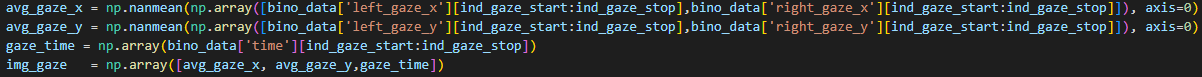
• le travail effectué

# Avancement

Lors du commencement de notre projet nous nous sommes intéressés principalement à la méthode de détection des fixations par fixation : I-DT. Pour cela, les programmes python déjà présents lors de notre arrivés, ont été utilisé comme base de notre travail. Ces algorithmes nous ont ainsi, pour l’instant, permis de développer plusieurs méthodes de détection par dispersion. Nous sommes donc arrivés à l’écriture de deux fichiers python fonctionnels : « dispersion\_map\_visualization.py » (Annexe 1)  et « Methode\_I\_DT.py » (Annexe 2)

En effet, la lecture des fichiers de données pour effectuer les méthodes suivantes, ont été développé grâce aux codes permettant d’afficher les coordonnées brutes du regard de l’utilisateur et la heat map. Cependant outre les coordonnées lues dans le fichier h5df, les données temporel sont en plus extraites et stockées dans un tableau Numpy (figure 1). Ces tableaux Numpy nous permettront par la suite d’effectuer les calculs de dispersion pour chaque image selon les deux méthodes I-DT et I-VT.

## Identification par dispersion

Figure Extraction des coordonnées X et Y, et des indices temporels de ces points dans des tableaux Numpy

Le travail dans cette partie fut donc de coder avec python un programme permettant :

- de repérer des fixations par dispersion (coordonnées et durées)

- d’analyser ces fixations (quelles sont les mouvements les précédant : saccade ou micro-saccade)

- d’afficher sur l’image correspondante ces fixations

- d’enregistrer ces fixations ou saccades dans un data-Frame (data-frame alors ensuite exporté dans un fichier Excel ou csv)

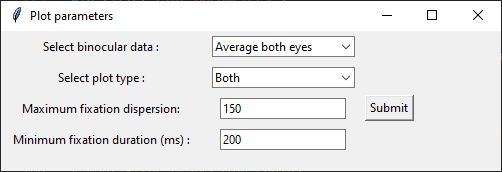
Pour cela nous avons crée le fichier « dispersion\_map\_visualization.py » en s’inspirant du fichier : « image\_gaze\_visualization.py ». Cependant le programme a du être adapté. Tout d’abord la boîte de dialogue (figure 2) s’ouvrant à l’ouverture du programme demande maintenant à l’utilisateur les paramètres de la dispersion étant : la dispersion maximum en pixel et sa durée minimale.

Figure Boîte de dialogue permettant à l'utilisateur de définir les options d'analyse des données de l'oculomètre

Par la suite il a été nécessaire d’adapter le programme pour qu’il puisse fonctionner sous différents systèmes d’exploitation : MacOs et Windows (changement de la syntaxe pour accéder aux fichiers) en utilisant deux séparateurs différents (‘/’ pour MacOs et ‘\’ pour Windows). De plus la structure du programme a été changé puisque le programme « image\_gaze\_visualization.py » balayait tout les point par image puis par fichiers, conditions nécessaire puisque la heat-map nécessite les données d’au moins deux participants. Le programme ré-ouvrait donc pour chaque image deux fichiers HDF, rendant le programme non optimisé pour ce que l’on souhaite en faire. Ainsi, notre code parcours pour chaque fichier HDF, correspondant à un participant, les images qu’il a regardés.

Puis le fichier « dispersion\_map\_visualization.py » inclus de nouvelles fonctions (affichage des graphes et enregistrements des data frames), ainsi qu’un nouveau fichier python : « Methode\_I\_DT.py ». Ce nouveau fichier que nous avons écrit permet de réaliser nos calculs de dispersion en utilisant les fonctions définis dans celui-ci. Ainsi notre programme suit le fonctionnement logique expliqué figure 3.

**Choix des paramètres de l’analyse par l’utilisateur** (affichage du graphe de fixations et/ou des données bruts, paramètres des fixations..).

**Boucle pour chaque fichier HDF** sélectionné au début du code (par défaut seul le fichier « ExploIMG\_PupilCore\_m\_001.hdf5 » est utilisé).

**Boucle pour chaque image** utilisée dans le test avec l’oculomètre.

**Extraction des données récupérées par l’oculomètre** en fonction des images et du fichier étudiés dans un tableau Numpyy.

**Envoie des tableaux Numpy contenant les données bruts** ainsi que les **paramètres sélectionnés par l’utilisateur** à la fonction « **Choix\_Methode\_Dispersion.py** » contenu dans le fichier « Methode\_I\_DT.py ».

