



Escuela
Politécnica
Superior

Entorno de interacción Hombre-Máquina mediante sistema multisensor 3D



Grado en Ingeniería Informática

Trabajo Fin de Grado

Autor:

John Alejandro Castro Vargas

Tutor:

José García Rodríguez

Junio 2016



Universitat d'Alacant
Universidad de Alicante

UNIVERSIDAD DE ALICANTE

Grado en ingeniería informática

Trabajo Fin de Grado

**Entorno de interacción Hombre-Máquina
mediante sistema multisensor**

Autor

John Alejandro Castro Vargas

Supervisor

José García Rodríguez

Departamento de Tecnología Informática y Computación (DTIC)

Alicante, Junio 20, 2016

Resumen

En la actualidad son muchas las personas que padecen de discapacidades físicas o cognitivas que necesitan la atención y ayuda de otros en su actividad diaria. Estos pacientes necesitan seguir tratamientos terapéuticos para mejorar sus capacidades motoras o cognitivas. Por ese motivo hay un fuerte interés en proporcionarles una asistencia permanente y con el mínimo coste, minimizando la intervención de terapeutas o cuidadores. Bajo esta situación es de donde surgen proyectos que buscan ofrecer a personas dependientes ayuda, seguimiento o tratamientos para mejorar su calidad de vida.

Entre los proyectos que buscan este objetivo podemos distinguir aquellos que lo realizan aplicando una asistencia con medios humanos y otros que contemplan el uso de la tecnología. En este último grupo es en el que se centran los objetivos de este proyecto, donde haremos uso de un dispositivo de seguimiento ocular y una Kinect, para reconocer gestos realizados con los ojos o las manos, permitiéndonos interactuar sobre un sistema en el que poder realizar ejercicios de rehabilitación cognitiva.

Para ello implementaremos un sistema de reconocimiento de gestos que utilice los joints o articulaciones proporcionados por la KinectV2. Con los joints obtenidos de las manos, reconoceremos las acciones que quiera realizar el usuario y ofreceremos la posibilidad de personalizar los gestos.

Para pacientes con un grado de discapacidad mayor y no puedan utilizar el sistema mediante gestos realizados con las manos, utilizaremos un dispositivo de seguimiento ocular para reconocer guiños o pestañeos largos como gestos de acción.

Por último realizaremos una pequeña aplicación de rehabilitación cognitiva, que consistirá en un puzzle que podrá ser resuelto utilizando cualquiera de los dispositivos.

Agradecimientos

Mientras escribo estas frases me recuerdo cuatro años atrás viéndome llenar la inscripción para Ingeniería Informática. Recuerdo que estaba ansioso, con ganas de aprender y adquirir habilidades que me permitiesen dar forma a cualquier cosa que imaginase. A día de hoy siento que he conseguido mis objetivos pero sigo ansioso y con ganas de aprender mucho más.

En primer lugar quiero agradecer a mi supervisor Jose, de quien he sentido un gran apoyo y con quien he podido mejorar mis habilidades. Sinceramente muchas gracias.

Me gustaría agradecer a los compañeros del DTIC por las sugerencias y ayudas prestadas, en especial a Francisco García y Albert. Y de DCCIA agradecer a Francisco Gómez por aguantar mis incansables preguntas.

Gracias a todos los compañeros y amigos que me habéis ayudado de manera directa e indirecta estos cuatro años. En especial a Felix, Paco (Fran), Pedro y Javier Hernández a quienes admiro mucho y han sido fuente de inspiración y motivación.

Por último quiero dar gracias a mis padres y a mi novia Libertad por aguantar mis épocas de stress con exámenes y animarme constantemente a mirar hacia adelante y no rendirme.

Muchas gracias a todos por ayudarme a crecer como persona y como profesional.

Contenidos

Lista de figuras	ix
Lista de tablas	xi
1 Introducción	1
1.1 Motivación	1
1.2 Trabajos relacionados	2
1.2.1 Seguimiento ocular	3
1.2.1.1 Registro por el movimiento de un objeto unido al ojo	4
1.2.1.2 Rastreo óptico sin contacto directo con el ojo	4
1.2.1.3 Potenciales Eléctricos	5
1.2.1.4 Aplicaciones	6
1.2.1.5 Dispositivos de seguimiento ocular	7
1.2.2 Detección de gestos con las manos	9
1.2.2.1 Guantes	9
1.2.2.2 Basados en visión	10
1.2.2.3 Dispositivos	11
1.3 Propuesta	12
1.4 Objetivos	13
1.5 Estructura	13
2 Metodología	15
2.1 Dispositivos	15
2.1.1 Kinect	15
2.1.2 Tobii EyeX	17
2.2 Herramientas Software	18
2.2.1 Qt	18
2.2.2 EyeX SDK	19
2.2.3 Kinect SDK 2.0	21
2.3 Técnicas aplicadas	22
2.3.1 Cambio de origen de coordenadas	22
2.3.2 Dynamic Time Warping	23
3 SIRMAVED	27
4 Reconocimiento de movimientos oculares	29
4.1 Desarrollo	29

4.1.1	Interfaz de personalización	29
4.1.2	Proceso de reconocimiento ocular	30
4.2	Experimentación	31
5	Reconocimiento de gestos con Kinect	33
5.1	Desarrollo	33
5.1.1	Interfaz de personalización	34
5.1.2	Transformación de coordenadas	34
5.1.3	Entrenamiento de gestos	36
5.1.4	Reconocimiento de gestos	36
5.2	Experimentación	37
6	Aplicación de rehabilitación: Puzzle	41
6.1	Desarrollo	41
6.1.1	Lógica de juego	41
6.1.2	Interfaz	42
7	Conclusión	45
7.1	Trabajos futuros	46
	Bibliografía	48

Lista de figuras

1.1	Robot Taizo	2
1.2	Funcionamiento con tambor ahumado giratorio	4
1.3	Headset EyesTracker	5
1.4	EyesTracking con potenciales eléctricos	5
1.5	Mapa de calor	6
1.6	Control mediante "EyesTracking"	7
1.7	Tobii EyeX	8
1.8	VT3 Mini	8
1.9	D7 Remote Desktop EyeTracking	9
1.10	Guantes con acelerómetros y flex sensors	10
1.11	LeapMotion	11
1.12	Kinect	12
1.13	Cibertouch	12
1.14	Planificación	13
2.1	Sensores de la Kinect	16
2.2	Sensores del TobiiEyeX	17
2.3	Qt Creator	19
2.4	EyeX SDK Querys	20
2.5	EyeX SDK Eventos	20
2.6	Dynamic Time Warping	24
3.1	Plataforma móvil	27
4.1	Personalización EyeX	29
4.2	Foco EyeX	30
5.1	Ventana deslizante	34
5.2	Configuración gestos Kinect	34
5.3	Esqueletos con Kinect	38
6.1	Puzzle	42
6.2	Puzzle win	43

Lista de figuras

Lista de tablas

2.1	Características Kinect v1 y v2	16
2.2	Características Tobii EyeX	17
2.3	Dynamic Time Warping	24
4.1	Distancia límites EyeX 1	31
4.2	Distancia límites EyeX 2	31
5.1	Acierto gestos Kinect	38
5.2	Acierto gestos Kinect Persona 2	39

Lista de tablas

1

Introducción

Son muchas las personas que hoy en día tienen discapacidades que les impiden llevar a cabo tareas cotidianas. Esta situación impide que puedan desarrollar determinadas actividades de manera independiente y necesiten la ayuda de otros para realizarlas. Algunas discapacidades pueden necesitar de un apoyo diario y rehabilitación, pero una prestación de rehabilitación constante puede suponer gastos elevados para las personas cuidadoras, provocando que no todas las personas que necesiten este servicio puedan acceder a él. En esta situación se propone utilizar medios tecnológicos para ofrecer un servicio de rehabilitación cognitiva con la finalidad de garantizar un servicio digno para aquellas personas que lo necesiten.

Los ejercicios de rehabilitación que debe realizar el paciente consisten en pequeños desafíos que deberá resolver. Debido a que el paciente puede presentar dificultades motoras, el sistema de rehabilitación debe estar capacitado para interactuar con él mediante gestos simples o la vista.

En este proyectoaremos uso de dos dispositivos, un dispositivo de seguimiento de los ojos y cámara RGB-D, que permitan que personas con discapacidades cognitivas o motoras puedan interactuar sobre un sistema que está siendo desarrollado por los profesores del Instituto Universitario de Investigación en Informática: José García Rodríguez y Miguel Ángel Cazorla Quevedo que tiene como objetivo dar asistencia a personas con discapacidades motoras o cognitivas.

1.1 Motivación

Este proyecto formará parte de un sistema multisensor desarrollado en el contexto del proyecto nacional denominado Sirmaved Desarrollo de un Sistema Integral Robótico de Monitorización e Interacción para Personas con Daño Cerebral Adquirido y Dependientes donde se hará uso de varios dispositivos para conseguir un sistema completo que permita la rehabilitación de pacientes con daño cerebral adquirido y/o dependientes.

Las personas con discapacidades que hagan uso del sistema pueden presentar problemas de movilidad que impidan su uso. Por ese motivo el sistema necesita estar dotado de varios dispositivos que permitan que en caso de que un paciente no pueda usar las manos para realizar los ejercicios, pueda usar un método alternativo. Del mismo modo, debido a que pueden presentar problemas de movilidad reducida, el sistema

debe ser capaz de adaptarse a los movimientos que pueda realizar el paciente.

1.2 Trabajos relacionados

En países como Japón se puede observar un especial interés en el uso de tecnologías para mejorar la calidad de vida de personas dependientes y/o ancianas. En ese país se plantea como robots sociales que proporcionan compañía a las personas y a su vez puedan ayudar en tareas diarias. Pero las funcionalidades que se quieren integrar en estos robots no se reducen a compañía y tareas, también hay interés en mejorar el estado físico de las personas mediante entrenamientos dedicados y personalizados.

En lo referente a entrenamiento para personas mayores podemos encontrar el robot Taizo (Figura 1.1) que está siendo desarrollado por el Instituto Nacional de Ciencia y Tecnología Industrial en Japón [Matsusaka et al., 2009]. Este robot tiene como objetivo ayudar a las personas mayores a realizar ejercicios para mejorar sus articulaciones.



Figura 1.1: Robot Taizo. (Imagen extraída de [Matsusaka et al., 2009])

Por otro lado, en el Instituto de Rehabilitación de la Universidad de Slovenia se realizó un trabajo [Debeljak et al., 2012] utilizando un dispositivos de seguimiento ocular en dos pacientes con Atrofia Muscular Espinal quienes solo podían realizar movimientos oculares. En él se demuestra que el uso de estos dispositivos puede ayudar a la comunicación e independencia de personas con esta enfermedad.

En lo referente a nuestro problema que es "la rehabilitación de personas con discapacidades motoras o cognitivas" se necesita de varios sensores para captar los gestos del usuario. Para usuarios con problemas en sus funciones motoras que le

impidan utilizar el sistema con gestos realizados con las manos será necesario el uso de un dispositivo de seguimiento ocular que le permita interactuar con él. En caso de que puedan realizar movimientos limitados con las manos se hará uso de un sistema de captura de gestos para identificar la acción a realizar por el usuario.

Partiendo de estos dos problemas en lo referente a la interacción con el sistema, realizaremos un pequeño análisis del marco teórico del seguimiento ocular junto con las técnicas utilizadas para realizar el rastreo de los ojos (Sección 1.2.1) y las técnicas utilizadas para la detección de gestos (Sección 1.2.2).

1.2.1 Seguimiento ocular

Los estudios relacionados con el seguimiento ocular no son un hecho reciente. En el siglo XIX surge el interés por conocer el modo en que se desplazan los ojos ante aquello en lo que centramos nuestra atención. En aquella época el interés nace ante las preguntas, *Cuando leemos... ¿En qué palabras se detienen nuestros ojos? ¿Por cuánto tiempo? o ¿Cuándo regresan de nuevo a las palabras ya vistas?*. El interés por estas preguntas se debe a que inicialmente se creía que los ojos realizaban un suave recorrido en la lectura, pero posteriormente se demostró que realmente realizan paradas cortas y movimientos rápidos durante la lectura.

En la actualidad la motivación en el estudio del "EyesTracking" se debe principalmente a intereses en marketing. La vista es una importante puerta de entrada de información del exterior a nuestro cerebro, de ahí que haya interés en estudiar y comprender el modo en que dirigimos nuestros ojos ante elementos que captan nuestra atención. En este sector, el objetivo es aprender la mejor manera de presentar un producto introduciendo información visual que impacte en el consumidor aumentando las probabilidades de venta del producto.

