar to Figure 28. It is clear that the resulting response maps are generally better than the SVM model.

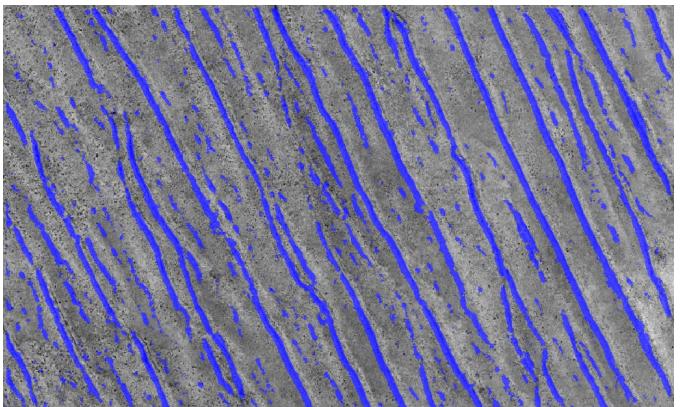
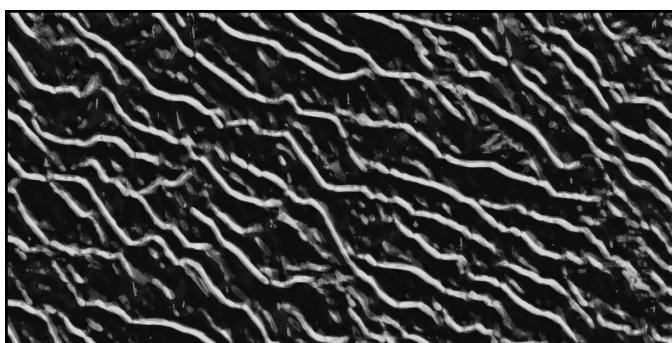




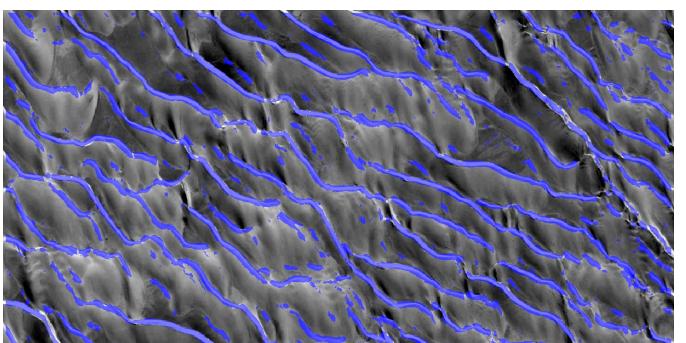
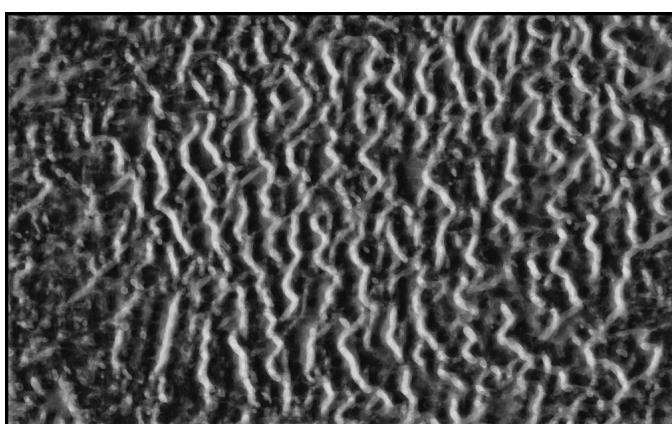
(c) Namib Response Map (GBT)

(d) Namib $R_{i,j} > 0$ 

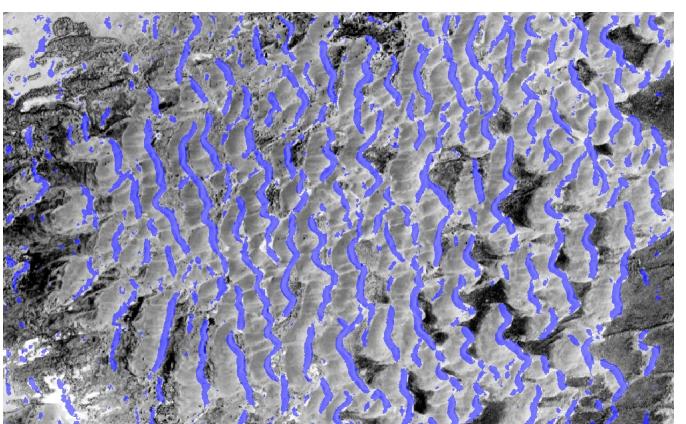
(e) Simpson Response Map (GBT)

(f) Simpson $R_{i,j} > 0$ 

(g) Skeleton Coast Response Map (GBT)

(h) Skeleton Coast $R_{i,j} > 0$ 

(i) WDC Response Map (GBT)

(j) WDC $R_{i,j} > 0$

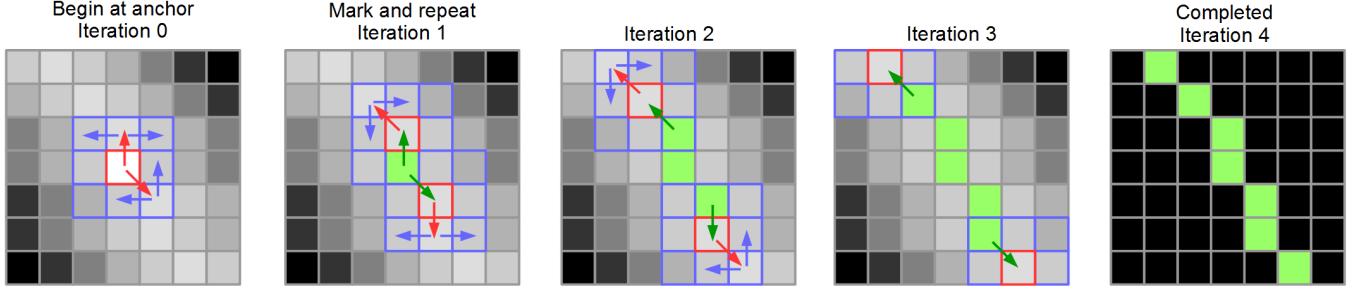


Figure 30: A recursive algorithm for ridge following: On the first iteration, the algorithm is initialized using the detected anchor pixels, and begin search for each neighbor. Neighbors which are peaks along the ridge are used as anchors for the iteration. This algorithm recursively follows the ridge of the response images.

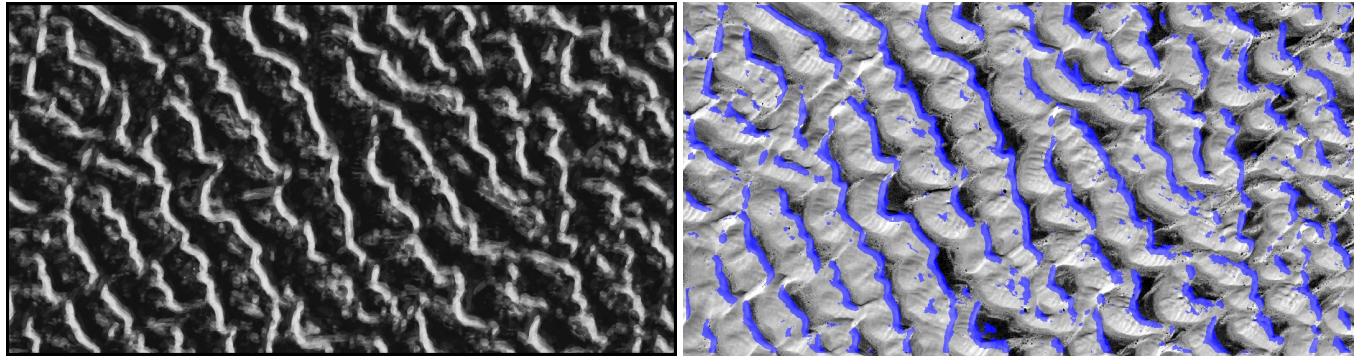


Figure 29: Results of the Gradient Boosted Tree classifier training for crest-line identification for the Terrestrial dataset shown in Figure 39. On the left is the response map (the output response of the GBT classifier using the SIFT features at each pixel). On the right, the thresholded the crest-line responses ($R_{i,j} > 0$) of the response map are overlapped on the original image.

With the classifier model trained and the response maps generated, crest-lines can be extracted from the maps, by finding the peaks in the responses. The desired output for crest-line detection is a single pixel wide line which is smooth and contiguous. The brighter response regions tend to be thick and noisy, requiring some thinning for proper crest-line candidates to be defined. The following section will discuss the methods used to thin the responses.

3.8.5 Crest-line Extraction: Thinning And Skeletonization

In order to extract the desired crest-lines, the response map must be thinned. The thinning process is typically a morphological operation which reduces the thickness of a region to a single pixel width. An overview of thinning algorithms is explained in [93, 39, 106] and many other publications.

The response map should have very large values where dune crest-lines are located. The goal is to find the local maximum of the responses along ridges. The assumption is that the crest-lines should form a ridge of high response values which can be traced. As a first attempt to solve this, a recursive algorithm was created to trace the crest-lines based on the response values. The algorithm is shown in Figure 30.

The first step in the algorithm is to find *anchor* points, which are starting points for the recursive algorithm. The so-called *anchor* points are pixels for which the response value is larger than a defined threshold, and which have greater value than the eight neighboring pixels. Then, for each neighbor, a comparison in response value is done with their neighbors. If a neighbor has a response value greater than its two neighbors, then it is marked, and inserted into the queue of the recursive algorithm. The process is repeated until all ridge points are marked and the end of the ridge has been reached. Results of this process are shown in Figure 31.

Although the algorithm does thin the response maps, the results have a tendency to be noisy and disjoint. To remove noise, some Gaussian smoothing can be applied, which improves the noisy output, but does not resolve the issue of disjoint segments. Ideally, the crest-line segments should be contiguous segments of a certain specified size. One way to resolve

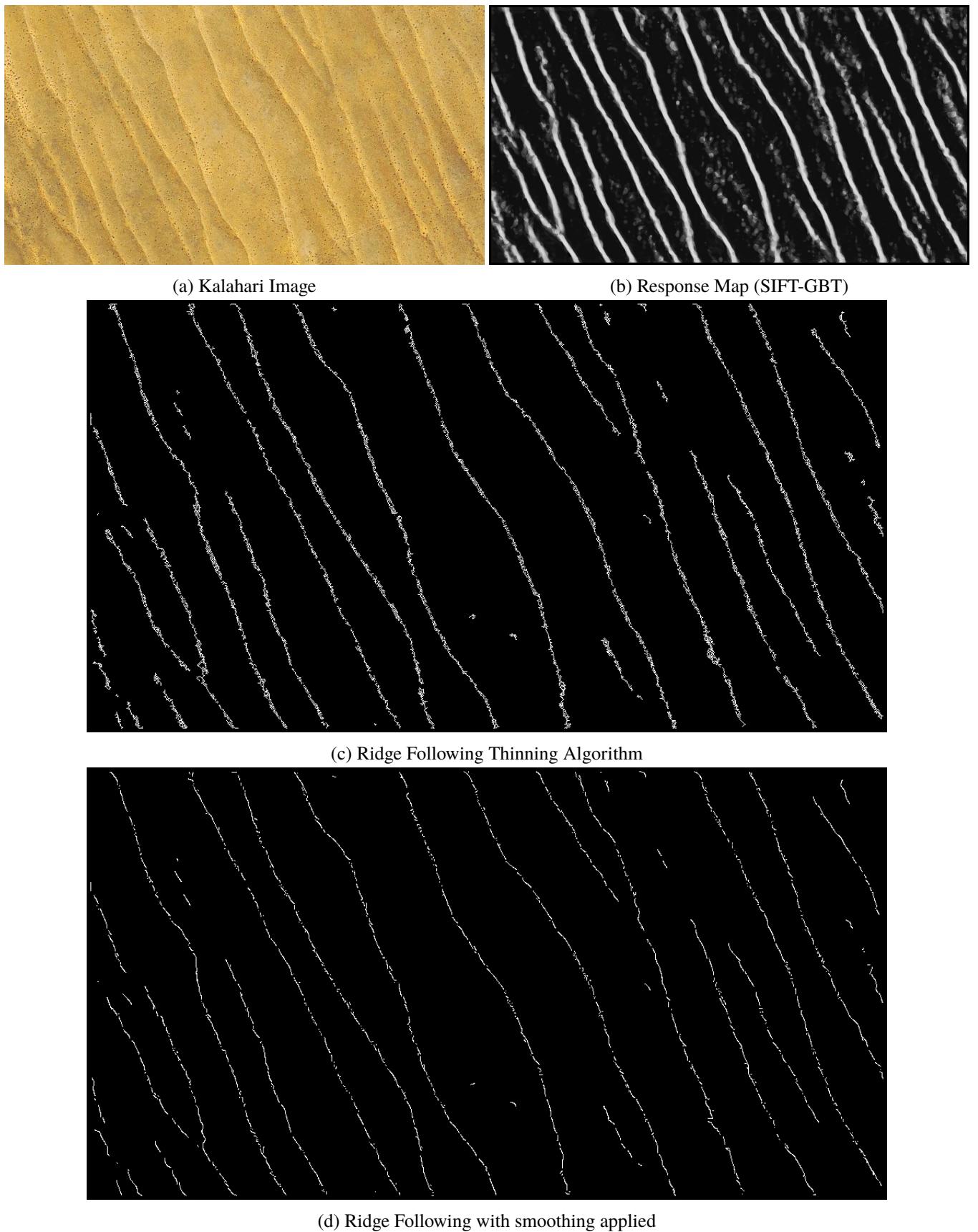


Figure 31: The results of the recursive ridge following algorithm applied on the response map of the Kalahari image trained with the SIFT-GBT classifier. The results produced appear noisy and disjoint even with smoothing applied.

this issue is to apply edge linking techniques such as defined in [12]. No attempt to apply edge linking was made as part of this project, therefore more research could be done in to resolve the problem.

A better alternative is to use Skeletonization algorithms, such as explained in [102, 121, 79, 111]. Skeletonization is the process of creating a skeleton (often referred to as a *topological skeleton*) of an object or region. The goal of skeletonization is to shrink a region into a thin skeleton of the object or region. It is used in a variety of applications for its geometrical and topological properties. It enables the computation of features such as length, orientation, connectivity, and other features.

The algorithm chosen to perform the thinning was the Zhang/Suen thinning algorithm from [111], due to it's simplicity and efficiency in implementation. Better algorithms may improve results of the thinning process, but for this application, the Zhang/Suen algorithm is sufficient. Before applying thinning to the response map, regions are created by thresholding the response map for all $R_{i,j} > 0$, as shown in Figure 32(c). Then, the Zhang/Suen thinning algorithm can be applied to the binary image, shown in Figure 32(d).

The results shown in Figure 32 are much better than the recursive algorithm shown in Figure 30. The skeletonization algorithm produces smooth contiguous segments which closely match the actual crest-lines. The final step of the method is to apply connected components to retrieve each dune segment. The length of each segment is measured and all segments which are shorter than a defined threshold can be filtered out as noise.

The results of are shown in Table 8. The next section discusses a mixed approach, which combines elements and algorithms of the machine learning approach and previous approaches.

3.9 Machine Learning and Gradient-Based Mixed Approach

The final approach which was studied as part of this research seeks to combine the concepts discussed in the previous approaches. From section 3.3, the use of contours to retrieve contiguous segments is used. The dominant orientation calculation, defined in section 3.4, is also a key component in this approach. The use of gradient orientation maps defined in section 3.7, and the machine learning concepts presented in 3.8 are also utilized to maximize performance of crest-line detection.

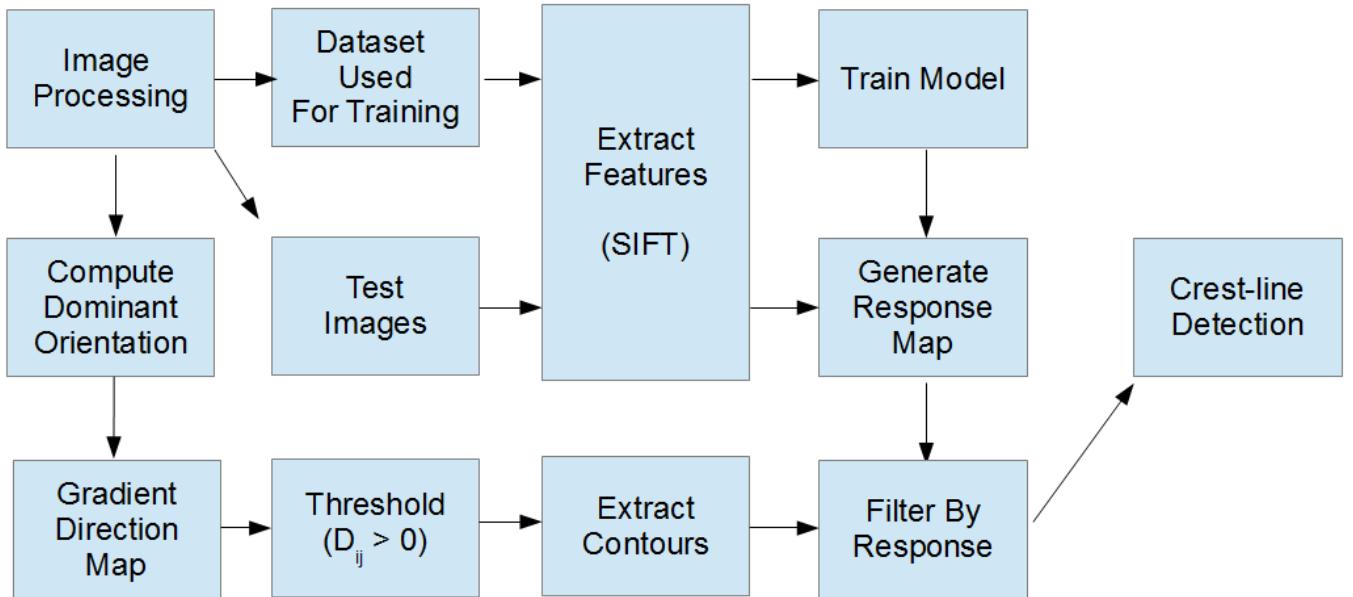


Figure 33: The process flow of the machine learning and gradient-based mixed approach. The image is first preprocessed, and the machine learning model training is identical to the approach presented in 3.8. Simultaneously, the dominant orientation is computed on the images, the gradient direction map is extracted, which is thresholded based on the dot product, and contours are extracted (in similar fashion to the approach explained in 3.7). Finally, the contours are filtered based on the response map, resulting in the detected crest-lines.

As usual, each image is preprocessed to remove noise and enhance the image quality. From the processed images, a set of sample points of crest-lines and non-crest-lines are chosen randomly from a set of training data, as explained in section 3.8.1. As we learned in sections 3.8.2, 3.8.3, and 3.8.4, the features are extracted (SIFT), the classifier is trained (Gradient Boosted Trees), and the response map is created. The process of creating the response map is identical to the previous section, which is used to improve the localization of crest-lines.

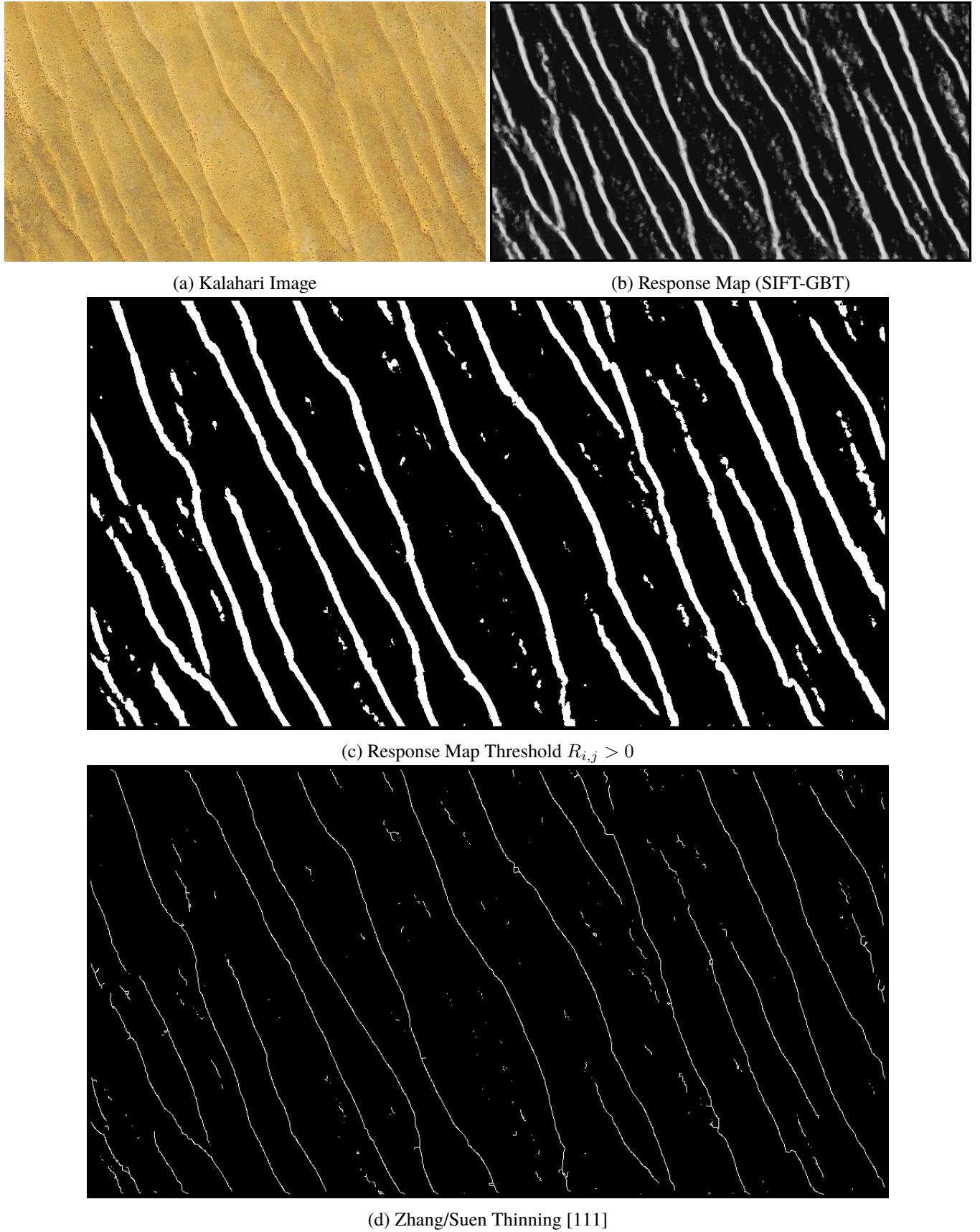


Figure 32: The results of Zhang/Suen Thinning [111] algorithm applied on the response map of the Kalahari image trained with the SIFT-GBT classifier. The output is much smoother and contiguous segments than the recursive ridge following method shown in Figure 30.

