Analysis of Blind Signal Separation of Mixed Signals Based on Fast Discrete Curvelet Transform

1S.A.Shehata, 2S. M. Diab, 3B. M. Salam, 4El-Sayed M. El-Rabaie, 5F. E. Abd El-Samie, 6S.S.Haggag, 7M.Y.Abbass
1Department of Engineering, Nuclear Research Center, Atomic Energy Authority, Egypt
2Department of Electronics and Electrical Communications, Faculty of Electronic Engineering, Menoufia University
Menouf, 32952, Egypt.

Abstract—This paper investigates the technique of Fast Discrete Curvelet Transform (FDCT) de-noising with the Independent Component Analysis (ICA) for the separation of noisy signals. Two approaches are presented for this purpose. In the first approach, noisy mixed signals are separated using fast ICA algorithm and then Curvelet thresholding is used to de-noise the results. The second approach uses Curvelet thresholding to de-noise the mixed signals and then the fast ICA algorithm to separate the de-noised signals. The simulation results show a better performance for image de-noising followed by separation. The Signal-to-Noise Ratio (SNR) and Root Mean Square Error (RMSE) are used as metrics to evaluate the quality of the separated signals.

Index Terms - Blind Source Separation (BSS), ICA, FDCT.

I. INTRODUCTION

Digital signals are invariably contaminated by noise. Noise arises due to imperfect instruments used in signal processing, problems with the data acquisition process, and interference which can degrade the data of interest. Also, noise can be introduced due to compression and transmission errors [1]. The overall noise characteristics in a signal depend on factors like type of sensor, exposure time, pixel dimensions, ISO speed, and temperature [2]. As most of the natural signals are assumed to have additive random noise, which is modeled as Gaussian noise, de-noising is the first step to be considered before the signal data is analyzed.

Blind source separation (BSS) is the method of extracting underlying source signals from a set of observed signal mixtures with little or no information to the nature of these source signals. Independent component analysis (ICA) is used for finding factors or components from multivariate statistical data and is one of the many solutions to the BSS problem [3]-[4]. ICA looks for the components that are both statistically independent and non-Gaussian. The various ICA algorithms extract source signals based on the principle of information maximization, mutual information minimization, maximum likelihood estimation and maximizing non-Gaussianity. ICA is widely used in statistical signal processing, medical image processing, economic analysis and telecommunication applications [5].

Most of the ICA methods are developed assuming noiseless data and these algorithms perform poorly in the presence of noise [6]-[7]-[8].

In this paper, noisy multiple channel blind signal separation algorithms based on FDCT thresholding are investigated. In the first approach, noisy mixed signals are separated using the fast ICA algorithm, and then hard FDCT thresholding is used for de-noising. The second approach uses hard FDCT thresholding to de-noise signals and then the fast ICA algorithm is used to separate the de-noised signals.

The organization of this paper is as follows. Section II, includes details about the independent component analysis. In Section III, fast ICA algorithm is explained. In Section IV, the curvelet transform are reviewed. Simulation results are presented in Section V followed by the conclusion and the more relevant references.

II. INDEPENDENT COMPONENT ANALYSIS(ICA)

Many popular ICA methods use a nonlinear contrast function to blindly separate the signals. Examples include equivariant adaptive source separation [9], fast ICA [10], and efficient fast ICA [11]. Adaptive choices of the contrast functions have also been proposed, in which the probability distributions are obtained by considering a Maximum Likelihood (ML) solution corresponding to some given distributions of the sources and relaxing this assumption afterwards [12]-[13]. This method is specially adapted to temporally independent non-Gaussian sources and is based on the use of nonlinear separating functions. Furthermore, Tichavsky et al. [14] have proposed two general purpose rational nonlinearities that have similar performance as tanh, but can be evaluated faster.
The basic ICA model, which is shown in Fig. 1, can be stated as,

\[ x(t) = As(t) + n(t) \]  

(1)

where \(x(t)\) is an \(N\) dimensional vector of the observed signals at the discrete time instant \(t\), \(A\) is an unknown mixing matrix, \(s(t)\) is the original source signal of \(M \times N\) \((M \leq N)\) and \(n(t)\) is the observed noise vector and \(M\) is number of sources. The purpose of the ICA is to estimate \(s(t)\), which is the original source signal from \(x(t)\), which is the mixed signal, i.e., it is equivalent to estimating the matrix \(A\). Assuming that there is a matrix \(W\), which is the de-mixing matrix or separation inverse matrix of \(A\), then the original source signal is obtained by:

\[ s(t) = Wx(t) \]  

(2)

The ICA algorithm assumes [15] that the mixing matrix \(A\) must be of full column rank and all the independent components \(s(t)\), with the possible exception of one component, must be non-Gaussian. Further, the number of observed linear mixtures \(M\) must be at least as large as the number of independent components \(N\) \((M \geq N)\).

