

# SurfOnHertz : un navigateur hertzien en radio logicielle pour l'indexation des bandes de radiodiffusion FM

B. HAPPI TIETCHE<sup>1</sup>, O. ROMAIN<sup>1</sup>, B. DENBY<sup>2</sup>, F. DE DIEULEVEULT<sup>3</sup>, B. GRANADO<sup>4</sup>, M. KERABERNOU<sup>4</sup>, H. KHEMIRI<sup>5</sup>, G. CHOLLET<sup>5</sup>, D. PETROVSKA<sup>5</sup>, R. BLOUET<sup>6</sup>, K. HACHICHA<sup>1</sup>, S. VIATEUR<sup>1</sup>

<sup>1</sup> Université Pierre et Marie Curie, Laboratoire d'Informatique de Paris 6 – LIP6 – UMR7606

<sup>2</sup> Université Pierre et Marie Curie, Laboratoire SIGMA, ESPCI-ParisTech

<sup>3</sup> Laboratoire LIST – CEA

<sup>4</sup> Laboratoire ETIS – Université Cergy Pontoise

<sup>5</sup> Institut Télécom Paris

<sup>6</sup> YACAST

<sup>1</sup>Brunel.Happi-Tietche@lip6.fr, <sup>1</sup>Olivier.Romain@upmc.fr, <sup>2</sup>denby@ieee.org

**Thèmes** – Architectures matérielles et logicielles, Adéquation Algorithme Architecture et indexation

**Problème traité** – Développement d'une architecture matérielle temps-réel qui extrait et indexe le contenu de toutes les stations radio FM en parallèle ;

**Originalités** – Démodulation simultanée de toutes les stations FM et indexation temps-réel ; solution globale de l'antenne à l'interface homme-machine interactive ;

**Résultats** – Prototype fonctionnel d'un navigateur hertzien pour la bande FM. Le prototype utilise une architecture massivement parallèle de démodulation portée sur FPGA pour l'extraction de toutes les stations dans la bande 87-108 MHz ; l'indexation des flux est effectué « offline » sur un GPU.

## 1 Introduction

Le passage de la radiodiffusion analogique (AM et FM) à l'ère du numérique va entraîner une modification importante des contenus diffusés. L'utilisation des techniques d'indexation audio dans les médias de radiodiffusion commerciale deviendra primordiale pour diverses applications, comme la radio à la demande et la pige musicale. Actuellement, les systèmes de surveillance commerciaux sont des installations massives, en raison du grand nombre de canaux à traiter, et nécessitent la présence d'un opérateur par flux annoté.

Les applications grands publics de la radio à la demande et d'indexation audio offriront de nouvelles possibilités telles que la programmation en fonction de critères (genre musical, évitement de publicités, etc.), ou des informations en temps réel sur les programmes l'ensemble des stations. Pour ces applications, la démodulation de toutes les stations en parallèle est nécessaire.

La radio logicielle permet de répondre à ce défi. Initialement orientée vers les radiocommunications, les radars et la gestion opportuniste du spectre, la radio logicielle est maintenant utilisée pour les applications commerciales de la radio aussi [1-8]. De part son procédé d'acquisition/numérisation au plus près de l'antenne, il est possible d'acquérir l'ensemble des canaux diffusés et de les traiter numériquement pour en restituer des informations sur les flux.

L'article présente le développement d'un prototype de « navigateur hertzien » pour la bande FM, utilisant une architecture de récepteur défini par logiciel pour la démodulation simultanée de toutes stations, et un algorithme de reconnaissance des média porté sur un GPU.

## 2 Navigateur pour la bande FM

Le prototype est composé d'un étage de pré-amplification et de filtrage RF (Front-End analogique), us dispositif de démodulation sur une carte FPGA ; et une liaison série USB connecté à un PC sur lequel une carte GPU permet de traiter les données démodulées.

