Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Mixing Matrix Identification for Underdetermined Blind Signal Separation: Using Hough Transform and Fuzzy K-means Clustering

Tsung-Ying, Sun[1], *IEEE Member*, Ling-Erh Lan[2], Chan-Cheng Liu[3], and Chih-Li Huo[4]
[1,2,4]Department of Electrical Engineering
National Dong Hwa University, Hualien,
[3]Institute of Information Science, Academia Sinica, Taipei,
Taiwan, R.O.C.
E-mail: [1]sunty@mail.ndhu.edu.tw

***Abstract*—This paper focuses on the underdetermined blind signal separation problem with sparse representation. The algorithm is proposed to identify the parameters of mixing model which are unknown. The distribution of mixtures are mapping to a new histogram domain by Hough transform which converts the Cartesian image space to the normal parameterization. And then, fuzzy k-means clustering is employed to seek the cluster centers, i.e. parameters of mixing model, on the histogram. Obtaining accurate estimates, the sources can be recovered clearly. The proposed algorithm and three existing algorithms are tested in the simulations. By the simulation results, our algorithm is able to perform a nice accuracy of estimation through a very low computational consumption.**

***Keywords*—Underdetermined Blind source separation, fuzzy k-means clustering, hough transform.**

I. Introduction

Blind source separation (BSS) involves recovering unobserved source signals from several mixed observations, which are typically obtained at the output of a set of sensors, where each sensor receives a different mixing matrix of the source signals. Some principle limitations of BSS is [1]:

- Mutual independent assumption on original signals.
- No information is available because of the unobservable original signals.
- Unobservable mixing environment. In general, a mixing model will be focused (known), but its parameters are assumed unknown.

BSS in signal processing has received considerable attention from researchers recently, due to its numerous promising applications in the areas of biomedical signal processing, digital communications and speech signal, sonar, image processing, and monitoring [1]-[6]. A number of blind separation algorithms have been proposed based on different separation models. These algorithms play increasingly important roles in many applications.

In reality, the number of source signals is different in variable environment. However, early BSS algorithms always require that the sensor number must equal to the source number (i.e. determined condition). Therefore, hardware could not suitably process the general cases, until Lee et al. The algorithm in [7] relied on the sparse representation to deal with the underdetermined BSS which has more source than sensor. As the mixtures are encoded with two sensors, each base vectors of unknown mixing matrix will display on the 2-D plane coordinate as a directional line. Some underdetermined algorithms based on neural network were proposed [8]-[12]. However, if there is not sufficient prior information about the mixing matrix or source, only a part of the source signals can be successfully extracted.

For the mixing matrix identification, Bofill and Zibulevsky proposed a potential function to transform an irregular mixture distribution to a simple curve involving some peaks (i.e. locations of mixing vectors) [13]. Lv and Zhang proposed a unified method for blind separation of sparse sources with unknown source number [14]. Shi et al. proposed the generalized exponential mixture model to approximate the distribution of the mixture, and this algorithm does not predefine any parameter [15]. The clustering methods as fuzzy clustering, *k*-means and some extension were proposed to search the mixing matrix [16]-[18]. These mentioned algorithms performed good performance in a well-condition mixing matrix, and assume source number is known.

Generally, in order to get a precise mixing matrix, it will take some time to achieve the goal. On the contrary, the error is bigger when the processing time is shorter relative to the former. This paper proposes a mixing matrix identification algorithm for underdetermined BSS. The proposed method associates Hough transform which is usually used on the image processing, and Fuzzy *k*-means clustering to obtain a set of precise mixing vectors and reduce computational time. According to the properties of our task, the mixing signals from two sensors, which are transferred from the time-domain to the frequency spectrum first via the Fourier Transform. Second, converting the data points on the *x*-*y* plane to the normal parameterization (ρ,θ) , then use the fuzzy *k*-means

978-1-4244-2794-9/09/$25.00 ©2009 IEEE

clustering to find the cluster centers. (Here we assumed that the number of sources is known.) After that, the algorithm searches a maximum in a radius around each cluster center (ρ_c, θ_c) , that is $(\rho_{\max}, \theta_{\max})$. After some mathematical calculation on $\theta_{\max}$ to get θ_{mlx} which are the mixtures of the mixing signals.

