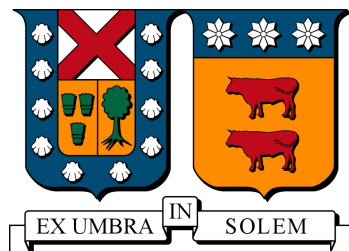


UNIVERSIDAD TÉCNICA FEDERICO SANTA MARÍA
DEPARTAMENTO DE ELECTRÓNICA
VALPARAÍSO - CHILE



**SISTEMA DE ESTIMULACIÓN VISUAL Y
REGISTRO DE MOVIMIENTOS OCULARES
PARA TAREAS SICOMOTORAS"**

CHRISTIAN ANDRÉS WICHE LATORRE

**MEMORIA DE TITULACIÓN PARA OPTAR AL TÍTULO DE INGENIERO CIVIL
ELECTRÓNICO MENCIÓN CONTROL E INSTRUMENTACIÓN**

PROFESOR GUÍA: MARÍA JOSÉ ESCOBAR SILVA
PROFESOR CORREFERENTE: MATÍAS ZAÑARTU SALAS

?? - 2018

Inserte su dedicatoria aquí

Firma

Agradecimientos

Resumen

Abstract

Índice general

Agradecimientos	I
Resumen	II
Abstract	III
1. Introducción	1
1.1. Motivación	1
1.2. Motivación y objetivos	2
2. Estado del arte	3
2.1. Sistemas de seguimiento ocular	4
2.1.1. Movimiento ocular	4
2.1.2. Métodos de captura	4
2.1.3. Sistemas comerciales más relevantes	4
2.2. Sistemas de estimulación visual	5
2.2.1. Hardware de estimulación	5
2.2.2. Software de estimulación	5
2.2.3. Experimentos de estimulación	5
3. Sistema propuesto	6
3.1. Alcances	6
3.1.1. Utilidad de la GUI	6
3.1.2. Experimentos a implementar	6
3.2. Tecnologías utilizadas	6
3.2.1. Hardware	6
3.2.2. Software	6
3.3. Diseño e implementación de la GUI	6
3.3.1. Arquitectura	6
3.3.2. Archivos de configuración	6
3.3.3. Script de ejecución	6
4. Resultados	7
4.1. Configuración de test de prueba	7
4.2. Mediciones obtenidas	7

5. Conclusiones y trabajo futuro	8
5.1. Conclusiones	8
5.2. Trabajo futuro	8
Referencias	9
A. Código fuente	11
B. Vistas de la GUI	17

Índice de figuras

2.1. Diagrama general de la interfaz hombre-máquina.	3
2.2. Ejemplo de setup experimental [1].	3

Índice de cuadros

Capítulo 1

Introducción

1.1. Motivación

Diariamente y sin prestar mayor atención la mayor parte de la población utiliza sus ojos para interactuar con su entorno: se detienen a admirar el paisaje, leer un libro, revisar su teléfono, navegar en internet, verificar que sus alimentos se encuentran en buen estado, etc. A pesar de lo simple que puede parecer esta acción, en la realidad corresponde a un proceso sumamente complejo que involucra la participación de un sinnúmero de estructuras del cuerpo.

El simple hecho de orientar nuestra vista hacia un nuevo objetivo desencadena una serie de eventos fascinantes: El globo ocular rota hasta lograr posicionarse en una determinada dirección de forma tal que los rayos de luz que son reflejados por el punto de interés se proyectan en la córnea, estructura que transduce esta información en impulsos eléctricos que son interpretados de forma posterior por el cerebro y que se traducen en información que interpretamos como una imagen.

El estudio de las dinámicas del ojo, incluyendo su movimiento (horizontal, vertical y rotacional) y el comportamiento de la pupila ha dado paso a una serie de avances que van desde detección de enfermedades (por ejemplo) hasta estudios de marketing y usabilidad (por ejemplo). El elemento común entre estas aplicaciones tan diversas corresponde a sistemas confiables que permitan correlacionar los diversos estímulos visuales con las respuestas motoras obtenidas.

