TIME-DOMAIN GCC-PHAT SOUND SOURCE LOCALIZATION FOR SMALL MICROPHONE ARRAYS

Bert Van Den Broeck [1,3], Alexander Bertrand [1,2], Peter Karsmakers [1,3], Bart Vanrumste [1,2,3], Hugo Van hamme [1], Marc Moonen [1,2]

[1]ESAT, KU Leuven
[2]IBBT, Future Healt Department
Kasteelpark Arenberg 10, 3001, Heverlee, Belgium

[3]MOBILAB, KH Kempen
Kleinhoefstraat 4, 2440, Geel, Belgium

phone: +32 (0)14 56 23 10, email: bert.van.den.broeck@.khk.be
web: [1] http://www.esat.kuleuven.be, [3] http://www.mobilab-khk.be

ABSTRACT

In sound enhancement, wireless acoustic sensor networks (WASN) are gaining popularity. This is due to recent progress on both the hard- and software side. Nowadays affordable, miniaturized and powerful signal processing hardware is available to build the individual sensor nodes. Combined with novel distributed software, allowing to share computational load among all network nodes, WASNs are able to handle complex signal processing tasks. One such task is the enhancement of speech recorded from a distance. A subtask in the enhancement process might be the estimation of the speakers position.

This paper focusses on implementing an accurate Sound Source Localizer (SSL) for estimating the position of a sound source on a digital signal processor (to be included in an individual WASN node) using as less CPU resources as possible. One of the least complex algorithms for SSL is a simple correlation, implemented in the frequency-domain for efficiency, combined with a frequency bin weighing for robustness. Together called Generalized Cross Correlation (GCC). One popular weighing called GCC PHAse Transform (GCC-PHAT) will be handled.

In this paper it is explained that for small microphone arrays, as mounted on a single acoustic node of a WASN, this frequency-domain implementation is inferior to its time-domain alternative in terms of algorithmic complexity. Therefore a time-domain PHAT equivalent will be described. Both implementations are compared in terms of complexity (clock cycles needed on a Texas Instruments C5515 DSP) and obtained results, showing a complexity gain with a factor of 146, with hardly any loss in localization accuracy.

1. INTRODUCTION

Nowadays, sound or speech enhancement finds its way to various applications such as e.g. speech controlled domotics, where speech signals need to be enhanced prior to recognition due to changing room acoustics and noise statistics. To achieve this speech enhancement, multiple microphone arrays may be used at different places in the target environment. Each microphone array is mounted on a device (further referred to as an acoustic node) with a local DSP processor and with wireless networking capabilities. Together, these acoustic nodes form a so-called wireless acoustic sensor network (WASN), where the nodes can work together to perform a certain task such as speech enhancement [5]. Recent research aims at distributed algorithms where all acoustic nodes share the computational load [1],such that one powerful processor can be replaced by multiple less performing types. This can make the overall WASN cheaper and less hungry for power. While acoustic node positions are often considered unknown, the microphone placement within one acoustic node is not. This prior knowledge on the local microphone positions can be exploited by applying a beamformer on the local microphone array. To do so position information should be estimated by each acoustic node individually, using an SSL. Since the algorithmic complexity is directly related to hardware costs and power consumption, it should be kept as small as possible while maintaining accuracy. Acoustic nodes need to be small in size since they often need to be discretely installed (e.g. for domestic purposes). As a consequence, the local SSL operates on a small-size microphone array, and this feature will be taken into account in the algorithmic design.

SSL algorithms can generally be split up in 3 major groups [2], based on: a) Time Delay Of Arrival (TDOA) estimation mostly by means of cross correlation [4] [2]. b) Steering out beams and finding high energy sound sources, often called steered power response [2]. c) Eigenvalue based algorithms such as MUltiple SIgnal Classification (MUSIC) [3].

