



FIGURE 3.9 – Interface graphique Matlab du programme GridCat [14]. Il permet une analyse des Grids-cells à partir de données, paramètres et réglages spécifiques. Son fonctionnement repose sur l’analyse des données, deux GLM et la visualisation des résultats.

3.7 Programme GridCat

Grid Code Analysis Toolbox (GridCat) est un programme open-source développé dans le cadre d’une thèse par le Dr M.Stangl [14]. Il permet d’analyser le grid code à partir d’IRMf acquises lors d’une tâche de navigation. Il reprend toutes les étapes, de l’estimation, en passant par les deux GLM, la génération des métriques et la visualisation des résultats. L’interface qui apparaît, à son lancement dans Matlab, est visible à la figure 3.9. Les différents champs à remplir sur cette figure sont développés ci-dessous.

3.7.1 Données en entrée

Afin de procéder à une analyse spécifique à notre expérience, il faut introduire différentes données spécifiques à chaque run. L'interface du programme étant en anglais, les différentes variables le seront également.

- **Functional scans** : Les images fonctionnelles doivent être fournies sous le format Nifti. Chaque run est caractérisé par une série de Nifti 3D par volume. Cela implique un changement de format car les images coregistrées sont stockées sous forme de fichier VTC. Ceux-ci seront donc transformés en plusieurs 3D-Nifti. Le preprocessing ainsi que la normalisation MNI doivent être réalisés avant l'utilisation de GridCat.
- **Event-table** : Fichier texte (.txt) décrivant les différents événements du jeu, qui sont synchronisés avec l'enregistrement de l'imagerie fonctionnelle. En effet, chaque ligne correspond à un événement. Ce dernier est caractérisé par un type (translation, rotation,...), un temps de départ, une durée et l'angle de direction du joueur pendant cette période. Un exemple d'Event-table spécifique à la tâche de ce mémoire est visible dans la section 4.2.
- **Additional regressors** : Cette entrée est optionnelle et permet d'ajouter des régresseurs supplémentaires, qui seront pris en compte dans le GLM. Par exemple, un fichier .txt comprenant les paramètres de réaligement des images fonctionnelles peut être inclus.

3.7.2 Spécifications

GLM data directories

Deux dossiers de sortie doivent être définis, un pour chaque GLMs. C'est dans ces dossiers que seront stockés les différents régresseurs calculés lors de l'exécution des modèles.

Scanning parameters

Le temps de répétition (TR) spécifique aux données doit être renseigné en secondes dans la case *Interscan Interval*. Il est également possible de spécifier d'autres paramètres, par exemple, pour l'application d'un filtre passe-haut.

Grid code symmetry

Cette section permet de sélectionner différentes symétries à étudier. La symétrie par défaut, est celle des Grids-cells, la sextuple. Ce paramètre permet de mettre en évidence leur présence ou non, car en comparant différentes symétries sur un

même ensemble de données, la sextuple doit être plus présente en cas d’activation des Grids-cells.

3.7.3 GLM1

Afin de générer le premier modèle linéaire généralisé, il faut décider comment seront séparés les données en un ensemble d’estimation et de test. Pour cela, différents choix sont possibles : utiliser la première ou la deuxième moitié ; les événements pairs ou impairs ou uniquement les runs pairs ou impairs.

Les événements sélectionnés pour le GLM1 peuvent être inclus (via Include unused grids events) ou non dans l’estimation sans leur angle correspondant, ce qui implique qu’ils ne contribueront pas à l’estimation de l’orientation de la grille.

Après avoir fixé les différents paramètres, le GLM1 peut être exécuté. L’orientation de la grille par voxel est estimée et les régresseurs sont calculés et sauves dans le dossier GLM1. Il est important de noter que les voxels dont la valeur est *NaN* (*Not a number*), sont ceux pour lesquels l’orientation de la grille n’a pas pu être estimée.

3.7.4 GLM2

ROI mask

La région d’intérêt peut être sélectionnée à l’aide d’un fichier Nifti contenant un masque de la région en question. A noter que ce masque doit être binaire et de mêmes dimensions que les images de la fonctionnelle. Après avoir fixé le masque, l’orientation moyenne de la grille dans la ROI est calculée. Dans le cadre de ce mémoire, c’est à ce moment que le masque pmEC ou alEC est utilisé.

Grid code regressors

Différents types de régresseurs sont disponibles dans le programme. Ils sont tous en lien avec l’orientation moyenne de la grille estimée avec le GLM1. Cependant, ils diffèrent en ce qui concerne les comparaisons avec l’orientation moyenne ou les événements. Les trois types de régresseurs sont :

- **Parametric modulation regressor** : Les angles des événements sont modélisés via leur alignement par rapport à l’orientation moyenne. Le régresseur varie de -1 (opposition) à 1 (alignement).
- **Aligned/misaligned regressors (1 each)** : Contrairement au Parametric, ce régresseur est binaire. En effet, chaque angle d’événement est comparé à l’orientation moyenne. Si la différence n’excède par 15° par rapport à celle-ci

ou à un multiple de 60° , l'angle est considéré comme *Aligned* ou *Misaligned* sinon.

- **Aligned/misaligned regressors (multiple)** : Le principe est le même que pour le régresseur "1 each" mise à part que dans ce cas-ci, la condition alignée sera représentée par plusieurs régresseurs qui sont les multiples de $60^\circ(0, 60^\circ, \dots)$. Ces derniers correspondent aux pics de signal attendus lors d'une symétrie sextuple. Dans le même raisonnement, la condition non-alignées sera modélisée par les angles décalés ($30^\circ, 90^\circ, \dots$) correspondant à un signal faible.

Grid events

Cette sélection d'événements est généralement automatique car elle contient la partie non-utilisée dans le GLM1.

3.7.5 Outils

La dernière partie du programme permet d'illustrer les différents régresseurs et résultats calculés lors des deux GLMs. Il est notamment possible d'illustrer l'orientation calculée pour chaque voxel ou encore leur stabilité d'orientation entre deux runs. Différents graphiques sont présentés dans la section 4.3.