

Capítulo 1

Resultados

1.1. Modo de uso

Debido a las características de este trabajo de título se opta por presentar los resultados obtenidos a modo de manual de uso. En este sentido, y dado que la aplicación puede ser utilizada tanto en forma de *API* como de *GUI*, se decide presentar desde estas dos perspectivas el proceso de configuración y ejecución de un experimento.

El experimento a configurar estará conformado por dos tareas que se ejecutarán una sola vez, de forma secuencial y luego de presionar la tecla espacio: La primera tarea a configurar consistirá en un ejercicio de movimiento prosacádico bajo el paradigma de *overlap*, lo que implica que el estímulo central se encontrará presente en todos los cuadros. De esta forma, se propone un cuadro inicial que contenga solo dicho estímulo identificado por una cruz blanca, con duración de $1,7[s]$. Luego, se presenta un cuadro que contenga además el punto objetivo identificado por un cuadro rojo, 16° a la derecha del estímulo inicial, durante $1,5[s]$. La segunda tarea consiste en la presentación de una imagen centrada para analizar patrones de búsqueda. En este caso el cuadro no debe ser temporizado: La tarea termina luego de que el usuario presione la tecla espacio.

La presentación de los estímulos se realizará utilizando el monitor secundario¹ que se encuentra ubicado a $53[cm]$ del paciente². Además, se hará uso de un *eye tracker* "Eyetribe" y los resultados obtenidos serán almacenados en el directorio "`C:/SaccadeApp/events/`".

¹Resolución de $1680 \times 1050[px]$ y una pantalla con ancho efectivo de $47,5[cm]$.

²La configuración del monitor debe ser realizada mediante el *Monitor center* de *psychoPy*. Al perfil correspondiente se le asigna el nombre "my_monitor" por simplicidad.

1.1.1. Usando la API

Configurar

En este proceso se utilizarán los siguientes módulos:

```
1 from saccadeapp.api import SaccadeDB
2 from saccadeapp.api import Configuration
3 from saccadeapp.api import Experiment, Test, Frame, Component
```

El primer paso es contar con una base de datos disponible. Para crear una en el directorio donde se ejecutan los comandos se utiliza la siguiente instrucción:

```
1 database = SaccadeDB()
```

Luego, es posible crear el experimento propuesto:

```
1 # Experiment:
2 experiment = Experiment(db=database)
3 experiment.set_code(code=u"exp_0001")
4 experiment.set_info(name=u"Example Experiment", version=u"coded_1.0")
5 experiment.set_description(text=u"This experiment was created with code!")
6 experiment.set_space_start(status=True)
7 experiment.set_dialog(status=False)
8 experiment.set_random(status=False)
9 experiment.set_rest_conf(status=False)
```

Las tareas que lo conforman:

```
1 ### Test_1: Prosaccade #####
2 test1 = Test()
3 test1.set_name(u"Test 1")
4 # Test_1 - Frame_1:
5 t1_fra1 = Frame()
6 t1_fra1.set_name(u"FP frame")
7 t1_fra1.set_color(u"black")
8 t1_fra1.set_as_task(False)
9 t1_fra1.set_time(1.7)
10 test1.add_item(t1_fra1)
11 # Test_1 - Frame_1 - FP
12 t1_fp = Component()
13 t1_fp.set_name(u"Fix Point")
14 t1_fp.set_shape(u"cross")
15 t1_fp.set_color(u"white")
16 t1_fp.set_position(0.0, 0.0)
17 t1_fra1.add_item(t1_fp)
18 # Test_1 - Frame_2:
19 t1_fra2 = t1_fra1.copy()
20 t1_fra2.set_name(u"TP frame")
21 t1_fra2.set_time(1.5)
22 test1.add_item(t1_fra2)
23 # Test_1 - Frame_2 - TP
24 t1_tp = Component()
25 t1_tp.set_name(u"Target Point")
26 t1_tp.set_shape(u"square")
27 t1_tp.set_color(u"red")
28 t1_tp.set_position(16.0, 0.0)
29 t1_fra2.add_item(t1_tp)
30 #Add test1 to experiment
31 experiment.add_item(test1)
32 #Add test1 to sequence
33 experiment.sequence_add(item_id=0, quantity=1)
```

```

1  ### Test_2: Couple Image ###
2  test2 = Test()
3  test2.set_name(u"Test 2")
4  # Test_2 - Frame_1:
5  t2_fra1 = Frame()
6  t2_fra1.set_name(u"Couple")
7  t2_fra1.set_color(u"black")
8  t2_fra1.set_as_task(True)
9  t2_fra1.set_keys_allowed(u"space")
10 t2_fra1.set_keys_correct(u"space")
11 test2.add_item(t2_fra1)
12 # Test_2 - Frame_1 - Image:
13 t2_img = Component()
14 t2_img.set_name(u"Couple Image")
15 t2_img.set_image(u"couple.png")
16 t2_fra1.add_item(t2_img)
17 #Add test2 to experiment
18 experiment.add_item(test2)
19 #Add test2 to sequence
20 experiment.sequence_add(item_id=1, quantity=1)