The eigenvalue based algorithms are quite complex due to their eigenvalue decomposition. Cross correlation and steered power response based algorithms are very much related. A major difference is that steered power response can estimate the delay in fractions of the sample period at a cost of an increased number of complex rotations. Since in our setup a) estimating a TDOA with a resolution close to the sampling period is fine and b) the computational complexity need to be kept as small as possible, it is preferred to use a cross correlation based algorithm. In order to make robust TDOA estimations the correlations should be filtered, e.g., with the so-called phase amplitude transform (PHAT), as envisaged in this paper. The resulting algorithm is then referred to as GCC-PHAT [4].

When only small microphone arrays are considered, the delays are small, and then a cross correlation can be done more efficiently in the time-domain. However, then the original frequency-domain-based PHAT weighing scheme cannot be used. Therefore this paper will introduce a low complexity time-domain PHAT alternative based on an adaptive linear prediction (LP) whitening filter.

In Section 2 of this paper the frequency-domain TDOA PHAT algorithm is briefly reviewed. In Section 3, a time-domain alternative is proposed. In Section 4 the computational costs are compared for a realistic sampling frequency and microphone array dimensions. In Section 5 the results of both the time and frequency-domain implementations are compared. Finally, conclusions are drawn in Section 6.

2. FREQUENCY-DOMAIN BASED GCC-PHAT IMPLEMENTATION

In this section the frequency-domain implementation of GCC is briefly reviewed. First, two time-domain signals are individually transformed into the frequency-domain in which they are multiplied with each other (after taking the complex conjugate of one of them), giving $G_{Left,Right}$. This result is then transformed back to the time-domain to obtain a correlation function. The position corresponding to the maximum cross correlation will indicate the TDOA. On its turn the TDOA will indicate the angle of sound incidence, given the array geometry. For robustness, the popular PHAT weighting scheme can be used to obtain a unity gain for all frequency components, while preserving phases which contain the actual delay information.

$$\hat{G}_{Left,Right}(f) = \frac{G_{Left,Right}(f)}{\left|G_{Left,Right}(f)\right|}$$

Theoretically, when transforming back to the time-domain, this should make the correlation function a unit impulse function (neglecting noise, echo,…). As a result, additional peaks in the correlation (from echo paths and noise sources) will not influence the spike from the direct path as much, giving better location estimations.

Considering small arrays, this initially brings a bad resolution. For instance, when using 2 microphones at 10cm distance at a sample frequency of 16kHz the correlation contains only 9 points of interest (others are related to impossible delays). This gives an average resolution of 20° which can be easily improved by a so-called quadratic interpolation [2]. Assuming a symmetric correlation function quadratic interpolation calculates a function $R = at^2+bt+c$ through the maximum correlation point and its left and right neighbors and then recalculates the position of the maximum at $-b/(2a)$. When using this quadratic interpolation, it is even more important that the correlation peaks of the direct and echo paths do not influence each other, which makes PHAT especially desirable in this case.

Reconsidering the limited resolution, it is noticed that the calculation of the cross correlation function in the frequency-domain is not very efficient in case of small microphone arrays. Block lengths of N=256 (16ms) and larger are not uncommon to compute reliable cross correlation functions in the GCC-PHAT algorithm. However, the correlation function contains only 9 points of interest, while a frequency-domain implementation of GCC-PHAT will necessarily compute 2N correlation points at once. As will be explained in Section 3 and 4, the required number of operations can be significantly reduced by using a time-domain correlation implementation if only a limited number of correlations points are required.

3. TIME-DOMAIN BASED GCC-PHAT IMPLEMENTATION

The inefficiency of the frequency-domain

processing for small arrays brings us at a time-domain version of GCC where only the middle correlation points are calculated. This will indeed strongly reduce the complexity compared to traditional frequency-domain GCC-PHAT implementations with large block sizes. Using two input blocks (left and right or L and R) of length N, the cross correlation can efficiently be calculated as:

$$R_{L,R}[m] = \begin{cases} \sum_{n=0}^{N-1-m} L[n+m]\, R[n], & 0 \le m \le N \\ \sum_{n=0}^{N-1+m} L[n]\, R[n-m], & -N \le m \le 0 \\ 0, & Else \end{cases}$$

It is noted that the number of multiplications to compute a single correlation coefficient is linear with the block size N. As explained in Section 2, only a limited number of cross correlation samples are required to be calculated.

