Automated Dune Crestline Detection

David Leblanc

03/13/2015

Abstract

TODO

1 Introduction

TODO

2 Previous Work

2.1 Dune Pattern Studies

In [3], an explanation of how dune-field patterns emerge and addresses the degree of complexity from the standpoint of self-organizing system. 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 meanful data such as junctions, terminations, mergers, linking, and other interesting processes explained in [3].

In later work, [10] studied the dune field pattern formations on the north polar region of Mars. These dunes are thought to be mostly inactive, with relative no movement over a period of 4 to15 martian years. The reason this region is interesting is because there are two nearly orthogonal crestline orientations present in the region. This complex system of superposition of multiple patterns in this example showcase a set of *primary* and *secondary* crestlines which would be valuable data to automatically segment. The *primary* dunes are the largest-scale dunes, extend over the entire lenght of the image set, and contain many Y junctions, an indication of well-organized linear dunes. In contrast, *secondary* dunes are rounded, have least defined features, and are perpendicular to the main primary crestlines.

Other interesting features are the so-called *Slipfaces*. These typically appear along the *primary* crestlines, in areas of intersection with the *primary* and *secondary* crestlines. 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 a 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 [10], have a polygonal shape, and are indicative of ice.

The rest of the paper describes an in depth statiscal analysis of the features to compute the flow fields and understand the geomorphic relationships which are present in the area[1].

In the most recent work, [9] 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 developped 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 are multiple scales. To summarize, being able to detect the crestlines and other types of features at multiple scales is invaluable.

In other related work, [2] 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 cor-

[1]The paper [10] 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.

relation matches found, the migration vector can be computed. The migration data processing can extract the flow fields of the dunes.

2.2 Image Processing

2.2.1 Edge and Line Detection

The Canny edge detector is a well known and used method to retrieve important edge features from an image. In [8], enhancement to the method is proposed by analysing 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 [14] 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 o f 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 [13], 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 chosing the high and low thresolds from the computed mean and standard deviation of the gradient magnitudes.

A line detection and image enhancement has been proposed in [4], for a digital image restoration of heritage art application. The images worked with have been deteriorated over time, and contain many cracks, faint lines, and broken stroke. 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 supression. This process preserves strong lines while removing texture lines. The image is the binarized to remove all remaining weak 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, [11] use rotating edge detection kernels and fuse the responses from the operations. What determines what an edge is a location and a direction, as a non-edge point such as a flat area or noise, which has no specific direction. Following this principle, the response in on 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[2].

An interesting alternative to the classical Hough transform is presented in [5] 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 remote high frequency noise, edges and lines are detected on the sonar based images. To detect potential linear features, they use a Phase Congruency detector[3]. The Hough transform is then used to find linear features, and some criterias are determined to reject false positives and preserve true positives.

2.2.2 Edge Linking

In [16], an edge linking approach is proposed which using 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 [16] define a windows 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 intensity values of the image, which reportedly gives better results.

2.3 Pattern Recognition

2.4 Automated Feature Detection

In [12], 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

[2]The approach proposed in [11] has definite potential for improved edge detection applied to dune crestline detection. It also has potential for improvement of the method itself

[3]The Phase Congruency detector requires more research to determine what type of features can be extracted from it.

proposed method is semi-automated because a user must provide and 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[4].

In [1], linear features of internal waves are extracted using a technique called *Multiscale Retinex* (MSR) feature extraction[5]. 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 accross 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.

There has been some work done in automated dune detection. In [6], 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 cells 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 labelled 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 [6], 230 labelled 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 largers cells will affect localization accuracy of the dunes. If the application requires higher localization accuracy, this type of surpervised learning approach may not be suitable.

