



**HAL**  
open science

# FuRIA : un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur utilisant modèles inverses et modèles flous

Fabien Lotte, Anatole Lécuyer, Bruno Arnaldi

► **To cite this version:**

Fabien Lotte, Anatole Lécuyer, Bruno Arnaldi. FuRIA : un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur utilisant modèles inverses et modèles flous. 21ème colloque GRETSI, Sep 2007, Troyes, France. inria-00134971v3

**HAL Id: inria-00134971**

**<https://inria.hal.science/inria-00134971v3>**

Submitted on 17 Oct 2007

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# FuRIA : un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur utilisant modèles inverses et modèles flous

Fabien LOTTE, Anatole LÉCUYER, Bruno ARNALDI

IRISA-INRIA Rennes, Avenue du général Leclerc, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France  
 fabien.lotte@irisa.fr, anatole.lecuyer@irisa.fr, bruno.arnaldi@irisa.fr

**Résumé** – Cet article propose un nouvel algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur. Cet algorithme utilise les modèles inverses ainsi que le nouveau concept de Région d'Intérêt (RI) floue. Il peut automatiquement identifier les RIs et bandes de fréquences pertinentes pour la discrimination. Les activités calculées dans ces RIs peuvent être utilisées comme caractéristiques pour n'importe quel classifieur. Une première évaluation de l'algorithme, utilisant une Machine à Vecteurs Supports (SVM) comme classifieur, est présentée sur le jeu de données IV de la "BCI competition 2003". Les résultats s'avèrent prometteurs avec une précision allant de 85% à 86% contre 84% pour le gagnant de la compétition sur ces données.

**Abstract** – In this paper, we propose a new feature extraction algorithm for Brain-Computer Interfaces (BCIs). This algorithm is based on inverse models and uses the novel concept of fuzzy Region Of Interest (ROI). It can automatically identify the relevant ROIs and their reactive frequency bands. The activity in these ROIs can be used as features for any classifier. A first evaluation of the algorithm, using a Support Vector Machine (SVM) as classifier, is reported on data set IV from BCI competition 2003. Results are promising with an accuracy on the test set ranging from 85% to 86% versus 84% for the winner of the competition.

## 1 Introduction

Les Interfaces Cerveau-Ordinateur (ICOs) sont des systèmes de communication permettant d'envoyer des commandes uniquement par le biais de l'activité cérébrale [1], celle-ci étant généralement mesurée par ÉlectroEncéphalographie (EEG). L'identification de tâches mentales dans une ICO repose sur deux points clés principaux : l'extraction de caractéristiques et la classification [2]. Le premier vise à décrire les signaux EEG par quelques valeurs pertinentes appelées "caractéristiques" tandis que le dernier a pour but d'attribuer automatiquement une classe à ces caractéristiques. Nous nous intéressons ici à l'extraction de caractéristiques, domaine pour lequel il est reconnu que l'exploration de nouveaux algorithmes est nécessaire [3].

Ces trois dernières années, les modèles inverses se sont avérées être des algorithmes d'extraction de caractéristiques prometteurs [4, 5, 6, 7]. De tels modèles peuvent calculer l'activité dans tout le volume cérébral, en utilisant uniquement les signaux EEG et un modèle de tête modélisant la conductivité dans le cerveau par un ensemble de voxels (éléments de volume). L'activité calculée ensuite dans quelques régions du cerveau peut fournir des caractéristiques efficaces pour les ICOs [7].

En dépit de résultats généralement bons, quelques limitations subsistent. Il s'avère en effet difficile de concilier à la fois la généralité, i.e., la capacité de traiter n'importe quel type de tâche mentale, et le fait de générer peu de caractéristiques. D'un côté, les méthodes génériques et automatiques génèrent un grand nombre de caractéristiques, car elles utilisent l'activité de voxels seuls comme caracté-

ristiques [5]. En effet, les voxels peuvent être dépendants les uns des autres et devraient donc être regroupés au sein de régions cérébrales. D'un autre côté, des méthodes générant peu de caractéristiques ont été proposées, mais elles ne sont plus génériques car nécessitant de la connaissance *a priori* sur le sujet et n'étant ainsi utilisable que pour de l'imagerie motrice [4, 6]. Enfin, une méthode proposée récemment est à la fois générique et génératrice de peu de caractéristiques étant donné que les voxels dépendants sont regroupés en Régions d'Intérêt (RIs). Cependant cette méthode a le défaut de ne pas être complètement automatique et d'être limitée à l'utilisation de deux RIs dont l'extension spatiale est difficile à identifier [7].