As explained in Section 2, a weighting scheme such as PHAT is required to make the GCC robust against echo paths and reverberation. PHAT alters the correlation function such that it has unity gain in the frequency-domain while the phase information is preserved. A unity gain can be relatively easily achieved by whitening the input signals before computing the cross correlation. In this paper we propose an adaptive linear prediction (LP) filter to whiten the input signals. Still keeping in mind that computational complexity is an issue, a stochastic gradient descent update formula [6] is proposed, which requires about 2K operations per filter update, with K the filter order. With $L_{in}[n]$ the left microphone signal, $\mathbf{W}[n]$ the momentarily adaptive filter coefficients, ρ the adaptive step size, $\mathbf{L}[n]$ the delay line of the filter and $L_{out}[n]$ the enhanced (whitened) left microphone signal, the adaptive LP filter can be implemented as follows:

$$\boldsymbol{L}[n] = \begin{bmatrix} L_{in}[n-1] \\ \dots \\ L_{in}[n-1-K] \end{bmatrix}$$

$$L_{out}[n] = L_{in}[n] - \boldsymbol{L}[n]^T \boldsymbol{W}[n]$$

$$\boldsymbol{W}[n+1] = \boldsymbol{W}[n] + \rho\, L_{out}[n]\, \boldsymbol{L}[n]$$

The preservation of phase information required for TDOA estimation can be guaranteed by not altering the relative phases between both input channels. This can be achieved by only updating one filter (for one channel) and using a copy of this filter on the second channel. Since the spectra of both input channels are almost equal in small-size microphone arrays, the channel using the copied filter is also partially whitened in practice.

		calls	cycles	total Cycles
f-dom	fft	2	11796	23592
	inverse-fft	1	11801	11801
	complex scaling	256	8079	2068224
	total			2143905
t-dom	Pre-whitening	1	6721	6721
	correlation sub	1	7749	7749
	total			14615

Table 1: Clock cycles for time and frequency-domain implementation on an TI C5515 DSP

Finally after the correlation a quadratic interpolation is used to increase the resolution of the TDOA estimate.

Remark that:

- While the frequency-domain implementation is bounded by block sizes of 2^x to be efficient (radix-2 FFT's), the time-domain implementation is not.
- The time-domain implementation leaves room for a balance between complexity and quality of results by altering the order of the LP filter. This cannot be done in the frequency-domain implementation.
- A downside of the time-domain implementation is that there are 2 extra parameters which need to be tuned, namely: LP filter order K and step size ρ. Where an increasing order K can improve whitening results at expense of an increased complexity. Step size ρ can be used to balance convergence speed and converged filter precision.

4. COMPUTATIONAL COSTS

The frequency-domain implementation will need two (zero-padded) FFTs, one inverse FFT and scaling for each frequency bin of the correlation. An (inverse) FFT (radix2) is known to have a complexity of O(2N.logN). Furthermore, each scaling for PHAT weighting requires a square root (to calculate the complex value's absolute value)

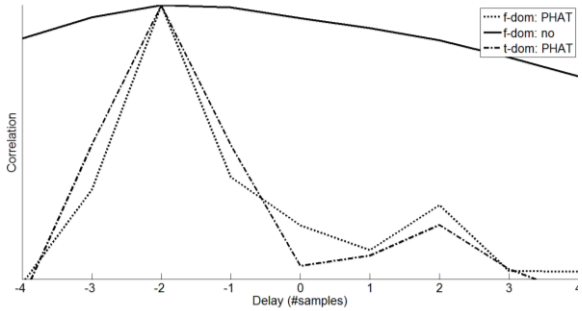

Figure 1: Averaged correlation (real angle: -45°)

