SOUND SOURCE SEPARATION WITH DISTRIBUTED MICROPHONE ARRAYS IN THE PRESENCE OF CLOCK SYNCHRONIZATION ERRORS

Zicheng Liu

Microsoft Research, Redmond,WA 98052

ABSTRACT

We first present our studies on how clock synchronization errors affect the performance of sound source separation with a distributed microphone array. We show that our previously-proposed energy-based sound source separation method is robust to constant clock shift errors but is more sensitive to clock drift errors. We then propose a novel technique to address the clock drift errors. The key observation is that as the amount of clock drift increases, so does the correlation between the energies of the separated sources which are obtained from the Independent Component Analysis (ICA). Based on this observation, we propose an optimization technique to solve the clock drifting parameter. Experiment results are shown to validate our approach.

Index Terms— Sound source separation, synchronization, clock drift

1. INTRODUCTION

In this paper, we report our studies about the effect of clock synchronization errors on sound source separation performance with a distributed microphone array. As indicated in [1, 2, 3], one typical scenario of distributed microphone array is in meeting rooms where many meeting participants bring portable devices such as laptops and PDAs to meeting rooms. The microphones on these devices form an ad hoc array. Such an ad hoc array is different from a conventional microphone array in multiple aspects. First, the microphones in an ad hoc array are spatially distributed. Typically the individual microphones are closer to the meeting participants. Second, the microphones may not share a common clock thus there may be synchronization errors. Third, the array geometry is unknown. Fourth, the microphones have different and unknown gains. Finally, the microphones have different signal to noise ratios.

Due to these differences, many audio processing tasks such as sound source localization and source separation require different algorithms [1, 4, 2, 3]. In particular for sound source separation, people observed that energy-based depermutation scheme works better than traditional phase-based depermutation method due to the fact that the microphones are spatially distributed.

The experiments in [1, 4] were conducted under the assumption that the microphones are synchronized. In many application scenarios, the individual microphones in an ad hoc microphone array are usually attached to separate capture cards each having their own clocks. Therefore, there may be clock synchronization errors between difference channels. The focus of this paper is to study the effect of clock synchronization errors on sound source separation performance with distributed microphone arrays.

Lienhart et. al [5] also studied the clock synchronization effect on sound source separation performance. One main difference between their work and ours is that they measured the performance of ICA with the conventional phase-based depermutation scheme while our focus is to measure the performance of ICA with energy-based depermutation scheme which, as reported in [1, 4], works better than phased-based depermutation scheme. In addition, we propose a novel technique to estimate clock drift parameter.

2. EFFECT OF CLOCK SYNCHRONIZATION ERRORS

Assume there are two microphones each with their own clocks and the two microphones have the same sampling rate (with respect to their own clocks). Suppose there is a reference clock which can be thought of as a ground truth clock. Denote $x_1(i)$ and $x_2(i)$, $i = 1, ...$, as the reference signals which are captured (virtually) by the two microphones according to the reference clock. Let $\hat{x}_1(i)$ and $\hat{x}_2(i)$ denote the actual signals which are captured by the two microphones according to their own clocks. Without loss of generality, we assume the first microphone's clock is the same as the reference clock, that is, $\hat{x}_1(i) = x_1(i)$. We assume there is a linear relationship between the second microphone's clock and the reference clock [6], that is, $\hat{x}_2(i) = x_2(k * i + b)$ where b corresponds to a constant shift error while k corresponds to clock drift errors which grow over time.

Let r denote the sampling rate of the two microphones. For convenience, we represent k as $k = \frac{r}{r+\mu}$ where $|\mu|$ is equal to the average number of samples drifted per second. We sometimes call μ the *drift rate*. For example when $\mu = 1$, one second on microphone#2's clock is equal to $\frac{r}{r+1}$ seconds on the reference clock. As a result, for each second on the

reference clock, the microphone#2 actually samples $\frac{r+1}{r} * r = r + 1$ samples. Note that μ is a real number which can be a fraction or negative number.

We use artificially generated data to simulate clock synchronization errors and measure the effect on sound source separation performance. First we use the image method [7] to synthesize two channel data with clocks perfectly synchronized. The data synthesis setting is similar to what was used in [1] where the signal-to-noise ratio is 20dB and the simulated room's reverberation time is $T_{60} = 300ms$. The two microphones are located at $(203.2, 228.6, 101.6)$m and $(101.6, 228.8, 101.6)$m, respectively. The two speakers are located at $(254, 228.6, 101.6)$m and $(50.8; 228.6; 101.6)$m, respectively. Each speaker speaks for 8 seconds with 100% overlap. The sampling rate is 8kHz. We use a frame length and frame-shift of 4096 and 1024 samples, respectively, for short window FFT transform.

