INTRODUCCIÓN A LAS INTERFACES NATURALES GESTUALES CON DISPOSITIVOS DE CAPTURA ÓPTICA

Julia Saenz julisaenz99@gmail.com

Alejo Schön aleeschon@gmail.com

Luciano Nahuel Espinosa nachuespinosa@gmail.com

Laboratorio emmeLab
Facultad de Artes
Universidad Nacional de La Plata
Argentina

Resumen

Las interfaces naturales gestuales son aquel tipo de interfaz de usuario en las que se interactúa con un sistema, aplicación, etcétera, sin utilizar métodos de entrada convencionales (como podrían ser un mouse, un teclado alfanumérico, un panel táctil, o joystick, etcétera) y en su lugar, se hace uso de movimientos gestuales del cuerpo o de alguna de sus partes, gestos faciales o sonidos. Pertenecen al grupo de las Interfaces Naturales, las cuales se basan en que la interacción con los dispositivos suceda de la misma manera que como lo hacemos con otros seres humanos, como por ejemplo a través del habla y los gestos.

Este artículo hace una recopilación de dispositivos, métodos de captura y gestos comúnmente utilizados en el desarrollo de interfaces naturales gestuales, en función de familiarizar al lector con algunos conceptos básicos, enfocándonos en el funcionamiento, ventajas y desventajas de cada elemento.

HCI

interfaces gestuales

detección del cuerpo

1. Introducción

Las interfaces naturales gestuales son aquel tipo de interfaz de usuario en las que se interactúa con un sistema, aplicación, etcétera, sin utilizar métodos de entrada convencionales (como podrían ser un mouse, un teclado alfanumérico, un panel táctil, o joystick, etcétera) y en su lugar, se hace uso de movimientos gestuales del cuerpo o de alguna de sus partes, gestos faciales o sonidos. Pertenecen al grupo de las Interfaces Naturales, las cuales se basan en que la interacción con los dispositivos suceda de la misma manera que como lo hacemos con otros seres humanos, como por ejemplo a través del habla y los gestos. Estas interacciones humanas, son de fácil comprensión para los seres humanos, ya que pertenecen a la más extrema cotidianidad, pero sin embargo, comprenden un complejo desafío de interpretación para una computadora, Lo interesante de las Interfaces Naturales de Usuario se basa en la capacidad que le da a las máquinas de entender mejor al mundo en el que están inmersas. [1]

No es un tipo de interfaz nueva y, como tal, existe una gran variedad de herramientas de software y dispositivos de detección, cada uno con sus propias ventajas y desventajas. La elección correcta de estos elementos se verá afectada por numerosos factores como el presupuesto, el diseño de la interfaz, el público al que va dirigido o el lugar en el que va a ser ubicada la interfaz.

Este artículo hace una recopilación de dispositivos, métodos de captura y gestos comúnmente utilizados en el desarrollo de interfaces naturales gestuales, en función de familiarizar al lector con algunos conceptos básicos, enfocándonos en el funcionamiento, ventajas y desventajas de cada elemento.

Durante las siguientes secciones del artículo analizamos diversos criterios de selección para abordar diferentes tecnologías de detección y sensado usadas en el desarrollo de este tipo de interfaces. A su vez, exploramos las distintas tipologías de gestos realizados por los humanos

Figura 1

Imagen promocional de Kinect Sports, un videojuego desarrollado para Kinect que funciona con detección de gestos y movimiento.

y cómo estos pueden relacionarse intrínsecamente con la interfaz en la que se utilizan y la tecnología que se usa para crearla. Finalmente, exploramos distintos softwares que se pueden utilizar para construir una interfaz natural gestual, considerando sus ventajas y desventajas.

2. Criterios de selección

Para este artículo, tendremos en cuenta solamente a aquellas interfaces naturales que realizan su detección a través de dispositivos ópticos y donde el medio de interacción natural es completamente gestual. Sin embargo, no se abarcan todas las opciones disponibles de captura óptica ni de gestos sino que se realizó un recorte pensado para proyectos de nivel inicial o de bajo presupuesto, apuntando a lectores con escaso conocimiento del tema y al desarrollo de una interfaz poco compleja. Esto es debido a las condiciones de producción presentes en nuestra región y la disponibilidad y precio de los dispositivos de captura. Sin embargo, esto no modifica el potencial y variabilidad que puede llegar a poseer una interfaz de esta tipología y construido con las tecnologías mencionadas.