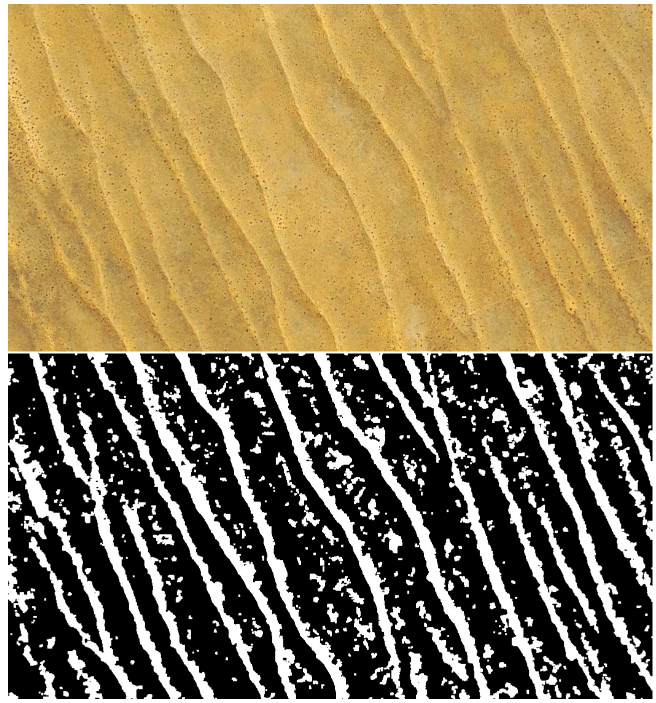


Figure 1: Example of an adaptive thresholding method applied on a dune crest image with poor illumination and contrast.

3 Methodology

3.1 Dune Detection

3.1.1 Method #1: Adaptive threshold and contour detection

Adaptive Threshold Image Process: Dune crest images typically have a shaded and a light side. Depending on the conditions and image quality, the actual edge or dune crest line may not be clearly defined. Some image processing typically improves the quality and allow the extraction of the dune crest lines.

Many approaches revolve around edge detection, but because the conditions and image quality can vary, edges may not be accurately extracted. In this approach, we use a thresholding method to extract the light side of a dune. Using a fixed threshold may not be sufficient to extract the dunes since the illumination across an image is usually not uniform. An adaptive threshold can be a useful tool to threshold an image with non-uniform illumination. An example of this type of thresholding is shown in Figure 1.

[4] The skeletonization procedure used in [12] may be useful for extracting the dune crestlines and junction points from segmented dune regions.

[5] Although the problem set in [1] is different, there appears to be significant overlap and comparison between dune crestline detection and oceanic internal wave detection.

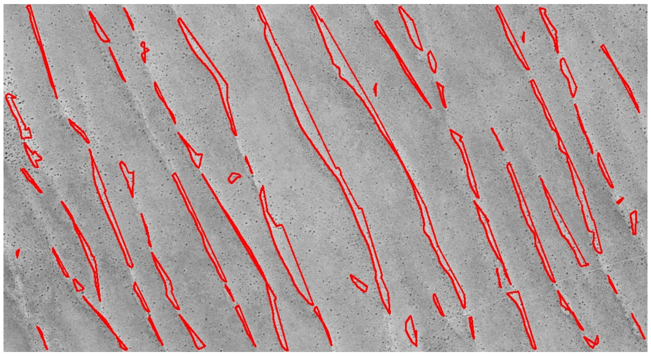


Figure 2: Contour extraction from binary processed image

Dune Crest Line Extraction: Once the image has been preprocessed, the bright regions can extracted from the resulting binary image. As shown in figure 1, the bright regions shown consist of the sun illuminated side of the dunes and some noise.

From the regions, contours can be extracted, which can be filtered by size of the region. Smaller regions are considered as noise, and can be rejected from the potential dune candidates. The remaining region contours are used as potential candidates for dune crest lines. From the coutours, figure 1 shows that typically one of the edges of the contour is located very nearly on the dune crest line. On the opposite side of the region is essentially the base of the dune. Figure 2 shows the results of the extracting the contours from the processed image.

Because the regions tend to be noisy, the contours are therefore smoothed using a Gaussian kernel, define as:

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

The Gaussian kernel is convolved on the contour, which a collection of two dimentional points (x,y). The result of this convolution is a smoother contour which can be more reliably used as a dune crest line candidate.

Typically, the dune crest lines are located on an area in the image where the derivatives are larger, or consistant 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 consistantly high gradients. To determine if a point $P(i)$ on a contour belongs on the dune crest line, we use the following criteria

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

where

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

and $0 < r \leq 1$, which is the ratio of how many strong consistant 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). More work needs to be done in this area.

3.1.2 Method #2: Unsupervized Edge-based line detection

Edge Detection Image Processing: This approach is based on the gradient information detect on the image. Dune crestlines typically have relatively large gradient magnitudes. There are a few challenges with extracting edges of the dune satelite images:

1. Images tend to be noisy and very textured, which results in many edges of varying orientations and magnitudes.

2. Many of the stronger edges may not necessarily be the crestlines themselves. Generally, dunes cast relatively large shadows which themselves produce relatively strong edges.

