Monitoring change between time-lapsed surveys using Spatial Cross Correlation

Garret P. Duffy* and John E. Hughes Clarke

Ocean Mapping Group, Department of Geodesy and Geomatic Engineering, University of New Brunswick, Fredericton, New Brunswick, CANADA, E3B 5A3.
E-mail: garret@omg.unb.ca; Fax: +1-506-4534943

Abstract

Six multibeam surveys of a headland-associated sand bank outside Saint John, NB were carried out every month from April to October 2002. The dynamic nature of the sand dunes was previously noted by analysis of four successive GSC cruises in 2000 and 2001. Quantifying the amount and direction of sand dune migration has proved an interesting problem and spatial cross correlation, which quantifies and locates a region of maximum similarity between two 2 dimensional spatial variables, seems to be a natural technique to choose. 8-bit sun illuminated images created from digital terrain models were chosen as the spatial variables so the correlation was executed over this representation of the shape of the seabed rather than a bathymetric surface. Vectors are drawn depicting the movement of a sand dune from time t_0 towards the peak in spatial correlation at time t_1. The two main parameters of interest are the window dimension and the step size and these must be selected for optimum resolution of movement with minimum aliasing. Application of this method shows that migration rates and directions vary substantially across the sand bank with increased migration in the shallowest part of the sand bank closest to the headland.

1 Introduction

Discrete correlation methods, both auto-correlation and cross-correlation methods, are widely applied to analyse temporal, spatial, and spatio-temporal data series. An example of the former is the measurement of time delay in an electronic system: simply by cross correlating the input signal with the output signal and locating the peak in the time series cross-correlogram the time delay reveals itself (Bendat, 1971). The cross-correlation operation simply identifies where, in time or space, two (*N*-dimensional) datasets are most 'similar'. Similarity in the cross-correlation is represented by a local maximum of the cross-correlation coefficient in the region of interrogation.

Spatial cross-correlation may be done in the space- or spatial frequency domains. Space domain cross-correlation is computationally very demanding so the Fast Fourier Transform may be used. However, with today's high-speed CPU's, the need for transformation into frequency space is becoming less imperative and spatial cross-correlation is routinely done by 'brute force' in the space domain (McKenna, 2002).

Two unusual applications of spatial correlation are in motion detection in fluids (Jambunathan, 1995) and the movie industry (Lewis). Scientists and engineers in the field of fluid dynamics suspend particles in a moving liquid and then film the motion of the suspended particles in the vicinity of obstacles. The movies are then analysed frame by frame to elucidate velocity information. In the movie industry special effects designers use cross-correlation techniques to do 'feature extraction', i.e. the cross-correlate each frame of the movie with a known pattern, e.g. the representation of an actor, remove the part of the image with high correlation and replace with something else.

This paper will concern itself with a new application of cross-correlation to consecutive digital terrain models (DTM's) of actively migrating sand dunes in Mispec Bay, Saint John, New Brunswick. Section 2 will deal with the mathematics of cross-correlation and the implementation of the Normalised Cross-Correlation Algorithm in C code. Section 3 of this paper will give a background to the datasets used in this paper. Section 4 will give some sample outputs with varying input parameters and discuss the importance of selecting suitable window sizes and search regions. Section 5 will be discussion and conclusion.

2 Theory

Spatial cross-correlation basically locates the point where two spatial datasets are most similar. Mathematically the concept of similarity is quantified as distance, more specifically the Euclidean distance:

$$d^2_{f,g}(k,l) = \sum_x \sum_y \left[f(x,y) - g(x-k, y-l)\right]^2$$

Eq. 1

Where *f(x,y), g(x,y)* are discrete spatial variables and the Euclidean distance is calculated at every point *(x,y)* in the space for all values of *k* and *l.* Where the minimum Euclidean distance exists, the spatial functions *f, g* are most similar with regard to the quantity measured by *f* and *g,* e.g. the 8 bit sun illuminated bathymetry of two successive DTM's. To clarify, we can regard the function *f(x,y)* as a reference array of elements, e.g. a 32 by 32 pixel window of a DTM at time t_2, and *g(x,y)* as some pattern that we want to find in the reference array, e.g. a 16 by 16 pixel sub window of the same region as *f* but at an earlier time t_1.