The remainder of this paper was organized as follow: Section II presents the BSS problem definition. Section III describes the related method which includes Hough transform and Fuzzy *k*-means clustering. Section IV illustrates the proposed method, describing how it works here. Section V presents some simulations and results. Section VI draws a brief conclusion for this work.

II. Blind Signal Separation

A. Description of Mixing Model

The sparse source signals are unobservable in a situation, $\mathbf{s}(t) = [s_1(t), \ldots, s_n(t)]^T$ is the zero-mean and is mutually statistically as independent as possible, where *n* denotes the number of sources and *t*=1, …, *N* is the instant sampling time. Sparse means that only a small number of the s_i differ significantly from zero. The degree of sparse is evaluated by the probability density function (pdf):

$$P_{sparse_i}(s_i) = \alpha_i \delta(s_i) + (1 - \alpha_i) f_i(s_i), \quad i = 1,2,\ldots,n \qquad (1)$$

where α_i is the probability that a source is inactive, $\delta(\cdot)$ denotes Dirac's delta and $f_i(s_i)$ is the pdf of the *i*th source when it is active [19]. The actual acoustics has higher degree of sparse in frequency domain than in time domain. The available sensor vector $\mathbf{x}(t) = [x_1(t), \ldots, x_m(t)]^T$, where *m* is the number of sensors, is given by:

$$\mathbf{x}(t) = \mathbf{A}\mathbf{s}(t) \qquad (2)$$

where $\mathbf{A} \in \mathbf{R}^{m \times n}$ is an unknown mixing matrix and non-singular. The definition of underdetermined case satisfies $n \geq m$, and the number of sensors *m*=2 is considered in this paper.

B. Source Signal Recovery

The sparsity means that only a small number of the s_i differ significantly from zero. The base vectors of mixing matrix would be exhibited on space domain when signals are sparse. The aim is to estimate the base vectors of mixing matrix with sensor signals only, and then to recover the original signals by maximizing the posterior distribution that is formed as [15]:

$$P(\mathbf{s}|\mathbf{x}, \mathbf{A}) = \prod_{t=1}^{T} P(\mathbf{s}(t)|\mathbf{x}(t), \mathbf{A}) \qquad (3)$$

According to (2) and Bayes' rule, we can obtain the log-likelihood by taking the logarithm of (2) as:

$$\begin{aligned} L(\mathbf{s}) = \sum_{t=1}^{T} \{ & -0.5(\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t))^T \Sigma^{-1} (\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t)) \\ & + \log\{P(\mathbf{s}(t))\}\} + \beta \end{aligned} \qquad (4)$$

where Σ indicates noise covariance matrix and β is a constant irrelevant to **s**(*t*). In order to maximize the (4), the gradient of (4) is derived with respect to **s**(*t*), as:

$$\nabla_{\mathbf{s}(t)} L(\mathbf{s}(t)) = \mathbf{A}^T \Sigma^{-1} (\mathbf{x}(t) - \mathbf{A}\mathbf{s}(t)) + \nabla_{\mathbf{s}(t)} \log\{p(\mathbf{s}(t))\} \quad (5)$$

Therefore, we can recover original signals gradually by follow iteration:

$$\mathbf{s}^j(t) = \mathbf{s}^{j-1}(t) + \eta \nabla_{\mathbf{s}(t)} L(\mathbf{s}^{j-1}(t)) \qquad (6)$$

where the superscript of **s** indicates the iteration index.

III. Related Method

A. Fuzzy k-means Clustering

Cluster analysis could be an effective approach to clearly differentiate various types of samples. It divides sampling data into several groups so that similar data objects belong to the same cluster and different data objects belong to separate ones. Here we use it on the normal parameterization to get the cluster centers which the number is assumed to be known already.

The fuzzy *k*-means clustering algorithm seeks a minimum of a heuristic global cost function:

$$J_{fuz} = \sum_{i=1}^{c} \sum_{j=1}^{n} [\hat{p}(\omega_i | x_j, \hat{\theta})]^b \left\| x_j - \mu_j \right\|^2 \qquad (7)$$

where *b* is a free parameter chosen to adjust the "blending" of different clusters and its range is between 1 and infinite, that is 1<*b*<∞. Further more, if *b* is set to 0, then J_{fuz} is merely a sum-of-squared errors criterion with each pattern assigned to a single cluster. For *b*>1, the criterion allows each pattern to belong to multiple clusters.

