Modelos innovadores de interacción Persona-Computador mediante reconocimiento gestual

Innovative Models of Human-Computer Interaction with gesture recognition

Josué Candela Perdomo
Departamento de Ingeniería Informática y Sistemas
Escuela Técnica Superior de Ingeniería Informática

Trabajo de Fin de Grado

La Laguna, 09 de julio de 2014

D. **Jesús Miguel Torres Jorge**, con N.I.F. 43.826.207-Y Profesor Contratado Doctor adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna

Y

D. **José Demetrio Piñeiro Vera** con NIF 43.774.048-B Profesor Titular Doctor adscrito al Departamento de Ingeniería Informática y Sistemas de la Universidad de La Laguna

C E R T I F I C A

Que la presente memoria titulada:

"Modelos innovadores de interacción Persona-Computador mediante reconocimiento gestual."

ha sido realizada bajo su dirección por D. Josué Candela Perdomo, con N.I.F. 78.587.291-V.

Y para que así conste, en cumplimiento de la legislación vigente y a los efectos oportunos firman la presente en La Laguna a 9 de julio de 2014

Agradecimientos

En primer lugar, mis padres y abuelos por haberme apoyado hasta conseguir todo lo que he querido y guiarme cuando más lo he necesitado.

A Jesús Torres, director del proyecto, por las horas de atención dedicadas y por toda la ayuda prestada.

Resumen

El objetivo de este trabajo ha sido la investigación del estado actual de los dispositivos de disponibles de interacción natural y el desarrollo de nuevos métodos de interacción Persona-Computador mediante el reconocimiento gestual.

Para ello, tras la elección del dispositivo más apropiado se ha realizado un trabajo de desarrollo para determinar las posibilidades y limitaciones actuales. En este desarrollo se ha elaborado una API para la creación de un prototipo que permitiese evaluar las capacidades del dispositivo.

Como entorno de pruebas, y una vez elaborada la API, se ha desarrollado una simulación de un sistema operativo que permite probar la estandarización planteada así como los controles de interfaz específicos desarrollados para la interacción gestual.

Palabras clave

Interacción Persona-Computador, Reconocimiento Gestual, LEAP Motion, Accesibilidad, Interacción Natural.

Abstract

The purpose of this work has been to investigate the current state of available natural interaction devices and the development of new methods of Human-Computer interaction through the use of gesture recognition.

In order to accomplish this, after selecting the most appropriate device, development work took place to determine current possibilities and limitations. Through this work an application programming interface (API) was developed with the intent to create a prototype that would allow for the evaluation of the device's capabilities.

As a testing environment, and once the API development took place, a simulation of an operating system was developed to allow for the testing of the suggested standardization as well as the specific interface controls developed for gesture interaction.

Keywords

Human-computer interaction, Gesture Recognition, LEAP Motion, Accessibility, Natural User Interface.

Índice General

Índice de figuras

Índice de tablas

Capítulo 1. Introducción

1.1 Antecedentes y Estado actual del tema

En los últimos años se va perfilando lo que parece ser la alternativa al sistema de interacción tradicional. Parece que esta evolución pasa por desprenderse cada vez en mayor medida del soporte físico, en busca de una interfaz más natural e intuitiva que dependa más del movimiento de nuestro cuerpo que de hardware externo.

En los últimos años las formas de comunicación entre el usuario y el ordenador han ido evolucionando. Los primeros ordenadores solo contaban con un lector basado en tarjetas perforadas como única forma de introducción de datos. Los avances en hardware desde entonces han sido numerosos y muy importantes, ya que gracias a ellos se ha mejorado notablemente la accesibilidad y la usabilidad de los propios ordenadores.
Merece especial mención la aparición del ratón, diseñado por Douglas Engelbart durante los años 60 y mejorado después por la compañía Xerox. Este artilugio, unido a la interfaz gráfica de usuario que hizo posible, abrió el mundo de los ordenadores a una gran parte de la población, que hasta entonces no había podido acceder a ellos. Con este sistema, un usuario podía utilizar un ordenador sin conocer ningún lenguaje de programación, algo impensable para la época. No obstante, pese a que siguen produciéndose avances en hardware año tras año, este sistema de comunicación basado en la combinación de teclado y ratón, aunque con ligeras mejoras, sigue intacto casi 40 años después. Actualmente, con cierta frecuencia, surgen nuevas tecnologías, como las basadas en pantallas táctiles o los últimos sistemas de detección de movimiento (con mayor importancia en el campo de los videojuegos) pero ninguna de éstas parece estar a la

altura de los sistemas tradicionales. Cabe preguntarse entonces si se ha alcanzado ya la mejor interfaz posible o si aún está por descubrir un sistema de interacción mejor. Realmente, si observamos detenidamente los avances que se van sucediendo, podemos apreciar que se va perfilando lo que parece ser la alternativa al sistema de interacción tradicional. Concretamente parece que esta evolución pasa por desprenderse cada vez en mayor medida del soporte físico, en busca de una interfaz más natural e intuitiva que dependa más del movimiento de nuestro cuerpo que de hardware externo. Esto se ha podido ver con claridad en un mundo siempre cercano a la tecnología, como es el de los videojuegos. Primero, el sistema Wii de Nintendo propuso un cambio radical en la forma de interactuar con los videojuegos con un mando con sensor de movimiento. Recientemente la empresa Microsoft ha querido dar un paso más en este sentido suprimiendo cualquier tipo de controlador físico con su revolucionario Kinect; una interfaz sin mandos compuesta por un dispositivo que cuenta con un proyector y una cámara IR y que reconoce directamente nuestro cuerpo, o una parte de él. Otra interfaz, a primera vista bastante atractiva, son los *data gloves* o *wired gloves*. Un dispositivo con forma de guante que se acopla a la mano y que cuenta con una serie de sensores que capturan el movimiento y los gestos de la mano. No obstante estos dispositivos no han tenido la aceptación que se esperaba, quizás debido a su alto coste y también a que no se ha dado el apoyo necesario por parte de los desarrolladores de software. Otro dispositivo que ha aparecido hace poco en el mercado es el sensor Leap. Su funcionamiento es muy similar a Kinect, pues parece que basa en un sistema de cámara y sensor IR que reconocen la posición y movimientos de la mano. Sus desarrolladores prometen que tiene una precisión mucho mayor que la del periférico de Microsoft. Además se ha puesto a disposición de los desarrolladores un SDK de forma gratuita. Todo esto unido a un precio de salida no demasiado elevado lo convierte en una opción a tener en cuenta en el futuro.

