

A semblance based TDOA algorithm for sound source localization

Aldeia, G. S. I., Crispim, A. E., Barreto, G., Alves, K., Ferreira, H., Nose-Filho, K.

Guilherme Seidyo Imai Aldeia

Universidade Federal do ABC

Petrópolis/RJ
2019

Índice

① Introdução

② Semblance

③ Algoritmo proposto

④ Metodologia

⑤ Resultados

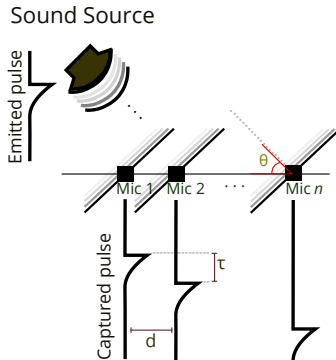
⑥ Conclusões

Contextualização

O uso de drones em cenários de busca e resgate está ganhando interesse na pesquisa. Podemos fazer o uso de microfones para localizar pessoas pedindo por ajuda, porém:

- ✗ A fonte sonora precisa estar ativa;
- ✗ O *ego-noise* (ruído produzido pelo próprio drone) tem grande interferência.

Problema de localização da fonte sonora

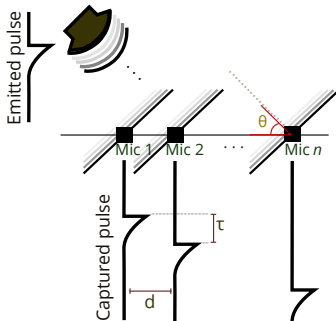


Temos:

- Um *array* de microfones com posições conhecidas (podemos medir d);
- Conhecimento da velocidade de propagação do som (v).

Problema de localização da fonte sonora

Sound Source

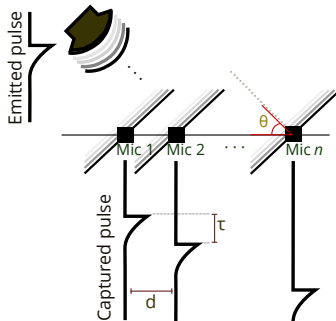


Assumimos que:

- O som se propaga com uma frente de onda plana;
- O som chega nos diferentes microfones em diferentes instantes de tempo.

Problema de localização da fonte sonora

Sound Source



Sejam:

- k_d a direção da fonte do som em relação à um ponto de referência (e.g. o primeiro microfone), parametrizado por θ (ou, no caso tridimensional, pelo par (θ, ϕ));
- τ uma função de correção dos tempos de atraso relativo em cada canal.

Função de correção τ

O atraso τ para o microfone de índice i e localização m_i , assumindo que a fonte do som esteja em k_d , é dado por:

$$\tau(m_i) = -\frac{k_d \cdot m_i}{v}, \quad (1)$$

onde k_d é um vetor que aponta para uma direção parametrizada por azimuth ($\theta_d \in [-\pi, \pi]$) e elevação ($\phi_d \in [-\frac{\pi}{2}, \frac{\pi}{2}]$).

Pela equação:

- Conhecendo d, τ, v podemos achar θ ;
- Conhecendo d, θ, v podemos achar τ .

Mas inicialmente não conhecemos nem θ nem τ .

Objetivos

- Gerais:
 - Propor um algoritmo para localização de fonte sonora utilizando sinais de microfones.
- Específicos:
 - Utilizar uma abordagem baseada no domínio do tempo;
 - Analisar a interferência do *ego-noise* no algoritmo proposto;
 - Comparar com um algoritmo do estado-da-arte.

Índice

① Introdução

② **Semblance**

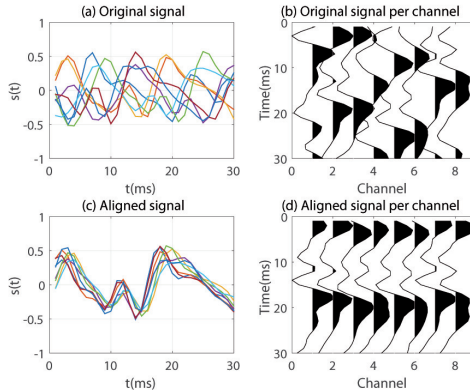
③ Algoritmo proposto

④ Metodologia

⑤ Resultados

⑥ Conclusões

Corrigindo os atrasos dos sinais no tempo



A correção calculada para a direção correta da fonte sonora gera o (provável) melhor alinhamento dos sinais.

