



**HAL**  
open science

## **MATTONG : Une interface graphique sous MatLab pour le suivi du contour de la langue à partir d'images ultrasons**

T Fux, A Amelot, L Crevier-Buchman, C Pillot-Loiseau, Martine Adda-Decker

### **► To cite this version:**

T Fux, A Amelot, L Crevier-Buchman, C Pillot-Loiseau, Martine Adda-Decker. MATTONG : Une interface graphique sous MatLab pour le suivi du contour de la langue à partir d'images ultrasons. 2014. halshs-01129180

**HAL Id: halshs-01129180**

**<https://shs.hal.science/halshs-01129180>**

Submitted on 10 Mar 2015

**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.

# MATTONG : Une interface graphique sous MatLab pour le suivi du contour de la langue à partir d'images ultrasons

T. FUX<sup>1</sup>, A. Amelot<sup>1</sup>, L. Crevier-Buchman<sup>1,2</sup>, C. Pillot-Loiseau<sup>1</sup>,  
M. Adda-Decker<sup>1,3</sup>

(1) Laboratoire de Phonétique et de Phonologie, CNRS, UMR7018, Paris, France

(2) Hôpital Européen Georges Pompidou, Univ. Paris-Descartes, Paris, France

(3) Traitement du Langage Parlée, LIMSI, CNRS, Orsay, France

thibaut.fux@hotmail.fr, angelique.amelot@univ-paris3.fr,  
lise.buchman@numericalbe.fr, claire.pillot@univ-paris3.fr, madda@limsi.fr

## RÉSUMÉ

---

L'imagerie ultrason (US) est un outil de plus en plus utilisé dans le domaine de la parole car elle est non invasive et permet de visualiser un articulatoire majeur dans la production vocale : la langue. Toutefois, sur ces images US, le contour de la langue doit y être identifié. MATTONG est une interface graphique conviviale développée sous MatLab permettant d'effectuer un suivi du contour de la langue à partir d'images ultrason. MATTONG reprend l'algorithme de suivi développé par (Li *et al.*, 2005a), utilisé dans le logiciel EdgeTrak (Li *et al.*, 2005b). La particularité de MATTONG est qu'il facilite l'utilisation de cet algorithme en apportant une interface graphique permettant d'accélérer les manipulations sur le contour de la langue et de les rendre plus fiables et reproductibles. Il intègre également plusieurs outils tels que des outils de filtrage d'images, la conversion de vidéos en série de fichiers image ou l'exportation des résultats sous divers formats.

## ABSTRACT

---

### **MATTONG : MatLab(R) Tongue Tracker on Graphical User Interface**

Ultrasound (US) imaging is widely used in the speech domain since it is non-invasive and allows viewing of one important articulator in speech production : the tongue. On these US images, the tongue contour need to be located. MATTONG is a graphical interface developed with MatLab allowing to track the tongue contour on US images. MATTONG is based on the algorithm developed by (Li *et al.*, 2005a) in the EdgeTrak software. The specificity of MATTONG is the following : it facilitates the use of this algorithm by adding a graphical interface enabling fast and easy tongue contours manipulations and making them more reliable and reproducible. It also proposes various tools such as image filtering, video file to images files or data exportation in different formats.

**MOTS-CLÉS :** Image, Ultrason, EdgeTrak, Contour, Langue.

**KEYWORDS:** Image, ultrasound, EdgeTrak, Contour, Tongue.

---

## 1 Introduction

L'utilisation d'imageries ultrasons constitue un moyen simple, efficace et non invasif d'observer les mouvements de la langue durant la production vocale. Cette technique est largement utilisée