3. Determining the parameters and appropriate scale for the parameters of the edge detector can be sensitive and affect detection rates.

To address these concerns, the Canny edge detector is used along side some of the techniques inspired from [13] and [14]. First, in order to reduce the amount of noise in the images, both a median and Gaussian filtering is applied to the image. The reasoning for 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.

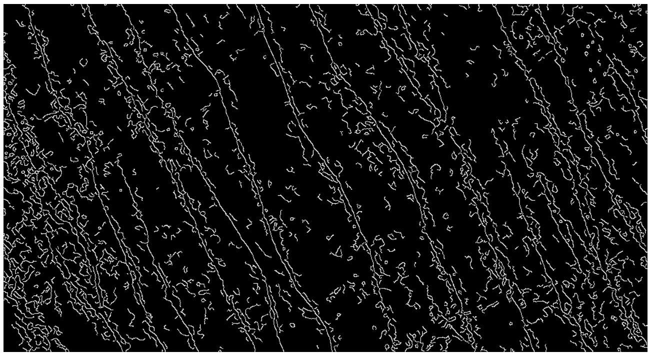


Figure 3: Canny edge detector after median and Gaussian filtering of the Kalahari dune image.

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

$$T_{high} = \mu + q \cdot \sigma$$

$$T_{low} = \frac{T_{high}}{2}$$

where q is so factor which can be determined based on the problem set. This makes the threshold parameter less sensitive to fixed threshold problems. The result of the Canny edge detector after filtering are displayed in figure 3.

Unsupervised Crestline Detection from Detected Edges

Many of the edges provided in the edge detection step are noise, shadows, and non crestline textures. Crestlines all have one feature in common: all of their gradient have relatively similar orientations. Dune crests typically appear to have larger gradient magnitures relative to other noisy edges or texture (but this may not always be the case).

Similarly, because of the position at which the sun, edges generated from the shadows of the dunes will also have similar orientations within themselves. Given these properties, there should be a clear deliniation between crestline edges, and non-crestline edges.

Using KMeans: Resolving these differences by grouping similar edges together can be achieved by clustering the gradients. There have been many approaches to clustering (REFERENCES NEEDED HERE???), but the most common is the K-Means algorithm. The benefits of using a clustering algorithm is that it is unsupervised, so that the differences should form themselves from the edge data itelf.

Typically, the main parameter that is required for K-Means is the value K, which is the number of clusters. In this problem set, the goal is to seperate the crestline group from all other non-crestline edges. Therefore, the parameter is simply set to $K = 2$, to seperate the gradients into two distinct groupings[6].

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:

$$\bar{\mu} = \frac{\sum_{i=0}^{P} \sqrt{\delta_{x_i}^2 + \delta_{y_i}^2}}{P}$$

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 ith point. Each gradient is then normalized by dividing by the average magnitude $\bar{\mu}$.

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

The set of normalized gradient vectors are then clustered using the K-Means algorithm with $K = 2$. In Figure 4, the results of the clustering method are shown. The blue points are gradient vectors which potentially belong to the dune crestline. 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 set which contains the crestline points.

The centroid values for each cluster are displayed in 4, which clearly indicates that the blue cluster contains the stronger edges, which are more than likely crestlines. To prove this, figure 5a. shows the results of rejecting all weaker cluster gradients, and keeping the dominant cluster gradients.

Once the weak edges have been filtered out, the remaining edges are used as crestline 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 connected components technique, these smaller less significant edge segments

[6] So far, only using two clusters has been investigated. Using more clusters may have more valuable information and better seperation.

