


2017-12-18

Tiled Time Delay Estimation in Mobile Cloud Computing Environments

Ruairí de Fréin

Technological University Dublin, ruairi.defrein@tudublin.ie

Follow this and additional works at: <https://arrow.tudublin.ie/engscheleart>

 Part of the [Computer Engineering Commons](#), [Signal Processing Commons](#), and the [Systems and Communications Commons](#)

Recommended Citation

de Fréin,Ruairí (2017)Tiled Time Delay Estimation in Mobile Cloud Computing Environments,2017 *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*,December 18-20, 2017. Bilbao, Spain. year={2017}, volume={}, number={}, pages={1-6}, month={Dec.}, keywords={Interference, Mobile communication, Noise cancellation, Transforms, Time-frequency analysis, Mobile handsets, Urban areas}, ISBN={978-1-5386-3840-8}, doi={To Appear}, url={To Appear} }

This Conference Paper is brought to you for free and open access by the School of Electrical and Electronic Engineering at ARROW@TU Dublin. It has been accepted for inclusion in Conference papers by an authorized administrator of ARROW@TU Dublin. For more information, please contact arrow.admin@tudublin.ie, aisling.coyne@tudublin.ie, vera.kilshaw@tudublin.ie.

Funder: Science Foundation Ireland



Tiled Time Delay Estimation in Mobile Cloud Computing Environments

Ruairí de Fréin[†]

[†]Dublin Institute of Technology,
Institiúid Teicneolaíochta Bhaile Átha Cliath,
Ireland

web: <https://robustandscalable.wordpress.com>

in: 2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT). See also `BIBTEX` entry below.

`BIBTEX`:

```
@article{rdefrein16Tiled,  
author={Ruairí de Fréin†},  
journal={2017 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)},  
title={Tiled Time Delay Estimation in Mobile Cloud Computing Environments},  
year={2017},  
volume={},  
number={},  
pages={1--6},  
month={Dec.},  
keywords={Interference, Mobile communication, Noise cancellation,  
Transforms, Time-frequency analysis, Mobile handsets, Urban areas},  
ISBN={978-1-5386-3840-8},  
doi={10.1109/ISSPIT.2017.8388656},  
url={https://ieeexplore.ieee.org/document/8388656},  
}
```

© 2017 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.



Tiled Time Delay Estimation in Mobile Cloud Computing Environments

Ruairí de Fréin

School of Electrical and Electronic Engineering
 Dublin Institute of Technology
 Dublin, Ireland
 Email: rdefrein@gmail.com

Abstract—We present a tiled delay estimation technique in the context of Mobile Cloud Computing (MCC) environments. We examine its accuracy in the presence of multiple sources for (1) sub-sample delays and also (2) in the presence of phase-wrap around. Phase wrap-around is prevalent in MCC because the separation of acoustic sources may be large. We show that tiling a histogram of instantaneous phase estimates can improve delay estimates when phase-wrap around is significantly present and also when multiple sources are present. We report that error in the delay estimator is generally less than 5% of a sample, when the true delay is up to 10 samples for three source mixtures.

I. INTRODUCTION

Time delay estimation between signals captured at spatially separated sensors has many important applications in radar, acoustic source separation [1], [2] and synchronized analysis [3]. Fifth generation mobile networks promise a higher density of mobile broadband users, support for device-to-device communication and greater data throughput [4]. Future advances in Mobile Cloud Computing (MCC) will be underpinned by this throughput increase: increased interaction with cloud data-centers may be used to provide prior information for on-device applications [5]. Fog computing architectures also facilitate greater device-to-device cooperation for edge analytics [6], by moving learning functionality closer to the end-user. This has the benefit of reducing latency and increasing reliability [7]. The disadvantage of this higher density of users is that greater acoustic interference will be experienced by mobile users. This interference is a crucially important consideration when the mobile is performing acoustic scene analysis. Time delay estimation is a first step in many acoustic denoising and source separation algorithms [8], [9]. The time delay estimation problem that arises in MCC comes in a different form compared to its traditional embodiment, this is because side-information is more easily available via cloud-servers/databases, networked sensors and peer-to-peer communications technologies, such as WIFIDirect [10]. However phase-wrap effects are more prevalent due to the larger distances between sensors.