dans divers domaines tels que la médecine, la phonétique, etc. Toutefois, les données issues de tels capteurs ne constituent pas des données prêtes à l'emploi mais se composent d'une série d'images brutes dans lesquelles un contour de langue peut être distingué. Ainsi, plusieurs auteurs se sont penchés sur la problématique de la détection automatique (ou semi-automatique) du contour de la langue, voir par exemple (Akgul *et al.*, 1999, 2000; Tang *et al.*, 2012; Roussos *et al.*, 2009; Wrench, 2007). Les méthodes les plus courantes sont basées sur le principe des contours actifs (communément appelé *snake*). L'un des algorithmes les plus répandus est celui développé par (Li *et al.*, 2005a) également connu sous le nom de EdgeTrak (nom du logiciel développé par les auteurs). Même si cet algorithme est très performant, son utilisation reste difficile à partir de l'interface graphique EdgeTrak et ceci pour plusieurs raisons. L'importation d'images demande un format spécifique et ne peut se faire que sur des images numérotées comportant le même nombre de chiffres. D'autre part, les corrections d'erreur de détection (inévitables dans tout type d'algorithme de suivi automatique) reste quelque peu difficiles. On peut imaginer plusieurs options d'affichage, d'exportation ou encore de filtrage pour donner à EdgeTrak un aspect bien plus attrayant. C'est pourquoi, nous avons décidé de développer une interface graphique conviviale (cf. figure 1) intégrant des fonctionnalités qui sont absentes dans EdgeTrak ou difficilement abordables. Ainsi, l'apport majeur de MATTONG est la possibilité pour un utilisateur novice de réaliser des analyses, des corrections de contours, des exportations de données et du filtrage d'images US tout en profitant de la puissance de l'algorithme EdgeTrak.

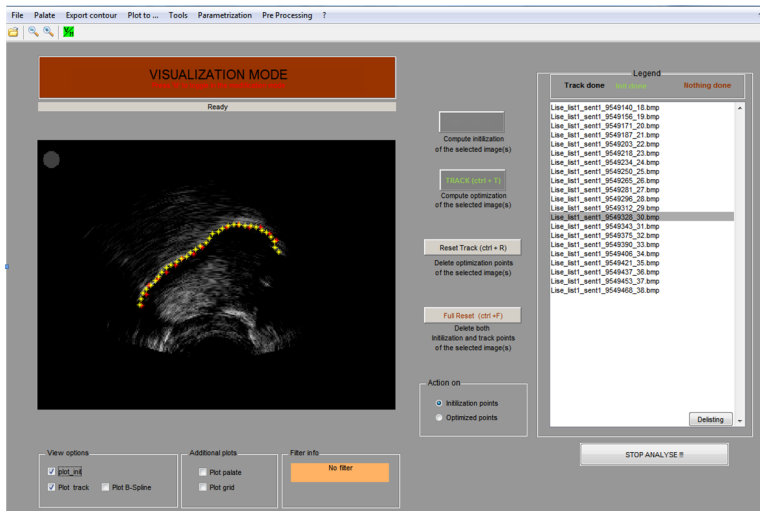


FIGURE 1 – Copie d'écran de MATTONG. On peut voir à droite la liste des fichiers images, au milieu une série de fonctionnalités pour le suivi automatique du contour de la langue. A gauche la partie visualisation inclue une zone de notification, un affichage des images et les contours de la langue ainsi que plusieurs options d'affichage et des informations concernant le filtrage de l'image. La barre des menus comprend plusieurs outils ainsi qu'un onglet de *paramétrisation* permettant de régler les paramètres propres au contour actif.

Dans cet article, nous exposerons tout d'abord les principes généraux de fonctionnement de l'algorithme EdgeTrak (Section 2) et nous préciserons les différences entre ce dernier et la version implémentée dans MATTONG (Section 3). Précisons ici que nous avons fait le choix dans un

premier temps de développer le code source sous MatLab plutôt que d'intégrer directement le code de EdgeTrak (développé en langage C++). Par la suite, nous exposerons les détails de son fonctionnement (Section 4) ainsi que les différents outils intégrés dans notre interface graphique (Section 5).