Las aplicaciones más llamativas, no obstante, corresponden a la detección temprana de

Aplicación particular: eye tracking en monitorización de funciones cognitivas

- Motivación ->Posible aplicación: Detección temprana de enfermedades neurológicas:
Demencia? Parkinson?

1.2. Motivación y objetivos

Los estudios sobre movimiento ocular permiten...

Lo que hace pensar que...

Por esto, se pretende...

Así, el objetivo principal de este trabajo de título consiste en el diseño y construcción de un sistema de estimulación visual y registro de movimientos oculares para tareas sicomotoras con el fin de facilitar el proceso de puesta en marcha de experimentos asociados al estudio de... . Con lo cual se hace necesario:

- (I) Definir y programar el mecanismo de estimulación visual en acorde con las características técnicas del despliegue.
- (II) Diseñar e implementar un sistema de sincronización entre el registro y la estimulación.
- (III) Integrar al diseño un sistema de captura del movimiento ocular.
- (IV) Asegurar el correcto registro de los datos.
- (V) Construcción de protocolos de estimulación para tareas sicomotoras.
- (VI) Integración de todas las etapas en una GUI.

Capítulo 2

Estado del arte

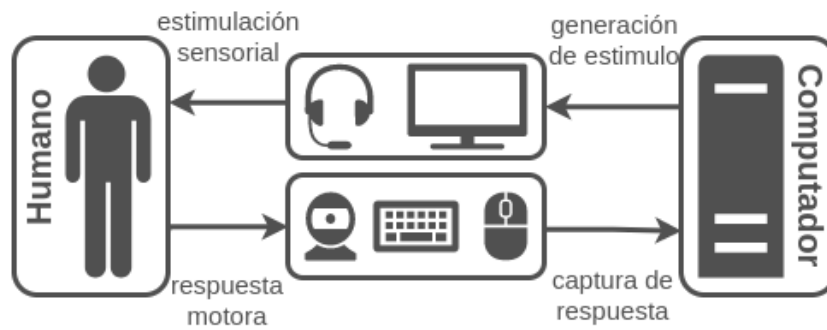


Figura 2.1: Diagrama general de la interfaz hombre-máquina.



Figura 2.2: Ejemplo de setup experimental [1].

2.1. Sistemas de seguimiento ocular

2.1.1. Movimiento ocular

Mirar u observar un objeto es parte fundamental del proceso de visión. Para poder lograr esta acción es necesario orientar los ojos en una determinada dirección con el fin de alinear el eje visual con el objetivo¹. Dicha orientación muchas veces implica movimientos coordinados de los ojos, cuello y cabeza, no obstante movimientos más pequeños son realizados solo por los ojos y son denominados.

Los movimientos sacádicos corresponden a rotaciones rápidas del ojo realiza

2.1.2. Métodos de captura

Un poco de historia

Tecnologías actuales

1. **De contacto directo**
2. **Seguimiento ocular**
3. **Medición de potencial eléctrico**

Comparativa

2.1.3. Sistemas comerciales más relevantes

1. **EyeGaze**
2. **EyeLink**
3. **EyeTribe**
4. **IViewX**
5. **Tobii**

¹Corresponde a la proyección de una línea recta que pasa simultáneamente por el centro de la fovea y la pupila.