En [Hassan Montero and Herrero Solana, 2007] exponen que los trabajos relacionados con el *Seguimiento ocular* han avanzado lentamente en lo referente a interacción hombre-máquina. Pero gracias a las nuevas soluciones comerciales que han aparecido en los últimos años con precios más bajos, cada vez es más accesible el disfrute de nuevas experiencias de usuarios y ayuda a personas con problemas de accesibilidad.

Dentro de este apartado vamos a dividir en tres categorías distintas las técnicas que han sido utilizadas para conseguir registrar los movimientos realizados por los ojos. Estas técnicas pueden distinguirse por: *Registrar el movimiento de un objeto unido al ojo* (Sección 1.2.1.1), *Obtener el rastreo del ojo sin contacto directo con este* (Sección 1.2.1.2) o *Potenciales eléctricos* (Sección 1.2.1.3). Mencionaremos aplicaciones en las que son utilizadas hoy en día estas técnicas (Sección 1.2.1.4) y ejemplos de dispositivos comerciales (Sección 1.2.1.5).

1.2.1.1 Registro por el movimiento de un objeto unido al ojo

Los primeros intentos para rastrear y analizar el seguimiento de los ojos entran dentro de esta categoría, en este tipo de dispositivos, se introducían pequeños objetos como lentes en el ojo y se les conectaba a estas lentes algún elemento más como palancas que permitiesen registrar el movimiento en un medio externo. [Wade and Tatler, 2009]

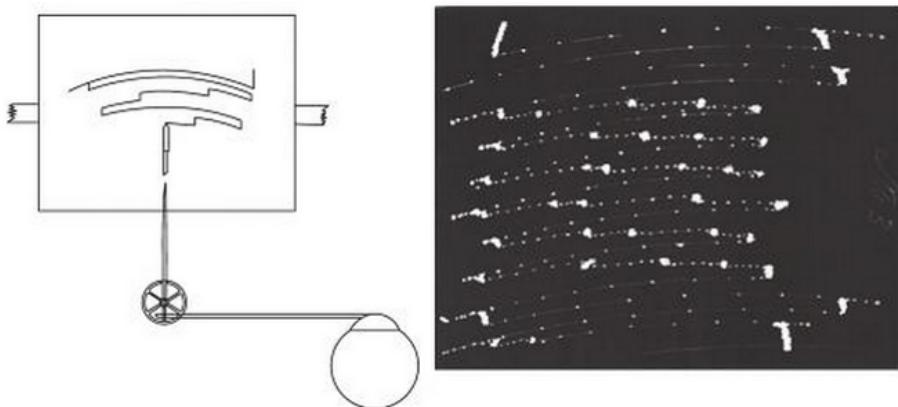


Figura 1.2: La imagen muestra un esquema del funcionamiento del tambor giratorio y su resultado. (Imagen extraída de ethnos.co.za)

Científicos del siglo XIX fueron los primeros en aplicar técnicas de quimografía (Figura 1.2) para intentar realizar una representación gráfica del movimiento del ojo. La técnica consistía en introducir en el ojo una pequeña lente de marfil con un filamento unido, un tambor giratorio ahumado se encargaba de registrar las oscilaciones del filamento al desplazarse el ojo.

1.2.1.2 Rastreo óptico sin contacto directo con el ojo

Los dispositivos de seguimiento ocular que abarcan esta categoría son aquellos que permiten captar el movimiento colocar dispositivos sobre el ojo para realizar el rastreo. Estos dispositivos normalmente hacen uso de cámaras y luces infrarrojas para analizar el movimiento de los ojos como se observa en [Hassan Montero and Herrero Solana, 2007], se distingue dos grupos en este apartado, dispositivos que se colocan en la cabeza y otros que funcionan a distancia.

EyesTracker Headset

Estos dispositivos resultan útiles cuando se necesita libertad de movimiento por parte del usuario del que se está obteniendo información de la mirada (Figura 1.3). El mecanismo es similar a los dispositivos que funcionan a distancia. Se monta una cámara y un emisor infrarrojo para captar el reflejo en la pupila y la córnea. Las imágenes captadas se analizan para deducir mediante el reflejo en los ojos la posición hacia la que dirige la mirada. Como ventaja tenemos que el dispositivo puede rastrear nuestra mirada sin limitarnos en espacio o posición delante del dispositivo.

Por otra parte, a pesar de que puede ser transportable al estar acoplado a la cabeza, el uso en exteriores puede verse afectado por la luz solar.



Figura 1.3: Ejemplo de EyesTracker de bajo coste acoplado en la cabeza (Imagen extraída de [Li et al., 2006])

Dispositivos a distancia

En este grupo los dispositivos usados para rastrear la mirada pueden usarse a distancia, pero se limita el área de uso. Debido a que la cámara y el emisor de infrarrojos no están acoplados al usuario resulta necesario que cuando se utilice se sitúen a una distancia suficiente para que pueda detectar los ojos. Como ventaja el usuario no tiene que llevar encima objetos que le puedan resultar incómodos (Figura 1.7).

1.2.1.3 Potenciales Eléctricos

Este método consiste en la utilización de electrodos alrededor del ojo para captar las diferencias de potenciales eléctricos de la piel.



Figura 1.4: (Imagen extraída de [Bulling et al., 2011])

Esta técnica tiene como ventaja que no necesita mucha capacidad de cómputo para ser procesada, siendo posible alcanzar frecuencias de muestreo de hasta 1000 Hz.

1. Introducción

Por otro lado puede trabajar en distintas situaciones de luminosidad, lo que lo hace perfecto para ser utilizado con la luz solar o mientras se duerme.

1.2.1.4 Aplicaciones

Uno de los problemas de los dispositivos de seguimiento ocular es que tradicionalmente resultaban poco accesibles con costes muy elevados, donde sus utilidades se limitaban a la investigación. Los precios de estos dispositivos han descendido a un punto más accesible y sus posibles aplicaciones comienzan a tener más relevancia. En este apartado comentaremos dos aplicaciones en los que nos podemos encontrar el uso de estos dispositivos: Mapas de calor y Aplicaciones de accesibilidad.

Mapas de calor

Los mapas de calor aplicados en al seguimiento ocular se basan en representaciones gráficas de las zonas en las que ha prestado atención un usuario. Estas representaciones muestran mediante colores las zonas que han sido observadas, donde los colores mas cálidos son los puntos que más interés han tenido para el usuario, y los colores fríos los que han tenido menos interés.

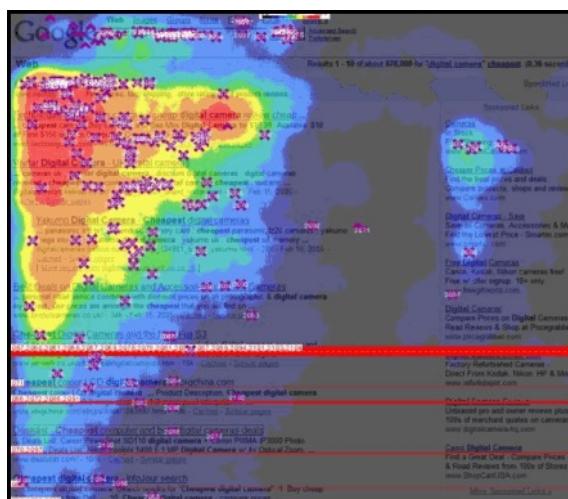


Figura 1.5: Este mapa de calor representa lo puntos en los que más se ha fijado una persona al realizar una búsqueda en google.(Imagen extraída de: ¹⁾)

Estos mapas de calor en ocasiones son usados en páginas web para encontrar aquellos puntos en los que mas se fija el usuario. Conociendo los principales puntos de interés, se modifica la interfaz para ofrecer el mejor contenido en las zonas que tienen más relevancia. Un ejemplo de esto es lo ocurrido en las búsquedas en google. Tal y como observamos en [Hof, 2015], se menciona un estudio realizado en el 2005 en el que utilizando técnicas de "EyesTracking" se genera un mapa de calor y se extrae como conclusión que la zona de mayor interés en una búsqueda en google, es el triángulo superior izquierdo o "Golden Triangle" y por tanto la zona que mayor valor tiene a la hora de publicar anuncios. A pesar de ello en el artículo se desmiente la existencia

¹⁾<http://www.parachutedigitalmarketing.com.au/tag/eye-tracking-heat-maps/>(03/2016)

del "Golden Triangle" (Figura 1.5), pues estudios más recientes afirman que esto no sucede del modo que afirmaba el estudio realizado en el 2005, y en realidad el usuario realiza un barrido rápido de abajo arriba de los resultados de la búsqueda. En cualquier caso, para poder realizar estos estudios fue necesario la utilización de dispositivos de seguimiento ocular y mapas de calor para extraer conclusiones.

Accesibilidad

Existen sistemas que permiten controlar el ordenador utilizando únicamente los ojos. Estos sistemas están enfocados principalmente para personas con discapacidades que tienen dificultades para interactuar de manera tradicional con ordenadores.

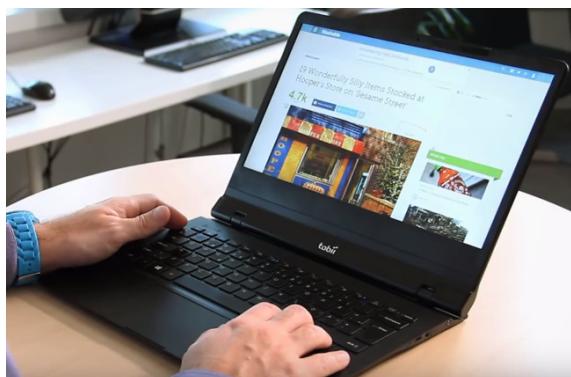


Figura 1.6: Ordenador siendo controlado mediante una tecla de acción y tobii eye.(Imagen extraída del sitio oficial Tobii)

Un ejemplo de sistema y dispositivo para el control del ordenador es el proporcionado por la empresa Tobii Tech (Figura 1.6). Este dispositivo permite ser acoplado en distintos ordenadores, y utiliza una técnica basada en visión para trazar el movimiento ocular. Este dispositivo viene acompañado de un software con el que podemos controlar el ordenador utilizando nuestros ojos para señalar un punto de acción y pulsar una tecla para ejecutar la acción, como puede ser un click de ratón. Este método es una pequeña aproximación al ideal del control total del ordenador utilizando únicamente los ojos, pero la empresa pone a disposición un SDK para desarrollar aplicaciones específicas que hagan uso del dispositivo, lo que permite ampliar las funcionalidades que tiene por defecto y crear sistemas específicos y adaptados para personas con discapacidades concretas.

1.2.1.5 Dispositivos de seguimiento ocular

Como dijimos anteriormente los dispositivos de seguimiento ocular tradicionalmente no han sido muy accesibles, reduciéndose su uso al campo de la investigación. Hoy en día podemos encontrar varias soluciones que se pueden adquirir y comentaremos unos pocos ejemplos de estos dispositivos.

Tobii EyeX

El Tobii EyeX (Figura 1.7) es desarrollado por la empresa sueca Tobii. Aplica la técnica de visión con luces de infrarrojo comentada anteriormente.

El dispositivo se puede colocar debajo de una pantalla y después de ser calibrado permitirá interactuar con ella haciendo uso de una tecla de acción a configurar. La tecla de acción es una tecla del teclado que aplicará una acción donde estén observando nuestros ojos.

La empresa proporciona una herramienta de desarrollo, lo que permite realizar aplicaciones específicas para el dispositivo.

El precio de este dispositivo oscila en los 100 euros, es el más accesible en el mercado y en los últimos años ha ganado bastante popularidad. La relevancia que ha conseguido, y su precio, llevó a empresas como "Mountain" a construir un portátil con el dispositivo integrado, debido a que se comienza a observar interés por empresas desarrolladoras de juegos en estos dispositivos.



Figura 1.7: Dispositivo de seguimiento ocular con funcionamiento a distancia (Imagen extraída de [Tech, 2001])

VT3 mini

El VT3 mini (Figura 1.8) es un dispositivo construido por la empresa "EyeTech Digital Systems". El dispositivo destaca por su tamaño, su facilidad de anclaje a portátiles y por su fiabilidad. El VT3 mini cuenta con la última tecnología "Aeye", desarrollada por EyeTech. Esta tecnología dota al dispositivo de especial resistencia a cambios bruscos al mover la cabeza, lo que se traduce en un rastreo muy robusto.



Figura 1.8: Dispositivo VT3 mini.(Imagen extraída del sitio oficial EyeTech Digital Systems)

D7 Remote Desktop EyeTracking Optics

Este dispositivo (Figura 1.9) es creado por ASL(Applied Science Laboratories) y tiene como principales características: tener una tasa de refresco de 60 Hz, integrar

reconocimiento facial y permitir la detección de los ojos en un área de un metro cuadrado en frente del dispositivo.