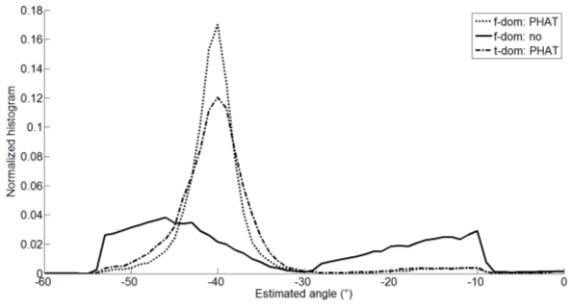

Figure 2: Histogram of estimated angles (real angle: -45°)

and a division. These are often expensive to implement on an embedded device in terms of clock cycles, but it heavily depends on the implementation (for instance, a square root is quite expensive to calculate, but more easily roughly estimated). In the time-domain implementation, all operations are linear in N(for each correlation point), but the expensive square roots and divisions are avoided.

To thoroughly compare required complexity, both frequency and time-domain implementations were implemented on an TI C5515 DSP. The frequency-domain implementation contains (inverse) FFT operations taken from the efficient TI DSPlib, the square root operation taken from the math toolbox and all of them are compiled by the TI v4.3.6 code generation tool. A block of data with size N=256 was processed by both algorithms. For the time-domain implementation a LP filter order of 10 was used. A microphone distance of 6.6cm and sample frequency of 16kHz were assumed (these are also used for the real-environment experiments explained later on), so that only 9 correlation samples were to be calculated. The required clock cycles for both algorithms were measured by means of simulation and are shown in Table 1.

When comparing the total number of clock cycles, the time-domain implementation is performed 146 times faster. However, this should be put into perspective due to the fact that the computation complexity is almost fully dominated by the scaling, more specifically in the calculation of the square root from the math toolbox. This can be performed faster by less accurate approximations, but obviously at a cost of a less accurate location estimation. Nevertheless, without considering the scaling operation, the time-domain implementation still outperforms the efficient (invers) FFTs by a factor 2.4, which is thereby the lower bound of the complexity reduction. Therefore, the time-domain implementation can perform 2.4 to 146 times faster than the frequency-domain implementation.

		Average abs. error (°)	Average estimated angle (°)	Average abs. dev. (°)
-45°	f-dom: no	20,53	-26,75	23,93
	f-dom: PHAT	8,53	-36,89	14,54
	t-dom: PHAT	8,92	-36,91	15,19
0°	f-dom: no	7,04	0,65	11,54
	f-dom: PHAT	3,18	0,20	8,40
	t-dom: PHAT	3,33	0,51	8,26
45°	f-dom: no	14,01	34,00	17,77
	f-dom: PHAT	10,54	35,60	19,42
	t-dom: PHAT	10,27	37,19	18,28
-45°	f-dom: no	24,20	-22,56	28,17
	f-dom: PHAT	11,49	-34,02	19,28
	t-dom: PHAT	12,79	-32,88	20,39
0°	f-dom: no	10,23	1,36	16,98
	f-dom: PHAT	7,24	-0,24	15,65
	t-dom: PHAT	7,75	0,76	16,05
45°	f-dom: no	13,21	34,91	17,90
	f-dom: PHAT	9,70	36,66	19,19
	t-dom: PHAT	10,77	37,15	18,94

Table 2: Results for all datasets

5. COMPARISON OF RESULTS

Six sets of data were recorded, using an Andrea Electronics microphone array at a talker distance of about 1.5m. The array consisted of 2 microphones separated 6.6cm from each other which were sampled at a frequency of 16kHz. Note that these parameters equal those from the example in Section 4, where it was seen that the time-domain implementation performs 2.4 to 146 times faster. The sets were recorded in a real home environment. Half a minute of speech recorded at 3 different incident angles, being -45°, 0° and 45°. Al further results are produced using MATLAB implementations which ease the analysis process.

For the frequency implementation an accurate square root was used so that (by results of Section 4) the time-domain implementation should perform 146 times faster.

