

IA-DATAHACK

Défi Accoustics

Localisation d'un humain dans un salon avec une impulsion sonore

Lilian SCHALL, Théophile STOURBE, Julien SCHAFFAUSER, Simon THUAUD et Elias TEBBANI

Notre méthodologie pour ce hackaton:

- Lecture de l'article de référence pour le jeu de données SoundCam [3]
- Découverte et analyse du jeu de données ainsi que les fichiers de méta-données fournies
- Implémentation et reproduction des différents modèles "*baseline*" conseillés par l'article: séparation du groupe en 2 sous-groupes pour optimiser notre productivité
- Analyse et visualisation des résultats
- Finalisation du rapport (qui a été écrit au fur et à mesure du hackaton, à la fin de chaque journée), et préparation de notre présentation

Contexte

Situation d'enregistrement

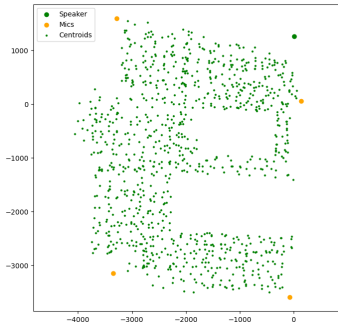


Figure: Visualisation du salon et des différentes positions enregistrées pour l'humain numéro 1. [3]

Contexte

Réponse sonore du salon

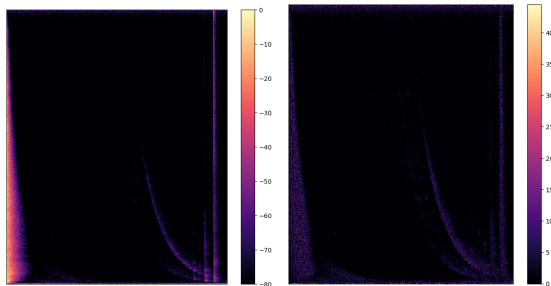


Figure: A gauche, spectrogramme de la *Room impulse response*[1] avec l'humain numero; à droite, la différence de RIR entre un salon vide et un salon avec humain.

Feature extraction

Valeur efficace entière

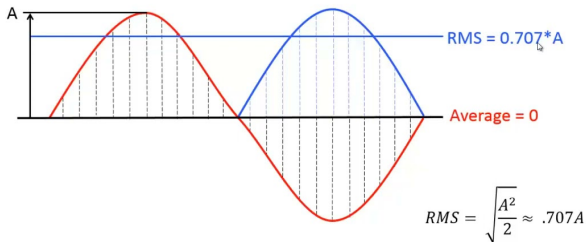


Figure: Visualisation d'un calcul de valeur efficace sur signal entier

source: <https://community.sw.siemens.com/s/article/root-mean-square-rms-and-overall-level>

Feature extraction

Valeur efficace avec fenêtre glissante

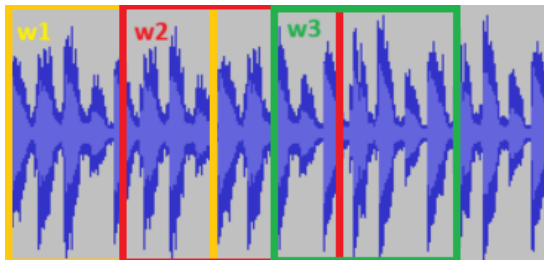


Figure: Visualisation des fenêtres glissantes pour le calcul des valeurs efficaces

Feature extraction

Benchmark

	Time to process	nb feature per mic-signal
RMS full	$\approx 1min30$	1
RMS rolling window	$\approx 20min$	1304

Table: Benchmark des méthodes d'extraction de la valeur efficace sur 4000 prises de son

Entraînement de modèles

Modèle linéaire: distance d'erreur

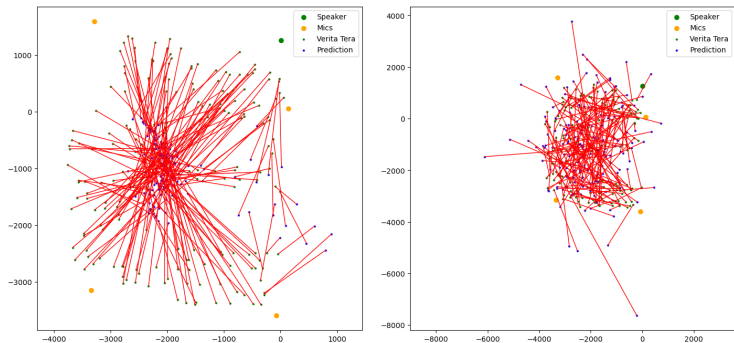


Figure: Distance d'erreur pour le modèle linear: à gauche RMS full, à droite RMS rolling window