1.2 Objetivos

Los objetivos que se siguen con el desarrollo de este proyecto son:

- Evaluar el rendimiento de los diferentes dispositivos de reconocimiento gestual que existen en el mercado.
- Desarrollar un software que sea capaz de determinar la postura de una mano usando la información de estos sensores.
- Desarrollar una aplicación de demostración que se pueda beneficiar del uso de este tipo de interfaces gestuales.

Capítulo 2. Revisión Bibliográfica

A continuación se expondrán las conclusiones obtenidas tras estudiar los dispositivos disponibles en el mercado para interacción natural.

2.1 Kinect

El dispositivo Kinect fue comercializado el 4 de Noviembre de 2010 por la compañía Microsoft para la consola Xbox 360. Basado en tecnología de la compañía PrimeSense Ltd,

Kinect cuenta con una cámara RGB (640x480), así como de una cámara IR de profundidad (640x480), un sensor CMOS, un micrófono de múltiples matrices y un procesador a medida. Todo ello permite un reconocimiento 3D del esqueleto del usuario así como reconocimiento facial. Su matriz de micrófonos permiten reconocimiento de voz, localización del usuario por audio y supresión de sonido de ambiente.

2.2 LEAP Motion

El dispositivo LEAP Motion fué comercializado en Julio de 2013 por la compañía Leap Motion, Inc. El dispositivo hace uso de una tecnología simple, basada en dos cámaras IR y 3 LEDs infrarrojos, transfiriendo la información con un máximo de 290 fps. Este sistema permite la detección de la posición con una precisión de 0.01mm.

El SDK de LEAP permite la detección de los dedos de la mano de forma independiente así como herramientas (como lápices o palillos) hasta 1 metro de distancia. Igualmente, incluye el reconocimiento de gestos por defecto, como círculos, deslizamientos y toques así como acciones de traslación, rotación y escalado.

Incluso en su temprano estado, LEAP Motion es un dispositivo con gran potencial para el reconocimiento gestual, suponiendo que este reconocimiento se centra en la mano del usuario. Actualmente se ha lanzado su actualización v2 que mejora considerablemente su capacidad y rendimiento.

2.3 SoftKinetic DS325

La empresa SoftKinetic se presenta como alternativa a LEAP Motion con su nuevo dispositivo DS325. Este dispositivo es un sensor de profundidad en tiempo de vuelo que permite la interacción y reconocimiento gestual a distancias cortas (de 0,15m a 1.0m). Dispone de dos lentes, una RGB a una resolución de 720p HD, y una de profundidad a 320 x 240 | QVGA. Toda esa información es tratada por un sensor CMOS DepthSense y por un controlador interno a 60fps. También dispone de un acelerómetro 3-axis así como de dos micrófonos.

2.4 PlayStation Eye

PlayStation Eye es un dispositivo, similar a una webcam, desarrollado para la consola PlayStation 3 y comercializado en Octubre de 2007. El dispositivo utiliza directamente (implementadas mediante software) tecnología de visión por computador así como reconocimiento gestual para procesar las imágenes capturadas por la cámara. La cámara es capaz de capturar video a 640x480 a 60Hz y 320x240 a 120Hz.Al igual que Kinect, dispone de una matriz de micrófonos que permiten la localización de la voz, cancelación de eco y supresión de ruido de ambiente.

2.5 Controladores

2.5.1 WiiMote

El Wii Remote Control fue comercializado en Noviembre de 2006 para la consola Wii de Nintendo y supuso la democratización de los dispositivos de reconocimiento gestual, al igual que el triunfo del reconocimiento gestual aplicado al mundo de las consolas. El dispositivo cuenta con un acelerómetro así como con un sensor óptico (PixArt) que permite determinar el lugar que se está apuntando. Por otro lado, el WiiMote viene acompañado de una barra de sensores, la cual distribuye LEDs a sus extremos para permitir el posicionamiento del cursor del WiiMote así como la distancia entre el WiiMote y la barra de sensores hasta 5 metros de distancia.

2.5.2 PlayStation Move

El dispositivo PlayStation Move salió a la venta en Octubre/Septiembre de 2010 como respuesta al WiiMote de Nintendo para la consola PlayStation 3 y, a su vez, como competencia de Kinect. PlayStation Move utiliza una esfera coloreada mediante LEDs en un extremo, que junto con PlayStation Eye, permiten el posicionamiento del dispositivo así como su distancia, ya que el tamaño de la esfera es conocida por la cámara. PlayStation Move también dispone de dos sensores de inercia, acelerómetro y un sensor de velocidad angular que permiten obtener la rotación así como el movimiento.

2.6 Próximas alternativas

2.6.1 Myo

El equipo de la startup Thalmic Labs plantea, para mediados de 2014, un sistema distinto para el reconocimiento gestual, basado en un brazalete que funciona a modo de escáner de nuestros impulsos eléctricos musculares. Esto lo consigue con un sensor de actividad muscular y una unidad de medición de inercia (9–axis).

2.6.2 ATAP Project Tango

Google lleva los smartphones un paso adelante con su proyecto Tango, un dispositivo móvil (actualmente de 5") que dispone de una cámara de 4 megapíxeles, una cámara de detección de movimiento y un sensor de profundidad. Para gestionar esta información, dispone de dos procesadores Myriad 1. Su finalidad actual es la elaboración de mapas 3D y se desconocen sus capacidades para el reconocimiento de gestos o figuras humanas.