Dado que el enfoque es en dispositivos de captura óptica, el recorte de los gestos analizados está delimitado por aquellos que puedan ser captados por una cámara.

Finalmente, para la selección de programas de detección, se tuvieron en cuenta ciertos requisitos: que el programa fuese de código abierto y que fuese compatible con Processing, ya sea mediante una conexión por OSC o una librería del programa.

3. Dispositivos de detección

Un dispositivo de detección es aquel periférico a través del cual se detecta algún tipo de acción del usuario. Los factores de mayor influencia en su elección son: el presupuesto disponible, el espacio donde quiera realizarse la detección (exterior o interior, con niveles de luz bajo o altos) y el o los gestos a detectar.

A continuación se detallan los tipos de detección óptica más utilizados, su funcionamiento y características y un cuadro comparativo de algunos dispositivos concretos.

3.1. Cámara RGB

Una cámara es un dispositivo utilizado para capturar imágenes. Aplicado al contexto de una interfaz natural gestual, una cámara es el dispositivo utilizado para sensar el espacio mediante la captura de imágenes y el procesamiento de las mismas. No todas las cámaras son iguales, y no todas pueden obtener los mismos datos de una captura realizada. Uno de los datos más comunes de registrar es el color, y si una cámara puede tomar esa información depende del tipo de sensor óptico que tenga. Es aquí, donde el modelo de color RGB cobra importancia. El RGB es un modelo de color basado en la síntesis aditiva, con el que es posible

representar un color mediante la mezcla por adición de los tres colores de luz primarios: rojo verde y azul. Las cámaras RGB, al igual que los ojos humanos, son capaces de descomponer la luz en los tres canales de informacion de color que la componen.

La ventaja de estás cámaras es que pueden combinarse con los tipos de cámaras que veremos a continuación y que, si solo se requiere información de color, son las más accesibles en cuanto a precio. La desventaja es que depende enormemente de la calidad de luz del espacio y no tiene ningún tipo de información de la escena además del color.

Es importante tener en cuenta a la hora de realizar una interfaz natural gestual, si la información de color es de vital importancia para el desarrollo de la misma. Como veremos al explorar distintos dispositivos, algunos no son capaces de detectar colores, pero poseen otras prestaciones que compensan esta falencia.[2]

3.2. Cámaras estereoscópicas

De la misma forma que nuestros ojos, una cámara estereoscópica utiliza dos cámaras separadas por una distancia conocida para calcular la profundidad de una escena. Dado un punto en el espacio, la separación de las cámaras lleva a una diferencia en la posición de ese punto en las dos imágenes, y mediante el cálculo de esa disparidad se puede discernir su distancia con respecto a la cámara, creando un mapa de profundidad de la escena.