So what is the relationship between the Euclidean distance and cross-correlation? Expansion of Equation 1 yields:

$$
\begin{aligned}
& d_{f,g}^2(k,l) = d_1(k,l) - 2d_2(k,l) + d_3(k,l) \\
& \textit{where} \\
& d_1(k,l) = \sum_x \sum_y \left[f(x,y) \right]^2 \\
& d_2(k,l) = \sum_x \sum_y f(x,y) g(x-k, y-l) \\
& d_3(k,l) = \sum_x \sum_y \left[g(x-k, y-l) \right]^2
\end{aligned}
$$

Eq. 2

The term d_3 represents the summation of the template energy and is constant valued independent of where the Euclidean Distance calculation is being carried out, defined by (*k,l*). The term d_1 represents the energy of *f* over the window defined by *g* centered at (*k,l*) and generally varies rather slowly with (*k,l*) (Pratt, 1991; Lewis). The remaining term, d_2, is the "cross-correlation" and a high $d_2(k,l)$ value indicates a high degree of similarity between *f* and *g* at that lag value (*k,l*). So we see that the cross correlation may simply be viewed as a term in the Euclidean Distance Metric. However, there are two disadvantages to using d_2 for pattern identification:

- If the image energy, d_1, varies substantially with position (*k,l*) the correlation of exactly matching regions in *f* and *g* measured by d_2 may be *less* than the d_2 measured when *g* overlaps a high energy region ("bright spot") of *f.* That is to say the assumption that *f(x,y)* is a stationary function is usually false.
- The range of d_2 is not standardized and is dependant on the size of the sought pattern, i.e. it depends on the number of terms in the summation, which depends on the dimensions of *g(x,y).*

Cross correlation is therefore corrected for these two factors (bias and normalisation) to give an expression for Normalised Cross Correlation, or the Correlation Coefficient:

$$\rho_{k,l} = \frac{\sum_x \sum_y \left[f(x,y) - \overline{f_{k,l}}\right]\left[g(x-k, y-l) - \bar{g}\right]}{\sqrt{\sum_x \sum_y \left[f(x,y) - \overline{f_{k,l}}\right]^2 \sum_x \sum_y \left[g(x-k, y-l) - \bar{g}\right]^2}}$$

Eq. 3

In Equation 2, $f_{k,l}$ '*bar'* is the mean value of *f(x,y)* over the range of *g* centered on *(k,l)*; the mean value of *g* is independent of *(k,l)* and so is a constant. The numerator is essentially the covariance evaluated at *(k,l)* and the denominator is the product of the standard deviations of *f* at *(k,l)* and *g*. The correlation coefficient thus described lies in the range $\{-1 \leq \rho_{k,l} \leq +1\}$. This is the expression for the Normalised Cross Correlation coefficient used in this research.

Figure 1 shows graphically how the cross-correlation code is executed in practice. The example shows the situation for a window size, or 'pattern', of 16 by 16 pixels and a search parameter (*k,l* values in Eq. 3) of ±8 pixels in the x and y directions. For execution of the code the green shaded area of 16x16 pixels from the DTM at time t_1 (*f(x,y)*) and the blue shaded search area of 32x32 pixels from the DTM at time t_2 (*g(x,y)*) are read into memory. A single value of $\rho_{k,l}$ is then calculated according to Eq. 3 at each lag value of *k* and *l* from -8 to plus 8 and this value is entered into one of the 17x17 spaces in the cross-correlation array (bottom left hand corner of Figure 1). If the bedforms had not migrated between times t_1 and t_2, there would be a maximum correlation at zero lag value, i.e. at the center of the cross-correlation array. Since migration *has* taken place, the maximum correlation is offset from the zero lag position. This is the position of the bedform at time t_2 relative to the center of windows *f* and *g.* We therefore can draw a vector from the zero lag position to the array value of maximum correlation (in Figure 1, this is $\rho_{3,-3}$). In theory we can then migrate this translation vector anywhere in the 'real world' pixel array, e.g. to the crest of the bedform at time t_1, but for simplicity's sake it is located at the shared centre of the windows *f* and *g*.