Capítulo 3. Planteamiento del desarrollo

Durante este capítulo se plantearán distintos prototipos a desarrollar haciendo uso del reconocimiento gestual con el fin determinar las posibilidades del hardware seleccionado e investigar nuevas vías de interacción natural. Posteriormente se realizará una selección para su desarrollo, no obstante, precederemos a detallas los demás planteamientos para dejarlos abiertos a nuevas líneas de investigación.

3.1 Navegación por gestos

3.1.1 Descripción

El objetivo de este prototipo sería utilizar la tecnología desarrollada de reconocimiento gestual para plantear un estándar de navegación en interfaces gráficas. Se intentaría crear una interfaz entre el usuario y el sistema operativo que permitiera, mediante gestos, realizar acciones normales (tales como mover, abrir, copiar, eliminar, ...) con la mayor naturalidad posible.

3.1.2 Planteamiento y factibilidad

Se trataría de un proyecto factible aún con las limitaciones del dispositivo, siendo posible siempre y cuando los gestos fuesen lo suficientemente sencillos para detectarlos, es decir, con los dedos lo más separados posible y sin ocultarse entre ellos a vista de la cámara de profundidad. La mayor dificultad la encontraremos al intentar conseguir una experiencia de usuario agradable con interfaces naturales mediante realización de los gestos por parte del usuario así como minimizar el cansancio del usuario durante la navegación. Para ello, el movimiento podría realizarse mediante *scroll* únicamente vertical u horizontal, la

selección de un fichero mediante la realización de un círculo sobre un fichero o al usar dos dedos.

Figura 1 - *Mockup* de sistema de navegación

Una vez seleccionado un elemento interactivo sería natural la necesidad de realizar acciones sobre él, para ello, deberán mostrarse menús contextuales relativos al elemento seleccionado.

Figura 2 - Mockup de menú contextual

3.2 Teclado virtual

3.2.1 Descripción

Este prototipo intentaría replicar el comportamiento de un teclado físico intentando encontrar la adaptación más correcta de este tradicional dispositivo de interacción al dispositivo LEAP. Para ello se podría hacer uso de triangulación respecto a posiciones de teclas solicitadas al usuario de forma previa a la inserción del texto o a la reagrupación de las teclas mediante sectores de dedos. Pudiendo incluso la opción al usuario de realizar esta agrupación por sí mismo. Así como cualquier otra adaptación para resolver el problema de la entrada de texto mediante dispositivos de interacción natural.

3.2.2 Planteamiento y factibilidad

Dada la dificultad de esta aplicación se han analizado diversas altern ativas para su desarrollo que se plantean a continuación:

Teclado tradicional

La elaboración de un teclado virtual al estilo tradicional, tal como se ha planteado inicialmente, se encuentra con el inconveniente de la pérdida d el tracking en dedos del usuario, así como la imposibilidad de permitir al us uario descansar las manos sobre una superficie. Aunque se ha planteado la p osibilidad de instalar el dispositivo LEAP en distintas orientaciones, no pare ce reaccionar de la forma esperada.

Teclado morse

Algo más plausible sería la elaboración de un teclado que requiera una interacción mínima por parte del usuario, una forma sería la elaboración de una codificación, por ejemplo morse, que permita escribir con

solo dos dedos (un para el punto y otro para raya) o incluso solo con uno (movimiento arriba correspondería a punto y abajo raya). Aunque se encuentra con la limitación de no ser nada intuitivo, contará con la ayuda visual de la GUI que indicará al usuario que letra puede formar con el movimiento de un dedo u otro. Por ejemplo:

Figura 3 - Edición de texto en morse trás una raya

El usuario se encuentra escribiendo en un campo de texto como el siguiente y quiere escribir el carácter O, la interfaz le mostraría los caracteres que comienzan por raya y que puede obtener si continua con raya o punto. Por defecto tendría el carácter 'T' ya que es el carácter asociado a raya. Una vez introducida una segunda raya, el editor mostraría los posible caracteres que comienzan con dos rayas.

Figura 4 - Edición de texto en morse trás dos rayas

En términos de usabilidad sería un poco desconcertante que los caracteres GQZ7 pasasen de estar en la parte inferior a la parte superior, pero la

intención, al introducir una segunda raya, es rechazar todos los caracteres BCDKNXY y dividir los caracteres GMOQZ6789.

Capítulo 4. Propuesta de Solución

Una vez planteadas las aplicaciones y determinadas las limitaciones del dispositivo nos decantamos por el desarrollo de la aplicación que nos ofrece una mayor propuesta de valor con el menor desarrollo. Determinando a su vez de gran urgencia e importancia, el desarrollo de una interfaz y un sistema de exploración de ficheros usable en la interacción con interfaces naturales.

El alcance de este proyecto abarcará lo siguiente:

- Desarrollo de una API para la elaboración de interfaces web para dispositivos LEAP Motion.
- Implementación gestual para la navegación entre directorios y ficheros.
- Implementación gestual para la ejecución de comandos básicos (copiar, pegar, eliminar, abrir, ...)

Para desarrollar el proyecto se diseñará una interfaz específica para la interacción gestual, de forma que la interacción se desarrolle de la forma más intuitiva posible. El flujo de la aplicación será diseñada teniendo en cuenta el tipo de navegación que se realiza actualmente en los *smartphones* ya que es la interacción natural más efectiva en la actualidad.

Para representar nuestro sistema de ficheros y dado que nuestro proyecto se realiza en HTML, crearemos un sistema simulado definido en JSON en el interior de nuestra aplicación a modo de prueba.

Planteamiento de funcionamiento

Inicialmente, encontraremos el escritorio donde el usuario podrá establecer sus accesos directos a lugares o aplicaciones localizados en el sistema de ficheros. Será desde el escritorio, desde donde podremos acceder a la exploración de las aplicaciones disponibles y desde ahí al explorador del sistema de ficheros.

