



Arrington Research head- fixed EyeTracker : A user guide

By

Anas EL KHALOUI, Agrocampus Ouest
Food technology department, HCMUT
Tutor: Mr. Nguyen Hoang Dzung



TABLE DES MATIERES

I.	Introduction	2
II.	Installation et réglage du materiel	3
1.	Description de l'appareil	3
2.	Besoins en matériel.....	3
3.	Brancher un deuxième écran sur l'ordinateur.....	5
4.	Disposition des caméras et de l'écran.....	5
5.	Brancher la machine sur l'ordinateur	6
III.	Installation, paramétrage et prise en main du logiciel ViewPoint EyeTracker	6
1.	Installation	6
a)	Configuration recommandée.....	6
b)	Installation des logiciels d'acquisition et d'analyse de données.....	7
c)	L'interface ViewPoint.....	7
2.	Paramétrage	9
a)	Configuration des cameras.....	9
b)	Modes: monoculaire et binoculaire.....	9
IV.	Déroulement des tests.....	10
1.	Avant de commencer	10
2.	Réglages préliminaires	11
3.	Calibration	12
4.	Enregistrement.....	13
a)	Lancer et stopper l'acquisition.....	13
b)	Structure de données	14
5.	Analyse des données.....	15
a)	Objectifs de l'analyse.....	15
b)	Utilisation du logiciel 'ViewPoint Data Analysis'	16
c)	Autres.....	17
V.	Annexes :.....	25
1.	Annexe 1 : Nom et signification des variables	25

I. INTRODUCTION

Ce document a pour ambition de servir de guide de mise en service et d'utilisation. Il sera aussi clair et précis que possible et restera très concret.

Nous disposons d'une machine de marque **Arrington Research**, se composant de 2 caméras de grande précision et d'un dispositif **HeadLock™ Ultra Precision Head Positioner™** permettant de garder la tête du sujet fixe durant l'expérience ; Ses yeux sont en effet supposés fixes par le logiciel pendant qu'il enregistre, d'où l'importance d'immobiliser la tête.



Figure 1 Vue générale de l'appareil

La machine est à la base destinée aux professionnels de la recherche dans plusieurs secteurs, touchant surtout à la santé : Electroencéphalographie, psychologie expérimentale, psycholinguistique, interfaces Homme-Machine, etc. mais nous voulons nous en servir ici en tant que source d'information à visée marketing : qu'est ce qui est attirant pour l'œil du consommateur et qu'est ce qui l'est moins ?

II. INSTALLATION ET REGLAGES DU MATERIEL

1. DESCRIPTION DE L'APPAREIL

L'appareil se compose de 2 caméras industrielles de très haute précision équipées chacune d'une lampe infrarouge, ainsi que d'une structure en métal et plastique qui se fixe sur une table et sert de support aux caméras et à la tête du sujet. Les positions de tous les éléments de la structure sont réglables à l'aide d'un ensemble de vis.



Figure 2 Les caméras, avec les lampes à infrarouges sur le côté.

L'appareil nécessite un ordinateur pour fonctionner et le logiciel est construit de manière à tourner avec deux écrans, le premier servant de panneau de commande, et le second servant à afficher les stimuli pour le sujet.

2. BESOINS EN MATERIEL

Afin de pouvoir faire marcher la machine, il faut :

- Un ordinateur de bureau avec une carte graphique et 2Go de RAM minimum.
- Deux moniteurs (ou un moniteur et un vidéoprojecteur avec son écran, si l'on veut travailler sur une grande surface d'affichage).

Attention : Bien vérifier les ports présents sur l'ordinateur et leur compatibilité avec les écrans disponibles. Il existe en effet 2 types de connectiques (VGA et DVI), et il faut parfois s'équiper d'un adaptateur.



Figure 3 Branchement des écrans sur la carte graphique. On utilise un adaptateur VGA->DVI pour brancher le 2nd écran

- Un hub USB 6 ports, autoalimenté. En effet, il faut un port USB par caméra et un second par lampe infrarouge. Nous aurons également besoin d'un clavier et d'une souris.



Figure 4 Hub USB 7 ports, autoalimenté (Se branche sur secteur)

- Un bureau calme et avec une lumière constante. Les yeux du sujet ne doivent pas être directement exposés à une source lumineuse (soleil par exemple) et l'on préférera les lampes à néon. Nous aurons également besoin d'un bureau situé idéalement face à un mur blanc ne comprenant aucune source de distraction pour le sujet.



Figure 5 L'environnement mis en place pour les expériences.

3. BRANCHER UN DEUXIEME ECRAN SUR L'ORDINATEUR

Nous avons besoin de deux écrans afin d'utiliser l'EyeTracker. Une fois les deux branchés et reconnus par l'ordinateur, il faut configurer un « Bureau étendu » (Extended desktop). Pour ce faire, sous Windows 7, il faut faire un clic droit sur le bureau puis sur *Résolution d'écran*. On voit alors les moniteurs connectés à son ordinateur qui portent les numéros d'identification **1** et **2**. On peut alors changer l'option *Affichage* Pour passer en mode *Bureau étendu*.

A présent, le bureau Windows s'étend sur 2 écrans et on utilise la même souris et le même clavier pour les deux. On peut par ailleurs afficher deux fenêtres différentes avec cette configuration matérielle simplement en faisant glisser une fenêtre de l'écran principal sur le second à l'aide de la souris.

4. DISPOSITION DES CAMERAS ET DE L'ECRAN

Le champ visuel du sujet ne doit en aucun cas être obstrué par les caméras, qui doivent se trouver en dessous de la ligne de vision. Il faut également surélever

l'écran en le posant sur un support stable de manière à ce que son centre se trouve au milieu de la ligne de vision du sujet lorsque la tête de celui-ci est fixée.

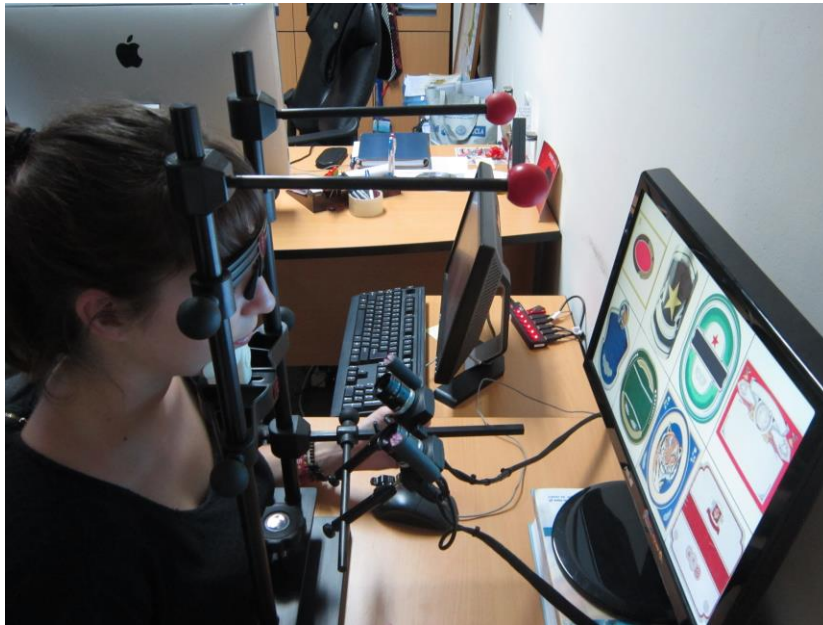


Figure 6 Positionnement du sujet durant les tests.

5. BRANCHER LA MACHINE SUR L'ORDINATEUR

Il suffit ici de brancher les 4 câbles USB de la machine (deux par caméra) sur le hub USB que l'on branche sur secteur (les caméras et les lampes ont besoin d'une alimentation propre). On connecte ensuite le hub à l'ordinateur.

III. INSTALLATION, PARAMETRAGE ET PRISE EN MAIN DU LOGICIEL VIEWPOINT EYETRACKER

1. INSTALLATION

A) CONFIGURATION RECOMMANDEE

ATTENTION : Les systèmes d'exploitation 64bit **ne sont pas supportés**. Seules les versions 32bit de Windows XP, Vista et Seven sont compatibles. Certains modèles d'ordinateurs de la marque DELL ne sont pas supportés à cause du BIOS installé par le fabricant (Modèles Optiplex et Dimension).

L'affichage doit être réglé sur 'True Color' (32bit), via les paramètres du gestionnaire d'affichage de la carte graphique. De même, il est important de régler

la résolution de l'écran d'affichage de manière optimale, le mieux étant d'opter pour la résolution recommandée. Il faut également veiller à ce que l'affichage soit neutre et qu'aucun réglage spécial ne soit appliqué (du type contraste ou luminosité augmentés) afin de ne pas apporter de biais visuel. Il vaut donc mieux réinitialiser les réglages à partir des boutons de l'écran (Reset) avant de commencer.

B) INSTALLATION DES LOGICIELS D'ACQUISITION ET D'ANALYSE DE DONNEES

L'installation des logiciels qui accompagnent l'Eye Tracker commence par une copie du contenu du CD d'installation dans un répertoire qu'on aura créé sur le disque dur de l'ordinateur (Le CD contient deux dossiers appelés 'Drivers' et 'ViewPoint 2.9.2.5').

Une fois la copie effectuée, il faut ouvrir le dossier 'drivers/PC-60/Windows 7 32 BIT/' et procéder à l'installation du programme appelé 'FALCON32_46000' qui permettra à l'ordinateur de reconnaître et utiliser les caméras. A ce stade, un redémarrage est recommandé pour que les modifications prennent effet.