Dans cet article, nous essayons de dépasser ces limitations. Nous proposons un algorithme d'extraction de caractéristiques générique qui peut automatiquement identifier n'importe quel nombre de RIs pertinentes et peut définir correctement leur extension spatiale grâce au nouveau concept de RI floue. Cet algorithme s'appelle FuRIA, acronyme de *Fuzzy Region of Interest Activity*.

## 2 L'algorithme FuRIA

FuRIA est un algorithme d'extraction de caractéristiques avec apprentissage. En effet, il peut automatiquement identifier quelles sont les RIs pertinentes pour discriminer les tâches mentales et les bandes de fréquences dans lesquelles ces RIs réagissent. Cet apprentissage est réalisé en 5 étapes qui sont : l'analyse statistique, le clustering, la fusion, l'adaptation et enfin la "fuzzification". Cette partie décrit

ces 5 étapes ainsi que l'utilisation de FuRIA pour l'extraction de caractéristiques.

## 2.1 Analyse statistique

Cette étape consiste à comparer, en utilisant un test statistique, l'activité de chaque voxel, éventuellement moyennée au cours du temps, entre les différentes classes. Cette analyse est effectuée pour chacune des  $N_f$  fréquences considérées. Pour cela les signaux EEG sont filtrés dans une bande de fréquence de 2 Hz autour de chaque fréquence considérée, et un modèle inverse est utilisé pour calculer l'activité des différents voxels. Une fois l'analyse effectuée, nous obtenons  $N_f$  cartes de statistiques, chacune révélant quels sont les voxels dont l'activité est statistiquement différente entre les classes pour la fréquence correspondante. Nous appelons ces derniers des "voxels significatifs".

## 2.2 Clustering

Pour chaque carte de statistiques calculée à l'étape précédente, un algorithme de clustering est utilisé dans le but d'agréger les voxels significatifs sous forme de RIs. Pour utiliser cet algorithme, un vecteur à 4 dimensions  $[x, y, z, s]$  est associé à chaque voxel significatif. Les coordonnées  $x, y, z$  sont les coordonnées 3D du voxel dans le modèle de tête choisi, et  $s$  la statistique du voxel obtenues à l'étape précédente. Le but est de trouver les voxels significatifs qui sont spatialement regroupés, et qui ont des statistiques communes. Ce dernier point permet principalement de ne pas agréger dans une même RI, un voxel avec une statistique positive et un voxel avec une statistique négative. Une fois le clustering effectué, tous les voxels dont les vecteurs associés appartiennent au même cluster, sont agrégés dans la même RI. Les RIs très petites, contenant seulement un ou deux voxels par exemple, peuvent être supprimées. A la fin de cette étape, un ensemble de RIs pertinentes a été identifiée pour chaque fréquence.

## 2.3 Fusion

Après avoir identifié quelles étaient les RIs pertinentes pour chaque fréquence individuelle, il faut trouver quelles RIs sont pertinentes dans quelles bandes de fréquences. Pour ce faire, chaque RI  $\Omega$  est associée à la fréquence  $f$  à laquelle elle a été trouvée, puis la procédure suivante est effectuée :

1. Parmi tous les couples  $(\Omega; f)$ , choisir deux couples  $(\Omega_1; f_1)$  et  $(\Omega_2; f_2)$  tels que le chevauchement entre  $\Omega_1$  et  $\Omega_2$  soit élevé et les bandes de fréquences  $f_1$  et  $f_2$  soient consécutives ou se chevauchent. Nous considérons qu'un chevauchement est élevé si  $card(\Omega_1 \cap \Omega_2) > 0.5 * \min(card(\Omega_1), card(\Omega_2))$  avec  $card(\Omega)$  étant le nombre de voxels dans  $\Omega$ ;
2. Remplacer les couples  $(\Omega_1; f_1)$  et  $(\Omega_2; f_2)$  par l'unique couple  $(\Omega_1 \cup \Omega_2; f_1 \cup f_2)$ . Cela signifie que nous avons choisi l'Union comme opérateur de fusion de RIs;
3. Retourner au point 1, jusqu'à ce que plus aucune RI ne puisse être fusionnée.