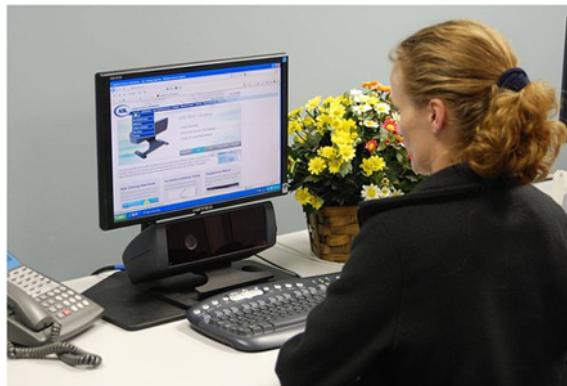


Figura 1.9: Dispositivo D7 Remote Desktop EyeTracking.(Imagen extraída de <http://www.est-kl.com>)

1.2.2 Detección de gestos con las manos

La detección de gestos es la capacidad que se le otorga a un computador para identificar los gestos realizados por una persona. En este punto hay que distinguir entre gestos estáticos y gestos en movimiento [Wu and Huang, 1999].

Gestos estáticos suponen reconocer el estado de la mano en un momento concreto y establecer una clasificación, por ejemplo tener la mano cerrada, la mano abierta o tener levantado un dedo concreto. Por otro lado los gestos en movimiento suponen la captura de varios estados de la mano en un periodo de tiempo, donde en función de la técnica utilizada para la captura de los gestos, podremos obtener información de solo la posición de la mano o también en información de los dedos. La información obtenida es procesada para distinguir el gesto que ha sido realizado en ese periodo de tiempo, por ejemplo desplazar la mano hacia a la izquierda.

Existen dos modos principalmente usados para la captura de gestos con las manos, "Guantes" y "Basados en visión". En este apartado comentaremos los métodos que hemos mencionado anteriormente que son utilizados para capturar gestos con las manos: Guantes(Sección: 1.2.2.1) y Basados en visión(Sección: 1.2.2.2). También comentaremos algunos de los dispositivos que podemos encontrar para realizar la captura de gestos(Sección: 1.2.2.3).

1.2.2.1 Guantes

Este grupo de técnicas se basan en el uso de sensores en un guante para detectar las posiciones de los dedos, de la palma de la mano o de la propia muñeca. Entre los sensores utilizados para obtener información de las distintas posiciones podemos encontrar el uso de acelerómetros [Kim et al., 2009] y flex sensors [Díaz-Tribaldos et al., 2015]. Haciendo uso de los acelerómetros(Figura: 1.10a) se puede obtener información del desplazamiento de los dedos y la posición de la palma de la mano para

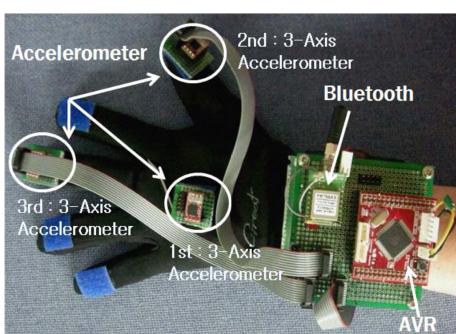
1. Introducción

obtener sus posiciones en un espacio tridimensional. Con los flex sensors obtenemos diferencias en las resistencias de los sensores en función de la curvatura experimentada, lo que nos permite obtener un ángulo de flexión en los dedos.

1.2.2.2 Basados en visión

Las técnicas basadas en visión consisten en la utilización de cámaras para el reconocimiento de los gestos. Este modo de realizar el reconocimiento conlleva varios problemas a abordar, como son: el sistema debe ser invariante ante el fondo de la imagen, insensibilidad a la luz y problemas de rendimiento al procesar las imágenes [Garg et al., 2009]. En ocasiones se puede acompañar el sistema basado en visión mediante guantes con colores o formas especiales que permitan una detección más sencilla por parte del software que analiza las imágenes tomadas por la cámara [Wang and Popović, 2009].

Estas técnicas pueden implementarse utilizando únicamente una cámara o acompañándola con un emisor infrarrojo. En el caso de usar solo una cámara, la información que recibiremos se verá limitada en profundidad [Wu and Huang, 1999], haciendo que la captura de gestos que podamos procesar se reduzca a dos dimensiones. Para obtener información de profundidad, se pueden aplicar técnicas de triangulación estéreo utilizando una segunda cámara que funcione en combinación con la primera. Otro modo es utilizar la cámara en combinación con luces infrarrojas para captar la variación en las intensidades de la luz infrarroja recibida.



(a) Guante con acelerómetros



(b) Guante con flex sensors

Figura 1.10: Guantes con los sensores usados para la captura del movimiento de la mano y/o dedos. (a) Guante con acelerómetros para la detección del movimiento de los dedos(Extraída de [Kim et al., 2009]). (b) Ejemplo de guante que hace uso de flex sensors para detectar la movilidad de los dedos(Extraída de [Díaz-Tribaldos et al., 2015]).

1.2.2.3 Dispositivos

En el mercado se pueden encontrar varias opciones para la captura de gestos con las manos, tanto dispositivos que utilizan técnicas basadas en visión como guantes. En este apartado mencionaremos algunos de estos dispositivos.

LeapMotion

El LeapMotion(Figura: 1.11) es un dispositivo al que se puede acceder por un coste muy bajo, desde los 40 euros a los 70 euros. Este dispositivo posee 25 grados de libertad, lo que permite reconocer una gran variedad de posiciones de las manos.

El dispositivo funciona mediante dos cámaras y emisores infrarrojos. Las cámaras capturan la luz infrarroja recibida por los emisores infrarrojos y mediante algoritmos de visión estereoscópica obtiene información de profundidad. El procesamiento de la imagen es realizado por el ordenador y no por el procesador del dispositivo, permitiendo que este sea pequeño y ligero, pero que el ordenador que lo utilice vea mermado su rendimiento.



Figura 1.11: Dispositivo LeapMotion utilizado para capturar gestos realizados con las manos(Imagen extraída del sitio oficial LeapMotion).

Kinect

Kinect(Figura: 1.12) es un dispositivo comercializado por Microsoft con un precio alrededor de los 140 euros.

El dispositivo dispone de un emisor infrarrojo y dos cámaras: una infrarroja y otra rgb. El emisor infrarrojo proyecta un patrón que es capturado por la cámara correspondiente, el dispositivo procesa la imagen para detectar los cambios producido en el patrón al rebotar el haz de luz infrarroja con objetos del entorno y estimar una profundidad para los elementos de la imagen.

El dispositivo dispone de un completo SDK que permite obtener varios tipos de información para ser utilizadas. Se puede obtener tanto imágenes de color como imágenes con profundidad. El propio SDK también nos permite obtener posiciones de joints que representarán distintas articulaciones de la persona que se encuentre en frente del dispositivo.



Figura 1.12: Dispositivo Kinect de la empresa Microsoft para sus consolas xbox.(Imagen extraída de msd.microsoft.com)

CyberTouch

Cybertouch(Figura 1.13) es un guante háptico que permite capturar los movimientos de dedos y manos. El guante también es capaz de ofrecer una respuesta física mediante vibraciones al interactuar con un elemento virtual.

Este guante captura información del estado de la mano 90 veces por segundos, puede reconocer tanto posición de los dedos como de las manos.

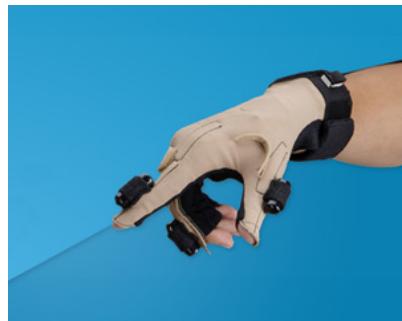


Figura 1.13: Guante de la empresa Cyber Glove Systems que permite obtener información de gestos realizados con las manos.(Imagen extraída de la web oficial de Cyber Glove Systems)

1.3 Propuesta

Después de describir la motivación de este trabajo y analizar el marco teórico del reconocimiento de gestos con las manos y el seguimiento ocular, proponemos el uso de un tobii eyeX y una KinectV2 para llevar a cabo el proyecto.

Mediante el TobiiEyeX daremos acceso al sistema a aquellas personas que no puedan utilizar las manos, y en aquellos que puedan, utilizaremos la KinectV2 para realizar capturas de los joins de las manos y detectar gestos simples con los que puedan interactuar con el sistema. Por otro lado implementaremos un pequeño ejercicio de rehabilitación cognitiva que constará de un pequeño juego de puzzle que deberá resolver utilizando el dispositivo que mejor se adapte a su condición física.

1.4 Objetivos

El principal objetivo de este proyecto es aportar al sistema SIRMAVED las funcionalidades que le faltan en lo referente a la interacción con el sistema e implementar uno de los ejercicios de rehabilitación. Para ello será necesario que el sistema identifique gestos simples realizados con las manos y pequeños gestos realizados con los ojos, como pueden ser un guiño o un parpadeo largo. Para conseguirlo será necesario que el sistema identifique gestos en movimiento capturados con KinectV2 y que el usuario pueda definir los gestos para las acciones. Del mismo modo necesitaremos que el usuario pueda elegir el gesto a realizar con los ojos para activar un evento.

La figura 1.14 muestra los tiempos estimados para cada apartado en la realización del proyecto. El desarrollo de la memoria y la implementación se plantea de una manera paralela, donde se desarrolle la memoria a medida que se desarrolla el proyecto.



Figura 1.14: Planificación para la realización del proyecto.

1.5 Estructura

Este documento está estructurado de la siguiente manera: El capítulo 1 muestra el marco teórico de los problemas expuestos en este proyecto. Se define la motivación y los objetivos concretos. El capítulo 2 presenta la metodología aplicada en el desarrollo y se describen las herramientas y técnicas aplicadas. El capítulo 3 describe el funcionamiento del sistema sobre el que trabajaremos. Los capítulos 4, 5 y 6 corresponden con el desarrollo de cada uno de los objetivos fijados en nuestro proyecto. En cada apartado comenzaremos con un breve definición del objetivo a conseguir, describiremos el desarrollo y finalizaremos con los resultados. Por último, el capítulo 7 detallará las conclusiones extraídas de nuestro proyecto.

1. Introducción

2

Metodología

Para desarrollar el proyecto necesitaremos utilizar varias herramientas hardware y software que nos permitirán alcanzar los objetivos definidos. En este apartado describiremos los dispositivos (Sección 2.1) y software utilizados (Sección 2.2).

2.1 Dispositivos

Para la interacción entre el usuario y el sistema utilizaremos dos dispositivos: KinectV2 y Tobii EyeX.

Kinect nos proporciona información en forma de esqueleto del cuerpo que podemos utilizar para obtener las posiciones de las manos en el espacio tridimensional. Y el Tobii EyeX nos permite conocer la posición en la pantalla a la que está mirando el usuario. En este apartado describiremos el modo en que funcionan ambos dispositivos.

2.1.1 Kinect

Kinect es un dispositivo desarrollado por microsoft y presentado al público en el 2009. El dispositivo cuenta con: una cámara RGB que permite obtener imágenes a color, un emisor infrarrojo que en conjunto con una cámara infrarroja permite obtener datos de profundidad y un micrófono para la captura de sonidos y una acelerómetro que permite obtener la orientación de la Kinect (Figura 2.1).

En la primera versión de kinect, para obtener información de profundidad, el dispositivo proyecta un haz de luz infrarroja que atraviesa una rejilla y es dividido en múltiples haces definiendo un patrón fijo [Khoshelham and Elberink, 2012]. La cámara infrarroja captura una imagen con los puntos en los que ha rebotado el patrón de luces infrarrojas y es comparada con una imagen de referencia que ha sido realizada a una distancia conocida. El punto del haz de luz se verá desplazado en comparación a la imagen de referencia en función de la distancia a la que se encuentre el objeto en el que haya rebotado el haz de luz. Se calcula el desplazamiento para todos los puntos mediante un procedimiento de correlación de imágenes y se obtiene una imagen de disparidad. Posteriormente haciendo uso de la imagen obtenida se utilizan cálculos de triangulación para obtener la distancia de cada punto.

2. Metodología

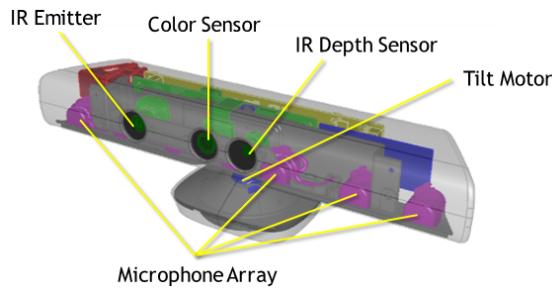


Figura 2.1: Sensores de la Kinect(Imagen extraída de ¹⁾)

En la segunda versión de kinect (KinectV2) el modo en que genera la nube de puntos cambia. Para obtener una nube de puntos el dispositivo utiliza un array de diodos láser que emiten pequeños pulsos de luz infrarroja. Cuando estos pulsos rebotan sobre los distintos objetos de la escena, la cámara infrarroja captura las señales y son usadas por el dispositivo para calcular el tiempo de vuelo de los pulsos capturados. Dichos tiempos de vuelo son utilizados para estimar la distancia de cada punto.