Para seleccionar un elemento interactivo, deberemos posicionarnos sobre el y cerrar completamente nuestra mano, lo que dará lugar, en el caso del explorador, a la expansión del contenido de dicha

El usuario siempre podrá abandonar la carpeta actual y volver a la anterior utilizando el botón designado para ello. A su vez, dispondrá en todo momento de un botón para volver al escritorio.

Como gesto para desplegar el total de las acciones disponibles, utilizaremos dos dedos, es decir, si nos localizamos sobre un elemento interactivo (carpeta o fichero) y dejamos dos dedos reconocibles, desplegaremos su menú contextual. Si un elemento solo tiene una acción contextual se ejecutará automáticamente.

Capítulo 5. Revisión del Hardware

Tratando de encontrar un modelo innovador para interactuar con ordenadores mediante reconocimiento gestual, se estima de gran importancia el reconocimiento de cada uno de los dedos de la mano con la finalidad de reconocer con la mayor precisión posible los gestos del usuario. Más aún, cuando se ha determinado que el reconocimiento gestual íntegro produce agotamiento al usuario.

Por ello, y teniendo en cuenta las especificaciones de los dispositivos estudiados, consideramos que LEAP Motion y SoftKinect DS325 son dispositivos que se ajustan de igual forma al reconocimiento con dichas especificaciones, es decir, realizando un reconocimiento a corta distancia, localizado en la parte relevante del usuario a la hora de realizar la interacción con el sistema.

Como elemento técnicamente trivial pero decisivo, nos quedamos con el precio, siendo LEAP Motion un producto mucho más asequible para el consumidor ($79 contra $249 de SoftKinect DS325).

5.1 Funcionamiento

A continuación detallaremos el funcionamiento del dispositivo LEAP Motion, así como los problemas encontrados durante el desarrollo.

El SDK (JavaScript) disponible para el dispositivo tiene una implementación sencilla que nos permite comenzar la recepción de datos prácticamente con unas líneas de código.

Los datos enviados por el dispositivo se basan en un objeto llamado Frame que contiene varios objetos y listas que describen la detección. Entre los campos más relevantes encontramos:

Fingers	Lista de dedos detectados en el frame
Hands	Lista de manos detectadas en el frame
Pointables	Lista de *pointables* (dedos y herramientas) detectadas en el frame
Gestures	Lista de gestos reconocidos en el frame

Tabla 1 - LEAP, Descripción del objeto Frame

Hay que tener en cuenta que el dispositivo es capaz de reconocer gestos de forma nativa, ya que el SDK nos da la posibilidad de detectarlos. En caso de que el usuario realizará alguno de los gestos definidos por el SDK, la lista de gestos lo contendría. Estos gestos son:

Swipe	Gesto horizontal o vertical brusco con alguno de los dedos
Circle	Gesto circular brusco con alguno de los dedos
Tap	Gesto punzante con alguno de los dedos

Tabla 2 - LEAP, Descripción de gestos disponibles

Todos estos gestos son parametrizables con el fin de ajustar lo máximo posible su detección.

Para este proyecto se ha hecho uso principalmente de la versión 0.6.1 del SDK del dispositivo. Durante el desarrollo se puso a disposición del público la versión v2 que permitía, entre otras características, obtener el esqueleto de la mano del usuario de forma similar a como funciona Kinect. A pesar de ello, el proyecto únicamente hace uso de las posiciones de los

pointables devueltos por la API, los cuales representan las posiciones de los dedos en el espacio independientemente de su conexión con la mano.

Otro punto mejorado con la nueva versión es la asignación y reasignación de Ids a los *pointables* al ser detectados, es decir, si un dedo ya no es detectado, la capacidad del sistema para reasignarle el mismo Id al detectarlo de nuevo.

Uno de los puntos mejorados en la nueva versión es el continuo reconocimiento de los dedos incluso en la acción de *grab*, lo cual en nuestro desarrollo (en donde nos basamos en los *pointables* detectados para determinar la acción a realizar) supuso un grave problema que se solucionó añadiendo flexibilidad a la hora de la detección y priorizando correctamente las acciones del usuario en función de la detección del dispositivo,

Los principales problemas detectados han sido ocasionados por la iluminación, añadiendo en algunas ocasiones excesivo ruido al reconocimiento y problemas de solapamiento al posicionar la mano en posición vertical o perpendicular al dispositivo, impidiendo así la correcta detección de los dedos.

Capítulo 6. Desarrollo de Solución

Durante este capítulo se explicará en detalle la solución desarrollada así como su estado actual, aplicaciones y limitaciones.

6.1 API JavaScript

Para llevar a cabo la solución planteada de manera reutilizable era de vital importancia abstraer el desarrollo de forma que las herramientas aplicadas para desarrollar nuestro sistema operativo fuesen aplicables a cualquier proyecto.

6.1.1 Descripción

La API desarrollada permite la creación de interfaces web adaptadas para el dispositivo LEAP Motion, Esta API funciona mediante la creación de escenas a través de las cuales podemos ir transitando. Estas escenas a su vez, están formadas por vistas que permiten la interacción del usuario en varios entornos conocidos como 'Interfaz de área' e 'Interfaz contextual', ambas interfaces funcionan de forma independiente de forma excluyente, es decir, solo una de las dos se encuentra en funcionamiento.

6.1.2 Estructura

El proyecto se encuentra estructurado de la siguiente forma:

tlp3dgui.js	Nucleo de la API que gestiona el funcionamiento de las interfaces así como las transiciones, creación y eliminación de escenas.
tlp3dscene.js	Clase base que contiene la definición de escena y que contiene las vistas interactivas asociadas
tlpleap.js	Clase que funciona como interfaz entre nuestra API para interfaces y la API oficial de LEAP

	Motion.
utils	Carpeta que contiene ficheros con funciones útiles variadas para la API. p.e: Cálculo de distancias.
views	Carpeta que contiene las clases de vistas interactivas que serán adjuntadas a las escenas.