III. FAST ICA ALGORITHM

The fast ICA is the most popular algorithm used in various applications as it is simple, fast convergent, and computationally less complex. It is a fixed point iterative algorithm that uses a nonlinear function \(g(y) = \tanh(ay)\), which is applied to the separation vector \(W\) that is recalculated at each iteration of the algorithm. The fixed point algorithm is to iterate to obtain a global minimum. Once you determine the vector \(W\), it is pointing to one of the independent components.

This algorithm is more efficient than the gradient algorithm [16]. The input to the fast ICA algorithm must first be whitened by three steps: 1) centering over the average, 2) normalization of the variance and 3) orthogonalization of the data. The steps to implement the fast ICA algorithm are as follows [17]:

1) Center the data to make its mean zero.
2) Whiten the data to give \(z\).
3) Choose an initial (e.g., random) vector \(w\) of unit norm.
4) Let \(w^' = E\{zg(w^Tz)\} - E\{g'(w^Tz)\}w\).
5) Let \(w = w^' / ||w^'||\).
6) If not converged, go back to step 4.

Convergence means that the old and new values of \(w\) point in the same direction i.e., their dot product is equal to 1. To estimate the several independent components, we need to run one unit of the fast ICA algorithm using several units with weight vectors \(w_{1,...,p}\). When we have estimated \(p\) independent components, or \(p\) vectors \(w_{1,...,p}\) we run the one unit fixed point algorithm for \(w_{p+1}\) and after every iteration step subtract from \(w_{p+1}\) the projections \(w_{p+1}^Tw_j\), \(j = 1,...,p\) of the previously estimated \(p\) vectors, and then renormalize \(w_{p+1}\).

Let

\[ w_{p+1} = w_{p+1} - \sum_{j=1}^{p} w_{p+1}^Tw_jw_j \]

\[ w_{p+1} = w_{p+1} / \sqrt{w_{p+1}^Tw_{p+1}} \]  

(3)

IV. THE CURVELET TRANSFORM

The curvelet transform, which inherited the ridgelet transform, was introduced to represent edges better than all known image transforms. The two-dimensional ridgelet transform is calculated as follow: for any 1-D univariate function \(\psi\), which satisfies the admissibility condition with sufficient decay, a 2-D univariate ridgelet functions is calculated as in Eqn. (4).

\[ \psi_{a,b,\theta}(x_1 + x_2) = a^{-\frac{1}{2}}\psi(x_1 \cos \theta + x_2 \sin \theta - b) / a \]  

(4)

where, \(a\) is a scalar parameter, such that \(a > 0\), \(b\) is a location parameter, and \(\theta\) is a rotation parameter. If \(\Psi\) is the mother wavelet function, then the ridgelets are constant along the lines \(x_1 \cos \theta + x_2 \sin \theta\), and along the orthogonal direction they are represented by the wavelets only.
Therefore for any bivariate function \( f(x,y) \), the ridgelet coefficients are calculated as in Eqn. (5), and the inverse reconstruction is calculated as in Eqn. (6). Also, the Radon transform of an object \( f(x,y) \) is calculated as in Eqn. (7).

\[
R_f (\theta, t) = \int f(x_1, x_2) \delta(x_1 \cos \theta + x_2 \sin \theta - t) dx_1 dx_2
\]

(5)

\[
R_f (a, b, \theta) = \int \psi_{a,b,\theta}(x) f(x) dx
\]

(6)

\[
f(x) = \int_{-\infty}^{2\Pi} \int_{-\infty}^{\infty} R_f (a, b, \theta) \Psi_{a,b,\theta}(x) \frac{da}{a^3} db \frac{d\theta}{4\Pi}
\]

(7)

Therefore, the ridgelet transform is an application of the 1-D wavelet transform to that part of the Radon transform where \( \theta \) is constant and \( t \) is varying. In a conclusion: first, samples are extracted from the Fourier transform along lines through the origin in the frequency domain. Second, these samples are used to compute the inverse Fourier transform, and finally the 1-D wavelet transform is calculated for these samples.