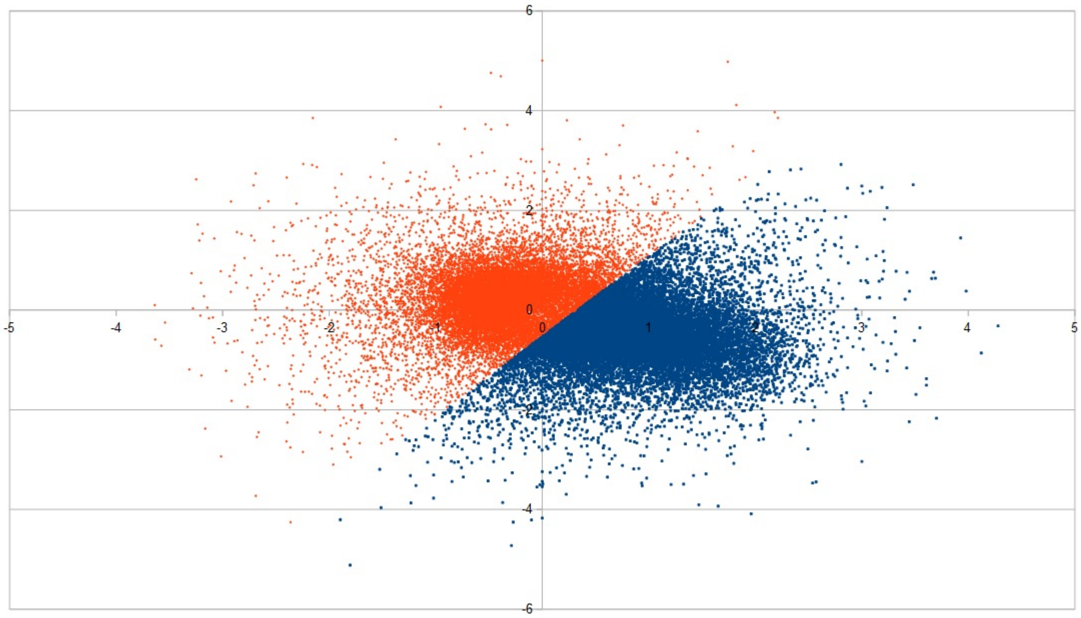
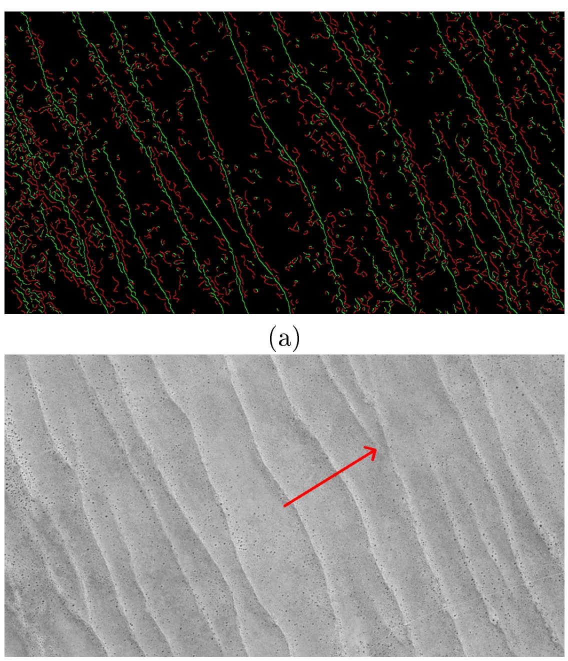


Figure 4: Results of K-Means clustering algorithm with $K = 2.$, on the edge gradients detected using Canny, from Kalahari. The blue cluster represents the dune crestline gradients, and the red represents all other non-crestline edges. The cluster centers of blue is (0.8958, -0.5496), and the cluster center of the red is (-0.4111, 0.2705).



(a)

(b)

Figure 5: (a) Results of filtering out edges (red) which belong to the cluster with the weaker edges. Potential crestline edges (green) are preserved for use. (b) Average orientation of the dominant cluster, which correctly appears to be perpendicular to the trend of the crestlines.

can be removed, keeping only longer contiguous segments[7].

Results of this processing is displayed in section 5.

Using Histograms of Gradients: An alternative to the clustering method is to compute a histogram of the gradients based on orientation (REFERENCE NEEDED!!!). 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 crestlines. The approach poses two 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 6 a value of $N = 18$ is used[8]

2. Multiple Peaks: Often, images may have multple peaks in the histogram, but only one of the peaks

[7] More research can be done here. There may be techniques used to fuse smaller segments into larger ones to achieve higher detection accuracy

[8] The optimal value for N has not yet been determined, but at first glance, smaller values for N will cause issues, while larger values seem to improve results.

should represents the true dominant orientation of crestlines.

One assumption made is that the bin with the largest value represents the orientation of the crestline 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 6. In this particular case, the are two major peaks in the histogram, where the stronger is not the part of the crestline edge group, and the weaker one is.