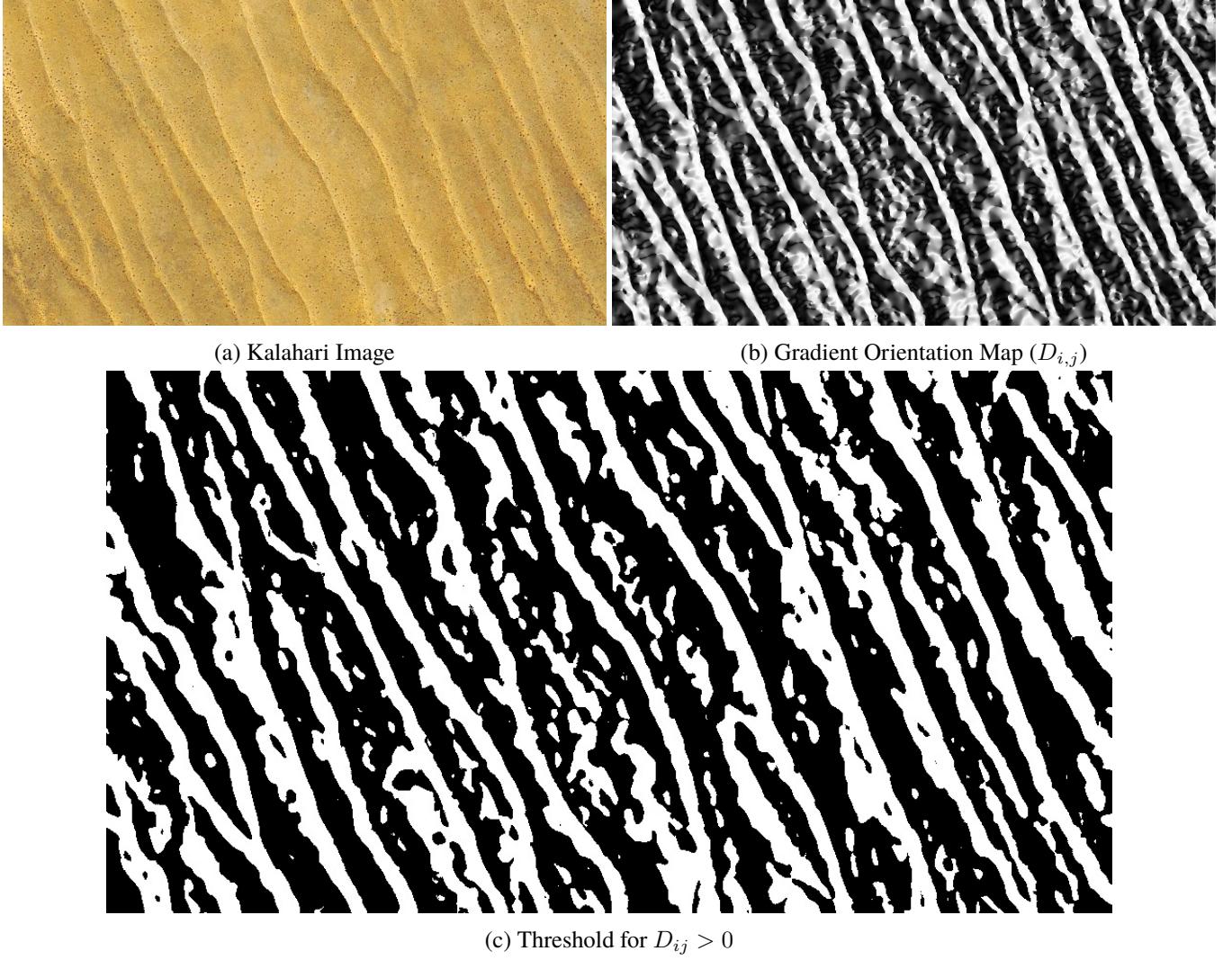


Figure 34: Computing the Gradient Orientation Map (b) ($D_{i,j}$) for the Kalahari image (a). The map is thresholded for all gradients which agree with the dominant orientation ($D_{ij} > 0$) in (c). The result produces smooth contiguous regions which contain candidate crest-lines.

3.9.1 Gradient Orientation Crest-line Extraction

In this process, extracting crest-line candidates is done using the method presented in section 3.7. To reiterate, the gradient orientation method computes the dominant orientation of the dune field, and computes the dot product of the gradients at each pixel with the computed dominant orientation, as shown in Figure 34(b). This produces an image where each pixels measures the *agreement* with the dominant dune field orientation.

The image produced, named the gradient orientation map, is thresholded resulting in a binary image containing regions which crest-line candidates. From the binary image, contours can be extracted from the regions using the same method shown in sections 3.3 and 3.7. The contours are typically smooth and contiguous, but result in poor localization. Also, there are two sets of contour around the regions. The goal is to therefore filter out one side of the contours, keeping only segments which land on the crest-line.

An additional step done in this approach is to improve the localization of the crest-lines based on the response map.

3.9.2 Localization Improvement of Crest-line Segments

As stated previously, the localization of the candidate crest-line contours do not land exactly on the crest-line ridges. We discussed in section 3.7 the issue of localization of the contours, the cause of which is illustrated in Figure 24. The shift explained in those sections was even more pronounced with larger values of K (the size of the kernel used to compute the gradients), as shown in Figure 25.

To further expand on the problem, Figure 35 shows the shift created from the binary image contours. In order to improve the localization of the candidate contour segments, each segment needs to be shifted towards the crest-line ridge. The result of the contour segment extraction is shown in Figure 36(b), clearly showing the localization issue. Two considerations must be made in order to properly shift the candidate crest-lines to the actual crest-line: the direction of

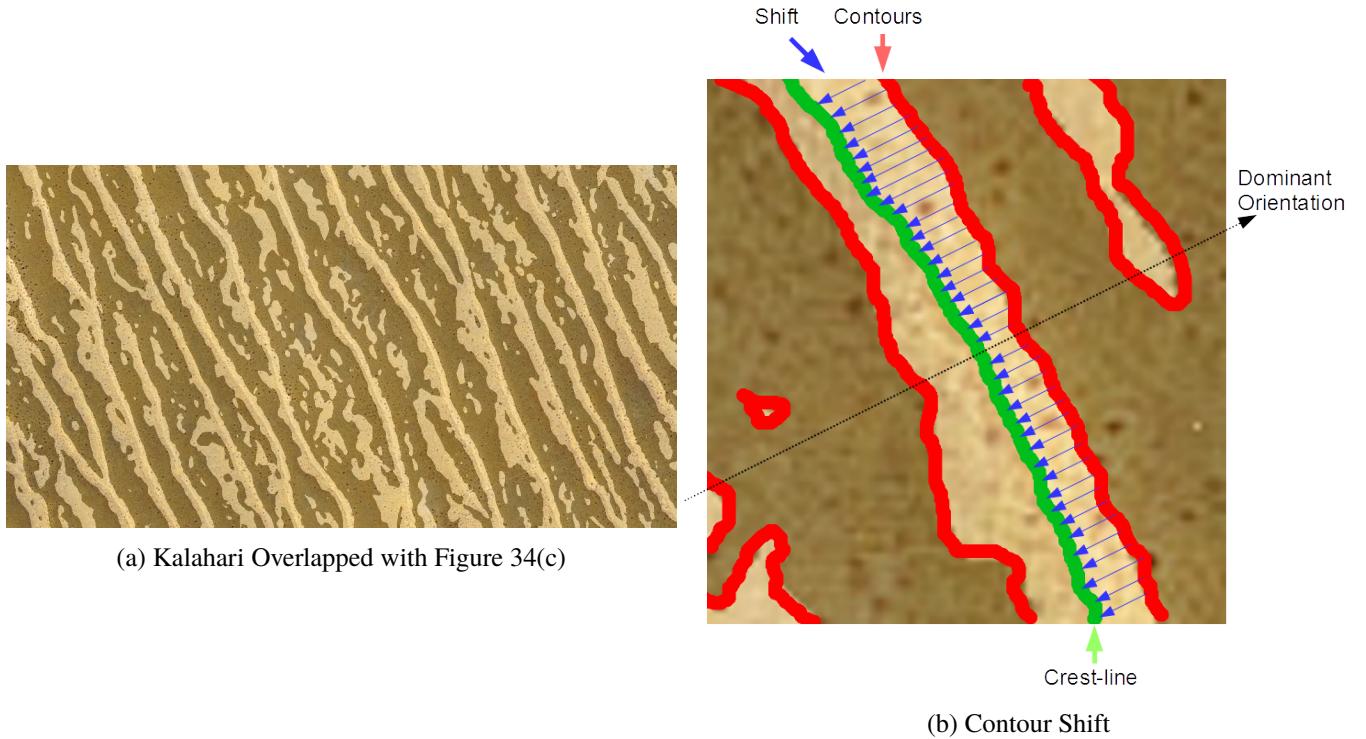


Figure 35: Illustration of the issue of poor localization of the extracted contours. In (a), the binary image shown in Figure 34(c) is overlaid on top of the Kalahari image 34(a). In (b), a zoomed in image patch reveals the localization issue with the contours (shown in red) from the actual crest-line (shown in green). In order to improve the localization, the contour segments must be shifted towards crest-line. The shift is applied in the direction opposite to the computed dominant orientation.

the shift, and the algorithm used to shift.

Because of the way the gradient orientation map was computed, an appropriate shifting direction can be in the opposite direction of the dominant orientation, as shown in Figure 35(b). There may be better solution for shifting, but using the dominant orientation makes sense in this case and generally produces good results. Since the direction of the shift is known, the next step is to find the actual crest-line ridge along the direction of the shift. The solution proposed in 3.7 was to find the nearest gradient magnitude peak. Unfortunately, as discussed in 3.1, some crest-line gradients may be relatively weak, making localization a difficult task.

Therefore, using the response map computed in the previous approach seems to be an appropriate way to resolve this problem. The response map will have larger values where crest-lines are located. So, the shift can be determined by doing a search along the opposite direction of the dominant orientation, and finding peaks in the response map. The shift can then be applied based on the computed distance, for each point along the contour segments. Smoothing can be applied in order to ensure that segments remain contiguous after the shift is applied. The results of the shift are shown in Figure 36(c).

Once the shift has been applied to every contour segments, filtering non-crest-line segments can be done by simply removing all contour segments which fall on weak responses. After all contours have been filtered, the remaining segments can be further filtered by segment length. All segments which are too short are removed from candidate crest-line segments. The output of this process is shown in Figure 36(d).

Overall, the mixed machine learning and gradient orientation based method produced excellent results in most cases but struggled on more complex dune systems. Section 4.3 summarizes the results of this approach on both our dataset and the dataset provided by [17]. The next section discusses the method and metrics computed from the detected crest-lines.

3.10 Dune Field Morphology Metrics

The final step of the process is to extract meaningful features and metrics for researchers to study from the dune fields. This process requires a good set of detected crest-lines. The output crest-line segments extracted in the previous methods are used as input for computing the features.

There are many features metrics of interests to scientists in this field, including dune spacings, orientation, shape, size, length, bifurcations, and many more. These metrics can be used to study and understand the movements of various patterns of dunes over time in a specific area.

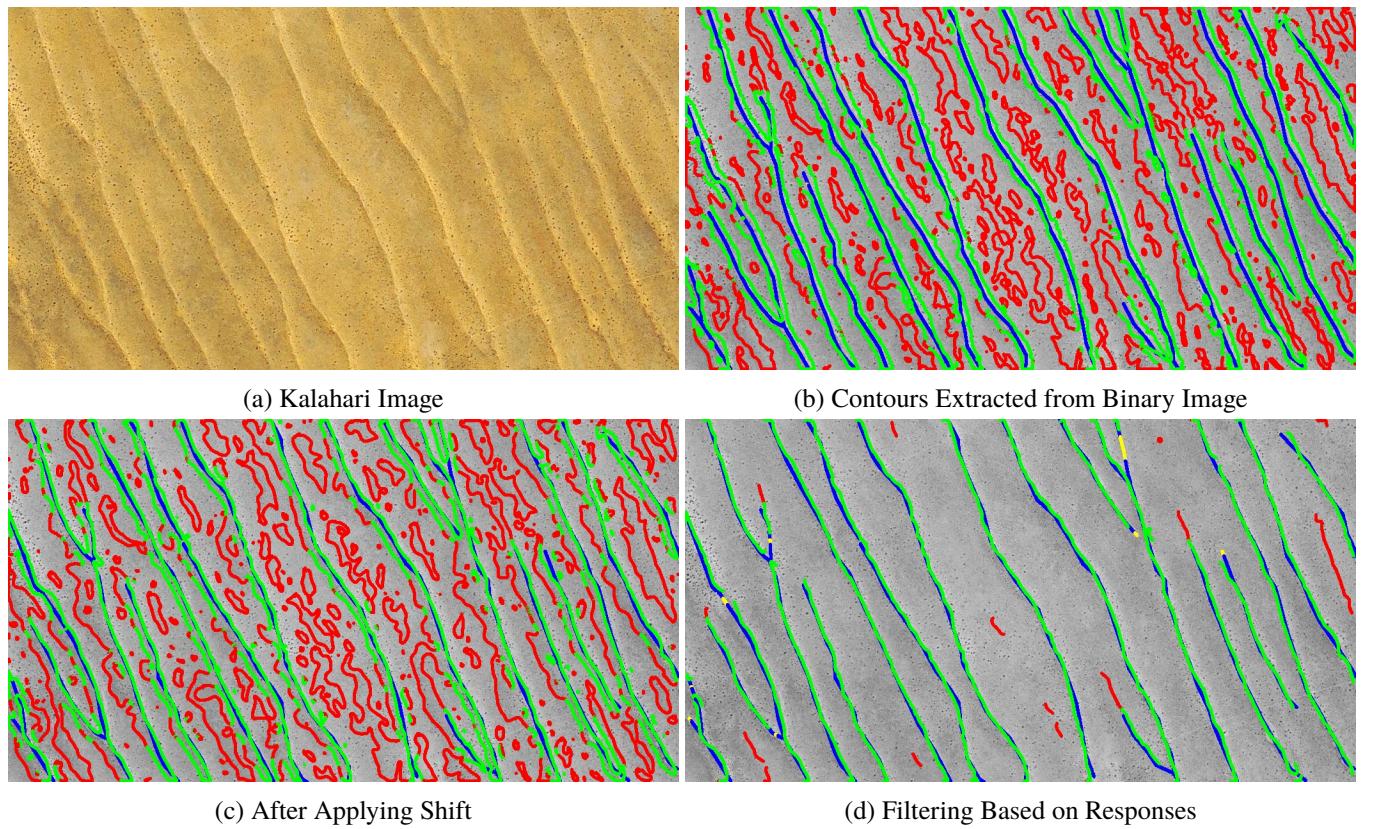
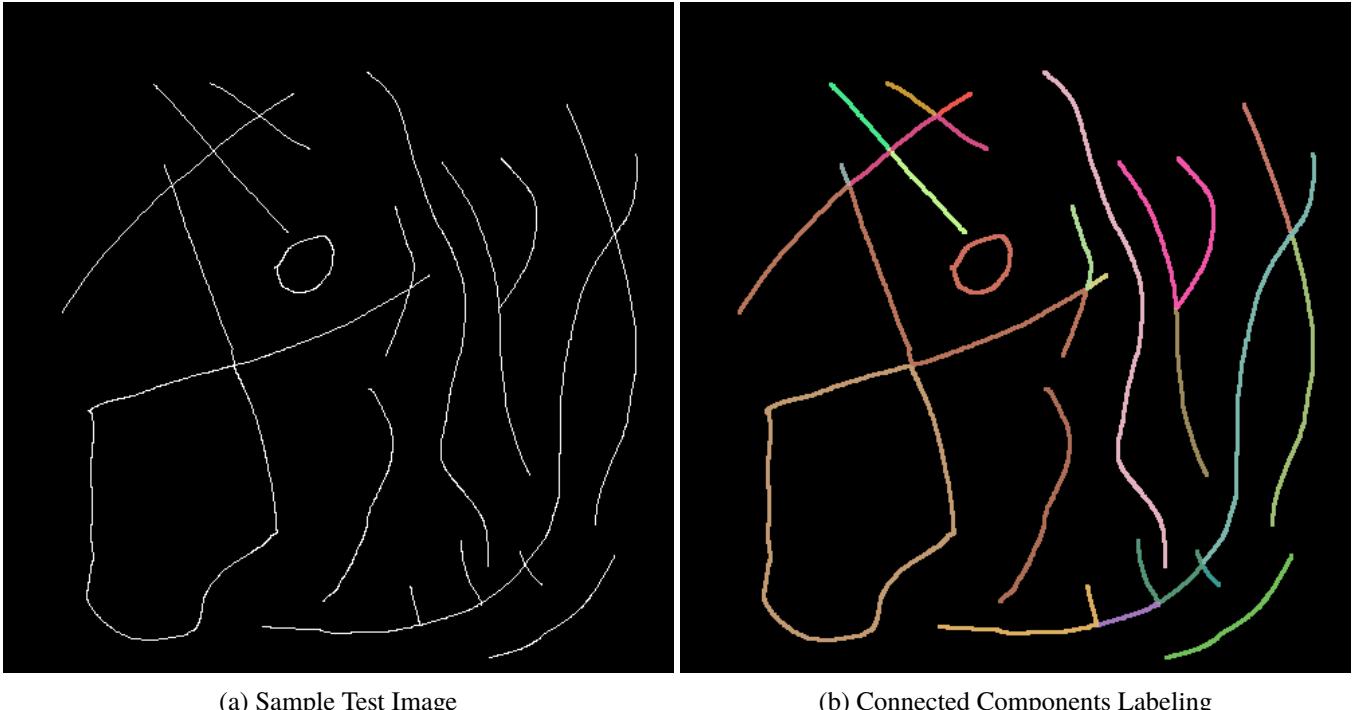


Figure 36: The process of extracting candidate crest-lines using the gradient orientation map, and the machine learning approach. The Kalahari input image shown in (a), is processed to extract contours shown in (b). A shift towards the nearest peak in response values is applied (c), and all segments which have low responses are filtered out in (d). The color scheme shown in (b-d) is true positive crest-lines detections (green), the positive identified ground truth (blue), false negatives (yellow), and false positives (red).



(a) Sample Test Image

(b) Connected Components Labeling

Figure 37: The connected component algorithm applied a sample test image (a). The results (b) show that segments can be extracted correctly.

As part of this research, we focused mainly on orientation and spacing of dunes, and used those metrics as validations for the effectiveness of the dune detection methods. Although these metrics are not a complete evaluation of the dune field, they are sufficiently complex to provide valuable feedback for scientists.

3.10.1 Dune Field Orientation Computation

One may argue that the orientation of the dune field could be simply computed in the same manner as explained in section 3.4. However, a more generalized approach is preferred which computes the metrics from any set of two dimensional points, regardless of the input image.

In order to compute the average dune field orientation from the detected points, the data must be formated appropriately. The detected crest-lines are typically a set of two dimensional points on the image. These points form lines and segments of various shapes and lengths. The simplest way to compute the orientation is to fit lines to each segments and compute the average orientation of the lines.

There are many commonly used methods to fit lines to segments. One popular method is the Hough Transform initially proposed in [45, 46, 89], which searches a space of parameters voting for segments which contribute to the line. The Hough transform can be generalized to many shapes suchs as circles and ellipses, but the parameter space gets much larger for complex shapes. The main drawback of the Hough transform is the computational cost of search a large parameter space.

An alternative method is the Ramer-Douglas-Peucker algorithm, proposed in [91, 19, 43], with the implementation provided within the OpenCV ([7]) framework. The method is used for simplifying or approximating any given curve to a subset of points. Thus, the dune segments can be reduced to subset of lines. The only parameter required is the allowable error in the approximation. Larger values allow more deviation, producing longer coarse approximations of the segments, while smaller values approximate the curve more closely, producing short segments which are sensitive to noise. The parameter is chosen to produce optimal results for the application.

Another critical requirement for this algorithm to work is the fact that each segment must be a single contiguous line or curve. This means that any segments extracted from the connected components algorithm ([101, 73]) must be sorted and organized in a list to ensure contiguous segments of points. All forks and bifurcations are detected and split into separate segments. The labeling of segment is demonstrated on a test image shown in Figure 37. The results in Figure 37(b) show that these segments can be extracted and split into many single line segments. Although the algorithm may not produce optimal results, the output is sufficient for this application.

Once the segments have been extracted and sorted, the line fitting algorithm presented in [91] can be applied. The result is a set of lines, which can be represented as vectors contain the position and orientation of each line. The orientation components are used to compute the average orientation of the dune field.