Semblance

A função de correlação cruzada *Semblance* (utilizada em física de reflexão) mede o **nível de similaridade** entre os sinais de diferentes sensores.

$$Z_d = \frac{\sum_n |\sum_k \hat{s}_k(n)|^2}{N_r \sum_n \sum_k |\hat{s}_k(n)|^2}, \quad (2)$$

onde k é o índice do microfone, n as amostragens no tempo, N_r o número de sensores, e $\hat{s}_k(n) = s_k(n - \tau_k)$ o sinal na amostragem no tempo n do k -ésimo microfone após ter a correção τ_k (para um dado Θ e Φ) aplicada.

Índice

- 1 Introdução
- 2 Semblance
- 3 Algoritmo proposto**
- 4 Metodologia
- 5 Resultados
- 6 Conclusões

Estratégia para solucionar o problema

Não temos informações sobre k_d ou τ , mas:

- Sabemos como calcular o atraso τ para uma direção k_d ;
- Temos uma função de coerência robusta para quantificar a similaridade dos sinais;

É esperado que o melhor alinhamento esteja relacionado com a correção aplicada para a direção correta da fonte sonora

Global Semblance

Nossa proposta inicial foi o algoritmo **Global Semblance**.

Algoritmo proposto - Global Semblance

- ❶ Criar uma grade de “chutes” uniformemente espaçados (pares (θ, ϕ));
- ❷ Calcular o atraso para cada possível direção;
- ❸ Medir a correlação nos sinais corrigidos;
- ❹ Escolher direção com maior correlação.

Global Semblance

Algoritmo 1: Find semblance global (find_global)

input : Δ : interval between angles to be tested

SoS : speed of sound on the medium

Fs : sampling rate

s : matrix containing the audio of the 8-channel microphones

$micPos$: array with coordinates $[x, y, z]$ of the microphones positions

output: z : matrix mapping correlation with angles

Θ : tested values for elevation

Φ : tested values for azimuth

$step = \Delta * \pi / 180$;

$\Theta = [\theta \mid \theta \leftarrow [-\pi, -\pi + step, \dots, \pi]]$;

$\Phi = [\phi \mid \phi \leftarrow [-\pi/2, -\pi/2 + step, \dots, \pi/2]]$;

$\tau = []$;

for (i, θ) **in** $(range(\Theta), \Theta)$ **do**

for (j, ϕ) **in** $(range(\Phi), \Phi)$ **do**

$kd = [\cos(\theta) * \cos(\phi), \sin(\theta) * \cos(\phi), \sin(\phi)]$;

for (k, mic) **in** $(range(micPos), micPos)$ **do**

$\tau[i, j, k] = round((kd * mic') / SoS) * Fs$;

for i **in** $range(\Theta)$ **do**

for j **in** $range(\Phi)$ **do**

for k **in** $range(numMic)$ **do**

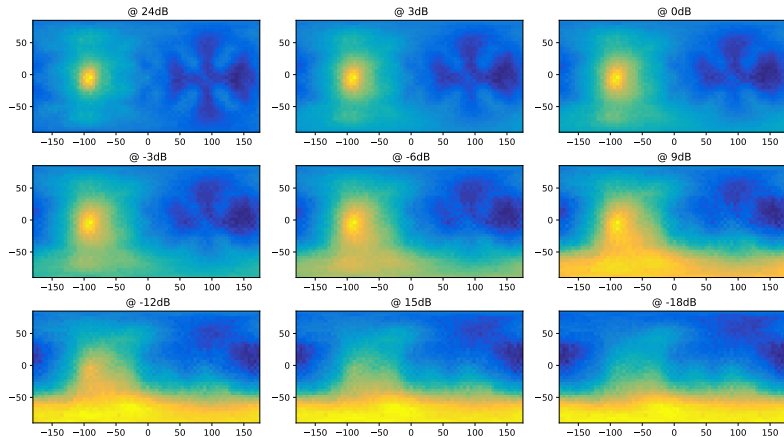
$\hat{s}_k(n) = s_k(n - \tau[i, j, k])$

$z[j, i] = semblance(\hat{s})$

return z, Θ, Φ ;

Painel de Semblance

Painel de Semblance: *Heatmap* obtido calculando a correlação Semblance para todas as combinações de (θ, ϕ) :



Local Semblance

Uma variação estudada em testes preliminares, chamada **Local Semblance**, se mostrou mais eficiente.