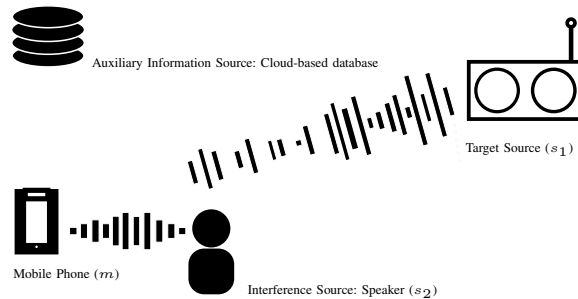


Fig. 1. Motivational Scenario: a music source is playing concurrently with the acoustic signal of a speaker who is communicating using a mobile phone. A cloud database, either recognizes the music track, or receives a (filtered) recording from an environmental sensor which is physically located close to the source. The database supplies the auxiliary input for the delay estimator on the phone $a[n] = s_1[n]$, e.g. the target source. The mobile phone observes the mixture $m[n] = \alpha_1 s_1[n + d_1] + \alpha_2 s_2[n - d_2]$.

Problem Statement: The present submission addresses the problem of delay estimation for a discrete time target source, $s_1[n]$, in a monaural discrete time mixture signal, $m[n]$, captured by a mobile phone.

$$m[n] = \sum_{j=1}^{N_s} \alpha_j s_j[n - d_j]. \quad (1)$$

The discrete time index is n . The attenuations and delays experienced by each source in the mixture are α_j and d_j respectively. Signals are scaled and delayed as they impinge on the mobile phone, relative to the versions of these signals observed at an auxiliary sensor. For the target source s_1 , the attenuation and delay are $\{\alpha_1, d_1\}$. The mobile phone senses N_s sources of acoustic energy. The problem we address is how to estimate d_1 , the delay experienced by s_1 .

In MCC environments we envisage that prior information could be made available by a number of mechanisms: (1) a peer mobile device, for example a cluster leader device [10], or an networked environmental sensor which provides a good monaural estimate of the signal, where by good we mean low-noise; and (2) a data-centre provides a clean

recording, given that the signal is hosted in a database, for example a commercially available music track or a public service announcement. In both cases the peer mobile device, the environmental sensor or the data-center are the auxiliary sensor. The estimate of the signal received by the mobile phone source will be attenuated and delayed relative to the version of the signal which is observed by the auxiliary sensor. The observation of the target source made by the auxiliary sensor may be corrupted due to other sources which are close to the auxiliary sensor. In case one, the auxiliary sensor may observe

$$a[n] = s_1[n] + z[n], \quad (2)$$

where the signal z captures a version of the signals s_j for $j = 2 \dots N_s$, or some other signal, y , which is not captured by the mobile phone. We will assume that the power of z is small relative to s_1 in order for the information provided by this sensor to be used for delay estimation. In other words, we only consider auxiliary sensors if they give a relatively clean version of the target signal. In the case illustrated in Fig. 1, the auxiliary sensor is a cloud hosted database and the target signal is a song played by a sound system. It provides $a[n] = s_1[n]$ exactly.

Recent acoustic source location and direction finding results [1], [8] underpin advances in acoustic scene interpretation applications on mobile handsets. They may not however be suitable for MCC type applications as they place strict constraints on the maximum displacement of the two sensors used for delay estimation in order to combat phase wrap-around effects. Similarly, cross-correlation based methods may not give sub-sample delay estimates [11] unless interpolation [12] or related methods are used [13]. In this paper we propose a delay estimator which is robust to phase wrap-around effects using a procedure called *Tiled Phase Estimation* (TPE) which addresses the short-comings of power-weighted type estimators [1], [8]. We address the problem of estimating sub-sample delays, by exploiting the linear phase assumption, that is, we fit linear template functions to time-frequency phase estimates.

This paper is organized as follows. Section II introduces a simple delay estimation algorithm. We explain that it is not suitable for many MCC scenarios. Section III outlines how a simple voting procedure can be used which makes sub-sample delay estimation possible. We then introduce a tiling procedure to tackle the phase-wrap around and the sub-sample delay estimation limitation of previous approaches in Section IV. In Section V we evaluate our approach under different environmental mixing conditions and discuss how this type of approach may be deployed.