**Récupération des listes, contenant les informations sur les saccades et les fixations,** retournées par la fonction appelée il y a quelques instants.

**Affichage des fixations** à l’aide des fonctions mathplotlib.

**Enregistrement des informations** reçues dans un tableau Numpy, puis dans un data Frame et enfin dans un xlxs.

Figure Pseudo-Code pour l'algorithme d'analyse des fixations

Vient alors les fonctions apportées par le fichier « Methode.py ». Ces fonctions permettent de calculer la dispersion pour chaque point des coordonnées du fichier HDF selon la méthode sélectionnée.

### Méthode Euclidienne

Première méthode abordé afin de reconnaître les dispersions, celle-ci s ‘intéresse à la distance euclidienne entre le premier point considéré et les points suivants. Si cette distance ne dépasse pas le seuil de dispersion maximum et que la fenêtre de point étudiée dépasse la durée minimale d’une fixation, alors on considère une fixation. Cependant cette méthode comporte des inconvénients et ne permet pas à l’heure actuelle de détecter les micro-saccades. Par ailleurs il sera plus intéressant par la suite, de comparer les distances par rapport au centre de gravité de nuage de point déjà traité que par rapport au premier point de la fenêtre en question (figure 4 & 5).

Figure 4 Calcul de la dispersion par la méthode euclidienne actuelle

x

y

X1,Y1

Di

Figure 5 Calcul de la dispersion par la méthode euclidienne future

x

y

CG

Di

De plus entre chaque fixation on attend une saccade, a qui l’on attribue un seuil de temps d’environ 40ms. La saccade n’a pas besoin d’être détecté puisque tout ce qui n’est pas fixation est considéré comme saccade.

### Méthode Salvucci, D.D. et Goldberg, J.H.

Cette méthode-ci, est la méthode la plus aboutie puisque on peut grâce à nos codes analyser et enregistrer les informations des différents mouvements suivants :

* Fixations
* Micro-Saccade (le regard se pose sur en endroit éloigné de la fixation mais y revient après seulement quelque millisecondes)
* Saccades

La méthode Salvucci et Goldberg consiste à détecter les fixations grâce à l’équation suivante :

Cette valeur de dispersion que l’on calcule sur une fenêtre de point choisie nous permet de définir, selon sa valeur, si l’on a affaire à une suite de saccade ou une fixation. Le seuil de dispersion que l’on a choisie est en pixel et est définie comme paramètre saisie par l’utilisateur. Cependant une valeur par défaut proposé est :

Ainsi le programme fonctionne de la manière suivante :