Características	Kinect v1	Kinect v2
Color Cámara	VGA 640 x 480 @30 fps	1920 x 1080 @30 fps
Profundidad Camara	QVGA 320 x 240	512 x 424
Profundidad máxima	4.5 M	4.5 M
Profundidad mínima	40 cm in near mode	50 cm
Número de articulaciones	20 joints	26 joints
USB Standard	2.0	3.0

Tabla 2.1: Características de las dos versiones de kinect.

Como se puede observar en la tabla 2.1, la kinect v2 tiene mayor resolución que kinect v1, permitiendo obtener imágenes a color con mayor detalle.

Kinect es una potente herramienta que permite obtener información de profundidad de los objetos que se encuentran en frente. El dispositivo procesa la información obtenida por los sensores para obtener una nube de puntos de la que luego extraerá información extra los controladores para ampliar las posibilidades que ofrece para desarrollo, como detectar gestos o articulaciones de un usuario.

En nuestro proyecto, la Kinect será utilizada principalmente para obtener enlaces (joints) de las articulaciones de la mano del paciente para detectar tres gestos simples que nos permitirán interactuar con el sistema. Se plantea el uso del algoritmo *Dynamic Time Warping* junto a los datos de los joints de las manos obtenidos para la captura de gestos. Estos es debido a que el sistema debe tener la capacidad de poder personalizar los gestos.

¹[https://decibel.ni.com/content/blogs/carlosotiniano/2014/04/09/aplicaciones-con-microsoft-kinect\(04/2016\)](https://decibel.ni.com/content/blogs/carlosotiniano/2014/04/09/aplicaciones-con-microsoft-kinect(04/2016))

2.1.2 Tobii EyeX

Tobii EyeX es un dispositivo desarrollado por la empresa Tobii Tech. El dispositivo tiene la función de rastrear los ojos y estimar el punto que está siendo observado en la pantalla por el usuario. También nos ofrece la posibilidad de desarrollar aplicaciones que utilicen el dispositivo mediante un completo SDK.

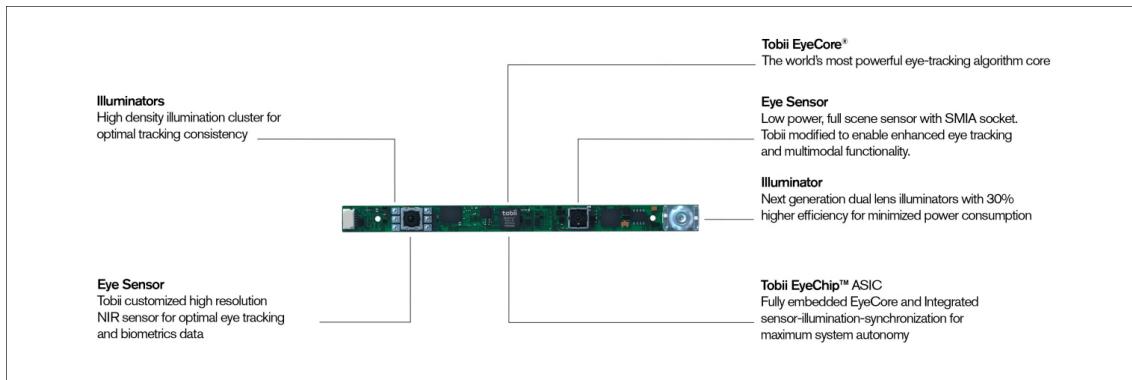


Figura 2.2: Esquema del dispositivo EyeX (Imagen extraída de ²)

El Tobii EyeX es capaz de rastrear la mirada del usuario gracias a los sensores (Figura 2.2) que posee. Utiliza varios emisores infrarrojos y un sensor NIR (Near Infrared) para obtener el punto que está siendo observado por un usuario. Para realizar el procesamiento se apoya de un procesador (Tobii EyeChip) desarrollado por la compañía que integra los algoritmos necesarios para su funcionamiento.

Características	Valores
Distancia máxima	100 cm
Distancia mínima	45 cm
Frecuencia	>60 Hz
Latencia con la aplicación	15 ms +/- 5 ms
Tamaño máximo de pantalla	27"
USB Standard	3.0

Tabla 2.2: Características del Tobii EyeX.

La tabla 2.2 muestra los rangos en los que trabaja el Tobii EyeX. Puede reconocer los ojos a una distancia máxima de 100 cm y una distancia mínima de 45 cm. Trabaja con una frecuencia mayor a 60 Hz y presenta una latencia de 15 ms al comunicarse con las aplicaciones. El dispositivo puede ser utilizado con pantallas de hasta 27" y necesita de un usb 3.0.

El dispositivo está provisto de un software (EyeX Engine) que permite interactuar con el ordenador utilizando los ojos y calibrar el dispositivo. El modo de uso empleado por defecto para interactuar con cualquier aplicación es mediante una tecla

²[http://www.tobii.com/tech/technology/system-overview/\(04/2016\)](http://www.tobii.com/tech/technology/system-overview/(04/2016))

de acción y la mirada, es decir, que el usuario podrá realizar un click en un elemento de la pantalla mirándolo y pulsando una tecla de acción.

Este dispositivo será utilizado para ofrecer a pacientes con problemas de movilidad la posibilidad de interactuar con el sistema, pero debido a que el modo que emplea necesita de una tecla de acción impide que algunos usuarios puedan utilizarlo. Por este motivo utilizaremos el SDK que nos ofrece la empresa para integrar directamente en el sistema la posibilidad de realizar acciones mediante guiños o parpadeos largos. Para ello se preparará el sistema de modo que una persona configure el gesto que mejor podrá utilizar un paciente determinado y el tiempo que debe estar realizando ese gesto para ser considerado una acción.

Por otro lado, debido a que cada usuario tendrá distintas condiciones en los ojos, el sistema tendrá que ser capaz de guardar una referencia a la calibración del dispositivo, para ello nos apoyaremos en el SDK que nos permite controlar los perfiles de calibración cargados en cada momento.

2.2 Herramientas Software

El trabajo lo realizaremos utilizando C/C++ en Visual Studio 2013. Para la generación del proyecto nos ayudamos de CMake, que procurará que las dependencias necesarias para el desarrollo estén disponibles en la máquina. Para el control de versiones usaremos Git.

La interfaz de la aplicación será realizada con Qt (Sección 2.2.1) y para el desarrollo del juego nos ayudaremos de opencv para dividir una imagen.

Por último, utilizaremos las herramientas para desarrolladores proporcionadas por las empresas distribuidoras de los dispositivos Kinect (Sección 2.2.3) y EyeX (2.2.2) para la interacción con el sistema.

2.2.1 Qt

Qt es una biblioteca multiplataforma desarrollado como software libre y código abierto. Es utilizada para la implementación de interfaces gráficas, aunque también puede ser usada para programas por consola. En el desarrollo de nuestro proyecto serán utilizados los recursos que nos ofrece para el desarrollo de interfaces gráficas.

Para trabajar con Qt podemos utilizar "Qt Creator", que es el software desarrollado para trabajar con esta herramienta. En nuestro caso nos ayudamos de CMake³ para generar un proyecto en "Visual Studio 2013" que realice las operaciones de generación de códigos moc(Meta-Object Compiler) que es requerido por Qt para el

³Instrucción usada para generar archivos MOC: qt4_wrap_cpp()

uso de signals y slots⁴, información de tipo de tiempo de ejecución y las propiedades dinámicas del sistema.

Los recursos principales que utilizaremos de esta herramienta serán "QStackedWidget" para navegar en el sistema, "QPushButton" y "QComboBox" para la selección y confirmación de algunas opciones.

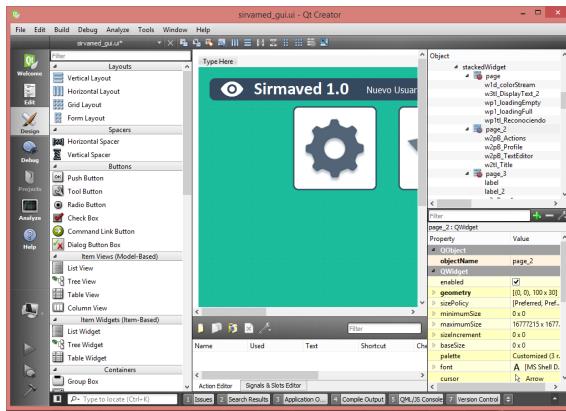


Figura 2.3: Heraminta Qt Creator para el diseño de las interfaces

Par el diseño de las interfaces utilizaremos "Qt Creator" o "Qt Designer" (Figure 2.3). Esta herramienta genera un archivo en formato ".ui" y mediante las herramientas "qmake" y "uic" genera código C++ que referencia a los elementos de nuestra interfaz. Estas acciones también las automatizaremos con CMake⁵.

2.2.2 EyeX SDK

EyeX SDK es un conjunto de herramientas preparadas para construir aplicaciones que utilicen dispositivos Tobi EyeX para interactuar con una aplicación.

El SDK ofrece las herramientas necesarias para obtener:

- El punto en la pantalla que está siendo observado por los ojos (Gaze point).
- La localización 3D en que se encuentra cada ojo en relación a la pantalla.
- Los puntos en la pantalla en el que los ojos se han parado y centrado la atención (Fixations). Estos puntos son una concentración de los "Gaze Point", lo que quiere decir que el usuario lleva cierto tiempo prestando atención a esa zona de la pantalla.
- Detección del ojo cuando entra y sale de una región.
- Regiones que pueden generar eventos utilizando los ojos y una acción (Activatable).
- Regiones que pueden realizar scroll sobre elementos de la ventana.

⁴Signals y Slots son utilizados por Qt para la comunicación entre objetos

⁵Instrucción usada para generar la cabecera de un archivo .ui qt4_wrap_ui()

2. Metodología

- Detección de presencia de usuarios.

Para definir elementos en una aplicación con las que pueda interactuar un usuario utilizando los ojos se define un tipo de elemento denominado "interactor". Estos elementos pueden tener distintos comportamientos como obtener el punto que está siendo observado por el usuario u ofrecer la posibilidad de generar un evento de tipo "activatable", como realizar click o posar el foco de la aplicación sobre ese elemento.

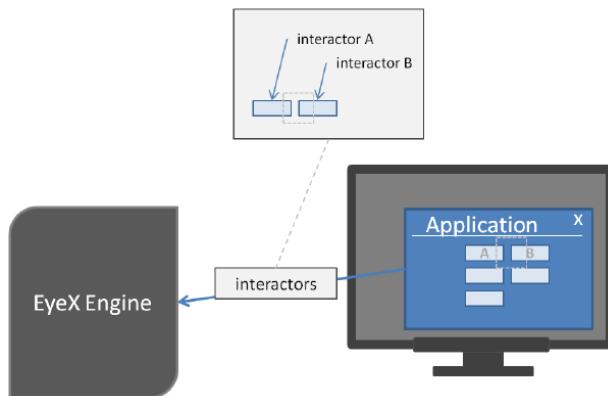


Figura 2.4: La caja definida por líneas discontinuas es la región devuelta por la query, y sobre los botones A y B que se encuentran en dicha región se crean dos "interactors" (Imagen extraída de la guía del SDK).

Debido a que una ventana puede tener distintos tamaños y posiciones en la pantalla no es posible utilizar "interactors activatables" fijos en la aplicación para interactuar con elementos, es necesario actualizarlos periódicamente para que la región que definen corresponda con el elemento con el que pretende interactuar. Para ello el SDK dispone de "querys", funciones que de manera periódica informan de la región que está siendo observada por el usuario para crear "interactors" temporales sobre la aplicación (Figura 2.4).

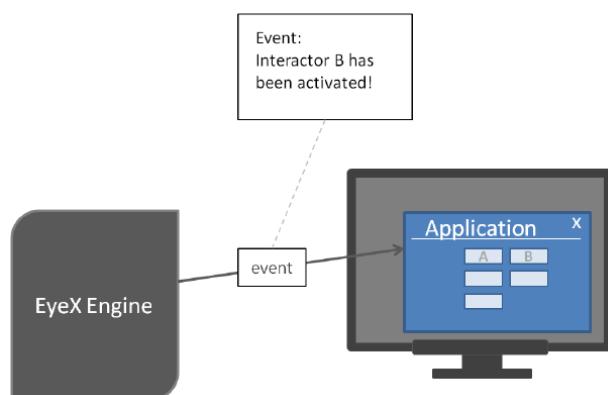


Figura 2.5: En función del comportamiento de los interactors utilizados se lanzarán eventos con la información pertinente, como pueden ser la posición de la mirada en la pantalla (Imagen extraída de la guía del SDK).