II. DELAY ESTIMATION

We introduce a novel Time-Frequency (TF) delay estimator. A TF linear transform of $s_1[n]$ positioned at sample

iR , is

$$s_{i,k}^1 = \sum_{n=iR}^{N-1+iR} s_1[n]w[n-iR]W^{k(n-iR)}. \quad (3)$$

The number of window hop-size samples is R , $w[n] \in \mathbb{R}^N$ is the analysis window function and $W = e^{-j\frac{2\pi}{N}}$. By multiplying $s_{i,k}^1$ by W^{kd} , accurate delay estimation can be achieved in TF. Care must be taken with the choice of analysis window [3]. The complex conjugate of $s_{i,k}^1$ is denoted $\bar{s}_{i,k}^1$.

Definition: Instantaneous TF phase estimates are achieved by computing the angle of the product of the observations at two sensors

$$\phi_{i,k} = \arg(m_{i,k}\bar{a}_{i,k}) = \arctan\left(\frac{l}{r}\right) \quad (4)$$

where $m_{i,k}$ and $a_{i,k}$ are m and a in TF respectively and the real and imaginary components of the product $m_{i,k}\bar{a}_{i,k}$ are $r = \text{Re}(m_{i,k}\bar{a}_{i,k})$ and $l = \text{Im}(m_{i,k}\bar{a}_{i,k})$. In general, instantaneous phases estimates in this unprocessed form are insufficient to provide accurate delay estimates.

Rationale: Instantaneous TF phase estimates, $\phi_{i,k}$, yield good estimates of d_1 if $a[n] = s_1[n]$ and $m[n] = s_1[n-d_1]$. In words, if one source is observed on both sensors. Then, $\arg(m_{i,k}\bar{a}_{i,k})$ reduces to $\arg(s_{i,k}^1 W^{kd_1} \bar{s}_{i,k}^1) = -\frac{2\pi}{N}kd_1$. The single source constraint is restrictive for real-world applications. It turns out that in many other cases the instantaneous TF phase estimator (Eqn. 4) gives good estimates, if a little more work is done. For example, consider the case where $m[n] = \sum_{j=1}^{N_s} \alpha_j s_j[n-d_j]$, $a[n] = s_1[n]$ and the source s_1 dominates in a set of TF bins, Ω . Then, the relationship $\arg(s_{i,k}^1 W^{kd_1} \bar{s}_{i,k}^1) \approx -\frac{2\pi}{N}kd_1$ holds for the set of TF bins Ω . The challenge lies in detecting which TF bins are in the set Ω . The source s_1 is dominant in a TF bin if its power is greater than the power of all of the other sources s_j for $j = 2 \dots N_s$. The assumption that speech is compactly supported in TF underpins many recent advances in sparse Signal Processing [1], [14]. In summary, the presence of other sources s_j for $j \neq 1$ causes certain TF bins to be dominated by those other sources, thus, the phase estimated in certain TF bins need not be due to the delay d_1 experienced by s_1 , in all frequency bins, k . Our approach to overcome this challenge is to use a voting procedure to vote for the best linear function through all of the instantaneous phase estimates $\phi_{i,k}$; we use a set of feasible lines to connect together the TF phase estimates corresponding to one source.

III. VOTING FOR THE BEST PHASE ESTIMATE

To set up the voting procedure, first we quantize the instantaneous phase estimates $\phi_{i,k}$ using L uniformly spaced quantization levels. Each level is spaced by Δ_d . We denote

the quantization levels $\hat{\phi}$. For a given frequency k , the set of TF bins contributing to quantization bin $\hat{\phi}$ is denoted

$$I_k(\hat{\phi}) := \{\{i, k\} : |\phi_{i,k} - \hat{\phi}| < \Delta_d\}. \quad (5)$$

A histogram that records the number of instantaneous phase estimates within Δ_d of the histogram bin value $\hat{\phi}$, for each discrete frequency k , is generated by computing the cardinality of each of these sets

$$H(\hat{\phi}, k) := |I_k(\hat{\phi})|. \quad (6)$$

We fit lines through the non-zero bins of the histogram $H(\hat{\phi}, k) > T$, where $T \in \mathbb{Z}$ is a user defined threshold. The most simple choice of T is $T = 1$, which means that only activated histogram bins can cast a vote. Introducing a threshold which can take an integer value greater than 1 allows the user to be more selective. Each of the test lines is parameterized by $\{r, \theta\}$, where r is the distance from the origin to the closest point on the line and θ is the angle between the x-axis and the line connecting the origin with that closest point, yielding the line

$$r_e = \hat{\phi} \cos \theta + k \sin \theta. \quad (7)$$

The set of test line angles used is $\nabla_\theta = 0, \Delta_\theta, \dots, \pi$. The set of test line distances from the origin is ∇_r . The step-size Δ_r and an upper and lower bound r_{min} and r_{max} determine the members of the set ∇_r . The distance bin index is denoted r . We use the following approach to select the maximum and minimum distances

$$r_{max} = \ell_2([L, N]^T), \quad r_{min} = -r_{max}, \quad (8)$$

where $\ell_2(\cdot)$ computes the Euclidean norm of a vector containing the height and breadth bin-count of the histogram.