The curvelet transform has gone through two major revisions. The first generation curvelet transform [18]-[19] used a complex series of steps involving the ridgelet analysis of the radon transform of an image. The performance was exceeding slow. The second generation curvelet transform [20] discarded the use of the ridgelet transform, thus reduced the amount of redundancy in the transform and increased the speed considerably. Two FDCT algorithms were introduced in [21]. The first algorithm is based on the unequally spaced FFT, while the second is based on the wrapping of specially selected Fourier samples. In this paper, we focus on the “wrapping” version of the curvelet transform.

Generally speaking, the curvelet transform is a special multi-scale pyramid with many directions and positions at each decomposition scale. Therefore, the curvelet transform is more suitable than all other multi-scale transforms including wavelet in some signal and image applications including, filtering, enhancement, compression, de-noising, and watermarking.

A. Curvelet Thresholding Approach

Curvelet de-noising is a simple operation, which aims at reducing noise in a noisy image. It is performed by selecting the FDCT coefficients below a certain threshold and setting them to zero as follows:

\[
y_d = \begin{cases} y_i & |y_i| \geq t_d \\ 0 & |y_i| < t_d \end{cases}
\]

(8)

Where \( t_d \) is the threshold and \( \lambda \) is the index.

The threshold used is \( t_d = k\sigma_d\sigma \), where \( \sigma \) is an estimation of the standard deviation of the noise, and \( \sigma_d \) is an approximation value for the standard deviation of each curvelet coefficient estimated by using the Monte-Carlo simulation [22]. \( k \) is taken as 4 for the finest scale and 3 for the other scales. In curvelet de-noising, the noise standard deviation is estimated from the finest scale coefficients corresponding to the diagonal orientation using the Median Absolute Deviation (MAD) estimate, which is given by:

\[
\sigma = \frac{MAD}{0.6745}
\]

(9)

Then, we use the inverse curvelet transform to get the de-noising image.

The various performance factors are then calculated using the following equations.

(i) SNR:

\[
SNR = 10 \log_{10} \left( \frac{\sum_{i=1}^{N} x^2(i)}{\sum_{i=1}^{N} (x(i) - y(i))^2} \right) dB
\]

(10)

Where \( x(i) \) is the original source signal, \( y(i) \) is the separated signal, \( i \) is the sample index and \( N \) is the number of samples of the signal.

(ii) RMSE:

\[
RMSE = \sqrt{\frac{1}{N} \sum_{i=1}^{N} (x(i) - y(i))^2}
\]

(11)
V. SIMULATION RESULTS

In this study, three signals were selected as shown in Fig. 2. These signals were mixed using a (3x3) random mixing matrix. White Gaussian noise was added to these mixed signals.

The noisy mixed signals so obtained are shown in Fig. 3.

Simulations were performed using Matlab® R 7.9 on a core i7 2.2 GHz PC. In the first approach, noisy mixed signals were separated by applying the fast ICA algorithm first and then de-noising. The separated independent components were thresholded using hard curvelet thresholding. Signals, which are separated by this approach, are shown in Fig. 4. The second approach uses hard curvelet thresholding to de-noise the signals, and then the fast ICA algorithm was used to separate the de-noised signals. The output signals of this approach are shown in Fig. 5. The output signals of the separated independent components without de-noising are shown in Fig. 6.

The performance evaluation metrics obtained using the hard curvelet thresholding approaches are showed in Figs. 7 to 12.
It is observed from Fig. 7 to 12 that the BSS using the second approach results in 10 to 15% improvement in the SNR, PSNR and RMSE.

Figure 7. Output SNR vs. input SNR for signal 1.

Figure 8. Output RMSE vs. input SNR for signal 1.

Figure 9. Output SNR vs. input SNR for signal 2.

Figure 10. Output RMSE vs. input SNR for signal 2.

Figure 11. Output SNR vs. input SNR for signal 3.
VI. CONCLUSION

In this paper, two approaches of noisy mixed signal separation have been studied to observe the effect of de-noising before and after signal separation. De-noising was done using hard curvelet thresholding and separation was based in using ICA. Results show that the performance metrics (SNR and RMSE) of the second approach (separation then de-noising) is better than the first approach (de-noising then separation).

REFERENCES

Shehata et. al., Analysis of Blind Signal Separation of Mixed Signals Based on Fast Discrete Curvelet Transform