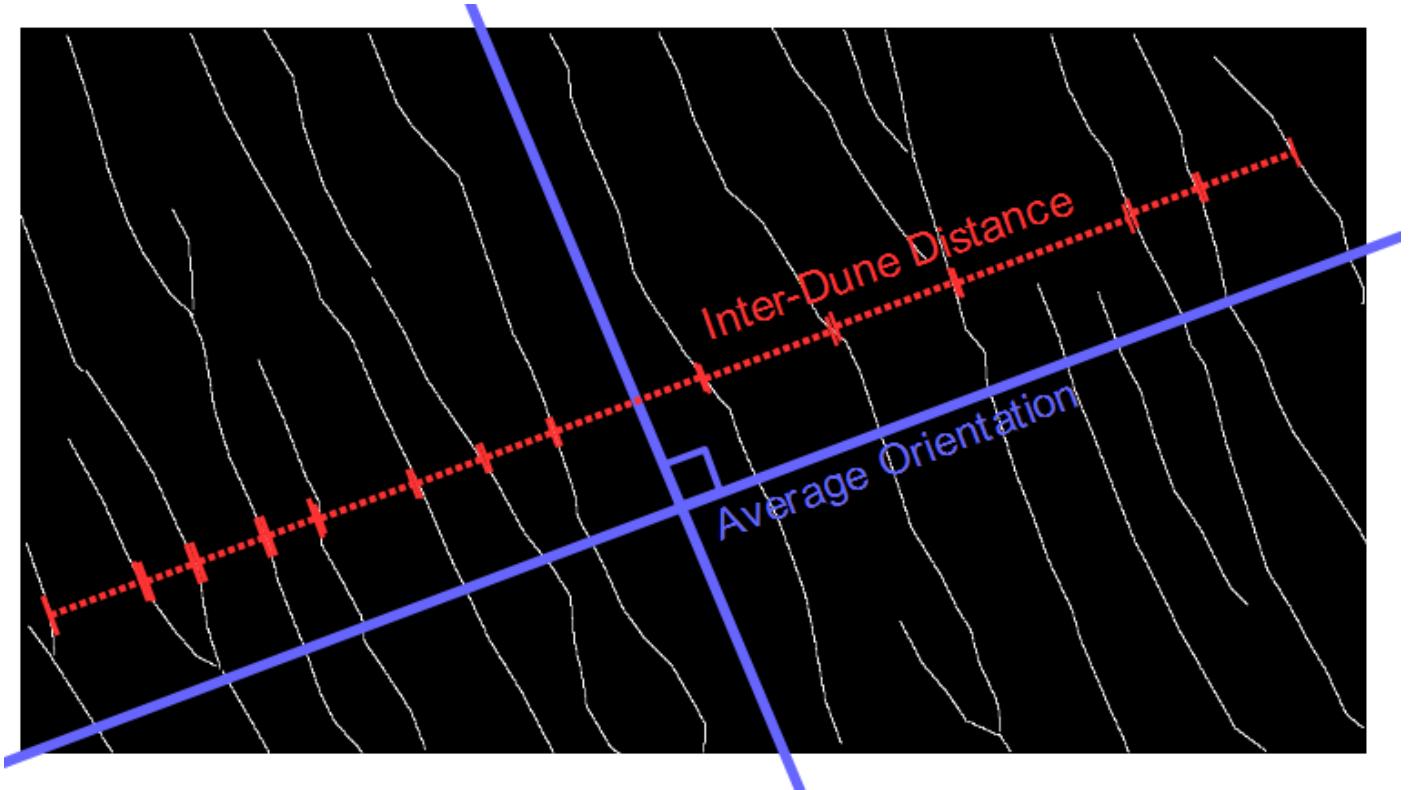


Figure 38: Computing the inter-dune distance using the average orientation. Once the average orientation has been computed, the inter-dune distance can be computed by measuring the distance between segments along an orthogonal axis to the average orientation.

As explained in [81], computing the average orientation of vectors is not as trivial as simply averaging the orientations. To illustrate the problem, the average of an angle of 350° and 10° would be 180° , when in fact a more appropriate average should be 0° . Averaging may produce an adequate approximation, but the von Mises distribution based method proposed in [81] typically produces much more accurate results. Therefore, this technique is used to compute the average orientation of the lines.

With the orientation of the dune field computed, the average dune distance can then be computed.

3.10.2 Inter-Dune Distance Computation

Computing the distance between crest-lines is a feature interesting to many researchers in the field. For simplicity, the inter-dune spacing is computed for the entire dune field image. The resulting measurement is in pixels in our case, but, if the scale of the satellite image is known (meters per pixels), then the measurement can be easily converted to a real world measurement. There are many ways to compute the inter-dune distance, but in this approach the simplest way was used, shown in Figure 38.

Using the binary image of the extracted dune crest-lines, and the average orientation computed, an iterative approach is used to measure the distance between the dunes. A search is done at each pixel along an orthogonal axis from the average orientation. When a crest-line is hit, the number of pixels traveled are counted until another dune is found. The process is repeated for the entire binary image. The lengths are averaged resulting in the average dune spacing for the entire dune field.

The results produced may not be optimal and accurate but can be used as an indicator to evaluate the effectiveness of the dune crest-line detections. The results and metrics are presented in section 4.3.

4 Experimental Evaluation

In this section, the experiments and results are presented. First, the datasets used to conduct the experiments and test the crest-line detection methods is presented in section 4.1. Two main datasets were used for the study:

Terrestrial Dataset A sample dataset of various regions created and labeled by an expert in the field, specifically for this research.

Mars Dataset The dataset from the Ganges Crater area on Mars, defined by [17].

The next section discusses the evaluation used for measuring the quality of each dune crest-line detector method. The main metrics used are the Precision, Recall, and dune features presented in section 3.10. Finally, section 4.3 presents the results along with a short discussion of the methods.

4.1 Study Areas

As part of this research project, two datasets were used in order to evaluate the robustness of each method. The datasets include a wide range of dune types with varying morphological properties.

The data provided in each set are satellite images of dune field regions available through Google Earth and NASA datasets. Included with the images is the ground truth which has been manually labeled by experts in the research field. The ground truth consists of crest-lines for each positive dune detection.

In each case, the scales of the satellite images were chosen such that the inter-dune distance appeared roughly normalized across the entire dataset. Scale selection of the images is important process for even comparison of the methods for different regions. The images are resized to an appropriate resolution for processing and even across the entire dataset.

The method described in this paper was tested on two distinct datasets: an terrestrial dataset which includes dune fields of various regions on Earth, along with the dataset provided in [17] which is located on Mars in the Ganges crater region. The same method was applied on both datasets.

4.1.1 Terrestrial Dataset

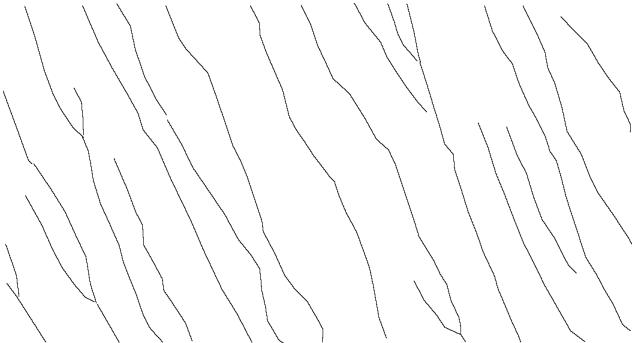
The first dataset is a small sample of a dozen satellite images from six separate desert regions on Earth (shown in Figure 39). Included are the Kalahari, Namib and Skeleton Coast sand sea regions in Namibia ([36]). Also represented in this dataset are the Simpson dune field in Australia, the Winnemucca Dune Complex in Nevada, and the White Sands National Monument. A wide range of landforms types are contained within each of these regions which provides a broad study for an automated crest-line detection method.

The Kalahari (Figure 39(a))sands span from the southeastern region of Namibia to South Africa. The region is a 100-200 km wide and is composed of mostly fixed dunes [60]. Most of the area is comprised of simple linear form dunes, although some small areas contain some compound linear dunes as well. Unlike other desert regions, the Kalahari contains areas which are well vegetated. On average, the dunes range from 2m to 15m in height, with a 150m to 250m width, and spaced from 200m to 240m. These measurements may be useful for determining the reliability of our metric calculations.

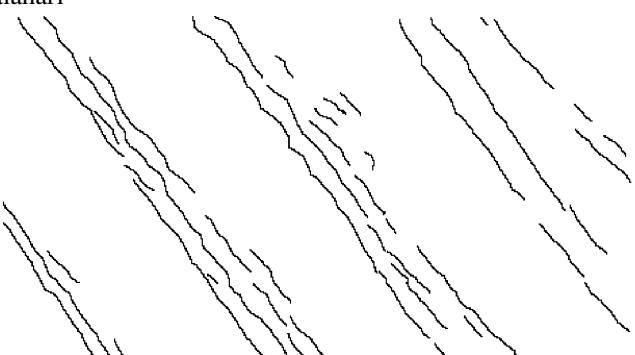
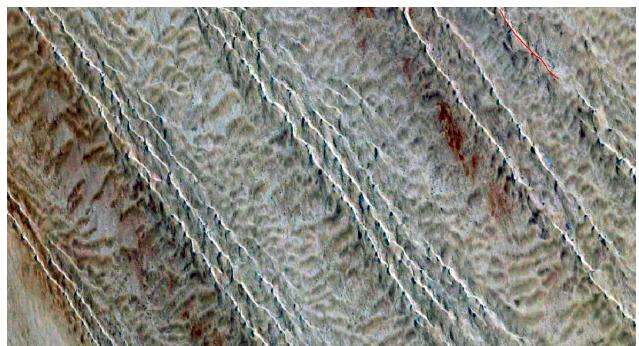
The Namib Sand Sea (Figure 39(b)) region spans approximately 34,000 km of the Altantic coast of Namibia, contains some of the largest and oldest sand dunes in the world according to [37]. High energy unimodal, bimodal and complex wind regimes create interesting dune field patterns in the Namib Sand Sea region of Namibia [59]. These wind patterns characterize the spatial variability of the dune types, sizes, and other morphological properties of the region, making it an interesting case study for this research.

The Skeleton Coast (Figure 39(d)) dune field contains simple, locally compound, transverse and barchanoid dunes over its 2000 km² span according to [58]. The dunes pattern in this region are formed due to onshore winds and surface roughness changes between the dunes and coastal plains. The dune field is roughly aligned with the coast and is characterized with a large slip face in which dunes range from 20m to 80m in height.

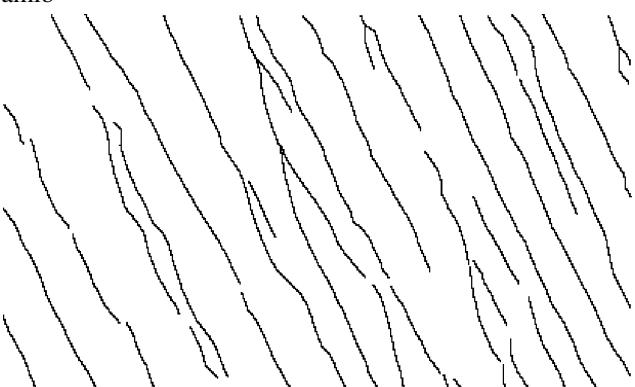
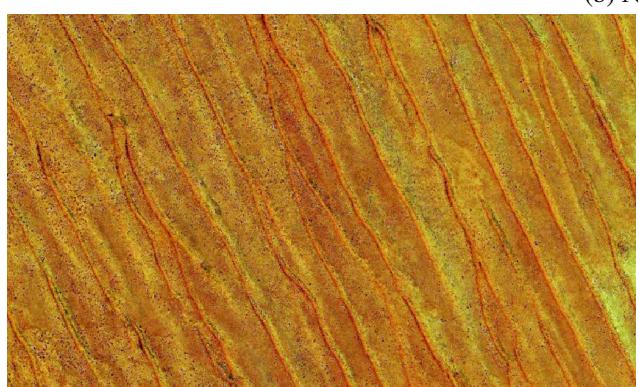
Another region represented in the dataset is the Simpson (Figure 39(c)) dune field in Australia. Much like the Kalahari sands in southern Africa, the Simpson dune fields contains many similar features. The areas are home to lush vegetation, and the dune field follow a mostly simple linear pattern where the dunes tend to be broad crested [44]. According to [115], some of the ridges continue unbroken for up to 200 km, where each crest measures 15m to 38m in height. The spacing between each crest varies depending on the height of the ridges. Areas with larger ridges may have one or two dunes per kilometer, while smaller ridges may have five or six dunes per kilometer. These factor make this area an interesting addition to the dataset because the scale of the images is much larger.



(a) Kalahari



(b) Namib



(c) Simpson

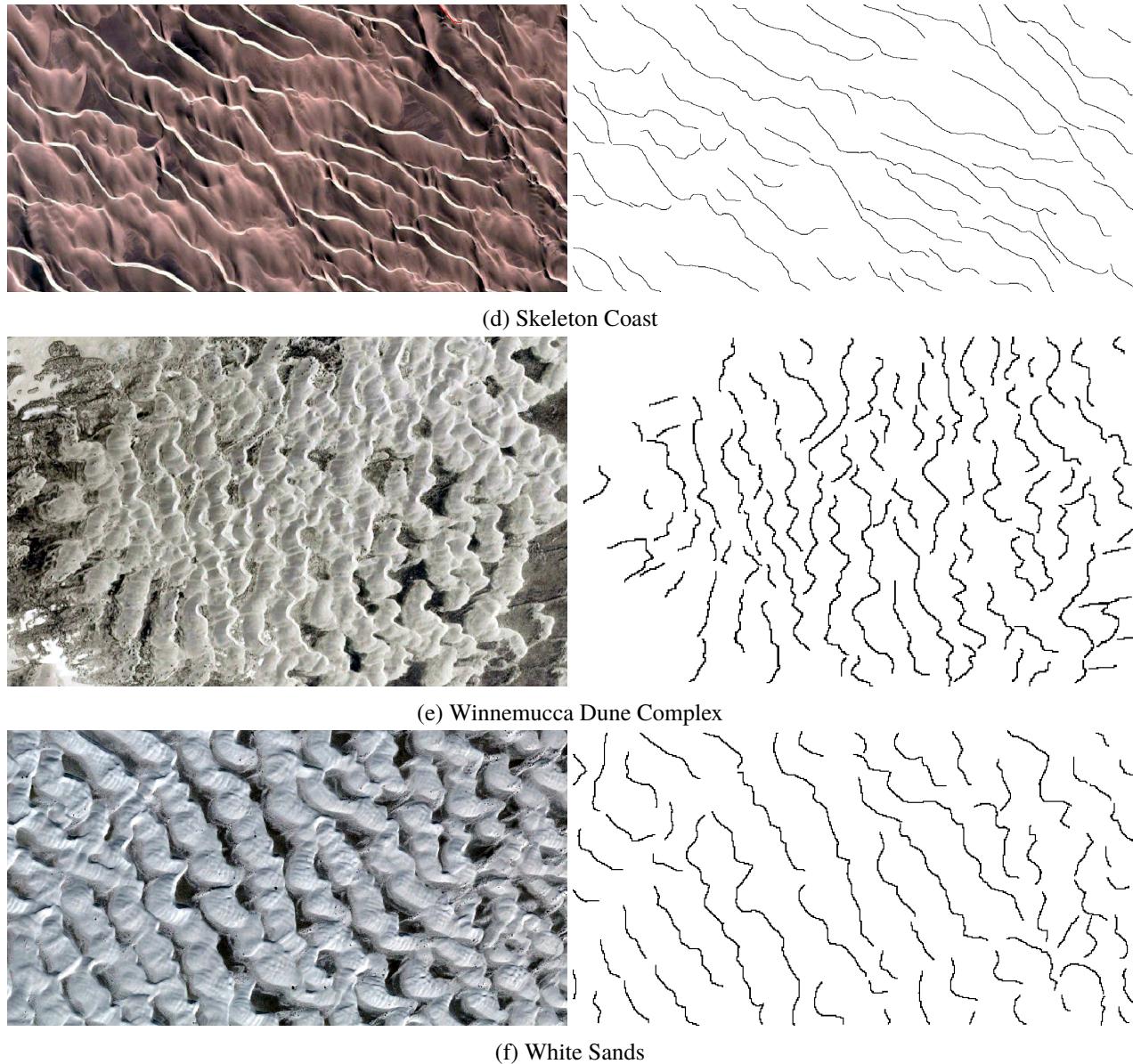


Figure 39: Terrestrial dataset of six regions with respective labeled ground truth: (39(a)) Kalahari (Namibia), (39(b)) Namib (Namibia), (39(c)) Simpson (Australia), (39(d)) Skeleton Coast (Namibia), (39(e)) Winnemucca Dune Complex (USA), and (39(f)) White Sands (USA)

Also present is the Winnemucca Dune Complex (WDC, Figure 39(e)) found in the western United States, in Nevada. The WDC covers an area of roughly 900 km^2 north of Winnemucca, Nevada. The most common dune type present are stabilized parabolic dunes, but barchans and transverse ridges can also be found scattered throughout the area [51]. In fact, according to [85], the WDC is primarily covered by six crescentic complexes, a large sand sheet, and discontinuous sets of compound barchanobolic-parabolic dune fields. The WDC contains a complex set of repetitive sequences of dunes which varying shapes and scales, which makes it an ideal candidate for this dataset.

Finally, the last area in the dataset is the White Sands National Monument (fig. 39(f)), located in the state of New Mexico, USA. This area boasts an interesting pattern of crescentic aeolian dunes which are formed in a systematically similar fashion to wind ripples and sub-aqueous dunes [98]. There are a wide range of features and properties in the White Sands dune field that merit study, such as described in [98]. These interactions include merging, lateral linking, defect repulsion, bed-form repulsion, off-center collision, defect creation and dune splitting. The details of these interactions are outside the scope of this research but crest-line detection may be an essential preliminary step towards extracting those features. Measuring the number of dunes, crest lengths, defect density, dune spacing, and dune height are all done manually by experts in the field. A move towards an automated process would greatly improve research efforts, and would be a helpful tool for scientist in the field.

4.1.2 Mars Dataset

To verify the robustness of this approach, the method was also tested on another dataset which was used in [17]. The dataset provided is from the Ganges Chasms on Mars, and includes satellite images, manually labeled crest-lines, and the results of the original author's method. The provided results lay a baseline benchmark for measuring quality and accuracy of crest-line detection algorithms. In order to more easily process the dataset, the Ganges region was split into sixteen areas of equal size, as shown in Figure 40. The ground truth included was not validated for correctness, the labeled data was used as-is, which may account for some errors (see Section 4.3).

The dataset is essentially a CTX mosaic of the Ganges chasma, which spans an area of 500 km^2 , and includes a wide variety of dune types and morphologies. According to [56], many aeolian fatures can be found in the Ganges Chasma. Sand sheets, dune fields, unidirectional features such as barchan dunes were all identified within this region. The overall structure of the Ganges dune field is a complex set of many diverging dunes, which makes it a challenging and appropriate area for testing this method.

4.2 Evaluation

For each method, a number of metrics were used to evaluate the quality of each dune detector. The results of interest are as follows:

True Positives A true positive (TP) is defined as a positively detected crest-lines. A detection is determined to be positive if it contains a ground truth within a radius of ϵ .

False Positives A false positive (FP) is defined to be a detection which is not an actual dune crest-line. A false positive occurs when a detection does not have a ground truth crest-line within a radius of ϵ .

False Negative A false negative (FN) is an actual ground truth crest-line which was no identified. A false negative occurs when no detection is found within a radius of ϵ .

Precision Precision is the ratio of correct detection to overall detection. It measures how precise the detections are. It can be computed from the true positives and false positives by $P = TP / (TP + FP)$.

Recall Recall is the percentage of crest-lines identified. It can be computed from the true positives and false negatives using $R = TP / (TP + FN)$.

Angular Error The difference between the dune field average orientation computed from the detected crest-lines, and the average orientation computed from the ground truth crest-lines (in degrees), represented by Δ_θ .

Inter-Dune Distance Error The difference between the dune distance computed from the detected crest-lines and the distance computed from the ground truth crest-lines (in pixels), represented by Δ_d .

In order to compute the results, some important considerations must be made in regard of determining what makes a detection a true positive or a false positive. An important point to note is that typically, in applications where objects are detected, each detection is matched to a single ground truth. Normally, each ground truth object is matched to each detection, and the closest match is marked. Once a detection is marked, it cannot be matched to any other ground truth.