We cast a vote as follows. For each non-zero entry of $H(\hat{\phi}, k) > T$ we compute a value r_e for each angle in the set ∇_θ . We determine the *distance* bin, r , associated with each value r_e for a given angle

$$I_\theta(r) := \{\{r, \phi\} : |r_e - r| < \Delta_r \text{ for a given } \theta\} \quad (9)$$

A distance-angle histogram is constructed by computing the cardinality of

$$L(r, \theta) = |I_\theta(r)|. \quad (10)$$

The time delay estimate of d_1 is obtained by considering the location of the peak of $L(r, \theta)$, e.g. $\{r_{max}, \theta_{max}\}$, and computing

$$d_{est} = -\frac{N}{L} \tan \theta_{max}. \quad (11)$$

IV. TILING PHASE HISTOGRAMS FOR DELAY ESTIMATION

One short-coming of the approach above is that we may potentially observe a number of peaks which have the same height in $L(r, \theta)$. Consider the three-pixel image in the

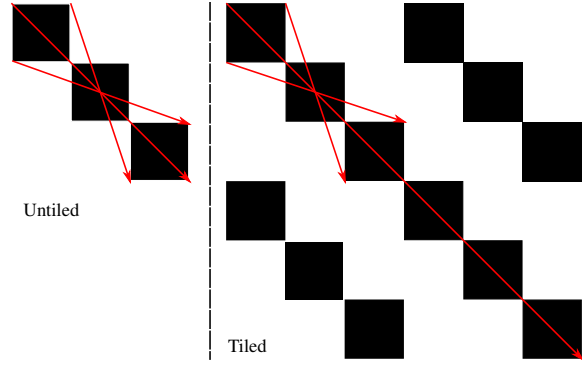


Fig. 2. Tiling Effects: The LHS (of dashed line) figure illustrates three black (non-zero) pixels in a 9×9 image. A number of different template lines (red arrows) pass through all three pixels. If we replicate the base image on the LHS (cf. RHS of dashed line) only one line passes through all 6 pixels, where the first 3 pixels correspond to the base histogram and the second 3 correspond to a copy of the base histogram.

LHS of Fig. 2, $H(\hat{\phi}, k)$. The three lines or votes (red-arrows) illustrated intersect all three pixels in $H(\hat{\phi}, k)$. This leads to a histogram $L(r, \theta)$ with three peaks of equivalent height. Given that the slope of line corresponding to the highest peak gives the delay estimate, which line should we choose?

Multiple peaks with the same height arise for the following reasons: (1) The number of un-corrupted instantaneous phase estimates, is related to the dominance of the target source, and therefore the number of sources present in the mixture. Target source dominance plays a crucial role in having a clear-cut outcome from the voting procedure. Interference sources may corrupt the phase estimates so that a number of lines provide feasible explanations for the instantaneous phase estimates. (2) The angle step-size parameter allows the user to obtain fine-scale delay estimates. Choosing smaller step-sizes causes the likelihood of a number of peaks with the same height in $L(r, \theta)$ occurring to increase. If we were to refine the angle step-size Δ_θ , it is probable that more lines would intersect all three pixels than the three lines illustrated in Fig. 2. In effect better resolution is not achieved by reducing the step-size Δ_θ . (3) With regard to the phase wrap-around problem, phase wrap around causes the instantaneous phase estimates, which are a linear function of the discrete frequency bin, k , to be wrapped around in some cases. The larger the delay, the more often the line is wrapped around. The slope of each line is unchanged, but the distance from the origin of each line (which arises due to phase wrap-around) is different for different portions of what should be one line.