```

Y el perfil de configuración requerido:

```

1  config_profile = Configuration(db=database)
2  config_profile.set_name(name=u"Test Profile (code)")
3  config_profile.set_events_path(path=u"C:/SaccadeApp/events/")
4  config_profile.set_tracker(tracker=u"eyetribe")
5  config_profile.set_monitor(monitor=u"my_Monitor")
6  config_profile.set_screen(screen=1)

```

Finalmente se almacena en la base de datos tanto el experimento como el perfil de configuración:

```

1  experiment.save()
2  config_profile.save()

```

Ejecutar

Para ejecutar el experimento que acaba de ser configurado y almacenado en la base de datos según las condiciones propuestas basta utilizar los siguientes comandos:

```

1  from saccadeapp.app import ExperimentHandler
2
3  handler = ExperimentHandler()
4  handler.prepare(exp_code=u"exp_0001", conf_name=u"Test Profile (code)")
5  handler.execute(frame_save=True)

```

1.1.2. Usando la GUI

Configurar

El primer paso consiste en añadir un perfil de configuración. Para esto es necesario ir a la pestaña `Configuration` en la ventana principal.

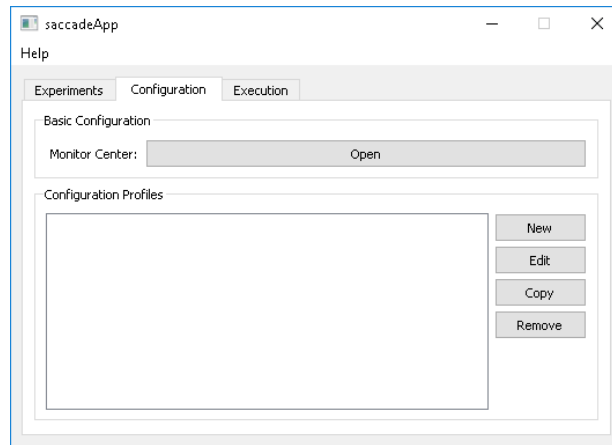


Figura 1.1: Pestaña de configuración.

Se crea un nuevo perfil donde se ingresan las configuraciones propuestas para el ejemplo, como se muestra a continuación:

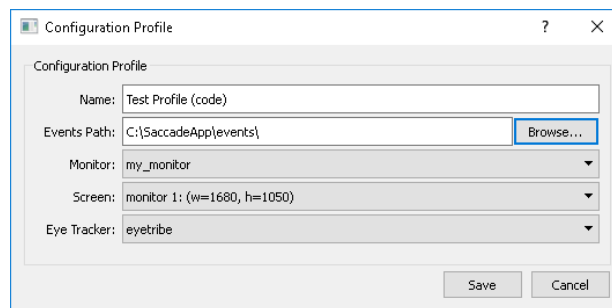


Figura 1.2: Perfil de configuración propuesto.