Ensuite, il suffit d'exécuter le logiciel 'ViewPoint-USB-90HZ' pour avoir accès à l'interface d'acquisition ViewPoint.

C) L'INTERFACE VIEWPOINT

La fenêtre principale de ViewPoint EyeTracker contient plusieurs autres petites fenêtres que l'on peut afficher ou masquer via l'onglet 'Windows'.



Figure 7 Aperçu de l'interface multifenêtres de ViewPoint

Celles qui nous intéressent sont :

EyeCamera : Elles sont au nombre de deux en *Binocular mode* et affichent ce que « voient » les caméras. Elles permettent de s'assurer que l'œil du sujet est bien centré, que la mise au point est correctement effectuée, et que le logiciel arrive bien à différencier la pupille de l'iris.

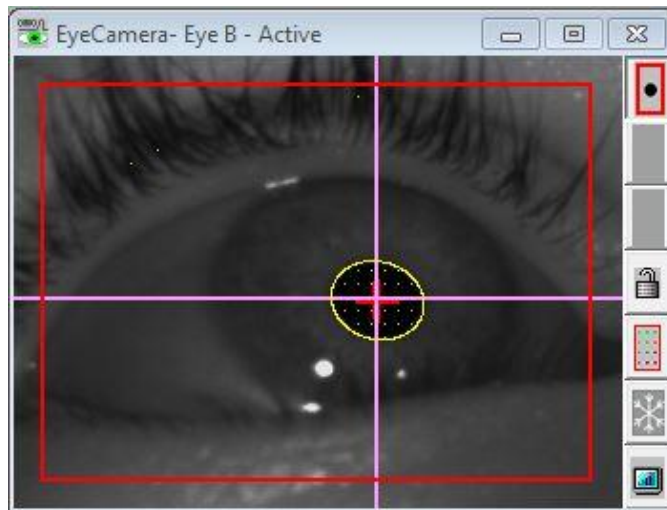


Figure 8 Fenêtre EyeCamera, affichant ce que 'voit' la caméra.

EyeSpace : Cette fenêtre sert à configurer puis lancer la procédure appelée *Calibration*, et de s'assurer que celle-ci s'est bien déroulée.

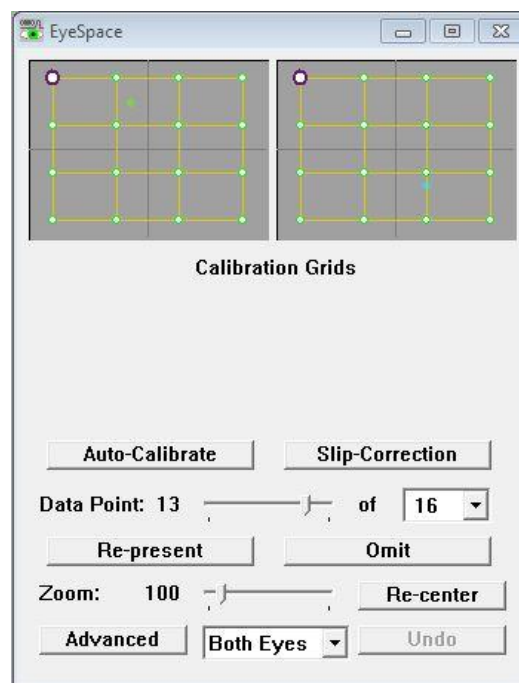


Figure 9 Fenêtre EyeSpace, utile pour la calibration.

GazeSpace : Affiche ce que le sujet voit, et permet de suivre le trajet de son regard en temps réel.

Controls : Permet notamment de lancer (*New Recording*) et suspendre l'acquisition (*Pause*), d'accéder aux fichiers de données (*Browse*) et de lancer l'utilitaire d'analyse (*Analyze*).

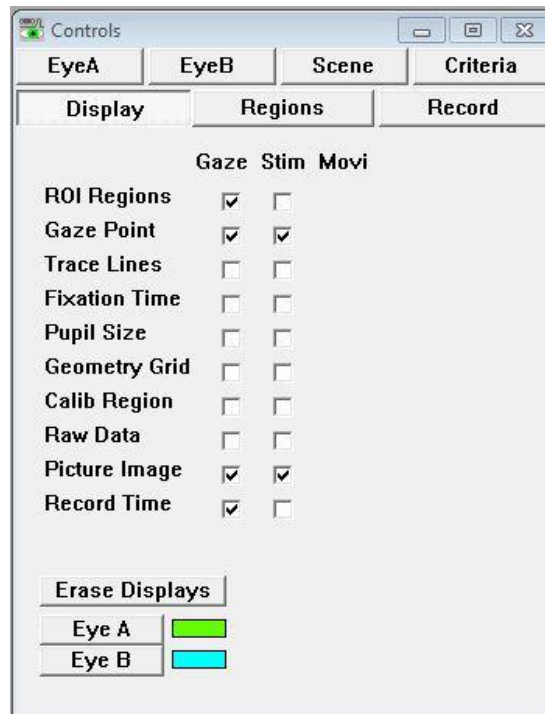


Figure 10 La fenêtre contrôle permet de contrôler l'affichage, de lancer les enregistrements et les analyses, de changer les paramètres avancés, etc...

2. PARAMETRAGE

A) CONFIGURATION DES CAMERAS

Afin d'avoir une précision optimale, il faut modifier les paramètres des caméras au sein du logiciel **uEye Cockpit**. Il s'agit, pour chacune des caméras, de changer la valeur des paramètres *Pixel Clock*, *Frame rate* et *Exposure time* en Optimum. Ainsi, les valeurs sont maximisées tout en tenant compte des performances de la machine.

B) MODES: MONOCULAIRE ET BINOCULAIRE

Lorsqu'on lance **ViewPoint EyeTracker**, il est en mode *Monocular*, c'est-à-dire qu'il utilise une seule caméra et donc qu'il suit le mouvement d'un seul œil.

Dans l'onglet *Interface*, on peut choisir de passer en mode *Binocular*, mais contrairement à ce que font les machines d'autres fabricants, la position moyenne du point de regard (Gaze point) n'est pas calculée automatiquement et on ne fait que doubler la quantité de données. Les données obtenues sont volumineuses et compliquées à traiter si l'on veut se servir d'un logiciel extérieur. Il vaut donc mieux se contenter du *Monocular mode*, sachant que les lignes de vision des deux yeux convergent en un seul point.

IV. DEROULEMENT DES TESTS

1. AVANT DE COMMENCER

Avant de commencer les tests, il est important de préparer un certain nombre de choses :

Tout d'abord, les stimuli, qui sont les images que l'on veut « tester » en les montrant aux sujets. Il faut les convertir au format bitmap (.bmp), le seul accepté par le logiciel **ViewPoint** et les adapter à la taille de l'écran d'affichage que l'on utilise. Par exemple, si l'on utilise un écran en 1366x768, il faudra que notre image soit de la même taille afin que l'on puisse l'afficher en plein écran sans devoir l'étirer (On évite ainsi les déformations et pertes de qualité). J'ai utilisé pour cela le logiciel d'édition et de retouche d'image **Photofiltre Studio X**. On peut également se servir du logiciel gratuit et libre GIMP, très polyvalent et multiplateformes.

Par ailleurs, l'écran d'affichage doit avoir un fond d'écran neutre et uni (de couleur grise claire par exemple), sans éléments graphiques qui pourraient attirer l'attention et ainsi biaiser la trajectoire du regard lorsque commence l'enregistrement.

Ensuite, la salle où se déroulent les tests doit être dûment préparée : L'environnement doit être calme : pas d'agitation, de discussion ou de passage. Il est extrêmement important d'informer les juges au sujet de ce à quoi ils vont participer et leur expliquer qu'il ne s'agit pas d'eux mais de l'image qu'ils regardent. Il faut leur éviter toute source de stress ou d'appréhension qui pourrait leur faire modifier leur comportement. Le but est que les participants parcourent l'image du regard de la manière la plus naturelle possible, comme il le ferait de manière inconsciente s'ils la voyaient par hasard. Pour ce faire, il faut les mettre à l'aise en les accueillant de manière chaleureuse et prendre le temps d'engager la discussion avec chacun.

Il faut également penser à préparer des fiches qui permettront de recueillir les dires des participants, car il est essentiel de les questionner après chaque test. En effet, l'eye tracking permet juste de savoir quels éléments sont regardés, mais ne permet en aucun cas de savoir pour quelles raisons. On repassera donc la vidéo du trajet parcouru par le regard de chacun sur le stimulus, tout en le questionnant sur les raisons qui font qu'il a regardé tel ou tel élément.

De même, il faut que le poste d'essai soit adapté aux sujets. Le bureau utilisé pour fixer l'eye tracker doit être suffisamment haut et il faut se servir d'un tabouret fixe pour que la tête reste immobile. Le tabouret doit être poussé vers le bureau jusqu'à ce que le dos du participant soit droit et qu'il se sente à l'aise.

Par ailleurs, il faut que toutes les vis de la structure soient correctement serrées et que les différents câbles passent derrière le bureau et soient hors d'atteinte.

2. REGLAGES PRELIMINAIRES

Une fois le sujet est assis et installé, il faut déplacer les caméras de manière à centrer ses pupilles, on s'assure que c'est bien le cas sur l'écran de contrôle. La position des caméras doit désormais rester inchangée (D'où l'importance d'un bon serrage des vis et d'une bonne organisation du poste de travail). Il faut ensuite faire les réglages de mise au point sur les caméras ; en effet les commandes de mise au point et d'ouverture sont placées sur l'objectif.