We propose the following approach for improving the resolution of delay estimation and resolving the challenges raised above. A tiled histogram is obtained by tiling replicates of the base histogram $H(\hat{\phi}, k)$ in the following

manner:

$$T(H(\hat{\phi}, k), K) = \begin{bmatrix} H(\hat{\phi}, k) & \dots & H(\hat{\phi}, k) \\ \vdots & \ddots & \vdots \\ H(\hat{\phi}, k) & \dots & H(\hat{\phi}, k) \end{bmatrix} \quad (12)$$

where the number of additional rows and columns in $T(H(\hat{\phi}, k), K)$ are $K - 1$. We apply the voting procedure defined above to this tiled histogram $T(H(\hat{\phi}, k), K)$ as before.

The RHS of the dashed line in Fig. 2 gives an illustration of the effect of tiling for $K = 2$. The base histogram, $H(\hat{\phi}, k)$, which gave rise to three peaks of equal in height in $L(r, \theta)$ now has one clear peak as only one of the lines intersects all six of the pixels in $T(H(\hat{\phi}, k), K)$. Similarly, multiple lines which arise due to phase wrap-around affects are unwrapped by tiling the base histogram $H(\hat{\phi}, k)$. The additional tiles provide a continuation of each of the wrapped-around phase estimates.

In summary, the reason that performing delay estimation using the untiled histogram gives peaks that are so broad, or that there are so many peaks of equal height, is that Δ_θ causes there to be a number of lines that fit through all of the pixels. The way to deal with this problem is to increase the number of pixels that the lines have to go through. We achieve this by replicating the base histogram. As the linear ridge is the phase histogram gets longer the number of peaks with equal heights gets smaller. We keep on adding tiles until we end up with only one peak. It is then possible to refine the delay estimate by decreasing Δ_θ and running the technique again with the appropriate number of tiles.

We present a synthetic example of the tiling process. Fig. 3 illustrates a synthetic phase histogram where the frequency index increases from the top to the bottom, giving rise to a line along the diagonal. One pixel, which is missing from the diagonal, represents a corrupted phase estimate due to an interference source.

We plot the angle-distance histogram $L(r, \theta)$ for the untiled phase histogram in the upper LHS in Fig. 4. In the upper RHS figure in Fig. 4 we plot the rows of the figure on the LHS. At least two peaks have flap-tops which are approximately 10 samples wide. This means that the estimator generates approximately 20 equally likely delay estimates. In the lower figure in Fig. 4 we illustrate the effect of tiling the base histogram before we perform the voting procedure. The lower RHS figure illustrates that the voting procedure produces a clearer outcome if more base histograms are analyzed using the tiling operator described above. The reason for this is that the length of the longest sequence of pixels in a line is increased. In the next section we demonstrate that this tiling operation also improves speech mixtures which have multiple interference sources

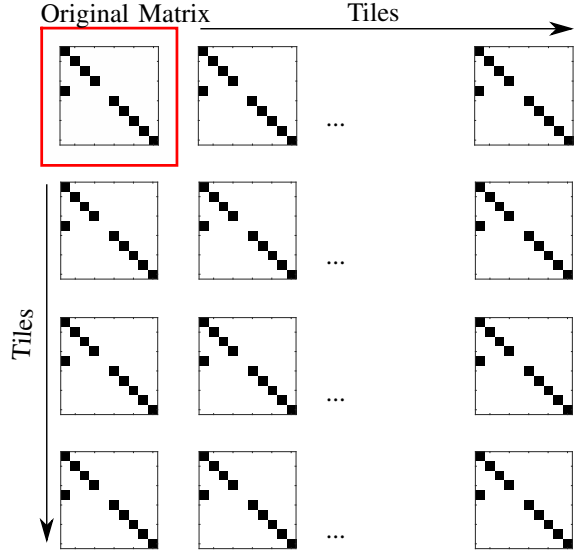


Fig. 3. Tiling Process: the base histogram $H(\hat{\phi}, k)$ is indicated by the bounding box in the upper LHS corner. Multiple copies of $H(\hat{\phi}, k)$ are concatenated in order to improve delay estimation resolution.