For a set of unlabeled data **x** = {x_1, x_2,…, x_{n}}, where *n* is the number of data points. The cluster membership functions are equivalent to the probabilities for each point are normalized as (8),

$$\sum_{i=1}^{c} \hat{p}(\omega_i | x_j, \hat{\theta}) = 1, \quad j = 1,2,\ldots,n \; , \qquad (8)$$

where $\hat{\theta}$ is the parameter vector for the membership functions, and the sum of all probabilities for each sampling point simply equals to the value of one. The distance between each data point and every cluster centers is normalized by (9).

$$d_{ij} = \left\| x_j - \mu_i \right\|^2 \qquad (9)$$

And the probability of category memberships for each point is going to be adjusted within each iteration loop of the fuzzy *k*-means clustering algorithm by (10).

$$\hat{p}(\omega_i|x_j,\hat{\theta})=\frac{(1/d_{ij})^{1/(b-1)}}{\sum_{r=1}^{c}(1/d_{rj})^{1/(b-1)}} \qquad (10)$$

At early iterations, the means locate near the center of the full data set because each point has a non-negligible "probability" in each cluster, and it will be updated by (11) during the loops of clustering.

$$\mu_i=\frac{\sum_{j=1}^{n}[\hat{p}(\omega_i|x_j,\hat{\theta})]^b x_j}{\sum_{j=1}^{n}\hat{p}(\omega_i|x_j,\hat{\theta})^b} \qquad (11)$$

All the means separate and each membership tends toward the value 1 or 0 at subsequent iterations. The cost function is minimized when cluster centers are near the points with high probabilities of presence. At the end of iterations, the change in means and probabilities are less so that clustering becomes convergent.[20]

B. Hough Transform

The Hough Transform (HT) was originally proposed by Hough in 1959. It is a technique to be used to isolate features of a particular shape within a digital image. The classical HT was used to detect the lines in an image at first, but later it was invented by Richard Duda and Peter Hart in 1972, who called it "generalized Hough Transform." The generalized HT can be used to identify the arbitrary shapes such as circles, ellipses.

Specifically applied in this research, in order to separate the mixing signals, the HT is used to detect the lines on the sparse distribution image such as Figure 4, and the slope of the lines represent the mixture of the mixing vectors. The way of how it works will be explained in Section IV later.

The HT maps the straight line on a Cartesian image space (x,y) into a point on another coordinate plane which is called normal parameterization (ρ,θ) via (12):

$$\rho = x\cos\theta + y\sin\theta, \qquad (12)$$

where ρ is the distance from the origin to the line which is passed through one point in the Cartesian plane, and the slope of line is varied from 0 to 2π. θ is the orientation of ρ with respect to the x-axis.

The HT can start by choosing a discrete set of ρ and θ to use. For each pixel (x,y) in the image, it computes (12) for each value of θ and place the result in the appropriate position in the (ρ,θ) array. At the end, the values of (ρ,θ) with the highest values in the array will correspond to strongest lines in the image. There are two derivative formulas which are showed below:

$$\rho=\sqrt{x^2+y^2}\left(\frac{x}{\sqrt{x^2+y^2}}\cos\theta+\frac{y}{\sqrt{x^2+y^2}}\sin\theta\right) \qquad (13)$$

$$\rho=\sqrt{x^2+y^2}\,\sin(\theta+\phi) \qquad (14)$$

It is obvious that if we plot the possible (ρ,θ) values defined by each one (x_i,y_i), everyone point in Cartesian image space is corresponded to one sinusoids curves in the polar Hough parameter space. This *point-to-curve* transformation is the Hough transform for straight lines.

For example, there are ten points illustrated in Figure1 that are on the same line. They are converted to the zero-polar plane as Figure 2 by (12) and the range of θ value here is $[-2\pi,2\pi]$. It is easy to see that the period in Figure 2 is π. Here in this research, the range of theta is set to be $[-\pi/2,\pi/2]$.

In Figure 2, the location (ρ,θ) in the *ρ-θ plane* where they crossover together indicates the line in the *x-y plane* that pass through both points. And the arctangent of θ_{mix} is exactly the slope of the line.