Luego, se finaliza la operación presionando el botón `Save`.

El segundo paso consiste en añadir un nuevo experimento. Para esto es necesario ir a la pestaña `Experiment` en la ventana principal.

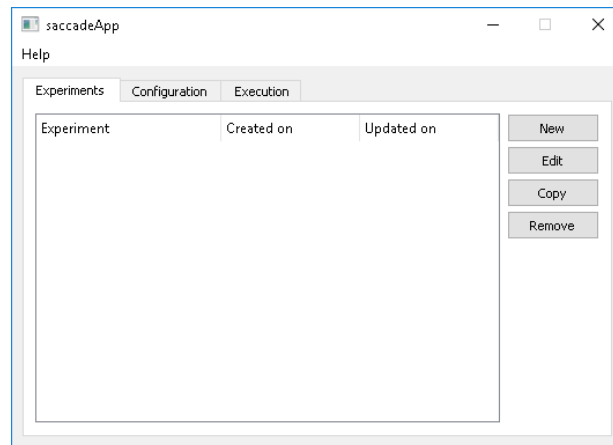


Figura 1.3: Pestaña de experimentos.

Se crea un nuevo experimento donde se ingresan las configuraciones propuestas y se añade la primera tarea.

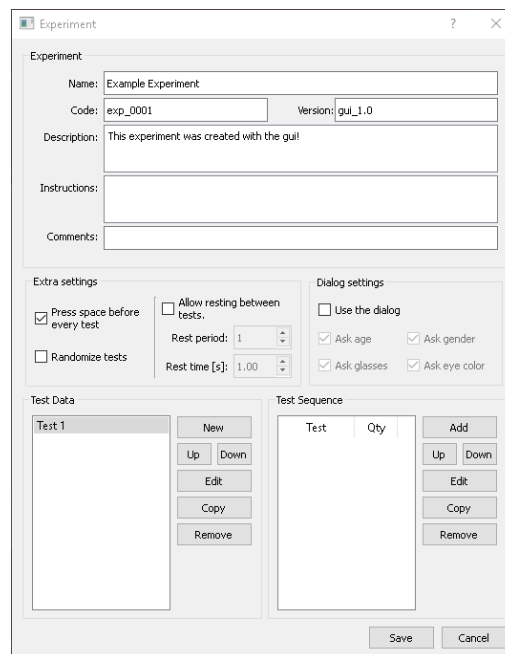


Figura 1.4: Experimento propuesto con la primera tarea configurada.

Dicha tarea se compone de dos cuadros:

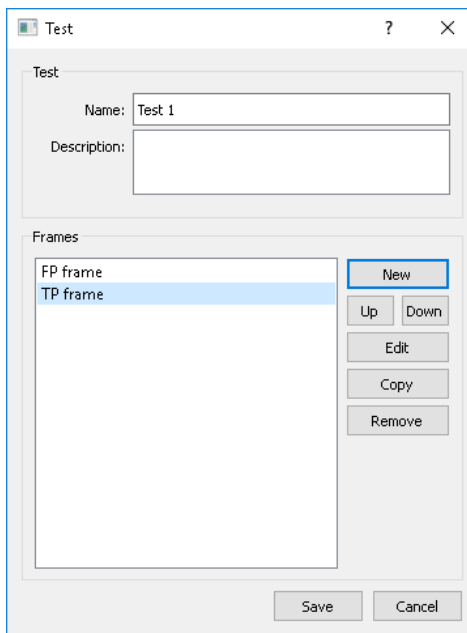
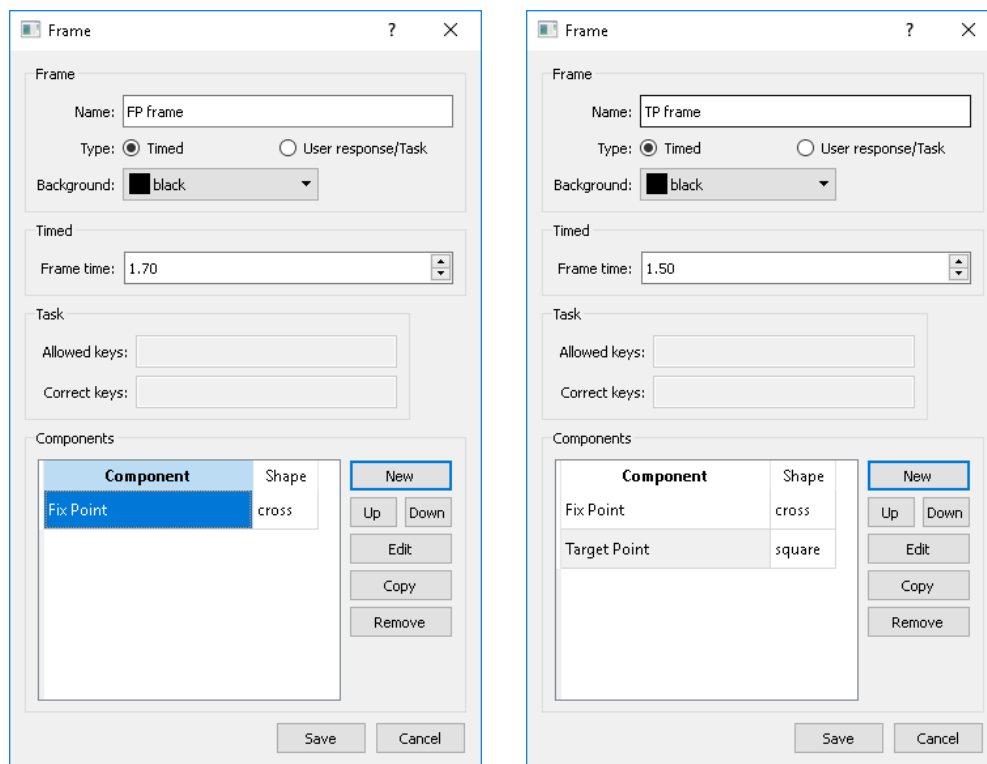


Figura 1.5: Ventana de configuración de la primera tarea.

Que se encuentran configurados de la forma:



(a) Primer cuadro.

(b) Segundo cuadro.

Figura 1.6: Configuración de los cuadros pertenecientes a la primera tarea.

Y donde los componentes que las conforman corresponden a:

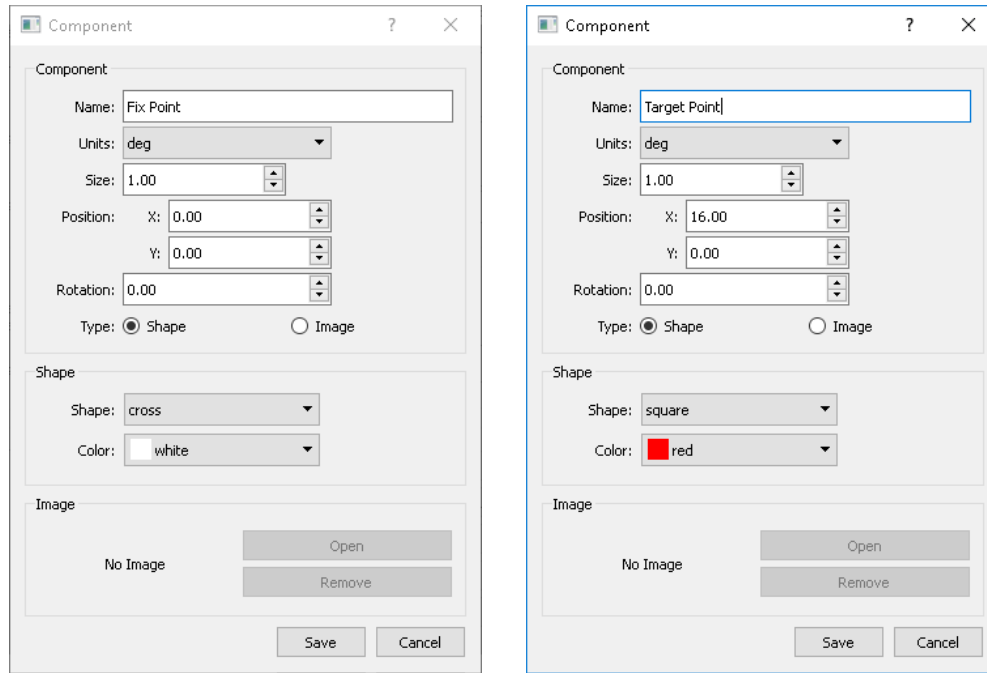


Figura 1.7: Componentes utilizados en la configuración de la primera tarea.

Para agregar la tarea a la secuencia de ejecución se utiliza el botón **Add** de la lista de secuencias, lo que despliega la siguiente ventana:

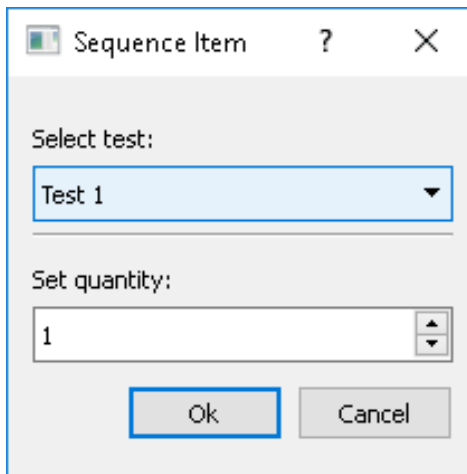


Figura 1.8: Agregando la primera tarea a la secuencia de ejecución.

Luego de añadir ese elemento se obtiene el siguiente resultado:

Experiment

Name: Example Experiment

Code: exp_0001 Version: gui_1.0

Description: This experiment was created with the gui!

Instructions:

Comments:

Extra settings:

☒ Press space before every test

☐ Allow resting between tests.

Rest period: 1

☐ Randomize tests

Rest time [s]: 1.00

Dialog settings:

☐ Use the dialog

☒ Ask age ☒ Ask gender

☒ Ask glasses ☒ Ask eye color

Test Data

Test
Test 1

New Up Down Edit Copy Remove

Test Sequence

Test	Qty
Test 1	1

Add Up Down Edit Copy Remove

Save Cancel

Figura 1.9: Experimento propuesto con la primera tarea configurada y en la secuencia de ejecución.

De forma posterior se añade la segunda tarea, que solo se compone de un cuadro.

Test

Name: Test 2

Description:

Frames

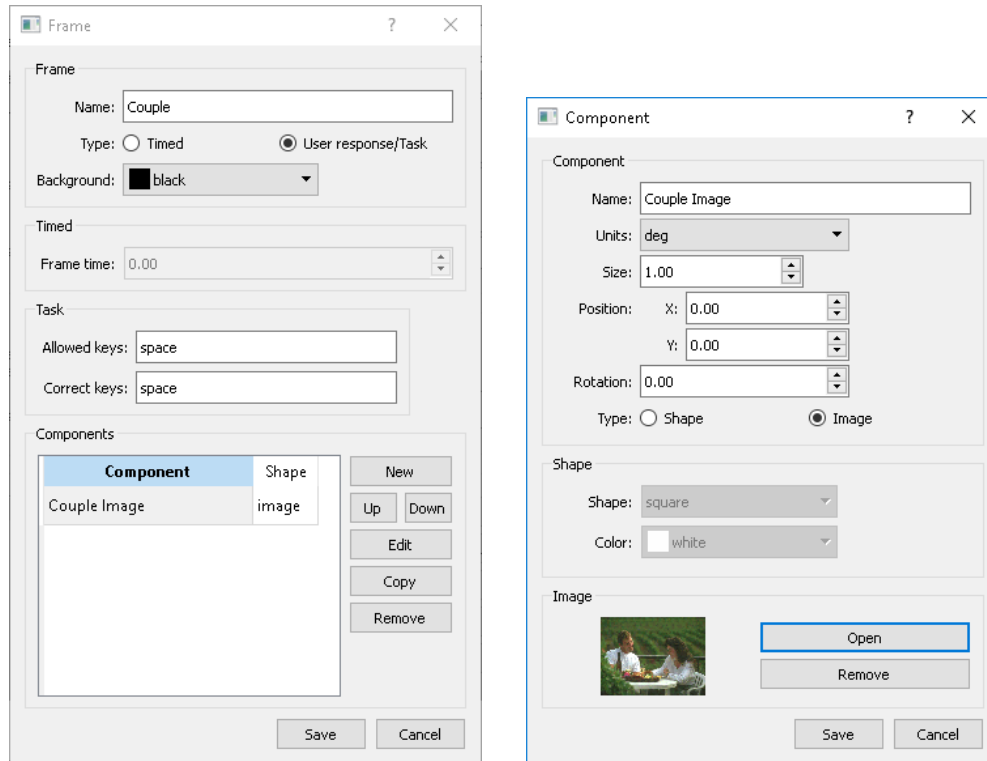
Couple

New Up Down Edit Copy Remove

Save Cancel

Figura 1.10: Ventana de configuración de la segunda tarea.

Que presenta las siguientes configuraciones:



(a) Cuadro que contiene la imagen.

(b) Imagen de la pareja.

Figura 1.11: Configuraciones de la segunda tarea.

Al añadir esta tarea a la secuencia de ejecución, el experimento queda de la forma:

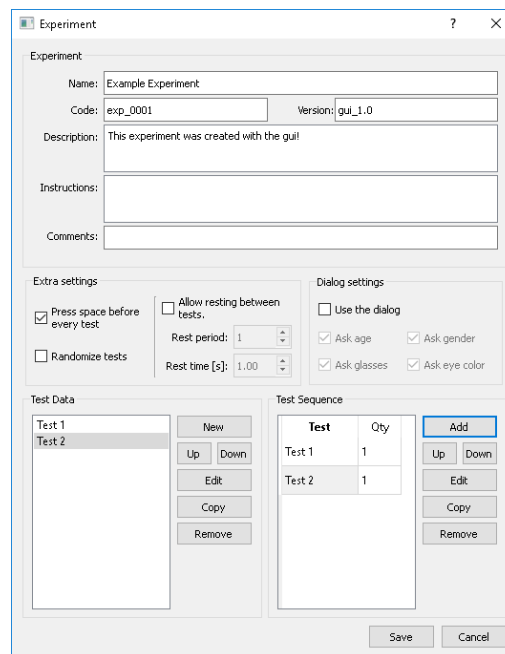


Figura 1.12: Experimento propuesto con configuración completa.

Luego, se finaliza la operación presionando el botón `Save`.

Con estas configuraciones ya es posible ejecutar un experimento. Las pestañas de configuración, luego de añadir estos elementos quedan de la siguiente forma:

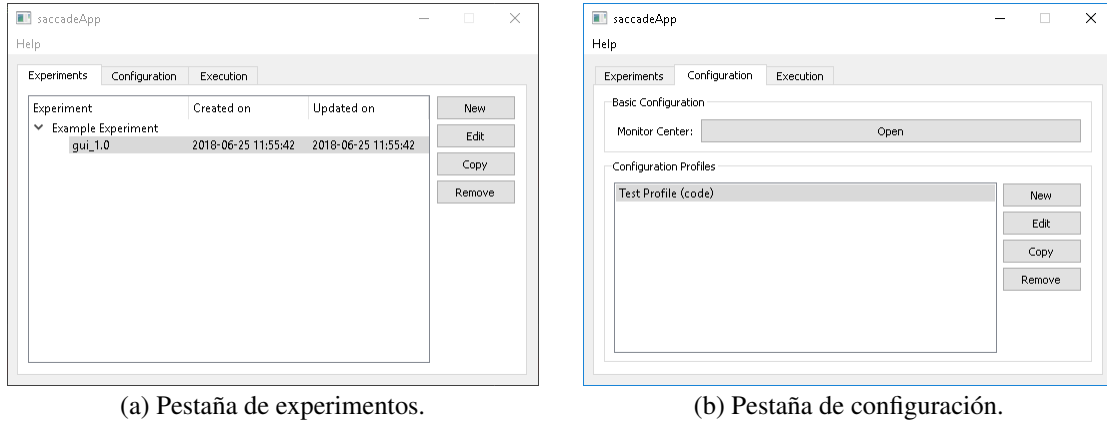


Figura 1.13: Resultado obtenido.

Ejecutar

Para ejecutar el experimento es necesario ir a la pestaña `Execution` en la ventana principal y seleccionar los elementos de interés de las listas desplegables:

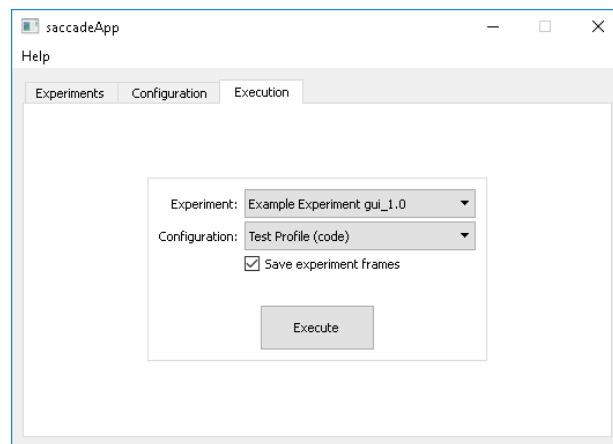


Figura 1.14: Aplicación lista para ejecutar *script*.

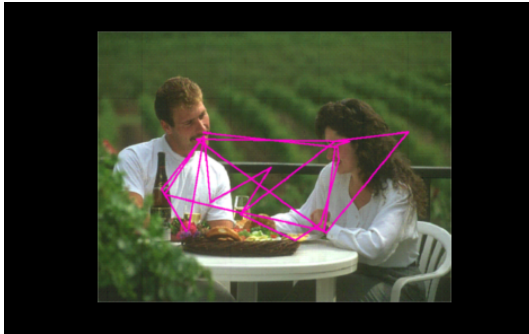
Finalmente se debe presionar el botón `Execute`.

1.2. Comprobación de funcionamiento.

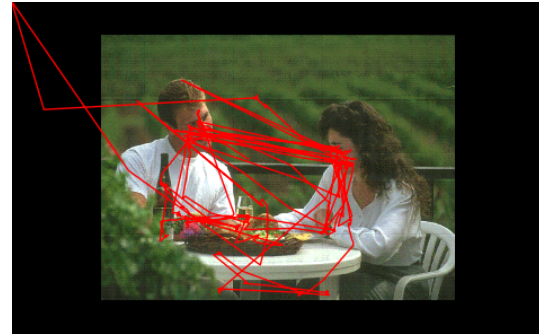
Con el fin de comprobar el funcionamiento de la aplicación y asegurar el correcto registro de los datos se creó un experimento cuya única tarea consistía en realizar un ejercicio de inspección visual sobre la imagen presentada en la figura 1.15 (a).



(a) Imagen original.



(b) Movimiento ocular esperado.



(c) Movimiento ocular registrado.

Figura 1.15: Datos registrados.

Los datos registrados desde el *eye tracker* en este experimento (1.15 (c)) muestran ser similares a los esperados (1.15 (b)), e indican que el movimiento ocular se realiza principalmente entre los rostros de la pareja y las proyecciones de hacia donde se encuentran dirigidas sus miradas.

Al consultar los registros de los puntos anómalos presentes principalmente en la esquina superior izquierda y que parecieran no corresponder al patrón de observación, se determina que se encuentran relacionados a los momentos en que el usuario cierra los ojos y no es posible realizar estimaciones de la posición ocular.