SCALING AND FILTERING OF A SAMPLED SIGNAL BY THE CONTINUOUS WAVELET TRANSFORM

Y. T. Chan, K. C. Ho, P. C. Ching

1 Royal Military College of Canada, Kingston, ON, Canada. K7K 5L0
2 University of Missouri-Columbia, MO 65211, USA
3 The Chinese University of Hong Kong, Shatin, N.T. Hong Kong

ABSTRACT

Scaling of a discrete time sequence is necessary in some applications. An example is in estimating the time delay and Doppler stretch between two waveforms received at spatially separated sensors, when there are relative motions between the signal source and the sensors. The scaling task is non-trivial because the signal to be scaled does not have an analytical form. This paper considers the use of continuous wavelet transform (CWT) to perform scaling of a discrete time signal. The method involves wavelet transforming the discrete sequence, thresholding coefficients for noise reduction and forming the scaled samples using the CWT reconstruction formula. Simulations are presented to evaluate the performance of the method.

1. INTRODUCTION

Some digital signal processing tasks require the scaling of a discrete time signal. One example is the estimation of time delay between two receiver outputs when the source and receivers have relative motion [1]. In such a case one needs to compress or expand the discrete time samples from one receiver before it can be correlated with the other sensor output to estimate the delay.

The problem is stated as follows. Given a noisy data sequence

\[ x(k) = s(k) + n(k) \]  

we are interested in finding \( s(ak) \), the scaled data samples of \( s(k) \), where \( a \) is the scaling factor which may be very close to unity.

Ho et al. [2] has recently proposed a method to solve the scaling problem. The technique is optimum in the sense that the mean-square error between the estimated signal and the true signal is minimized. The method first applies Wiener filtering on the input data to reduce noise. A sinc function interpolator then performs scaling on the filtered data. The design of a Wiener filter requires known signal and noise statistics but this knowledge may not be available in many practical situations.

This paper considers the use of continuous wavelet transform (CWT) to scale a discrete time signal. We first transform the input data to wavelet coefficients. The scaled data samples are then generated using the CWT signal reconstruction formula. Noise filtering is achieved by thresholding the CWT coefficients. The advantage of the proposed technique is that no prior knowledge about the signal and noise statistics is necessary.

The paper is organized as follows. The next section gives a brief review to CWT and shows how to compute the CWT of a discrete time signal. Based on the signal reconstruction expression, a formula that determines the scaled signal samples from CWT coefficients is developed. Section 3 presents the method to perform discrete time signal filtering and scaling. Section 4 provides simulation results to illustrate the performance of the method and conclusions will be drawn in Section 5.

2. CWT OF A DISCRETE TIME SIGNAL AND ITS CONTINUOUS TIME RECONSTRUCTION

The CWT is originally developed to operate on a continuous time signal. This section derives the computation of CWT for a discrete time signal. The technique is to generate the continuous time equivalent of a discrete signal using sinc function interpolation. The CWT values are continuous in translation and sampling is necessary to reduce the CWT data size. The reconstruction of a continuous time waveform from the CWT samples will be given.

The CWT of a signal \( x(t) \) is defined as [3-4]

\[ CWT_x(\alpha, \tau) = \frac{1}{\sqrt{\alpha}} \int_{-\infty}^{\infty} x(t) \psi^* \left( \frac{t-\tau}{\alpha} \right) dt \]  

where \( \alpha \) and \( \tau \) are the scale and translation parameters and \( \psi(t) \) is the mother wavelet that must satisfy the admissibility condition. The superscript * denotes complex conjugate. In practice, only a certain number, say \( N \), of data samples \( x(k) \),
Upon using the linearity property, the CWT of the noisy signal in (1) is

$$CWT_x(a, r) = CWT_s(a, r) + CWT_n(a, r).$$

We shall assume the noise is Gaussian with zero mean and variance \( \sigma_n^2 \). In such case it can be easily shown that \( CWT_n(a, r) \) is also Gaussian, with the mean equal to zero and variance \( \sigma_n^2 \).

If the input is noise only, we can find from the Gaussian distribution a threshold level, denoted by \( TH \), for a given probability of miss of a noisy data sample.

The proposed scaling method uses simple thresholding to reduce noise. That is, for all possible \( u \) and \( v \),

$$\chi(u, v) = \begin{cases} 0, & \text{if } |CWT_x(u, v)| < TH \\ CWT_x(u, v), & \text{otherwise}. \end{cases}$$

Putting (12) into (10) yields the noise reduced signal estimate:

$$\hat{s}(t) = \frac{1}{c_v} \sum_{u \neq 0, v} CWT_x(u, v) \tilde{\psi}(u, t - v).$$

When \( t \) is substituted by \( ak \), we have the scaled data sequence:

$$\hat{s}(ak) = \frac{1}{c_v} \sum_{u \neq 0, v} CWT_x(u, v) \tilde{\psi}(u, ak - v).$$

To summarize, the steps to compute the scaled discrete time samples are as follows:

1. apply eq. (4) to compute \( CWT_x(u, v) \),
2. use eq. (12) to reduce noise,
3. reconstruct the scaled samples from eq. (14).

Let the total number of scale levels used in (14) be \( U \) and that of the translations be \( V \). Step 1 requires a computational complexity of \( O(UV \log_2 V) \) when using FFT to compute convolution. Step 2 can be accomplished using a simple thresholding circuit. The computation of \( N \) scaled samples in step 3 requires \( N \) times \( O(UV) \) operations. Simulation study shows that choosing \( V \) to be slightly larger than \( N \), say \( 1.5N \) is enough. Thus the total complexity is \( O(UN^2) \) in order to compute \( N \) scaled samples from \( N \) input data samples.