2.2. Sistemas de estimulación visual

2.2.1. Hardware de estimulación

Un poco de historia

A lo largo de la historia las tecnologías utilizadas

Tecnologías actuales

1. **Monitores CRT**
2. **Monitores LED, oLED, LCD**

Comparativa

2.2.2. Software de estimulación

Software más relevante

1. **PsychoPy**
2. **PsychoToolbox**
3. **VissionEgg**
4. **Presentation**

Comparativa

2.2.3. Experimentos de estimulación

Capítulo 3

Sistema propuesto

3.1. Alcances

3.1.1. Utilidad de la GUI

3.1.2. Experimentos a implementar

3.2. Tecnologías utilizadas

3.2.1. Hardware

3.2.2. Software

3.3. Diseño e implementación de la GUI

3.3.1. Arquitectura

3.3.2. Archivos de configuración

3.3.3. Script de ejecución

Capítulo 4

Resultados

4.1. Configuración de test de prueba

4.2. Mediciones obtenidas

Capítulo 5

Conclusiones y trabajo futuro

5.1. Conclusiones

5.2. Trabajo futuro

Referencias

- [1] S. MacKenzie, “Evaluating eye tracking systems for computer input,” [consulta: 10 diciembre 2017]. [Online]. Available: <http://www.yorku.ca/mack/cogain.html> VI, 3
- [2] K. R. Gegenfurtner, “The interaction between vision and eye movements,” *Perception*, vol. 45(12), pp. 1333–1357, 2016.
- [3] J. Findlay and R. Walker, “Human saccadic eye movements,” [consulta: 10 diciembre 2017]. [Online]. Available: http://www.scholarpedia.org/article/Human_saccadic_eye_movements
- [4] T. Eggert, “Eye movement recordings: methods,” *Neuro-Ophthalmology*, vol. 40, pp. 15–34, 2007.
- [5] D. C. Richardson and M. J. Spivet, “Eye tracking: Characteristics and methods,” *Encyclopedia of biomaterials and biomedical engineering*, vol. 3, p. 10281042, 2004.
- [6] E. Dalmaijer, “Is the low-cost eyetribe eye tracker any good for research?” *PeerJ Pre-Prints*, vol. 2:e585v1, 2014.
- [7] S. Popelka [et al]., “Eyetribe tracker data accuracy evaluation and its interconnection with hypothesis software for cartographic purposes,” *Computational Intelligence and Neuroscience*, vol. 2016, pp. 1–14, 2016.
- [8] B. Bauer, “A timely reminder about stimulus display times and other presentation parameters on crts and newer technologies,” *Canadian Journal of Experimental Psychology*, vol. 69, pp. 264–273, 2015.
- [9] P. Wang, “An lcd monitor with sufficiently precise timing for research in vision,” *Encyclopedia of biomaterials and biomedical engineering*, vol. 5, 2011.
- [10] T. Elze, T. G. Tanner, and B. Krekelberg, “Temporal properties of liquid crystal displays: Implications for vision science experiments,” *PLoS ONE*, vol. 7(9), 2012.
- [11] N. B. Systems, “Presentation software,” [consulta: 13 junio 2017]. [Online]. Available: <http://www.neurobs.com/>
- [12] J. Peirce, “Psychopy,” [consulta: 13 junio 2017]. [Online]. Available: <http://www.psychopy.org/>

- [13] M. Kleiner, “Psychtoolbox,” [consulta: 13 junio 2017]. [Online]. Available: <http://psychtoolbox.org/>
- [14] A. D. Straw, “Vission egg,” [consulta: 13 junio 2017]. [Online]. Available: <https://visionegg.readthedocs.io/en/latest/index.html>
- [15] J. Pierce, “Generating stimuli for neuroscience using psychopy,” *Frontiers in Neuroinformatics*, vol. 2, p. 10, 2009.
- [16] S. Simpson, “iohub,” [consulta: 23 agosto 2017]. [Online]. Available: <http://www.isolver-solutions.com/iohubdocs/index.html>
- [17] S. Amador [et al]., “Dissociating cognitive deficits involved in voluntary eye movement dysfunctions in parkinson’s disease patients,” *Neuropsychologia*, vol. 44(8), pp. 1475–1482, 2006.
- [18] V. Goyal [et al]., “Saccadic eye movements in parkinson’s disease,” *Indian Journal of Ophthalmology*, vol. 62(5), 2014.