## 2.4 Adaptation

À cause de l'utilisation de l'Union comme opérateur de fusion, une RI  $\Omega$  identifiée comme réactive dans la bande de fréquence  $[f_a; f_b]$  peut contenir des voxels qui étaient significatifs à une fréquence comprise entre  $f_a$  et  $f_b$  mais qui ne le sont plus dans  $[f_a; f_b]$ . Par conséquent, pour chaque RI trouvée à l'étape précédente, nous effectuons une autre analyse statistique pour les voxels appartenant à la RI, dans le but de retirer ceux qui ne sont plus significatifs.

## 2.5 Fuzzification et concept de RI floue

Dans les RIs définies jusqu'à présent, nous pouvons remarquer que tous les voxels n'ont pas, statistiquement, le même pouvoir discriminant. Cependant, ces voxels portent tous plus ou moins d'information. De plus, pour un sujet donné, les régions cérébrales correspondant à des fonctions cérébrales particulières ne sont pas bien définies : les frontières de ces régions sont naturellement floues. De ce fait, identifier quels sont les voxels qui doivent être gardés dans une RI est un problème difficile.

Nous pensons que tous les voxels contenant de l'information devraient être dans la RI, mais que ceux qui en contiennent moins, devraient être moins dans la RI. Nous proposons ainsi le concept de RI floue, en s'appuyant sur la théorie des ensembles flous développée par Zadeh [8]. Une RI classique  $\Omega_c$  est définie par l'ensemble des voxels qu'elle contient. L'activité  $\gamma_{\Omega_c}$  de cette RI  $\Omega_c$  est classiquement calculée comme suit :

$$\gamma_{\Omega_c} = \sum_{v \in \Omega_c} \gamma_v \quad (1)$$

où  $\gamma_v$  est l'activité du voxel  $v$ .

Une RI floue  $\Omega_f$  n'est plus définie par un ensemble de voxels, mais par une fonction d'appartenance floue  $\mu$ . Cette fonction donne le degré d'appartenance, dans  $[0; 1]$ , de n'importe quel voxel existant dans la RI  $\Omega_f$ . De cette façon, les RI floues ont des frontières "floues". L'activité dans une RI floue est calculée de cette façon :

$$\gamma_{\Omega_f} = \sum_{v=1}^{N_v} \mu(v) \gamma_v \quad (2)$$

Avec  $N_v$  le nombre total de voxels dans le modèle de tête utilisé. Ce formalisme permet de pondérer chaque voxel en fonction de sa contribution relative au sein de la RI, et ainsi, d'utiliser efficacement toute l'information disponible.

Pour terminer l'apprentissage de FuRIA, toutes les RIs  $\Omega$  sont transformées en RIs floues en associant à chacune d'entre elles une fonction d'appartenance floue  $\mu_{\Omega}$  qui utilise les statistiques calculées lors de l'étape précédente. La forme de cette fonction dépend du modèle inverse et du test statistique utilisé. Dans la partie 3 nous proposons une fonction d'appartenance floue adaptée aux algorithmes que nous avons utilisés.

Une fois l'apprentissage terminé, l'extraction de caractéristiques consiste à filtrer les signaux EEG dans les bandes

de fréquences réactives de chaque RI et à utiliser un modèle inverse pour calculer son activité. Ces activités sont ensuite concaténées en un vecteur de caractéristiques qui peut alors être utilisé avec n'importe quel classifieur.