Tabla 3 - API, Estructura de ficheros

6.1.3 Interfaz de área

Las escenas permiten asociar elementos interactivos a ellas (vistas) y a su vez, estas escenas serán representadas en 3D en la interfaz de área. La interfaz de área se posiciona delante de la cámara y funciona como 'base' para la escena que se representa actualmente. Cuando se solicita un cambio de escena, se realiza una transición que deja a la interfaz de área fuera del rango de cámara lo que permite eliminar los objetos actuales y crear la escena solicitada. Al terminar la transición, la nueva escena está creada en la misma interfaz de área.

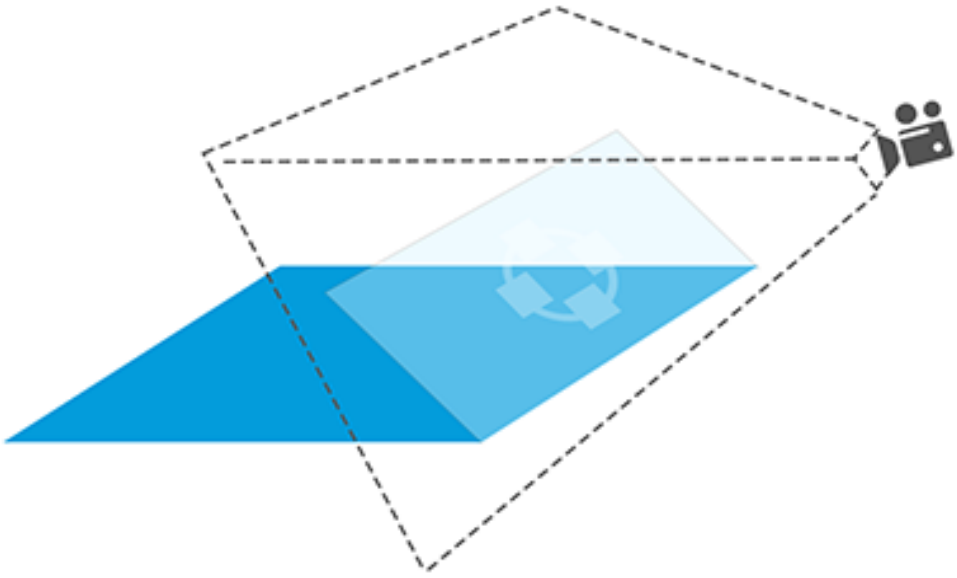

Figura 5 – Representación de la interfaz de área

La interfaz de área representa a modo de puntero, la posición normalizada de la mano del usuario, y es este puntero y su posición el que usamos para determinar si estamos o no interactuando con algún elemento de la interfaz de área.

6.1.4 Interfaz contextual

La interfaz contextual permite mostrar las acciones disponibles sobre un elemento interactivo que no sea de tipo SCROLL, La interfaz contextual se muestra cuando una vista se encuentra a menos de cierta distancia del puntero de la interfaz de área (vista enfocada) y el usuario usa solo dos dedos para interactuar. En ese momento, entendemos que el usuario está realizando una acción secundaria, que simboliza la apertura de la interfaz contextual en caso de que hayan múltiples acciones donde elegir. Si solo encontramos una acción disponible se realizará automáticamente.

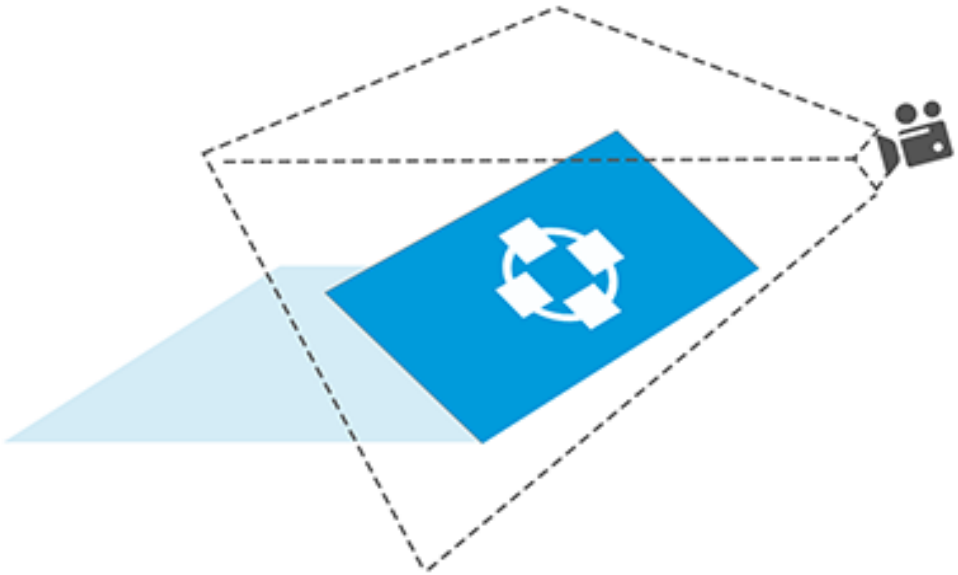

Figura 6 - Representación de la interfaz contextual

La interfaz contextual se reestructura automáticamente en forma radial dependiendo del número de acciones a mostrar.

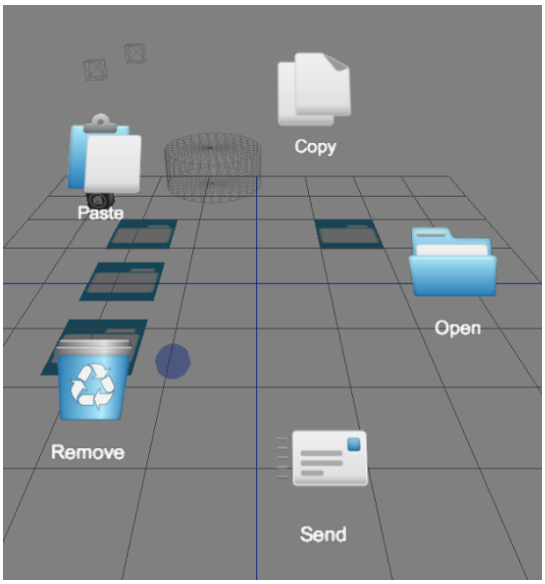

Figura 7 - Ejemplo de interfaz contextual

6.1.5 Vistas

Interactive3DView

Las vista interactiva permite al usuario interaccionar con el entorno 3D. Asociadas a ellas estás las acciones contextuales que se mostrarán en la interfaz contextual y de las cuales una está marcada como acción primaria. Al actualizarse, detecta si está cerca del puntero del usuario determinando que es una vista enfocada permitiendo, en cuanto el usuario realice un gesto de apretar (*grab*), ejecutar la acción contextual marcada como primaria, o en caso de interactuar con dos dedos, mostrar en la interfaz contextual las acciones disponibles sobre la vista.