Let $x_1(i)$ and $x_2(i)$ denote the two channel data in time domain, respectively. For any given μ and b, we generate $\hat{x}_2(i)$ through resampling based on the formula $\hat{x}_2(i) = x_2(\frac{r}{r+1}i + b)$. The resampling is performed by using linear interpolation. We apply the energy-based source separation technique [1] to $x_1(i)$ and $\hat{x}_2(i)$. The energy-based source separation technique is also an ICA-based source separation technique. It differs from conventional ICA-based source separation technique such as [8] in the depermutation phase where energies of the source separation filters are used for depermutation.

To measure the effect of constant shift error, we first set $\mu = 0$ and vary b. Figure 1 shows the Signal to Interference Ratio (SIR) of the separated signals as a function of b. We can see that the source separation performance is quite robust to clock shift errors. When b is less than 100 which is equal to 12.5 milliseconds of clock misalignment, the SIR only drops 0.3dB.

We then set $b = 0$ and vary μ to measure the effect of drift error. Figure 2 shows the SIR as a function of μ. We can see that the sound source separation performance is more sensitive to drift errors. But it is much better than TDOA-based depermutation scheme as reported [5]. According to [5], there is a 5dB drop for a drift rate of one sample per second with sampling rate of 16kHz. Note that a drift rate of one sample per second with 16kHz sampling rate is equivalent to drift rate of half second with 8kHz sampling rate. As shown in Figure 2, the SIR drops less than 1dB when $\mu = 0.5$. Therefore, energy-based depermutation scheme is significantly more robust against drift errors than TDOA based depermutation.

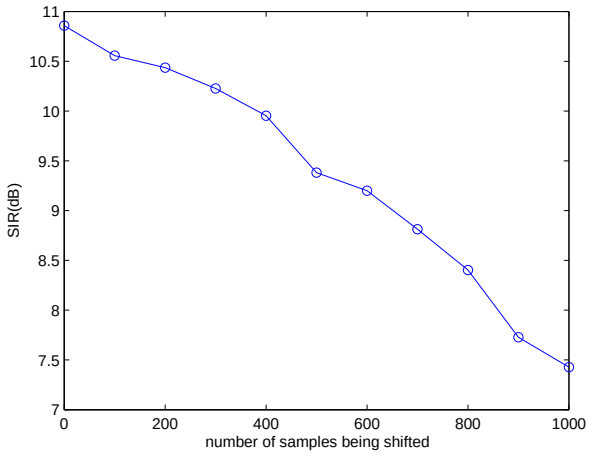

Fig. 1. Sound source separation performance vs. the amount of shift between two channels. The horizontal axis is the number of samples being shifted. The vertical axis is the Signal to Interference Ratio (SIR). The sampling rate is 8kHz.

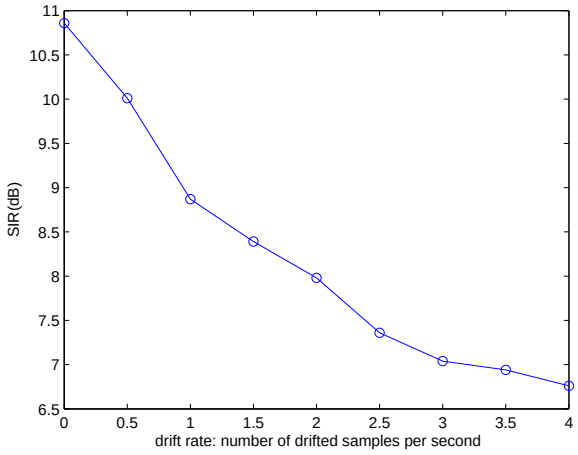

Fig. 2. Sound source separation performance vs. the clock drift errors. The horizontal axis is μ: the number of drifted samples per second. The vertical axis is the Signal to Interference Ratio (SIR). The sampling rate is 8kHz.

3. DRIFT PARAMETER ESTIMATION

In this section, we present a technique to estimate the drift parameter. We focus on clock drift error because, as shown in previous section, the shift errors do not cause serious performance degradation.