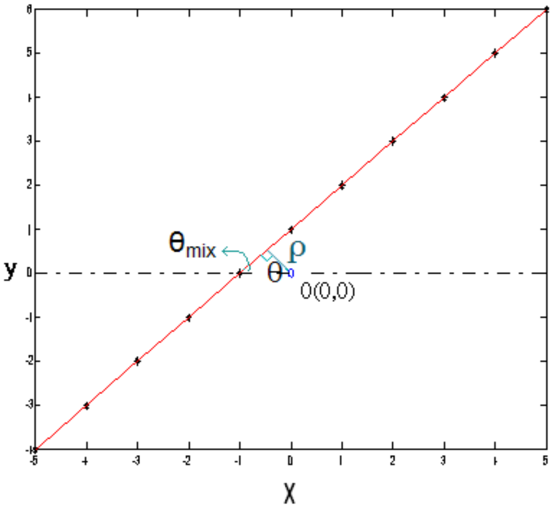

Figure.1 Ten points on the x-y plane

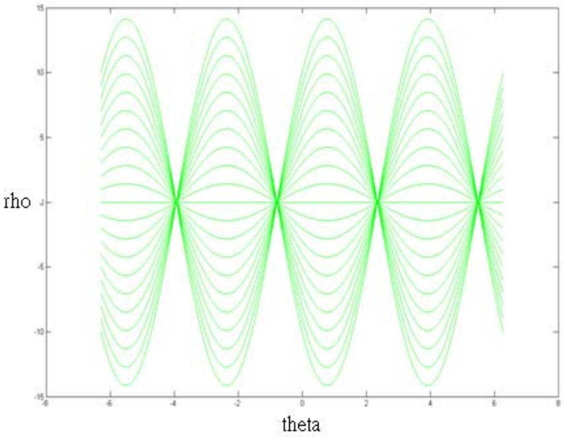

Figure. 2 Two points correspond to two sinusoid on the zero-polar space

IV. Proposed Method

This section describes how the algorithm works and the flow chart is shown in Figure 3. In the beginning, three continuous and non-periodic source signals are received by two sensors, that the mixed signals are named m_1 and m_2 and the mixing matrix $\boldsymbol{A}$ is assumed already. Because the sparse characteristic is more obvious in frequency domain, we use Fourier transform to transfer data in time domain to frequency spectrum, M_1 and M_2 are denoted the real based of FT representation of m_1 and m_2 separately. For the property of sparse representation, the vectors of M_1 and M_2 form the gathered lines of mixtures on a 2-D coordinate plane in Figure 4. But in reality, there may be some noise to cause non-sparse points in general cases. Therefore, the gathered lines often seemed as distributed tendency of mixtures.

The next step is using HT to convert data points from the 2-D coordinate plane to zero-polar plane. The vectors M1 and M2 are corresponded to x and y in (12). In Figure 4, it is obvious that the distance from the original point (0,0) to all the directional lines is closed to zero. Each one point in Cartesian plane is converted to a sinusoid in the zero-polar plane, and those sine waves are crossoverred to form three central intersections near around the theta axis, i.e., rho is closed to zero. Due to the theorem of HT, we know that the three gathered lines in Figure 4 are corresponded to three intersections in Figure 5.

Figure 5 shows all the sine curves transferred from the data points in the Cartesian plane, and Figure 6 shows the amount of superimpositions where the curves are overlapped in Figure 5. Because the amount of superimpositions are less at $|\rho| \geq 20$ and very unclearly, the scale bound in Figure 6 is set to be $|\rho| \leq 20$.

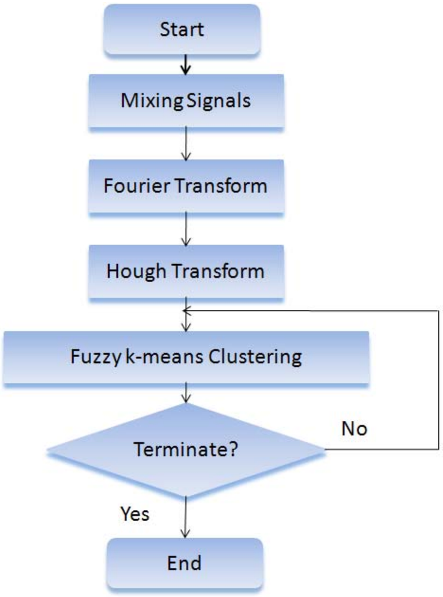

Figure. 3 The flow chart of the proposed algorithm

Fuzzy k-means clustering is employed to cluster the superimposed centers in the polar space directly. The three means vectors are given in random at first, the particles (ρ_j, θ_j) are equal to x_j in (9), then it starts to update the values of means (11) and membership functions (10) in terms till there is almost nothing to change. After the three cluster centers are found, in order to find the maximum of superimposed number in this region, we chose a radius r to form three circles around them. The maximum superimpositions are indicated to three coordinates (ρ_i, θ_i) in each circle, where i is the number of cluster centers. These three points are corresponded back to three straight lines in Cartesian plane in Figure 4.