### 2.1 Front-End analogique

Dans l'architecture retenue, tous les canaux de la bande FM sont convertis par un CAN sans transposition de fréquence. Le front end doit présenter à l'entrée du convertisseur des signaux d'amplitude suffisante en minimisant l'intermodulation. La difficulté, dans ce cas, réside dans la conception d'un amplificateur à grand gain, faible facteur de bruit et fort IP3. La bande FM (87 à 108 MHz) comporte une cinquantaine de stations espacées en moyenne de 400kHz, les paramètres les plus importants deviennent gain et linéarité. Il est clair que dans ce cas la difficulté provient du nombre de canaux simultanés et de l'amplitude requise.

L'architecture se résume à une chaîne d'amplification et filtrage dans la bande concernée. Les paramètres, gain, facteur de bruit, et IP3 ont ensuite été optimisés en sacrifiant la consommation et la miniaturisation du front end. Des simulations conduites sous Genesys (Agilent) ont permis de valider la configuration optimale comportant deux filtres passe-bande passifs de Butterworth d'ordre 3, deux amplificateurs à gain fixe et forte linéarité (ADL5535), et un amplificateur à gain commandé, le LMH6514. Le gain global de la chaîne est de 60 dB, la dynamique, dû à l'amplificateur à gain commandé, de 43 dB. Le point d'interception du 3<sup>ème</sup> ordre global est de 42 dBm, ce qui permet de ne pas dégrader le rapport C/N de plus de 62 dB lorsque le niveau de sortie vaut -10 dBm.

## 2.2 Algorithme de démodulation parallèle

Le synoptique de l'architecture de démodulation parallèle à quatre branches est donné sur la figure 1. La figure 2 présente la prototype expérimental réalisé.