Let $\hat{x}_1(i)$ and $\hat{x}_2(i)$ denote two signals in time domain with clock drift errors between the two channels. Let $y_{\lambda,2}(i) = \hat{x}_2(\frac{r+\lambda}{r}i)$ where r is the sampling rate. We would like to find the correct λ so that $y_{\lambda,2}(i)$ is aligned with $\hat{x}_1(i)$.

For notational convenience, denote $y_{\lambda,1} = x_1$. Let $Y_{\lambda,j}(\omega,t)$ denote the FFT transform of $y_{\lambda,j}$ where ω is the frequency index and t is the frame index. Denote $Y_\lambda = (Y_{\lambda,1}, Y_{\lambda,2})^T$. Let $W_\lambda(\omega) = \phi(Y_\lambda(\omega,\cdot))$ denote the separation matrix resulted from ICA by maximizing the Kurtosis of $W_\lambda(\omega)Y_\lambda(\omega,t)$ [9].

Let $S_\lambda(\omega,t) = (S_{1,\lambda}(\omega,t), S_{2,\lambda}(\omega,t))^T$ denote the separated result, that is, $S_\lambda(\omega,t) = W_\lambda(\omega)Y_\lambda(\omega,t)$. The key observation is that if $Y_{\lambda,1}$ is aligned with $Y_{\lambda,2}$, the separated signals $S_{1,\lambda}$ and $S_{2,\lambda}$ should have the least amount of correlation. Since the phase information is noisy, we use the correlation of the magnitude while ignoring the phases. We call it the *energy correlation score*, which is defined as

$$\Pi((S_{1,\lambda}, S_{2,\lambda})^T) = \frac{1}{F}\sum_{\omega=1}^{F} \frac{\sum_{t=1}^{N} |S_{1,\lambda}(\omega,t)||S_{2,\lambda}(\omega,t)|}{\sqrt{\sum_{t=1}^{N} |S_{1,\lambda}(\omega,t)|^2}\sqrt{\sum_{t=1}^{N} |S_{2,\lambda}(\omega,t)|^2}} \quad (1)$$

where F is number of frequency bands and N is the number of frames. Note that to evaluate energy correlations score, all we need is the ICA results for each frequency band. There is no need to perform de-permutation.

λ is then estimated by solving the following optimization problem

$$Minimize\ \Pi(W_\lambda Y_\lambda). \quad (2)$$

We use an iterative quadric function approximation approach to solve this optimization problem. Figure 3 is an outline of the algorithm. The algorithm maintains three points $\lambda_1^k < \lambda_2^k < \lambda_3^k$ so that

$$\Pi(W_{\lambda_1^k} Y_{\lambda_1^k}) \geq \Pi(W_{\lambda_2^k} Y_{\lambda_2^k}) \quad (3)$$
$$\Pi(W_{\lambda_3^k} Y_{\lambda_3^k}) \geq \Pi(W_{\lambda_2^k} Y_{\lambda_2^k}) \quad (4)$$

where k is the iteration counter. Denote

$$g_j^k = \Pi(W_{\lambda_j^k} Y_{\lambda_j^k}), j = 1,2,3. \quad (5)$$

We approximate function $\Pi(W_\lambda Y_\lambda)$ by a quadratic function $h(\lambda)$ which is defined by the three points (λ_j^k, g_j^k), $j = 1,2,3$. Assume $h(\lambda)$ has the form:

$$h(\lambda) = a\lambda^2 + b\lambda + c. \quad (6)$$

The coefficients a, b, c are determined the following equations:

$$\begin{aligned} a(\lambda_1^k)^2 + b\lambda_1^k + c &= g_1^k \\ a(\lambda_2^k)^2 + b\lambda_2^k + c &= g_2^k \\ a(\lambda_3^k)^2 + b\lambda_3^k + c &= g_3^k \end{aligned} \quad (7)$$

Initialization: Obtain $\lambda_1^0, \lambda_2^0, \lambda_3^0$ through linear search. Set $g_j^0 = \Pi(W_{\lambda_j^0} Y_{\lambda_j^0})$,j=1,2,3. Set $k = 0$.

Step 1: Solve equation 8 to obtain a, b, c. Set $\lambda_4^k = \frac{b}{2a}$.