Los "interactors" pueden tener distintos comportamientos, como ofrecer información de la posición de los ojos en relación a la pantalla, informar sobre el punto que está siendo observado en la pantalla por el usuario, o activar una acción sobre una región "activatable". Para obtener la información que nos proveen esos interactors el engine lanza eventos con la información que tiene relación con el comportamiento del interactor (Figura 2.5).

El modo que tienen el engine de realizar acciones es utilizar una tecla de acción (Por defecto CTRL derecho). Cuando el un usuario se encuentra mirando la aplicación y pulsa el botón de acción, el engine envía información a la aplicación con el "interactor" con el que estuviese interactuando en ese punto el usuario.

Debido a que los usuarios que utilicen el sistema pueden estar en situación de no poder utilizar las manos para pulsar una tecla de acción, resulta necesario configurar el servicio que haga uso del Tobii EyeX de manera que podamos reconocer guiños con el ojo izquierdo, guiños con el ojo derecho o pestañeos largos. Para lograrlo crearemos "interactors" con un comportamiento de tipo "TX_BEHAVIORTYPE_EYEPOSITIONDATA", este comportamiento genera eventos con información de la posición de los ojos en relación a la pantalla. Si el usuario se encuentra en frente del dispositivo de manera que pueda detectar los ojos y este se desplaza provocando que pierda la información de rastreo, el último valor de posición que devolverá el evento del interactor con comportamiento "TX_BEHAVIORTYPE_EYEPOSITIONDATA" será la última posición de los ojos en relación a la pantalla. Sin embargo, si el dispositivo tiene el rastreo de la posición del ojo y uno de los dos se cierra o se tapa mientras el otro se mantiene abierto la posición que devolverá el evento en el ojo del que se ha perdido el rastreo será "0". Utilizando estos valores configuraremos el servicio de modo que emita eventos en función del gesto realizado con los ojos que se pretenda utilizar como gesto de acción.

2.2.3 Kinect SDK 2.0

Kinect SDK 2.0 es un conjunto de herramientas que del mismo modo que con el EyeX SDK nos permite construir aplicaciones que utilicen un dispositivo. En este caso el dispositivo es la KinectV2, y utilizaremos los recursos que nos proporciona para obtener información de las articulaciones o joints de las manos para realizar el reconocimiento de gestos utilizando el Dynamic Time Warping.

La KinectV2 puede ofrecer distintos flujos de datos como: nubes de puntos, imágenes RGB, o joints que representarán las articulaciones de las personas que se encuentren en frente del dispositivo. Para nuestro proyecto son estos joints los que tienen mayor importancia, utilizándolos para la detección de gestos realizados con las manos, de modo que la información que utilizaremos de la kinect son los joints que nos aporta y configuraremos el servicio de modo que nos devuelva esa información.

Los joints que nos interesan son los que corresponde con las manos, el cuello, la cabeza y uno de los hombros. Los joints del cuello, cabeza y hombro los utilizare-

mos para realizar una transformación de coordenadas de las posiciones de los joints de las manos, de modo que el origen de coordenadas pase de ser la KinectV2 a ser el cuello. Explicaremos este proceso más adelante.

2.3 Técnicas aplicadas

En esta sección explicaremos procesos y algoritmos utilizados para alcanzar los objetivos de nuestro proyecto, como el cambio de coordenadas (Sección 2.3.1) necesaria para normalizar las capturas y hacerlas invariantes a distancia y ángulo con respecto a la cámara y Dynamic Time Warping (Sección 2.3.2) utilizado para comparar las trayectorias definidas por los gestos y encontrar la de mayor similitud.

2.3.1 Cambio de origen de coordenadas

La transformación de coordenadas es un proceso por el cual cambiamos el origen de coordenadas de un punto a otro.

Para realizar la transformación de coordenadas deberemos escoger un nuevo punto de origen y crear un sistema de referencia entorno a él.

El proceso de cambio de coordenadas conlleva a realizar dos pasos, la rotación y traslación del punto de modo que la nueva coordenada tenga la misma distancia que tenía anteriormente con el nuevo origen de coordenadas.

Para realizar la rotación del punto necesitaremos calcular los ángulos definidos por cada eje con su correspondiente en el nuevo sistema de coordenadas, es decir, se deberá calcular el ángulo entre el eje X del antiguo sistema de referencia con el eje X del nuevo y de la misma manera lo realizaremos con el resto de ejes. Dichos ángulos los utilizaremos en las matrices de rotación para rotar el punto.

$$R_x(\alpha) = \begin{pmatrix} 0 & 0 & 0 \\ 0 & \cos\alpha & -\sin\alpha \\ 0 & \sin\alpha & \cos\alpha \end{pmatrix} R_y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix} R_z(\gamma) = \begin{pmatrix} \cos\gamma & -\sin\gamma & 0 \\ \sin\gamma & \cos\gamma & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Estas tres matrices podemos representarlas en una matriz general [Murray, 2013], donde la matriz será construida por el producto de las tres. La matriz general se verá condicionada por el orden de rotación, es decir, la matriz obtenida de multiplicar $R_\alpha R_\beta R_\gamma$ será distinta de la obtenida al multiplicar $R_\gamma R_\beta R_\alpha$. En nuestro caso

trabajaremos con la matriz general obtenida del producto $R_zR_yR_x$.

$$R_zR_yR_x = \begin{pmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \cos \alpha \cos \gamma \sin \beta + \sin \alpha \sin \gamma \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\cos \gamma \sin \alpha + \cos \alpha \sin \beta \sin \gamma \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta \end{pmatrix}$$

La matriz de rotación general cambiará las coordenadas del punto, aplicando giros sobre todos los ejes para adecuar la orientación al nuevo sistema de referencia, pero la posición seguirá siendo relativa al dispositivo por lo que será necesario aplicar una traslación.

El proceso de traslación solo implica incrementar el tamaño de la matriz y añadir en esta el punto que será el nuevo origen de coordenadas [Murray, 2013]:

$$T =$$

$$\begin{pmatrix} \cos \beta \cos \gamma & \cos \gamma \sin \alpha \sin \beta - \cos \alpha \sin \gamma & \cos \alpha \cos \gamma \sin \beta + \sin \alpha \sin \gamma & O_x \\ \cos \beta \sin \gamma & \cos \alpha \cos \gamma + \sin \alpha \sin \beta \sin \gamma & -\cos \gamma \sin \alpha + \cos \alpha \sin \beta \sin \gamma & O_y \\ -\sin \beta & \cos \beta \sin \alpha & \cos \alpha \cos \beta & O_z \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Con la matriz de transformación definida solo tendríamos que aplicar el producto para los puntos que queremos cambiar su origen.

$$P' = T \cdot P \text{ para } P = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

2.3.2 Dynamic Time Warping

Dynamic time warping es una técnica utilizada para relacionar dos secuencias temporalmente dependientes. Dichas secuencias pueden ser señales discretas o señales continuas que han sido muestradas en puntos equidistantes en relación al tiempo [Müller, 2007].

Para realizar el cálculo se utiliza una función de coste local que permita calcular la distancia entre dos puntos de ambas secuencias, como la distancia euclídea o manhattan. Dicha función de coste local es utilizada por el algoritmo *Dynamic Time Warping* para acumular de manera óptima las distancias y obtener un valor de similitud que relacione ambas secuencias. El coste acumulado en la distancia de ambas secuencias se puede definir mediante la siguiente función:

$$D(n, m) = \min D(n - 1, m - 1), D(n - 1, m), D(n, m - 1) + c(x_N, y_m)$$

donde 'c' es la función de coste local y 'n' y 'm' definen la posición en una matriz construida a partir de las secuencias X e Y.

2. Metodología

La matriz se construye utilizando los puntos de una secuencia como eje X y la otra secuencia como eje Y. La primera fila y columna serán iniciadas con valor infinito, y el elemento (0,0) de la matriz tendrá como valor inicial 0. Con los valores de la primera fila y columna iniciados aplicamos la función D(N,M), dicha función es aplicada para cada celda de las siguientes filas de izquierda a derecha (Tabla 2.3).

A →	(0,0)	(0,0)	(1,1)	(2,0)	(3,1)	(4,0)	(5,1)	(6,0)
B ↓	0,0	inf						
(-1,1.5)	inf	1,8	3,86	7,21	11,25	16,47	22,49	29,65
(0,2)	inf	3,80	3,22	6,05	9,21	13,68	18,78	25,1
(1,1.5)	inf	5,61	3,72	5,02	7,08	10,44	14,47	19,69
(2,2)	inf	8,43	5,13	5,72	6,43	9,26	12,42	16,9
(3,1.5)	inf	11,79	7,19	6,93	6,22	8,02	10,08	13,44
(4,2)	inf	16,26	10,36	9,76	7,63	8,22	9,43	12,26
(5,1.5)	inf	21,48	14,39	13,12	9,69	9,43	8,72	10,52
(6,2)	inf	27,8	19,49	17,59	12,86	12,26	10,13	10,72
(7,1.5)	inf	34,96	25,51	22,81	16,89	15,62	12,19	11,93

Tabla 2.3: Ejemplo de cálculo en el Dynamic Time Warping para dos secuencias. La primera columna muestra los puntos para la secuencia B y la primera fila los puntos para la secuencia A. Los valores del resto de celdas es el resultado de aplicar el algoritmo *Dynamic Time Warping*.

El resultado de aplicar el Dynamic Time Warping sobre las secuencias A y B da como resultado en la última posición de la matriz obtenida una distancia óptima que permite estimar la similitud entre ambas secuencias (Valor 11.93 de la tabla 2.3). Para alcanzar ese valor de similitud el algoritmo acumula las distancias óptimas entre los puntos de ambas secuencias (Figura 2.6).

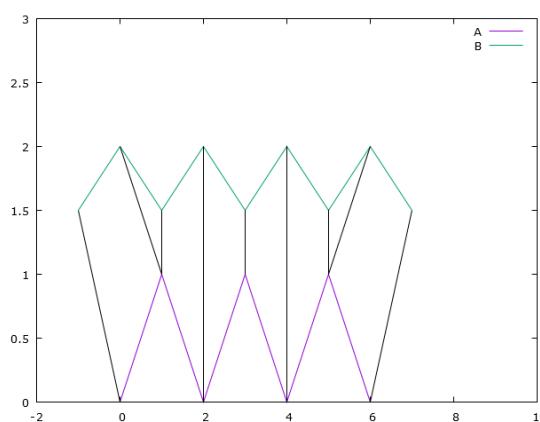


Figura 2.6: En la imagen se observa como rectas de color negro las distancias óptimas para relacionar dos secuencias A y B utilizando como función de coste local la distancia euclídea.

El algoritmo nos resulta de utilidad para reconocer el gesto realizado por un usuario en el sistema. Almacenando la secuencia de posiciones de la mano que captura la kinect y comparándolas con otras muestras previamente almacenadas, podemos obtener el gesto que más se asemeja al que está siendo realizado por el usuario.

2. Metodología

3

SIRMAVED

SIRMAVED (Desarrollo de un Sistema Integral Robótico de Monitorización e Interacción para Personas con Daño Cerebral Adquirido y Dependientes) es un proyecto nacional que está siendo desarrollado por los profesores del Instituto Universitario de Investigación e Informática: José García Rodríguez y Miguel Ángel Cazorla Quevedo que tiene como objetivo dar asistencia a personas con discapacidades motoras o cognitivas.

SIRMAVED tiene como objetivo hacer uso de varios sensores para dotar a un robot con la capacidad de interactuar con usuarios que tenga discapacidad motora o cognitiva. El sistema tendrá la capacidad de reconocer gestos del usuario y responder a sus necesidades de modo que pueda ser utilizado en un entorno doméstico con un seguimiento mínimo por parte de un doctor o especialista .

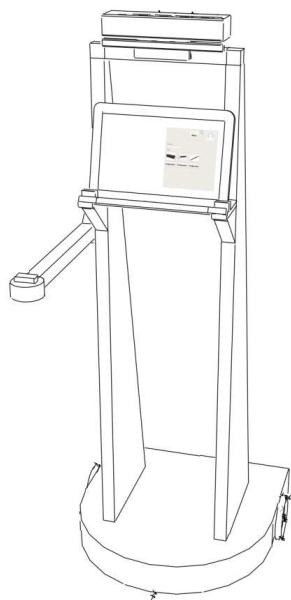


Figura 3.1: Plataforma móvil sobre la que estarán situados el conjunto de sensores de SIRMAVED.