In this application and in the results presented, this rule was violated for a few reasons:

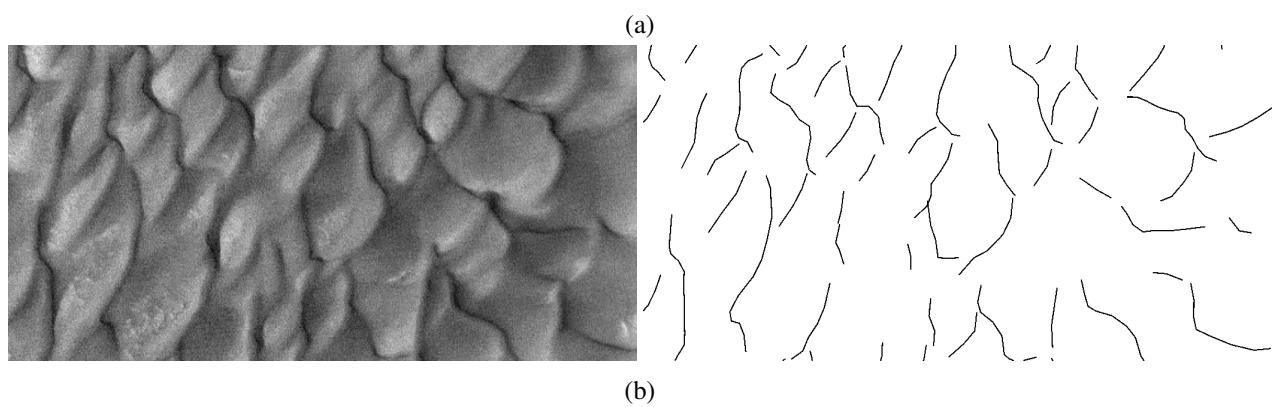
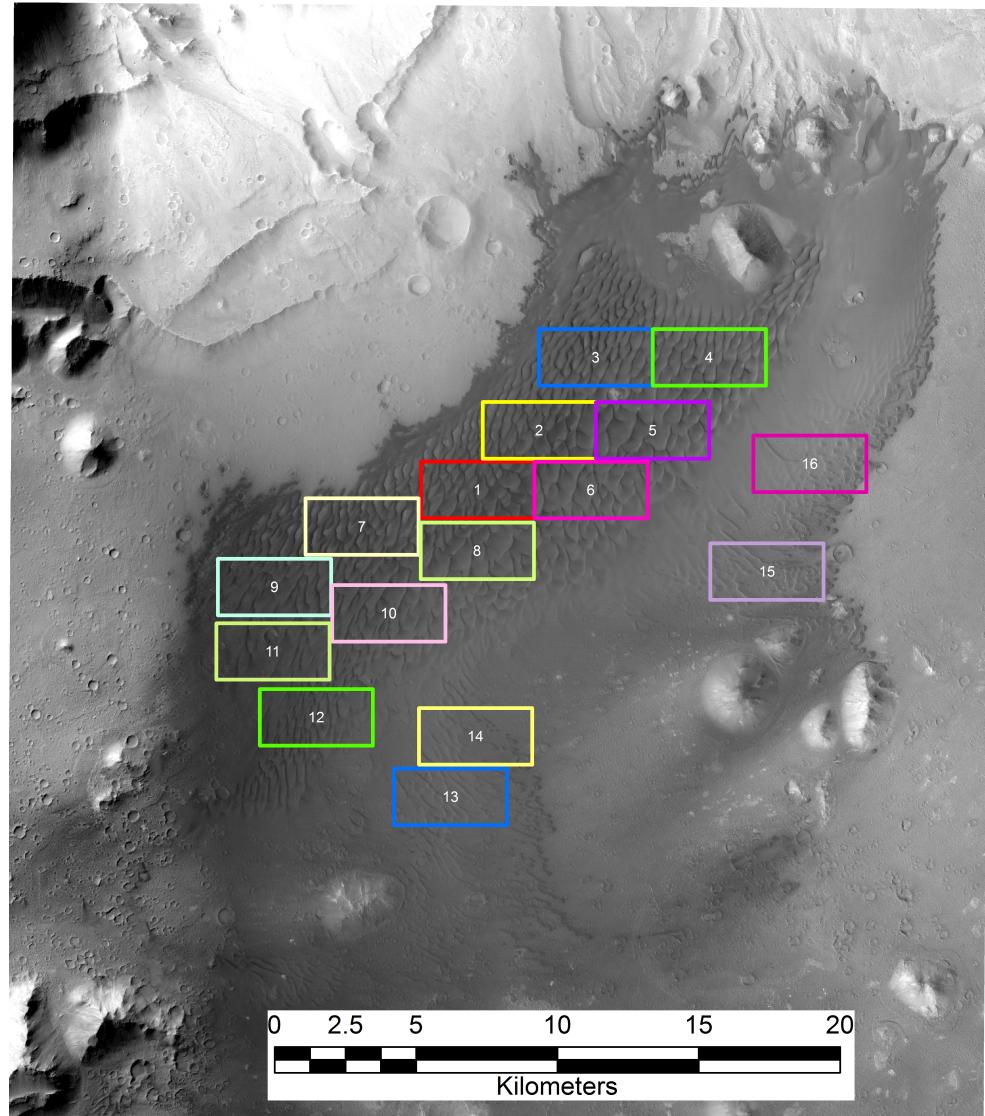


Figure 40: Ganges Chasms Mars dataset from [17] which includes (40(a)) 16 regions extracted, (40(b)) sample image from region 1 with corresponding labeled ground truth

Table 1: Precision and recall results of the appearance-based approach described in section 3.3 for the Terrestrial dataset.

Dataset	Precision	Recall
Kalahari	0.7651	0.7094
Namib	0.5543	0.7493
Simpson	0.1732	0.1364
Skeleton Coast	0.7117	0.8033
WDC	0.5993	0.5581
White Sands	0.2740	0.2614

- The ground truth and detected crest-lines are segments of 2D points or pixels. In some cases, the ground truth may not be single pixel-wide segments, meaning the expected output is thicker than the detected segments. The output is that some of the ground truth pixels do not get matched with any detections, resulting in inaccurate False Negative rates.
- The detections are also segments of 2D points which form a line which may not match the exact shape of the ground truth. Although it is close to the ground truth, the number of pixels in a given detection segment may not match the number of points found in the ground truth, resulting in loss of accuracy.
- Errors in the ground truth are common in each dataset. Even though the datasets were labeled by experts, they were in fact manually drawn and are not perfect. In many cases, some crest-lines are either missing, or offset by a few pixels from the actual crest-lines.

For all these reasons, the standard approach to generating results is omitted in favor of a simpler way. The approach used is to search in a window of radius ϵ around each detection and each ground truth points. If a ground truth is found in the window around a detection, that detection is marked as a true positive. On the other hand, if no ground truth is found in that window, it is a false positive. Finally, if no detections are found within the ϵ window around a ground truth, that ground truth is marked as a false negative.

Determining an appropriate value for ϵ affects the overall results as well. Based on the datasets used in this research, and accounting for error in labeling and image resolution, an appropriate value was chosen to be $\epsilon = 10$ (pixels). If the scale of the image is known (meters per pixel), pixels can be converted to a more meaningful metric scale.

For each method the results are computed using the dataset presented in section 4.1. For the machine learning methods, the terrestrial dataset was extended with test images. Proper evaluation of machine learning method require test images in order to test the model on samples not trained on.

Additionally, the terrestrial dataset includes many representative regions with various dune types. There are two options for training models:

Unique Model Per Region In this approach, a machine learning model is trained for each region. The result are more accurate within each regions, but the solution is not as generalized.

Cross-Region Model In this approach, a single model is trained using all regions. The results produced are typically less accurate than the unique model per region approach, but provides a more generalized solution.

Another dataset evaluated was the Mars dataset from section 4.1.2. Included are the results from the research done in [17], which includes the ground truth and results. Our machine learning method was compared side-by-side with the results from [17] on each of the 16 regions sampled. The 16 regions are split half and half for training and testing respectively, where odd regions were used for training and even regions for testing.

4.3 Results

In this section, the results for each approach are presented, which includes the appearance-based, edge-based, Watershed Transform, frequency domain, gradient orientation-based, machine learning, and final mixed methods.

Appearance-Based Approach Results

The precision and recall results for the appearance-based approach explained in section 3.3 are presented in Table 1. The results are quite poor but are a good basis starting point for evaluation of dune detectors.

Table 2: Precision and recall results of the edge-based line detection approach described in section 3.4 for the Terrestrial dataset. The results are shown for both the K-Means (section 3.4.1) and the histogram of gradients (section 3.4.2) approaches to computing the dominant orientation.

Dataset	K-Means		HoG	
	Precision	Recall	Precision	Recall
Kalahari	0.7902	0.9396	0.5793	0.9852
Namib	0.6534	0.9281	0.5842	0.9626
Simpson	0.5999	0.7464	0.5808	0.8977
Skeleton Coast	0.9122	0.7605	0.7899	0.8187
WDC	0.5132	0.5111	0.5807	0.7478
White Sands	0.5681	0.6231	0.5708	0.7634

Table 3: Precision and recall results of the watershed transform segmentation approach described in section 3.5 for the Terrestrial dataset.

Dataset	Precision	Recall
Kalahari	0.9087	0.8093
Namib	0.6115	0.6617
Simpson	0.6992	0.4893
Skeleton Coast	0.5700	0.7489
WDC	0.6279	0.2532
White Sands	0.4763	0.4068

Edge-Based Approach Results

The results for the edge-based line detection approach discussed in 3.4 are shown in Table 2. The approach proposed implemented two methods to compute the dominant orientation, using K-Means (section 3.4.1) and using the histograms of gradients (section 3.4.2), therefore results are shown for both are shown in the table. The HoG method generally provides higher recall, while the K-Means generally produces higher precision. However, the results of this approach are better overall than the appearance-based approach.

Watershed Transform Approach Results

Table 3 shows the results of the watershed transform segmentation approach described in section 3.5. The results of this approach are better than the appearance based approach, but still are quite poor.

Frequency Domain Approach Results

The frequency domain approach presented in section 3.6 also has somewhat poor results, but does have a slightly higher overall recall rate. The results of this approach are shown in Table 4.

Gradient Orientation-Based Approach Results

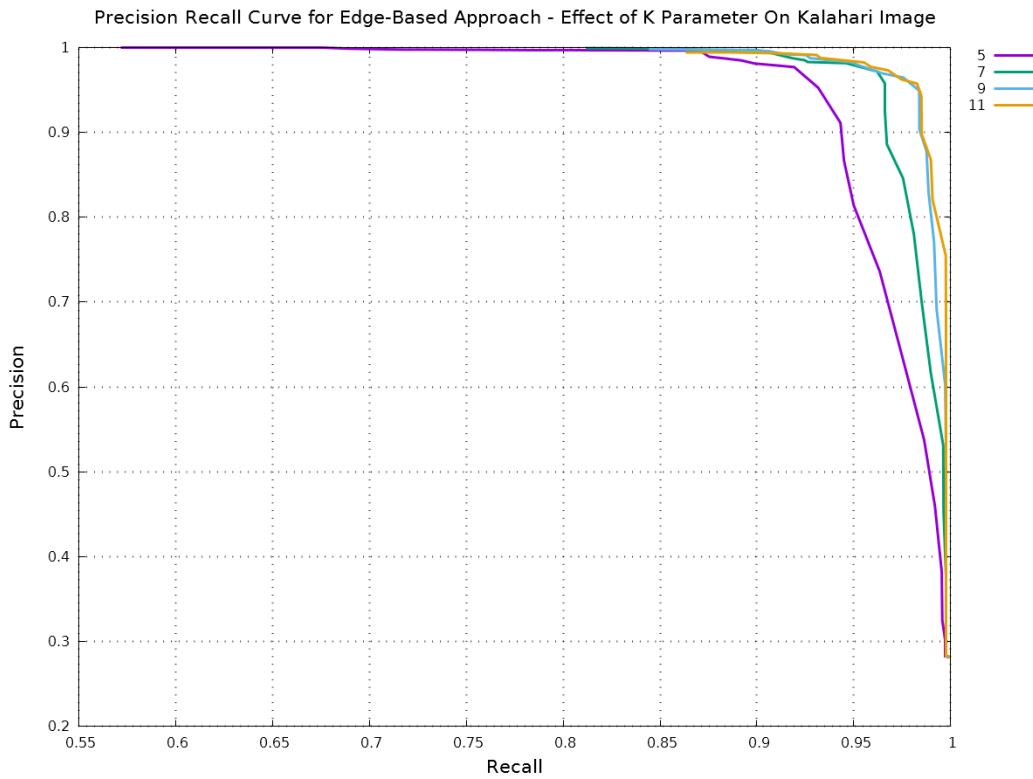
The gradient orientation-based approach presented in section 3.7 is a much better improvement overall, as shown in Table 5. The precision and recall rates are better than the previous methods, but there still appears to be issues with the *Simpson*

Table 4: Precision and recall results of the frequency domain based approach described in section 3.5 for the Terrestrial dataset.

Dataset	Precision	Recall
Kalahari	0.6329	0.9730
Namib	0.5835	0.9811
Simpson	0.6450	0.6857
Skeleton Coast	0.6448	0.7767
WDC	0.7221	0.6571
White Sands	0.4703	0.3601

Table 5: Precision and recall results of the gradient orientation approach described in section 3.7 for the Terrestrial dataset.

Dataset	Precision	Recall
Kalahari	0.7181	0.9370
Namib	0.6297	0.9883
Simpson	0.4604	0.5697
Skeleton Coast	0.7697	0.9935
WDC	0.7589	0.9302
White Sands	0.5004	0.6429

Figure 41: Precision and Recall curves of the Gradient Orientation based approach on the Kalahari image for various values of K (the size of the gradient kernel).

(linear dunes) and *White Sands* (complex dunes). Figure 42 shows the results of the approach projected on the Skeleton Coast and WDC images. The results clearly smooth contiguous crest-line segments on both images.

An important aspect to note with the gradient orientation-based approach is the selection of the kernel size for computing the gradients. In Figure 25, we showed that the size of the gradient kernel K had a significant impact on the results. To illustrate, Figure 41 shows the precision/recall curves results for various values of K .

Machine Learning Approach Results

Finally, the results for the machine learning approaches are presented. To recap, the machine learning approaches require the training of the models using a set of data and the model is then tested on a separate set of data. Therefore, a separate set of data was created to evaluate the performance of the machine learning approach, shown in Figure 43.

The addition of the test images allows us to better evaluate the effectiveness of the various models and methods used to train the machine learning system. To train each model a set of data is collected from the both the training and test sets using the method described in section 3.8.1. A set of pixels are randomly selected for each image, preserving a equal count of crest-line pixels and non-crest-line pixels. Crest-line pixels are determined using the provided ground truth. The number of pixels extracted was varied to determine the effectiveness of the model. When fewer data was used, the model could be trained very quickly but the representation of the model achieved poorer results. A larger training set provided a much better representation of crest-lines, at the cost of efficiency. The number of pixels used found to be optimal was in the range of 1000 to 5000.

As described in section 3.8.2, the features were extracted at each sampled pixel. Many feature types were used including BRIEF, ORB, BRISK, SURF, HOG, LBPH, and even using raw pixel values but the SIFT performance was far

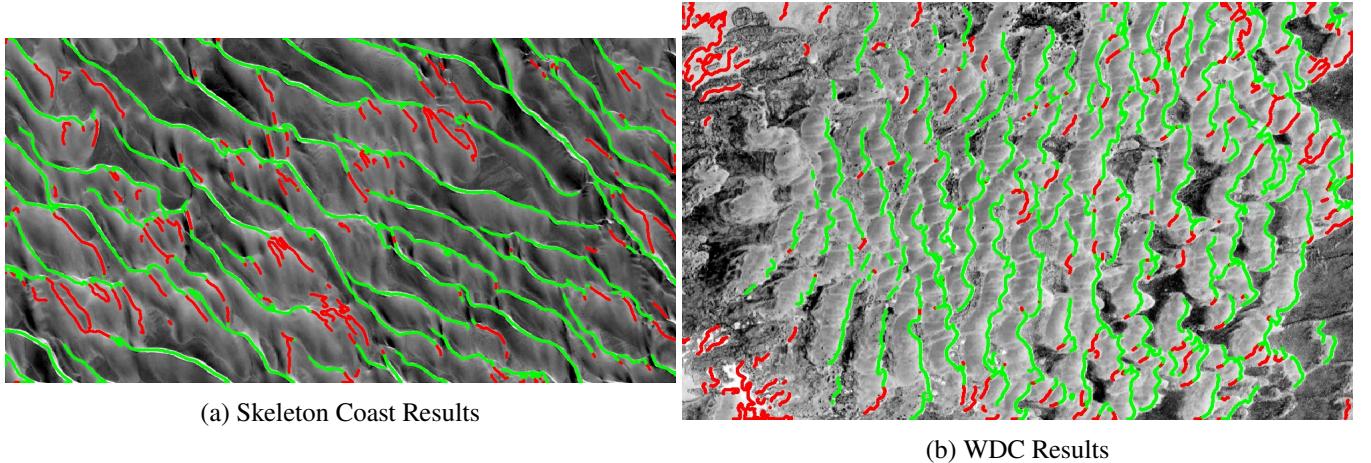


Figure 42: Results of the gradient based approach projected on the image for the Skeleton Coast and WDC images. The true positives are shown in green and the false positives are shown in red. The overall precision and recall results are shown in Table 5.

superior to all others. The traditional 128 bin SIFT descriptor vector was extracted at each pixel as the feature to use for the model.

The next consideration is the type of machine learning classifier to use for the model. Again, as described in section 3.8.3, there are many classifier options available such as Bayesian, Neural Networks, Decision Trees, Support Vector Machines.

In Table 6 shows the results of the training process. The results show that SVM and the decision trees perform the best. The Normal Bayes classifier has a tendency to over-fit the training data. This is evident by the fact that it learns the training data very well but has poor performance on the test set. Overall, the Gradient Boosted Tree results shown in Table 6d has the best results.

Next, the machine learning method results (using Gradient Boosted Trees on the extracted SIFT descriptors) explained in section 3.8 are discussed. The method was applied in two experiments:

1. Training one model for each region: For each region, training data and test data was collected, and a single model is created and trained for that specific region. The results of this approach are shown in Table 8.
2. Cross-Region model: A single model is used for all the regions. The training data was collected from all regions, the model was trained using the features from all the regions. The results of this approach are shown in Table 7

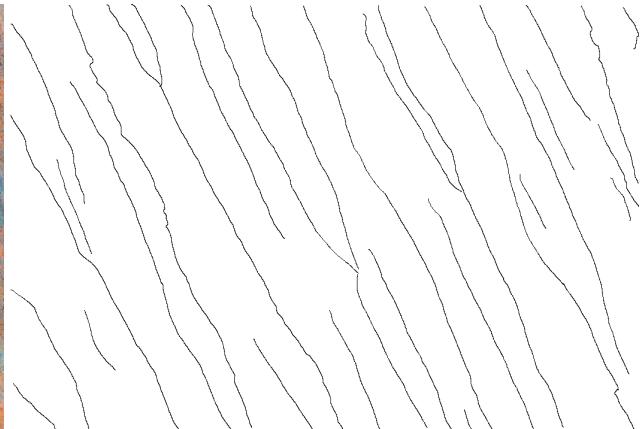
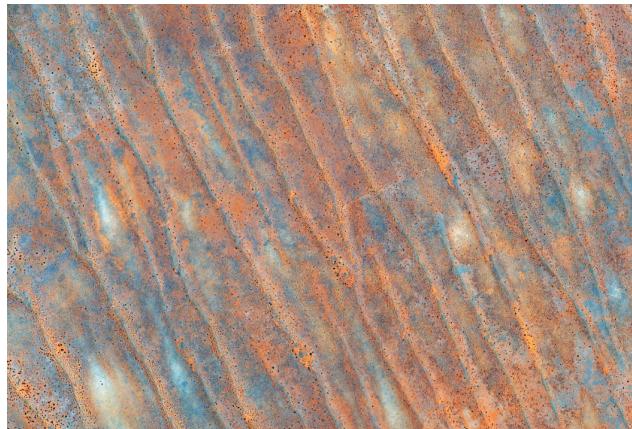
The results clearly show that the cross-region model struggles at learning the general concept of a dune across many regions. The precision and recall values of the single model per region approach shown in Table 7a are measurably better than the cross-region approach results shown in Table 9a. The difference is especially large with complex dune structures such as the Winnemucca Dune Complex and the White Sands regions. The dune metrics computed in Tables 7b and 9b are also much worse, since the calculation of these metrics heavily depends on accurate detection of crest-lines.

The results of the machine learning crest-line detection are shown in Figure 44. The true positive detections are shown in green, along with the correctly identified ground truth in blue. The false negatives (crest-lines not correctly detected) are shown in yellow, and false positives are shown in red. The results show that the overall performance of this approach is quite good.

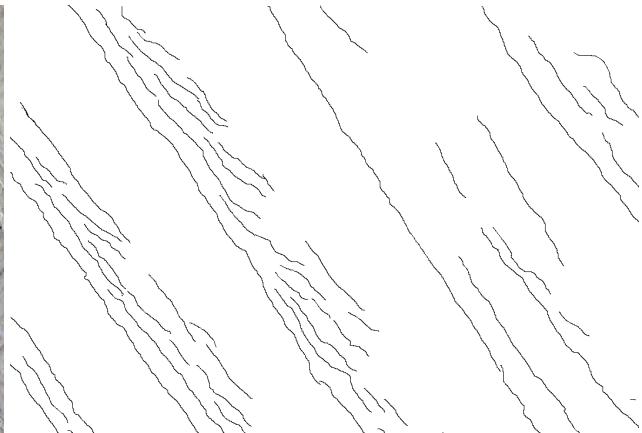
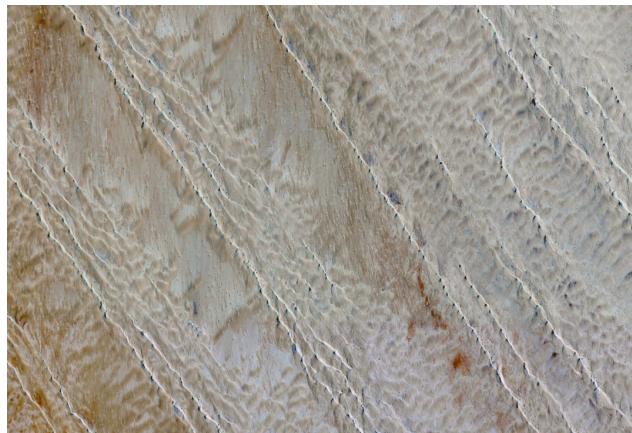
One important point to note is that there are inaccuracies in the ground truth itself. In many cases, some false positives may not actually be false positives because some crest-lines were not labeled in the ground truth. Additionally, some false negatives may be caused by poor localization of the ground truth labels. For the most part, the ground truth is fairly accurate, but a second pass at labeling would increase the overall precision and recall rates results.

Mixed Machine Learning and Gradient Orientation Approach Results

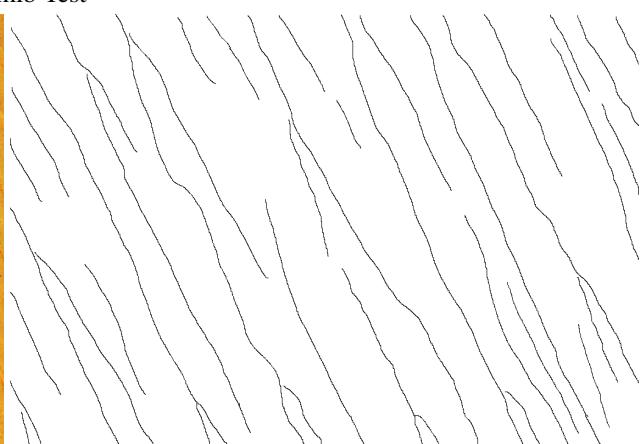
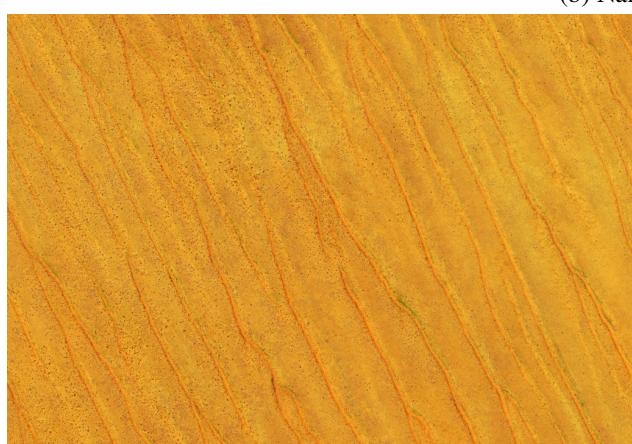
The final results presented are for the mixed machine learning and gradient orientation approach presented in section 3.9. The approach essentially combines the methods from sections 3.7 and 3.8 to yield improved results. Similarly to the previous method, the machine learning model in this approach can also be trained for each region or across all the regions. The results of these methods are presented in Tables 10 and 11. The results projected onto the images are shown in Figure 45.



