ABSTRACT

Backscatter ionograms are often used in HF communication and over-the-horizon radar systems as a means of assessing the available propagation. In such situations, the main feature of interest is the leading edge. An algorithm is outlined for the real time extraction of leading edges from backscatter ionograms via the two-dimensional discrete wavelet transform.

INTRODUCTION

Backscatter sounding is a powerful technique for deriving the propagation characteristics of the ionosphere for a large class of range and frequency [1-3]. Essentially, power is transmitted for sample frequencies over a wide range of elevations. The power is then scattered back to the sounder by the ground where it lands. This produces a 2D plot of group range against frequency for the scattered returns. This is the backscatter ionogram, which can be used to infer useful information about propagation and the ionosphere that produced it. In particular, the leading edge of the ionogram provides information about the plasma density at the range for which the ionospheric “reflection” takes place [2]. Thus, backscatter ionograms are an important frequency management tool for radio communication and HF over the horizon sky wave radars [3]. Traditionally, only the first leading edge is used, but the second leading edge can provide important information about changes in plasma as the range increases. As a consequence, the current paper seeks to develop an algorithm for deriving both first and second leading edges. Such an algorithm can then provide an important tool for long-range ionospheric evaluation.

APPLYING THE DISCRETE WAVELET TRANSFORM

The aim is to fit a curve to the first and second leading edges. Note that the first leading edge corresponds to 1-hop F-layer reflections and the second leading edge to 2-hop F-layer. A key feature of ionograms is the so-called “nose” position of each leading edge. The nose position corresponds to the highest frequency at which there are large power returns for a leading edge, and the range at which this frequency occurs. A very simple heuristic method for locating the nose is to classify the nose frequency as being the highest frequency bin that contains power greater than a certain threshold value, and the nose range as the range bin with the highest power in that frequency bin. Providing only a certain range of frequencies and ranges are searched, this routine will return the nose location for the desired leading edge.

All figures and data in this paper are from the Jindalee over-the-horizon radar at Alice Springs, Australia. Fig.1 shows a typical backscatter ionogram and the positions of the first and second leading edge nose positions as found by this heuristic. The chance of any problems caused by the background clutter is removed by ignoring all pixels with a power less than a certain minimum threshold and at frequencies higher than the first leading edge’s nose. The first step in the fitting of curves to the leading edges is to automatically extract the pixels that lie on each leading edge. One way of doing this is to use the Two-Dimensional Discrete Wavelet Transform (2-D DWT) [4]. Note from Fig.1 that there is a large rapid increase in power with increasing range and decreasing frequency at each leading edge. The 2-D DWT is good at detecting such edges in images, that is, such large rapid changes in image intensity, and is therefore ideal for this problem. Furthermore, the algorithm we use is a fast 2D-DWT, which is a faster algorithm than the 2D-FFT.

Fig.2 shows the result of applying the 2-D DWT to Fig.1, where Fig.2(a) shows vertical edge information and Fig.2(b) horizontal edge information. Note how most of the line of blue pixels (large negative returns, i.e. increases in power with increasing range) in Fig.2(a) corresponds to the position of the first leading edge, and similarly the red pixels (large positive returns, i.e. decreases in power with increasing frequency) in Fig.2(b). Since the leading edge corresponds to the case where a region of low power returns changes rapidly to a region of high power returns with decreasing

This work was funded by the DSTO and the author would like to thank them for their support and data.
frequency and with increasing range, Fig. 2 indicates a method of obtaining samples for the curve fitting; simply
threshold the vertical DWT returns to leave only the blue pixels, and note their locations, or threshold the horizontal
DWT returns to leave only the red pixels. In actual fact, experiments have shown that it is worthwhile to combine the
horizontal and vertical DWT returns to reinforce the location of the leading edges. A simple way to do this is to
subtract the powers of the horizontal DWT returns from the corresponding pixels in the vertical, so that the resulting
image has its most positive pixels on the leading edge.

Fig. 1 Original backscatter ionogram, from Jindalee OTHR at Alice Springs, year 2000, day 240 at 7:39:46 UT. The
black crosses show the nose position of the first and second leading edges as detected by the algorithm.

Images returned by the DWT are downsampled by a factor of two. However, what is needed is the locations of the
pixels in the original image (Fig. 1) corresponding to the strongest returns from the combined vertical and horizontal
DWT. To obtain these it is necessary to choose a threshold power, and keep those pixels that have a DWT power return
above that threshold and then map these back to the corresponding pixels in the original image. One method is to
simply take the desired pixels from the DWT image, multiply their indices by two, and then keep this pixel from the
original image, as well as all eight surrounding pixels from this location. Note that the original power of these pixels is
no longer relevant as the aim is to fit to the pixels on the leading edge where there was a large gradient and hence only
the location of these pixels is necessary.

Fig. 2. Relevant pixels after applying the DWT to Fig. 1; (a) DWT vertical returns and (b) DWT horizontal returns.

FITTING A QUADRATIC TO THE LEADING EDGES

Due to the slightly curved nature of the leading edges, it is sufficient to fit a quadratic to the leading edges, using a
weighted least squares data fitting criteria. Data points are those obtained using the procedure described in the previous
section. Note from Fig. 1 that in this typical example the first and second leading edges are well separated. It is
assumed that this will be the case in general, and therefore it is possible to fit two curves to the two separated leading
edges. This means that it is necessary to have a reliable means in the software of ensuring separation of the two leading edges before a curve is fitted to the pixels, or a means for fitting the curve that will reliably throw away points from the wrong leading edge.