Variação - Local Semblance

- ➊ Dividir o áudio em vários *frames* (podendo ter sobreposição ou não);
 - ➋ Aplicar o Global Semblance para cada *frame*;
 - ➌ Combinar os painéis em um único utilizando os maiores valores (utilizando o *pooling max*);
 - ➍ Escolher direção de maior correlação.
- ✓ Capaz de realçar a relação SNR nos *frames* onde a fonte sonora está ativa, obtendo melhores resultados.

Local Semblance

Algoritmo 2: Find semblance local (find_local)

input : *frameSize*: size of the frames

overlap: overlap between frames

Δ : interval between angles to be tested

SoS: speed of sound on the medium

Fs: sampling rate

s: the data of the 8-channel

micPos: array with coordinates [x, y, z] of the microphones positions

output : *z*: matrix mapping correlation with angles

Θ : tested values for elevation

Φ : tested values for azimuth

sTotal = *length*(*x*); // total samples

sSize = *round*(*frameSize* * *Fs*); // sample size

sOverlap = *round*(*overlap* * *sSize*); // sample overlap

nFrames = *ceil*((*sTotal* - *sSize*)/(*sSize* - *sOverlap*)) + 1;

painels = [];

for *i* *in* *range*(*nFrames*) **do**

begFrame = *i* * (*sSize* - *sOverlap*);

endFrame = *begFrame* + *sSize*;

sFrame = *s*[*begFrame* : *endFrame*, :];

painels[*i*] = *find_global*(Δ , *SoS*, *Fs*, *sFrame*);

return *pooling*(*painels*), Θ , Φ ;

Índice

- ① Introdução
- ② Semblance
- ③ Algoritmo proposto
- ④ Metodologia**
- ⑤ Resultados
- ⑥ Conclusões

Benchmark

Temos:

- 3 gravações de um drone com 8 microfones acoplados, com apenas sinal de fala (com direção conhecida);
- 1 gravação de puro ruído dos rotores do drone, obtida com os mesmos microfones.

Foram feitas diversas combinações com diferentes relações SNR: [24, 21, 18, ..., 3, 0, -3, -4, -5, ..., -19, -20, -21]

Isso permite encontrar um limiar de SNR para dizer onde o algoritmo começa a apresentar problemas.

Métrica de erro

O erro foi calculado pela equação *Great circle distance*:

$$\Delta\sigma = \arctan \frac{\sqrt{(\cos \phi_2 \sin(\Delta\theta))^2 + (\cos \phi_1 \sin \phi_2 - \sin \phi_1 \cos \phi_2 \cos(\Delta\theta))^2}}{\sin \phi_1 \sin \phi_2 + \cos \phi_1 \cos \phi_2 \cos(\Delta\theta)}. \quad (3)$$

Essa equação dá o ângulo entre 2 pontos na superfície de uma esfera. Foi considerado um erro aceitável até 10°.

Ajuste de hiper-parâmetros

Os algoritmos possuem um grupo de hiper-parâmetros que influenciam na qualidade do resultado:

- Em comum aos algoritmos Local Semblance e Global Semblance:
 - Δ - tamanho do espaçamento da grade de chutes;
- No caso do Local Semblance:
 - *Overlap* - sobreposição entre os frames;
 - *FrameSize* - tamanho de cada frame.

Para encontrar a melhor configuração, foi utilizado o *gridsearch*.

Gridsearch

Gridsearch: técnica comum no campo de aprendizado de máquina para ajuste de hiper-parâmetros. Faz uma busca exaustiva entre todas combinações, obtendo aquela que minimiza o erro.