(a) Kalahari Test



(b) Namib Test



(c) Simpson Test

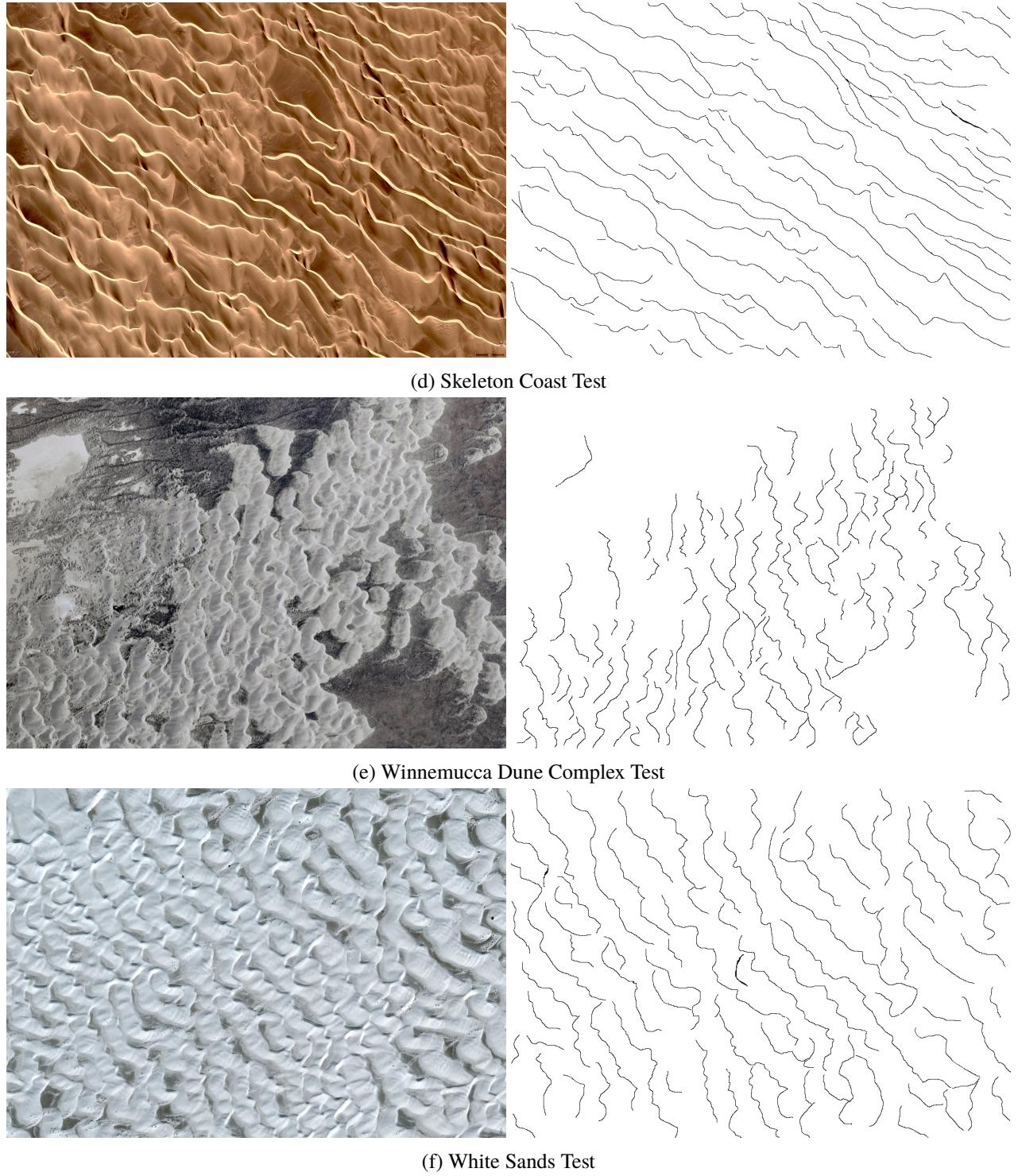


Figure 43: Terrestrial test dataset of six regions with respective labeled ground truth: (43(a)) Kalahari (Namibia), (43(b)) Namib (Namibia), (43(c)) Simpson (Australia), (43(d)) Skeleton Coast (Namibia), (43(e)) Winnemucca Dune Complex (USA), and (43(f)) White Sands (USA), used for evaluation of machine learning approaches. Test

Table 6: True positive and false positive rates results of the training process to build the crest-line classifier models as described in section 3.8.3, using the SIFT descriptors for various classifiers (a) Support Vector Machines, (b) Normal Bayes, (c) Random Trees, (d) Gradient Boosted Trees

SVM Results	Training Set		Test Set	
	TPR	FPR	TPR	FPR
Kalahari	0.9939	0.0151	0.8703	0.1383
Namib	0.9896	0.0183	0.9054	0.1300
Simpson	0.9959	0.0078	0.7950	0.1584
Skeleton Coast	0.9917	0.0195	0.8031	0.1393
WDC	1.0000	0.0018	0.6421	0.1874
White Sands	0.9913	0.0099	0.7081	0.1495

(a) Support Vector Machine Results

Normal Bayes Results	Training Set		Test Set	
	TPR	FPR	TPR	FPR
Kalahari	1.0000	0.0491	0.4440	0.0468
Namib	0.9913	0.0754	0.6359	0.1081
Simpson	0.9937	0.0428	0.6015	0.0823
Skeleton Coast	0.9959	0.1074	0.5714	0.1045
WDC	0.9950	0.0499	0.8511	0.1624
White Sands	0.9978	0.0812	0.5483	0.1111

(b) Normal Bayes Classifier Results

Random Tree Results	Training Set		Test Set	
	TPR	FPR	TPR	FPR
Kalahari	0.9837	0.0981	0.9037	0.1277
Namib	0.9809	0.1181	0.9504	0.1493
Simpson	0.9586	0.0895	0.8317	0.2078
Skeleton Coast	0.9793	0.1191	0.8764	0.1844
WDC	0.8856	0.0942	0.7107	0.2048
White Sands	0.9524	0.0990	0.7955	0.1960

(c) Random Trees Classifier Results

GBT Results	Training Set		Test Set	
	TPR	FPR	TPR	FPR
Kalahari	0.9817	0.0792	0.8900	0.1213
Namib	0.9688	0.1201	0.9267	0.1415
Simpson	0.9400	0.0972	0.8279	0.1975
Skeleton Coast	0.9813	0.1133	0.8764	0.1762
WDC	0.8706	0.0943	0.7090	0.1895
White Sands	0.9351	0.1129	0.7677	0.1636

(d) Gradient Boosted Trees Results

Table 7: Results of the machine learning method presented in section 3.8 using a single Gradient Boosted Tree classifier and the SIFT features trained across all the regions, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.

Datasets	Training Set		Test Set	
	Precision	Recall	Precision	Recall
Kalahari	0.9379	0.9726	0.7694	0.9232
Namib	0.8491	0.9676	0.7867	0.8778
Simpson	0.9084	0.9249	0.7658	0.8398
Skeleton Coast	0.7632	0.9536	0.7938	0.9393
WDC	0.8981	0.8554	0.7558	0.8559
White Sands	0.9370	0.8784	0.8695	0.8001

(a) Machine Learning Precision and Recall Results (GBT-SIFT)

Datasets	Training Set		Test Set	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Kalahari	0.2437°	3.9228	1.5047°	7.6993
Namib	8.0655°	13.2756	2.8757°	4.0135
Simpson	13.0593°	8.3083	0.7329°	2.4516
Skeleton Coast	0.0434°	9.5747	1.4801°	1.5155
WDC	1.9937°	1.9692	2.1001°	1.7801
White Sands	10.6052°	4.3457	2.5835°	7.8096

(b) Dune Metrics Results for Angular Error (Δ_θ) and Inter-Dune Distance Error (Δ_d)

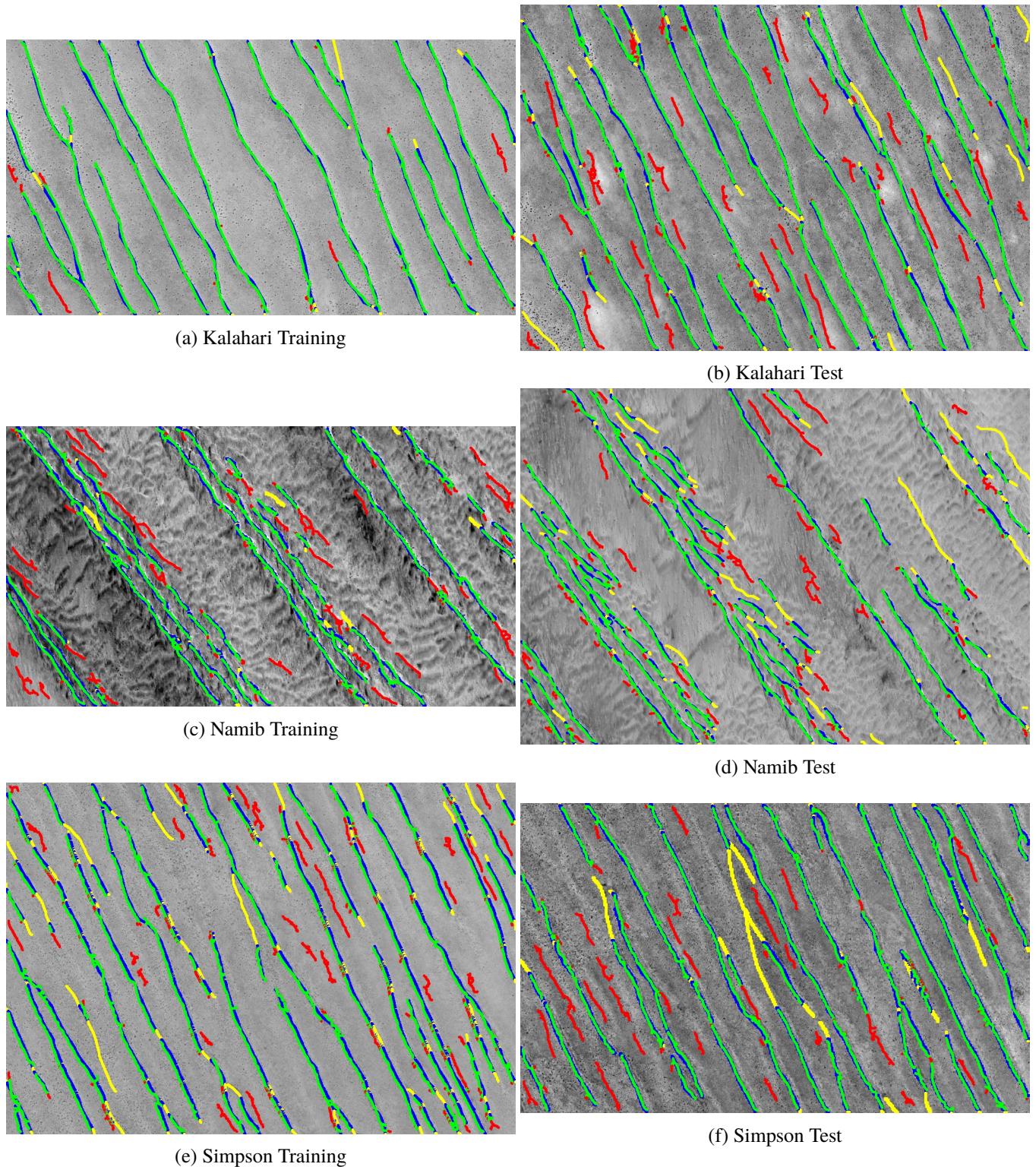
Table 8: Results of the machine learning method presented in section 3.8 using the Gradient Boosted Tree classifier and the SIFT features, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.

Datasets	Training Set		Test Set	
	Precision	Recall	Precision	Recall
Kalahari	0.9258	0.9254	0.7300	0.8444
Namib	0.7942	0.9590	0.6637	0.8634
Simpson	0.8491	0.7854	0.7375	0.7579
Skeleton Coast	0.5768	0.9096	0.6135	0.8967
WDC	0.9315	0.6155	0.8585	0.5771
White Sands	0.9374	0.6431	0.7909	0.3973

(a) Machine Learning Precision and Recall Cross-Region Results (GBT-SIFT)

Datasets	Training Set		Test Set	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Kalahari	0.4803°	5.0142	1.0343°	5.9187
Namib	2.8976°	16.7863	9.5319°	2.4645
Simpson	13.938°	9.0225	0.2434°	2.7176
Skeleton Coast	10.4224°	18.6251	7.7421°	8.3473
WDC	12.7138°	11.1957	15.8434°	21.4145
White Sands	11.0979°	15.9853	1.2744°	53.3311

(b) Dune Metrics Results for Angular Error (Δ_θ) and Inter-Dune Distance Error (Δ_d) for Cross-Region



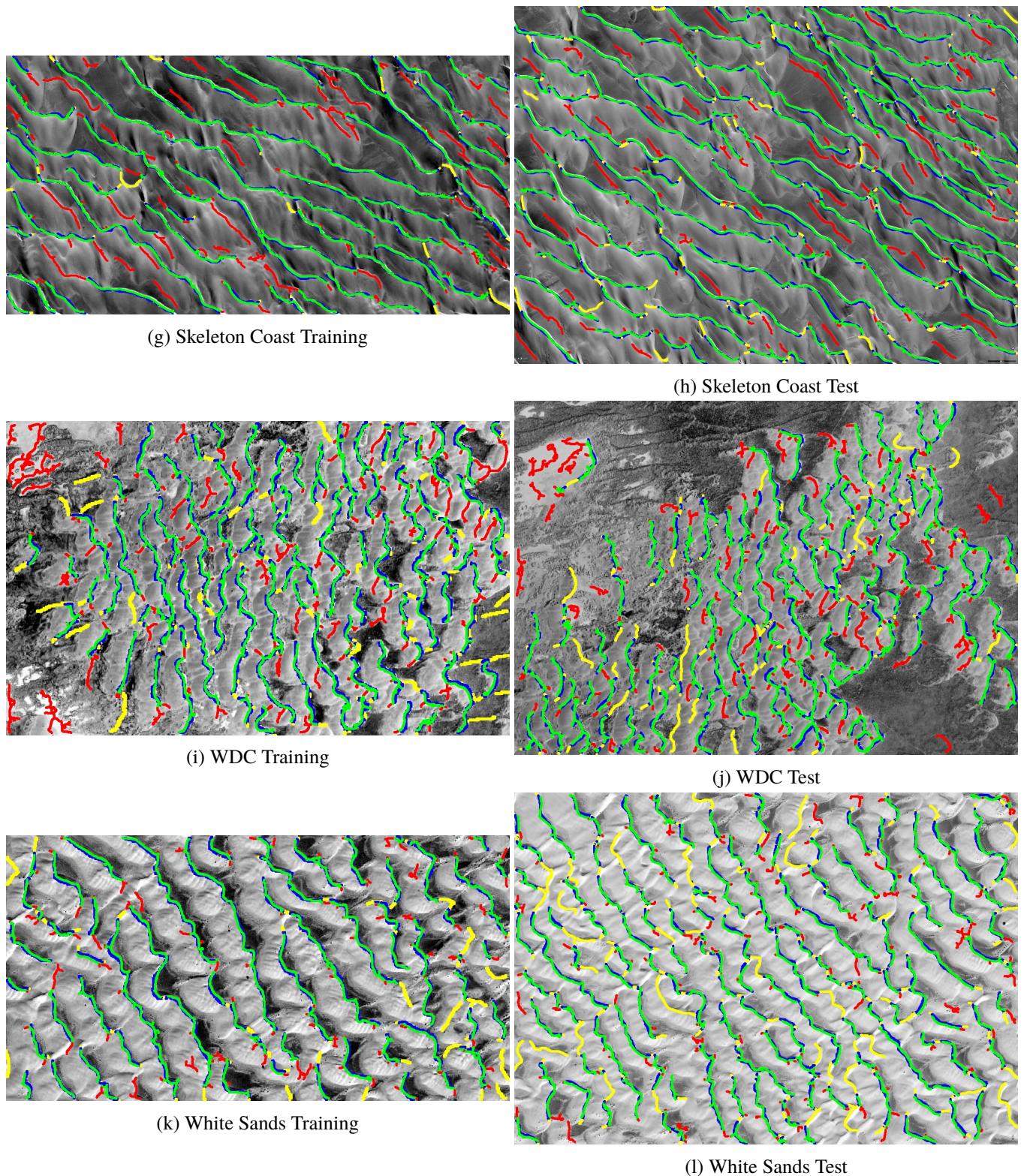


Figure 44: The results of the crest-line detection for the machine learning approach on terrestrial dataset for both the training images (a), (c), (e), (g), (i), (k), and the test images (b), (d), (f), (h), (j), (l) using the GBT classifier and SIFT features. **Legend:** *Green* - True Positives, *Red* - False Positives, *Yellow* - False Negatives, *Blue* - Ground truth correctly detected.

Table 10: Results of the mixed machine learning and gradient-based method presented in section 3.9 using the Gradient Boosted Tree classifier and the SIFT features, includes the Precision and Recall (a) and the Dune Metrics error results (b) for both the training and test image sets.

Datasets	Training Set		Test Set	
	Precision	Recall	Precision	Recall
Kalahari	0.9523	0.9904	0.9003	0.8979
Namib	0.9512	0.9831	0.9366	0.8443
Simpson	0.9542	0.8341	0.9111	0.8335
Skeleton Coast	0.9356	0.9806	0.9829	0.8974
WDC	0.9711	0.7996	0.9145	0.8006
White Sands	0.9789	0.8328	0.9540	0.7948

(a) Mixed ML/Grad Precision and Recall Results (GBT-SIFT)

Datasets	Training Set		Test Set	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Kalahari	0.5643°	8.4518	0.5811°	9.4325
Namib	6.2212°	2.36	3.5316°	19.8735
Simpson	12.5965°	9.8947	1.0102°	5.3046
Skeleton Coast	0.5231°	5.2738	1.9851°	16.7215
WDC	2.6216°	6.4317	4.0554°	6.8127
White Sands	10.2219°	10.2207	17.0219°	14.2709

(b) Dune Metrics Results for Angular Error (Δ_θ) and Inter-Dune Distance Error (Δ_d)

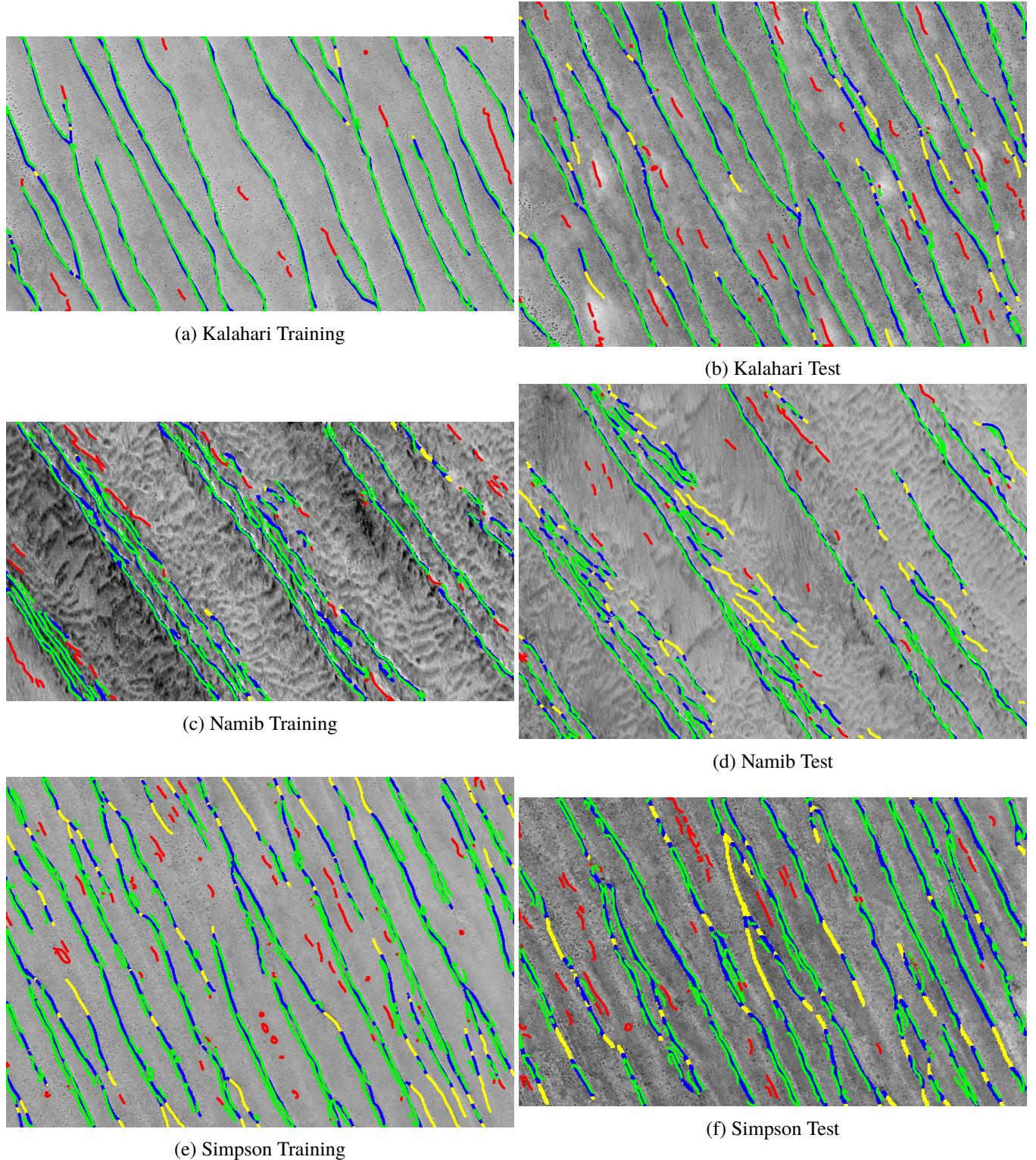
Table 11: Results of the mixed machine learning and gradient-based method presented in section 3.9 using the Gradient Boosted Tree classifier and the SIFT features trained across all the regions, includes the Precision and Recall a and the Dune Metrics error results b for both the training and test image sets.