The latter paragraph described how a migration vector for a small window of the dtm is calculated. For calculation of the entire migration vector field of the dataset, the cross-correlation algorithm is iterated over the whole data set in such a way that successive *g(x,y)*'s adjoin and do not overlap. Sample output from cross-correlating June and July 2002 is shown in Figure 2.

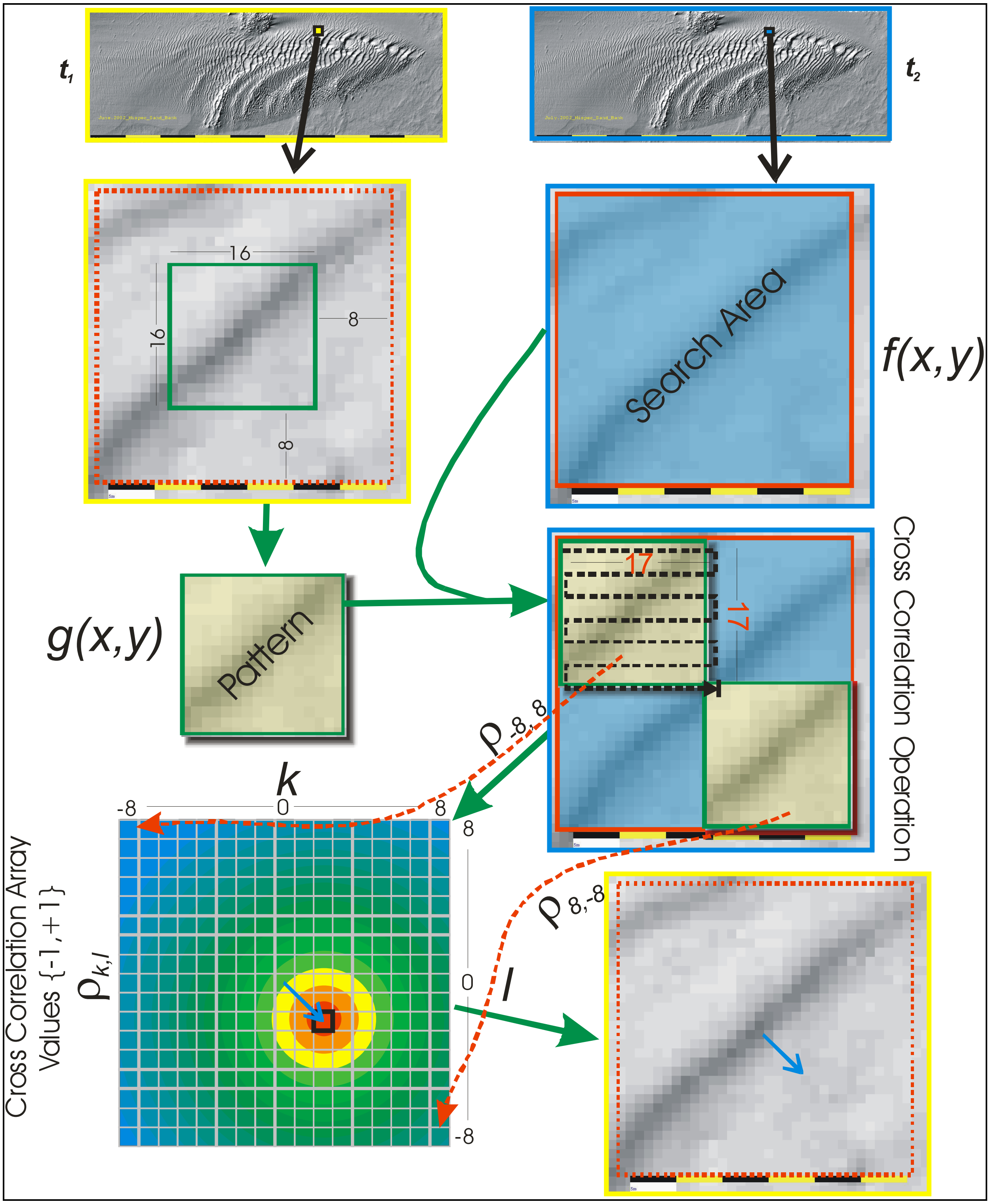


Figure 1 Schematic representation of the execution of the cross-correlation algorithm on a single window of the 8 bit dataset. Explanation in the text.

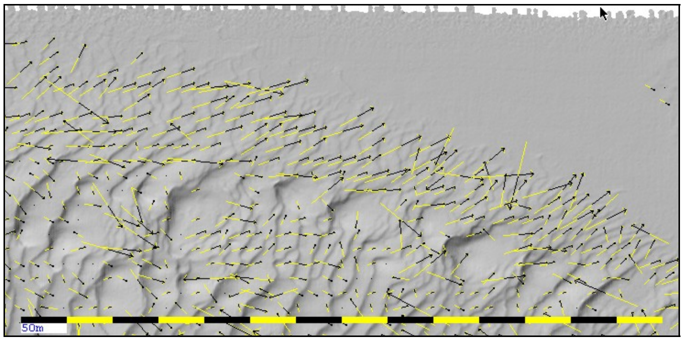


Figure 2 Cross-correlation output showing migration vectors for a window size and search parameter of 16 and 8 respectively.

3 Data Description

Six consecutive EM-3000 multibeam surveys of dynamic sand dunes in Mispec Bay, Saint John, NB were carried out from April to September in 2002. Coastguard differential GPS gave horizontal position and a CHS operated tide gauge at Saint John gave tide data necessary to correct for depth variations due to tide. Vessel attitude and heading were given by a variety of instruments for different surveys such as POS-MV 320, Seapath 200 RTK, Seatex MRU-6 and Octans Gyrocompass.

The EM-3000 measures depths in equiangular sectors port and starboard so the resolution is strongly nadir biased, therefore to ensure a high sounding density and resolution we kept a line spacing of typically 30 m and with an average water depth of around 30 m (with a tidal range of ~ 7 m). Soundings were processed to remove major motion associated artifacts and digital terrain models (DTM's) with a 1 metres resolution were constructed for analysis.

The cross-correlation algorithm implemented in this study is executed on 8 bit data arrays, i.e. pixel integer values from 0 to 255. Therefore the aforementioned 1 metre DTM's (with floating point values 7 to 37 m below chart datum) need to be transformed to an 8 bit DTM. This is simply done by creating a sun-illuminated version of the DTM which is effectively a representation of the shape of the DTM and has *no depth information*. This is fine for the cross-correlation since change in pixel value not due to bathymetry can affect the successful execution of the algorithm.