Choosing the higher peak will cause invalid crestlines to be chosen. In order to determine which peak best represents the crestline 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:

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

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 ith point. The mean orientation is computed from the mean vector as:

$$\bar{\theta_\mu} = \arctan\left(\frac{\mu_{\bar{y}}}{\mu_{\bar{x}}}\right)$$

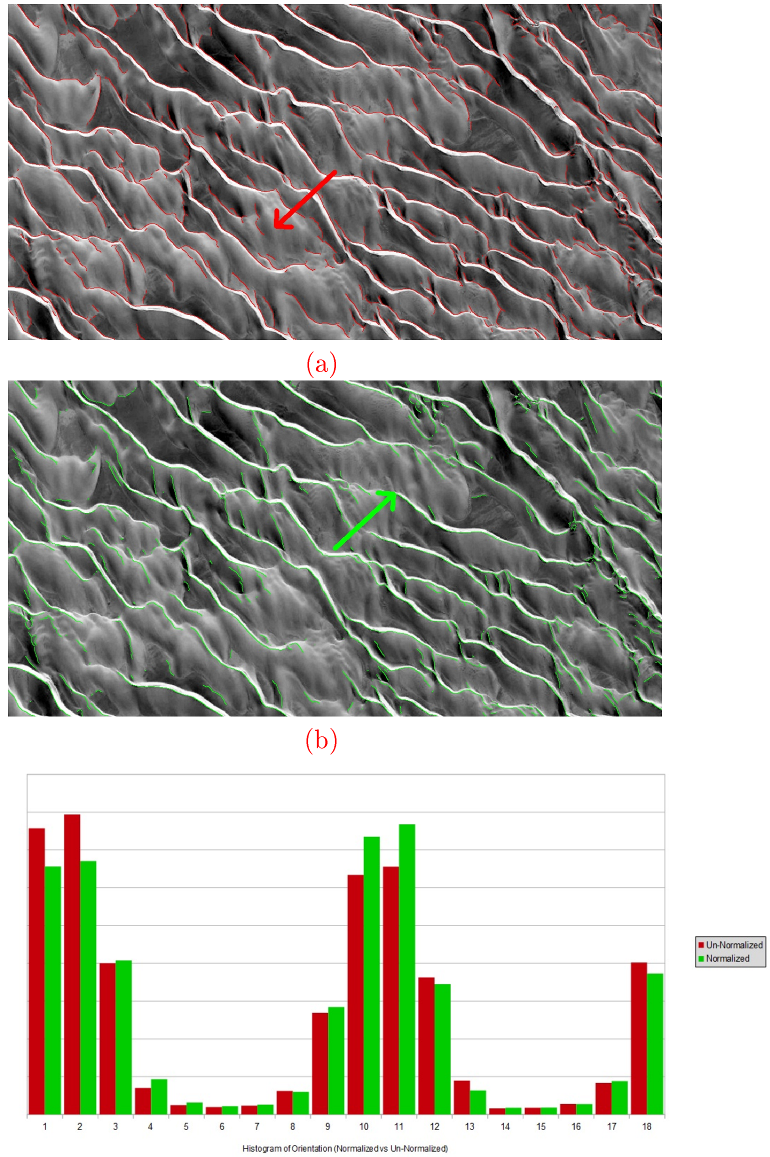
The gradients are then normalized by simply subtracting the mean vector from each gradient:

$$\dot{\delta}\left(x_i, y_i\right) = \left(\delta_{x_i} - \mu_{\bar{x}}, \delta_{y_i} - \mu_{\bar{y}}\right)$$

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

$$\dot{\theta}_i = \arctan\left(\frac{\dot{\delta_{y_i}}}{\dot{\delta_{x_i}}}\right) - \bar{\theta_\mu}$$

To determine which bin the ith edge point belongs to, we simply calculate $\left\lceil \frac{\dot{\theta}_i . N}{2\pi} - 0.5 \right\rceil$, and increment that bin by the magnitude of the normalized gradient by $\sqrt{\dot{\delta}_{x_i}^2 + \dot{\delta}_{y_i}^2}$. In essense, this normalization process removes the uneven skew of the gradients in the overall image. Removing this skew allows true crestline edges to be fairly compared with other stronger edges. As shown in Figure 6, the normalization process softens the stronger dominant edge and enhances the true crestline edges. This process enables crestlines to be accurately detected in images where the valleys of dunes are sharp and contain strong edges[9].