Step 2:

- Run the fixed point ICA algorithm on $Y_{\lambda_4^k}(\omega)$ per frequency band ω and denote $W_{\lambda_4^k}(\omega)$ as the resulting separation matrix.
- Compute the energy correlation of the separated signals: $g_4^k = \Pi(W_{\lambda_4^k} Y_{\lambda_4^k})$.
- Discard the λ_j^k with the largest correlation score g_j^k, and set $\lambda_1^{k+1} < \lambda_2^{k+1} < \lambda_3^{k+1}$ to be the rest three λ_j^k in sorted order, and set g_j^{k+1}, $j = 1,2,3$, to be their corresponding correlation scores.

Step3 : If $max(|\lambda_1^{k+1} - \lambda_2^{k+1}|, |\lambda_3^{k+1} - \lambda_2^{k+1}|) < \epsilon$, stop. Otherwise, set $k = k + 1$ and goto Step 1.

Fig. 3. Algorithm outline

It is simple to solve this 3×3 linear system of equations and obtain a, b, c. The minimum value of the quadratic function $h(\lambda)$ is achieved as

$$\lambda_4^k = -\frac{b}{2a}. \quad (8)$$

From λ_1^k, λ_2^k, λ_3^k, λ_4^k, we remove the one that corresponds to the largest energy correlation score, and denote $\lambda_1^{k+1} < \lambda_2^{k+1} < \lambda_3^{k+1}$ as the remaining three λ_j^ks. The algorithm stops if the difference between the current estimate and the optimal value, which is upper bounded by $max(|\lambda_1^{k+1} - \lambda_2^{k+1}|, |\lambda_3^{k+1} - \lambda_2^{k+1}|)$, is smaller than a predefined threshold ϵ. Otherwise, it sets $k = k + 1$ and goes to the next iteration.

To initialize, we need to search for three initial values $\lambda_1^0 < \lambda_2^0 < \lambda_3^0$ satisfying

$$\Pi(W_{\lambda_1^0} Y_{\lambda_1^0}) \geq \Pi(W_{\lambda_2^0} Y_{\lambda_2^0}) \quad (9)$$
$$\Pi(W_{\lambda_3^0} Y_{\lambda_3^0}) \geq \Pi(W_{\lambda_2^0} Y_{\lambda_2^0}) \quad (10)$$

This is done by starting with an arbitrary initial value λ_0, and evaluate objective function $\Pi(W_\lambda Y_\lambda)$ at $\lambda_0 - \Delta\lambda$, λ_0, and $\lambda_0 + \Delta\lambda$ where $\Delta\lambda$ is a user specified parameter. If $\Pi(W_{\lambda_0-\Delta\lambda} Y_{\lambda_0-\Delta\lambda}) \leq \Pi(W_{\lambda_0} Y_{\lambda_0})$, we search for $u \geq 1$ so that $\Pi(W_{\lambda_0-(u+1)\Delta\lambda} Y_{\lambda_0-(u+1)\Delta\lambda}) > \Pi(W_{\lambda_0-u\Delta\lambda} Y_{\lambda_0-u\Delta\lambda})$. We then use the last three points $(\lambda_0 - (u+1)\Delta\lambda, \lambda_0 - u\Delta\lambda, \lambda_0 - (u-1)\Delta\lambda)$as the initial $\lambda_1^0, \lambda_2^0, \lambda_3^0$.

The procedure is similar if $\Pi(W_{\lambda_0+\Delta\lambda} Y_{\lambda_0+\Delta\lambda}) \geq \Pi(W_{\lambda_0} Y_{\lambda_0})$.

4. EXPERIMENT RESULTS

We use simulation data to evaluate the performance of our algorithm. The data is generated in the same way as described

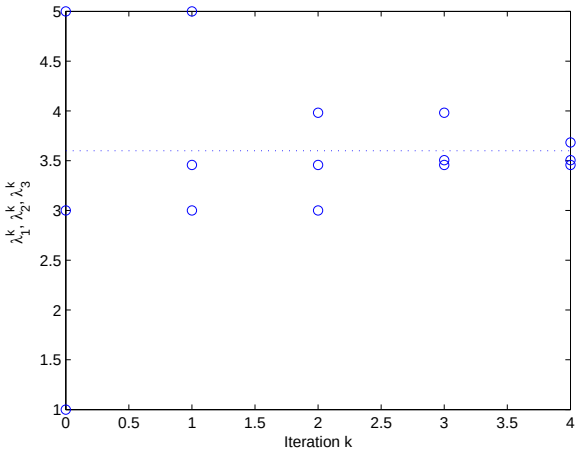

Fig. 4. Experiment result: The horizontal axis is the iteration number. The vertical axis are values of λ_1^k, λ_2^k, and λ_3^k.