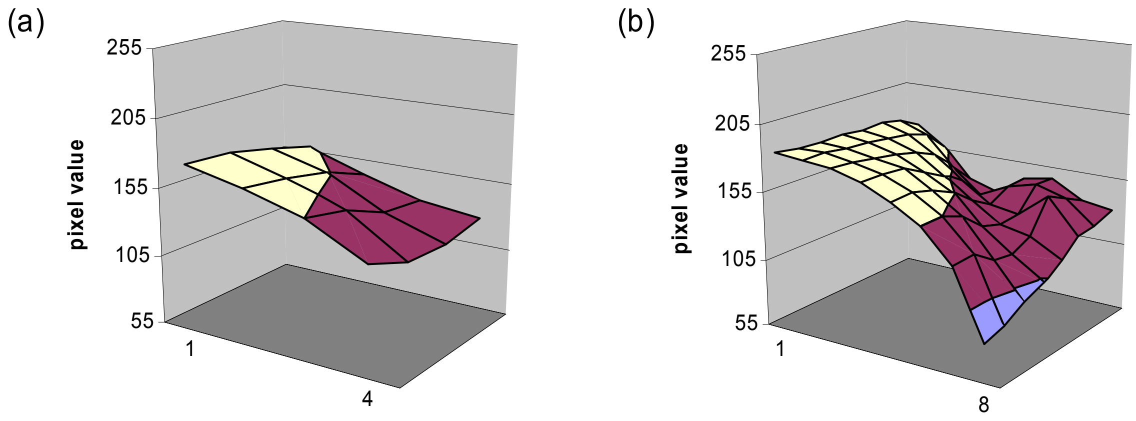
The execution of the code is a processor intensive process with execution times being from 25 minutes (for a window and search parameter of 8 and 8 pixels) to several hours (for parameters of 32 and 32 pixels) for a 1300 by 4400 pixel array.

3.1 Choice of 8 bit Digital Terrain Model

For cross-correlation analysis the input DTM's must show the dynamic features of interest, i.e. the bedforms, in good definition. The sun-illuminated DTM's therefore must be created in such a way so that the bedforms are emphasized. This is done by selecting suitable azimuth and altitude angles of the virtual sun. In this study angles of 115° and 45° respectively gave best definition of the bedforms under investigation.

4 Results

Selecting a suitable window size and search parameter is necessary for the output from the code to realistically show how the bedforms are migrating. The window size must be large enough to capture any systematic 'pattern' of pixels in the image, e.g. stoss slope of a sand dune or the crest or trough of a sand dune, but not too small so that there is no characteristic pattern that the algorithm cannot 'lock on' to (Figure 3(a)) and not too large so that the spatial resolution of migration is not compromised (Figure 3(d)). Implicit in the last sentence is that the window size will affect the total number of migration vectors in the field and therefore the spatial resolution of bedform migration. In practice, this selection procedure done by trial and error, comparing the vector fields for different input parameters and deciding, albeit subjectively, which combination gives the most realistic result.