Entraînement de modèles

Modèle K-nearest-neighbours[2]: distance d'erreur

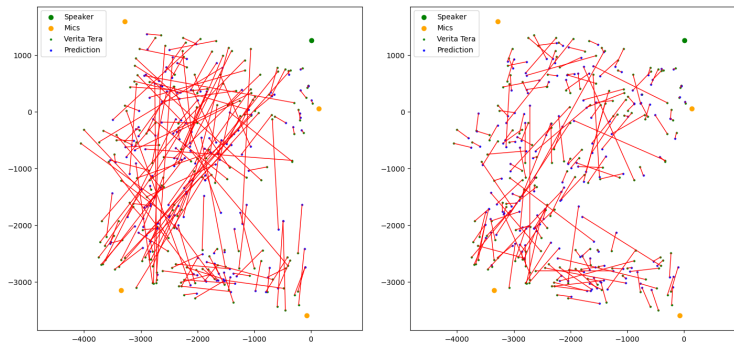


Figure: Distance d'erreur pour le modèle KNN: à gauche RMS full, à droite RMS rolling window

Entraînement de modèles

Modèle linéaire: heatmap d'erreur

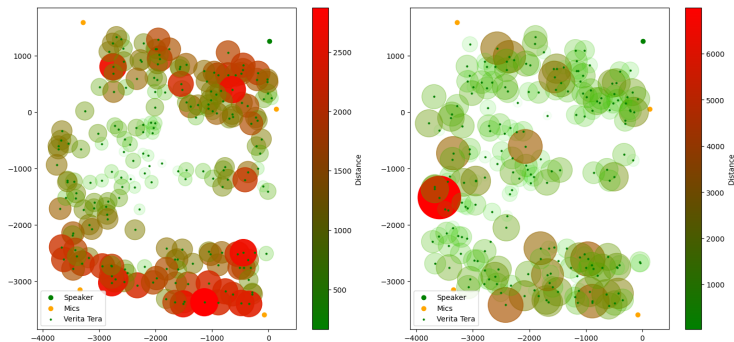


Figure: Distance d'erreur pour le modèle linear: à gauche RMS full, à droite RMS rolling window

Entraînement de modèles

Modèle K-nearest-neighbours[2]: heatmap d'erreur

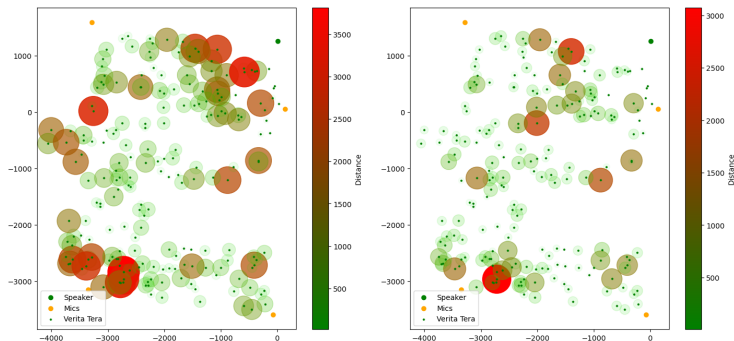


Figure: Distance d'erreur pour le modèle KNN: à gauche RMS full, à droite RMS rolling window

Conclusion

Analyse et conclusion

	Linear : RMS full	Linear : RMS rolling window	Linear : Paper
Avg distance error (in mm)	1498	1812	1545
Std deviation error (in mm)	582	1040	547
	K-NN : RMS full	K-NN : RMS rolling window	K-NN : Paper
Avg distance error (in mm)	957	528	932
Std deviation error (in mm)	844	512	771

Figure: Résultats de nos différents modèles, comparés aux résultats du papier[3]

- [1] Diego Di Carlo et al. *dEchorate: a Calibrated Room Impulse Response Database for Echo-aware Signal Processing*. 2021. [arXiv: 2104.13168 \[eess.AS\]](#).
- [2] Jingwen Sun, Weixing Du, and Niancai Shi. "A Survey of kNN Algorithm". In: *Information Engineering and Applied Computing 1* (May 2018). DOI: [10.18063/ieac.v1i1.770](#).
- [3] Mason Wang et al. *SoundCam: A Dataset for Finding Humans Using Room Acoustics*. 2024. [arXiv: 2311.03517 \[cs.SD\]](#).

*Merci de votre attention !
Des questions ?*