## 2 Algorithme de détection du contour de la langue de Edge-Trak

L'algorithme EdgeTrak (Li *et al.*, 2005a) se base sur un modèle de contour actif (ou *snake*). Un *snake* est formé d'une série de points mobiles le long d'une courbe pouvant se rapprocher de plus en plus près d'un contour d'une image en minimisant une fonction d'énergie. Cette fonction d'énergie tient compte de plusieurs paramètres directement issus de l'image (énergie externe,  $E_{ext}$ ) mais également des informations liées à la forme du contour et la position des points qui le composent (énergie interne,  $E_{int}$ ) telles que la distance entre les points, la courbure, etc. L'énergie interne a pour rôle de contraindre la courbe en termes de continuité et de régularité alors que l'énergie externe contraint la courbe à épouser une région d'intérêt dans l'image. Le modèle de contour actif proposé par (Li *et al.*, 2005a) se base sur trois énergies de base ainsi qu'une fonction d'énergie spécialement développée par les auteurs pour le suivi du contour de la langue sur des images ultrasons. Ces énergies sont pour les énergies internes : (1) énergie de courbure  $E_{smooth}$  (2) énergie de continuité (ou de distance)  $E_{dist}$ . Pour l'énergie externe, les auteurs utilisent (3) l'énergie du gradient  $E_{grad}$ . A cette énergie externe, les auteurs rajoutent une énergie de bande  $E_{band}$  qui permet de tenir compte de l'intensité dans les régions voisines du contour. En effet, une image ultrason de la langue aura la particularité de présenter une ligne blanche plus ou moins distinctement formée par la réflexion des ondes ultrasons au niveau de la partie supérieure de la langue. Ainsi, le haut de la langue se situera juste en dessous de cette ligne blanche. Les auteurs proposent d'intégrer une énergie qui correspond au rapport d'intensité entre une région située au dessus de la courbe par rapport à une région située en dessous et ce localement (i.e. pour chaque point constituant le contour, cf. figure 2). Ainsi, la fonction de coût total ( $E_{TOTAL}$ ) du modèle de (Li *et al.*, 2005a) se présente de la façon suivante :

$$E_{TOTAL} = \alpha E_{int} + \beta E'_{ext} = \alpha E_{int} + \beta (E_{band}(v_i) \cdot E_{ext}(v_i))$$

avec

$$E_{int} = \alpha_1 E_{smooth} + \beta_1 E_{dist} = \alpha_1 \left( 1 - \frac{\overrightarrow{v_{i-1}v_i} \cdot \overrightarrow{v_i v_{i+1}}}{|\overrightarrow{v_{i-1}v_i}| \cdot |\overrightarrow{v_i v_{i+1}}|} \right) + \beta_1 ||v_i - v_{i-1}| - d|$$

$$E_{band}(v_i) = \begin{cases} pen & \text{si } dif(v_i) < 0 \\ 1 - dif(v_i) & \text{sinon} \end{cases} \quad \text{où, } dif(v_i) = \frac{1}{n \cdot N} \cdot \left( \sum_{p_j \in R_i} I(p_j) - \sum_{p'_j \in R'_i} I(p'_j) \right)$$

et

$$E_{ext}(v_i) = 1 - |\nabla I(v_i)| / M$$

Dans ces équations  $\alpha$  et  $\beta$  ainsi que  $\alpha_1$  et  $\beta_1$  correspondent à des facteurs de pondération des énergies,  $d$  est la distance moyenne entre deux éléments du contour,  $I$  l'image considérée,  $M$  une constante de normalisation,  $\nabla$  l'opérateur de gradient,  $pen$  une valeur de pénalité définie à 2,  $n$  le nombre de pixels dans la région  $R_i$  ou  $R'_i$ , et  $N$  un facteur de normalisation d'intensité égale à 255. Pour plus de précisions nous renvoyons le lecteur vers l'article originel (Li *et al.*, 2005a).

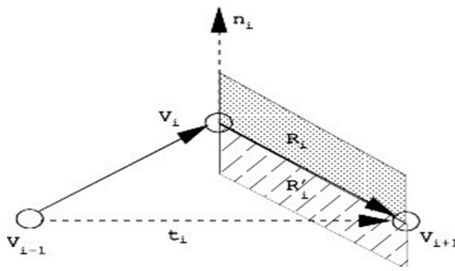


FIGURE 2 – Définition des régions locales d'intensité (source : (Li *et al.*, 2005a)).  $v_i$  est le point considéré,  $v_{i-1}$  et  $v_{i+1}$  respectivement le point précédent et suivant.  $t_i$  est le vecteur tangent en  $v_i$  et  $n_i$  le vecteur normal. La zone  $R_i$  est définie comme un quadrilatère dont un côté est défini entre  $v_i$  et  $v_{i+1}$  et l'autre correspond à la normale.

### 3 Suivi du contour de la langue : MATTONG Vs EdgeTrak

L'algorithme de suivi du contour de la langue que nous avons implémenté pour MATTONG est directement issu de l'interprétation que nous avons faite de l'article d'origine (Li *et al.*, 2005a) avec quelques simplifications.