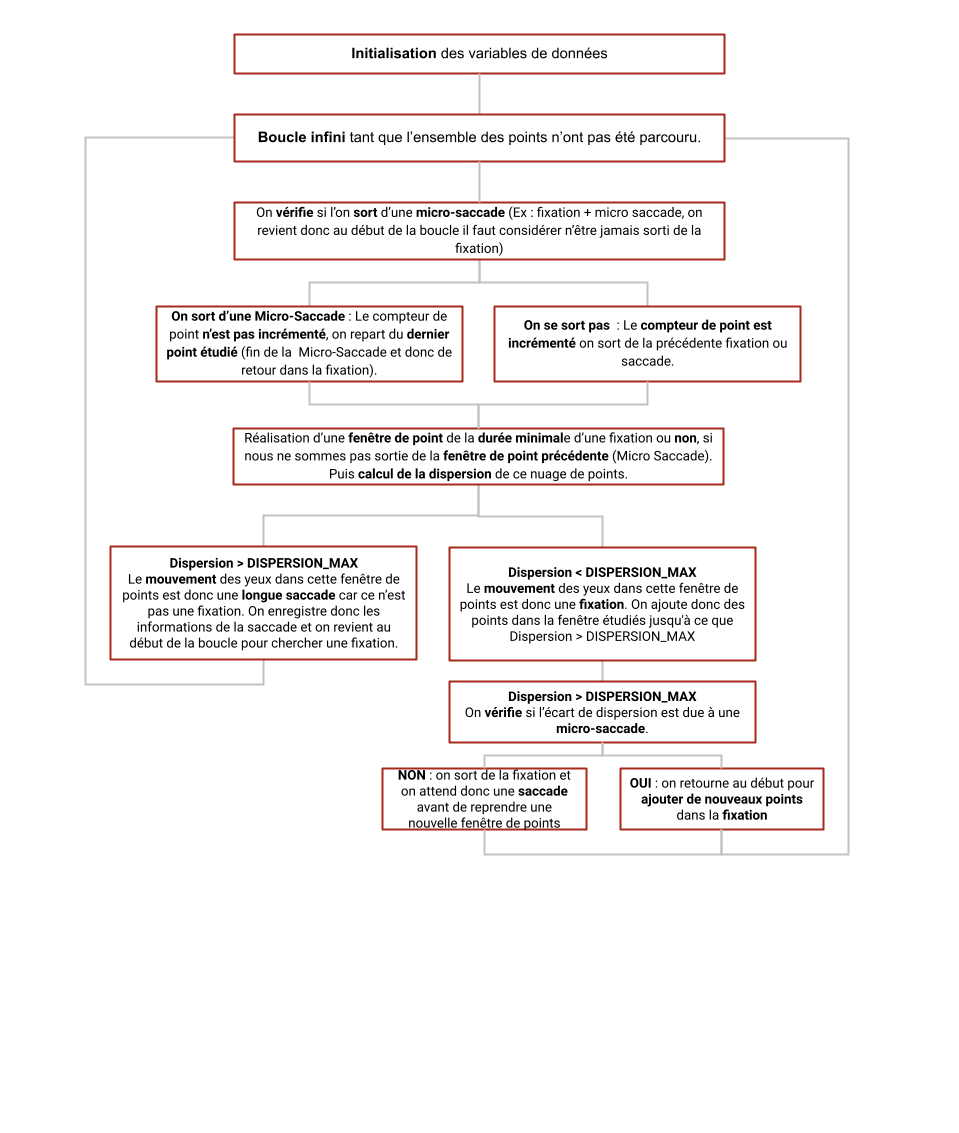


Figure Représentation logique du programme de détection des fixations et saccades

## Exports des données

Afin de sauvegarder les valeurs de fixation et de saccades nous avons choisies de réaliser des DataFrame de la forme suivante :

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | **INFORMATIONS** | **Fixation\_x (px)** | **Fixation\_y (px)** | **rayon (px)** | **Time Start (s)** | **Duration (s)** |
| **0** | m n° 001 | 27 | 0 | 51,1627388 | 64,51167297 | 0,299863994 |
| **1** | img\_1.jpg | 93 | -100 | 57,85558701 | 64,85163879 | 0,811013997 |
| **2** |  | -2 | -33 | 53,24966431 | 65,70362091 | 0,462042511 |
| **3** |  | 110 | 90 | 51,72835159 | 66,40760803 | 0,340027004 |
| **4** |  | 8 | 71 | 56,54074478 | 66,78765869 | 0,72794199 |
| **5** |  | 107 | 70 | 51,78007889 | 67,55755615 | 0,350091994 |
| **6** |  | 12 | -85 | 53,62353516 | 67,9515152 | 0,496073514 |
| **7** |  | -127 | -88 | 53,35621643 | 68,49163818 | 0,472258508 |
| **8** |  | 45 | 34 | 67,75315857 | 69,00756836 | 1,474084973 |
| **9** |  | 27 | -97 | 59,97525406 | 70,52758026 | 0,956054509 |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Type** | **X\_start (px)** | **Y\_start (px)** | **X\_end (px)** | **Y\_end (px)** | **Time Start (s)** | **Time End (s)** | **Duration (s)** |
| **0** | Normal | 119 | -52 | 44 | -73 | 53,23162842 | 53,27566528 | 0,044038001 |
| **1** | Normal | 47 | -69 | -110 | -53 | 53,27154541 | 53,47562027 | 0,204077497 |
| **2** | Normal | 9 | -100 | -14 | 104 | 54,01604462 | 54,05765152 | 0,041607499 |
| **3** | Normal | -34 | -55 | -67 | -127 | 54,81957626 | 54,86157227 | 0,041995998 |
| **4** | Normal | -34 | 51 | -5 | 124 | 55,94355011 | 55,98361588 | 0,040068001 |
| **5** | Normal | 90 | -31 | 52 | 41 | 56,76163864 | 56,80354309 | 0,041905001 |
| **6** | Micro | 85 | 107 | 84 | 109 | 57,29579163 | 57,29965973 | 0,0038685 |
| **7** | Micro | 74 | 110 | 76 | 107 | 57,30368805 | 57,30768967 | 0,0040025 |
| **8** | Micro | 80 | 109 | 80 | 107 | 57,3117981 | 57,31578445 | 0,0039855 |
| **9** | Micro | 80 | 106 | 81 | 103 | 57,3196373 | 57,32362366 | 0,0039885 |
| **10** | Micro | 85 | 107 | 84 | 104 | 57,32758331 | 57,33161163 | 0,0040265 |
| **11** | Normal | 87 | 104 | 88 | 108 | 57,33960342 | 57,38351059 | 0,0439055 |
| **12** | Normal | 88 | 112 | 122 | -13 | 57,37959671 | 57,58359146 | 0,203994006 |
| **13** | Normal | 123 | 119 | -124 | 92 | 58,08774567 | 58,13164139 | 0,0438965 |
| **14** | Normal | 42 | 36 | 52 | 38 | 58,50365067 | 58,54961395 | 0,045965999 |
| **15** | Normal | -128 | 83 | 11 | -10 | 59,18439102 | 59,22768021 | 0,043288998 |
| **16** | Normal | 88 | 108 | 121 | -4 | 57,38351059 | 57,58760452 | 0,204095498 |
| **17** | Normal | 123 | 119 | -124 | 92 | 58,08774567 | 58,13164139 | 0,0438965 |
| **18** | Normal | 42 | 36 | 52 | 38 | 58,50365067 | 58,54961395 | 0,045965999 |
| **19** | Normal | -128 | 83 | 11 | -10 | 59,18439102 | 59,22768021 | 0,043288998 |