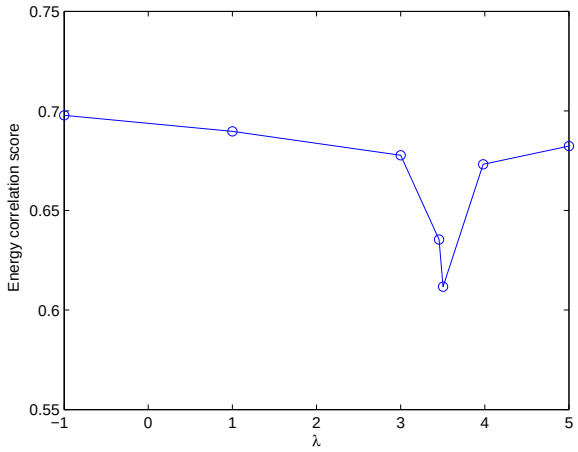

Fig. 5. Plot of the energy correlation scores at the λ values which are generated during the optimization procedure including both the linear search at the initialization phase and the quadratic search phase. The horizontal axis is λ. The vertical axis is the energy correlation score.

in section 2. There are two speakers and two microphones. The two speakers talk simultaneously for 8 seconds. The sampling rate is 8kHz. We generate drifted signal $\hat{x}_2$ by setting the drift rate μ to be equal to 4.6 samples per second. We then apply our drift estimation algorithm to x_1 and $\hat{x}_2$. We use $\lambda_0 = -1$, $\Delta\lambda = 2$, and $\epsilon = 0.2$.

At the initialization phase, it takes three iterations of linear search to obtain $\lambda_1^0 = 1$, $\lambda_2^0 = 3$, and $\lambda_3^0 = 5$.

After that, the algorithm enters the quadratic search. It finishes after 4 iterations. Figure 4 shows the resulting λ_1^k, λ_2^k, and λ_3^k for each iteration.

At the end of the optimization procedure, the estimated solution is $\lambda = 3.5044$ which is very close to the ground truth solution of $\lambda = 3.6$. Figure 5 plots the energy correlation scores at those λ values which are generated during the optimization procedure.

5. CONCLUSIONS

We have presented our studies of clock asynchronization effect on the performance of sound source separation with a distributed microphone array. Our numerical experiments showed that the energy-based sound source separation method is robust to clock shift error. We proposed a technique to estimate the clock drift error by minimizing the energy correlation scores of the separated signals. We presented experiment results validating the effectiveness of the proposed algorithm.

6. REFERENCES

[1] Jacek Dmochowski, Zicheng Liu, and Phil Chou, "Bline source separation in a distributed microphone meeting environment for improved teleconferencing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, Las Vegas, Nevada*, 2008.

[2] Zicheng Liu, Zhengyou Zhang, Li-Wei He, and Phil Chou, "Energy-based sound source localization and gain normalization for ad hoc microphone arrays," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, Honolulu, Hawaii*, 2007.

[3] V. C. Raykar, I. Kozintsev, and R. Lienhart, "Position calibration of microphones and loudspeakers in distributed computing platforms," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 1, pp. 70–83, January 2005.

[4] E. Robledo-Arnuncio and Biing-Hwang (Fred) Juang, "Blind source separation of acoustic mixtures with distributed microphones," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2007.

[5] R. Lienhart, I. Kozintsev, S. Wehr, and M. Yeung, "On the importance of exact synchronization for distributed audio processing," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2003.

[6] Carlos H. Rentel and Thomas Kunz, "A clock-sampling mutual network time-synchronization algorithm for wireless ad hoc networks," in *IEEE Wireless and Communications and Networking Conference*, 2005.

[7] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating smallroom acoustics," *J. Acoust. Soc. Am.*, vol. 65, pp. 943–950, April 1979.

[8] H. Sawada, R. Mukai, S. Araki, and S. Makino, *Frequency-Domain Blind Source Separation, in Speech Enhancement, Eds. Benesty, S. Makino, and J. Chen*, Springer-Verlag, 2005.

[9] Ella Bingham and Aapo Hyvarinen, "A fact fixed-point algorithm for independent component analysis of complex valued signals," *International Journal of Neural Systems*, vol. 10, pp. 1–8, February 2000.