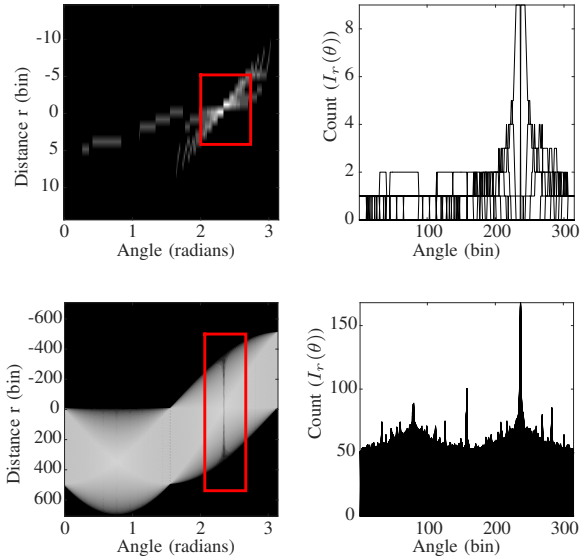


Fig. 4. Utility of Tiling: The LHS column of figures illustrate phase histograms for one tile (upper) and fifty tiles (lower). The plots in the RHS column plot the number of elements in each bin $L_r(\theta)$. The count in the untiled plot (upper RHS) is equal in a number of bins, which implies that many delay estimates are feasible. In the tiled case (lower RHS) there is one peak, and thus one feasible delay estimate. Boxes indicated the corresponding peaks in the LHS column.

and where the target source is delayed by an arbitrary delay.

V. NUMERICAL EVALUATION

Time delay estimation is evaluated in a number of different scenarios to demonstrate (1) the range of its performance, e.g. delay in samples; (2) the effectiveness of tiling the base histogram; (3) that time delay estimates are accurate in the presence of interference sources; and finally (4) that we can improve the resolution of the estimator by tuning the angle step-size parameter.

We use the following parameters in our experiments. All mixtures are sampled a 16kHz. The window length used to perform TF analysis is $N = 1024$ samples. The analysis window is advanced by $N/2$ samples when we compute the TF representation. The number of bins used in the phase histogram is $L = 100$. The angle step-size used in the first experiments is fixed as $\Delta_\theta = 0.01$.

First we evaluate the accuracy of the estimator over a range of delays. The first mixture consists of three speech sources (two female voices and one male speaker). With regard to the mixing scenario illustrated in Section I a database provides the input from the auxiliary sensor. It provides a clean version of the target signal, which we use to estimate the relative delay parameter d_1 . The mobile phone provides a mixture of three signals. Source s_1 is a female speaker reciting a public service announcement perhaps and sources s_2 and s_3 are female and male speakers respectively. The mixtures are

$$m[n] = \alpha_1 s_1[n - d_1] + \alpha_2 s_2[n - d_2] + \alpha_3 s_3[n - d_3] \quad (13)$$

and

$$a[n] = s_1[n]. \quad (14)$$

We assume that the knowledge of the auxiliary source input was provided by a cloud-based song-recognition system. Each source is normalized to sum to one and the weights of each component sources in the mixture are $\alpha_1 = \alpha_2 = \alpha_3 = 1$.

We analyze the performance of the estimator over the range of delays $-10 \leq d_t \leq 10$ samples, where d_t corresponds to the true delay. For each mixture one speaker is the target speaker and two speakers are interference speakers. We increase the number of tiles in the delay estimator over the range $K = 1, 2, 3$ in order to demonstrate the efficacy of the approach. In each experiment phase wrap around effects are present in the instantaneous phase histograms that are computed, $H(\hat{\phi}, r)$.

Fig. 5 illustrates the error in delay estimates ($d_{true} - d_{est}$ in samples) as the absolute value of the true delay $|d_{true}|$ increases. In all experiments phase-wrap around does not significantly affect the estimated time delay estimates. The estimated delay is d_{est} . Increasing the number of tiles used in the estimator from $K = 1$ to $K = 2$ tiles increases the accuracy of the delay estimates. Increasing the number of tiles from $K = 2$ to $K = 3$ tiles has no effect on

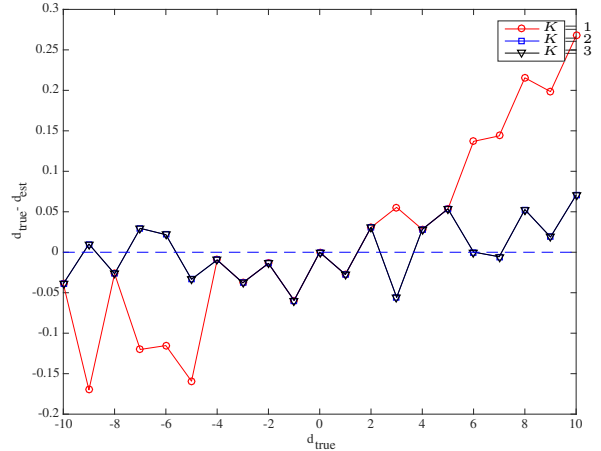


Fig. 5. Effect of tiling: Increasing the number of tiles from one to two tiles increases the accuracy of the delay estimates. Increasing the number of tiles from two to three tiles has no effect on accuracy. The error in the estimates is generally less than 5% of one sample.