Figura 2

Esquema explicativo del funcionamiento de un sensor RGB

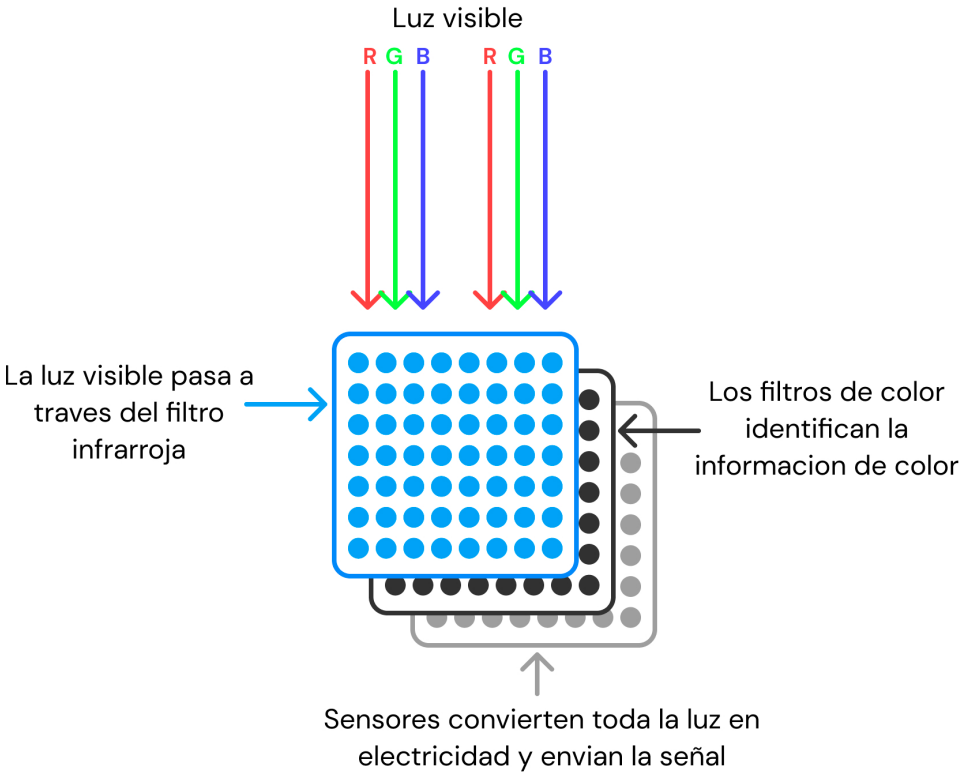


Figura 3

Ejemplo de fotografía tomada con una cámara estereoscópica.

La ventajas de este tipo de cámara son que puede obtenerse información de la profundidad de la escena a bajo costo, pudiendo incluso armarse una cámara estereoscópica básica utilizando dos cámaras comunes; el rango de detección puede modificarse fácilmente aumentando o disminuyendo la distancia entre lentes; y puede utilizarse para diferenciar gestos o partes del cuerpo más específicas, dependiendo del software con el que se la complemente. Sin embargo, la precisión de la detección depende del nivel de luz y la textura de la escena: es poco útil en espacios poco iluminados o, por ejemplo, frente a una pared vacía.

3.3. Cámaras Infrarrojas

Un sensor infrarrojo (IR) detecta y mide la radiación infrarroja en su campo de visión[1] la cual, al tener un largo de onda mayor que el de la luz visible, no es observable para el ojo humano. Todo objeto o sujeto que tenga temperatura, emite radiación infrarroja.

Hay dos tipos de sensores infrarrojos: activos y pasivos. Los sensores activos cuentan de un diodo emisor de luz (LED) y un receptor; cuando un objeto se acerca al sensor, la luz del emisor refleja ese objeto y el emisor detecta el reflejo con el cual puede calcular la forma y distancia del objeto. Un ejemplo de un sensor activo es el control remoto de cualquier televisión: el control tiene un emisor que transforma el toque de un botón en un haz de luz infrarroja que es detectada por el receptor de la televisión. Los sensores pasivos, en cambio, solo cuentan con el receptor, por lo que solo detectan las radiaciones infrarrojas que ya tiene cada objeto. La medición de cada sensor puede variar según el largo de onda que capta y el tiempo de respuesta.

El término cámara infrarroja incluye cualquier cámara que utilice un sensor infrarrojo para detectar una escena; la cual puede ser solo de profundidad[2], o de tipo RGB-D. [3]Las ventajas de estos dispositivos son que, al igual que las cámaras estereoscópicas, pueden detectar gestos específicos; y no necesitan luz para producir una imagen definida, aunque son especialmente sensibles a materiales transparentes, semi-transparentes o reflectivos como vidrios, espejos o metales.

Algunos de los dispositivos más utilizados en interfaces naturales gestuales son cámaras infrarrojas, como la Kinect [3], y como tal suelen tener programas diseñados para trabajar específicamente con ellos. Sin embargo este tipo de cámaras suelen ser las más costosas y más difíciles de conseguir.

Figura 4

Una Microsoft Kinect 1.0, que contiene un sensor e emisor de luz infrarroja

Los dos métodos más usados para este tipo de detección son:

3.3.1 Tiempo de Vuelo (ToF)

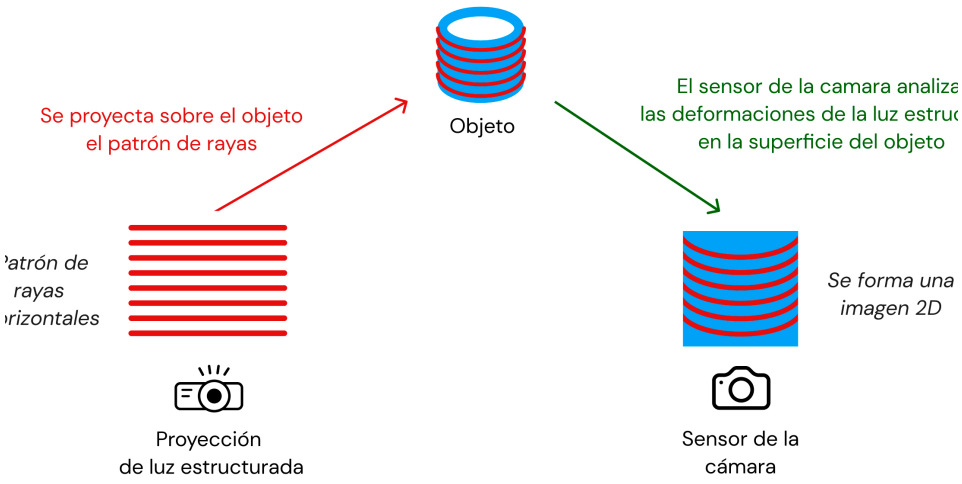
El tiempo de vuelo (ToF) es un método para calcular la distancia de objetos haciendo uso de un haz de luz infrarroja. Se calcula el tiempo transcurrido entre la emisión del haz de luz, su reflejo en un objeto y su retorno al sensor; mientras más tiempo haya entre la emisión y la recepción, más lejos se encuentra el objeto. De esta forma se crea un arreglo de puntos medidos individualmente, que son luego reconstruidos en forma de una imagen bidimensional o representaciones tridimensionales básicas de la escena. En comparación con la proyección de luz estructurada, este método es más fácil de calibrar y tiene mayor velocidad de procesamiento, aunque la cámara suele tener menor resolución; por estas razones en los últimos años compañías como Apple y Kinect pasaron a utilizar medición por proyección de luz estructurada a tiempo de vuelo.

3.3.2. Proyección de Luz Estructurada

La detección a través de la proyección de luz estructurada funciona de forma similar a la cámara estereoscópica, solo que en lugar de dos cámaras, utiliza una cámara y un proyector láser. Funciona mediante un proyector infrarrojo que barre una línea de luz sobre la escena, aunque también puede proyectarse un patrón de puntos o líneas; la cámara

Figura 5

Esquema explicativo del funcionamiento del sistema de captura de ToF (Tiempo de Vuelo) utilizando una Microsoft Kinect 1.0



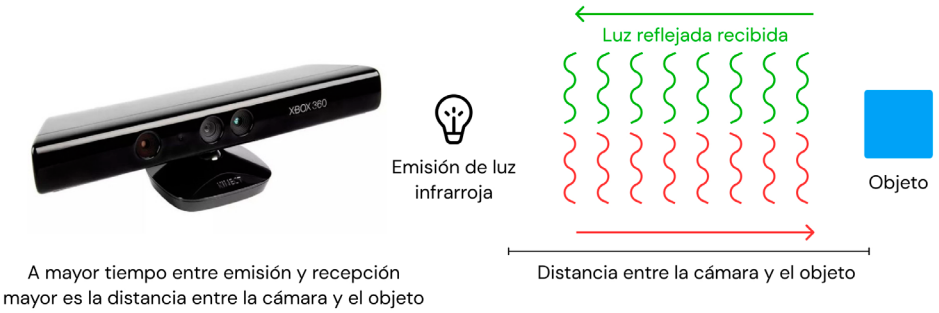


Figura 6

Esquema explicativo del funcionamiento del sistema de captura de Luz Estructurada

recibe la imagen con la luz proyectada y a través de las diferencias de esta imagen iluminada con el haz proyectado se reconstruye la superficie tridimensional. Este método tiene mejor definición a poca distancia, pero es especialmente sensible al exceso de luz ambiente.