Datasets	Training Set		Test Set	
	Precision	Recall	Precision	Recall
Kalahari	0.9502	0.9908	0.9007	0.9244
Namib	0.9336	0.9705	0.8926	0.9137
Simpson	0.8833	0.8117	0.7108	0.7943
Skeleton Coast	0.8946	0.9089	0.9284	0.8740
WDC	0.9906	0.4903	0.9780	0.3942
White Sands	1.000	0.3122	0.9930	0.1347

(a) Mixed ML/Grad Precision and Recall Results (GBT-SIFT)

Datasets	Training Set		Test Set	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Kalahari	0.5742°	0.1654	0.1318°	2.1444
Namib	4.9791°	0.9722	5.7731°	11.1356
Simpson	14.0118°	15.2678	0.4280°	14.03
Skeleton Coast	2.8091°	2.6331	1.3421°	11.9770
WDC	7.3042°	45.1448	6.8027°	63.6407
White Sands	11.8688°	74.6094	12.6240°	112.5056

(b) Dune Metrics Results for Angular Error (Δ_θ) and Inter-Dune Distance Error (Δ_d)



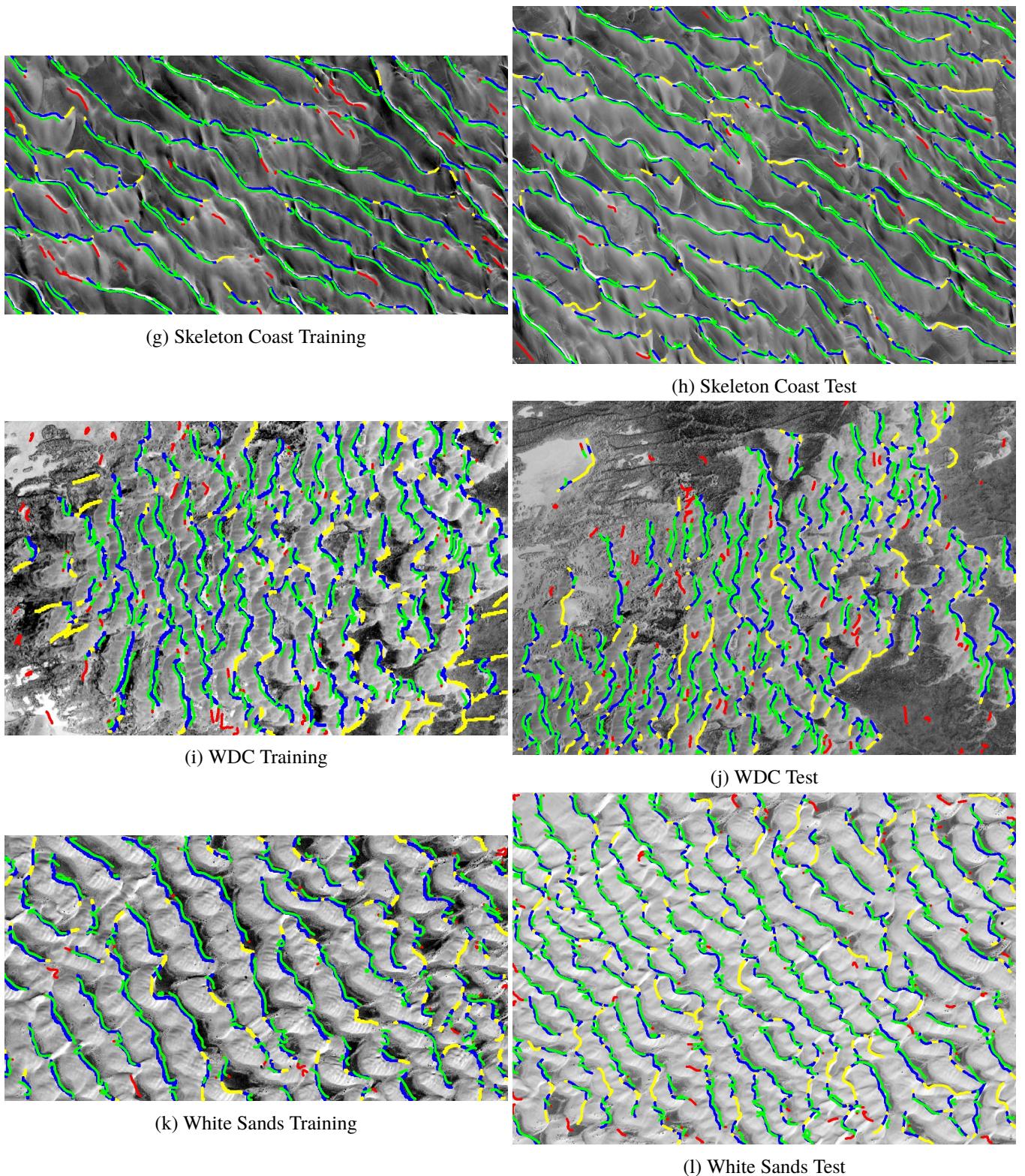


Figure 45: The results of the crest-line detection for the mixed machine learning and gradient approach on terrestrial dataset for both the training images (a), (c), (e), (g), (i), (k), and the test images (b), (d), (f), (h), (j), (l) using the GBT classifier and SIFT features. **Legend:** *Green* - True Positives, *Red* - False Positives, *Yellow* - False Negatives, *Blue* - Ground truth correctly detected.

Table 12: Precision and Recall results of the mixed machine learning and gradient-based method presented in section 3.9 (GBT-SIFT) trained on the Mars dataset presented in 4.1.2, compared to the results in [17]. Shown are the results of both the training set (odd areas) in (a), and test set (even areas) in (b).

Sub-Regions	Vaz [17] Results		Our Results	
	Precision	Recall	Precision	Recall
Area 1	0.4256	0.8454	0.6532	0.8563
Area 3	0.4725	0.9770	0.7367	0.9102
Area 5	0.5145	0.9067	0.7147	0.8346
Area 7	0.3055	0.9618	0.5496	0.9213
Area 9	0.3904	0.9126	0.3751	0.9486
Area 11	0.2491	0.7691	0.2151	0.9053
Area 13	0.3436	0.7702	0.3573	0.8489

(a) Precision Recall Training Region Results: Vaz [17] vs Our Approach

Sub-Regions	Vaz [17] Results		Our Results	
	Precision	Recall	Precision	Recall
Area 2	0.4451	0.8538	0.7823	0.9001
Area 4	0.4521	0.8750	0.7189	0.8274
Area 6	0.4516	0.9126	0.3564	0.4792
Area 8	0.4330	0.8518	0.6726	0.7991
Area 10	0.4541	0.8463	0.5259	0.8618
Area 12	0.4140	0.8736	0.4374	0.9043
Area 15	0.1404	0.9449	0.0880	0.2777

(b) Precision Recall Test Region Results: [17] vs Our Approach

The results shown are an overall improvement on the machine learning method itself, notably an increased precision. Combining the machine learning and gradient orientation approaches results in many less false positives while preserving similar recall rates as shown in Tables 10a and 11. The metrics measured shown in Tables 10b and 11b are similar to the standard machine learning approach. Unfortunately, there are no improvements for the cross region machine learning models. The method still struggles with more complex dune patterns, especially the White Sands region, shown in Figures 45(k) and 45(l).

The method was also tested on another dataset. The Mars dataset, described in section 4.1.2 was constructed and used in the research presented in [17]. The results presented in this paper were acquired and the same evaluation methodology was applied on their results. We ran our method on the dataset, therefore we can compare the results side-by-side.

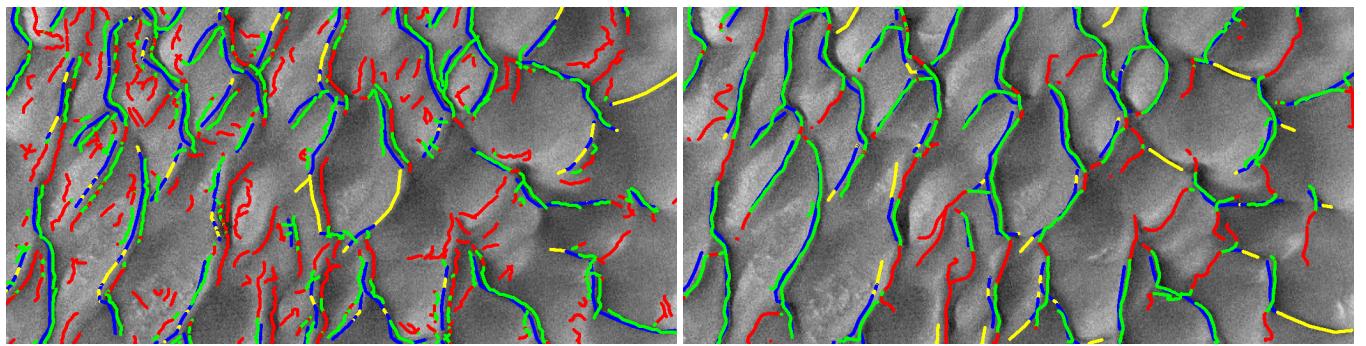
The region was sub-sampled into sixteen sub-region for efficient processing. The machine learning method requires a training set, therefore, half of the regions where used for training and the remaining half for testing. For simplicity, odd regions were used for training and even for testing. The precision/recall and dune metrics results for both the Vaz and our method are presented in Tables 12 and 10b.

Clearly, the results show that the performance of our method is superior, mainly in precision. Recall values are roughly equivalent in most areas, but our precision is typically much better. This can be attributed to the improved localization of the crest-lines using the response maps. There are a few cases in which our method fails to deliver high precision and recall. These cases can be easily explained simply by examining the projected results in Figure 46.

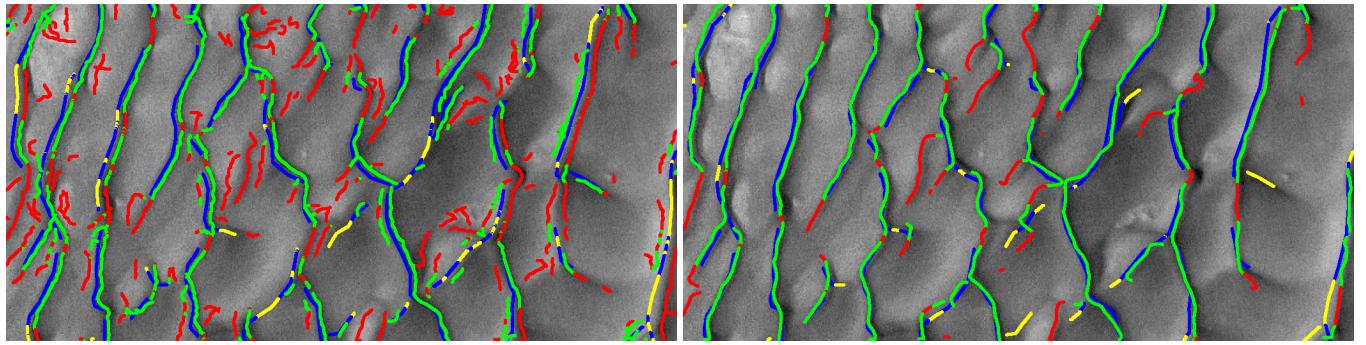
As with Figure 45, the results shown in Figure 46 display the detected crest-lines overlapped with the ground truth of each area. The color schemes shows green lines as correctly detected crest-lines and red lines as false positives. Blue lines are ground truth crest-lines which have been correctly identified, and yellow lines are unidentified ground truth crest-lines (false negatives).

The first notable issue with the Vaz results are the noisiness and disjointness of the detections, while our results are much smoother, less noisy, contiguous segments. Also, in many cases, the Vaz results show duplicate parallel detections next to each crest-lines which falsely increase the apparent precision, while our results are more accurate single segments for each crest-lines. This especially noticeable in areas 1, 5, 6, 7 and 10 (Figures 46(a), 46(e), 46(f), 46(g) and 46(j)).

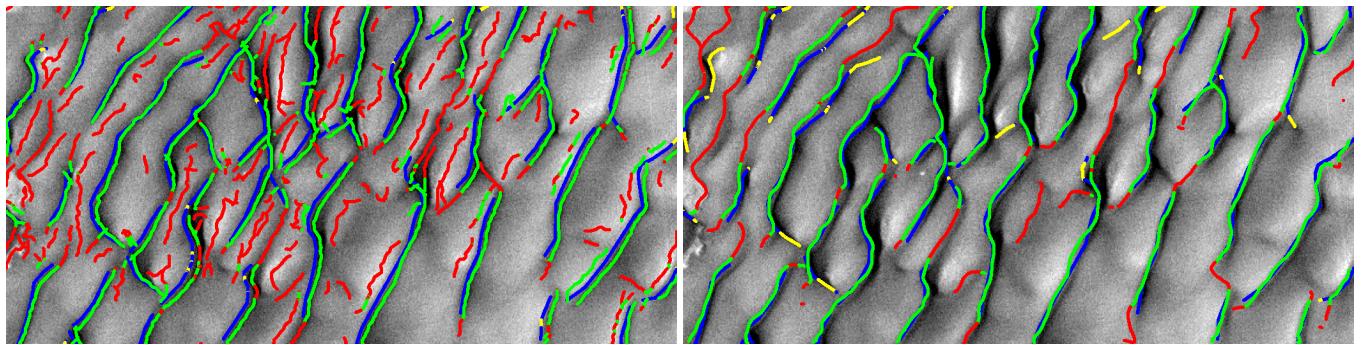
The final issue is the ground truth itself. The ground truth was labeled by the Vaz researchers and is for the most part accurate, but there are many discrepancies. In many cases, some crest-lines are missing from the ground truth, and often the localization of the ground truth itself is quite poor. Areas 9 and 15 (46(i), 46(n)) are good examples of poor labeling of crest-lines in the ground truth. Many of the detections in those areas should in fact be true positives. Area 6 (46(f)) is a good example of poor localization. Our results show that the actual crest-lines were detected but because the ground truth



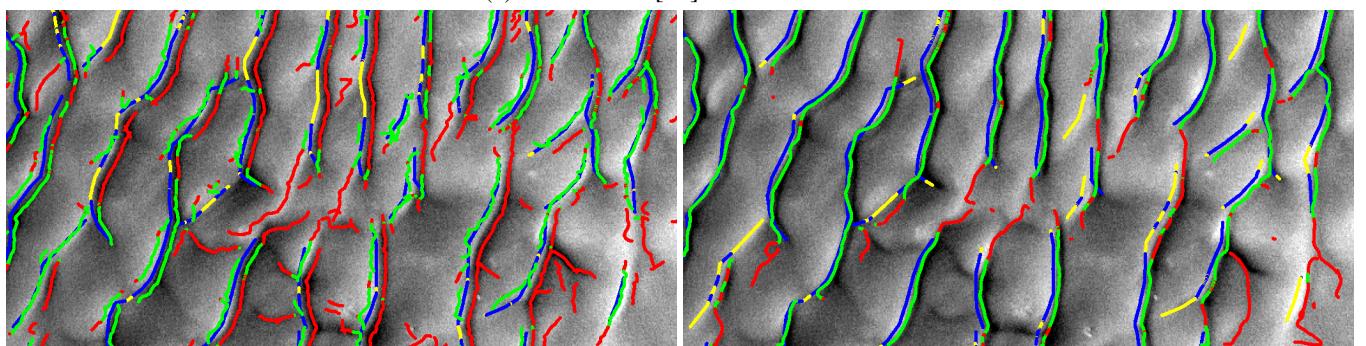
(a) Area 1: Vaz [17] and Our Results



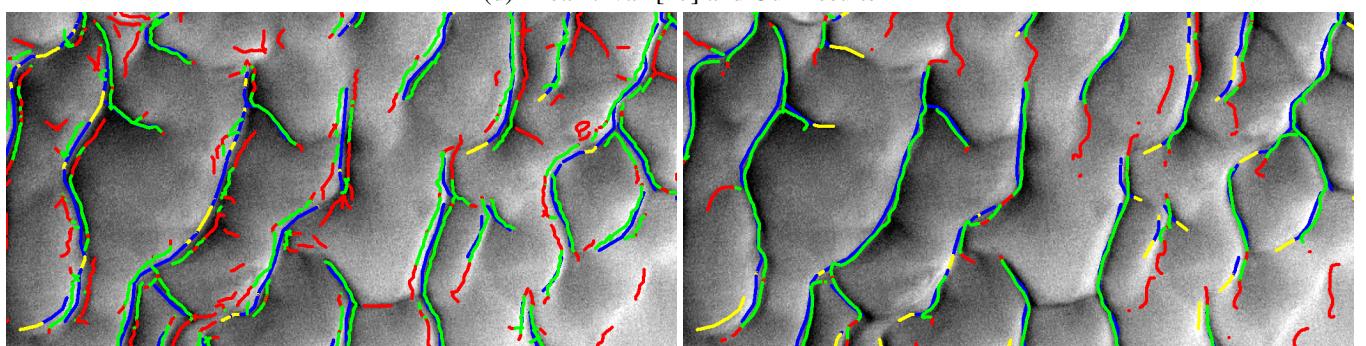
(b) Area 2: Vaz [17] and Our Results



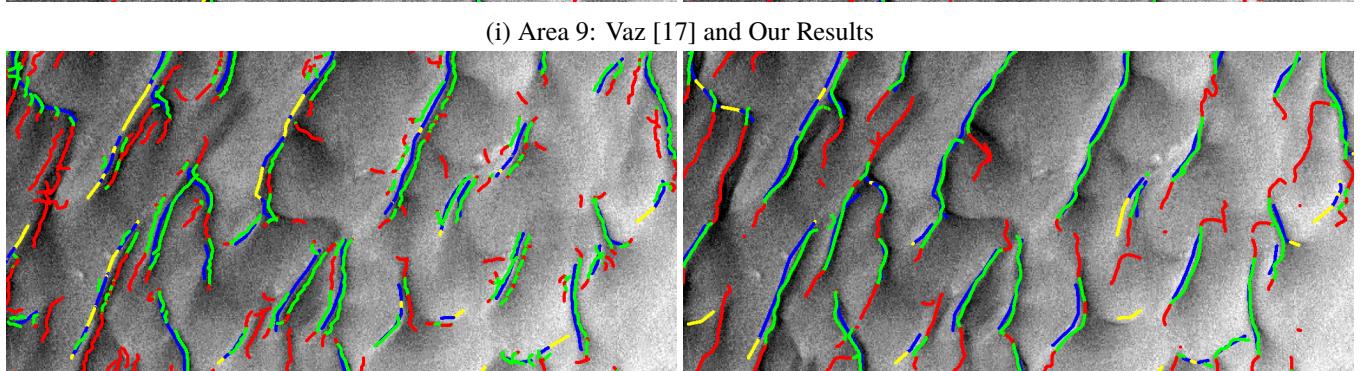
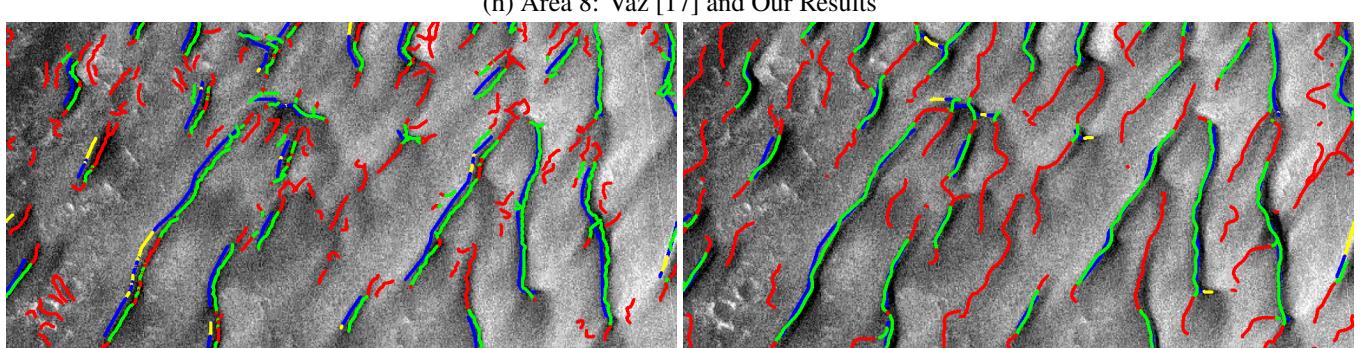
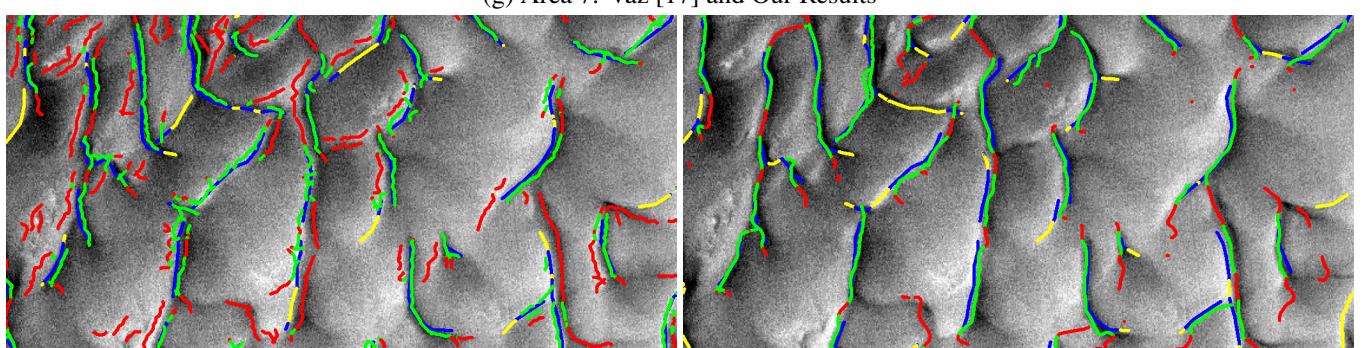
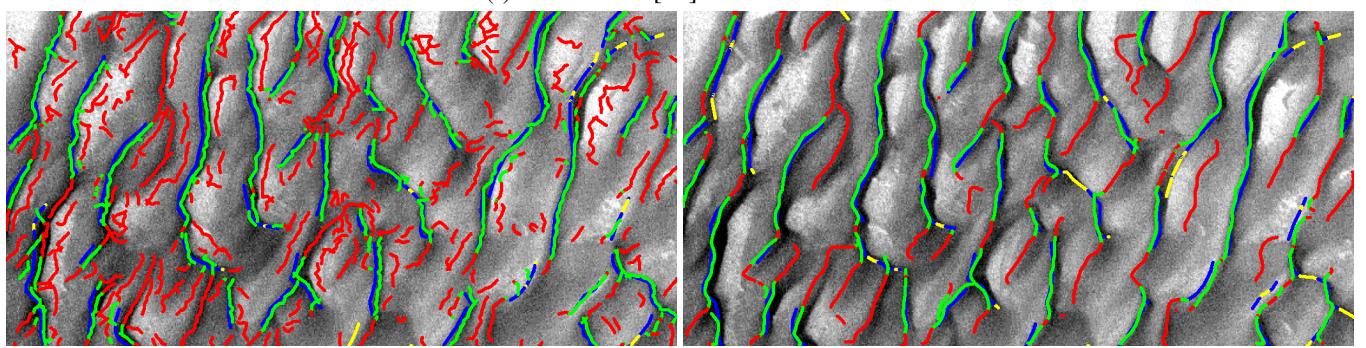
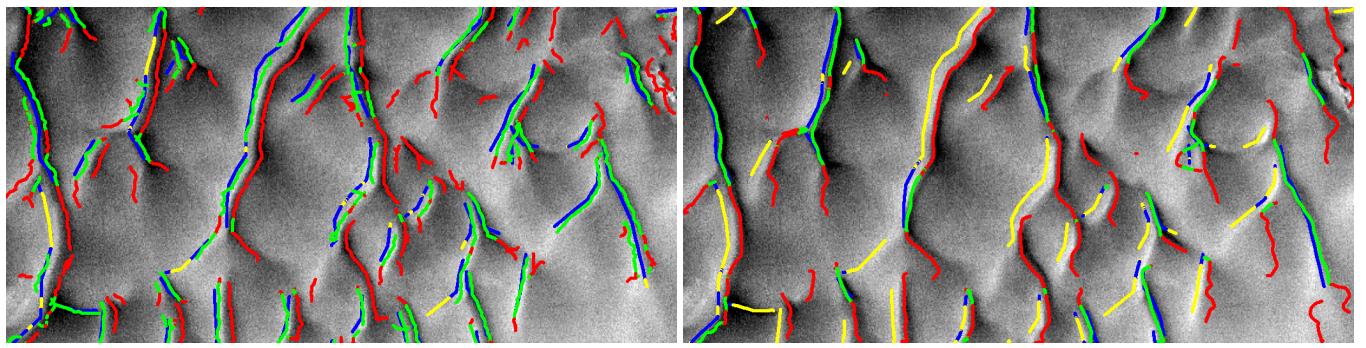
(c) Area 3: Vaz [17] and Our Results