Figure 11 Commandes d'ouverture et de mise au point

Dans des conditions normales d'éclairage, il est recommandé de laisser l'ouverture au maximum afin que la caméra puisse recueillir un maximum de lumière infrarouge réfléchi par l'œil. Il faut que la différence de couleur entre l'iris et la pupille soit franche, l'iris apparaissant en gris et la pupille en noir sombre. Il est recommandé de faire des tests en condition réelle avec une pupille en mouvement et voir si le logiciel arrive à suivre la position de cette dernière.

Pour ce qui est des sujets portant des lunettes, il vaut mieux les retirer pendant le test. Si ce n'est pas possible, elles peuvent être gardées mais les caméras doivent être alors rapprochées de manière à ce que le cadre des lunettes ne soit pas dans le champ des caméras. En effet, les cadres en métal ou plastique réfléchissent la lumière et peuvent fausser la détection de la pupille.

3. CALIBRATION

Etape essentielle, elle permet de donner des repères dans l'espace réel (l'écran) au logiciel **ViewPoint**, il s'agit d'une série de point formant une grille qui s'affichent dans un ordre aléatoire et que le sujet doit fixer un à un afin que le logiciel puisse délimiter le champ de vision du sujet et associer à chaque position de la pupille une position dans l'espace réel du plan de l'écran d'affichage.

Il est possible de choisir un nombre de points de calibration, le minimum étant de quatre. Seize points permettent d'avoir une très bonne précision, mais les sujets ont tendances à perdre patience et à se déconcentrer avec un nombre de points aussi élevé. Par ailleurs, un nombre de neufs points de calibrations permet d'avoir une précision équivalente sinon supérieure.

Une fois le nombre de points choisi, il faut cliquer sur *Auto calibrate* ou utiliser le raccourci clavier Ctrl+A. On aura expliqué au préalable au sujet comment se passait la calibration, et l'on s'efforcera de garder un environnement exempt de toute source de distraction visuelle ou auditive.

A la fin de la procédure, on peut voir l'allure de la grille formée par les points enregistrés par les caméras. Celle-ci doit dessiner une forme aussi proche que possible d'un rectangle régulier, celui-ci n'étant jamais parfait à cause de l'erreur induite par le facteur humain.

Le fait que la grille soit légèrement déformée ou penchée est tout à fait normal et n'affecte en rien la précision de l'appareil, cependant il arrive que, à cause d'un souci lors de l'acquisition, un ou plusieurs points soient totalement en dehors de la grille et la déforment fortement, auquel cas il faut sélectionner ces points un à un sur la grille et cliquer sur *Re-present* afin de les réafficher au sujet. La calibration est terminée une quand la grille a une forme régulière.

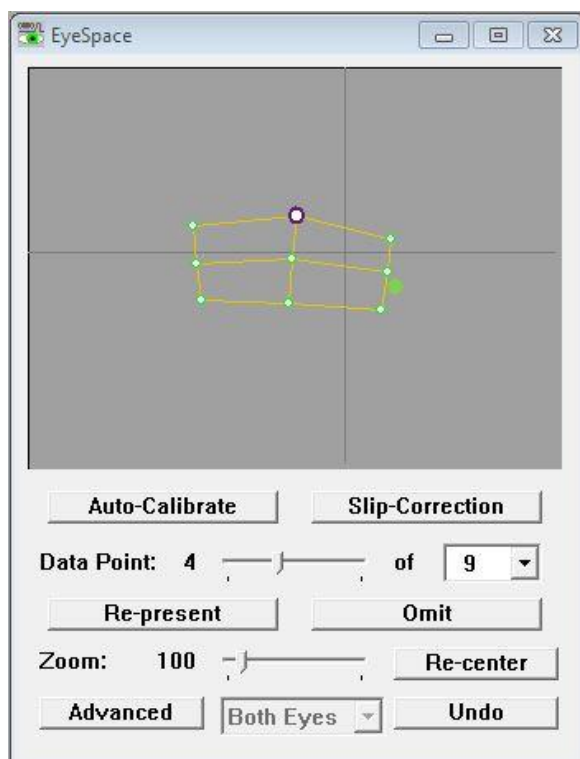


Figure 13 Ceci est une bonne calibration réalisée avec 9 points, même si le rectangle n'est pas parfait

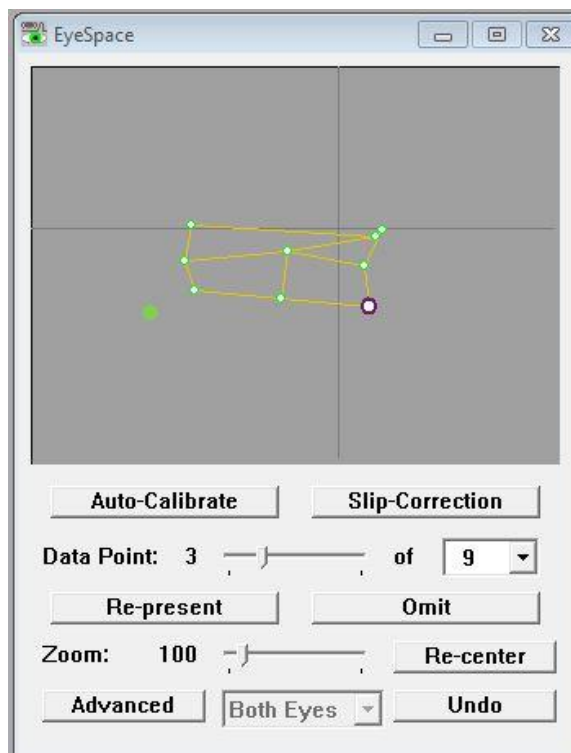


Figure 12 Voici une calibration qu'il faut corriger en sélectionnant le point défaillant (celui que doit être en haut au milieu) et en cliquant sur Re-present.

Attention : Il arrive que la calibration ne marche pas avec certains sujets et donne des résultats aberrants même au bout de plusieurs essais. Les yeux de ces personnes sont simplement incompatibles avec l'eye tracking. On estime leur proportion dans la population à près de 5%, et ce sont plutôt des personnes âgées ou avec une acuité visuelle très basse.

4. ENREGISTREMENT

A) LANCER ET STOPPER L'ACQUISITION

Une fois la calibration faite, on peut passer à l'enregistrement. Il faut laisser quelques secondes entre la fin de la calibration et le lancement de l'acquisition pour que le regard du sujet soit « libre ».

L'image à tester étant chargée (*File > Images > Load image...*), il faut cliquer sur F2 pour l'afficher et lancer l'acquisition en même temps en cliquant sur *New Recording*. On ne peut malheureusement pas fixer une durée d'acquisition en avance, et on doit l'arrêter à la main, en cliquant sur *Pause*.

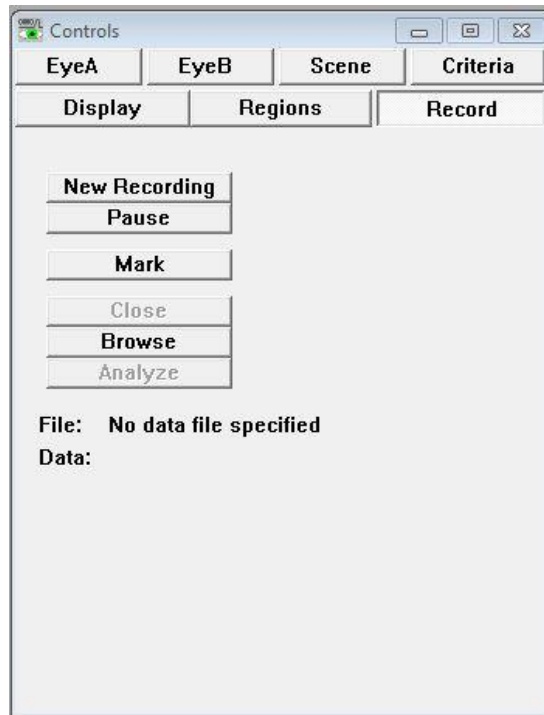


Figure 14 Panneau de contrôle de l'acquisition.

Durant l'acquisition, on peut visualiser le trajet du regard du sujet sur la fenêtre *GazeSpace*, ce qui donne une première idée de ce qui a été regardé ou pas.

B) STRUCTURE DE DONNEES

Les données sont sauvegardées dans le répertoire « \ViewPoint 2.9.2.5\Data » au format TXT selon une nomenclature de la forme AAAA-MM-JJ ;HH-MM-SS.txt (Date et heure exactes de l'acquisition).

Les fichiers de données peuvent être ouverts dans un tableur pour une meilleure lisibilité et se présentent comme suit :

- Les trente-cinq premières lignes du fichier permettent seulement d'identifier le fichier et fournissent des notes de version et autres informations sur les données et la machine.
- Les lignes 36 et 37 donnent les noms des colonnes (variables. Voir Annexe 1).
- La ligne 38 fournit le nombre d'images par seconde perçues par la cameras (FrameRate).
- La ligne 39 donne le nom de l'image testée (le stimulus).
- Les données enregistrées commencent à la ligne 40 et sont marquées par le nombre 10 en début de ligne. L'appareil enregistre

les fixations, qui correspondent à des phases où l'œil est immobile
Il peut arriver que l'appareil « rate » un point de regard, auquel cas les valeurs normalement enregistrées sont remplacées par un message du type 'Movie Frame' ou 'lost a buffer(0)'.