### 3 Implémentation de l'algorithme

Dans le but d'évaluer FuRIA, nous avons utilisé sLORETA comme modèle inverse [9], un test  $t$  avec permutations comme analyse statistique [10] avec  $\alpha = 0.01$ , et enfin l'algorithme "Mean Shift" [11] en tant qu'algorithme de clustering avec une valeur de 2 pour le paramètre de lissage  $H$ .

Grâce à sLORETA (standardized Low Resolution Electromagnetic Tomography), calculer l'activité dans une RI floue peut être fait grâce à un simple produit matriciel :

$$\gamma_{\Omega_f} = m^T R_{\Omega_f} m \quad \text{avec} \quad R_{\Omega_f} = \sum_{v=1}^{N_v} \mu(v) R_v \quad (3)$$

Ici,  $m$  est le vecteur instantané de mesures pour chacune des  $N_e$  électrodes utilisées, et  $R_v$  une matrice  $N_e * N_e$  telle que  $\gamma_v = m^T R_v m$  [7]. Ainsi, calculer l'activité dans une RI floue est très rapide étant donné que  $R_{\Omega_f}$  peut être calculée hors-ligne. Dans cette étude, nous avons utilisé un modèle de tête standard contenant 2394 voxels.

Enfin, durant l'étape de fuzzification, nous avons eu à choisir les fonctions d'appartenance les plus adaptées. Comme sLORETA possède une faible résolution spatiale, les voxels significatifs avec une relativement forte probabilité d'erreur de type I sont beaucoup plus nombreux que les voxels significatifs avec une relativement faible probabilité d'erreur. Par conséquent, pour chaque RI  $\Omega$ , nous avons choisi une fonction  $\mu_{\Omega}$  donnant un degré d'appartenance qui décroît exponentiellement avec la statistique du voxel :

$$\mu_{\Omega}(v) = \begin{cases} e^{-\frac{1}{2}(\frac{p_{max}-p_v}{\sigma_{max}})^2} & v \in \Omega \\ 0 & v \notin \Omega \end{cases} \quad (4)$$

avec  $\sigma_{max} = 1/(N_{\Omega_s} \sum_{p_v > 0} (p_{min} - p_v)^2)$

$N_{\Omega_s}$  est le nombre de voxels dans la RI  $\Omega$ ;  $p_{min}$  et  $p_{max}$  sont respectivement la probabilité minimale et maximale, parmi les voxels de la RI, de rejeter correctement l'hypothèse nulle  $H_0$ ; et  $p_v$  est la probabilité de rejeter correctement  $H_0$  pour le voxel  $v$ . Plus la RI est large, plus le nombre de voxels avec une faible  $p_v$  est grand lorsque l'on utilise sLORETA. Cela a été pris en compte en mettant  $N_{\Omega_s}$  au dénominateur de  $\sigma_{max}$ .

## 4 Évaluation

Cette section présente une première évaluation de l'algorithme FuRIA. Tout d'abord les données EEG utilisées sont présentées puis les résultats obtenus sont décrits.

### 4.1 Jeu de données EEG

Nous avons utilisé le jeu de données IV de la "BCI competition 2003" [12]. Ces données contiennent des signaux

EEG enregistrés alors qu'un sujet tapait sur un clavier avec un doigt de sa main gauche ou droite. Les signaux EEG ont été échantillonnés à 100 Hz, enregistrés avec 28 électrodes et contenaient les 500 ms précédant la frappe effective sur le clavier. Les données contenaient 314 essais pour l'apprentissage et 100 pour le test. Le but des participants à la compétition était d'identifier, pour chaque essai, la main qui allait être utilisée, à savoir "gauche" ou "droite".

### 4.2 Résultats de l'extraction de caractéristiques avec FuRIA

Nous avons entraîné l'algorithme FuRIA sur l'ensemble d'apprentissage fourni. Nous avons travaillé sur les fréquences comprises entre 3 Hz et 45 Hz, avec un pas de 1 Hz et nous sommes focalisés sur la fenêtre temporelle comprenant les dernières 250 ms pour chaque essai. Cela signifie que FuRIA travaillait avec l'activité moyenne au cours de cette période temporelle. FuRIA a trouvé cinq RIs flous discriminantes, générant ainsi des vecteurs de caractéristiques de dimension 5. Ces RIs et leurs bandes de fréquences correspondantes sont visibles sur la Figure 1.