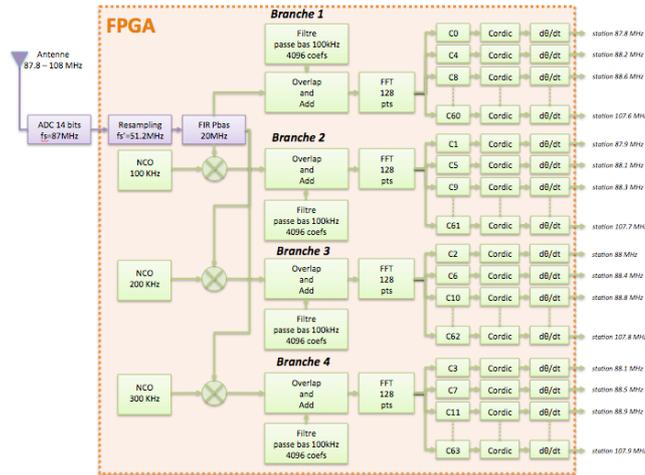


Figure 1 : Architecture de démodulation

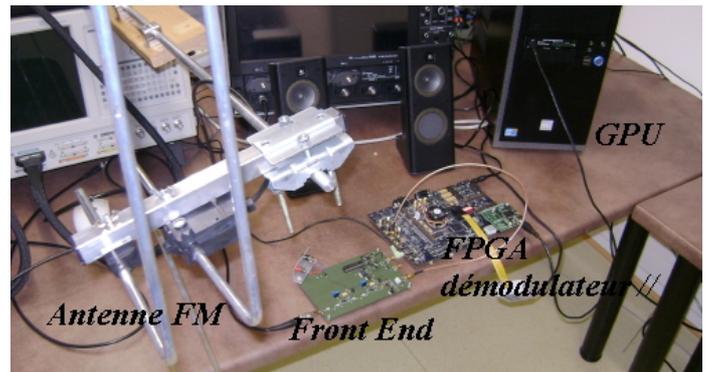


Figure 2 : Prototype FPGA de démodulateur parallèle

Le signal en sortie du front-end est échantillonné à 14 bits à 87 MS/s. C'est un « bandpass sampling » [9], qui effectue une transposition des stations FM vers une bande s'étendant de 0.8 MHz et 20 MHz. Le signal est ensuite ré-échantillonné à 51.2 MHz avant d'être partitionné en fréquence par 4 bancs de filtres polyphases. Ces filtres sont implémentés selon la technique du « Overlap/Add » [10 - 12], suivie d'une TFD sur 128 points. Les décalages fréquentiels des branches sont obtenus avec un NCO avant le « Overlap/Add ». La branche 1 démodule en parallèle les stations 87.8MHz à 107.6 MHz par pas de 400kHz ; la branche 2 les stations aux fréquences de 87.9MHz à 107.7MHz par pas de 400kHz, et ainsi de suite. Cette structure permet ainsi d'obtenir l'équivalent de 32 filtres passe-bande de 100 kHz d'espacement. La démodulation FM est ensuite réalisée au moyen d'un opérateur de type Cordic, pour le calcul de la fonction « arctan », et d'une dérivée temporelle première.

## 2.3 Implémentation matérielle

L'implémentation matérielle de l'architecture de démodulation a été réalisée en VHDL sur une carte de développement à base de FPGA de type Altera Stratix II (EP2S180) (figure 2). Le tableau ci-dessous donne les principales caractéristiques de cette IP. Le transfert des données démodulées au PC intégrant le GPU est réalisé au moyen d'une liaison USB2.

Table 1 : Caractéristiques de l'IP démodulateur parallèle en VHDL

ALUTs	36913/143520 (26%)
Registres dédiés	44498/143520 (31%)
Total registers	44498
Memoire bits	1301457/9383040 (14%)
DSP block 9 bit	768/768 (100%)
Fréq max	101.74 MHz

## 3 Indexation audio

Deux types d'analyse des flux démodulés ont été implémentés ; un algorithme de reconnaissance de média (développé par la société YACAST), porté sur GPU, et une détection automatique des publicités.

### 3.1 Détection automatique publicités connues

Nous proposons ici un nouveau système de détection des publicités basé sur l'acquisition automatique des unités acoustiques à travers les outils ALISP (Automatic Language Independent Speech Processing) [13]. Les outils ALISP

ont été développés initialement pour le codage de la parole à très bas débit, et ont été exploités avec succès pour d'autres tâches, telle que la reconnaissance du locuteur.

### 3.1.1 Acquisition et modélisation des unités ALISP

Une phase d'apprentissage des HMM (modèles de Markov cachés) est d'abord nécessaire pour définir les unités acoustiques ALISP. Soixante-cinq classes HMM ont été apprises d'une façon itérative. La topologie utilisée de 3 états gauche-droite avec une gaussienne modélisant la probabilité d'émission. Pour initialiser l'apprentissage, une première segmentation est obtenue en deux étapes : une de segmentation par la décomposition temporelle [14] et une de classification par quantification vectorielle [15]. Les paramètres audio sont établis à travers les coefficients MFCCs.

### 3.1.2 Protocol expérimental

Nous avons établi un protocole expérimental sur 5 jours d'enregistrement de données. Les données sont exploitées ainsi :

- Données de développement: 24h00 d'enregistrement audio contenant 12 radios (288 heures) pour l'apprentissage des modèles ALISP et 72h00 d'enregistrement pour étudier la stabilité des transcriptions ALISP des publicités.
- Données de référence : 2172 publicités à détecter.
- Données d'évaluation : 24h00 d'enregistrement audio contenant 11 radios contenant 802 publicités à détecter.

La comparaison des transcriptions ALISP se fait à travers la distance de Levenshtein [16].

### 3.1.3 Résultats

Pour évaluer les performances de notre système, 24h00 d'enregistrement contenant 11 radios contenant 802 publicités a été utilisé. Les tests réalisés ont montré un taux de précision de 98% et un taux de rappel de 97%. Puis, 783 publicités ont été correctement détectées et 27 fausses alarmes signalées. Nous avons découvert que 9 publicités n'ont pas été détectées à cause des erreurs dans les annotations, et les 10 autres publicités étaient différentes des références correspondantes. Ces erreurs corrigées, le nombre des publicités détectées a passé à 792.