36	6	ATT	ADT	ALX	ALY	ACX	ACY	ARI	APW	APH	AQU	AFX
37	5	TotalTime	DeltaTime	X_Gaze	Y_Gaze	X_CorrectedGaze	Y_CorrectedGaze	Region	PupilWidth	PupilHeight	Quality	Fixation
38	7	0.000000	EyeA: FrameRate	90.5256								
39	16	0.000000	labels_patchwork.bmp									
40	10	0.0000	0.0000	0.4503	0.3813	0.4520	0.3797	-1	0.1028	0.0860	1	0.0879
41	10	0.0227	22.7007	0.4561	0.3776	0.4527	0.3795	-1	0.1050	0.0867	1	0.1106
42	10	0.0333	10.5857	0.4802	0.3504	0.4596	0.3722	-1	0.1059	0.0834	1	0.1211
43	777	0.1525	MovieFrame	1	0							
44	10	0.1531	119.8320	0.5925	0.3307	0.4948	0.3600	-1	0.1002	0.0848	1	0.2410
45	10	0.1570	3.8885	0.5938	0.3161	0.5307	0.3437	-1	0.1020	0.0795	1	0.0039

Figure 15 Ceci est un extrait d'un fichier de données brutes Arrington Viewpoint.

- Le dernier point enregistré porte le marqueur 'END'.

5. ANALYSE DES DONNEES

A) OBJECTIF DE L'ANALYSE

Selon le type d'étude que l'on mène, on s'intéressera à des éléments différents. Si l'on effectue une étude du type 'Shelf testing' où l'on présente une série de produit (Comme sur une étagère de supermarché) par exemple, on s'intéressera, sur une durée de quelques secondes aux produits qui attirent le regard en premier. Le but est de reproduire les conditions dans lesquelles se trouve un consommateur ayant un temps limité pour faire ses courses, avec un comportement d'achat qui lui est propre. On est alors dans une étude de type comparative destinée à mesurer l'attractivité visuelle d'emballages ou de marques du même segment (celui des bières par exemple).

On adaptera les modalités d'expérimentation et d'acquisition à ses objectifs.

B) UTILISATION DU LOGICIEL 'VIEWPOINT DATA ANALYSIS'

Fourni avec **ViewPoint**, il permet de visualiser la trajectoire du regard au ralenti, afin de se rendre compte des points d'intérêt et des fixations. On peut appliquer plusieurs réglages à la visualisation via la fenêtre *Controls* : Afficher les temps de fixation, trajectoire qui trace des lignes, ajuster le lissage ...

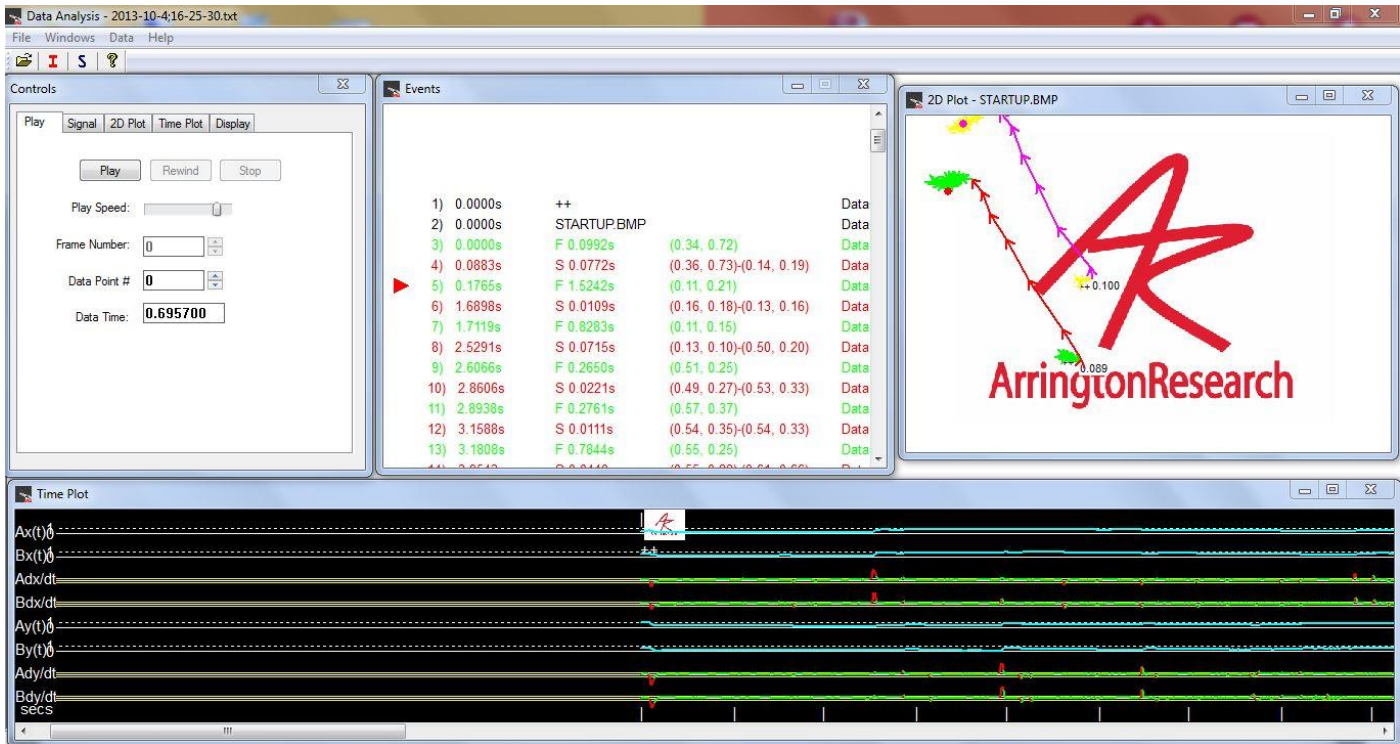


Figure 17 Aperçu de l'interface du logiciel Data Analysis.

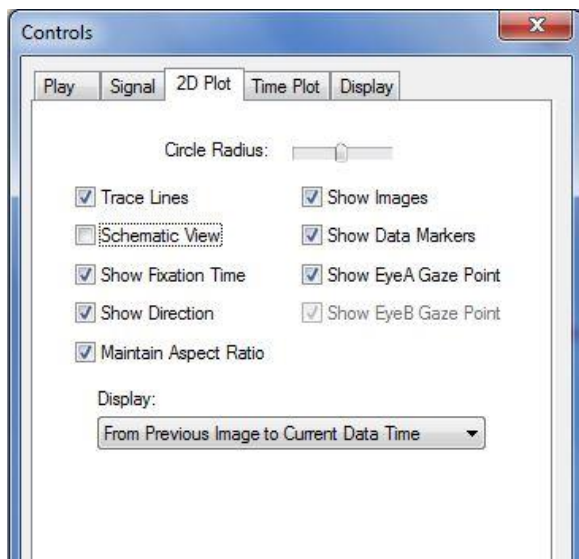


Figure 16 Fenêtre de contrôle de ce qui est affiché lors de la rediffusion du trajet du regard.

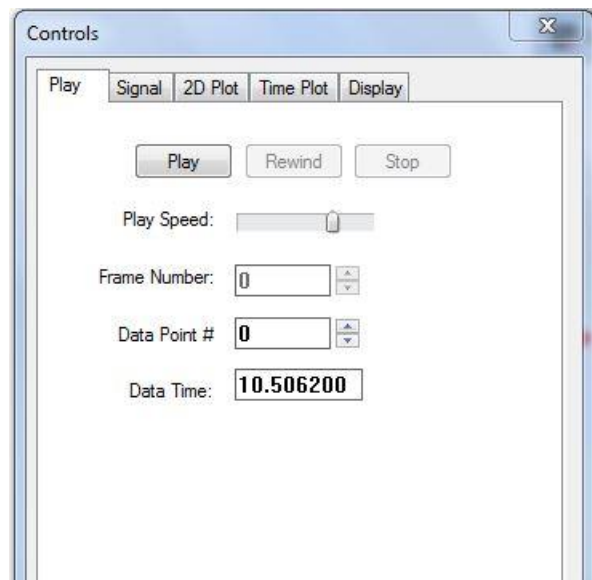


Figure 18 Fenêtre de contrôle de la lecture.

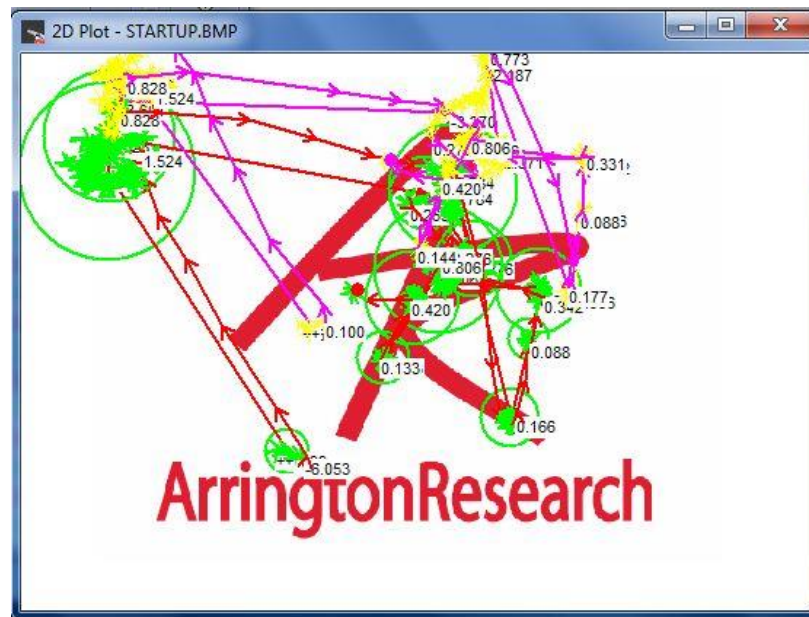


Figure 19 Rediffusion vidéo d'un trajet de regard.