accuracy. This suggests that unless Δ_θ is decreased the time delay estimator will not yield improved estimates. Alternatively, it is possible that the two other interference signals corrupt instantaneous phase estimates to such an extent that improvement in the time delay estimates is not possible. For all but one estimate the error in the time delay estimate is less than 5% of a sample when the number of tiles is equal to $K = 2$ or $K = 3$. This performance is note-worthy given that it is achieved when the delay is $d_{true} = 10$ samples.

In the second case we evaluate the effect of varying both the angle step-size parameter, using the values $\Delta_\theta = 0.0005, 0.001, 0.005$, and the number of tiles, using the values $K = 1, 2, 3$, to see if the performance of the estimator can be improved. We focus on positive delays $0 \leq d_{true} \leq 10$ because of the symmetry of the results through the x-axis. A number of results illustrated in Fig. 6 warrant reporting.

The accuracy of the time delay estimates improve as the number of tiles increases. When the angle step-size is $\Delta_\theta = 0.005$ increasing the number of tiles from $K = 1$ to $K = 2$ yields an improvement. However, increasing the number of tiles from $K = 2$ to $K = 3$ yields no improvement. In contrast for a smaller angle step-size $\Delta_\theta = 0.0005$ increasing the number of tiles from $K = 1$ to $K = 2$ and then to $K = 3$ improves the time delay estimates. This result supports the assertion that the resolution of time delay estimates may be improved by increasing the number of tiles and also by decreasing the angle step-size parameter in tandem.

Discussion: The scenarios we described above arise in many real-world applications. Some of the harmful effects of community noise on communications are outlined by the

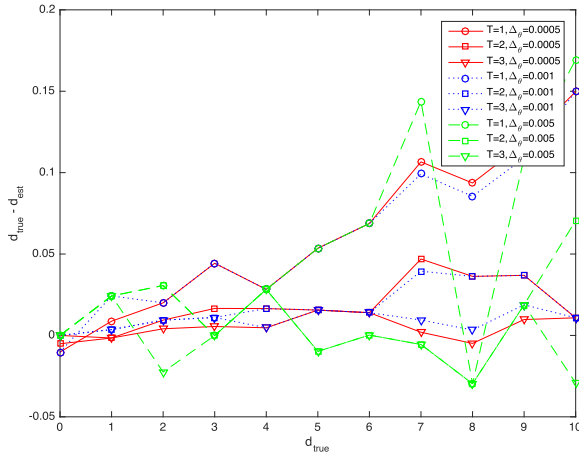


Fig. 6. Effect of tiling K and the angle step-size $\Delta\theta$: The resolution of time delay estimates may be improved by increasing the number of tiles and decreasing the step-size parameter.

World Health Organization in [15]. We have considered the case where the target source was a recording of a well-known piece of spoken text or song. A database such as Spotify may be used as the auxiliary sensor to estimate the time delay experienced by this source on the mobile phone’s recording. On the other hand, an environmental sensor in an underground commuter train system might capture a clean recording of the train arriving in the underground station. This recording may then be used by the target mobile device to obtain a time delay estimate of when the train arrives, or a platform announcement, relative to the mobile phone. Moreover, peer mobile devices which are closer to some source of interference signal may contribute a corrupted single channel recording of the interference signal [16]. In each of these scenarios the ability to be able to estimate time delays is important. This submission is underpinned by the fact that voice based search is becoming more popular in mobile search [17]. For example, Google reported that 20% of Android-based searches were made by voice in 2016 –Amazon Echo and Siri are also gaining increasing acceptance in the home as voice assistants.

VI. CONCLUSION

We have presented a tiled time delay estimator which computes accurate sub-sample time delay estimates in challenging scenarios where a number of interference signals are present. We demonstrated that increasing the number of tiles used, along with decreasing the angle step-size parameter $\Delta\theta$, improved time delay estimates in many cases. We also demonstrated that phase wrap-around does not significantly affect the estimator. This result is notable given the increased delays that we might expect in Mobile Cloud Computing environments. Numerical evaluation of

the technique supports the claim that the accuracy of the estimator is tun-able. We posit that the results of this study will be useful for the design and optimization of time delay estimators on Mobile Cloud Computing platforms, which are tasked with performing acoustic scene analysis.