Nos simplifications concernent les différents termes de la fonction de coût et notamment l'énergie correspondant à la distance entre les points du contour ( $E_{dist}$ ). Nous avons choisi de supprimer cette fonction et d'interpoler les points de façon équidistante après chaque optimisation du contour. Ceci est possible dans notre algorithme actuel car la recherche du contour ne se fait que dans une seule direction. En effet, dans l'algorithme de base, il est mentionné que d'abord la zone de recherche ne s'oriente que dans la direction de la normale ( $n_i$  dans la figure 2) et qu'après plusieurs itérations une recherche est effectuée suivant la tangente ( $t_i$ ). Dans MATTONG la recherche suivant la tangente est en cours de développement. Cependant, d'après plusieurs tests effectués sur des images US, l'absence de cette direction de recherche affecte peu la recherche du contour. Enfin, la recherche du contour optimal par programmation dynamique est dans l'algorithme d'origine celui proposé par (Amini *et al.*, 1990). Dans MATTONG nous avons codé un algorithme de Viterbi (Viterbi, 1967) permettant d'aboutir à des résultats similaires. Nous avons comparé quelques suivis réalisés par EdgeTrak et MATTONG : nous obtenons des contours très similaires ce qui conforte nos choix et notre méthodologie (cf. figure 3).

### 4 MATTONG – principe de fonctionnement

Dans EdgeTrak, le processus de suivi consiste à annoter manuellement la première image puis à démarrer le processus d'optimisation sur cette image. Ensuite il est possible de lancer l'analyse de toutes les images automatiquement. Le contour une fois optimisé sur une image est ensuite propagé sur l'image suivante en tant que contour d'initialisation avant d'être à son tour optimisé. Sur ce point, le fonctionnement de MATTONG diffère quelque peu de celui de EdgeTrak. Nous avons remarqué qu'il était bénéfique de procéder en premier lieu à une estimation approximative de la position du contour avant la phase d'optimisation. Ainsi dans MATTONG il existe deux

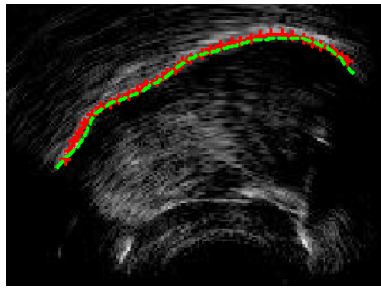


FIGURE 3 – Exemple de contours de langue issus de MATTONG (en vert) et de EdgeTrak (en croix rouge).

processus distincts : (1) un processus d'initialisation de toutes les images (2) un processus de localisation précise.

Le premier processus est semi-automatique et est réalisé sur des images dont la résolution est réduite tandis que le second est entièrement automatique et est appliqué sur les images à pleine résolution. Dans MATTONG, il est possible à tout moment d'arrêter ou de reprendre l'analyse sans effectuer aucune action explicite de sauvegarde. Après chaque action MATTONG enregistre les données. Ceci permet également de stopper l'analyse afin d'apporter des corrections manuelles directement en cliquant sur le graphique pour retirer ou ajouter des points sur le contour. Voyons à présent un peu plus en détail ces deux processus et la façon dont MATTONG gère les données.

### **Processus 1 : initialisation sur les images réduites**

Pour la première image d'une série il est nécessaire de tracer manuellement un contour approximatif de la langue en positionnant manuellement des points proches du contour de la langue directement sur l'image.

Comme le temps d'exécution de scripts MatLab est relativement long, tout le processus d'initialisation est effectué sur des images dont la résolution a été réduite. Ceci permet d'accélérer le processus et d'augmenter la zone de recherche à moindre frais. La zone de recherche est par défaut de  $\pm 3$  pixels dans l'algorithme de suivi du contour de la langue de MATTONG. Par défaut l'image est réduite de 50% ce qui permet ainsi, pour un temps de calcul équivalent, d'étendre cette zone de recherche à  $\pm 6$  pixels (théorique).

Le nombre de points définissant un contour est également réduit par rapport à l'algorithme d'origine. Par défaut ce nombre est de 18 points.

Après annotation manuelle, la position de ce contour est optimisée, c'est à dire que l'énergie de ce contour est minimisée. Pour chaque nouvelle image le contour de la langue est initialisé de manière automatique par copie du contour précédent (optimisé) sur l'image actuelle, de la même manière que dans EdgeTrak.