Una vista permite realizar cualquier acción, y una acción puede contener cualquier código, al crear una acción contextual le pasamos como

parámetro una función a ejecutar al detectar que el usuario tiene esa intención.

ScrollView

La vista *scroll* está compuesta por múltiples vistas que son reestructuradas automáticamente a la hora de mostrarse. Esta vista, permite al usuario hacer *scroll* en un grupo de vistas siguiendo una orientación definida, que puede ser vertical u horizontal. Esta vista es altamente parametrizable y permite, además de indicar su orientación, definir el tamaño de los ítems que va a contener así como la separación entre ellos.

6.1.6 Ejemplos de uso

Dentro del proyecto encontraremos varios ficheros de ejemplo donde podremos ver cómo realizar su integración en HTML así como los ficheros que se deben incluir para su correcto funcionamiento.

Para demostrar la simplicidad de la API se mostrará cómo crear una escena básica con un elemento interactivo en ella (vista):

```
TLPLeap.init();
TLP3DGui.init();

//ID unico que usaremos para cargar o eliminar la escena
var SCENE_ID = "miIdDeEscena";
var scn = new TLP3DScene(SCENE_ID);
scn.addView(new Interactive3DView({ // añadimos la vista
    actions : [ // determinamos sus acciones contextuales
        new ContextualAction(
            "res/icon.png", //icono de la accion
            "Say Hello",    //nombre de la accion
            function(){     //funcionamiento al ejecutar la acción
                alert("HOLA MUNDO");
        }, true),  //indicamos que es una acción primaria
    ],
    position : {x: -100, y:100},
  size: 80
}));
```

```
TLP3DGui.sceneManager.add(scn);
TLP3DGui.sceneManager.select(SCENE_ID, false);
```

Este ejemplo nos introduce al gestor de escena que se encarga de cargar y eliminar las vistas de la interfaz de área. En su interior, contiene todas las escenas añadidas por el usuario e indexadas por un Id único que permite identificarlas.

6.1.7 Modo depuración

Con la finalizad de agilizar y mejorar el uso de la API como del ejemplo desarrollado, se ha establecido una variable global que permite indicar si estamos en modo depuración o no.

El modo depuración nos permite visualizar mediante *helpers* los límites virtuales de nuestra interfaz de área así como una representación de la mano y los dedos del usuario.

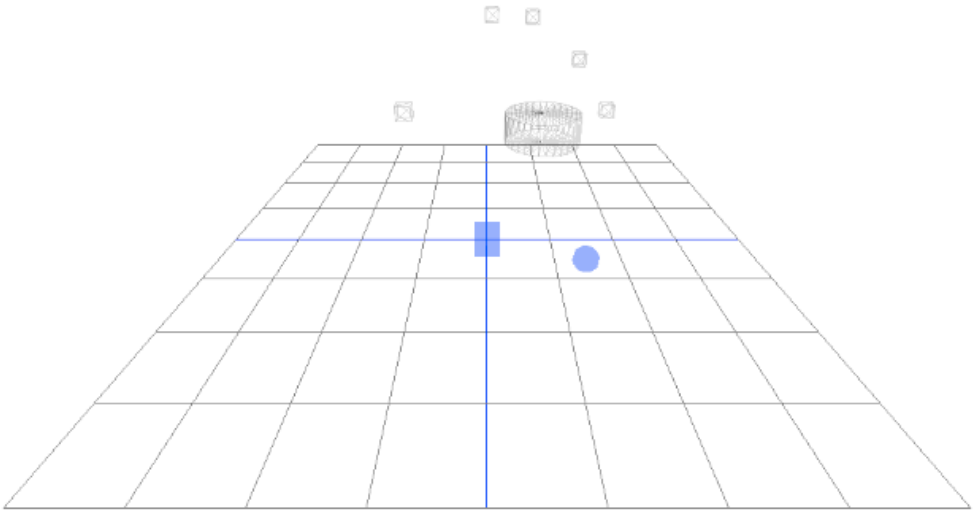

Figura 8 - Visualización en Modo Depuración activo

El modo depuración nos permite al mismo tiempo, alternar entre la cámara estándar y una cámara perpendicular a la interfaz de área para visualizar más fácilmente la posición de nuestro puntero y la fiabilidad del reconocimiento.

6.2 Reconocimiento gestual

La implementación del reconocimiento gestual supone la parte más importante del desarrollo ya que determina en gran medida como de usable es nuestra interfaz gráfica.

En nuestro sistema, hemos procurado detectar la interacción del usuario de la forma más sencilla posible y para ello hacemos principalmente uso del posicionamiento, tanto de la mano del usuario como de los dedos detectados. Con estos datos procedemos a detectar que tipo de interacción es la pretendida por el usuario.

En primer lugar, la localización de la mano nos permite posicionar el puntero tanto en la interfaz contextual como en la interfaz de area, lo que nos permite determinar que objeto es el enfocado por el usuario.

La detección de los dedos del usuario nos permite deducir la interacción deseada; en caso de detectar uno o ningún *pointable,* concluimos que el usuario se encuentra realizando una acción de *grab* lo cual puede suponer una acción de *scroll* o de apertura dependiendo del elemento enfocado.