(d) Area 4: Vaz [17] and Our Results



(e) Area 5: Vaz [17] and Our Results



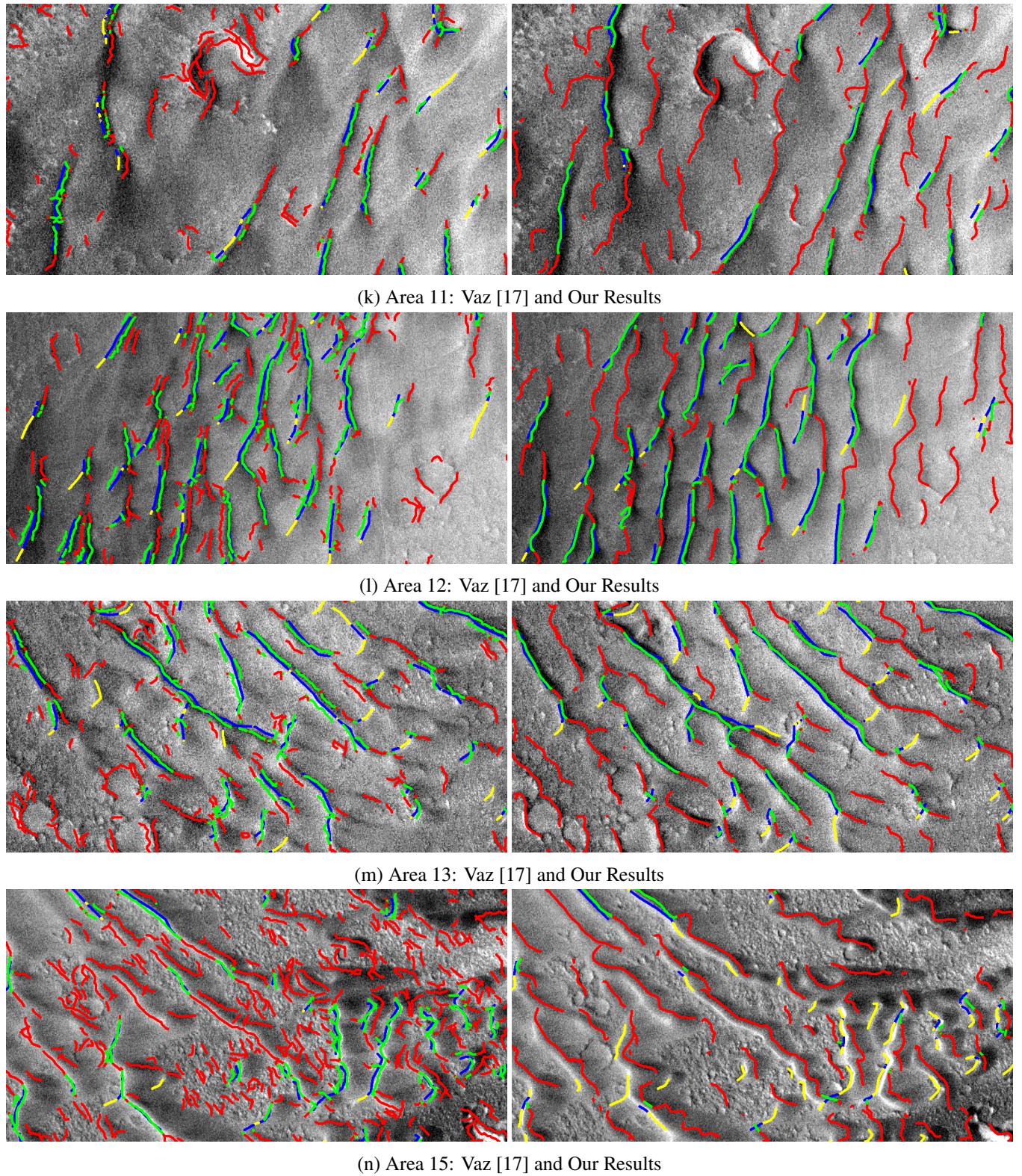


Figure 46: Crest-line detection results plotted for all regions used in the Mars dataset. The Vaz [17] results (on the left) compared to our results (on the right). **Legend:** *Green* - True Positives, *Red* - False Positives, *Yellow* - False Negatives, *Blue* - Ground truth correctly detected.

Table 13: Dune metrics results of the mixed machine learning and gradient-based method on the Mars dataset presented in 4.1.2. The dune metrics were computed on our crest-line detection results, the ground truth, and the crest-lines detected in [17]. Shown are the results of both the training set (odd areas) in (a), and test set (even areas) in (b).

Sub-Regions	Vaz [17] Results		Our Results	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Area 1	5.0668°	40.6086	1.6556°	12.748
Area 3	5.5938°	52.1585	2.3637°	7.4039
Area 5	8.12824°	93.7826	2.0505°	4.8174
Area 7	6.39212°	28.3885	2.5625°	10.6288
Area 9	5.5570°	74.2544	9.8787°	8.1765
Area 11	22.1351°	40.2464	3.3544°	8.9837
Area 13	20.5087°	89.5503	14.651°	9.7079

(a) Computed Dune Metrics Results (Training Regions): Vaz [17] vs Our Approach

Sub-Regions	Vaz [17] Results		Our Results	
	Δ_θ	Δ_d (Pixels)	Δ_θ	Δ_d (Pixels)
Area 2	5.5938°	52.1585	5.8210°	35.053
Area 4	6.3921°	28.3885	1.9697°	21.4148
Area 6	2.51432°	12.6348	9.0910°	17.5553
Area 8	1.7801°	35.8738	3.4364°	5.8379
Area 10	1.7096°	54.185	2.6398°	6.9980
Area 12	8.3404°	63.7937	2.0865°	46.1008
Area 15	17.1538°	0.9386	11.7136°	55.664

(b) Computed Dune Metrics Results (Test Regions): [17] vs Our Approach

is offset from the actual crest-lines (by a distance greater than our error threshold ϵ), the detections count as false positives when they really should be true positives. The effect of these issues are reflected in the results generated in Tables 12 and 10b.

As stated previously, determining true positives was dependent on the selection of the distance error threshold parameter ϵ . The parameter was admittedly arbitrary selected to be $\epsilon = 10$, based on what seems to be a reasonable allowable error in localization of crest-lines. The parameter ϵ directly affects the reported precision and recall results. In order to have a better understanding and a better comparison, the plots in Figure 47 show the precision and recall results versus the distance error threshold ϵ parameter, for both our approach and the Vaz results.

In areas 1 and 2 (Figures 47(a), 47(b)), the recall values stay roughly equivalent for both methods, with our approach slightly leading in area 1 and with the Vaz results having a slight edge in area 2. The reason for this is most likely because the machine learning method was trained on odd areas and tested on even areas, therefore, a lower recall should be expected. The main distinction in the two approaches is the precision results. Our approach offers vastly and consistently better precision than the Vaz approach.

To summarize the results, the machine learning approaches are by far the best methods for detecting the crest-lines. This is easily seen simply in the numbers, shown in Tables 8 to 13. Not only are the precision, recall, and metrics results better in almost every case, the quality of the produced crest-lines are much better. The detected crest-lines in the machine learning methods are generally smoother and are more contiguous than the previous methods.

The main benefits of the machine learning is that it is a much simpler approach than other appearance based approach. By simpler, we mean that fewer parameters and optimization are required to achieve good results. In appearance based approaches presented in sections 3.3, 3.4, 3.5, 3.6, and 3.7, many parameters are required in order to tune the performance of crest-line detection. Typically, the parameters also require optimization based on the appearance, scale, and contrast in each region. These factors are solved in machine learning model which simply learns the distinctions of crest-lines versus non-crest-lines.

The benefits of the machine learning method come at a cost. Labeled data is required in order to train the models, which requires the crest-lines in an image region to be manually and accurately traced by an expert. The process of creating the ground truth for training can be tedious and error prone. Also, there seems to be some struggles when training a model across regions of different types of dunes. If the dunes of two different regions are similar, then a single model might be sufficient. However, when training a model on linear dunes for example, the results of using this model on a complex dune structure may not be optimal.

The final section summarizes and discusses the work done in this research, along with offering potential future work

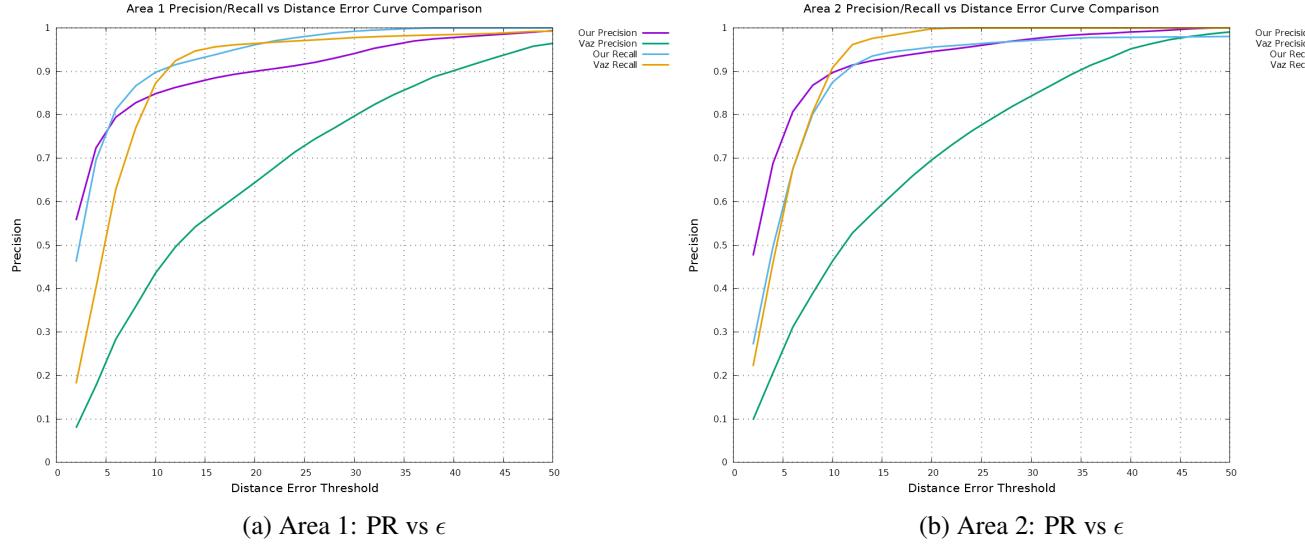


Figure 47: Plots of the Precision and Recall comparison between our method and Vaz [17] for Area 1 (a) and Area (b) as the allowable distance threshold value ϵ increases.

methods.

5 Conclusion

Detecting dune crest-lines in satellite images is a challenging application of computer vision. Although there have been many computer vision methods used to solve similar problems, the space of research in this particular application is relatively sparse. The main goal of the research presented is to widen the understanding of the challenges in crest-line detection, and to propose candidate solution. Ultimately, crest-line detection is used as a stepping stone for extracting higher level information and dune metrics such as orientation, spacing, length, bifurcations, dune movement, and many other properties inherent to dune fields.

5.1 Discussion

The most obvious and main challenge discovered during this research lies in the bimodal distribution of the gradient histograms of the images. That is to say, the transitions from darker areas to brighter and making the determination of which gradients belong to the crest-lines. Additionally, we discovered that often gradient magnitudes are not necessarily a sure way to determine crest-lines.

In this research, many approaches were tried with varying levels of success. Approaches varied from appearance-based, to gradient-based, to machine learning methods. Although individually many of the approaches had poor performance, valuable information about the problem space was extracted.

The main contribution of the gradient-based methods was the computation of the dominant orientation, which turned out to be a key piece of information in determining the crest-lines. Although the results are not perfect, computing the normalized histogram of gradients proved to provide consistent results in computing the dominant orientation of the dune field.

Another important contribution is the use of the gradient direction map, which enabled good segmentation of regions in the image, in order to determine the crest-line. A drawback of this method is that the computation of the gradient direction map is heavily dependent on a reliable computation of the dominant orientation. For simpler linear dune fields, the dominant orientation is typically accurate. However, in complex dune field structures, the dominant orientation computation may not be as accurate, which can affect the gradient direction map.

The approaches with the most promising results are without doubt the machine learning based methods. Machine learning techniques enable us to automatically create an accurate model representation which characterizes the problem space. The drawback of the machine learning methods is the need for large datasets for accurate model representation. The datasets used in this research were relatively small, including a terrestrial dataset spanning six distinct regions, and a Martian dataset, which included images of a specific region on Mars. In order to construct a more fool-proof model, more data would need to be collected.

Fortunately, we were able to successfully train models to recognize the difference between crest-line, and non-crest-line image patches with our limited dataset, as a proof of concept. When the model is applied to each pixel in the image, a response map could be generated. The response map could then be used as a filter for detecting crest-lines.

Two main approaches utilized the response maps. The first simply thresholded the response maps and applied non-maxima suppression or skeletonization in order to recover the crest-lines. The second combined the gradient direction map segmentation concept with the response maps to filter out crest-line candidates from the detected contours.

Along with the crest-line detection algorithms presented in this research, an attempt at computing dune metrics was also proposed. The two main metrics computed were the dune field orientation and the inter-dune spacing. These two were chosen mainly for their simplicity of computation. Although the proposed methods to compute these metrics may not be the most optimal, the results generated were sufficient enough for evaluating the approaches presented.

Section 4.3 presented the results of each approach applied on both the terrestrial and Martian datasets. Table 14 presents a summary of the results of each approach on both linear and complex dune patterns. The most promising approaches clearly are the machine learning-based approaches, which yielded much better precision, recall, and dune metric calculation results. The mixed machine learning and gradient based approach presented in section 3.9 provides the best overall results for dune detection.

Since the results presented in [17] were made available, we were able to compare side-by-side their results with ours. It is clear that our results are superior than the method proposed in [17], not only in precision and recall, but also in dune metric calculation and in overall quality of crest-line extraction.

The crest-lines extracted by [17] tended to be noisy, duplicated, and have poor localization. On the other hand, our results provide much smoother, contiguous, and better localized crest-lines. In many of the case where the results appeared lower were due to poor quality of the labeled ground truth. Often, in the ground truth of the Martian dataset, many of the crest-lines were incorrectly localized, or simply absent, in locations where our approach did in fact correctly detect a crest-line. Correcting the ground truth would most likely yield higher precision and recall for our reported results. We

Table 14: Summary of the precision and recall results for each approach applied to the Kalahari, Namibia, Skeleton Coast, and WDC terrestrial images.

Kalahari	Precision	Recall
Appearance-Based	0.7651	0.7094
Edge-Based	0.5793	0.9852
Watershed Transform	0.9087	0.8093
Frequency Domain	0.6329	0.9730
Gradient Orientation	0.7181	0.9370
Machine Learning	0.9379	0.9726
Mixed ML-Gradient	0.9523	0.9904

(a) Kalahari Results

Namibia	Precision	Recall
Appearance-Based	0.5543	0.7493
Edge-Based	0.5842	0.9626
Watershed Transform	0.6115	0.6617
Frequency Domain	0.5835	0.9811
Gradient Orientation	0.6297	0.9883
Machine Learning	0.8491	0.9676
Mixed ML-Gradient	0.9512	0.9831

(b) Namibian Results

Skeleton Coast	Precision	Recall
Appearance-Based	0.7117	0.8033
Edge-Based	0.7899	0.8187
Watershed Transform	0.5700	0.7489
Frequency Domain	0.6448	0.7767
Gradient Orientation	0.7697	0.9935
Machine Learning	0.7632	0.9536
Mixed ML-Gradient	0.9356	0.9806

(c) Skeleton Coast Results

WDC	Precision	Recall
Appearance-Based	0.5993	0.5581
Edge-Based	0.5807	0.7478
Watershed Transform	0.6279	0.2532
Frequency Domain	0.7221	0.6571
Gradient Orientation	0.7589	0.9302
Machine Learning	0.8981	0.8554
Mixed ML-Gradient	0.9711	0.7996

(d) WDC Results

decided against attempting to correct the ground truth for two main reasons: it is a very tedious and error prone task, and tampering with the data would not produce a fair comparison of the Vaz [17] and our approach.

Overall, the research presented offers many different approaches and many more avenues to explore. The results are very promising, however there are still many improvements and more research to be done.

5.2 Future Work

There is a multitude of potential methods to be tried and a fair amount of future work to be done in this application to make it a reliable tool for researchers. Obvious improvements can be made in the computation of the dune metrics. The current methods used to compute the metrics are somewhat minimal and simplistic.

Additionally, many of the metrics desired by scientist in the field of dune research are absent. Computing bifurcation and studying the movement of dunes is an interesting problem, but would require a large amount of data to be further studied. Using temporal data could provide valuable insight on the formation and complex patterns inherent to dune fields.

Another area which requires further study is the computation of the dominant orientation of the dune field. The bimodal distribution of the gradients still poses problems in complex dune structures. There may be options for computing the orientation in a more optimal way. Alternatively, computing the local dominant orientation may provide more useful information and could be a solution to the problem. Computing the local orientation has merit but poses other problems in the computation of the gradient direction map. Also, determining the optimal local scale then becomes a crucial task for determining the local orientation of dunes.

Other approaches have been discussed but were not included in this research. Namely, the use of Tensor Voting methods to determine and trace dune crest-lines. Another interesting approach is to use the sun's orientation relative to the dune field to generate a pseudo elevation map using shape-from-shading techniques. If the light source is known, it may be possible to reconstruct a 3D model of the dune field, which in turn could facilitate the detection of crest-lines.

References

- [1] The fast fourier transform and its applications. *IEEE Transactions on Education*, 12(1):27–34, 1969.
- [2] The finite fourier transform. *IEEE Trans. Audio Electroacoustics*, 17(2):77–85, 1969.
- [3] A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, Signal Processing*, 27(1):13–18, 1979.
- [4] H. Siegelmann A. Ben-Hur, D. Horn and V. Vapnik. Support vector clustering. *Journal of Machine Learning Research*, (2):125–137, 2001.
- [5] Lehman C. Mulla D. Barnes, R. Priority-flood: An optimal depression-filling and watershed-labeling algorithm for digital elevation models. *Computers and Geosciences*, 62:117–127, 2014. doi:10.1016/j.cageo.2013.04.024.
- [6] Serge Beucher and Christian Lantu  j. Workshop on image processing, real-time edge and motion detection. 1979. doi:10.1016/0165-1684(94)90059-0.
- [7] G. Bradski. *Dr. Dobb's Journal of Software Tools*, 2000.
- [8] Hanson A.R. Riseman E.M. Burns, J.B. Extracting straight lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):425–455, 1986. doi:10.1109/TPAMI.1986.4767808.
- [9] V. Vapnik C. Cortes. Support-vector networks. *Machine Learning*, 20(3):273–297, 1995. doi:10.1007/BF00994018.
- [10] P. Gouinaud C. Gouinaud, A. I. Doutoum and M. K. Traore. Sar image automated detection of dune area. *Geoscience and Remote Sensing Symposium (IGARSS), 2013 IEEE International*, pages 1489–1492, 2013. doi:10.1109/IGARSS.2013.6723068.
- [11] R. Manduchi C. Tomasi. Bilateral filtering for gray and color images. *Proceedings of the 1998 IEEE International Conference on Computer Vision*, 1998. doi:10.1109/ICCV.1998.710815.
- [12] John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (8):679–698, 1986. doi:10.1109/TPAMI.1986.4767851.
- [13] F. Cazals and M. Pouget. Topology driven algorithms for ridge extraction on meshes, 2005.
- [14] Chih-Chiang Tai Yung-Chung Wei Chih-Chung Kao, Liang-Hwei Lee. Extracting the ocean surface feature of non-linear internal solitary waves in modis satellite images. *Intelligent Information Hiding and Multimedia Signal Processing*, 2007. doi:10.1109/IIHMSP.2007.4457485.
- [15] Franklin C. Crow. Summed-area tables for texture mapping. *Computer Graphics*, 1984. doi:10.1145/800031.808600.
- [16] Navneet Dalal and Bill Triggs. Histograms of oriented gradients for human detection. *CVPR*, 2005. doi:10.1109/CVPR.2005.177.
- [17] Maria Barata Lori Fenton Timothy Michaels David Vaz, Pedro Sarmento. Object-based dune analysis: Automated dune mapping and pattern characterization for ganges chasma and gale crater, mars. *Geomorphology*, 250:128–139, 2015. doi:10.1016/j.geomorph.2015.08.021.
- [18] Moisan L. Morel J.-M. Desolneux, A. Meaningful alignments. *International Journal of Computer Vision*, (40):7–23, 2000. doi:10.1023/A:1026593302236.
- [19] David Douglas and Thomas Peucker. Algorithms for the reduction of the number of points required to represent a digitized line or its caricature. *The Canadian Cartographer*, 10(2):112â€¢122, 1973. doi:10.1002/9780470669488.ch2.
- [20] L. Dr  gu   and T. Blaschke. Automated classification of landform elements using object-based image analysis. *Geomorphology*, 81(3-4):330–344, 2006. doi:10.1016/j.geomorph.2006.04.013.
- [21] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Comm. ACM*, (15):11–15, 1972. doi:10.1145/361237.361242.