### **Processus 2 : Localisation**

Le processus d'initialisation peut être suffisant car il offre une approximation raisonnable de la position de la langue. Toutefois, afin d'obtenir des contours plus précis, un processus de localisation plus précise est possible. La première étape appelée initialisation nous a permis d'obtenir un contour approximatif qui sera par la suite utilisé comme contour d'initialisation pour chaque image de façon indépendante. Dans ce processus, il n'y a

plus de copie du contour précédent (optimisé) sur l'image actuelle. Ainsi, il s'agit ici d'un processus entièrement automatique. Ce contour comporte par défaut 33 points et la zone de recherche quant à elle est de +/- 2 points par défaut. Afin d'augmenter la précision des contours issus du processus d'initialisation, ici le suivi est effectué sur les images d'origine (pas de diminution de la résolution). Du fait qu'un contour d'initialisation est déjà présent et dû à la zone de recherche réduite ceci assure une optimisation efficace présentant moins de divergence lors du suivi du contour de la langue. Notons que tous les paramètres de recherche ainsi que les coefficients de pondération pour les deux processus sont réglables dans MATTONG via les menus.

### Gestion des données par MATTONG

Comme précédemment mentionné, les analyses peuvent être arrêtées et reprises à tout moment. Dans MATTONG, après chaque modification, un fichier est créé contenant les informations liées aux contours (initialisation et localisation). Ainsi, pour chaque image, après analyse, il existera deux fichiers *.mat* contenant les informations des contours. Le premier fichier ayant le même nom que l'image en question suivi du suffixe *\_init.mat* et contient les informations du contour d'initialisation. De la même manière le fichier avec le suffixe *\_track.mat* contient les informations sur le contour précis de localisation. D'autres fichiers peuvent également être présents dans le dossier où se situe les images US tel qu'un fichier *palate.mat*, dont nous parlerons plus tard, qui contient les informations liées à la position du palais. Bien que des tests ont montré que les temps d'accès sont négligeables, une solution future pourra s'orienter vers des fichiers binaires uniques.

L'intégralité de MATTONG n'a pas été codée par nosCi-dessous le détail de ces options d'affichage. soins. En effet, MATTONG utilise un certain nombre de codes sources disponibles sur internet. Pour respecter le travail de ces personnes, nous détaillons ci-dessous ces scripts en mentionnant les sources et les auteurs. Nous en profitons pour remercier l'ensemble des développeurs qui publient gratuitement leurs codes sur internet permettant ainsi d'aider et de faire avancer un grand nombre de projets.

- Interpolation B-splines : (Hunyadi, 2012)
- Filtrage d'images US : (Meshral, 2013)
- Lecture de fichiers vidéo : (Richert, 2009)
- Extraction des pixels inclus dans un polygone : (Engwirda, 2007)
- Interpolation 2D équidistante : (D'Errico, 2012)
- Clavier virtuel : (Zimmerman, 2012)
- Activation d'un objet graphique : (Weber, 2002)

## 5 MATTONG et ses fonctionnalités

Le premier objectif de MATTONG est de fournir une interface graphique conviviale incluant plusieurs outils pour l'affichage, le tracé de courbes, l'exportation des données ou encore le filtrage des images US. Dans cette partie, nous décrivons ces outils.

**Options de visualisation :** Plusieurs options (présentes en dessous de l'image dans l'interface graphique) permettent d'afficher ou non certains détails. Ci-dessous le détail de ces options d'affichage.

- *Palais* : la référence correspondant à la position du palais peut être définie et affichée par MATTONG si une image US permet de l'observer. Le menu « Palate » permet de définir et de modifier cette courbe de référence. Les points de ce contour sont enregistrés dans un fichier *palate.mat*.
- *Grille* : Il est également possible de superposer aux images US une grille angulaire en fonction de la sonde US utilisée. (Dans la version actuelle, cette grille n'est pas encore paramétrable).
- *Visibilité des contours* : pendant et après le suivi des contours de la langue, MATTONG permet de personnaliser l'affichage des contours. Ainsi, dans les options d'affichage il est possible d'afficher ou non, les points d'initialisation (en rouge), les points de localisation (en jaune) ainsi qu'une courbe interpolée par B-splines sur le contour de la localisation (en bleu), le palais ou encore la grille. Notons que ces options de visualisation sont à prendre en compte dans les procédures d'exportations décrites par la suite.

**Options d'exportation des courbes** : Plusieurs outils permettent d'exporter les données ainsi que de réaliser des copies d'écran permettant l'intégration facile d'images dans des documents scientifiques sont également disponibles.