3.4. Cuadro comparativo

	ZED	Kinect 1.0	Xtion Pro	Kinect 2.0	LeapMotion	Realsense	Azure
Tipo de dispositivo	Cámara estereoscópica	Cámara Infrarroja	Cámara Infrarroja	Cámara Infrarroja	Cámara Infrarroja	Cámara estereoscópica e infrarroja	Cámara Infrarroja
Modo de detección	Medición estereoscópica	Proyección de luz estructurada	Proyección de luz estructurada	Tiempo de Vuelo	Tiempo de Vuelo	Medición estereoscópica	Tiempo de Vuelo
Año	2014	2010	2011	2014	2013	2015	2020
Activo o pasivo	Pasivo	Activo	Activo	Activo	Activo	Activo	Activo
Información de color	Si	Si	Si	Si	No	Si	Si
FOV	90° x 60° x 100°	57° x 43°	58° x 45° x 70°	70° x 60°	180°	**Sensores BN:** 86° x 57° (±3°) **Sensor RGB:** 69.4° x 42.5° x 77° (± 3°)	90° x 74.3°
Rango mínimo	0.3 m	0.8 m	0.8 m	0.5 m	60 cm	0.4 m	0.5 m
Rango máximo	25 m	4 m	3.5 m	4.5 m	80 cm	10 m	5,46 m
Ambientes	Interior o Exterior	Interior	Interior	Interior	Interior o Exterior	Interior o Exterior	Interior
Conexión	USB 5V	USB 2.0	USB 2.0	USB 3.0	USB 2.0	USB-C* 3.1 Gen 1	USB 3.0

Figura 7

Cuadro comparativo entre diversos tipos de dispositivos ópticos utilizados para la detección de cuerpos y objetos en el espacio

4. Gestos

El segundo elemento a tener en cuenta es el gesto o acción de entrada que se requiere del usuario. Las cuestiones más importantes en la elección de gestos son: el espacio en el que va a realizarse, el tipo de público al que está orientada la interfaz y la cantidad de opciones diferentes de interacción que tendrá el usuario.

Estos gestos pueden ser complejos ya que culturalmente pueden tener connotaciones muy diferentes. Sin embargo, para facilitar su análisis, podemos clasificarlos en tres distinciones que también pueden utilizarse como categorías [4]:

- **Estático o dinámico:**
    - En función de sus características temporales, los gestos pueden

clasificarse como estáticos o dinámicos.

- Los gestos estáticos son aquellos en los que solo se observa la posición estática de una mano.
- Los gestos dinámicos son aquellos en los que una mano se mueve entre una serie de posiciones para formar un gesto completo.

- **Comunicativo o manipulativo:**
  - En función de su uso y del contexto en el que se utilizan, los gestos también pueden clasificarse como comunicativos o manipulativos
  - El primer grupo de gestos, los gestos comunicativos, son aquellos que, aunque siguen teniendo una finalidad comunicativa y se utilizan con mayor frecuencia junto con el habla, pueden ser independientes del habla y no la necesitan para transmitir un significado. Tienen una traducción directa en palabras y pueden reemplazarlas. Se utilizan deliberadamente para enviar un mensaje concreto y tienen un significado ampliamente aceptado, aunque pueda ser específico de un grupo, clase o cultura. Algunos ejemplos son “pulgares arriba” para indicar aprobación agitar la mano como saludo o “frotar el dedo índice y el pulgar para referirse al dinero” .
  - Por otro lado, los gestos manipulativos se utilizan para comunicar la posición espacial de los objetos, o las formas en que se manipulan y tienen un fuerte vínculo con el habla. Tan estrecho es este vínculo que no funcionan como elementos comunicativos separados.

- **Predefinidos o de forma libre:**
  - En función de los niveles de instrucción que se dan para guiar la ejecución de los gestos, éstos pueden clasificarse como gestos predefinidos o de forma libre.
  - Los gestos prescritos son aquellos en los que se define un diccionario de gestos antes de ser utilizados. Los usuarios de una aplicación tienen que aprender estos gestos, y la realización de un gesto predefinido desencadena una acción predefinida. Los gestos prescritos pueden aumentar la carga cognitiva, sus tasas de aprendizaje dependen de las habilidades cognitivas de los usuarios, y su uso obliga a los usuarios a aprender y utilizar gestos que quizás no elegirían ellos mismos.
  - Los gestos de forma libre no tienen restricciones y no suelen desencadenar acciones específicas y uniformes predefinidas. En el contexto de las interfaces interactivas, normalmente se copian en el sistema para el que se utilizan las interfaces, y suelen utilizarse para formar superficies, o para mover objetos en un espacio virtual. Esto significa que no comunican los significados simbólicos o metafóricos que pueden transmitir los gestos prescritos. Por lo tanto, los gestos de forma libre, en contra de la implicación de no ser restrictivos que su nombre infiere, tienen

una amplitud de aplicación limitada en su forma actual.