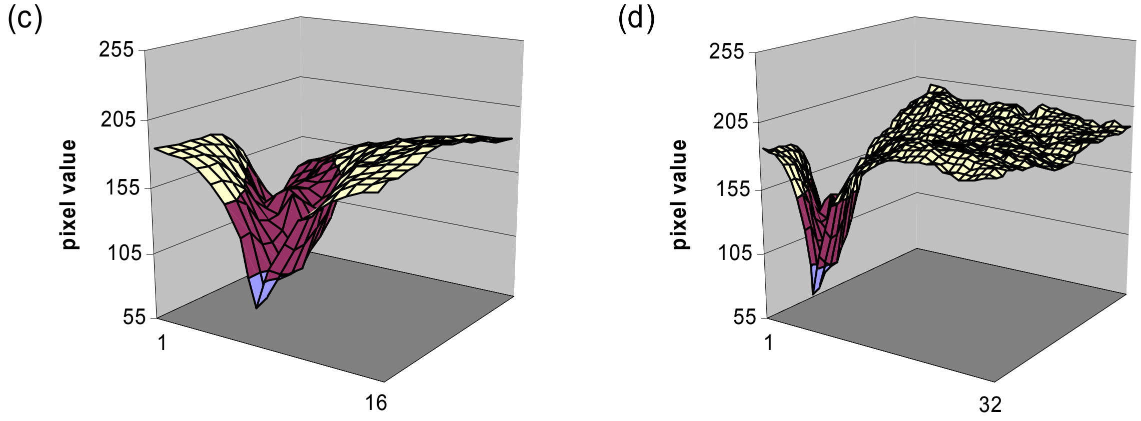
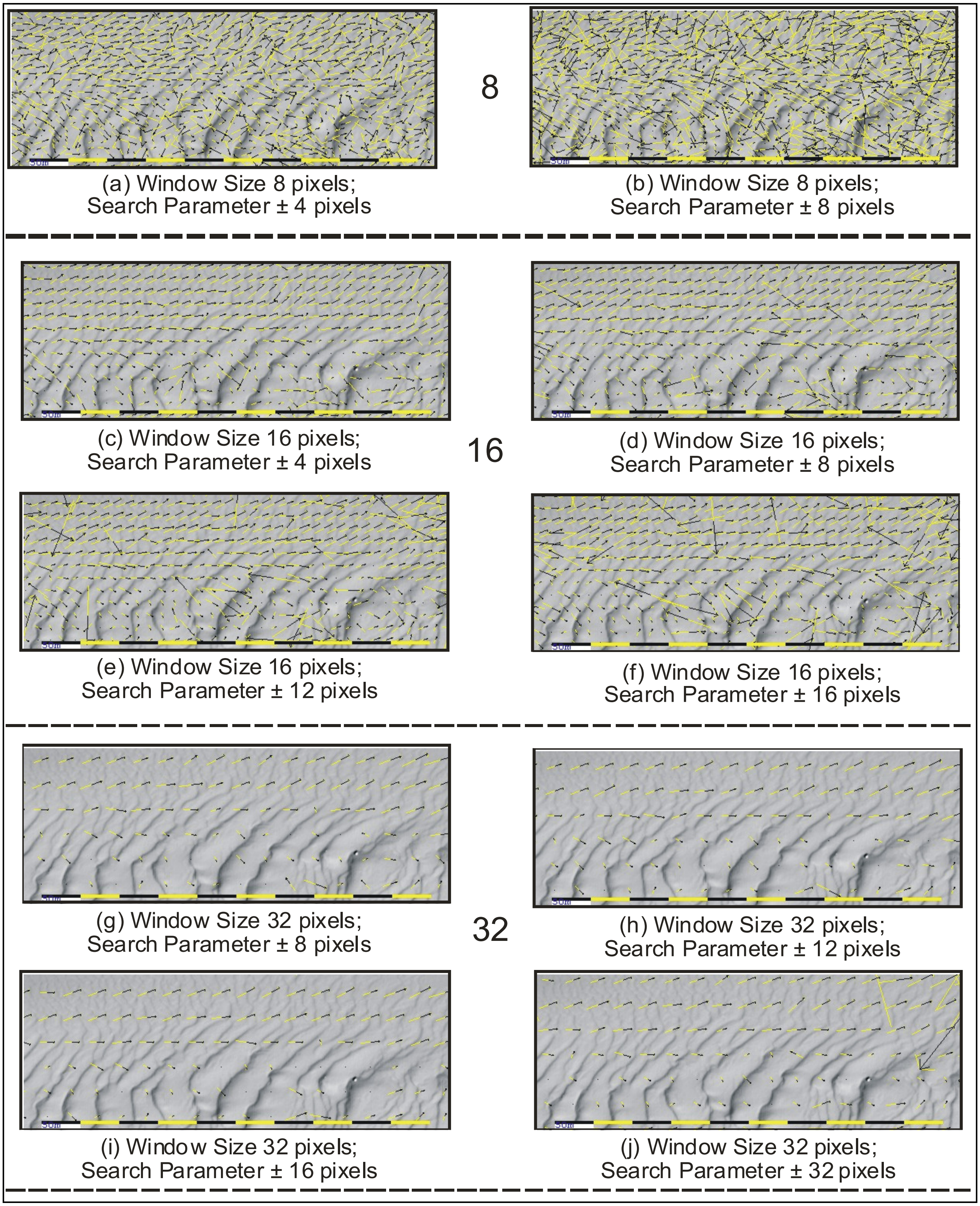


Figure 3 Some views of the sun illuminated DTM at different window sizes.