C) AUTRES

Utiliser le logiciel **DataAnalysis** est simple et pratique lorsqu'on travaille avec un petit nombre de sujets, mais il peut être très pratique d'avoir un moyen de visualisation globale du type carte de chaleur. Le tracé de ce type de graphiques n'est malheureusement pas implémenté dans la suite **ViewPoint**, et il faut se tourner vers d'autres solutions comme R ou Matlab.

Pour utiliser les données enregistrées par ViewPoint avec d'autres programmes, il faut commencer par les nettoyer pour ne garder que 3 variables : Les coordonnées X et Y de chaque fixation, et le durée de cette dernière en secondes.

1	X	Y	Fixation
2	0.2907	0.3548	0.0332
3	0.2907	0.3542	0.0642
4	0.2906	0.354	0.0661
5	0.2906	0.3539	0.0772
6	0.2904	0.3538	0.0893
7	0.2904	0.3546	0.0993

Figure 20 Exemple de données 'nettoyées'.

Il faut ensuite pour gagner en lisibilité ramener les coordonnées X et Y, enregistrées entre 0 et 1 aux dimensions de l'écran. Si l'on considère encore une fois un écran de 1366x768, il faut multiplier les coordonnées X par 1366 et Y par 768.

Il s'agit ensuite de tracer une carte de chaleur où la couleur du point de coordonnées (x,y) dépend de la durée de fixation en ce même point. Voici un script R commenté qui permet de faire cela.

Code commenté:

```
heat <- read.table("D:/Anas/COURS/Stage 4A
HCMUT/Donnees/heatproject.txt", header=TRUE, sep="\t",
na.strings="NA", dec=".", strip.white=TRUE)

#on charge nos données brutes stockées dans heatproject.txt.

x<-heat$X*1366
y<-heat$Y*768
z<-heat$Fixation

#on stocke les colonnes de notre jeu de données dans les variables x,y
et z.

plot(x, y, col=c("green","red")[cut(z, c(0,0.5,1.5))])
gr <- colorRampPalette(c("green","red"))
with(heat, plot(x, y, col=gr(30)[cut(z,
seq(0,1.5,length.out=30))]) ,width=1366,height=768, units='px')

#On génère une palette de couleur grâce à la fonction
ColorRampPalette() allant du vert au rouge et trace le nuage de point
de coordonnées x,y en déterminant la couleur de chaque point en
fonction de la valeur de z. Plus la fixation est longue, et plus on
tend vers le rouge saturé.
```

On peut se servir des fonctions emf() ou pdf() pour exporter directement le graphique dans un fichier image vectorielle Windows Metafile ou PDF. Il faudra bien veiller à utiliser la fonction dev.off() après l'exportation.

Voilà le résultat obtenu avec un jeu de données générées aléatoirement :

Code :

```
plot(x, y, col=c("green","red")[cut(z, c(0,0.5,1.5))])  
dat <- data.frame(x=rnorm(500),y=rnorm(500),z=runif(200, 0, 1.3364))  
with(dat, plot(x, y, col=c("green","red")[cut(z, c(0,0.5,1.5))]))  
gr <- colorRampPalette(c("green","red"))  
with(dat, plot(x, y, col=gr(30)[cut(z, seq(0,1.5,length.out=30))]))
```

Sortie :

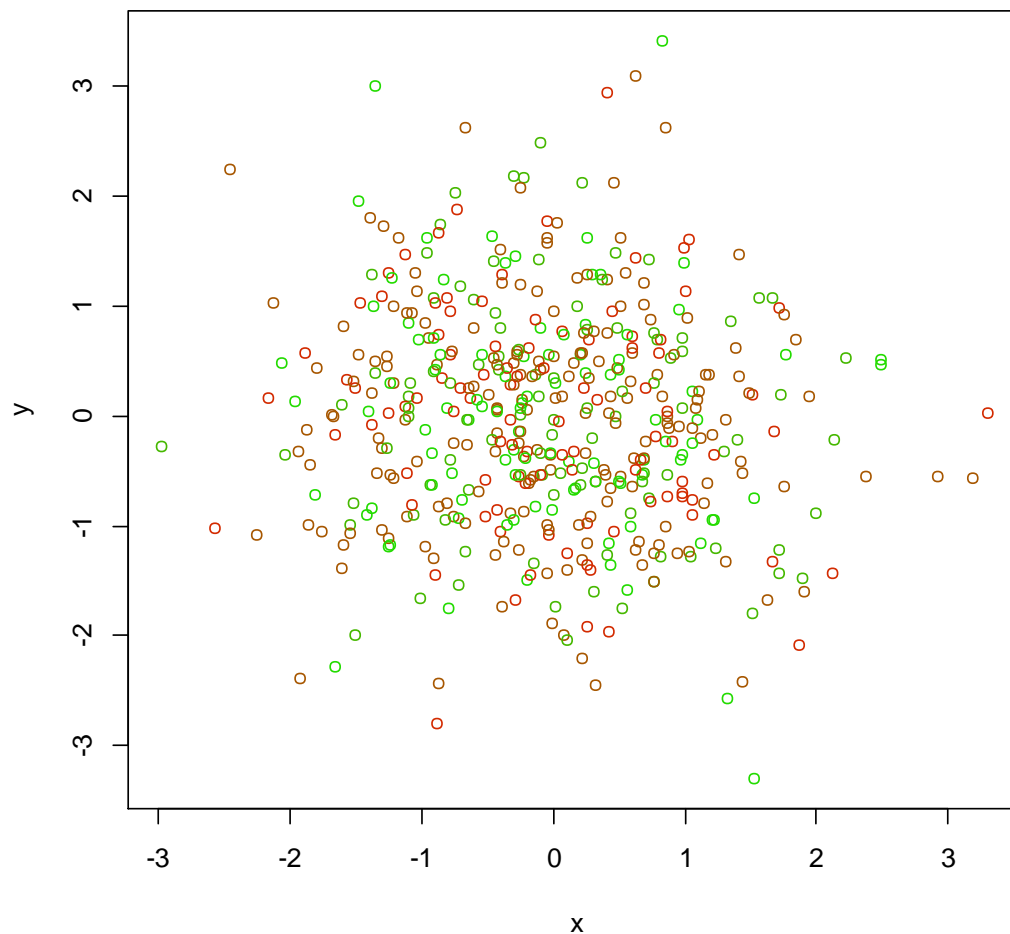


Figure 21 Nuage de points générés aléatoirement. Couleur = fixation, selon une échelle vert-rouge.

Il faut ensuite travailler la sortie (exportée au format vectoriel EMF) afin de mettre un fond transparent et la superposer à l'image stimulus. On utilise pour cela le logiciel Photofiltre.

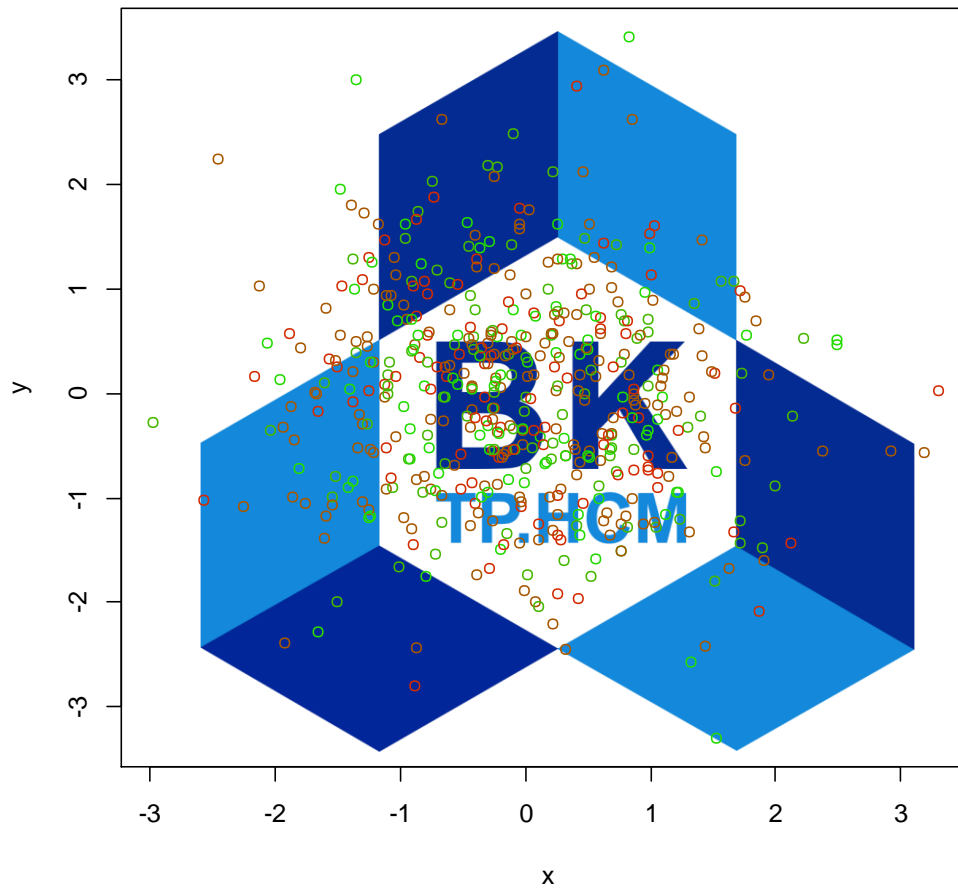


Figure 22 Nuage de points, avec effet transparence et une image insérée en fond

Voici à présent comment faire avec le logiciel Matlab, bien plus simple à utiliser que R lorsqu'il s'agit de tracer des graphiques car les paramètres avancés sont accessibles par une interface graphique pour la grande majorité. Nous utiliserons également le logiciel gratuit d'édition d'images Photofiltre pour obtenir notre image finale.