① $\text{overlap} = [0, 0.1, 0.2, 0.3, 0.4, 0.5]$

② $\Delta = [17.5, 15, 12.5, 10, 7.5, 5]$

③ $\text{frameSize} = [0.064, 0.128, 0.256, 0.512, 1.024]$

Comparação com o estado-da-arte

Os resultados foram comparados com o GCC-PHAT (*Generalized Cross Correlation PHase-Transform method*), utilizando dois *poolings* diferentes - *max* e *sum*.

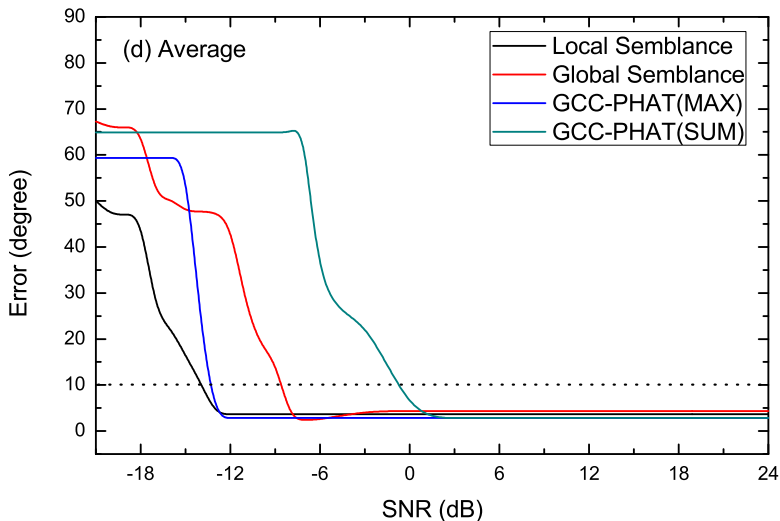
Índice

- ① Introdução
- ② Semblance
- ③ Algoritmo proposto
- ④ Metodologia
- ⑤ Resultados**
- ⑥ Conclusões

Resultados do Gridsearch

- Melhor resultado: $\Delta = 10$, $FrameSize = 0.064s$, 20% *Overlap* entre os *frames*;
 - No geral, valores de Δ menores que 10 não tem diferença significativa no resultado;
 - Para o *FrameSize*, quanto menor o valor, melhor o resultado;
 - Um pequeno grau de *Overlap* (20%) tem melhores resultados.

Desempenho médio dos métodos



Discussão

À respeito dos métodos propostos:

- Apresentaram um erro de $\approx 2^\circ$ em vários casos;
- Local Semblance leva cerca de 6.8 segundos para executar em cada áudio, com duração de 5s;
- Abordagem Global usa 1 *core*, a Local é paralelizada em todos os *cores* de um i7@1.3GHz;
- O Local Semblance é capaz de acertar a direção até em casos de SNR de -16dB.

Discussão

Comparado com o GCC-PHAT:

- GCC-PHAT foi comparado nas mesmas condições: $\Delta=10$ e *frames* de 0.064s;
- o Local Semblance superou o GCC-PHAT;
- O Global Semblance tem um resultado intermediário entre o GCC-PHAT (max pooling) e o GCC-PHAT (sum pooling);

Índice

- ① Introdução
- ② Semblance
- ③ Algoritmo proposto
- ④ Metodologia
- ⑤ Resultados
- ⑥ Conclusões

Conclusões

- Nesse artigo propomos uma nova técnica baseada na função de coerência Semblance para o problema de localização de fonte sonora;
- O algoritmo foi testado com 3 áudios em diferentes configurações de SNR;
- Algoritmo apresenta boa performance.

Continuação da jornada

Um próximo artigo sobre o método está para ser publicado!

*4th Workshop on Communication Networks and Power Systems
(WCNPS 2019)*

**On the application of SEGAN for the attenuation of the
ego-noise in the speech sound source localization
problem**

Conferência dias 3 e 4 de outubro 2019

<https://ieee-wcnps.org/>

Agradecimentos

Os autores agradecem à Universidade Federal do ABC pelo apoio financeiro.

Contato

Guilherme Aldeia (discente e apresentador do trabalho)

✉ `guilherme.aldeia@aluno.ufabc.edu.br`

Kenji Nose-Filho (professor docente orientador):

✉ `kenji.nose@ufabc.edu.br`

Link da apresentação:

🔗 `galdeia.github.io/presentations/SBrT.pdf`