GCC-PHAT Re-Imagined - A U-Net Filter for Audio TDOA Peak-Selection

Gulin, Jens; Åström, Kalle

Published in:
ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

DOI:
10.1109/ICASSP48485.2024.10447558

2024

Document Version:
Peer reviewed version (aka post-print)

GCC-PHAT RE-IMAGINED - A U-NET FILTER FOR AUDIO TDOA PEAK-SELECTION

Jens Gulin[‡⋆] *Kalle Åström*[†]

jens.gulin@sony.com karl.astrom@math.lth.se
[†] Centre for Mathematical Sciences, Lund University, Lund, Sweden
[‡] Department of Electrical and Information Technology, Lund University, Lund, Sweden
[⋆] Sony Europe B.V., Lund, Sweden

ABSTRACT

Time-difference-of-arrival (TDOA) estimation from GCC-PHAT is not always as straight forward as finding the maximum peak. This work views the GCC output as an image, with time on the vertical axis and TDOA horizontally, to explore if image-to-image machine learning methods can make a more robust filter. The Structure from Sound Database provides audio recorded with a distributed microphone setup and a moving sound source. The audio was fed to GCC-PHAT without pre-processing, and images were produced for batch processing. The ground truth, the direct-path TDOA, shows a continuous curve through time. The GCC output image has a similar curve, but obscured by noise and not at all times texturally different from the multi-path components. The main approach tested is binary semantic segmentation with a U-Net. A challenge is the extreme class imbalance within the image. Preliminary results indicate that the method is valid to detect curves, yet more work is needed to single out the direct path TDOA with confidence.

***Index Terms*—** Time-difference-of-arrival, Generalized Cross-Correlation, U-Net, noise reduction, curve detection

1. INTRODUCTION

The scope of this paper is *image-to-image filtering of GCC-PHAT output*, for the purpose of improved TDOA estimations. In a larger context, the goal is to produce stable methods for moving sound source localization (SSL) when the microphone positions are unknown. A secondary objective is to learn from little data.

Precise localization of sender/receiver node positions using radio or sound signals is a key enabler in numerous applications such as microphone array calibration, speaker diarization, beam-forming, radio antenna array calibration, mapping and positioning [1]. A key component of systems for calibration of node positions, e.g. [2], is the first signal processing step. The state-of-the-art method is the GCC-PHAT method and variants thereof, [3]. It has been observed in previous papers that the potential for improvement is large here. For example in [4], the authors studied a dataset of seven recordings where state of the art on average were within the set threshold in less than 40% of the estimations.

In this paper we study if machine learning methods could be used to filter the result from the GCC-PHAT detections, to obtain fewer false positives while keeping the true positives sufficiently high. We show how these filtering methods can be trained, and evaluate the resulting algorithms on real world data to ascertain the validity of the approach.

2. BACKGROUND

2.1. GCC-PHAT

The *Generalized Cross-Correlation with Phase Transform* (GCC-PHAT) [5] is still after almost 50 years one of the most useful algorithms for TDOA estimation. However, in many cases conditions are not optimal. To get the appropriate results, both input and output may need further processing based on heuristics for that specific use-case.

The **cross-correlation** of a pair of signals is highest when they are the most similar, time-shifting one of the signals thus gives a peak corresponding to the time difference. Correlation corresponds to multiplication in the frequency domain. Effectively, when the signal pair has matching (amplitude of each) frequency component in the Fourier transform, the correlation is high. GCC allows an arbitrary factor to adjust the importance of each frequency band. This factor is what makes it generalized. The **PHAT factor** performs *whitening*, to make each frequency count equally regardless of amplitude. Compared to unadjusted GCC, the GCC-PHAT gives very distinct peaks. One drawback is a risk of giving even weak noise high impact in the output.

As the GCC output is correlation estimates for a set of proposed time-differences, a TDOA window is chosen to limit the relevant range. In the optimal case, the true TDOA is found from identifying the highest peak, the TDOA that gave the maximum correlation value. The optimal condition states that any noise in the signal is uncorrelated, otherwise a pair of signals will have matching noise. Unless a single sound source, in an un-echoic environment, additional peaks is to be expected. In a real-world scenario, the peak corresponding

to the highest GCC-score could be noise, and further possible peaks may need to be evaluated to get accurate TDOA.

Usually GCC is computed using a form of short-time Fourier transform (STFT) [6], separating the output into a sequence of GCC estimates, each specific to a time-window. With a moving sound source, a short window length is required to get accurate time resolution. Unfortunately, high time resolution means less detailed frequency resolution.

