

Rapport groupe 7 - Point Of Regard

Louis Arys Anh-Emile Pham Edwin Automne Filipp Shatskiy
Melvin Campos Casares

9 décembre 2020

Traitement des Signaux et données - 3TI - EPHEC 2020/2021

Table des matières

1	Présentation	2
2	Réalisation pratique	2
2.1	Méthodologie	2
2.2	Organisation	2
2.3	Implémentation	3
3	Pistes d'amélioration	5
3.1	Possibilité de choisir la vidéo et le fichier de données	5
3.2	Interface graphique simplifiant l'utilisation du programme	5
3.3	Heatmap de différentes couleurs	5
3.4	Détection des mouvements plus conséquents en un court instant	5
3.5	Remplacement du heatmap pour afficher les endroits ou le sujet à principale- ment regardé	5
4	Conclusion	6
5	Conclusions personnelles	6
5.1	Louis Arys	6
5.2	Anh-Emile Pham	6
5.3	Edwin Automne	6
5.4	Filipp Shatskiy	7
5.5	Melvin Campos Casares	7
6	Sources	8

1 Présentation

Le projet Point of Regard concerne le suivi du regard et sa représentation sur une vidéo regardée par une personne. L'objectif est d'intégrer à la vidéo les données récupérées via un casque électroencéphalogramme afin d'indiquer les zones où le regard de la personne s'est attardé et ainsi comprendre ce qui peut, potentiellement, attirer notre intérêt et regard.

2 Réalisation pratique

La vidéo ainsi que la base des données oculaires ont été fournies.

Il est important de noter qu'il nous a été nécessaire de traiter les données. En effet, elles ont été reçues en format *.TXT* alors que le contenu respectait la norme du format *.CSV*.

Nous étions libres de choisir notre environnement de travail. Ne sachant pas quel environnement aurait été le plus apte, nous avons commencé par travailler en scindant l'équipe afin d'avancer en parallèle sur Python et Matlab. Très rapidement, nous avons jugé de l'intérêt du langage de programmation Python et avons travaillé avec ce langage.

2.1 Méthodologie

Nous avons travaillé via pair-programming afin de subdiviser les tâches entre nous tout en organisant des réunions à intervalles réguliers via Discord, avec rappel des réunions via un groupe de discussion sur Facebook Messenger.

Nous avons choisi Discord comme moyen de réunion car nous partageons nos écrans aisément par ce biais et profitons d'un canal de discussion entièrement dédié au projet, nous semblant plus simple d'emploi que Messenger.

Notre méthodologie de travail s'apparente donc à de l'Agile/Scrum.

2.2 Organisation

Nous avons tout d'abord procédé à la recherche de documentation afin de voir plus concrètement comment réaliser le projet.

Nous avons ensuite scindé le groupe afin de travailler sur Python et Matlab, visualiser lequel serait le plus approprié pour ce projet en réalisant une lecture rapide de la vidéo et quelques manipulations simples à mettre en place.

Melvin et Filipp ont tout d'abord travaillé sur la lecture du fichier contenant les données et en extraire les informations utiles en format JSON. Pendant ce temps Louis et Emile ont

travaillé sur la lecture du fichier vidéo sous Python. Edwin à quant à lui travaillé sur la lecture du fichier vidéo sous Matlab.

Ensuite, Melvin et Filipp ont réalisé, en pair-programming, le traitement des données récoltées afin d’y effectuer une moyenne. Louis et Emile ont avancé sur l’affichage d’un point représentant le point de regard sur une vidéo. Edwin à vérifié la lecture des données et l’affichage de ce dernier via Matlab avant de comparer cela avec Louis et Emile.

Nous avons déduit que Python étant plus intéressant que Matlab, nous nous sommes ensuite focalisé sur Python uniquement et Edwin a procédé via pair-programming avec Louis et Emile.

Après l’entrevue avec *Mathieu Petieau* le 24 novembre à 9h30 via Microsoft Teams, nous avons eu quelques détails complémentaires afin de guider au mieux notre travail.