Nous utiliserons la version R2013a de Matlab.

Une fois le logiciel lancé, cliquer sur Import Data sous la rubrique HOME pour entrer les données dans le logiciel

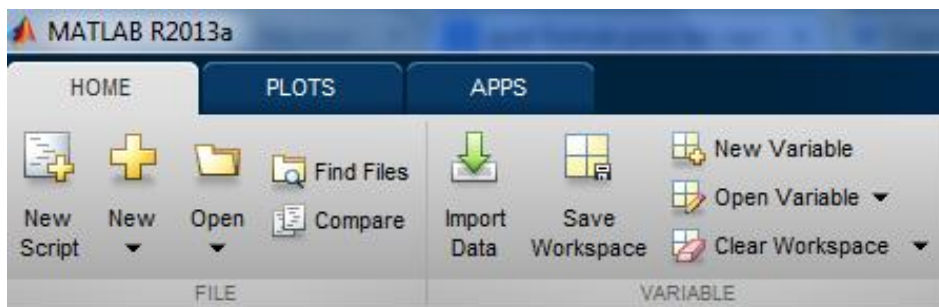


Figure 23 Menu principal Matlab (partie gauche).

On sélectionne le jeu de données rapporté aux dimensions de l'écran (dans l'exemple, il est au format txt et rapporté à un écran 1366x768), la fenêtre Import s'ouvre alors d'elle-même.

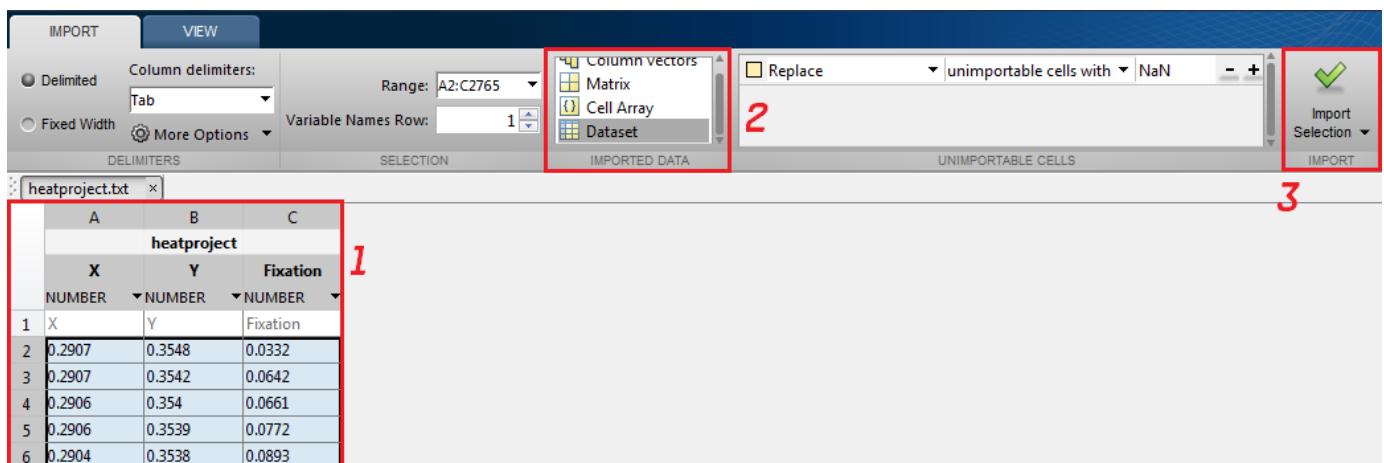


Figure 24 Fenêtre import

Après avoir vérifié que Matlab a bien reconnu la structure des données (1), on sélectionne l'option *Dataset* (2), puis *Import Selection* (3). On peut à présent retourner à la fenêtre principale de Matlab. Le jeu de données s'affiche alors dans l'encart *Workspace* à droite.

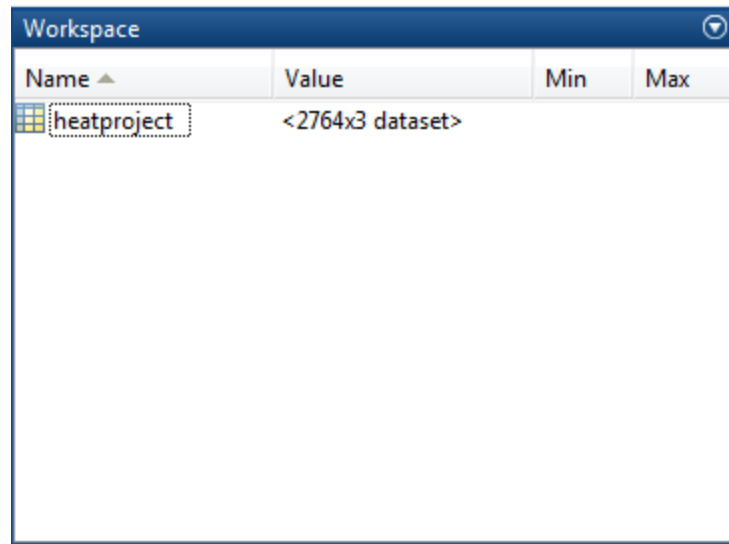


Figure 25 Encart Workspace, affichant les données en mémoire.

Cliquer alors sur l'onglet *Plots* puis sur le nom du jeu de données puis sélectionner les valeurs à représenter dans le tableur de l'encart *Variables*.

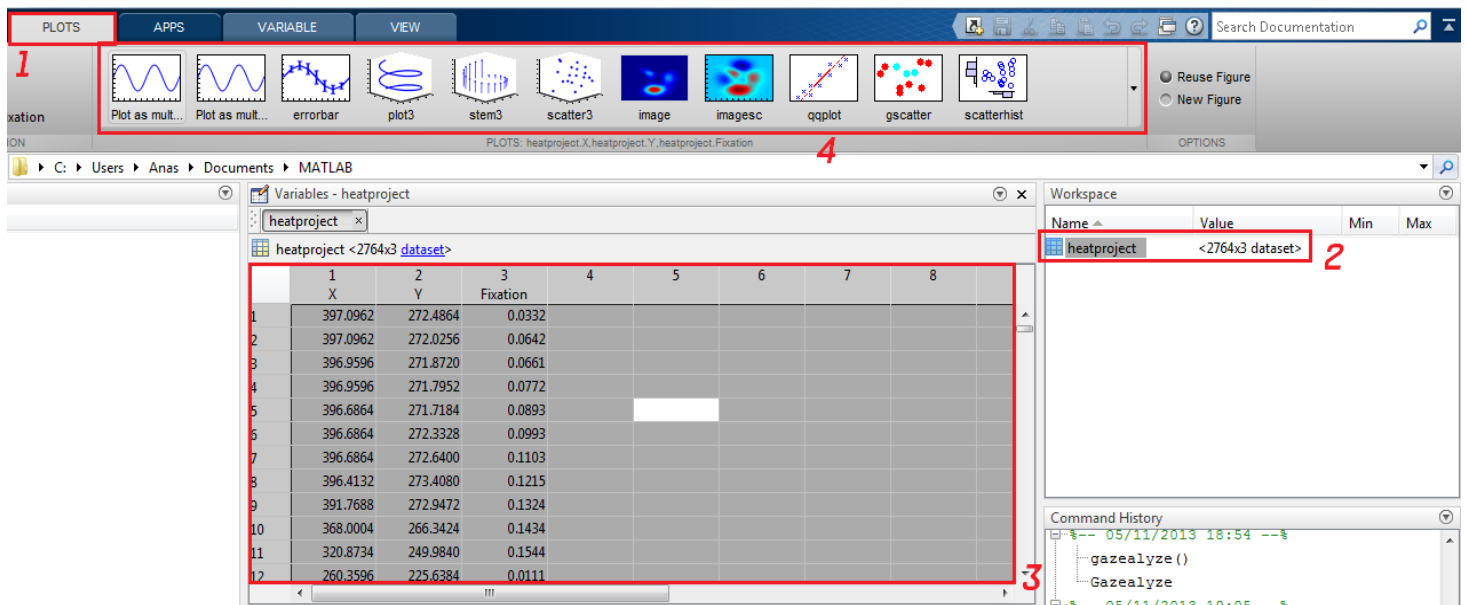


Figure 26 Fenêtre Plots, permettant de tracer toutes sortes de graphiques paramétrables.

On a alors accès à une grande variété de graphiques. On peut choisir le type que l'on veut.

La manière la plus élégante est encore d'utiliser le package ggplot2, une implémentation de Grammar Of Graphics sous R :

On commence par installer le package ggplot2 avec `install.packages('ggplot2')`, puis le jeu de données étant chargé en mémoire :

```
win.metafile('humex.wmf')
library(ggplot2)
library(grid)
attach(heat1366)
dat<-data.frame(X,Y,Fixation)
p<- ggplot(data=dat,aes(x=X,y=Y,colour=Fixation)) +
  geom_point(alpha=1/4) +
  scale_colour_gradient(low='green',high='red') +
  xlim(0,1366) + ylim(0,768) + labs(x=NULL,y=NULL) +
  theme(legend.position="none")
detach(heat1366)
dev.off()
```

Commentaires :

ggplot2 permet d'enchaîner les instructions grâce au signe '+' et ainsi de rajouter des attributs à son graphique, au fur et à mesure. On commence par charger ggplot2 et grid qui va permettre de manipuler notre graphique avant exportation.

ggplot2 ne prend que des données sous forme data.frame, nous en fabriquons donc une à partir de notre jeu de données grâce à la fonction data.frame() que nous stockons dans l'objet dat.