Apéndice A

Código fuente

```
1  %Programa Principal
2  tstart = 0; tend = 20; dt = 0.01;
3  data = simulacion(tstart, tend, dt)
4  visualize_test(data);
5  Graficos(data);

1  # -*- coding: utf-8 -*-
2  # =====
3  # Modules
4  # =====
5  import os
6  import platform as pt
7  from PyQt4 import QtGui
8  import ConfigParser as cp
9
10 # =====
11 # ID Definitions
12 # =====
13 TEST_TYPE = u'.tst'
14 EXPS_TYPE = u'.exp'
15 # =====
16 NOT_VALID_CHARS = [u'', u'.', u'..']
17
18
19 # =====
20 # Config Files Handler
21 # =====
22 class confHandler(cp.ConfigParser):
23     def __init__(self):
24         cp.ConfigParser.__init__(self)
25
26     # ===== File Handler
27     def openConf(self, fileName):
28         fileName = unicode(fileName)
29         if os.path.isfile(fileName):
30             fileItem = open(fileName)
31             self.readfp(fileItem)
32             fileItem.close()
33             return True
34         else:
35             return False
36
```

```

37     def saveConf(self, fileName, isRW=True):
38         fileName = unicode(fileName)
39         isFile = os.path.isfile(fileName)
40         if not isFile or (isFile and isRW):
41             fileItem = open(fileName, 'w')
42             self.write(fileItem)
43             fileItem.close()
44             return True
45         else:
46             return False
47
48     # ===== Data Handler
49     def getSections(self):
50         sections = self.sections()
51         if sections:
52             return sections
53         else:
54             return []
55
56     def getValue(self, section, option, mode=None):
57         if mode is int:
58             return self.getint(section, option)
59         elif mode is float:
60             return self.getfloat(section, option)
61         elif mode is bool:
62             return self.getboolean(section, option)
63         else:
64             return self.get(section, option)
65
66     def setValue(self, section, option, value):
67         try:
68             sections = self.getSections()
69             if section in sections:
70                 self.set(section, option, value)
71             else:
72                 self.add_section(section)
73                 self.set(section, option, value)
74             return True
75         except:
76             return False
77
78     # ===== Children
79     def getConf(self, fileName=u''):
80         pass
81
82     def putConf(self, fileName=u''):
83         pass
84
85
86     # =====
87     # Config File Class
88     # =====
89     class confFile(confHandler):
90         def __init__(self):
91             confHandler.__init__(self)
92             # =====
93             self.fileName = u'./saccadeApp.ini'
94             # =====
95             if pt.system() is 'Windows':
96                 self.secConf = u'ConfigWindows'
97             else:
98                 self.secConf = u'ConfigUnix'
99             # =====
100             self.testDir = u''
101             self.expsDir = u''
102             # =====
103             self.getConfig()
104