(a) (b) (c)

Figure 6: Results of unnormalized (red) and normalized (green) histograms of gradients ($N = 18$). The histogram contains two major peaks, but the crestline is the weaker of the two peaks. (a) Edges which belong to the dominant orientation whic are invalid crestlines. (b) With normalization, the true crestlines now have the higher overall magnitude. (c) The 18 bin histogram of gradients with and without normalization.

[9]The gradient normalization process has not be proven to work for all cases, only for the images from the test set. There

The dominant orientation can be used to filter out most non-crestline edges by keeping all edges using the following criteria:

$$P(x_i, y_i) = \begin{cases} 1 & \left\| \arctan\left(\frac{\delta_{y_i}}{\delta_{x_i}}\right) - \Phi \right\| \leqq \varphi \\ 0 & otherwise \end{cases} \tag{1}$$

P is a crestline candidate if it satisfies the orientation criteria, where φ is an angle tolerance value which is experimentally set to $\varphi = \frac{\pi}{2}$[10] .

In order to improve the robustness, a connected components method is used to find all segments from the canny image. For each segments if the number of points along the segment satisfy the criteria from 1 is greater than some ratio R, then the segment is considered to be a dune segment, where $0 < R < 1$. Choosing a small R generally improves the true positive detection, while increasing false positive rate[11].

3.2 Dune Morphology

With the reliable crest line data extracted, morphology parameters of the target image can be computed. The desired properties include crest line lenght, dune scale, distance between dune crests, and other properties. TODO (more research on this later).

4 Experiments

Since the goal of this project is to automatically segment dune crest lines, a dataset of images with labeled ground truth is used to measure the accuracy of the dune detection algorithm. The current dataset is a small sample set of six images taken from different regions, at various scale and various illuminations. The regions include: Kalahari, Namib, Simpson, Skeleton Coast, WDC, and White Sands, as shown in figure 7.

TODO.

5 Results

The dune line crest detector is applied to the labeled benchmark images shown in figure 7. The results we are interested in for detection purposes are the true positives (correctly detected dune lines) and false positive (segments that are not dunes). The results are computed in a brute force manner:

1. **True Positives**: The TPs are computed by taking each ground truth pixel, and finding the nearest detected dune segment pixel. If the distance between the nearest detected and the ground truth is smaller than some error e, the ground truth is considered to be detected. The TP rate is then the number of correctly identified ground truth points, over the total number of ground truth points.

2. **False Positives**: The FPs are computed by taking each detected dune candidate, and finding the nearest ground truth point. If that distance is greater than some error e, then the detected dune candidate is a false positive. The FP rate is then expressed as the number of incorrectly detected dune points over the total number of dune points detected.

The TP and FP rates are shown in table 1. The results shown in figure 8 visually overlaps the ground truth with the dune detected.

6 Conclusion

TODO.

References

[1] 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. *???????????????*, ??????????????

[2] 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*, 2005.

[3] Ryan C. Ewing Gary Kocurek. Aeolian dune field self-organization - implications for the formation of simple versus complex dune-field patterns. *Geomorphology*, 2005.

[4] 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.

may be cases where the normalization may have the opposite effect and remove all true crestline. More investigation and sample images are required to determine the effectiveness of the normalization.

[10] The value of φ has not been extensively tested yet, and may have an effect on the results, more testing is required to determine the sensitivity of this parameter.

[11] During initial testing, R (~0.1) was set to a very low value, which allowed many of the true positive to be found at the expense of a higher false positive rate. This technique is very simplistic and does require more investigation to improve on.