**Tableau 2 : Précision, Rappel, publicités non détectées et fausses alarmes calculés pour chaque radio.**

Radio ID	P%	R%	Publicités à détecter	Publicités non détectées	Fausse alarmes
1	100	98	76	1	0
165038	92	96	26	1	2
3	94	98	82	1	5
541	90	100	28	0	3
547	94	96	66	2	4
548	99	97	111	3	1
552	96	96	82	3	3
553	95	98	72	1	3
555	84	100	21	0	4
557	99	96	112	4	1
558	98	97	126	3	2
Total			802	19	27

## 3.2 Reconnaissance des médias

Le système présenté dans cette section vise à identifier le média diffusé dans l'environnement immédiat du démodulateur. Il s'agit d'une tâche d'identification dans un ensemble fermé constitué de l'ensemble des radios captés par le démodulateur. Un tel système permet par exemple d'imaginer la réalisation d'un appareil mobile de mesure d'audience ou encore d'adapter le comportement d'un site web en fonction du média écouté par l'utilisateur.

### 3.2.1 Description du système

**Modélisation des médias :** Le système cherche à identifier l'un des  $K$  médias démodulés par la plateforme SurfOnHerz. Toutes les 3 secondes,  $K$  ensembles de signatures chacun associé à l'un des médias sont créés. Pour chaque média  $k$ , les signatures sont ajoutées à un buffer circulaire. Un ensemble de signatures correspond à une suite de vecteurs MFCC de dimension 16. Les  $K$  buffers circulaires permettent d'avoir une mémoire de 30 secondes sur les contenus diffusés sur chaque média.

**Reconnaissance des médias :** Lorsqu'une requête d'identification est traitée, une suite de vecteur MFCC est extraite du signal sonore capté par le micro de la plateforme SurfOnHerz. Le signal capté a une durée d'environ 5 secondes. Pour chaque média  $k$ , on cherche ensuite à l'intérieur du buffer circulaire  $b_k$  la séquence de 3 secondes qui

minimise la distance  $D$  avec la séquence extraite du signal capté. Le média candidat est celui pour lequel cette distance est minimale. Si cette distance est inférieure à un seuil fixé expérimentalement, l'environnement sonore autour de la plateforme SurfOnherz est qualifié sinon il est marqué comme étant inconnu.

### 3.2.2 Résultats

Une base de test est actuellement en cours de réalisation. Pour cela nous avons développé une application iPhone qui permet d'enregistrer et d'envoyer le signal sonore correspondant à l'environnement de l'iPhone ainsi que la qualification du média parmi une liste de 60 médias. Nous disposons actuellement d'environ 100 extraits. Sur ces extraits, les tests réalisés ont montré un taux de bonne détection d'environ 96% alors que le taux de fausse alarme était nul. Ces résultats sont très bons mais peu significatif. L'ajout de nouveaux extraits à la base de test permettra d'obtenir une estimation des performances plus fiables. En outre il faudrait qualifier l'environnement de captation en fonction du rapport signal sur bruit.

## 4 Implémentation sur GPU de la reconnaissance de médias

Les architectures des processeurs des cartes graphiques, autrement appelés GPU, exploitent un parallélisme de type SPMD permettant d'accélérer des algorithmes possédant un fort taux de parallélisme de données. Les algorithmes développés par Yacast, basés notamment sur les MFCC, exhibent ce type de parallélisme. Nous avons donc choisi une carte graphique Tesla C1060 de Nvidia pour mettre en œuvre les traitements liés aux algorithmes de Yacast. Cette carte se programme aisément grâce aux extensions Cuda du langage C. Pour paralléliser les traitements des MFCC, nous avons choisi de traiter séquentiellement les canaux et de paralléliser le calcul sur les 340 vecteurs et les 16 MFCC considérés, offrant ainsi l'exploitation d'un taux de parallélisme de 5440. Les résultats obtenus indiquent un temps moyen de traitement de 1ms par vecteur sur une carte de moyenne gamme type GeForce 9600 et un temps moyen de traitement de 0.26 ms par vecteur sur la carte Tesla C1060. Le traitement global de 50 canaux sur la carte Tesla C1060 prend 6.24 secondes comparé à un temps de 25 secondes sur une carte GeForce 9600.