```

```

105 # =====
106 def getConfig(self, filename=u''):
107     # =====
108     if self.openConf(fileName=self.fileName) and self.secConf in self.getSections():
109         self.testDir = self.getValue(self.secConf, u'testDir', mode=str)
110         self.expsDir = self.getValue(self.secConf, u'expsDir', mode=str)
111     else:
112         self.testDir = u'Tests/'
113         self.expsDir = u'Experiments/'
114         self.putConfig()
115     # =====
116     if not os.path.isdir(self.testDir):
117         os.mkdir(self.testDir)
118     if not os.path.isdir(self.expsDir):
119         os.mkdir(self.expsDir)
120
121     def putConfig(self, filename=u''):
122         self.setValue(self.secConf, u'testDir', self.testDir)
123         self.setValue(self.secConf, u'expsDir', self.expsDir)
124     # =====
125     self.saveConf(fileName=self.fileName, isRW=True)
126
127 # =====
128 # Test File Class
129 # =====
130
131 class testFileConf(confHandler):
132     def __init__(self, fileName=u''):
133         confHandler.__init__(self)
134         # =====
135         self.fileName = fileName
136         # =====
137         self.imagData = []
138         self.extraFlag = False
139         self.extraImag = u''
140         self.extraKeys = []
141         self.extraCAns = u''
142         # =====
143         if self.fileName is not u'':
144             self.getConfig()
145
146 # =====
147 def getConfig(self, fileName=u''):
148     # =====
149     if fileName is not u'':
150         self.fileName = fileName
151     # =====
152     if self.fileName is not u'' and self.openConf(fileName=self.fileName):
153         self.imagData = self.getImagData()
154         self.extraFlag = self.getValue(u'finalSel', u'extraFlag', mode=bool)
155         self.extraImag = self.getValue(u'finalSel', u'extraImag', mode=str)
156         self.extraKeys = self.getExtraKeys()
157         self.extraCAns = self.getValue(u'finalSel', u'extraCAns', mode=str)
158         return True
159     else:
160         return False
161
162 def putConf(self, fileName=u''):
163     # =====
164     if fileName is not u'':
165         self.fileName = fileName
166     # =====
167     self.setValue(u'mainTest', u'imagData', self.setImagData())
168     self.setValue(u'finalSel', u'extraFlag', self.extraFlag)
169     self.setValue(u'finalSel', u'extraImag', self.extraImag)
170     self.setValue(u'finalSel', u'extraKeys', self.setExtraKeys())
171     self.setValue(u'finalSel', u'extraCAns', self.extraCAns)
172     # =====

```

```

173         return self.saveConf(fileName=self.fileName, isRW=True)
174
175     # =====
176     def setImagData(self):
177         if self.imagData:
178             auxStr = u''
179             for line in self.imagData:
180                 auxStr += u'\n%s\t%s' % tuple([line[0], line[1]])
181             return auxStr
182         else:
183             return u''
184
185     def setExtraKeys(self):
186         if self.extraKeys:
187             auxStr = u''
188             for key in self.extraKeys:
189                 auxStr += u'%s;' % key
190             return auxStr
191         else:
192             return u''
193
194     # =====
195     def getImagData(self):
196         auxStr = self.getValue(u'mainTest', u'imagData', mode=str)
197         if auxStr is not u'':
198             auxOut = []
199             auxArr = auxStr.split(u'\n')
200             for line in auxArr:
201                 try:
202                     auxLne = line.split(u'\t')
203                     auxOut.append([auxLne[0], float(auxLne[1])])
204                 except:
205                     pass
206             return auxOut
207         else:
208             return []
209
210     def getExtraKeys(self):
211         auxStr = self.getValue(u'finalSel', u'extraKeys', mode=str)
212         if auxStr is not u'':
213             auxOut = []
214             auxArr = auxStr.split(u';')
215             for line in auxArr:
216                 if line is not u'':
217                     auxOut.append(line)
218             return auxOut
219         else:
220             return []
221
222
223     # =====
224     # Experiment File Class
225     # =====
226     class expsFileConf(confHandler):
227         def __init__(self, fileName=u''):
228             confHandler.__init__(self)
229             # =====
230             self.fileName = fileName
231             # =====
232             self.testData = []
233             self.randFlag = False
234             self.restFlag = False
235             self.restTest = 0
236             self.restTime = 0.0
237             # =====
238             if self.fileName is not u'':
239                 self.getConf()
240