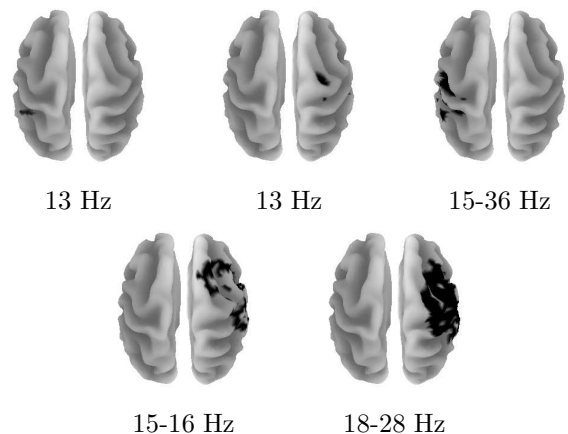


FIG. 1 – Les RIs flous (en noir) et leur fréquences correspondantes qui ont été automatiquement obtenues.

Nous pouvons remarquer que FuRIA a trouvé deux RIs dont les fréquences correspondantes sont dans le rythme  $\mu$  (13 Hz) et trois RIs avec des fréquences dans le rythme  $\beta$  (15-16 Hz, 15-36 Hz, 18-28 Hz). De plus, les RIs trouvées se situent dans le cortex moteur gauche et droit, ce qui est cohérent avec la littérature concernant l'intention de mouvements [13].

### 4.3 Classification

Pour classifier les caractéristiques extraites en utilisant FuRIA, nous avons utilisé une machine à vecteurs supports (SVM) linéaire. En effet, ce classifieur est l'un des plus populaires et des plus efficaces pour la conception d'ICO [2]. Ce classifieur a été classiquement entraîné sur l'ensemble d'apprentissage et ensuite utilisé pour classifier les caractéristiques de l'ensemble de test. Les taux de reconnaissance obtenus après 100 répétitions d'appren-

tissage/test sont rapportés dans la Table 1 et comparés avec deux autres méthodes : celle du gagnant de la compétition et une méthode utilisant les modèles inverses que nous avons proposée précédemment. Les résultats obtenus avec FuRIA sans l'étape de fuzzification (cf partie 2.5) sont aussi rapportés pour comparer.

TAB. 1 – Taux de reconnaissance sur l'ensemble de test

Méthode	Précision
Gagnants "BCI competition 2003" [12]	84 %
Travaux précédents [7]	83 %
FuRIA sans fuzzification	80-82 %
FuRIA avec fuzzification	<b>85-86 %</b>

Nous pouvons remarquer que la méthode utilisant FuRIA avec fuzzification atteint le plus haut score avec un taux de reconnaissance allant de 85% à 86%. Ces résultats suggèrent que les caractéristiques extraites avec FuRIA sont très discriminantes, étant donné qu'elles sont peu nombreuses mais permettent de bons résultats de classification. Nous pouvons aussi remarquer que lorsque l'on utilise FuRIA sans fuzzification, les résultats sont moins bons ce qui montre l'intérêt d'utiliser le concept de RI floue. Enfin, on peut remarquer que dans les méthodes proposées dans [7] et [12], des filtres spatiaux dépendants des données ont été utilisés pour augmenter les performances. Cela suggère que combiner FuRIA avec des filtres tels que ceux décrits dans [7] pourrait donner de meilleurs résultats.

## 5 Conclusion et travaux futurs

Dans cet article<sup>1</sup>, nous avons proposé un algorithme d'extraction de caractéristiques pour les interfaces cerveau-ordinateur appelé FuRIA (*Fuzzy Region of Interest Activity*). Cet algorithme utilise des modèles inverses ainsi que le nouveau concept de Régions d'Intérêt (RIs) floues. FuRIA peut automatiquement identifier quelles sont les RIs pertinentes pour discriminer des classes, ainsi que les bandes de fréquences dans lesquelles ces régions réagissent. L'activité dans ces RIs peut ensuite être utilisée comme caractéristiques avec n'importe quel classifieur. FuRIA a été évalué sur le jeu de données IV de la "BCI competition 2003" en utilisant une Machine à Vecteur Support comme classifieur. Les résultats sont prometteurs étant donné que cette méthode a obtenu un meilleur score que les gagnants de cette compétition.