Debido a que los pacientes pueden presentar distintos grados de discapacidad el sistema tendrá la capacidad de adecuar sus sensores para reconocer las intenciones del paciente (Figura 3.1). Para ello el sistema estará equipado con un sensor de

3. SIRMAVED

largo alcance en la parte superior de la plataforma permitiendo una identificación biométrica basada en el reconocimiento de rostros y el reconocimiento de gestos o poses. También utilizaremos un sensor de medio alcance para el reconocimiento de gestos que necesiten mayor precisión. Este sensor se encontrará situado en un brazo giratorio lateral con un grado de libertad permitiendo que cuando el sensor se encuentre desactivado no estorbe al usuario que esté utilizando el sistema.

El sensor situado en la parte superior de la plataforma también está equipado con una matriz de micrófonos permitiendo capturar órdenes por voz.

En la parte central de la plataforma estará situada una tableta digital que servirá como medio de procesamiento del sistema ofreciendo también un medio de interacción táctil. Debajo de la tableta se encontrará situado un dispositivo de seguimiento ocular que permita a usuarios con dificultades motoras más grave interactuar con el sistema.

El software del que estará dotado la tableta digital permitirá controlar el conjunto de sensores y guardará un historial de las actividades del paciente para que un especialista pueda observar los progresos vía online. En el sistema se implementaran varias fichas o aplicaciones que sirvan como ejercicios de rehabilitación cognitiva y dichas aplicaciones podrán ser utilizadas por los pacientes con los sensores comentados anteriormente. Los actividad de los sensores será dividida en varios hilos y solo podrá haber un sensor activo para cada usuario. Para ello el sistema activará y desactivará los sensores en función de la configuración del usuario.

La plataforma sobre la que se encuentran los sensores será móvil, con el fin de poder desplazarlo de un modo sencillo en un entorno doméstico.

4

Reconocimiento de movimientos oculares

Debido a que hay personas que pueden presentar un grado de discapacidad, que les impida utilizar el sistema con las alternativas de interacción de las que se dispone y que el modo básico de interacción que utiliza Tobii EyeX es pulsando una tecla de acción, añadiremos la posibilidad de interactuar con el sistema mediante guiños o pestañeos largos.

4.1 Desarrollo

El usuario deberá configurar el gesto a realizar en la pantalla de configuración (Sección 4.1.1), de modo que el servicio (Sección 4.1.2) realice una acción al reconocer el gesto seleccionado.

4.1.1 Interfaz de personalización

Para el control del sistema utilizando el dispositivo de seguimiento ocular hemos definido una pantalla de configuración que vincule un guiño o un pestañeo al usuario que haya iniciado sesión y el tiempo de duración de los gestos para ser aceptado como una acción (Figura 4.1).



Figura 4.1: Interfaz con opciones para personalizar los gestos a realizar por el usuario con los ojos.

El botón calibrar y el desplegable que se observa encima de él en la figura 4.1 se corresponden con la opción de calibración del dispositivo. El botón realiza una llamada al sistema para ejecutar el asistente de calibración, donde una vez finalizado el proceso se genera un evento que permite actualizar el desplegable y escoger el nuevo perfil de calibración para el usuario.



Figura 4.2: Botón con el foco al estar siendo observado utilizando el Tobii EyeX.

Con el dispositivo calibrado al observar un botón este obtendrá el foco (Figura 4.2) y cambiará de color para diferenciarse de los otros hasta que el usuario deje de mirarlo. Si uno de los botones tiene el foco y el usuario quiere pulsarlo, deberá mantener el gesto que ha seleccionado en el desplegable del centro, el tiempo que ha fijado en el desplegable de la izquierda.

4.1.2 Proceso de reconocimiento ocular

Como comentamos en el apartado 2.2.2, para reconocer los gestos realizados con los ojos definiremos "interactors" con comportamiento "TX_BEHAVIORTYPE_EYE-POSITIONDATA". Los "interactors" definidos con ese comportamiento lanzarán eventos con las posiciones 3D de los ojos relativas al dispositivo, y en función de que los valores sean 0 ó distintos diferenciaremos los posibles gestos.

Los posibles gestos serán *Guño derecho* si la posición del ojo izquierdo es distinta de 0 y la del ojo derecho es igual a 0. *Guño izquierdo* si sucede al revés y *Pestaño* si ambos tienen como posición 0.

Si queremos realizar la captura de un pestaño, en ocasiones los valores de ambos ojos a 0 no se realiza a la vez. Es decir, puede ocurrir que cuando realizamos el gesto *Pestaño largo* el dispositivo capture una primera muestra con la posición del ojo izquierdo a 0 y el derecho distinto a 0 ó al revés, ojo derecho a 0 y el izquierdo distinto a 0. Para solucionar este problema y evitar que una persona tenga que mantener durante el tiempo fijado alguno de los ojos abiertos con el cansancio visual que ello implica, hemos utilizado un buffer para acumular todos los gestos que son capturados en cada muestra durante el tiempo seleccionado. El gesto que será reconocido cuando haya finalizado el tiempo será el que más veces aparezca en el

buffer.

El tiempo de aceptación para reconocer los gestos comenzará a contar cuando se detecte la posición de uno de los ojos a 0. Si ambos ojos se abren antes de reconocer un gesto el servicio reiniciará el contador a la espera que la posición de alguno de los ojos vuelva a estar a 0.

4.2 Experimentación

Las pruebas con el dispositivo las hemos realizado de dos modos distintos a varias distancias, por un lado realizando los gestos cerrando uno o ambos ojos (Tabla 4.1) y por otro tapando el ojo con una o ambas manos (Tabla 4.2).

	Guiño Izquierdo	Guiño Derecho	Pestañeo Largo
0.5m	SI	SI	SI
0.7m	SI	SI	SI
1m	NO	NO	NO

Tabla 4.1: Tabla con las distancias a las que es capaz de reconocer el gesto el Tobii EyeX realizando el gesto.

	Guiño Izquierdo	Guiño Derecho	Pestañeo Largo
0.5m	SI	SI	SI
0.7m	SI	SI	SI
1m	NO	NO	NO

Tabla 4.2: Tabla con las distancias a las que es capaz de reconocer el gesto el Tobii EyeX tapando los ojos con las manos.

En ambos casos el sistema fue capaz de reconocer los gestos hasta una distancia de 0.7 metros, superados los 0.7 metros el dispositivo tiene problemas para identificar la posición de los ojos devolviendo valores a 0 cuando el ojo se encuentra abierto.

En el siguiente enlace se puede ver el proceso de configuración de Tobii EyeX y su funcionamiento: <https://goo.gl/photos/wuCMbs9WNzmW8bCA8>.

4. Reconocimiento de movimientos oculares

5

Reconocimiento de gestos con Kinect

La captura de gestos con kinect tienen como objetivo que personas con movilidad reducida, que puedan realizar movimientos con el codo o/y hombro puedan controlar el sistema mediante tres gestos simples, que corresponderán con las acciones Anterior, Siguiente y Aceptar. Dichas acciones permitirán cambiar el foco entre los distintos elementos de la interfaz gráfica o activar una acción. Debido a que los usuarios pueden presentar problemas para realizar algunos movimientos, se dará la posibilidad de personalizar los gestos mediante la grabación de las posiciones de la mano para cada una de las posibles acciones.

5.1 Desarrollo

El sistema de reconocimiento de gestos consistirá en un servicio que se ejecutará para aquellos usuarios que tengan configurado en su perfil el uso de este dispositivo como medio de interacción con el sistema.

El primer paso para realizar el reconocimiento de gestos es utilizar los recursos ofrecidos por el SDK de KinectV2 para obtener joints del esqueleto del usuario que utilice el sistema. Con los joints podremos guardar las posiciones rastreadas en un dataset (Sección 5.1.3) si estamos en modo captura de datos o utilizarlos para comprobar si se corresponde con alguno de los gestos almacenados (Sección 5.1.4) pero en ambos casos necesitaremos realizar una transformación de coordenadas (Sección 5.1.2) para situar las posiciones de las manos en un punto de origen invariante a la orientación y posición en relación a Kinect.

Los joints de las manos son almacenados en un buffer circular que funcionará a modo de ventana deslizante (Figura 5.1). Cada captura realizada de los joints entrará dentro del buffer y si estamos en modo reconocimiento de gestos se aplicará Dynamic Time Warping, para el conjunto de los elementos que estén dentro de la ventana, sin embargo si estamos en modo entrenamiento se guardará el contenido del buffer en un dataset.

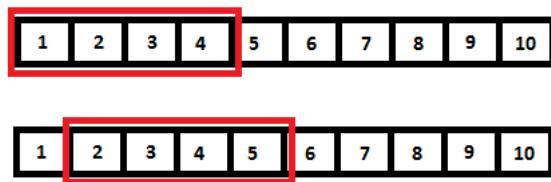


Figura 5.1: En este ejemplo la venta tiene un tamaño para cuatro elementos. Desplazamos por unidades la ventana e intentamos reconocer el gesto en cada desplazamiento.

5.1.1 Interfaz de personalización

Para la configuración de los gestos de la kinect definimos una interfaz (Figura 5.2) que permita configurar las distintas posibilidades que tenemos para la captura de gestos con la kinect. Los gestos podrán ser capturados siguiendo los movimientos realizados con la mano izquierda o derecha, de modo que se deberá seleccionar la mano con la que realizará los movimientos el paciente. Por otro lado, debido a que un paciente podría llegar a realizar movimientos muy lentamente se da la posibilidad de cambiar el tiempo de grabación del gesto.



Figura 5.2: Pantalla de configuración para la grabación de gestos realizados con la Kinect.

5.1.2 Transformación de coordenadas

La Kinect devuelve la posición de los joints del esqueleto relativas a su posición, provocando que realizar el mismo gesto en distintas posiciones y orientaciones tengan distintas coordenadas. Para afrontar este aspecto realizaremos un cambio de coordenadas de las posiciones de las manos de manera que indiferentemente de la posición u orientación en relación a la cámara tengamos coordenadas similares para el mismo gesto. Para conseguir lo que hemos mencionado anteriormente usaremos el joint del cuello como nuevo origen de coordenadas de modo que nuestro cuerpo

sea el nuevo sistema de referencia.

Para obtener el nuevo sistema de referencia utilizaremos joints del esqueleto para definir nuevos ejes X, Y y Z. Estos joints serán los que corresponden con la cabeza, cuello y hombro derecho respectivamente, que nos permitirán definir un plano sobre el cual extraer una normal y definir un sistema de referencia en base a ella. Debido a que los usuarios del sistema van a estar principalmente sentados, creemos conveniente utilizar el cuello como origen de coordenadas a otras alternativas.

Lo primero que haremos será utilizar los tres joints seleccionados para definir dos vectores. Con estos vectores definidos podremos calcular un vector perpendicular a la unión de ambos para obtener el eje Z.

$$\overrightarrow{NS} = S - N \quad \overrightarrow{NH} = H - N$$

S Es el punto correspondiente al hombro, N al cuello y H a la cabeza.

Para calcular un vector perpendicular a los vectores \overrightarrow{NS} y \overrightarrow{NH} calcularemos su producto vectorial.

$$Z = \overrightarrow{NS} \times \overrightarrow{NH}$$

Por último definimos los ejes X e Y a partir del eje Z calculado.

$$X = \begin{pmatrix} -z \\ y \\ x \end{pmatrix} \quad Y = \begin{pmatrix} x \\ -z \\ y \end{pmatrix}$$

Con el nuevo sistema de referencia definido, el siguiente paso es calcular la matriz de transformación de coordenadas explicada en la sección 2.3.1. Para ello deberemos obtener primeramente los ángulos entre los ejes.

$$\theta(V_1, V_2) = \arccos \left(\frac{V_1 \cdot V_2}{|V_1| \cdot |V_2|} \right)$$

Con $\theta(V_1, V_2)$ calcularemos los ángulos que definen cada par de ejes del nuevo sistema de referencia con el antiguo, es decir:

$$\alpha = \theta(X_k, X) \quad \beta = \theta(Y_k, Y) \quad \gamma = \theta(Z_k, Z)$$

Donde X_k , Y_k y Z_k son los ejes correspondientes al sistema de referencia de kinect.

Por último nos queda sustituir en la matriz de transformación construida en la sección 2.3.1 los ángulos obtenidos y las coordenadas del joint correspondientes al cuello, como punto de origen y realizar el producto de la matriz de transformación

por los puntos que queremos transformar.

5.1.3 Entrenamiento de gestos

La grabación de los gestos será una opción que podrá realizar el usuario para definir movimientos personalizados con los que interactuar con el sistema.

Para realizar la grabación del gesto se deberá pulsar el botón "Grabar gesto" (Figura 5.2). Entonces aparecerá una cuenta atrás de tres segundos donde al llegar a 0 aparecerá el mensaje "Ya" y el usuario tendrá que realizar el gesto en el tiempo definido por el desplegable correspondiente. Esta acción se repetirá una vez más para tener una segunda muestra del gesto que se pretende reconocer.