Les données des saccades sont stockées dans un excle

## Observation

Enfin, dans le but d’observer et d’analyser nos résultats, deux solutions ont été choisies :

* La première est d’afficher les tracés X et Y en fonctions du temps où l’on peut donc observer pour chaque palier présents simultanément en X et Y une fixation (figure 7).
* La seconde est d’afficher les fixations sous forme de cercle avec pour rayon la valeur de dispersion ; et comme centre, le centre de gravité de la fenêtre de point constituant la fixation (figure 8).

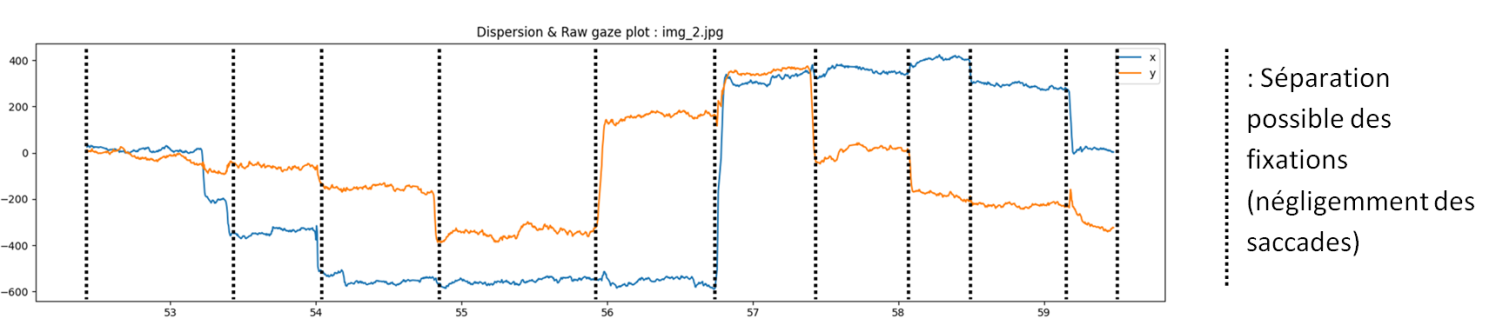


Figure 7 Affichage des coordonnées X et Y selon le temps pour l'image n°2

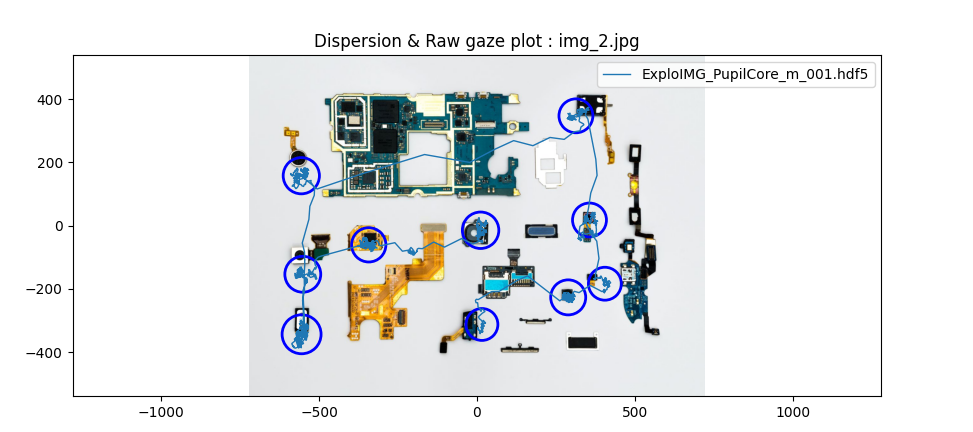


Figure 8 Affichage des fixations sous forme de cercle