```

```

241 # =====
242 def getConf(self, fileName=u''):
243     # =====
244     if fileName is not u'':
245         self.fileName = fileName
246     # =====
247     if self.fileName is not u'' and self.openConf(fileName=self.fileName):
248         self.testData = self.getTestData()
249         self.randFlag = self.getValue(u'testSort', u'randFlag', mode=bool)
250         self.restFlag = self.getValue(u'testRest', u'restFlag', mode=bool)
251         self.restTest = self.getValue(u'testRest', u'restTest', mode=int)
252         self.restTime = self.getValue(u'testRest', u'restTime', mode=float)
253         return True
254     else:
255         return False
256
257 def putConf(self, fileName=u''):
258     # =====
259     if fileName is not u'':
260         self.fileName = fileName
261     # =====
262     self.setValue(u'mainExps', u'testData', self.setTestData())
263     self.setValue(u'testSort', u'randFlag', self.randFlag)
264     self.setValue(u'testRest', u'restFlag', self.restFlag)
265     self.setValue(u'testRest', u'restTest', self.restTest)
266     self.setValue(u'testRest', u'restTime', self.restTime)
267     # =====
268     return self.saveConf(fileName=self.fileName, isRW=True)
269
270 # =====
271 def setTestData(self):
272     if self.testData:
273         auxStr = u''
274         for line in self.testData:
275             auxStr += u'\n%s\t%s' % tuple([line[0], line[1]])
276         return auxStr
277     else:
278         return u''
279
280 # =====
281 def getTestData(self):
282     auxStr = self.getValue(u'mainExps', u'testData', mode=str)
283     if auxStr is not u'':
284         auxOut = []
285         auxArr = auxStr.split(u'\n')
286         for line in auxArr:
287             try:
288                 auxLne = line.split(u'\t')
289                 auxOut.append([auxLne[0], float(auxLne[1])])
290             except:
291                 pass
292         return auxOut
293     else:
294         return []
295
296
297 # =====
298 # Experiment File Class
299 # =====
300 def getFilePath(path, ext, newName=u'', oldName=u'', isRW=False):
301     newName = unicode(newName) if newName is not None else u''
302     oldName = unicode(oldName) if oldName is not None else u''
303     auxName = oldName
304     newPath = None
305     # =====
306     isFirst = True
307     isReady = False
308     # =====

```



```

309 while not isReady:
310     if isFirst and newName != u'':
311         isFirst = False
312         isOk = True
313     else:
314         isFirst = False
315         newName, isOk = QtGui.QInputDialog.getText(None, u'New file name', u'Insert new
file name:', text=auxName)
316         newName = unicode(newName)
317         # =====
318         if isOk:
319             newPath = unicode(path + u'/' + newName + ext)
320             isNewPathOk, isNewPathExist = isPathAvailable(newPath)
321             # =====
322             if isNewPathOk and newName not in NOT_VALID_CHARS and len(newName) >= 4:
323                 if isRW and isNewPathExist and newName != oldName:
324                     print u'Error: This file name affects not-involved files.'
325                     QtGui.QMessageBox.about(None, u'Error!', u'This file name affects\nnot-
involved files')
326                 elif not isRW and isNewPathExist:
327                     print u'Error: Cannot copy to a existing file.'
328                     QtGui.QMessageBox.about(None, u'Error!', u'Cannot copy to a existing
file')
329                 else:
330                     isReady = True
331             else:
332                 if len(newName) < 4:
333                     print u'Error: file name too short'
334                     QtGui.QMessageBox.about(None, u'Error!', u'Please use a name larger\
n than 4 letters.')
335                 else:
336                     print u'Error: Bad file name'
337                     QtGui.QMessageBox.about(None, u'Error!', u'Bad file name')
338                 # =====
339                 auxName = newName
340             else:
341                 print u'Operation cancelled.'
342                 newName = None
343                 newPath = None
344                 isReady = True
345             # =====
346             return newName, newPath
347
348
349 def isPathAvailable(path):
350     # Return: [nameOk, exists]
351     if not os.path.isfile(path):
352         try:
353             auxFile = open(path, 'w')
354             auxFile.close()
355             os.remove(path)
356             return True, False
357         except:
358             return False, False
359     else:
360         return True, True
361     pass

```

Apéndice B

Vistas de la GUI