2.2. Semantic segmentation using U-Net

Many methods to process images for various effects are available. Most state of the art methods are learning based, implemented with convolutional neural networks [7] and more recently using transformers [8]. Here we adapt *semantic segmentation*, which is the task of separating known object types from the background. It can be considered a classification task that evaluate each pixel. The U-Net [9] was introduced in year 2015 as a means to do biomedical image segmentation. It is a fully convolutional network, which makes it possible to use on larger patches, yet with reasonable training requirements. It can be seen as a U-shaped encoder-decoder, the bottleneck at the bottom, with added skip-connections across (Figure 1a) to retain both high-level and detailed context. Lower levels have smaller feature maps, a quarter of the size above ($\frac{W}{2} \times \frac{W}{2}$ versus $W \times W$). To retain the needed expressiveness when the size of the feature map is reduced, the U-Net doubles the number of maps in each layer.

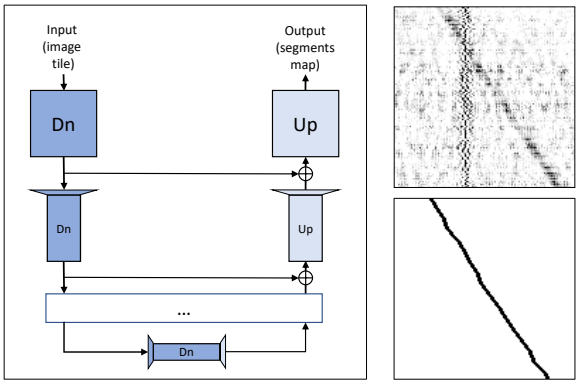


Fig. 1. **(a)** U-Net architecture overview. See section 2.2 for description. **(b)** Example input (128x128 patch). **(c)** Ground truth matching input.

The boxes in Figure 1a are fully convolutional blocks, each comprising of two chained 3×3 convolutions with padding to retain size[1]. The *Dn* blocks increase the number of channels and pairs with down-sampling, while the *Up* blocks reduce channels with up-sampling. As shown, the output from a down-block is also fed unaltered through the skip-connection. This set of feature maps is concatenated to the feature maps in the decoder path (effectively stacked to make the double number of maps).

An important consideration is the **loss function**. [10] gives an overview of popular choices for semantic segmentation. For example, focal losses are applied to prioritize learning from more difficult examples.

In this case, the loss is a measure of the difference in two binary images, G (ground truth) and P (prediction), where each pixel should be either 1 (positive) or 0 (negative). Where P and G intersect, there is a True Positive (TP), where they mismatch it is either a False Positive (FP) or a False Negative (FN).

The *Dice coefficient* (or F_1-score) is a positive metric for pixel-wise similarity of images. It is the "intersection over sum" score, similar to the "intersection over union" (IoU, Jaccard index). The *Tversky index*[2] [12]

$$T_{\alpha,\beta}(G,P) = \frac{\text{TP}}{\text{TP} + \alpha\text{FN} + \beta\text{FP}} \tag{1}$$

is a generalization that introduces constants to balance the penalty of FP and FN.

With Equation 1, IoU is $T_{1,1}$ and Dice is $T_{0.5,0.5}$. Enforcing $\alpha = 1 - \beta$ is common, to limit number of hyperparameters. An important problem for segmentation, as with most ML methods, is class imbalance: When there are only few foreground pixels in a large image patch, it is hard for the loss function to guide the optimization in a meaningful direction. Collapsing into all-zero predictions may get 99% of the (background) pixels right, yet be utterly useless. Setting a higher α is a way to promote more positive predictions, with the hope that some will eventually lead to TP catching on. [11] suggests $\alpha = 0.7$ and notes that higher values pushes for more FP, leading to good *sensitivity* but worse overall score.

From a positive metric, such as Tversky index, the loss function is essentially $1 - T_\alpha(G,P)$, with added smoothing in T to ensure continuity. *Focal Tversky Loss* is $(1 - T_\alpha(G,P))^\gamma$ [10]. A $\gamma > 1$ gives less weight to mispredictions in easy training examples, to focus training towards the more difficult examples (low Tversky index) with a steepened gradient.

3. GCC-PHAT RE-IMAGINED

This section briefly describes the work presented. Full details are available in the implementation, with high-level design as well as low-level hyperparameters.