Melvin et Filipp ont finalisé l’exportation de la moyenne des données par seconde pour ensuite s’attaquer à une réadaptation par paquet de 20 (*principalement*) et 19 données afin de correspondre au nombre d’images par seconde de la vidéo. Louis, Emile et Edwin ont intégré la lecture du fichier JSON exporté afin que le rendu vidéo puisse se faire correctement.

De là nous avons ensuite travaillé par réunions Discord tous ensemble afin de finaliser notre projet au mieux.

2.3 Implémentation

Au niveau du choix d’implémentation, nous avons scindé le traitement et le formatage des données du traitement vidéo avec inclusion du heatmap.

Dans la partie du formatage des données, nous avons d’abord formaté les données en JSON, format standardisé et facilement utilisable par différents langages de programmation. Ceci nous permet de parcourir plus aisément les données utiles que nous avons extraites afin d’en faire un calcul de moyenne et intégrer cela dans le traitement et exportation vidéo.

Ensuite, nous avons fait une moyenne des données qu’on a séparé par des paquets de 20 (nombre total de données / nombre de frames de la vidéo). Ceci nous permet non seulement d’avoir qu’un seul échantillon de données par paquet, mais aussi de diminuer légèrement les saccades dont le nombre était quand même considérable. Mais cela restait néanmoins insuffisant, donc nous avons rajouté un filtre Savitzky-Golay. L’avantage de ce filtre par rapport à un filtre moyenneur classique est qu’il permet de faire une moyenne plus précise, en limitant les erreurs dû justement au moyennage des valeurs. Cela est utile si on veut voir avec plus de précisions les différents endroit regarder par le sujet. Pour lisser une dernière fois les données, nous les interpolons avec une spline du cinquième degré.

Pour finir, nous avons utilisé la librairie Open-CV 2 pour lire la vidéo et récupérer les images une par une, afin de pouvoir les traiter à l'aide de la librairie Matplotlib en plottant les points sur les images et puis afficher à nouveau celles-ci avec Open-CV 2.

Concernant la heatmap, nous avons créé une image composée de pixels blancs grâce à la fonction "ones", qui en parallèle avec la vidéo, rassemblait tous les points de données les uns à la suite des autres. Ceci nous a permis de voir des zones par où les points passaient le plus durant la vidéo. En effet, ils n'indiquaient p

3 Pistes d'amélioration

3.1 Possibilité de choisir la vidéo et le fichier de données

Nous avons intégré cette fonctionnalité pour le traitement des données. L'intérêt serait donc de pouvoir utiliser le programme, quelque soit les fichiers sources.

Il suffirait de les mentionner via un paramètre entré au lancement du programme, comme nous l'avons fait pour le traitement des données.

3.2 Interface graphique simplifiant l'utilisation du programme

Une possible piste d'amélioration serait la mise en place d'une interface graphique (*GUI*) permettant une utilisation plus simple et plus intuitive.

3.3 Heatmap de différentes couleurs

Mise en place d'une heatmap plus précise avec plus de couleurs. Cela permettra, par exemple, de pouvoir mieux différencier le temps que le sujet passe à regarder une zone.

3.4 Détection des mouvements plus conséquents en un court instant

Mise en place d'un système qui permet de détecter les moments où l'observateur cherche ou suit du regard quelque chose en particulier dans la vidéo.

3.5 Remplacement du heatmap pour afficher les endroits où le sujet à principalement regardé

Mise en place d'un système qui permet de détecter les points d'intérêt qu'observe le sujet dans la vidéo. Avec cette information, il serait possible de mieux visualiser et mieux comprendre ce qui peut attirer le regard du sujet.

4 Conclusion

Nous sommes arrivés à un projet fonctionnel et affichant la zone de regard du sujet sur une vidéo exporté. Nous considérons que le projet est donc concluant même si nous aurions voulu avoir un peu de temps supplémentaire afin de développer l’une ou l’autre piste d’amélioration.

Ce projet nous a également permis de voir la puissance derrière Matlab et Python puisque nous les avons chacun comparés au départ. De plus, travailler sur un projet de la sorte nous a permis de nous ouvrir à une façon de travailler nouvelle pour certains membres du groupe.