Por otra parte, para detectar si un usuario quiere mostrar las opciones contextuales de un elemento, detectamos si el número de punteros es dos, para ello, nos hemos basado en el *click* secundario de los *trackpads* de Apple, los cuales hacen uso de este gesto para abrir los menús contextuales.

Con estos gestos y una detección básica podemos desarrollar una interfaz completamente interactiva que limita en menor medida al usuario.

6.3 Sistema Operativo (Ejemplo de navegación)

Siguiendo las indicaciones descritas en el planteamiento de la solución se ha desarrollado un proyecto de ejemplo que hace uso de la API.

En este proyecto se han investigado las posibilidades del dispositivo y se ha intentado desarrollar una interfaz interactiva que haga uso de los datos que nos devuelve.

En el siguiente enlace muestra un video con todo el contenido desarrollado en el ejemplo:

https://www.youtube.com/watch?v=Mvguyy6N9ho

6.3.1 Descripción

El sistema operativo permite navegar entre varias escenas que simulan el comportamiento de un *smarphone* pero aplicado a entornos web.

A grandes rasgos el proyecto nos permite navegar por las aplicaciones y ficheros de un sistema operativo ficticio y realizar acciones sobre su árbol de directorios.

6.3.2 Estructura

El proyecto se encuentra estructurado de la siguiente forma:

osgui.js	Núcleo del sistema operativo que gestiona el funcionamiento del explorador así como la creación de las escenas que lo componen y sus transiciones
osfileview.js	Clase que contiene la definición del objeto interactivo que representa un fichero

osappview.js	Clase que contiene la definición del objeto interactivo que representa una app

Tabla 4 - Sistema Operativo, Estructura de ficheros

6.3.3 Funcionamiento

El siguiente diagrama explica el flujo realizado por la aplicación:

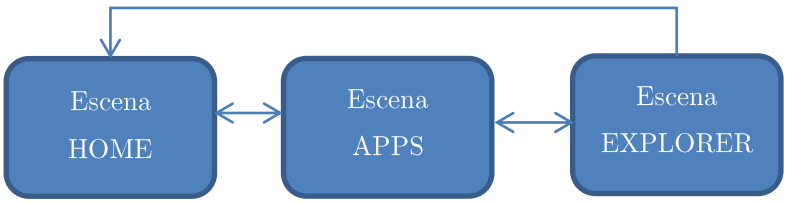

Figura 9 - Flujo del sistema operativo

El proyecto contiene las siguientes escenas:

- Home : Es una escena sencilla cuyo propósito es, en futuros desarrollos, contener *widgets* o accesos directos a aplicaciones o URLs.
- Apps : En esta escena se listan las aplicaciones disponibles en el sistema operativo, permitiendo abrirlas y hacer *scroll* en ellas.

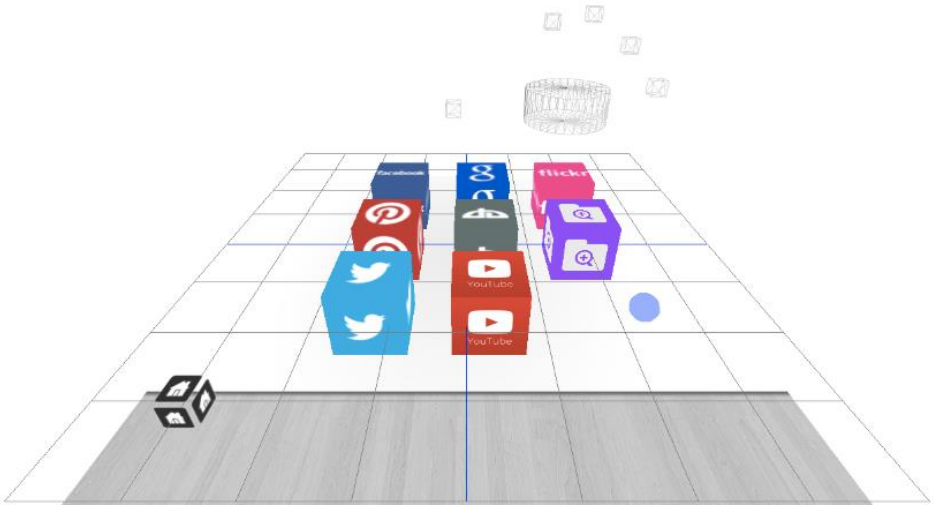

Figura 10 - Escena de listado de *apps*

- Explorer: El *explorer* es el explorador de ficheros, permite realizar acciones de borrado y apertura, otras operaciones como copiar, pegar y compartir se muestran pero no están disponibles.

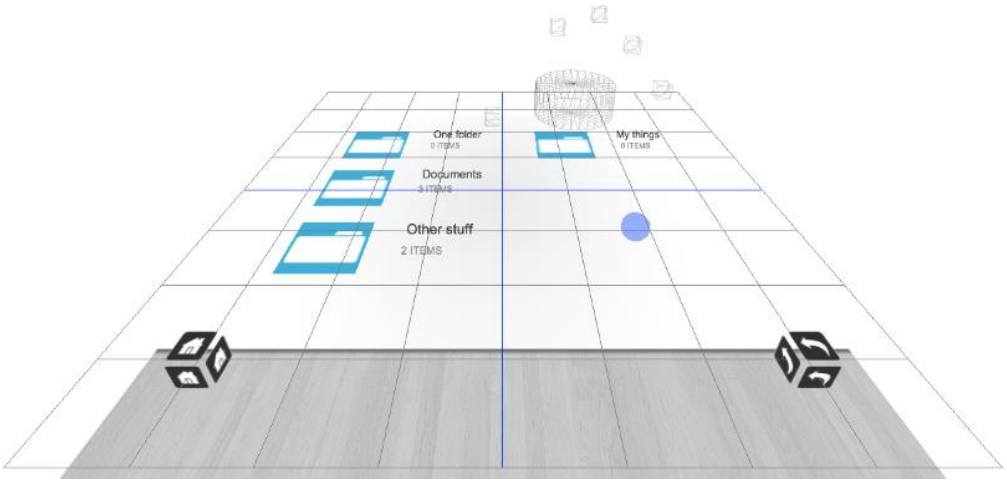

Figura 11 - Escena del explorador de ficheros

6.3.4 Vistas

OSAppView

Esta vista extiende de Interactive3DView y define en su interior el comando del sistema operativo a ejecutar y los parámetros necesarios.