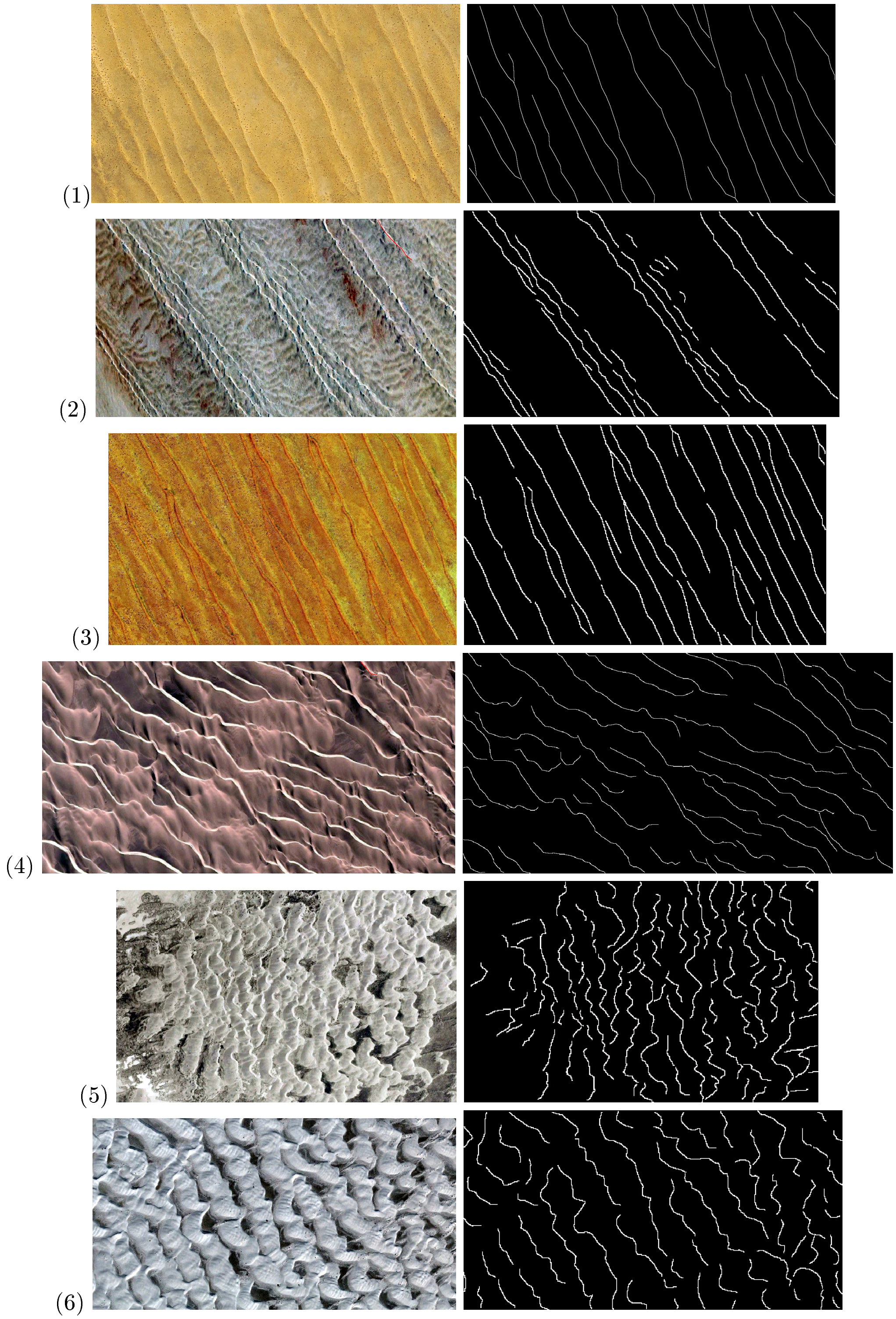


Figure 7: Dune dataset with corresponding dune crest line ground truth map. The images are labeled as (1) Kalahari, (2) Namib, (3) Simpson, (4) Skeleton Coast, (5) WDC, and (6) White Sands.

Images	Method #1		Method #2 (K-Means)		Method #2 (HoG)	
	TP	***FP***	***TP***	***FP***	***TP***	***FP***
Kalahari	0.7094	**0.2178**	**0.9396**	0.2495	0.9852	0.7155
Namib	0.7493	0.6026	**0.9281**	**0.4924**	0.9626	0.6850
Simpson	0.1364	0.6512	**0.7464**	**0.4979**	0.8977	0.6478
Skeleton Coast	**0.8033**	0.3253	0.7605	**0.1517**	0.8187	0.2178
WDC	**0.5581**	**0.3731**	0.5111	0.4849	0.7478	0.5399
White Sands	0.2614	0.6926	**0.6231**	**0.4762**	0.7634	0.5740

Table 1: Results of the dune detection algorithms for the benchmark dataset using a value of $e = 5$ pixels. Method #2 used $R = 0.1$