- [22] Garret P. Duffy and John E. Hughes-Clarke. Application of spatial cross correlation to detection of migration of submarine sand dunes. *Journal of Geophysical Research*, 110(F4), 2005. doi:10.1029/2004JF000192.
- [23] C.-Kocurek G. Lake.L.W. Ewing, R. Pattern analysis of dune-field parameters. *Earth Surface Processes and Landforms*, 31(9):1176–1191, 2006. doi:10.1002/esp.1312.
- [24] Peyret-A.-P.B. Kocurek-G. Bourke M. Ewing, R.C. Dune field pattern formation and recent transporting winds in the olympia undae dune field, north polar region of mars. *Journal of Geophysical Research*, (115):E08005, 2010. doi:10.1007/s11071-006-2016-3.
- [25] Michaels-T.I.-Beyer R.A. Fenton, L.K. Inverse maximum gross bedform-normal transport 1: How to determine a dune-constructing wind regime using only imagery. *Icarus*, 230:5–14, 2014. doi:10.1016/j.icarus.2013.04.001.
- [26] Michaels-T.I.-Chojnacki M.-Beyer R.A. Fenton, L.K. Inverse maximum gross bedform-normal transport 2: Application to a dune field in ganges chasma, mars and comparison with hirise repeat imagery and mramps. *Icarus*, 230:47–63, 2014. doi:10.1016/j.icarus.2013.07.009.
- [27] Kenneth E. Foote and Margaret Lynch. Geographic information systems as an integrating technology: Context, concepts, and definitions. *he Geographer's Craft Project, Department of Geography, The University of Colorado at Boulder*, 2015.
- [28] E. W. Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, (21):768–769, 1965.
- [29] J. H. Friedman. Greedy function approximation: A gradient boosting machine. 29(5):1189–1232, 2001. doi:10.1214/aos/1013203451.
- [30] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics and Data Analysis*, 38(4):367–378, 2002. doi:10.1016/S0167-9473(01)00065-2.
- [31] Ryan C. Ewing Gary Kocurek. Aeolian dune field self-organization - implications for the formation of simple versus complex dune-field patterns. *Geomorphology*, 72(1-4):94–105, 2005. doi:10.1016/j.geomorph.2005.05.005.
- [32] Nicola L. C. Talbot Gavin C. Cawley. On over-fitting in model selection and subsequent bias in performance evaluation. *Journal of Machine Learning Research*, 11:2079–2107, 2010.
- [33] Mrinmoy Ghorai and Bhabatosh Chanda. A robust faint line detection and enhancement algorithm for mural images. *Fourth National Conference on Computer Vision, Pattern Recognition, Image Processing and Graphics*, 2013. doi:10.1109/NCVPRIPG.2013.6776175.
- [34] Maheshkumar H. Kolekar G.Lloyds Raja. Illumination normalization for image restoration using modified retinex algorithm. *India Conference (INDICON), 2012 Annual IEEE*, 2012.
- [35] Dian Gong and Gerard Medioni. Probabilistic tensor voting for robust perceptual grouping. *Computer Vision and Pattern Recognition Workshops (CVPRW)*, 2012. doi:10.1109/CVPRW.2012.6238926.
- [36] Andrew S. Goudie. Desert landforms in namibia - a landsat interpretation.
- [37] Andrew S. Goudie. Namib sand sea: Large dunes in an ancient desert.
- [38] Qiang Chen Guoji Zhang and Quansen Sun. Illumination normalization among multiple remote-sensing images. *IEEE Geoscience and Remote Sensing Letters*, 11(9), 2014. doi:10.1109/LGRS.2013.2295794.
- [39] Robert M. Haralick. Performance characterization in image analysis: Thinning, a case in point. *Pattern Recognition Letters*, 13(1):5–12, 1992. doi:10.1016/0167-8655(92)90108-C.
- [40] Douglas M. Hawkins. The problem of overfitting. *Journal of Chemical Information and Computer Sciences*, 44(1):1–12, 2004. doi:10.1021/ci0342472.
- [41] D. Hebb. *The Organization of Behavior*. 1949.
- [42] Tinne Tuytelaars Herbert Bay, Andreas Ess and Luc Van Gool. Speeded up robust features. *ETH Zurich, Katholieke Universiteit Leuven*, 2006. doi:10.1016/j.cviu.2007.09.014.

- [43] John Hershberger and Jack Snoeyink. Speeding up the douglasâŠpeucker line-simplification algorithm. *Proc 5th Symp on Data Handling*, page 134âŠ143, 1992.
- [44] Paul P. Hesse. The australian desert dunefields: formation and evolution in an old, flat, dry continent.
- [45] P.V.C. Hough. Machine analysis of bubble chamber pictures. *International Conference High Energy Accelerators and Instrumentation*, 1959.
- [46] P.V.C. Hough. Method and means for recognizing complex patterns, 12 1962.
- [47] David J. Livingstone Igor V. Tetko and Alexander I. Luik. Neural network studies. 1. comparison of overfitting and overtraining. *Journal of Chemical Information and Computer Sciences*, 35(5):826–833, 1995. doi:10.1021/ci00027a006.
- [48] Jason C. Isaacs and Ross Goroshin. Automated cable tracking in sonar imagery. *OCEANS 2010*, 2010. doi:10.1109/OCEANS.2010.5664414.
- [49] C. Tomasi J. Shi. Good features to track. *9th IEEE Conference on Computer Vision and Pattern Recognition*, 1994. doi:10.1109/CVPR.1994.323794.
- [50] Duin R.P.W. Jianchang M. Jain, A.K. Statistical pattern recognition: a review. pattern analysis and machine intelligence. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(1):4–37, 2000. doi:10.1109/34.824819.
- [51] Steven H. Williams James R. Zimbelman. Eolian dunes and deposits in the western united states as analogs to wind-related features on mars.
- [52] Tai Y. Wu T. Tang C.K. Jia, J. Video repairing under variable illumination using cyclic motions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(5):832–839, 2006. doi:10.1109/TPAMI.2006.108.
- [53] Tang C.K. Jia, J. Tensor voting for image correction by global and local intensity alignment. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(1):36–50, 2005. doi:10.1109/TPAMI.2005.20.
- [54] K. Polthier K. Hildebrandt and M. Wardetzky. Smooth feature lines on surface meshes. *Symposium on geometry processing*, (85):85âŠ90, 2005.
- [55] Ewing R. C. Kocurek, G. Aeolian dune field self-organization - implications for the formation of simple versus complex dune field patterns. *Geomorphology*, (72):94–105, 2005. doi:10.1016/j.geomorph.2005.05.005.
- [56] R. A. Beyer L. K. Fenton, T. I. Michaels. Aeolian sediment sources and transport in ganges chasma, mars: Morphology and atmospheric modeling. 2012.
- [57] M. Schmitt L. Najman. Watershed of a continuous function. *Signal Processing (Special issue on Mathematical Morphology.)*, 38(1):99–112, 1994. doi:10.1016/0165-1684(94)90059-0.
- [58] N. Lancaster. Dunes on the skeleton coast, namibia (south west africa): Geomorphology and grain size relationships. *Earth Surface Processes and Landforms*, 7:575–587, 1982. doi:10.1002/esp.3290070606.
- [59] N. Lancaster. Winds and sand movements in the namib sand sea. *Earth Surface Processes and Landforms*, 10(6):607–619, 1985. doi:10.1002/esp.3290100608.
- [60] N. Lancaster. Development of linear dunes in the southwestern kalahari southern africa. *Journal of Arid Environments*, pages 233–244, 1987.
- [61] Sang Uk Lee, Seok Yoon Chung, and Rae Hong Park. A comparative performance study of several global thresholding techniques for segmentation. *Computer Vision, Graphics, and Image Processing*, 52(2):171–190, 1990. doi:10.1016/0734-189X(90)90053-X.
- [62] Samuel H. Lewis and Aijuan Dong. Detection of breast tumor candidates using marker-controlled watershed segmentation and morphological analysis. *Image Analysis and Interpretation (SSIAI), 2012 IEEE Southwest Symposium on*, 41:187–228, 2012. doi:10.1109/SSIAI.2012.6202438.
- [63] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision*, 2(30):77–116, 1998. doi:10.1023/A:1008045108935.

- [64] S. P. Lloyd. Least square quantization in pcm. *IEEE Transactions on Information Theory*, 28(2):129–137, 1982. doi:10.1109/TIT.1982.1056489.
- [65] Bebis G. Nicolescu M. Skurikhin A. Loss, L. An iterative multi-scale tensor voting scheme for perceptual grouping of natural shapes in cluttered backgrounds. *Computer Vision and Image Understanding*, 113(1):126–149, 2009. doi:10.1016/j.cviu.2008.07.011.
- [66] Jose Saraiva Lourenco Bandeira, Jorge S. Marques and Pedro Pina. Advances in automated detection of sand dunes on mars. *Earth Surface Processes and Landforms*, 38(3):275–283, 2013. doi:10.1002/esp.3323.
- [67] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2), 2004. doi:10.1023/B:VISI.0000029664.99615.94.
- [68] B. Koch M. Ghadiry, A. Shalaby. A new gis-based model for automated extraction of sand dune encroachment case study: Dakhla oases, western desert of egypt. *The Egyptian Journal of Remote Sensing and Space Sciences*, 15(1):53–65, 2012. doi:10.1016/j.ejrs.2012.04.001.
- [69] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, (1):281–297, 1967. doi:10.1.1.308.8619.
- [70] W. McCulloch and W. Pitts. A logical calculus of ideas immanent in nervous activity. *Bulletin of Mathematical Biophysics*, 5(4):115–133, 1943.
- [71] Ameet Talwalkar Mehryar Mohri, Afshin Rostamizadeh. *Foundations of Machine Learning*. The MIT Press, 2012.
- [72] F. Meyer. Color image segmentation. *Image Processing and its Applications*, 1992., International Conference on, 1992.
- [73] Hannan Samet Michael B. Dillencourt and Markku Tamminen. A general approach to connected-component labeling for arbitrary image representations. *Journal of the ACM*, 39(2):253, 1992. doi:10.1145/128749.128750.
- [74] T. Mitchell. *Machine Learning*. McGraw Hill, 1997.
- [75] Medioni G. Mordohai, P. G. Stereo using monocular cues within the tensor voting framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(6):968–982, 2006. doi:10.1109/TPAMI.2006.129.
- [76] S. Lewin M.W. Telfer, R.M. Fyfe. Automated mapping of linear dunefield morphometric parameters from remotely-sensed data. *Aeolian Research*, 19:215–224, 2015. doi:10.1016/j.aeolia.2015.03.001.
- [77] B. Triggs N. Dalal. Histograms of oriented gradients for human detection. *CVPR*, pages 886–893, 2005. doi:10.1109/CVPR.2005.177.
- [78] Medioni G. Nicolescu, M. Voting-based computational framework for visual motion analysis and interpretation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(5):739–752, 2005. doi:10.1109/TPAMI.2005.91.
- [79] Robert L. Ogniewicz. Automatic medial axis pruning by mapping characteristics of boundaries evolving under the euclidean geometric heat flow onto voronoi skeletons. *Harvard Robotics Laboratory, Tech. Rep. 95-4*, 1995. doi:10.1.1.45.3291.
- [80] J. Ogor and B. Zerr. Comparisons of crest line extraction algorithms applied to marine dunes. *OCEANS 2016 MTS/IEEE Monterey*, 2016. doi:10.1109/OCEANS.2016.7761318.
- [81] Edwin Olson. On computing the average orientation of vectors and lines. *IEEE International Conference on Robotics and Automation*, pages 3869–3874, 2011. doi:10.1109/ICRA.2011.5979562.
- [82] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Trans. Sys., Man., Cyber.*, 9(1):62–66, 1979. doi:10.1109/TSMC.1979.4310076.
- [83] Lei Zhang Paul Bao and Xiaolin Wu. Canny edge detection enhancement by scale multiplication. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 27(9):1485–1490, 2005. doi:10.1109/TPAMI.2005.173.
- [84] C. Nishan Canagarajah Paul R. Hill and David R. Bull. Image segmentation using a texture gradient based watershed transform. *IEEE Transactions on Image Processing*, 12(12):1618–1633, 2003. doi:10.1109/TIP.2003.819311.

- [85] Nathaniel Pepe. *The Geomorphology, Eolian Activity, and Petrology of the Winnemucca Dune Complex, Humboldt County, Nevada, USA*. PhD thesis, University of Nevada, Reno.
- [86] J. R. Quinlan. Simplifying decision trees. *International Journal of Man-Machine Studies*, 27(3):221, 1987.
- [87] R. E. Woods R. C. Gonzalez. *Digital Image Processing*. 2008.
- [88] K. Shanmugam R. M. Haralick and I. Dinstein. Textural features for image classification. *IEEE Transactions on Systems Man, and Cybernetics*, SMC-3(6):610–621, 1973. doi:10.1109/TSMC.1973.4309314.
- [89] P. E. Hart R. O. Duda. Use of the hough transformation to detect lines and curves in pictures. *Communications, ACM*, 15(1):11–15, 1972. doi:10.1145/357994.358023.
- [90] A. K. Goswami R. R. Chowdhury, B. Deshmukh and S. S. Prasad. Neural network based dunal landform mapping from multispectral images using texture features. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 4(1):171–184, 2011. doi:10.1109/JSTARS.2010.2062491.
- [91] Urs Ramer. An iterative procedure for the polygonal approximation of plane curves. *Computer Graphics and Image Processing*, 1(3):244–256, 1972. doi:10.1016/S0146-664X(72)80017-0.
- [92] Jason D. M. Rennie, Lawrence Shih, Jaime Teevan, and David R. Karger. Tackling the poor assumptions of naive bayes text classifiers. pages 616–623, 2003. doi:10.1.1.13.8572.
- [93] Ashley Walker Robert Fisher, Simon Perkins and Erik Wolfart. Thinning. *Hypermedia Image Processing Reference (HIPR2)*, 2003.
- [94] Jos B.T.M. Roerdink and Arnold Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 2001. doi:10.1.1.10.3852.
- [95] John Collomosse Rui Hu, Mark Barnard. Gradient field descriptor for sketch based retrieval and localization. *2010 IEEE International Conference on Image Processing*, pages 1025–1028, 2010. doi:10.1109/ICIP.2010.5649331.
- [96] S. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. 2nd ed. edition, 2003.
- [97] Alex G. Hayes Ryan C. Ewing, George D. McDonald. Multi-spatial analysis of aeolian dune-field patterns. *Journal of Geomorphology*, 240(1):44–53, 2014. doi:10.1016/j.geomorph.2014.11.023.
- [98] Gary A. Kocurek Ryan C. Ewing. Aeolian dune interactions and dune-field pattern formation: White sands dune field, new mexico. *Sedimentology*, 57:1199–1219, 2010. doi:10.1111/j.1365-3091.2009.01143.x.
- [99] Gary Kocurek Ryan C. Ewing, Aymeric-Pierre B. Peyret and Mary Bourke. Dune field pattern formation and recent transporting winds in the olympia undae dune field, north polar region of mars. *Journal of Geophysical Research*, 115(E8), 2010. doi:10.1029/2009JE003526.
- [100] C. Meena S. Malathi, S. Uma Maheswari. Fingerprint pore extraction based on marker controlled watershed segmentation. *Computer and Automation Engineering (ICCAE)*, 2010. doi:10.1109/ICCAE.2010.5451426.
- [101] H. Samet and M. Tamminen. Efficient component labeling of images of arbitrary dimension represented by linear bintrees. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(4):579, 1988. doi:10.1109/34.3918.
- [102] Wah Chiu Tao Ju Sasakthi Abeysinghe, Matthew Baker. Segmentation-free skeletonization of grayscale volumes for shape understanding. *IEEE International Conference on Shape Modeling and Applications*, pages 63–71, 2008. doi:10.1109/SMI.2008.4547951.
- [103] M. Sezgin and B. Sankur. Survey over image thresholding techniques and quantitative performance evaluation. *Journal of Electronic Imaging*, 13(1):146–165, 2004. doi:10.1117/1.1631315.
- [104] Qing Guo Fangmin Dong Shuai Wanh, Shuifa Sun and Chunyan Zhou. Image edge detection based on rotating kernel transformation. *7th International Congress on Image and Signal Processing*, 2014. doi:10.1109/CISP.2014.7003813.
- [105] Xiangrong Zhang Shuang Wang, Xuili Ma and Licheng Jiao. Watershed-based textural image segmentation. *International Symposium on Intelligent Signal Processing and Communications Systems*, 2007. doi:10.1109/IITA.2008.285.

- [106] J. M. Brady S.M. Smith. Susan - a new approach to low level image processing. *International Journal of Computer Vision*, 1997. doi:10.1023/A:1007963824710.
- [107] Irwin Sobel. History and definition of the sobel operator. 2014.
- [108] M. Pietikäinen T. Ojala and D. Harwood. Performance evaluation of texture measures with classification based on kullback discrimination of distributions. *Proceedings of the 12th IAPR International Conference on Pattern Recognition*, 1:582–585, 1994. doi:10.1109/ICPR.1994.576366.
- [109] M. Pietikäinen T. Ojala and D. Harwood. A comparative study of texture measures with classification based on feature distributions. *Pattern Recognition*, 29:51–59, 1996. doi:10.1016/0031-3203(95)00067-4.
- [110] S. Ghosh T. Stepinski, R. Vilalta. Machine learning tools for automatic mapping of martian landforms. *IEEE Intelligent Systems*, 22(6):100–106, 2007. doi:10.1109/MIS.2007.114.
- [111] C. Y. Suen T. Y. Zhang. A fast parallel algorithm for thinning digital patterns. *Communication of the ACM*, 27(3):236–239, 1984. doi:10.1145/357994.358023.
- [112] G. Medioni G. Lee M.S. Tang, C.K. N-dimensional tensor voting, and application to epipolar geometry estimation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (23):829–844, 2001. doi:10.1109/34.946987.
- [113] Jeffrey Brokish Trish Keaton. A level set method for the extraction of roads from multispectral imagery. *Applied Imagery Pattern Recognition Workshop*, 2002. doi:10.1109/AIPR.2002.1182268.
- [114] Tinne Tuytelaars and Krystian Mikolajczyk. Local invariant feature detectors: A survey. *Foundations and Trends in Computer Graphics and Vision*, 3(3):177–280, 2007. doi:10.1561/0600000017.
- [115] C. R. Twidale. The simpson desert, central australia. *South African Geographic Journal*, 65:3–17, 1980. doi:10.1080/03736245.1980.10559618.
- [116] Viola and Jones. Rapid object detection using a boosted cascade of simple features. *Computer Vision and Pattern Recognition*, 2001. doi:10.1109/CVPR.2001.990517.
- [117] Paul Viola and Michael J. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 2001. doi:10.1.1.110.4868.
- [118] Jakubowicz J. Morel J.M. Randall G. von Gioi, R.G. Lsd: A fast line segment detector with a false detection control. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, (32):722–732, 2010. doi:10.1109/TPAMI.2008.300.
- [119] Wei Zhang Weibin Rong, Zhanjing Li and Lining Sun. An improved canny edge detection algorithm. *IEEE International Conference on Mechatronics and Automation*, 2014. doi:10.1109/ICMA.2014.6885761.
- [120] P. J. Werbos. Beyond regression: New tools for prediction and analysis in the behavioral sciences. 1975.
- [121] Liu Wenyu Xian Bai, Latecki Longin. Skeleton pruning by contour partitioning with discrete curve evolution. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(3):449–462, 2007. doi:10.1109/TPAMI.2007.59.
- [122] Qingbo Geng Xiaobing Wang, Baokui Li. Runway detection and tracking for unmanned aerial vehicle based on an improved canny edge detection algorithm. *4th International Conference on Intelligent Human-Machine Systems and Cybernetics*, 2012. doi:10.1109/IHMSC.2012.132.
- [123] A. Belyaev Y. Ohtake and H.-P. Seidel. Ridge-valley lines on meshes via implicit surface fitting. *ACM Transactions on Graphics (TOG)*, 23(3):609–612, 2004. doi:10.1145/1015706.1015768.
- [124] Zhigang Jin Yu Cheng and Cunming Hao. Illumination normalization based on 2d gaussian illumination model. *International Conference on Advanced Computer Theory and Engineering (ICACTE)*, 2010. doi:10.1109/ICACTE.2010.5579870.
- [125] Hong Zhang Zhijie Wang. Edge linking using geodesic distance and neighborhood information. *International Conference on Advanced Intelligent Mechatronics*, 2008. doi:10.1109/AIM.2008.4601650.