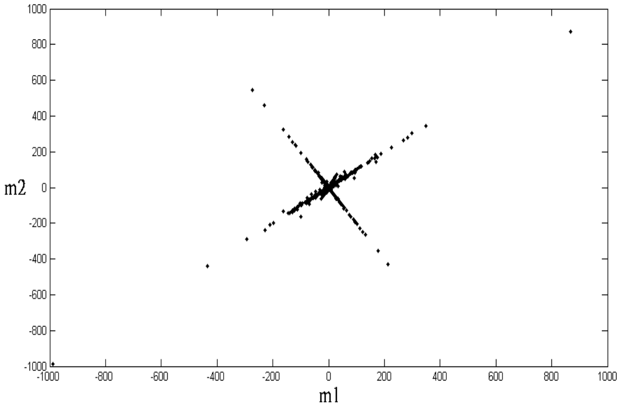

Figure. 4 The sparse mixture in a 2-D space (involve three distinguishable base vectors)

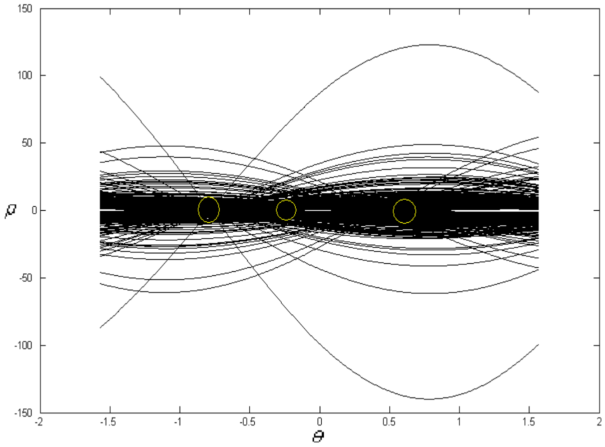

Figure. 5 The sparse mixture in a 2-dimensional space is converted to the zero-polar plane

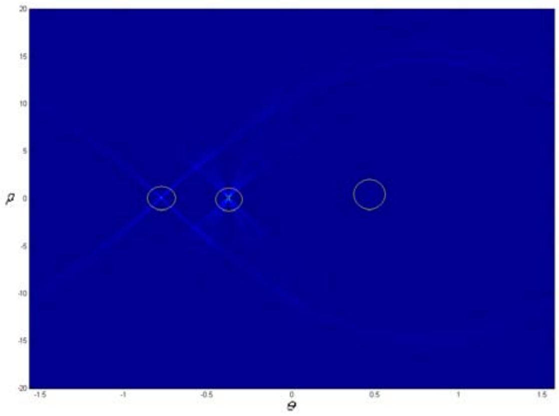

Figure. 6 The amount of superimpositions at $\rho \le 20$

The mixing matrix are equaled to the slope of directional lines in Figure 4, also the θ_{mix} as mentioned in Figure 1. In the end, if the value of θ_i is positive, then it has to be subtracted by $\pi/2$; On the contrary, if it is negative, then add $\pi/2$ on it.[21]

V. SIMULATION

In order to confirm the efficiency and robustness of the proposed algorithm, the BSS simulations consider one kind of mixing matrix and three numbers of source signals, and implement three existing algorithms for performance comparison, conveniently they are named CPSO [22], PF [13], FC [17] later. The sampling number we take from each mixing signals is 30,000, and the source signals are shown in the Figure 7, which the frequency spectrum is shown in Figure 8 in order to present the sparse characteristic.