5 Conclusions personnelles

5.1 Louis Arys

Le début du projet fût assez laborieux, car le manque de clarté des données nous a demandé pas mal de recherche avant de savoir lesquelles étaient intéressantes et lesquelles ne nous seraient pas utiles. Mais une fois passée cette difficulté, le projet est devenu beaucoup plus agréable. J’ai bien aimé de pouvoir imprimer les données sur la vidéo. Ce qui était le plus surprenant, c’était qu’on pouvait (une fois le lissage fait), voir la personne lire les différents morceaux de textes présent dans la vidéo.

Pour ce qui est de l’utilisation de Python dans le projet, cela m’a permis de remettre à jour mes connaissances en Python, et de découvrir différentes bibliothèques mathématiques que je n’avais jamais utilisées.

5.2 Anh-Emile Pham

Ce projet de traitement de signal fut une expérience très enrichissante pour moi, car nous avons dû nous essayer à une technologie qui nous était inconnu et que l’on avait jamais vu en cours, oculométrie (Eye-tracking). L’apprentissage du Python pour la réalisation de ce projet fut pour le moins intéressante et m’a permis de découvrir que je n’appréciais pas les langages non-typé.

5.3 Edwin Automne

Les deux éléments les plus compliqués de ce projet furent de coupler la charge de travail qu’il représente à celle déjà présente avec les autres cours et projets, ainsi que sa réalisation en Python, un langage de programmation que je n’avait jamais pratiquer et qui ne m’avait jamais attiré tant que cela à cause de son auto-typage et de son écriture basée sur l’indentation plutôt que des parenthèses et des points-virgules.

Malgré cela, le projet fut très intéressant et l’utilisation du pair-programming a fortement améliorer la qualité et le plaisir de réalisation du travail.

5.4 Filipp Shatskiy

J'ai, personnellement, trouvé ce projet très fructueux. En effet, toute l'équipe était toujours présente aux rendez-vous fixés, faisait le travail demandé à temps, était très positive et on n'hésitait pas à s'entraider dès que quelqu'un était coincé dans son développement.

Ce que j'ai trouvé de plus intéressant dans ce projet, était le fait qu'on n'arrêtait pas de rencontrer des problèmes assez intéressants à résoudre au fur et à mesure qu'on avançait dans le développement.

Nous avons bien sûr trouvé quelques difficultés comme par exemple au niveau de l'intégration des points dans la vidéo, au niveau de la décomposition de la vidéo en images ou encore au niveau de la "heatmap". Finalement, j'ai réalisé que beaucoup de fonctionnalités, encore méconnues par moi même, étaient réalisables grâce à nos yeux. C'était vraiment très passionnant de découvrir encore un milieu dont je ne connaissais pas en informatique.

5.5 Melvin Campos Casares

J'ai déjà eu l'occasion par le passé de travailler dans des projets respectant les techniques Agile en intégrant les Scrum meeting, mais l'intérêt du pair-programming me semblait quelque peu flou jusqu'à ce projet.

Ayant quelques affinités avec Python, c'était l'occasion d'approfondir un peu ce langage, visualiser un aspect plus mathématique/scientifique et également remarquer les limitations dues à un langage de programmation non typé.

Avec Filipp, nous avons rencontré quelques difficultés liées au traitement des données et le calcul des moyennes mais avons fait notre possible pour travailler en équipe, chercher dans la documentation ce qui pouvait nous être utile et nous avons finalement parvenu à atteindre notre objectif.

6 Sources

1. *PyGaze - Open-source toolbox for eye tracking in Python*, visité le 03/11/2020, <http://www.pygaze.org/>
2. *PyGaze Analyser*, visité le 03/11/2020, <http://www.pygaze.org/2015/06/pygaze-analyser/>
3. *iView X Manual- 8.2.2 Data section*, visité le 03/11/2020, https://psychologie.unibas.ch/fileadmin/user_upload/psychologie/Forschung/N-Lab/SMI_iView_X_Manual.pdf
4. *Matplotlib - Documentation*, visité le 10/11/2020, <https://matplotlib.org/contents.html>
5. *Visualizing Gaze Data*, visité le 10/11/2020, <https://vpixx.com/vocal/visualizing-gaze-data/>
6. *ScyPi - Documentation*, visité le 12/11/2020, <https://docs.scipy.org/doc/scipy/reference/>