3.1. Implementation

The experiment is driven by a Python 3 implementation, code available[3]. The code was developed iteratively and ad-hoc, as

[1] No padding was used in the original U-Net, but same size is more convenient and the more popular choice since.

[2] The original paper regards similarity of sets. FP and FN (thus α and β) are interchangeable until one set is regarded as GT, current notation following [10, 11].

[3] https://github.com/JensGulin/GCC-REIMAGINED-UNET

a proof-of-concept. There are three main parts.

GCC-conversion ran on a local Jupyter environment on Intel Xeon W-2135 CPU @ 3.70GHz × 12 with 32 Gb RAM and an Nvidia GTX 1080Ti GPU with 11 Gb memory. This notebook does the audio processing and outputs the proper 2D image representation. The GCC-PHAT implementation used is from [13][4].

Training and inference was developed in the Google Colab environment, executing on a T4 GPU with 16 Gb memory. The U-Net implementation, in TensorFlow/Keras, was based on a tutorial[5].

Benchmarking was performed on the same environment as the training.

3.2. Audio processing

The audio is fed to GCC-PHAT in windows of size 1024 samples. This is a basic short-time Fourier transform (STFT) that provides time-resolution. The step size is 128 samples, giving 8 overlapping windows. The TDOA range is cut off at ± 800 samples delay. These arbitrary numbers can be adjusted with knowledge of the nature of the audio.

No pre-filtering of the audio is done, the idea of this experiment is to push the proposed filtering in the image domain to the limits.

3.3. 2D representation

The GCC output is in a (TDOA $\times$ time) array that translates directly to an image of (width $\times$ height). Each row is one GCC estimate, with each cell value a correlation score, a negative or positive number.

To ensure values in range [0,1], all negative values are zeroed, and each row is divided by the maximum of that row. The reason for dropping the negative values, rather than rescaling, is to focus training towards positive peaks. It has not been evaluated if this is helpful or not. A nice side-effect is that printed example images have more contrast[6].

The data is saved to file using *pyplot.imsave()*, which will rescale the [0,1] range to [0,max]. The file format chosen is PNG, a gray-scale image represented as 8 bpc RGB. One image is produced per microphone pair, organized in folders to separate different recording sessions.

Following image-to-image learning practice, the ground truth must be an image of the same dimensions. For semantic segmentation, each pixel in GT is given a number corresponding to the intended class. As a binary classification task, the true direct-path TDOA and the background are the two classes. For each window, a single pixel is thus set to 1 in a row otherwise 0. To provide a single TDOA per row, given GT samples are averaged within the time window. This also bridges the different sampling rates of GT and audio. Again, one GT image is produced per microphone pair.

3.4. U-Net design

The U-Net architecture is designed for the semantic segmentation learning task. After some experimentation, an input size of 1024x1024 pixels is chosen. The aim is to get as much context as possible in each patch, while not exhausting GPU memory.

Following the design in Figure 1, each layer down-sample with a 2x2 kernel MaxPool, effectively passing on a feature map with half the side of the layer input. The white box illustrates stacking layers as the one above. Starting with 1024x1024, 6 middle layers reach 16x16, then downsampled in the bottle-neck to 8x8. This is the most abstract, high-level representation of the patch. The decode-chain upsample at matching levels, to end again with a 1024x1024 segmentation mask after a regular sigmoid activation function.

For this design, a maximum of 512 maps per layer is set. This cap is not in the usual architecture, but added to keep memory usage down. The reasoning is that this restriction of the lowest layers is in line with the bottleneck strategy anyway. Due to the large input size and the many layers, the network has in total around 31.5 million parameters to learn.

3.5. Loss function and training

For this experiment, the *bassh3* recording data from SFSDB[7] is used. This data has a moving speaker playing continuous music for 15 seconds, with a 9 second intro and 10 second outro without music. There are 8 microphones synchronously recorded, in total 28 possible pairs. Arbitrarily, mic 6 is kept as test data, and mic 2 for validation data. This leaves 15 pairs for training. Since microphone pair (2,6) was never seen in training, this is the pair used for final evaluation.

During training, random patches are taken from the large image. Each batch uses a new selection. To help guide training, the patches are chosen so that there are no fully negative examples, at least some pixels are ground truth.

Although the decision was to have 1024x1024 input, to verify the model and setup, smaller patches were used to start with. The initial training experiments failed promptly due to class imbalance. Tversky loss with $\alpha = 0.7$ did not give a consistent result. The larger patches increase the problem, since maximum one pixel per row is ground truth, thus a quadratic increase of background pixels.