On commence ensuite à construire notre graphique grâce à la fonction ggplot() qui prend en argument le nom de l'objet contenant les données ainsi que les variables à représenter. Nous traçons X en abscisses, Y en ordonnées (comme à l'écran) et la longueur de fixation sera quant à elle attribuée à une échelle de couleur de point. `geom_point(alpha=1/4)` veut dire que le graphique est sous la forme nuage de points, et le paramètre alpha fixé à 1/4 permet d'appliquer une transparence aux points, qui permet des graphiques de haute densité avec beaucoup de point qui se superposent. Nous remplaçons ensuite l'échelle de

couleur par défaut (nuances de bleu) par une échelle allant du vert au rouge. Nous fixons ensuite les tailles des axes grâce aux paramètres `xlim()` et `ylim()`, puis supprimons les noms d'axes via le paramètre `labs()`. On supprime ensuite la légende (échelle de couleur).

Les fonctions `win.metafile('heatmap.wmf')` et `dev.off()` encadrent le code et permettent d'exporter le graphique dans un fichier Windows Metafile, un format redimensionnable et compatible Microsoft Office. Les fichiers images vectoriels sont manipulables avec le logiciel libre InkScape.

On peut par ailleurs tracer un nuage de point général, qui résume ce qu'ont regardé tous les juges de l'étude. Il suffit alors de mettre bout à bout les données « nettoyées » dans une base de données Access que l'on aura créé (Les feuilles Excel ne sont pas optimisées pour contenir plusieurs dizaines de milliers de lignes), avant de les exporter au format texte pour traitement.

V. ANNEXES :

1. ANNEXE 1 : NOM ET SIGNIFICATION DES VARIABLES

Table 7. DataFile: EyeData Record Structure, Tag = 10		
Column Heading	Type	Description
Tag	integer	The value 10 in the first column indicates an eye data record.
TotalTime	float	TotalTime = time elapsed in seconds
DeltaTime	float	dt = delta time in milliseconds since the previous data entry
X_Gaze	float	X = direction of gaze normalized with respect to the x-axis
Y_Gaze	float	Y = direction of gaze normalized with respect to the y-axis
Region	list	Which ROI or ROIs the gaze point is in
PupilWidth	float	Pupil width normalized with respect to the EyeCamera window width
PupilAspect	float	Dimensionless aspect ration of the pupil, i.e. 1.0 is a perfect circle
Quality	integer	Quality of eye movement data. See section 11.8, for a complete description of the codes.
Fixation	float	Fixation duration in seconds. A zero value indicates a saccade.
Torsion	float	Torsion in degrees. -998 indicates Torsion not being calculated. -999 indicates "Range Error". Only displayed if torsion is being measured.
Count	integer	Eye movement data record count, useful for sorting.
Mark	char	Any printable ASCII character, e.g., {a-z, A-Z, 0-9, =, #, +, %, etc.}. See Table 8, below.

TABLE DES FIGURES :

Figure 1 Vue générale de l'appareil	2
Figure 2 Les caméras, avec les lampes à infrarouges sur le côté.	3
Figure 3 Branchement des écrans sur la carte graphique. On utilise un adaptateur VGA->DVI pour brancher le 2nd écran	4
Figure 4 Hub USB 7 ports, autoalimenté (Se branche sur secteur)	4
Figure 5 L'environnement mis en place pour les expériences.	5
Figure 6 Positionnement du sujet durant les tests.	6
Figure 7 Aperçu de l'interface multifenêtres de ViewPoint	7
Figure 8 Fenêtre EyeCamera, affichant ce que 'voit' la caméra.	8
Figure 9 Fenêtre EyeSpace, utile pour la calibration.	8
Figure 10 La fenêtre contrôle permet de contrôler l'affichage, de lancer les enregistrements et les analyses, de changer les paramètres avancés, etc...	9
Figure 11 Commandes d'ouverture et de mise au point	11
Figure 12 Voici une calibration qu'il faut corriger en sélectionnant le point défaillant (celui que doit être en haut au milieu) et en cliquant sur Re-présent.	13
Figure 13 Ceci est une bonne calibration réalisée avec 9 points, même si le rectangle n'est pas parfait	13
Figure 14 Panneau de contrôle de l'acquisition.	14
Figure 15 Ceci est un extrait d'un fichier de données brutes Arrington Viewpoint.	15
Figure 16 Fenêtre de contrôle de ce qui est affiché lors de la rediffusion du trajet du regard.	16
Figure 17 Aperçu de l'interface du logiciel Data Analysis.	16
Figure 18 Fenêtre de contrôle de la lecture.	16
Figure 19 Rediffusion vidéo d'un trajet de regard.	17
Figure 20 Exemple de données 'nettoyées'.	17
Figure 21 Nuage de points générés aléatoirement. Couleur = fixation, selon une échelle vert-rouge.	19
Figure 22 Nuage de points, avec effet transparence et une image insérée en fond	20
Figure 23 Menu principal Matlab (partie gauche).	21
Figure 24 Fenêtre import	21
Figure 25 Encart Workspace, affichant les données en mémoire.	22
Figure 26 Fenêtre Plots, permettant de tracer toutes sortes graphiques de paramétrables.	22

Run an Eye Tracking study : a quick guide.

We are using here the Arrington fixed-head eye tracker, and we assume that the software and hardware are already installed and configured. This document comes as a handy addition to the Arrington Research User Guide (available on the install disk).

VI. PREPARATION:

2. THE TESTING ENVIRONMENT

The office used as a laboratory is kept very quiet and with no one inside except the experimenters and the judge who is being tested. There must be no visual distractions for the person while she is recorded. Here is the arrangement we used:

We use a desktop computer with a dual screen. The first one is used by the experimenter as a control panel to operate the software, and the second is used to display the stimuli for the respondents.

3. THE STIMULI FILES

They should exactly match the screen resolution (1366x768 for the current setup) and be in the Bitmap file format (.bmp), the only format accepted by the ViewPoint EyeTracker program. They can be made easily with an image editing software like Photofiltre or GIMP. The quality must be as high as possible in order to avoid pixelation or stretching issues.

4. THE JUDGES

First, given that the experiment involves human contact and interaction, it is very important to make the respondents feel as comfortable as possible, and let them freely express their thoughts. In order to do that, they all read an instructions sheet before coming to the experimentation room. It is written in an informal tone and clearly explains how the event goes in English and Vietnamese. The sheet stipulates that the study is about beers and not about the judges, and that they don't need to know or do anything in order to prepare for the test. The purpose is to make them feel comfortable and act spontaneously.

Also when the respondents come to the ‘lab’, it is important to welcome them in a warm way and talk to them ordinarily, they should not feel like participating at something important so that they do not think about where they look or change their behavior.

5. THE DOCUMENTS

It is essential to understand that an eye tracker is just cameras capturing a mountain of data, but it doesn’t deliver recommendations and conclusions itself, and doesn’t answer the ‘Why?’ It basically just tells you what people look at. That makes eye tracking data very difficult to interpret.

That’s why it’s crucial to take a lot of notes, and ask the judges about what they looked at, in a very smooth way. Documents must be prepared for this:



Figure 27 A stimuli image



Figure 28 A sheet we printed to take notes about what the judge looked at

VII. RUNNING THE EXPERIMENT

6. INSTALL THE JUDGE

Before starting the experiment, the judge must be well installed. He must be at ease, and the cameras must be positioned in front of the eyes, focus and aperture adjusted. We use a fixed stool, and make sure all the screws of the eye tracker are tight.

7. RUN CALIBRATION

Once everything is set up, we can start the step called **Calibration**. We can launch it using the Ctrl+A shortcut or by clicking *Auto-Calibrate* on the **EyeSpace** window in the **ViewPoint EyeTracker** software.

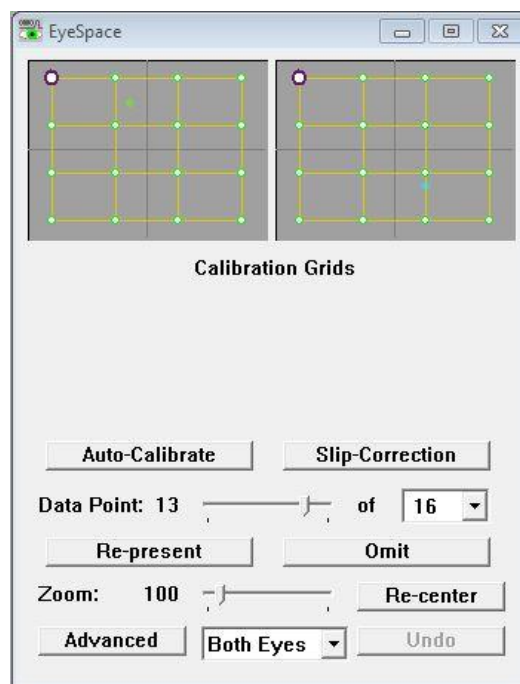


Figure 29 EyeSpace window, the calibration control panel

Once launched, this procedure will display a set of points on the screen that the judge must follow with his eyes. We can choose the number of points to display, nine appears to be a good compromise between rapidity and quality of calibration. We also obtain very good accuracy and precision.