Cada vez que queremos grabar un nuevo gesto el buffer es vaciado y se capturan posiciones de las manos hasta que finalice el tiempo. A esas posiciones se les realiza una transformación de coordenadas y se normaliza su punto inicial al punto de origen antes de ser almacenadas en un dataset temporal.

El proceso de normalización consiste en restar las coordenadas del primer punto al resto de puntos de la secuencia y dejar el primero con sus coordenadas a 0.

El dataset temporal no será vinculado con el usuario hasta que este pulse "Guardar".

5.1.4 Reconocimiento de gestos

Los joints o articulaciones de interés para nuestro proyecto serán los del cuello, hombro derecho, cabeza y manos. Los joints del cuello, hombro y cabeza serán utilizados para realizar una transformación de coordenadas de la posición de las manos. Esta transformación de coordenadas la realizaremos cada vez que obtengamos nueva información del esqueleto antes de introducir la posición de la mano en el buffer.

Cada vez que entra un nuevo elemento en el buffer realizaremos una copia de todo el buffer en otro temporal. El contenido copiado será normalizado al origen de coordenadas.

Al buffer normalizado le aplicaremos Dynamic Time Warping (Sección 2.3.2) con todos los gestos almacenados en el dataset y nos quedaremos con el gesto que menor valor devuelva.

Aplicando Dynamic Time Warping con todos los elementos del dataset siempre obtendremos un gesto que será más cercano al movimiento ejecutado con las manos a pesar de que este no se corresponda con dicho gesto. Para evitar este comportamiento estableceremos umbrales como distancias mínimas que debe cumplir el valor devuelto por el algoritmo para ser aceptado como gesto.

Debido a que damos la posibilidad de cambiar el tiempo de grabación de gesto debemos definir varios umbrales de aceptación. Cuanto mayor sea el tiempo del gesto que se quiere reconocer más difícil será que el usuario realice exactamente el mismo movimiento implicando un aumento en el umbral a utilizar, por otro lado este aumento en el tiempo también implica cambiar el tamaño de la ventana del buffer para recoger información suficiente en ese tiempo de las posiciones de las manos.

5.2 Experimentación

Con la implementación realizada procedemos a la realización de pruebas para comprobar las limitaciones en la captura de gestos con kinect. Para ello realizamos la grabación de un dataset para el conjunto de las tres acciones posibles "Siguiente", "Anterior" y "Aceptar". La grabación fue realizada situándonos en frente del dispositivo a una distancia de un metro con el codo de la mano derecha pegado al costado (Figura fig:kinect-frente) desplazamos la mano hacia la izquierda para definir el gesto anterior, hacia la derecha para definir el gesto siguiente y hacia adelante para el gesto aceptar.

Una vez finalizado el entrenamiento procedimos a comprobar la efectividad en el reconocimiento con cada uno de los gestos. Para ello nos posicionamos con distintas orientaciones en relación a la Kinect para comprobar como afectaba en el acierto del gesto. La tabla 5.1 muestra los resultados de dichas pruebas. Realizando el gesto en la misma posición en la que fue grabado, es decir mirando la kinect (0°) el programa es capaz de reconocer los 3 gestos y realizar la acción que le corresponde, sin embargo a medida que giramos el cuerpo los aciertos comienzan a bajar.

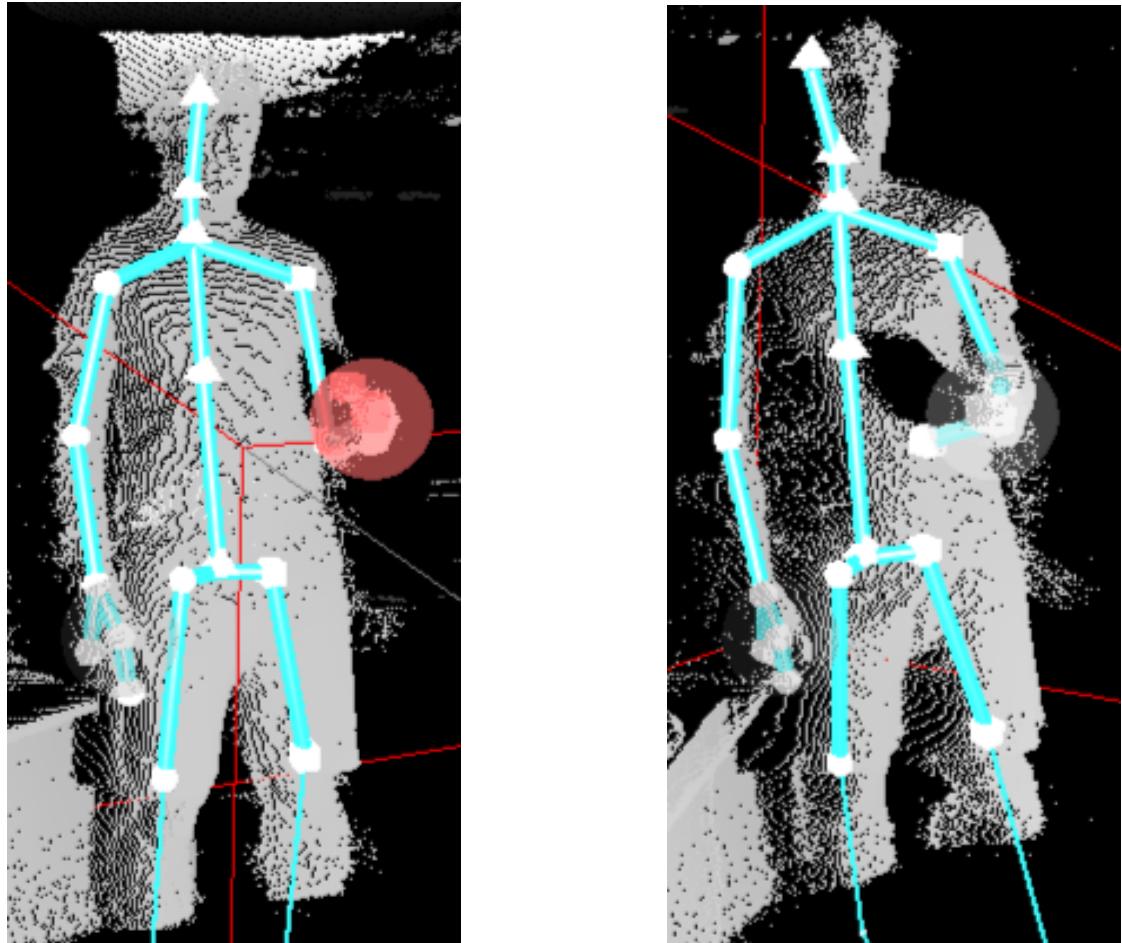
Observando los resultados podemos apreciar que habiendo realizado la grabación de los gestos con la mano derecha, a medida que giramos a la izquierda la tasa de aciertos de los gestos siguiente y anterior se reducen, alcanzando el 80% de aciertos, posicionándonos con una orientación de -30° en relación a la kinect y un 20% con una orientación de -45° . Este descenso se debe a la correlación que realiza la kinect con las manos cuando no tiene una perspectiva completa del cuerpo del usuario. En las muestras examinadas observamos que girando el cuerpo hacia la izquierda manteniendo el codo pegado al costado con el puño hacia el frente, la kinect realiza una estimación de la posición de la mano acercando virtualmente la posición al cuerpo del usuario (Figura 5.3). Esta correlación cambia virtualmente las posiciones en los ejes x,y en relación al cuello como origen de coordenadas, pero al desplazar la mano hacia delante la profundidad no se ve alterada, pues la kinect si es capaz de estimar ese movimiento con esa orientación, permitiendo el reconocimiento de gestos como el definido para aceptar, donde dicho movimiento es un desplazamiento de la mano hacia el frente.

Cuando giramos el cuerpo hacia la derecha la tasa de aciertos para "siguiente" y "anterior" aumentó en comparación cuando teníamos una orientación de -45° o 30° . Este aumento es producido gracias a que en esa posición la kinect ha podido es-

timar mejor la posición de la mano para movimientos realizados en los ejes x e y en relación al cuello como origen de coordenadas. Sin embargo los movimientos en profundidad sufren mayor dificultad en el reconocimiento, en esta posición la estimación de esta coordenada se ve más afectada.

	-45°	-30°	0°	30°	45°
Siguiente	20%	80%	100%	100%	70%
Anterior	0%	80%	100%	20%	10%
Aceptar	100%	100%	100%	0%	0%

Tabla 5.1: Tabla de resultados al realizar gestos en frente de la Kinect a distintas distancia y orientaciones.



(a) Esqueleto con Kinect a 0°

(b) Esqueleto con Kinect a -45°

Figura 5.3: Esqueletos visualizados mediante kinect. a) Esta visualización es con el usuario posicionado con una orientación de 0° grados en relación a la kinect. b) En este caso el usuario ha girado -45° manteniendo la postura, se puede observar como la kinect acerca virtualmente la mano al cuerpo a pesar de mantener la misma postura.

Por último hemos realizado pruebas con una segunda persona que no había utilizado el sistema de reconocimiento de gestos, estas pruebas fueron realizadas con el fin de comprobar la efectividad del sistema con los gestos grabados anteriormente y con otros grabados por él mismo. Antes de realizar las pruebas se explicó brevemente al usuario las acciones a realizar para la detección de los gestos y se realizaron breves pruebas. Pudimos detectar dificultad en el reconocimiento de gestos para este usuario, debido a lo "enérgicos" que eran sus movimientos. El usuario realizaba los movimientos a una velocidad elevada dificultando la recogida de información suficiente para la ventana del buffer y su posterior reconocimiento con Dynamic Time Warping. Después de un poco de entrenamiento fue capaz de realizar los movimientos con la soltura suficiente, para que el sistema de reconocimiento de gestos reconociese sus movimientos.

Una vez el usuario era capaz de realizar los movimientos con una soltura mínima, le indicamos que se posicionase en frente del dispositivo a una distancia de un metro y una orientación de 0° grados a la kinect. Indicamos al usuario que hiciese los gestos "Siguiente", "Anterior" y "Aceptar" (Gestos grabados en la prueba anterior) diez veces. Posteriormente realizamos una grabación personalizada por el usuario donde se le solicitó que hiciese los mismos movimientos para cada gesto y repetimos la prueba anterior.

Los resultados de dichas pruebas los podemos observar en la tabla 5.2, donde podemos ver un aumento en la tasa de acierto cuando el usuario realizaba gestos grabados por él. Comprobamos que esto es debido al modo en que realiza cada movimiento. A pesar de que la descripción dada al usuario para grabar los movimientos fue la misma utilizada por nosotros, es decir con el codo pegado al costado con movimientos hacia los lados para "Siguiente" y "Anterior", y en el caso de "Aceptar" moviendo la mano hacia el frente, el usuario realizaba los gestos con pequeñas diferencias, donde al realizar la grabación personalizada mejoró la tasa de aciertos. Por otro lado, el usuario al ser consciente del movimiento que había grabado, intentaba realizarlo lo más parecido posible en velocidad y movimiento.

	Pregrabados	Personalizados
Siguiente	70%	90%
Anterior	70%	80%
Aceptar	60%	80%

Tabla 5.2: Tabla con tasa de aciertos al realizar gestos por una persona que no había utilizado antes el sistema reconocimiento de gestos desarrollado.

En el siguiente enlace puede observarse el entrenamiento con los gestos y su reconocimiento: <https://goo.gl/photos/WEGoedMhsZedGvEZ6>.

5. Reconocimiento de gestos con Kinect

6

Aplicación de rehabilitación: Puzzle

El puzzle tiene como objetivo ser un ejercicio sencillo para personas con discapacidad cognitiva. Las personas que presentan esta situación, pueden tener dificultades para enfrentarse a diversos problemas, pudiendo ir acompañado también con limitaciones en las funciones motoras. Por ese motivo el sistema puede ser configurado con el dispositivo que mejor se adapte a su condición y resolver un puzzle muy sencillo que permita evaluar o mejorar el estado del paciente.

6.1 Desarrollo

Para realizar el puzzle que servirá de ejercicio de aplicación de rehabilitación cognitiva, hemos personalizado los botones de qt de manera que cada botón represente una pieza. Implementar el juego de esta manera nos permitirá trabajar de manera mas sencilla, con los recursos que nos ofrece las herramientas de desarrollo para el Tobii EyeX. Las piezas del puzzle tendrán asociadas trozos de una imagen que cargaremos en el sistema para utilizar en el puzzle.

La implementación del puzzle la hemos realizado separando "Lógica de juego" (Sección 6.1.1) de "Interfaz" (Sección 6.1.2).