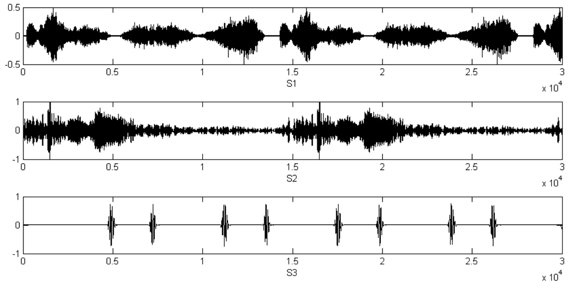

Figure. 7 The three source signals in time domain

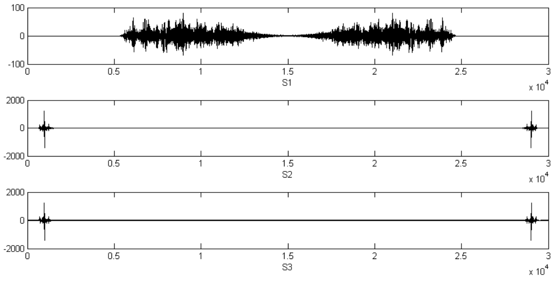

Figure. 8 The three source signals in the frequency spectrum

Each simulation is carried out 20 independent runs. The sparse source signals are the actual sounds. A performance evaluation is defined as:

$$PI = \frac{\sum_{i=1}^{k} \left| Rbv_i - Ebv_i \right|}{k} \tag{21}$$

where Rbv_i denotes the ith actual mixing vector, and Ebv_i denotes the ith estimation. PI means the error percent of the mixing matrix, that is the larger the PI is, the more unprecise result is. In final, the average of *PI* by 20 independent runs will be presented. In this case, three source signals and a mixing matrix are involved. The mixing matrix with scatter base vectors, $Rbv = [2.5,\ -2,\ 1]$, are presented as:

$$\mathbf{A} = \begin{pmatrix} 0.3714 & 0.4472 & 0.7071 \\ 0.9285 & -0.8944 & 0.7071 \end{pmatrix}$$

The received mixtures are assumed already and the 2-D plane are plotted in Figure 4, and the predefined parameters of involved algorithms are:

- Proposed algorithm: grid scale of θ=0.01 and ρ =0.1, b=1.5, r=0.05
- CPSO: $sz = 10$, $k = 6$, $\alpha = 95$ and $g_{\max} = 50$
- PF: the grid scale I=720 and λ=55
- FC: the tolerant error ε is 10^{-3}

And the averages of estimated base vectors and PI are present in TABLE I.

TABLE I The comparisons of identity accuracy and run time

	Algorithms				
	Real	Proposed method	CPSO	PF	FC
bv1	2.5	2.4979	2.5000	2.5882	2.4161
bv2	-2	-2.0024	-1.9959	-1.9522	-2.0386
bv3	1	1.0092	0.9999	1.0220	0.9588
PI		0.0031	0.0018	0.1027	0.0819
Time(s)		0.8299	1.6165	2.1676	1.5120

Apparently, under the same condition, the proposed algorithm is the fastest one. Compare with the CPSO, although the mixture they got was almost two times precisely than us, but the time they took was two times than us, too. The FC uses Fuzzy *C*-means clustering to cluster the data in the 2-D plane directly. The difference between FC and us is that we use Hough transform before we do the fuzzy *k*-means clustering. After we converted the coordinate and use another type of data, the result we got was more precise than FC. Also the computational time is much less than them.

In conclusion, on the viewpoint of saving time and get a precise value that the error is small and acceptable meanwhile, our proposed algorithm can achieve both of them in the same time.

VI. Conclusion

The mixing matrix identification which has good accuracy and low cost of computation was proposed in this paper. The received mixtures are converted to frequency spectrum. Parameters of the blind mixing matrix can be clearly visible on the zero-polar plane when mixtures had been transferred by Hough transform. The precise parameters are further found out by fuzzy *k*-means clustering. At last, because the θ value we get so far is the orientation of ρ with respect to the x-axis in Figure 1, the mixture is the slope of the directional line. In order to get the mixture, the θ value need to be plus or subtracted by $\frac{\pi}{2}$ to get θ_{mix} .

From the simulation results, performance of the proposed algorithm is better than those of PF algorithm and FC algorithm, and as accuracy as that of CGPSO. Additionally, it only spends about half computational time of others. By the advantages, the proposed algorithm has a benefit to deal with an on-line BSS problem.

References

[1] A. Cichocki, and R. Unbehauen, "Robust neural networks with on-line learning for blind identification and blind separation of sources," *IEEE Trans. on circuits and systems-I: fundamental theory and applications*, vol. 43, 1996.

