IMAGE COMPRESSION BY MEANS OF WAVELET TRANSFORM – APPLICATION TO SOLAR OBSERVATIONS

WERNER MÜHLMANN and ARNOLD HANSLMEIER
Institut für Astronomie, Universitätsplatz 5, A-8010 Graz, Austria

(Received 27 October, 1995; in final form 26 January, 1996)

Abstract. Images and spectra contain a large amount of data. Therefore the question arises, how this data can be decreased or compressed without losing important detail. The discrete wavelet transform is a tool which can be used to compress data because of its good approximation properties. It is very easy to implement and requires approximately the same amount of calculation as the fast Fourier transform. It has the advantage of giving information both in time and frequency. Since most of the coefficients in the transformed data are very small compared to the maximum values, which means that they do not contribute much to the transform, a large number of them can be neglected. Although some data get lost, the physical results deduced from the data remain the same, as is demonstrated by various examples. By this method it is even possible to compress data containing much noise to high-compression ratios.

1. Introduction

The wavelet transform has become a versatile tool in recent years. Many applications are responsible for its success. It has been used for image restoration and noise suppression (Starck and Murtagh, 1994), for determining solar diameter variabilities (Vigouroux and Delache, 1993), analysis of cosmic velocity fields (Rauzy and Lachieze-Rey, 1993), photometric analysis of astronomical images (Coupinot et al., 1992) and for sound, images and fractal structures (Arnéodo et al., 1988). The mathematical base was provided by Grossmann, Kronland-Martinet, and Morlet (1989) and Daubechies (1992).

In this article we want to put stress on the use of the wavelet transform for data compression and the application to solar images and spectra.

High-resolution images of solar granulation contain many structures and a large amount of noise. Compressing them with common lossless algorithms such as gzip or compress leads to a compression with data reduction of only 15 to 20%. The noise in these images is responsible for the bad compression rate, since it is incompressible. Therefore, it is necessary to make use of a lossy algorithm if high compression ratios are to be achieved.

Another very interesting problem is the compression of spectra. The spectra contain many small structures, such as weak lines, which should not disappear in such a compression method.

In this Letter the Daubechies wavelets are used, which are very easy to implement. They are also integrated into IDL and are in the Numerical Recipes packages (Press et al., 1992).
2. The Discrete Wavelet Transform

Since the wavelet transform is not so familiar, we want to give a short introduction defining the wavelet transform. The wavelet transform is given by the following equation:

\[
(T^{\text{wav}} f)(a, b) = |a|^{-1/2} \int f(t) \psi \left( \frac{t - b}{a} \right) \, dt.
\]  

(1)

The values \(a\) and \(b\), the translation and dilation parameters are discretized. The wavelets constitute an orthonormal basis for the Hilbert space. The transformation matrix is:

\[
T = \begin{bmatrix}
S \\
D
\end{bmatrix} = \begin{pmatrix}
h_0 & h_1 & \cdots & h_{2N-1} \\
h_0 & h_1 & \cdots & h_{2N-1} \\
& \ddots & \ddots & \ddots \\
g_0 & g_1 & \cdots & g_{2N-1} \\
g_0 & g_1 & \cdots & g_{2N-1} \\
& \ddots & \ddots & \ddots \\
g_0 & g_1 & \cdots & g_{2N-1}
\end{pmatrix},
\]  

(2)

\(S\) stands for smooth and \(D\) for detail, one part smoothes the data and the other points out differences. The \(h\) and \(g\) are the filter coefficients; only a small number are different from zero; they can be found in Daubechies (1992) and Mühlmann (1995). The dimension of the matrix is \(n \times n\), the length of the data vector is \(n\), and \(n\) is a power of 2. If \(n\) is not a power of two, a zero padding has to be performed. To reduce boundary effects the matrix is made cyclic.

Since the transformation matrix is orthogonal, the inverse transform is just done with the transposed matrix. The generalization to more than one dimension is also very easy. In the 2-dimensional case first all row vectors and then all column vectors are transformed. The order of the calculation is approximately \(n\); it depends upon the number of filter coefficients, whether it is more or less.

3. Application to Data Compression

In Figure 1(b) a logarithmic scale is necessary to make most of the coefficients visible. The first coefficients, the smooth ones, are much larger than the detail coefficients.

Now all values which are lower than a certain threshold can be truncated. In this form the data can be compressed by the other data compressors without any problem. If 80% of the transformed data is set to zero and compressed with an
ordinary compression code, such as Ziv and Lempel (1978) or Huffman (1952), a compression-ratio of 80%±1% can be achieved.

4. Results

For our calculations we used images of solar granulation and infrared spectra of the Sun. The images contain a smooth pattern of granulation and noise, which can be described as detail signal. The spectrum on the other hand is relatively smooth, but it contains also details that are very different in their size (thick and very thin lines).

In Figure 2(a) the r.m.s. between compressed and original is given; it is normalized to 1. In Figures 2(b) and 2(d) the ratio of original value minus compressed value to the original value \((o - c)/o\) is plotted against the percentage of data reduction. Figure 2(c) shows the displacement of a line minimum in comparison to the original position. It can clearly be seen that the errors up to 80% data reduction are almost negligible. Also better results can be achieved when more filter coefficients are used, but the advantage is not very good.

4.1. Further Applications

Another interesting application would be the transform and compression of time series. Time series with images of solar granulation, for instance, show small changes from one image to the next. A three-dimensional wavelet transform will approximate such a series very well. The only problem is the large amount of memory, which is needed for the transform. The compression ratios should increase due to the third dimension. This dimension is the most continuous one, the detail coefficients are therefore very small.
Werner Mühlmann and Arnold Hanslmeier

Figure 2. (a) R.m.s. error between original and compressed granulation image with 16 and 4 filter coefficients. (b) Variation of the equivalent width of a line, 8 and 16 coefficients (c) Displacement of a line minimum, 16 coefficients. (d) Variation of $\delta I_{r.m.s.}$ of a column of a granulation image with 16 and 8 filter coefficients.

References