- *Fichier image* : La première des options d'exportation des courbes est l'enregistrement de l'affichage actuel (image+contour(s)+palais+grille) directement en fichier image.
- *Figure MatLab* : Une seconde option permet de dupliquer l'affichage actuelle dans une fenêtre figure de MatLab. Ceci permet ainsi de réaliser toutes les manipulations courantes sur les figures disponibles dans MatLab (ajout de titre, modification des couleurs, légende, etc....)
- *Superposition* : Une troisième option, permet de tracer l'ensemble des contours des images sélectionnées dans les options de visualisation sur une seule figure. Cet affichage utilise un dégradé de couleurs permettant de différencier les contours des premières images et des dernières images.

**Options d'exportation des données** : Même si tous les résultats sont enregistrés dans des fichiers *.mat* propres à chaque image et chaque processus, MATTONG intègre plusieurs options pour l'exportation des données.

- *Exportation directe dans MatLab* : il est possible d'exporter directement les informations des contours ainsi que les images directement dans l'espace de travail MatLab.
- *Fichiers textes* : Une autre solution consiste à utiliser les formats d'exportation MATTONG. Ces formats sont des fichiers en format texte incluant les données des contours et les noms des images US correspondantes. Deux types de format existent nommés *long* et *short*. Le format *long* inclus des balises permettant d'avoir un format lisible par l'utilisateur. Le deuxième format contient les mêmes informations mais aucune balise n'est présente. L'utilisateur de MATTONG pourra se servir d'un script *mtg\_reader.m* fourni dans MATTONG pour lire ces fichiers dans un programme MatLab annexe.

**Options de filtrage** : Notre logiciel intègre également un outil de filtrage des images US. Ceci permet soit de réaliser un filtrage directement sur les images avant de réaliser un suivi ou d'enregistrer les images filtrées dans un nouveau dossier permettant ainsi dans certains cas de ne pas trop ralentir l'analyse. Parmi les filtres disponibles, on pourra citer les filtres médians, spectraux ou par ondelettes. Notons ici que les scripts MatLab utilisés pour le filtrage sont issus de (Meshral, 2013). Plus de détails sur ces méthodes de filtrage sont disponibles dans (Juan et Fernández-Caballero, 2009).



**Conversion de vidéo en images** : Une option très pratique est intégrée dans MATTONG permettant de convertir une vidéo en fichier images en quelques clics. En sélectionnant une vidéo puis un dossier de destination, cette fonction extrait toutes les images de cette vidéo en les nommant comme la dite vidéo suivi d'une numérotation sur 4 chiffres.

## 6 Prérequis

Notre logiciel contient des scripts codés en langage C permettant de générer des fichiers *.mex* utilisable par MatLab. Ceci permet entre autres d'accélérer certains processus coûteux en temps de calcul. Ainsi, pour pouvoir utiliser MATTONG ces scripts doivent être compilés par MatLab sur la machine sur laquelle MATTONG va être utilisé. Notre logiciel se charge de réaliser ces compilations ; toutefois il est nécessaire de disposer d'un compilateur C intégré à MatLab. Pour connaître comment installer ce compilateur et choisir une version compatible, nous renvoyons le lecteur au site web de Mathworks. Une option permettant de lancer MATTONG sans ce compilateur, au détriment du temps de calcul, est en cours de développement. Il est toutefois indispensable de disposer de la boîte à outils MatLab pour le traitement d'images (image processing toolbox). De plus, afin de profiter au maximum de l'outil de filtrage dans son intégralité la boîte à outils ondelettes est nécessaire (Wavelet Toolbox).

## 7 Conclusion

Notons que l'algorithme de suivi de MATTONG, même s'il est très proche de celui de (Li *et al.*, 2005a), ne respect pas à la lettre ce dernier. Précisons ici que MATTONG est encore en phase de développement et d'amélioration et que cet outil que nous proposons n'a pas pour but de proposer une alternative à EdgeTrak, ce dernier ayant largement fait ces preuves, mais d'apporter des fonctionnalités satellites et une interface conviviale permettant son utilisation. De plus, l'architecture du code de MATTONG offre la possibilité d'inclure assez facilement un algorithme de suivi autre que EdgeTrak. Ainsi, dans des futures versions il sera possible d'inclure différents algorithmes afin de faire de MATTONG un outil convivial et facile d'utilisation permettant la gestion d'image US. MATTONG est téléchargeable sur le site du Laboratoire de Phonétique et de Phonologie (LPP) à partir du lien suivant : [http://lpp.in2p3.fr/article.php3?id\\_article=400&var\\_mode=calcul](http://lpp.in2p3.fr/article.php3?id_article=400&var_mode=calcul)