[2] Y. cZhang and S. A. Kassam, "Robust rank-EASI algorithm for blind source separation," *IEE Proceedings-Communications*, vol. 151, issue: 1, pp. 15-19, 2004.

[3] A. Tonazzini, L. Bedini, and E. Salerno, "A markov model for blind image separation by a mean-field EM algorithm," *IEEE Trans. on Image Processing*, vol. 15, issue: 2, pp. 473-482, 2006.

[4] I. Schiessl, M. Stetter, J. E. W. Mayhew, N. McLoughlin, J. S. Lund, and K. Obermayer, "Blind signal separation from optical imaging recordings with extended spatial decorrelation," *IEEE Trans. on Biomedical Engineering*, vol. 47, issue: 5, pp. 573-577, 2000.

[5] E. Tangdiongga, N. Calabretta, P. C. W. Sommen, and H. J. S. Dorren, "WDM monitoring technique using adaptive blind signal separation," *IEEE Photonics Technology Letters,* vol. *13*, issue: 3, pp. 248-250, 2001.

[6] O. Yilmaz and S. Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 52, issue: 7, pp. 1830-1847, 2004.

[7] T. W. Lee, M. S. Lewicki, M. Girolami, and T. J. Sejnowski, "Blind source separation of more sources than mixtures using overcomplete representations," *Signal Processing Letters*, vol. 6, issue: 4, pp. 87-90, 1999.

[8] Y. Li and J. Wang, "Sequential blind extraction of instantaneously mixed sources," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 50, issue: 5, pp. 997-1006, 2002.

[9] J. Even and E. Moisan, "Blind source separation using order statistics," *Signal Processing*, vol. 85, issue: 9, pp. 1744-1758, 2005.

[10] P. Pajunen, "Blind source separation using algorithmic information theory," *Neurocomputing,* vol. 22, issue: 1-3, pp. 35-48, 1998.

[11] D. T. Pham and F. Vrins, "Local minima of information-theoretic criteria in blind source separation," *IEEE Signal Processing Letters,* vol. 12, issue: 11, pp. 788-791, 2005.

[12] T. P. Dinh, "Mutual information approach to blind separation of stationary sources," *IEEE Trans. on Information Theory*, vol. 48, issue: 7, pp. 1935-1946, 2002.

[13] P. Bofill and M. Zibulevsky, "Underdetermined blind source separation using sparse representations," *Signal Processing*, vol. 81, pp. 2353-2362, 2001.

[14] Q. Lv and X. D. Zhang, "A unified method for blind separation of sparse sources with unknown source number," *IEEE Signal Processing Letters,* vol. 13, issue: 1, pp. 49-51, 2006.

[15] Z. Shi, H. Tang, W. Liu, and Y. Tang, "Blind source separation of more sources than mixtures using generalized exponential mixture models," *Neurocomputing*, vol. 61, pp. 461-469, 2004.

[16] P. O. Grady and B. Pearlmutter, "Soft-LOST: EM on a mixture of oriented lines," in *Proc. of ICA 2004*, ser. *Lecture Notes in Computer Science Granada,* Spain. 3195, pp. 430-436, 2004.

[17] C. C. Liu, T. Y. Sun, K. Y. Li, and C. L. Lin, "Underdetermined blind signal separation using fuzzy cluster on mixture accumulation," in *Proc. Int. Symp. on Intelligent Signal Processing and Communication Systems*, pp. 455-458, 2006.

[18] S. V. Vaerenbergh and I. Santamaria, "A spectral clustering approach to underdetermined postnonlinear blind source separation of sparse sources," *IEEE Trans. on Neural Networks,* vol. 17, issue: 3, pp. 811-814, 2006.

[19] D. Luengo, I. Santamaria, and L. Vielve, "A general solution to blind inverse problems for sparse input signals: deconvolution, equalization and source separation," *Neurocomputing*, vol. 69, pp. 198-215, 2005.

[20] R.O.Duda, P.E.Hart, and D.G.Stork, *Pattern classification,* Canada,2001

[21] A.Mcandrew, *Introduction to digital image processing with matlab,* Course techonology,2004

[22] C. C. Liu, T. Y. Sun, K. Y. Li, S. T. Hsieh, and S. J. Tsai, "Blind sparse source separation using cluster particle swarm optimization technique," in *Proc. Int. Conf. on Artificial Intelligence and Applications*, pp. 549-217, 2007.