6.1.1 Lógica de juego

La lógica del juego funcionará mediante un vector que tendrá la función de tablero. Este vector será iniciado con valores -1 y cada pieza del puzzle insertadas en él tendrán valores desde 1 a N, donde N es la cantidad de piezas y el tamaño del tablero. Para que el puzzle sea resuelto las piezas insertadas en el tablero deberán encontrarse en orden, es decir en la primera posición del tablero deberá encontrarse la pieza con identificador 1, en la siguiente posición la pieza con identificador 2, en la siguiente con identificador 3 y de manera consecutiva y ordenada las siguientes piezas hasta llegar a N.

Las acciones que permite el tablero son "Mover Pieza" e "Intercambiar piezas". Distinguimos como "Mover Pieza" cuando colocamos una pieza en el tablero o cuando la retiramos de este. La acción "Intercambiar piezas" se corresponde al objetivo de

6. Aplicación de rehabilitación: Puzzle

intercambiar las posiciones de dos piezas ya colocadas en el tablero.

Para informar a la interfaz de las acciones que son realizadas en el juego serán lanzados 5 posibles eventos:

- Pieza extraída del tablero.
- Pieza insertada en la posición X del tablero.
- Piezas intercambiadas en el tablero.
- Victoria.
- Partida reiniciada.

6.1.2 Interfaz

La pantalla para resolver el puzzle (Figura 6.1) está compuesta por dos zonas, una columna a la izquierda con las piezas del puzzle y un tablero a la derecha donde serán colocadas las piezas. El puzzle puede resolverse utilizando cualquiera de los dispositivos o la pantalla táctil. Cuando usamos los dispositivos estos desplazarán el foco (Punto B de la figura 6.1) entre las distintas piezas del puzzle y el tablero hasta seleccionar el punto que desean y ejecutar la acción aceptar. Dicha acción seleccionará la pieza cambiando el tamaño y borde de la pieza seleccionada (Punto A de la figura 6.1) para seleccionar luego en el tablero la zona de destino.

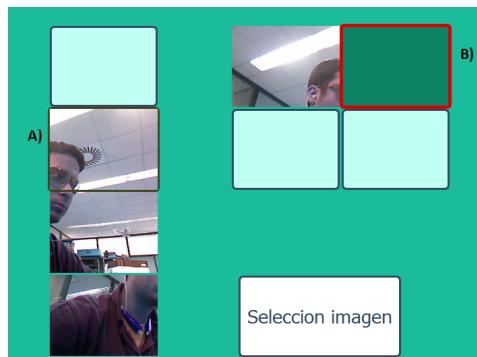


Figura 6.1: Puzzle de cuatro piezas con imagen seleccionable. A) Pieza seleccionada. B) Zona del tablero que tiene el foco y donde puede colocarse la pieza seleccionada.

Para implementar la interfaz hemos utilizado dos vectores que contendrán varios "QPushButtons" vinculados a un identificador. Los vectores funcionarán a modo de "mesa" y "tablero". El vector "tablero" contendrá las piezas que han sido colocadas para resolver el puzzle y el vector "mesa" contendrá las piezas que aún no han sido colocadas en el tablero. En cada movimiento la interfaz comunicará al servicio con la lógica del juego la acción a realizar.

Cuando seleccionamos una imagen la dividimos utilizando funciones de la librería

OpenCV¹ en la cantidad de piezas que tendrá el puzzle y vinculamos cada trozo de izquierda a derecha, de arriba a abajo, de manera ordenado un identificador del 1 a N, donde N es la cantidad de trozos de la imagen. Esta vinculación la realizamos en el vector "mesa".

Cuando la imagen ha sido troceada y vinculada, se informa al servicio con la lógica del juego un reinicio de partida, generando un evento que mezclará las piezas en la mesa. Para que la interfaz manifieste los eventos generados por el servicio, le hemos suscrito la interfaz para recibir los eventos generados y manifestar visualmente cada acción.

Para implementar las funciones de la interfaz hemos definido la clase "QPiecePuzzle" heredando de la clase "QPushButton" de Qt. Esta herencia permitirá controlar los efectos visuales utilizados para señalar el momento en que una pieza se encuentra seleccionada o tiene el foco. El control de los efectos visuales lo realizamos de esta manera, en vez de utilizar únicamente "Qt Style Sheet", porque resulta más conveniente para controlar un pequeño ajuste de posición al cambiar el tamaño de las piezas. Este variación del tamaño lo realizamos para resaltar el cambio en el grosor de los bordes, cuando los botones se encuentran con el foco o seleccionadas. Por otro lado facilita el configurar una restricción, en la que como máximo solo podrá haber una única pieza seleccionada en el tablero y la mesa. Teniendo como atributo estático de clase un puntero, señalando a la última pieza del puzzle que generó el evento "click", podemos controlar la restricción mencionada. Si ya hay una pieza seleccionada y se selecciona otra distinta a ella misma, el puntero cambiará su valor a NULL, en cambio si no hay ninguna pieza seleccionada en el puntero y se selecciona una, se guarda en el puntero una referencia a dicha pieza.

Al completar el puzzle la lógica de juego genera un evento que es capturado por la interfaz para escribir un mensaje de victoria en la pantalla (Figura 6.2). Para volver a empezar la partida se deberá seleccionar otra imagen.



Figura 6.2: Mensaje de victoria

¹Libería para procesamiento de imágenes y visión por computador. Versión utilizada 2.4. Sitio web: <http://opencv.org/downloads.html>

6. Aplicación de rehabilitación: Puzzle

En los siguientes enlaces se puede observar la resolución del puzzle utilizando la Kinect o el Tobii EyeX:

- Kinect <https://goo.gl/photos/U9VC9e9f44HPpHKy8>.
- Tobii EyeX <https://goo.gl/photos/HqN2opEJv71HTLN7A>.

7

Conclusión

En este trabajo hemos realizado la implementación de un sistema de reconocimiento de gestos con las manos que permite personalizar los gestos a realizar, hemos añadido la posibilidad de generar acciones con los ojos sin utilizar una tecla de acción y por último hemos realizado la implementación de un aplicación sencilla con el objetivo de servir como ejercicio de rehabilitación cognitiva y en el que poder utilizar los sistemas de interacción implementados. El conjunto de las funciones implementadas en este proyecto se integran dentro del contexto del proyecto nacional SIRMAVED, que es un proyecto que tiene como objetivo dar asistencia personalizada a pacientes con daño cerebral mediante un sistema multisensor robotizado.

En lo referente al reconocimiento de gestos con las manos utilizando KinectV2, observando los resultados obtenidos en las pruebas (Sección 5.2) el sistema funciona bastante bien cuando se conoce el modo de empleo y se entrena los gestos a utilizar. Por otro lado encontramos limitaciones cuando se utiliza el sistema con orientaciones laterales, dificultando la correlación de las posiciones de las manos realizada por Kinect y en consecuencia dificultando el reconocer los gestos. En nuestro proyecto decidimos escoger únicamente uno de los joints de las manos que nos proporciona el esqueleto generado por la Kinect. Esta decisión fue tomada para evitar los errores en la correlación de los dedos de las manos, de manera que un usuario al personalizar los gestos no tenga problemas en la grabación de ellos.

Por otro lado nuestro sistema de reconocimiento de gestos es robusto para su uso en entorno con baja luminosidad, permitiendo utilizarlo en sitios oscuros.

En el caso de TobiiEyeX hemos dotado al sistema con la capacidad de interactuar con él reconociendo guiños o pestañeos y por otro lado podemos vincular perfiles de calibración con los usuarios. Las limitaciones antes las que nos encontramos en el reconocimiento de guiños y pestañeos con este dispositivo es su rango de distancia, limitando el espacio sobre el que podrá interactuar el paciente. Personalmente me ha sorprendido la precisión que tiene el dispositivo, si bien es verdad que en objetos muy pequeños puede tener problemas para detectar que están siendo observados, si se calibra de la manera adecuada y las dimensiones del elemento de la interfaz son razonables se pueden realizar aplicaciones potentes y accesibles a todo tipo de usuarios.

Por último la aplicación de rehabilitación cognitiva puede ser utilizada con cualquiera de los dispositivos disponibles en el sistema SIRMAVED, de este modo indiferente-

mente del dispositivo que haya escogido el usuario para interactuar con el sistema podrá ser utilizada.

Los resultados de nuestro proyecto han sido integrados en SIRMAVED, proyecto que ha sido desplegado para ser probado con pacientes reales.

7.1 Trabajos futuros

Observando los resultados obtenidos con la KinectV2 comprobamos que los usuarios podían presentar problemas al interactuar con el sistema de reconocimiento de gestos, debidos principalmente por el tiempo de espera entre gestos. Esta acción resulta necesaria para evitar que acepte varias veces el mismo gesto o se confundan al volver a un punto inicial. Por ese motivo como una mejora visual planteamos el añadir en la interfaz una pequeña esfera que cambie de color cuando se pueda realizar un gesto.

Por otro lado también podrían realizarse más fichas de rehabilitación y dar la posibilidad de reconocer más gestos con las manos.

En lo relacionado con el dispositivo de seguimiento ocular, también sería interesante el generar mapas de calor en terapias de rehabilitación alternativa, con el fin de dotar al sistema con más posibilidades para los terapeutas y pacientes.

Bibliografía

- [Bulling et al., 2011] Bulling, A., Ward, J. A., Gellersen, H., and Tröster, G. (2011). Eye movement analysis for activity recognition using electrooculography. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(4):741–753.
- [Debeljak et al., 2012] Debeljak, M., Ocepek, J., and Zupan, A. (2012). Eye controlled human computer interaction for severely motor disabled children. In *Lecture Notes in Computer Science*, pages 153–156. Springer Science + Business Media.
- [Díaz-Tribaldos et al., 2015] Díaz-Tribaldos, M. R., Escobar-Ocampo, J. M., and Vivas-Albán, Ó. A. (2015). Interfaz haptica tipo guante con realimentación vibratoria. *Revista EIA*, 12(23):29–39.
- [Garg et al., 2009] Garg, P., Aggarwal, N., and Sofat, S. (2009). Vision based hand gesture recognition. *World Academy of Science, Engineering and Technology*, 49(1):972–977.
- [Hassan Montero and Herrero Solana, 2007] Hassan Montero, Y. and Herrero Solana, V. (2007). Eye-tracking en interacción persona-ordenador. *No solo usabilidad*, (6).
- [Hof, 2015] Hof, R. (2015). How Do You Google? New Eye Tracking Study Reveals Huge Changes. <http://www.forbes.com/sites/roberthof/2015/03/03/how-do-you-google-new-eye-tracking-study-reveals-huge-changes/#280d89b71252>. [Online; accedido 22-Marzo].
- [Khoshelham and Elberink, 2012] Khoshelham, K. and Elberink, S. O. (2012). Accuracy and resolution of kinect depth data for indoor mapping applications. *Sensors*, 12(2):1437–1454.
- [Kim et al., 2009] Kim, J.-H., Thang, N. D., and Kim, T.-S. (2009). 3-d hand motion tracking and gesture recognition using a data glove. In *Industrial Electronics, 2009. ISIE 2009. IEEE International Symposium on*, pages 1013–1018. IEEE.
- [Li et al., 2006] Li, D., Babcock, J., and Parkhurst, D. J. (2006). openeyes: a low-cost head-mounted eye-tracking solution. In *Proceedings of the 2006 symposium on Eye tracking research & applications*, pages 95–100. ACM.
- [Matsusaka et al., 2009] Matsusaka, Y., Fujii, H., Okano, T., and Hara, I. (2009). Health exercise demonstration robot taizo and effects of using voice command in robot-human collaborative demonstration. In *Robot and Human Interactive Communication, 2009. RO-MAN 2009. The 18th IEEE International Symposium on*, pages 472–477. IEEE.
- [Müller, 2007] Müller, M. (2007). Dynamic time warping. *Information retrieval for music and motion*, pages 69–84.

- [Murray, 2013] Murray, G. (2013). Rotation about an arbitrary axis in 3 dimensions. *Online* <http://inside.mines.edu>.
- [Tech, 2001] Tech, T. (2001). Tobii EyeX. <http://www.tobii.com/xperience/>. [Online; accedido Marzo 2016].
- [Wade and Tatler, 2009] Wade, N. J. and Tatler, B. W. (2009). Did java measure eye movements during reading? *Journal of Eye Movement Research*, 2(5):1–7.
- [Wang and Popović, 2009] Wang, R. Y. and Popović, J. (2009). Real-time hand-tracking with a color glove. *ACM transactions on graphics (TOG)*, 28(3):63.
- [Wu and Huang, 1999] Wu, Y. and Huang, T. S. (1999). Vision-based gesture recognition: A review. In *Gesture-based communication in human-computer interaction*, pages 103–115. Springer.