OSFileView

Esta vista extiende igualmente de Interactive3DView. Al igual que OSAppView, se definen todas las operaciones contextuales propias, en este caso, de un fichero. Determina, en función de si es carpeta o fichero, que tipo de icono se va a mostrar teniendo en cuenta su extensión y finalmente añade a la vista base textos informativos como el nombre, tamaño o número de ficheros que contiene.

Figura 12 – Apariencia de OSFileView

6.4 Herramientas utilizadas

Para llevar a cabo el proyecto se ha partido de HTML y JavaScript como base, haciendo uso de la API de LEAP Motion v2. Durante todo el desarrollo se han utilizado librerías de terceros que se han ido importando desde la carpeta **3rdparty** del proyecto. Estas librerías son:

Tween.js	Librería para la creación de interpolaciones y animaciones.
Three.js	Librería para la creación y manipulación de entornos tridimensionales.

6.5 Localización del proyecto

El proyecto se encuentra disponible bajo la licencia MIT en el repositorio GIT con url :

https://jcandela@bitbucket.org/jcandela/tlp-leap-motion.git

Capítulo 7. Conclusiones y Trabajos Futuros

Durante este proyecto se ha tratado de encontrar nuevas formas de interacción que permitan al usuario final interactuar de la forma más intuitiva posible con los sistemas informáticos.

Como punto de partida, nuestro proyecto pretende suponer una base para el desarrollo de interfaces para dispositivos naturales en la web. La API planteada, con los gestos de interacción reconocidos, permiten una navegación fluida que aumentará la capacidad de reacción del sistema y permitirá al usuario realizar las tareas objetivo sin ningún tipo de interfaces adicionales.

A continuación se listan las conclusiones obtenidas del proyecto:

- Los dispositivos de interacción natural son cada vez más corrientes y asequibles para el usuario final y la tendencia es a estar cada vez más presentes en los sistemas informáticos.
- La usabilidad de las interfaces gráficas que hace uso de interfaces naturales dependen en gran medida de la implementación desarrollada.
- Es relativamente sencillo elaborar una aplicación que haga uso de los datos ofrecidos por los dispositivos de reconocimiento gestual y la dificultad radica en la elaboración de una aplicación que reaccione correctamente.
- La estandarización de los gestos de interacción natural se llevará a cargo de la empresa u organización que logre implementarlos de la forma más efectiva (e.g.: iPhone y las pantallas táctiles).

Como trabajos futuros planteamos la posibilidad de extender la API con nuevas vistas y elementos de interfaz para permitir crear interfaces ricas y variadas pensadas para el dispositivo objetivo. A su vez, la adición de nuevos gestos destinados a complementar la interacción así como la posibilidad de gestionar extensiones para los diseños de interfaz.

Otra posibilidad podría ser implementar la API para otros dispositivos que permitan un mejor reconocimiento

Para concluir, se puede decir que el sistema construido ha explotado un porcentaje mínimo de la capacidad del dispositivo y las posibilidades de ampliación son infinitas.

Capítulo 8. Summary and Conclusions

The intention throughout this project has been to look for innovative methods to help facilitate Human-Computer interaction by way of gesture recognition.

Therefore, an evaluation of the current devices that implement gesture recognition was carried out, followed by a selection of the most appropriate device to be used in the elaboration of a project to improve usability for graphical interfaces that make use of these gesture recognition devices.

Once the device selection was made, various alternatives for development were considered where the possibility to introduce some improvements and innovations could exist in development.

The chosen proposal was that of a web graphical interface system that allows for web browsing through the use of gesture recognition devices running on a test operating system that will be used to evaluate the capabilities and usability of the system.

Once developed the following conclusions were reached:

- Natural User Interface devices are increasingly becoming more common and affordable for the end user and the tendency is for them to become even more present in information systems.
- The usability of graphical interfaces that make use of Natural User Interfaces depends heavily on the implementation that was developed for them.
- It is relatively simple to create an application that makes use of the data collected by gesture recognition. The real difficulty lies in the creation of an application that will react in the appropriate manner.

- The standardization of natural interface gestures will be carried out by the organization or business that is able to implement them in the most effective manner (e.g. iPhone and touch screens).

In conclusion, it can be said that the system that was built has only exploited a minimal percentage of the capacity for the device and the possibilities for expansion are infinite.

Bibliografía

[1] Kinect, Wikipedia <http://en.wikipedia.org/wiki/Kinect>

[2] LEAP Motion, Wikipedia
<http://en.wikipedia.org/wiki/Leap_Motion>

[3] PlayStation Eye, Wikipedia
<http://en.wikipedia.org/wiki/PlayStation_Eye>

[4] WiiMote, Wikipedia <http://en.wikipedia.org/wiki/Wii_Remote>

[5] PlayStation Move, Wikipedia
<http://en.wikipedia.org/wiki/PlayStation_Move>

[6] Myo Gesture Control Armband, Thalmic Labs
<https://www.thalmic.com/en/myo/>

[7] LEAP Motion, Leap Motion, Inc <https://www.leapmotion.com/>

[8] Project Tango, Google ATAP
<https://www.google.com/atap/projecttango/>

[9] DepthSense Cameras, Softkinect <http://www.softkinetic.com/en-us/products/depthsensecameras.aspx>

[10] SoftKinect DS325 Datasheet, SoftKinect
<http://www.softkinetic.com/Portals/0/Documents/PDF/WEB_20130527_SK_DS325_Datasheet_V4.0.pdf>

[11] LEAP Motion Javascript Documentation, Leap Motion, Inc.
<https://developer.leapmotion.com/documentation/javascript/index.html>