Training is executed for 25 epochs, using the Adam optimizer [14] with initial learning rate 0.0001 and 10-fold decay every 10 epochs. The current loss function is the average of two Focal Tversky loss functions, one with $\gamma = 3$ and one with $\gamma = 0.3$, both with $alpha = 0.9$. The focus on both difficult and easy examples resembles [15].

[4] https://github.com/axeber01/ngcc

[5] https://github.com/nikhilroxtomar/UNet-Segmentation-in-Keras-TensorFlow

[6] Example images were inverted, so that positive peaks print as black.

[7] https://vision.maths.lth.se/sfsdb/

4. EXPERIMENTAL VALIDATION

To validate the approach, the trained U-Net is applied on the evaluation data, the *bassh3_mic_2_6* input image. The large file is processed as 1024x1024 patches with 512 pixel stride. This sums four estimations of each pixel, to alleviate differences due to context. Each estimate is [0,1], with 0.5 assumed the threshold for predicting class 1 (positive, direct path TDOA). The sum range [0,4] changes the threshold, 0.5 meaning "likely at least one predicted positive". For this evaluation, no threshold is set, instead peaks are extracted as the most likely TDOA candidates.

For convenience, the output is again formatted as an RGB image, where the summed prediction is the green component, GT is red and input is blue.

4.1. Qualitative evaluation

From visual inspection, it is clear that the U-Net is able to suppress the image background noise. It is also apparent that there are still plenty of false positives. Although the GT indicated only one curve, the direct path, the current method predicts any curve-segment. The prominent noise line at 0 TDOA is not removed. Current model prioritize positive predictions, even when this straight line is clearly different, both in texture and from context (see Figure 1b).

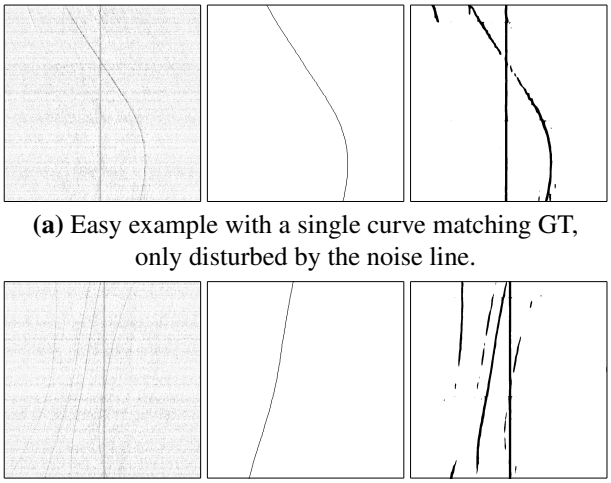
A few illustrative examples are shown for reference in Figure 2. **Example (a)** illustrates a successful segmentation. The output has a strong peak matching the GT. The output also includes the noise line, and in this case it even gets priority at the crossing. **Example (b)** is hard even for a human. There are four similar curves and the one slightly more accentuated is not the actual target. This shows that texture (pixel value and adjacent pixel patterns) is not enough to distinguish all cases. Another indicator is context, how strong the curve is over time. Apparently the context of this patch is not enough: A human will agree with the GT only when given even more context (bigger patch). The "false curves" (echoes) are prominent in *this* cut-out, but the true TDOA is actually more consistent over time.

A third example, not depicted, confirms the ability to filter out noise also during intro and outro, when no music is playing. The prediction is then dominated by the noise line.

5. CONCLUSIONS & FUTURE WORK

This paper evaluates semantic segmentation with U-Net as a method to filter out false peaks from GCC-PHAT output. This novel approach can successfully suppress background noise and pick out curves and lines from the image.

Further research will be able to continue from this baseline. Evaluation in a downstream task is needed to show actual accuracy improvement and if added computational cost can be fully regained by less false peaks.



(a) Easy example with a single curve matching GT, only disturbed by the noise line.

(b) Hard to see that GT is second curve from left. Prediction gives more confidence to the third.

Fig. 2. Example cut-outs (1024x1024) from bassh3_mic_2_6. Images from left to right: Input, Ground Truth, U-Net Output.

The model currently map both the direct-path and sufficiently strong multi-path echoes as semantically same, even though GT did only mark direct-path. This is likely due to the limited training (relatively few samples and few epochs) and a loss function promoting positive predictions over false negatives. In this first experiment, it was not regarded as a problem that there were more than one peak remaining, as a robust multilateration method can handle some putative entries. Finding reflections may also be useful [16, 17]. Traditional image manipulation methods, such as noise suppressing and erosion, might be used to improve results, both before and after U-Net processing.