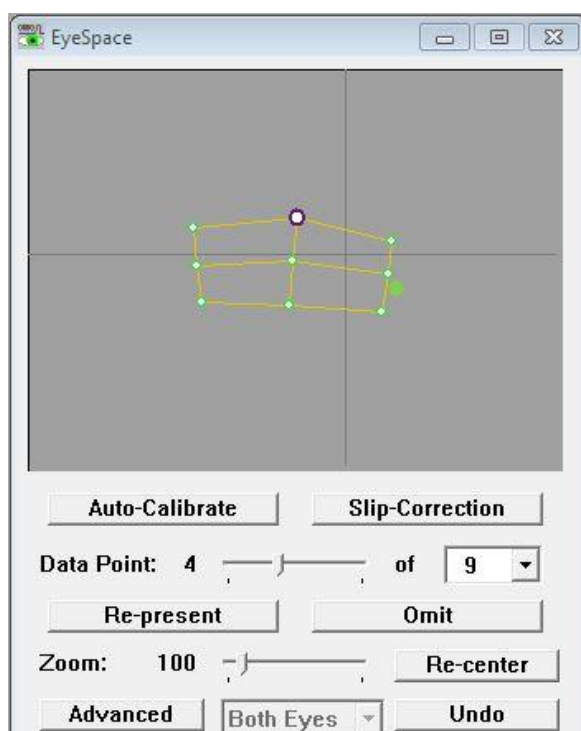


Figure 31 this is how a good calibration looks like

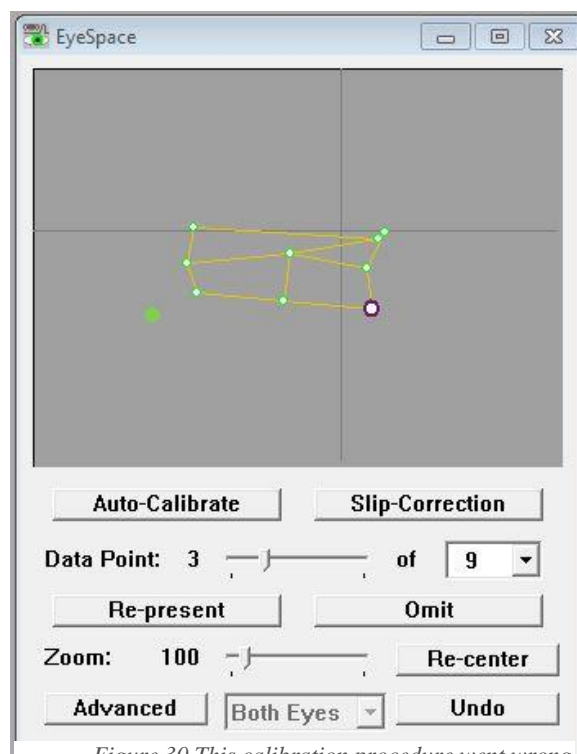


Figure 30 This calibration procedure went wrong and needs to be corrected

8. RECORD DATA

We use the exact same hardware and software settings for all the respondents. In order to make a test, we have to display the stimuli image for a chosen duration (Use **F2** to display or hide the stimuli image), and record gaze data during that same duration. The *New Recording* button (Controls > Record window) will start making a new data file, and we have to stop the recording using *Pause*.

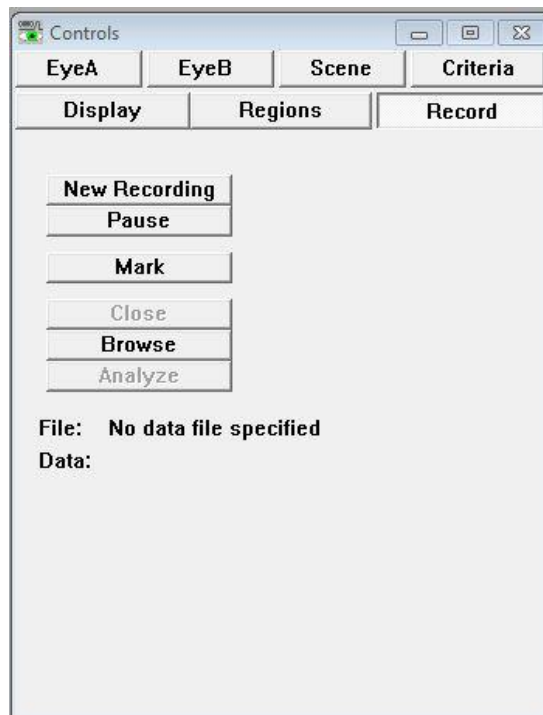


Figure 32 The Controls window

The recorded data is stored in the « \ViewPoint 2.9.2.5\Data » folder, in text format.

9. ANALYZE-IT!

The *ViewPoint Eyetracker* software suite has a built-in data analysis program. It is soberly called **Data Analysis** and can be launched from the *Controls* window by clicking *Analyze*. It provides gaze path enhanced video, and a lot of other information that we can chose to hide or show using the *Controls* window.

It is also possible to plot gaze data, using any plotting tool : R, Matlab, etc ... in order to do so, it is important to 'clean' the data up to keep only data points, characterized by 3 variables : X and Y coordinates and duration of fixation.

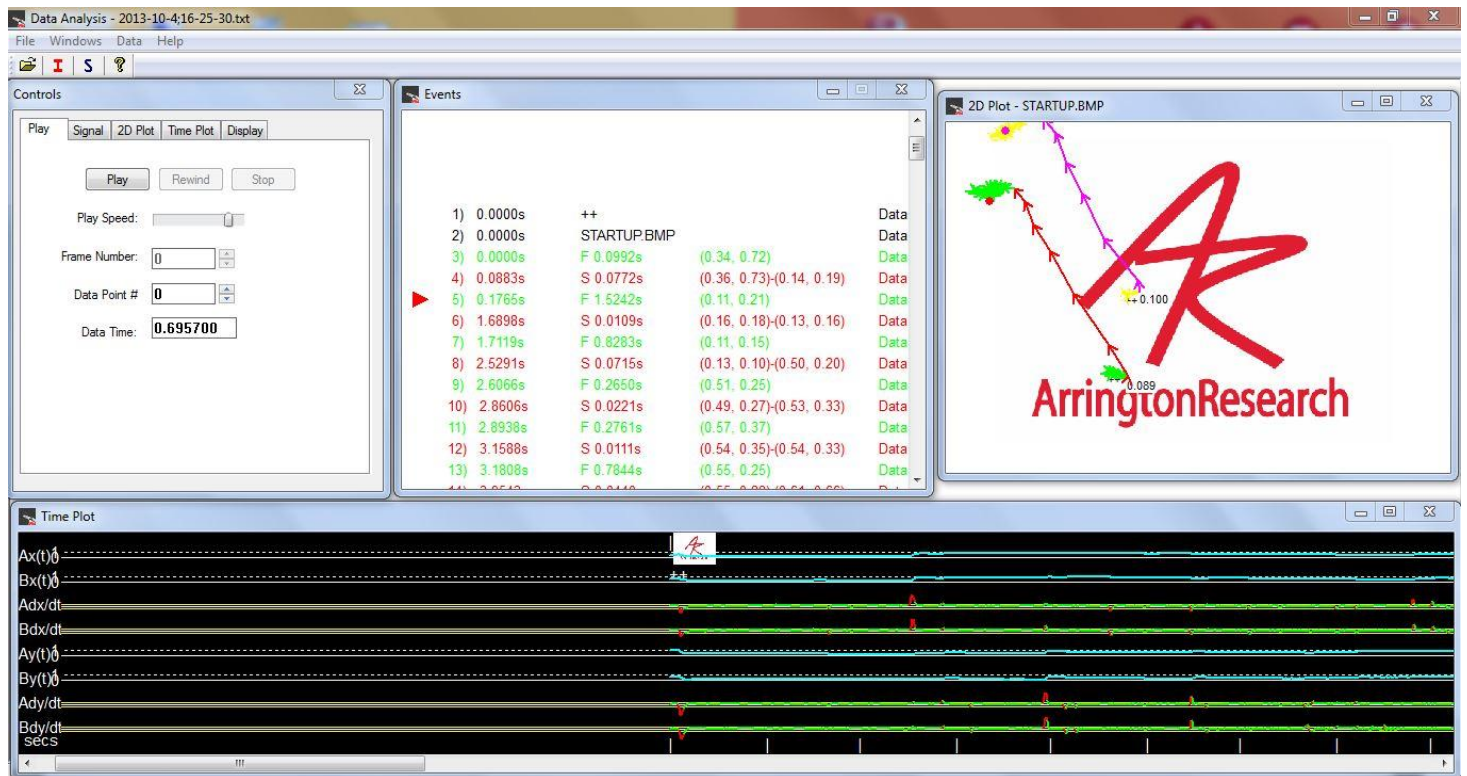


Figure 33 The Data Analysis interface

It is also possible to plot gaze data, using any plotting tool : R, Matlab, etc ... in order to do so, it is important to 'clean' the data up to keep only data points, characterized by 3 variables : X and Y coordinates and duration of fixation.

1	X	Y	Fixation
2	0.2907	0.3548	0.0332
3	0.2907	0.3542	0.0642
4	0.2906	0.354	0.0661
5	0.2906	0.3539	0.0772
6	0.2904	0.3538	0.0893
7	0.2904	0.3546	0.0993

Figure 34 This is how 'clean' data looks like

After that, and given that each coordinate belongs to a [0,1] interval, we can multiply the X variable by the screen width and Y by the screen height. For example, if we used a 1366x768 screen resolution, we will import our dataset into R and run the following command lines:

```
Attach(dat)
X<-X*1366
Y<-Y*768
Detach(dat)
```

We can then plot the (x,y) data points on a scatterplot, while associating the Fixation variable to the color of the points (like on a heatmap) or their size. This can be done using the

plot() or ggplot() R functions (for more details, please refer to the french user guide, it contains a ready-made code sample).