The effect of different window sizes and search parameters is shown in Figure 4. Figure 4(a and b) show the bedform migration field with a window size of 8 pixels. Here the field is densely populated because of the small window size (even though the displayed vector field has omitted every second vector for clarity) but there are some outliers because the terrain captured by the small window is not uniquely characteristic of the bedform (see an example in Figure 3(b)). However if some filtering was applied to omit the outliers the migration field in Figure 4(a) may still be useable and interpretable since with the eye of faith a systematic pattern may be perceived. In Figure 4(b), the search parameter is the same dimension as the window itself causing increased noise due to 'bad locks' or aliasing as the algorithm searches farther for an ambiguous pattern to start with. This situation is perhaps too noisy to be useful.

Figure 4(c,d,e and f) show a less dense and undecimated vector field due to a larger window size of 16 pixels. The vector field is also initially cleaner and show well the pattern of bedform migration, this is because the window captures a more characteristic pattern of pixels. However as we increase the search parameter again, especially as the search parameter exceeds half the window size, we see the vector field gets noisier due to bad locks.

Figure 4(g,h,i and j) show a sparsely populated field due to 32 pixel window size. Notable is the invariance of the vector field with increasing search parameter indicating possibly that an optimal match for the window is found within 8 pixels and indicating that the 32 pixel window is sufficiently unique to get a good lock and so is independent of search size.



(a) Window Size 8 pixels; Search Parameter ± 4 pixels

(b) Window Size 8 pixels; Search Parameter ± 8 pixels

(c) Window Size 16 pixels; Search Parameter ± 4 pixels

(d) Window Size 16 pixels; Search Parameter ± 8 pixels

(e) Window Size 16 pixels; Search Parameter ± 12 pixels

(f) Window Size 16 pixels; Search Parameter ± 16 pixels

(g) Window Size 32 pixels; Search Parameter ± 8 pixels

(h) Window Size 32 pixels; Search Parameter ± 12 pixels

(i) Window Size 32 pixels; Search Parameter ± 16 pixels

(j) Window Size 32 pixels; Search Parameter ± 32 pixels

Figure 4 Sample cross-correlation code with different window sizes and search parameters.

5 Conclusion

Whilst movies are a useful method with which to visualise changes in bathymetry, they are difficult to interpret and can be confusing, especially over large areas. This paper has shown that spatial cross-correlation is an effective tool for interrogating time lapsed bathymetric datasets when the correlation parameters are carefully selected. These parameters are of course fundamentally controlled by the dimension and the migration rate of the bedforms: the window size must be chosen to correspond to a characteristic length scale of the bedform and the search parameter must correspond to the average displacement of the bedforms over the area of investigation.

Finally, a powerful corollary of this research is that if we have multiple vector fields, e.g. from time t_1 to t_2, t_2 to t_3, t_3 to t_4, etc. from repeat bathymetric surveys we can potentially improve the resolution of the migration field by temporally averaging the vector fields, assuming the same migration conditions exist over that period.

6 References

Bendat, J.S. and Piersol, A.G. (1971). *Random Data: Analysis and Measurement Procedures.* Wiley.

Jambunathan, K., Ju, X.Y., Dobbins, B.N. and Ashforth-Frost, S. (1995). *An improved cross correlation technique for particle image velocimetry.* Meas. Sci. Technol. 6, 507-514.

Lewis, J.P.. *Fast Normalized Cross Correlation.* Industrial Light and Magic. http://www.idiom.com/~zilla/Work/nvisionInterface/nip.pdf

Lipkin, B.S. and Rosenfeld, A. (1970). *Picture Processing and Psychopictorics.* Academic Press.

McKenna, S.P., McGillis, W.R. (2002). *Performance of digital image velocimetry processing techniques.* Experiments in Fluids **32** 106-115

Pratt, W.K. (1991). Digital Image Processing. Wiley.