The performance of the proposed method is dependent on the mother wavelet chosen. Ideally one should choose a wavelet close to the signal pattern. In the noise filtering perspective, the noise thresholding scheme works best if one of the baby wavelets matches exactly to the signal. Considering computational requirement, if the signal is close to the wavelet there will only be a small number of non-zero

3. THE PROPOSED SIGNAL SCALING METHOD

Upon using the linearity property, the CWT of the noisy signal in (1) is

$$CWT_x(\alpha, \tau) = CWT_s(\alpha, \tau) + CWT_n(\alpha, \tau).$$

We shall assume the noise is Gaussian with zero mean and variance \( \sigma_n^2 \). In such case it can be easily shown that \( CWT_n(\alpha, \tau) \) is also Gaussian, with the mean equal to zero and variance \( \sigma_n^2 \). If the input is noise only, we can find from the Gaussian distribution a threshold level, denoted by \( TH \), for a given probability of miss of a noisy data sample.

The proposed scaling method uses simple thresholding to reduce noise. That is, for all possible \( u \) and \( v \),

$$\chi(u, v) = \begin{cases} 0, & \text{if } |CWT_x(u, v)| < TH \\ CWT_x(u, v), & \text{otherwise}. \end{cases}$$

Putting (12) into (10) yields the noise reduced signal estimate:

$$\hat{s}(t) = \frac{1}{c_v} \sum_{u \neq 0, v} CWT_x(u, v) \tilde{\psi}(u, t - v).$$

When \( t \) is substituted by \( ak \), we have the scaled data sequence:

$$\hat{s}(ak) = \frac{1}{c_v} \sum_{u \neq 0, v} CWT_x(u, v) \tilde{\psi}(u, ak - v).$$

To summarize, the steps to compute the scaled discrete time samples are as follows:

1. apply eq. (4) to compute \( CWT_x(u, v) \),
2. use eq. (12) to reduce noise,
3. reconstruct the scaled samples from eq. (14).

Let the total number of scale levels used in (14) be \( U \) and that of the translations be \( V \). Step 1 requires a computational complexity of \( O(UV \log_2 V) \) when using FFT to compute convolution. Step 2 can be accomplished using a simple thresholding circuit. The computation of \( N \) scaled samples in step 3 requires \( N \) times \( O(UV) \) operations. Simulation study shows that choosing \( V \) to be slightly larger than \( N \), say \( 1.5N \) is enough. Thus the total complexity is \( O(UN^2) \) in order to compute \( N \) scaled samples from \( N \) input data samples.
\( \text{CWT} \) coefficients. As a result, \( U \) is small and the complexity can be significantly reduced.

The proposed method may not work well to scale a signal that has a rich low frequency content. This is due to the bandpass nature of a wavelet in order to satisfy the admissibility condition. The scales must be very large to generate low frequency components. Hence \( U \) ought to be large as well and this is prohibited by the extensive computational requirement. Fortunately, most signals in rader, sonar etc. are bandpass and the bandpass signal requirement should not present any restriction of the proposed method in most applications.

4. SIMULATIONS

We have studied two wavelet functions for use in the scaling algorithm proposed. One is the real component of a Morlet wavelet:

\[
\psi(t) = e^{-t^2/2\sigma^2} \cos(2\pi f_o t), \quad \sigma = 0.5, \ f_o = 1 \quad (15)
\]

and the other is the Mexican Hat wavelet:

\[
\psi(t) = (1 - t^2 / 2.25)e^{-t^2/(2.25 \sigma^2)}, \quad \sigma = 0.25 \quad (16)
\]

In both cases the \( \sigma \) were set so that the wavelets cover the high frequency region around \( f = 0.5 \). Note that the Mexican Hat wavelet starts at a scale of 1.5.

The signal was either a Morlet signal

\[
s(t) = e^{-t^2/2\sigma^2} \cos(2\pi f_o t), \quad \sigma = 3.0, \ f_o = 0.5, -16 \leq t \leq 16 \quad (17)
\]

or a Mexican Hat signal

\[
s(t) = (1 - t^2 / 2.25)e^{-t^2/(2.25 \sigma^2)}, \quad \sigma = 1, -8 \leq t \leq 8. \quad (18)
\]

We chose the signals to have analytical forms so that exact signal scaling is available when evaluating the performance of the proposed algorithm. The parameters are set such that the signals were not equal to one of the baby wavelets. The scale factor \( a \) was 1.1 throughout the study.

Figure 1 shows the dependency of the square error on the number of scale \( U \) used in the reconstruction formula eq. (14), where the square-error is defined as

\[
SE = \frac{\Sigma (\hat{s}(ak) - s(ak))^2}{\Sigma s(ak)^2} \quad (19)
\]

5. SUMMARY

Scaling of a discrete time sequence is a challenging problem because in most cases a discrete time sequence does not have an analytical form. The problem is further complicated when the scale factor is close to unity. This paper proposed the use of an intermediate wavelet that has an analytical form to achieve scaling of a discrete time sequence. It consists of wavelet transforming the input data, thresholding CWT coefficients for noise reduction, and reconstructing the scaled signal samples from the CWT samples. The relevant formulae for the proposed method were derived. Simulations have verified the validity and showed the performance of the method.

6. REFERENCES

Fig. 1 Square-error versus $U$ for Morlet wavelet

Fig. 2 Square-error versus $U$ for Mexican Hat wavelet

Fig. 3 Performance of the proposed method with Morlet wavelet. $U=40$, $V=48$

Fig. 4 Performance of the proposed method with Mexican Hat wavelet. $U=60$, $V=24$

Fig. 5a Signal before scaling and noise filtering

Fig. 5b Signal after scaling and noise filtering