Why edge detection?
???
So, we have motivated why use edge detection for the detection of sand ripples.
How will we be using edge detection?
What makes edge detection the right tool for this job?

"*Another important question is the reliability of the edge estimates. We do not only want to find an edge but also to know how significant it is. Thus, we need a measure for edge strength. Closely related to this issue is the question of optimum edge detection. Once edge detectors deliver not only edges but also an objective confidence measure,* ***different edge detectors can be compared to each other and optimization of edge detection becomes possible****.*" - B. Jähne, Digital Image Processing ISBN 3–540–67754–23, 2002 by Springer-Verlag.
Once we have accomplish(or incorporate it in from the beginning) ed detection of sand ripples and maybe even removed them, the maybe we can even do further analysis. Further analysis in-terms-of compare different edge detectors.

"*Edge detection is always based on differentiation in one or the other form. In discrete images, differentiation is replaced by discrete differences, which only approximate to differentiation. The* ***errors*** *associated with these approximations require careful consideration. They cause effects that are not expected in the first place. The two most serious errors are: anisotropic edge detection, i. e., edges are not detected equally well in all directions, and erroneous estimation of the direction of the edges.*"
These errors can be calculated.

Edge detection can be done in various ways.

- 1st derivative - gradient based edge detection
- 2nd derivative - Laplacian - zero crossing
- Regularized edge detector - Sobel edge detector
- Regularized edge detector - Canny edge detector (using Gaussian filter for smoothing)

What about noise?
Noise inside an image, applying these edge detectors on a noise image has

some effects on the results. How would we deal with this?

"filters that perform a derivation in one direction but also smooth the signal in all directions. Smoothing is particularly effective in higher dimensional signals be- cause smoothing in all directions perpendicular to the direction of the gradient does not blur the edge at all. Derivative filters that incorporate smoothing are also known as regularized edge detectors because they result in robust solutions for the ill-posed problem of estimating derivatives from discrete signals."

e-magnitude & e-theta a measures that compare various edge detectors in-terms-of there errors: magnitude & angle errors respectively.

Patrick (based on an initial comment of mine relating to steering filter implementation):

"not sure how the matlab node works, but what you want to do is find the average angle of all the edge pixels in the image. If ripples dominate, and they lie at a rough angle of x, then you should be able to find x (+- error). If your run an edge filter on the image, it will give, per pixel, a gradient magnitude (edge strength) and a gradient angle (basically the angle of greatest change across the edge, so edge direction). You can then create a histogram over the range of angles, maybe bins a few degrees wide, and throw each pixel into the bin in which it's edge angle falls. You may need to set a gradient threshold for magnitude, and just ignore pixels which correspond to weak edges. When the count the pixels in each histogram bin, there should be one which has a maximum : this should be the approximate angle of the dominant edge orientation. Note that the ripples have changing edge orientation, so there will be a spread of angles. So you would need to look at a range of angles around the peak in the histogram. To see if this did prove useful, you could plot back the pixels you put into those bins in a color and see where they cam from in the image. You'd need to do more processing to select only ripple pixels, but this gives you information too help.

Does that make sense? This doesn't really require any not complex filtering, just a good edge detector . "

- Detect angles:

- Edge operator
- Gradients
- Gradient magnitude = edge strength
- Gradient direction = edge direction + 90 degrees
  - Detect/find angles
    - Find all edge angle pixels
      - With ripples dominating images, the bulk of detected edge pixels belong to ripples
      - Histogram of angles
        - A histogram is a graphical representation of the distribution of numerical data
        - **the first step** is to "bin" the range of values (bin edge pixels)
          - that is, divide the entire range of edge pixel values into a series of intervals—and then count how many edge pixel values fall into each interval
          - Bins - a few degrees wide
          - Categorise (using bins) pixels relating to detected angles
        - Gradient magnitude threshold - to ignore weak gradient magnitudes
    - To box detected gradient?
      - Look for degrees either above or below threshold
        - To pickup only strong edge magnitudes (ignore weak magnitudes)
  - Count pixels in bins
    - Find maximum
      - approximate angle of the dominant edge orientation
- Detected angles - wide range of angles detected - spread of angles
  - A number of bins will exist to reflect wide range of angles detected
- look at a range of angles around the peak in the histogram