Several design options are open for future work; training strategy, loss function and network design are such examples. A preliminary conclusion is that a larger time-context would benefit differentiation of direct-path and multi-path curves. The *Distance Map Loss* [18] is under consideration, as a way to promote the slim GT. Another suggestion may be *Sharp U-Net* [19], that introduce fixed sharpening filters on the skip connections, to reduce "not only blurred feature maps, but also over- and under-segmented target regions".

6. ACKNOWLEDGMENTS

This work was partially supported by the strategic research project ELLIIT, and by the Wallenberg AI, Autonomous Systems and Software Program (WASP) funded by the Knut and Alice Wallenberg Foundation. The authors gratefully acknowledge Lund University Humanities Lab.

For support and discussions, we thank Amir Aminifar and Erik Tegler. Gratitude is further extended towards the entire open source community. In memoriam of Jan Eric Larsson.

7. REFERENCES

[1] A. Plinge, F. Jacob, R. Haeb-Umbach, and G. A. Fink, "Acoustic microphone geometry calibration: An overview and experimental evaluation of state-of-the-art algorithms," *IEEE Signal Processing Magazine*, vol. 33, no. 4, pp. 14–29, 2016.

[2] Simayijiang Zhayida, Simon Segerblom Rex, Yubin Kuang, Fredrik Andersson, and Kalle Åström, "An automatic system for acoustic microphone geometry calibration based on minimal solvers," *arXiv preprint arXiv:1610.02392*, 2016.

[3] M. Larsson, G. Flood, M. Oskarsson, and K. Åström, "Robust self-calibration of constant offset time-difference-of-arrival," in *International conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[4] Kalle Åström, Martin Larsson, Gabrielle Flood, and Magnus Oskarsson, "Extension of time-difference-of-arrival self calibration solutions using robust multilateration," in *2021 29th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 870–874.

[5] C. Knapp and G. Carter, "The generalized correlation method for estimation of time delay," *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 24, no. 4, pp. 320 – 327, aug 1976.

[6] J. Allen, "Short term spectral analysis, synthesis, and modification by discrete fourier transform," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 25, no. 3, pp. 235–238, 1977.

[7] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[8] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[9] O. Ronneberger, P.Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*. 2015, vol. 9351 of *LNCS*, pp. 234–241, Springer, (available on arXiv:1505.04597 [cs.CV]).

[10] Shruti Jadon, "A survey of loss functions for semantic segmentation," in *2020 IEEE Conference on Computational Intelligence in Bioinformatics and Computational Biology (CIBCB)*. oct 2020, IEEE.

[11] Nabila Abraham and Naimul Mefraz Khan, "A novel focal tversky loss function with improved attention u-net for lesion segmentation," *CoRR*, vol. abs/1810.07842, 2018.

[12] Amos Tversky, "Features of similarity," *Psychological Review*, vol. 84, pp. 327–352, 1977.

[13] Axel Berg, Mark O'Connor, Kalle Åström, and Magnus Oskarsson, "Extending GCC-PHAT using Shift Equivariant Neural Networks," in *Proc. Interspeech 2022*, 2022, pp. 1791–1795.

[14] Diederik P. Kingma and Jimmy Ba, "Adam: A method for stochastic optimization," 2017.

[15] Ken C. L. Wong, Mehdi Moradi, Hui Tang, and Tanveer Syeda-Mahmood, *3D Segmentation with Exponential Logarithmic Loss for Highly Unbalanced Object Sizes*, p. 612–619, Springer International Publishing, 2018.

[16] Inkyu An, Myungbae Son, Dinesh Manocha, and Sung-Eui Yoon, "Reflection-aware sound source localization," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 66–73.

[17] Inkyu An, Youngsun Kwon, and Sung-eui Yoon, "Diffraction- and reflection-aware multiple sound source localization," *IEEE Transactions on Robotics*, vol. 38, no. 3, pp. 1925–1944, 2022.

[18] Francesco Caliva, Claudia Iriondo, Alejandro Morales Martinez, Sharmila Majumdar, and Valentina Pedoia, "Distance map loss penalty term for semantic segmentation," 2019.

[19] Hasib Zunair and A. Ben Hamza, "Sharp U-Net: Depthwise convolutional network for biomedical image segmentation," *Computers in Biology and Medicine*, vol. 136, pp. 104699, 2021.