For the time-domain implementation, all data sets were processed by the proposed adaptive LP filter with order 10. Then multiple blocks of 256 samples (16ms) were cut out of this datasets. Each block that passed a predetermined energy threshold of $5e^{-5}$ were further processed. Other blocks were considered as only noise. For each dataset there were between 240490 and 319470 blocks of data used. Furthermore, an appropriate adaption step size ρ of 128 was used.

First, it was investigated if the cross correlation peaks indeed became smaller in case PHAT was used. Results of the averaged correlation function over all blocks for the first dataset are shown in Figure 1, results for all other datasets are similar. It is clearly seen that both the frequency and time-domain implementations produce huge improvements compared to the correlation without PHAT. Furthermore it is noticed that the correlation peak using frequency-domain PHAT is still somewhat smaller than the time-domain implementation. It can be noticed that the time-domain results should be better when using larger LP filter orders.

In a second test these correlation functions were used to estimate angles. From all these angles normalized histograms were calculated and are shown in Figure 2 (only results of the first dataset are shown, the others are again similar). These can be viewed as probability functions describing the probability of an estimated angle given a random pair of input blocks. It can be seen that both GCC-PHAT estimations have a maximum around 40°, whereas the method without PHAT has no clear peak in the histogram. The 5° error can be explained by the low initial resolution, and the fact that quadratic interpolation is observed to be quite limited in improving the resolution. Detailed results are shown in Table 2, listing the average absolute error, the average estimated angle and the average absolute deviation for both PHAT methods and for the method without any weighing. Once again, there is no significant difference between results obtained by the time and frequency-domain PHAT algorithms. If PHAT is not used, the performance is rather poor.

6. CONCLUSIONS

It has been explained that, when using small microphone arrays, the frequency-domain GCC-PHAT is suboptimal in terms of computational complexity. Therefore, a time-domain GCC-PHAT equivalent has been proposed. Computational complexities have been compared in terms of required clock cycles on a Texas Instruments C5515 DSP. Here it has been demonstrated that a computational complexity gain of a factor of 2.4 to 146 can be achieved by using the time-domain implementation while the angle of arrival estimations are comparable to those obtained using the frequency-domain implementation.

ACKNOWLEDGMENTS

This research was carried out at the ESAT and MOBILAB laboratories of respectively the Katholieke Universiteit Leuven and Katholieke Hogeschool Kempen, in the frame of following projects: IWT doctoral scholarship nr.111433 (‘Audio based home observation system for the elderly’), IWT-SBO/100049 (‘ALADIN’), KU Leuven Research Council CoE EF/05/006 ‘Optimization in Engineering’ (OPTEC) and PFV/10/002 (OPTEC), Concerted Research Action GOA-MaNet, the Belgian Programme on Interuniversity Attraction Poles initiated by the Belgian Federal Science Policy Office IUAP P6/04 (DYSCO, ‘Dynamical systems, control and optimization’, 2007-2011), Research Project IBBT, and Research Project FWO nr. G.0763.12 ('Wireless acoustic sensor networks for extended auditory communication’). The work of A. Bertrand was supported by a Postdoctoral Fellowship of the Research Foundation - Flanders (FWO).

REFERENCES

[1] A. Bertrand, M. Moonen, "Distributed node specific LCMV beamforming in wireless sensor networks," *Transactions on Signal Processing*, vol. 60, pp. 233-246, Jan. 2012.

[2] I. Tashev, *Sound Capture and Processing*. John Wiley and Sons, 2009.

[3] R. Schmidt, "Multiple emitter location and signal parameter estimation," IEEE *Transactions on Antennas and propagation*, vol. 34(3), pp. 276 280, 1986.

[4] C. Knapp, G. Carter, "The generalized correlation method for estimation of time delay," IEEE *Transactions on Acoustics Speech and Signal Processing*, vol. 24(4), pp. 320-327, 1976.

[5] A. Bertrand, "Applications and trends in wireless acoustic sensor networks: a signal processing perspective," in *Proc. of the IEEE Symposium on Communications and Vehicular Technology (SCVT)*, Ghent, Nov. 2011.

[6] S. Haykin, *Adaptive Filter Theory*. Prentice Hall, 2001.