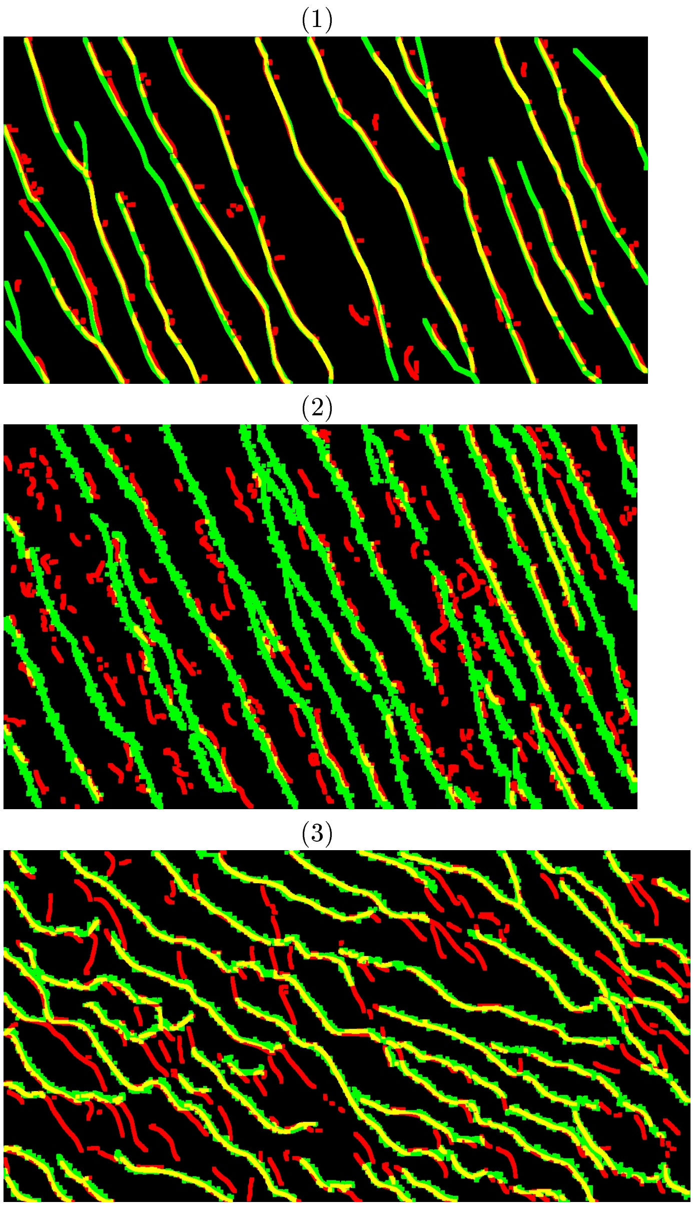


Figure 8: Dune detection results (red) compared to the ground truth (green), overall is shown in yellow. Shown are results for (1) Kalahari, (2) Simpson, and (3) Skeleton Coast benchmark images

[5] Jason C. Isaacs and Ross Goroshin. Automated cable tracking in sonar imagery. *????????????????????*, ???????????????????

[6] Jose Saraiva Lourenco Bandeira, Jorge S. Marques and Pedro Pina. Advances in automated detection of sand dunes on mars. *Earth Surface Processes and Landforms*, 2013.

[7] David G. Lowe. Organization of smooth image curves at multiple scales. *????????????????????*, 1988.

[8] Lei Zhang Paul Bao and Xiaolin Wu. Canny edge detection enhancement by scale multiplication. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 2005.

[9] Alex G. Hayes Ryan C. Ewing, George D. McDonald. Multi-spatial analysis of aeolian dune-field patterns. *Journal of Geomorphology*, 2014.

[10] 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*, 2010.

[11] 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.

[12] Jeffrey Brokish Trish Keaton. A level set method for the extraction of roads from multispectral imagery. *Applied Imagery Pattern Recognition Workshop*, 2002.

[13] Wei Zhang Weibin Rong, Zhanjing Li and Lining Sun. An improved canny edge detection algorithm. *IEEE International Conference on Mechatronics and Automation*, 2014.

[14] 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.

[15] Xueliang Zhang Xiaomin Ji and Lifeng Zhang. Sequential edge linking method for segmentation of remotely sensed imagery based on heuristic search. *????*, ????

[16] Hong Zhang Zhijie Wang. Edge linking using geodesic distance and neighborhood information. *International Conference on Advanced Intelligent Mechatronics*, 2008.