## Remerciements et licence

Ce travail a bénéficié d'une part d'une aide de la commission européenne à travers le projet i-Treasures (Intangible Treasures - Capturing the Intangible Cultural Heritage and Learning the Rare Know-How of Living Human Treasures FP7-ICT-2011-9-600676- i-Treasures) ; d'autre part d'une aide de l'Etat gérée par l'Agence Nationale de la Recherche au titre du programme Investissements d'Avenir portant la référence ANR-10-LABX-0083.



MATTONG est mis à disposition selon les termes de la licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Partage dans les Mêmes Conditions 4.0 International.

# Références

- AKGUL, Y. S., KAMBHAMETTU, C. et STONE, M. (1999). Automatic extraction and tracking of the tongue contours. *IEEE Transactions on Medical Imaging*, 10(18):1035–1045.
- AKGUL, Y. S., KAMBHAMETTU, C. et STONE, M. (2000). A task-specific contour tracker for ultrasound. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis*, page 135–142, Hilton Head Island, South Carolina.
- AMINI, A. A., WEYMOUTH, T. E. et JAIN, R. C. (1990). Using dynamic programming for solving variational problems in vision. *PAMI*, 9(12):855–867.
- D'ERRICO, J. (2012). Interparc. <http://www.mathworks.in/matlabcentral/fileexchange/34874-interparc>. [consulté le 01/02/2014].
- ENGWIRDA, D. (2007). Fast points-in-polygon test. <http://www.mathworks.com/matlabcentral/fileexchange/10391-fast-points-in-polygon-test>. [consulté le 01/02/2014].
- HUNYADI, L. (2012). B-splines. <http://www.mathworks.com/matlabcentral/fileexchange/27374-b-splines>. [consulté le 01/02/2014].
- JUAN, L. M. et FERNÁNDEZ-CABALLERO, A. (2009). Finding out general tendencies in speckle noise reduction in ultrasound images. *Expert Systems with Applications*, 4(36):7786–7797.
- LI, M., KAMBHAMETTU, C. et STONE, M. (2005a). Automatic contour tracking in ultrasound images. *International Journal of Clinical Linguistics and Phonetics*, 6/7(19):545–554.
- LI, M., KAMBHAMETTU, C. et STONE, M. (2005b). Edgetrak - automatic contour tracking in ultrasound images. <http://http://speech.umaryland.edu/edgetrak.html>. [consulté le 01/02/2014].
- MESHRAI, A. (2013). Speckle noise reduction in ultrasound images. <http://www.mathworks.com/matlabcentral/fileexchange/41240-speckle-noise-reduction-in-ultrasound-images>. [consulté le 01/02/2014].
- RICHERT, M. (2009). Mmread. <http://www.mathworks.com/matlabcentral/fileexchange/8028-mmread>. [consulté le 01/02/2014].
- ROUSSOS, A., KATSAMANIS, A. et MARAGOS, P. (2009). Tongue tracking in ultrasound images with active appearance models. In *Image Processing (ICIP), 2009 16th IEEE International Conference on*, pages 1733–1736.
- TANG, L., BRESSMANN, T. et HAMARNEH, G. (2012). Tongue contour tracking in dynamic ultrasound via higher-order MRF and fusion moves. *Medical Image Analysis (MedIA)*, 16(8):1503–1520.
- VITERBI, A. (1967). Error bounds for convolutional codes and an asymptotically optimum decoding algorithm. *Information Theory, IEEE Transactions on*, 13(2):260–269.
- WEBER, M. (2002). Setfocus. <http://www.mathworks.com/matlabcentral/fileexchange/1898-setfocus>. [consulté le 01/02/2014].
- WRENCH, A. (2007). Articulate assistant advanced :ultrasound module. In *Ultrafest IV*. Oral Presentation.
- ZIMMERMAN, M. (2012). Have matlab type a string as if from keyboard. <http://stackoverflow.com/questions/13004695/have-matlab-type-a-string-as-if-from-keyboard>. [consulté le 01/02/2014].