## 5 Conclusion

Nous avons présenté dans cet article un prototype de navigateur hertzien pour la bande FM. Le système utilise conjointement une architecture massivement parallèle de démodulation portée sur FPGA qui permet d'extraire toutes les stations contenues dans la bande 87-108 MHz. Les flux sont traités par un GPU pour la reconnaissance des médias et « offline » par une méthode de détection automatique de publicités connues. Le portage de cette dernière méthode sur GPU constitue une des perspectives à courts termes. La seconde portera sur le développement d'un navigateur hertzien pour les stations DRM émettant dans la bande HF.

## 6 Références

- [1] M.L. Dickens, B.P. Dunn, and J.N. Laneman, "Design and Implementation of a Portable Software Radio", IEEE Communications Magazine, vol. 46, no. 8, pp. 58-66, August 2008
- [2] K.T. Lee, S.J. Kim, J.H. Paik and J.S. Seo, "Implementation of Portable Digital Radio Mondiale (DRM) Receiver", in Proc. of the 13th IEEE International Symposium on Consumer Electronics (ISCE2009).
- [3] GNURadio, <http://gnuradio.org/redmine/wiki/gnuradio>
- [4] Récepteur multistandard de NXP, <http://www.eetimes.com/design/automotive-design/4011449/Automotive-radio-receiver-harnesses-Software-Defined-Radio>
- [5] DREAM Project. [http://sourceforge.net/apps/mediawiki/drm/index.php?title=Main\\_Page](http://sourceforge.net/apps/mediawiki/drm/index.php?title=Main_Page)
- [6] Liste des 991 récepteurs numériques commercialisés à ce jour. [http://www.worlddab.org/products\\_manufacturers/receiver\\_manufacturers#S](http://www.worlddab.org/products_manufacturers/receiver_manufacturers#S)
- [7] O. Romain & B. Denby, "Prototype of a Software Defined Broadcast Media Indexing Engine", in proc IEEE International Conference on Acoustic Signal Speech and Signal Processing, ICASSP 2007, 15-20 April 2007, Honolulu, Hawaii USA. ISBN :1-4244-0727-3. doi :10.1109/ICASSP.2007.366360
- [8] O. Romain & B. Denby, "A Real Time Software Radio for the AM Long Wave Band", 11th International Workshop on Systems, Signals and Image Processing, IWSSIP'04 Ambient Multimedia, September 13-15, 2004, Poznan, Poland. ISBN : 83-906074-8-4.
- [9] R. G. Vaughan, N. L. Scott, and D. R. White, "The theory of bandpass sampling," IEEE Transactions on Signal Processing, vol. 39, no. 9, pp. 1973-1984, 1991.
- [10] R.E. Crochiere, "A weighted overlap-add method of short time Fourier analysis/synthesis," IEEE Trans. Acoust. Speech Signal Processing, ASSP-28 (1980) 99-102.
- [11] H. Wang and Al., "Channalizer receiver with WOLA filterbank", in Proc. IEEE International Conference on Radar, 2006, CIE'06, 16-19 october, Shanghai, China.
- [12] G. Savir, "Scalable and Reconfigurable Digital Front-End for SDR Wideband Channelizer », Msc Thesis, 2006, Delf University of Technology, Israel.
- [13] G. Chollet, J. Cernocky, A. Constantinescu, S. Deligne, et F. Bimbot. Towards ALISP: a proposal for Automatic Language Independent Speech Processing, pages 357-358. NATO ASI Series. Springer Verlag, 1999.
- [14] B. Atal. Efficient coding of lpc parameters by temporal decomposition. ICASSP, pages 81-84, April 1983.
- [15] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for Vector quantizer design. IEEE Transactions on Communication, 28(1):84-95, January 1980.
- [16] G. Navarro. A guided tour to approximate string matching. ACM Computing Surveys 33 (1): 31-88. 2001.