REFERENCES

- [1] O. Yilmaz and S. T. Rickard, “Blind separation of speech mixtures via time-frequency masking,” *IEEE Trans. on Sig. Proc.*, vol. 52, no. 7, pp. 1830–47, Jul. 2004.
- [2] J. Chen, J. Benesty, and Y. Huang, “Time delay estimation in room acoustic environments: An overview,” *EURASIP Journal on Advances in Signal Processing*, no. 1, May 2006.
- [3] R. de Fréin and S. T. Rickard, “The synchronized short-time-Fourier-transform: Properties and definitions for multichannel source separation,” *IEEE Trans. Sig. Proc.*, vol. 59, no. 1, pp. 91–103, Jan. 2011.
- [4] A. Gohil, H. Modi, and S. K. Patel, “5G technology of mobile communication: A survey,” in *2013 Int. Conf. Intell. Sys. and Sig. Proc.*, Mar. 2013, pp. 288–92.
- [5] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, “Heterogeneity in mobile cloud computing: Taxonomy and open challenges,” *IEEE Coms. Surv. Tut.*, vol. 16, no. 1, pp. 369–92, 2014.
- [6] P. Patel, M. I. Ali, and A. Sheth, “On Using the Intelligent Edge for IoT Analytics,” *IEEE Intelligent Systems*, vol. 32, no. 5, pp. 64–69, Sep. 2017.
- [7] S. Yang, “IoT Stream Processing and Analytics in the Fog,” *IEEE Communications Magazine*, vol. 55, no. 8, pp. 21–27, 2017.
- [8] R. de Fréin and S. T. Rickard, “Power weighted divergences for relative attenuation and delay estimation,” *IEEE Sig. Proc. Letters*, vol. 11, no. 4, May 2016.
- [9] S. T. Rickard, C. Fearon, R. Balan, and J. Rosca, “Minuet: Musical interference unmixing estimation technique,” *Conf. on Inf. Sc. and Sys.*, pp. 1–6, 2004.
- [10] R. Loomba, R. de Fréin, and B. Jennings, “Selecting energy efficient cluster-head trajectories for collaborative mobile sensing,” in *2015 IEEE GLOBECOM*, Dec. 2015, pp. 1–7.
- [11] C. Knapp and G. Carter, “The generalized correlation method for estimation of time delay,” *IEEE Trans. Acoust., Sp., Sig. Proc.*, vol. 24, no. 4, pp. 320–7, Aug. 1976.
- [12] X. Lai and H. Torp, “Interpolation methods for time-delay estimation using cross-correlation method for blood velocity measurement,” *IEEE Trans. Ultrason Ferroelect. Freq. Contr.*, vol. 46, pp. 277–90, 1999.
- [13] B. Qin, H. Zhang, Q. Fu, and Y. Yan, “Subsample time delay estimation via improved GCC PHAT algorithm,” in *2008 9th Int. Conf. on Sig. Proc.*, Oct. 2008, pp. 2579–82.
- [14] R. de Fréin, “Ghostbusters: A parts-based NMF algorithm,” *24th IET Irish Signals and Systems Conference*, pp. 1–8, Jun. 2013.
- [15] B. Birgitta and L. Thomas, “Community noise,” *Stockholm: Archives of the Centre for Sensory Research*, vol. 2, 1995.
- [16] R. de Fréin, “Learning and storing the parts of objects: IMF,” *IEEE International Workshop on Machine Learning for Signal Processing (MLSP)*, pp. 1–6, Sep. 2014.
- [17] J. Schalkwyk, D. Beeferman, F. Beaufays, B. Byrne, C. Chelba, M. Cohen, M. Kamvar, and B. Strope, ““Your word is my command”: Google search by voice: A case study,” in *Advances in Speech Recognition: Mobile Environments, Call Centers and Clinics*, A. Neustein, Ed. New York: Springer, 2010, ch. 4, pp. 61–90.

Acknowledgment: This publication has emanated from research conducted with the financial support of Science Foundation Ireland (SFI) under the Grant Number 15/SIRG/3459