Les travaux futurs vont être dédiés à l'évaluation de FuRIA sur des jeux de données différents, correspondant à d'autres tâches mentales, d'autres sujets et à un plus grand nombre de classes. Il est également envisagé d'améliorer FuRIA en prenant en compte la dimension temporelle. Enfin, il pourrait être intéressant de comparer les RIs automatiquement obtenues par FuRIA et leurs per-

formances avec celles de RIs définies par des experts en neurophysiologie.

## Références

- [1] J. R. Wolpaw, N. Birbaumer, D. J. McFarland, G. Pfurtscheller, and T. M. Vaughan. Brain-computer interfaces for communication and control. *Clinical Neurophysiology*, 113(6) :767–791, 2002.
- [2] F. Lotte, M. Congedo, A. Lécuyer, F. Lamarche, and B. Arnaldi. A review of classification algorithms for eeg-based brain-computer interfaces. *Journal of Neural Engineering*, 4 :R1–R13, 2007.
- [3] D. J. McFarland, C. W. Anderson, K.-R. Muller, A. Schlogl, and D. J. Krusienski. Bci meeting 2005-workshop on bci signal processing : feature extraction and translation. *IEEE Trans. on Neural Syst. and Rehab. Eng.*, 14(2) :135 – 138, 2006.
- [4] L. Qin, L. Ding, and B. He. Motor imagery classification by means of source analysis for brain computer interface applications. *Journal of Neural Engineering*, 1(3) :135–141, 2004.
- [5] R. Grave de Peralta Menendez, S. Gonzalez Andino, L. Perez, P.W. Ferrez, and J. del R. Millán. Non-invasive estimation of local field potentials for neuro-prosthesis control. *Cognitive Processing*, 2005.
- [6] M. G. Wentrup, K. Gramann, E. Wascher, and M. Buss. Eeg source localization for brain-computer-interfaces. In *2nd International IEEE EMBS Conference on Neural Engineering*, pages 128–131, 2005.
- [7] M. Congedo, F. Lotte, and A. Lécuyer. Classification of movement intention by spatially filtered electromagnetic inverse solutions. *Physics in Medicine and Biology*, 51(8) :1971–1989, 2006.
- [8] L. A. Zadeh. Fuzzy sets. *Fuzzy sets, fuzzy logic, and fuzzy systems : selected papers by Lotfi A. Zadeh*, pages 19–34, 1996.
- [9] R. D. Pascual-Marqui. Standardized low resolution brain electromagnetic tomography (sloreta) : technical details. *Methods and Findings in Experimental and Clinical Pharmacology*, 24D :5–12, 2002.
- [10] A. P. Holmes, R. C. Blair, J. D. Watson, and I. Ford. Nonparametric analysis of statistic images from functional mapping experiments. *Journal of Cerebral Blood Flow and Metabolism*, 16 :7–22, 1996.
- [11] D. Comaniciu and P. Meer. Mean shift - a robust approach toward feature space analysis. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(5) :603–619, 2002.
- [12] Y. Wang, Z. Zhang, Y. Li, X. Gao, S. Gao, and F. Yang. Bci competition 2003-data set iv : an algorithm based on cssd and fda for classifying single-trial eeg. *IEEE Trans. on Biomedical Engineering*, 51(6) :1081–1086, 2004.
- [13] G. Pfurtscheller. Eeg event-related desynchronization (erd) and event-related synchronization (ers). *Electroencephalography, 4th ed.*, pages 958,967, 1999.

<sup>1</sup>Les auteurs souhaitent remercier Oscar Yanez-Suarez, Morgane Rosendale, Stéphanie Gerbaud, Yann Renard, Marco Congedo et des rapporteurs anonymes pour leur aide. Ce travail est soutenu par l'ANR à travers le projet Open-ViBE (ref. ANR05RNTL01601).