Note that the first leading edge has a large number of pixels associated with it, and that its nose range is lower than many of the points on the second leading edge. To remove some of the second leading edge, the number of possible pixels is restricted by discarding those with frequency indices higher than the nose frequency, and those with range indices above the nose range. The fact that some of the remaining data is not on the first leading edge is dealt with by performing several passes and eliminating points well away from the fitted curve. It is assumed that the location of the nose is correct, and should definitely be a point on the fitted curve. Therefore the weighting of the nose location is set to be twenty times larger than any other point.

The reason a weighted least squares quadratic fit is used is that one of the major problems encountered with a basic least squares fit is that when there are not many data points returned, the fitted curve can easily be lured away by pixels that are well away from where the fit should be. This occurs fairly frequently with the second leading edge. To try and counter this problem, note that both leading edges lie very roughly along a straight line from the origin to the nose location. Hence, data points are weighted according to how far away they are from this straight line, using a Gaussian weighting, so that points a long way from this line are virtually discarded, points close to this line are heavily weighted in comparison, and points on the line have a weight of one.

Assume \( N \) data points (i.e. pixels obtained using the procedure outlined above) are returned. Denote the location of the \( i \)-th data point as \((x_i, y_i)\). The aim is to fit a quadratic of the form \( y_i = ax_i^2 + bx_i + c \), and therefore there are three unknowns. Define the following matrices

\[
X = \begin{bmatrix} x_1 & x_1 & 1 \\
x_2 & x_2 & 1 \\
\vdots & \vdots & \vdots \\
x_n & x_n & 1 \end{bmatrix}, \quad Y = \begin{bmatrix} y_1 \\
y_2 \\
\vdots \\
y_n \end{bmatrix}, \quad P = [a \ b \ c].
\]

Then the least squares quadratic fit to the data is given by \( P = Y^T X (X^T X)^{-1} \). If the \( i \)-th data point has a weighting of \( A_i \), then define

\[
A = \begin{bmatrix} A_1 & 0 & \cdots & 0 \\
0 & A_2 & \cdots & 0 \\
\vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & A_n \end{bmatrix}
\]

and the weighted least squares quadratic fit is given by \( P_w = Y^T A X (X^T A X)^{-1} \).

Denote the nose position as \((x_{\text{nose}}, y_{\text{nose}})\). As stated, it is desired that the weights decrease in a Gaussian fashion as data points get further away from the line \( y = Bx \) where \( B = y_{\text{nose}} / x_{\text{nose}} \). Hence define the weights as

\[
A_i = \exp \left( -\frac{(y_i - Bx_i)^2}{2\sigma^2} \right)
\]

where \( \sigma \) is the standard deviation for the weight, and becomes a control parameter in the algorithm. Note that for data points on the line \( y = Bx \) that \( A_i = 1 \), and data points a long way from that line have \( A_i \) approach zero. Experiments indicate that a value of \( \sigma = 6 \) is good for the first leading edge and \( \sigma = 3 \) is a good value for the second leading edge. Trials have shown that with a single attempt at this fitting, the resultant curve for the first leading edge is lured away by points on the second leading edge. This is allowed for by a second pass at the fitting. A crude method for doing this is to discard data points that are more than some fixed arbitrary \( y \)-distance away from the fitted curve and finding a new fit with these points. This process is then repeated. It is clear from trials that this third pass gives a very good fit. The same procedure is used for fitting the second leading edge, except that prior to fitting, the first leading edge is removed.
from the data. This is done by removing all pixels from range bin 1 to the range bin 10 pixels above the fitted first leading edge.

CONCLUSIONS

Experiments have shown that the procedure outlined above works very well throughout a whole day’s worth of data for the first leading edge for the Jindalee OTHR at Alice Springs. Several aspects of the algorithm combine to cope with ionograms where both leading edges are very close together; firstly, the 2-D DWT ensures that only pixels close to the leading edges are obtained, secondly, we use a weighting least squares fitting, and thirdly, several passes of the fitting are performed. However, the second leading edge almost disappears at certain times, and consequently this fitting procedure sometimes breaks down for the second leading edge, but works very reliably otherwise.

Fig. 3 shows the original data from Fig. 1 with lines plotted showing the quadratic fits to both leading edges, plus the pixels used for the fitting. Clearly for this sample, the fitting has been quite good for both leading edges. It is also clear from trials that the returns for the first leading edge are far more distinct than the second leading edge. Hence greater reliability is expected in fitting to the first leading edge regardless of the time of day, and this has been found to be the case. Algorithms such as this one will always run into problems if they receive unexpected data, and hence the algorithm has been designed to cope with typical backscatter ionograms, and provide parameters that can be modified should typical conditions change, or for a specific radar. Finally, we note that the average computational time for our algorithm is negligible compared to the typical time between obtaining ionogram samples, and therefore this algorithm could therefore easily be used in realtime.

ACKNOWLEDGEMENTS

The author wishes to thank Assoc. Prof. Chris Coleman, of The School of Electrical and Electronic Engineering at The University of Adelaide for valuable comments and discussion.

REFERENCES