A diferencia de las interfaces táctiles, en la que la interacción es casi intuitiva debido a que los botones suelen utilizar iconografía conocida (la cruz para cancelar, por ejemplo) o son autoexplicativos (el botón para cancelar dice “Cancelar”), las interacciones naturales son mucho menos directas, necesitando que el usuario realice un trabajo de aprendizaje rápido para poder interactuar cómodamente, sin una carga cognitiva muy elevada. Por esta razón, generalmente este tipo de interfaces no pide a los usuarios que recuerden más de 3 o 4 entradas diferentes.

4.1. Gestos faciales

Los humanos pueden realizar una infinidad de gestos con el rostro, proporcionando una amplia gama de entradas posibles, ya sea cerrar o abrir los ojos, sonreír, abrir la boca o alguna combinación más complicada. Sin embargo, estos gestos pueden ser muy complejos o sutiles para una detección confiable.

El uso más conocido de este tipo de entrada es la cámara que saca foto cuando una o más personas están sonriendo, pero no es ampliamente utilizada ya que necesita un dispositivo de entrada con buena calidad de imagen para asegurar la correcta detección de los gestos. Este tipo de entrada puede detectarse tanto con cámaras estereoscópicas como infrarrojas.

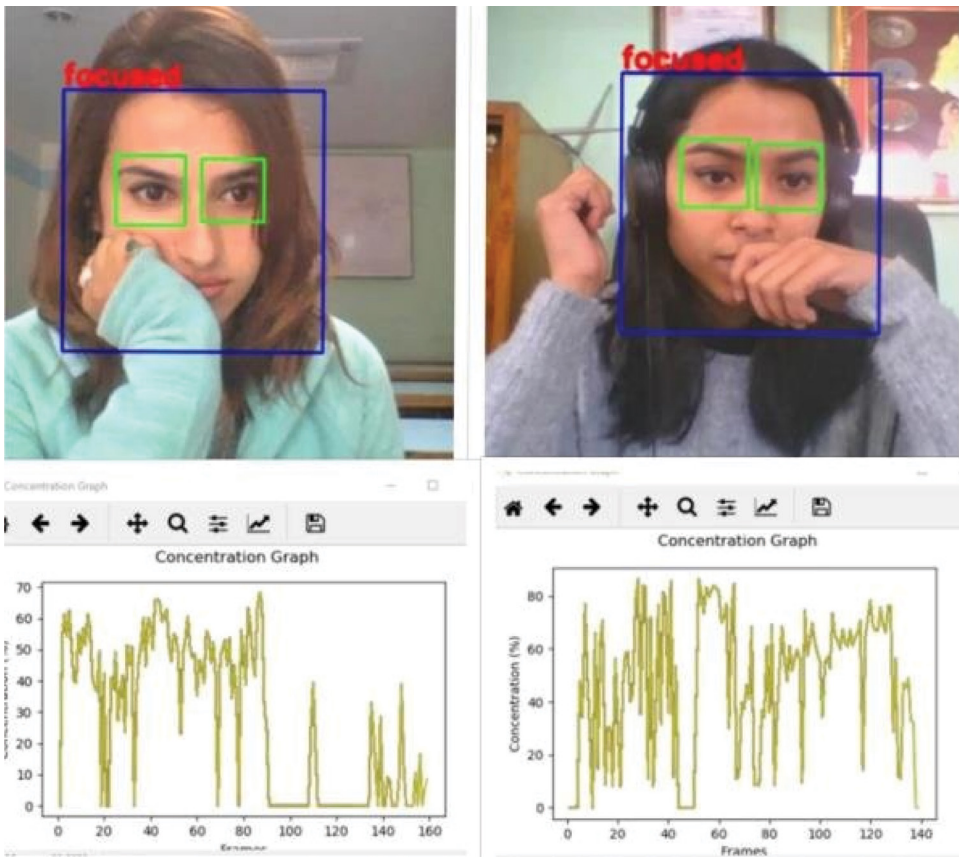


Figura 7

Reconocimiento de gestos faciales a través de Computer Vision

(Student Engagement Detection Using Emotion Analysis,

Eye Tracking and Head Movement with Machine Learning)

También, existen proyectos que utilizan la detección de estos gestos y el seguimiento del rostro para ofrecer mayor accesibilidad a personas con movilidad reducida. Por ejemplo, existe el caso de EVA Facial Mouse, una aplicación móvil que por medio del seguimiento del rostro del usuario captado a través de la cámara frontal de un celular permite controlar un puntero en pantalla (a modo de mouse) que permite el acceso a la mayor parte de elementos de la interfaz.

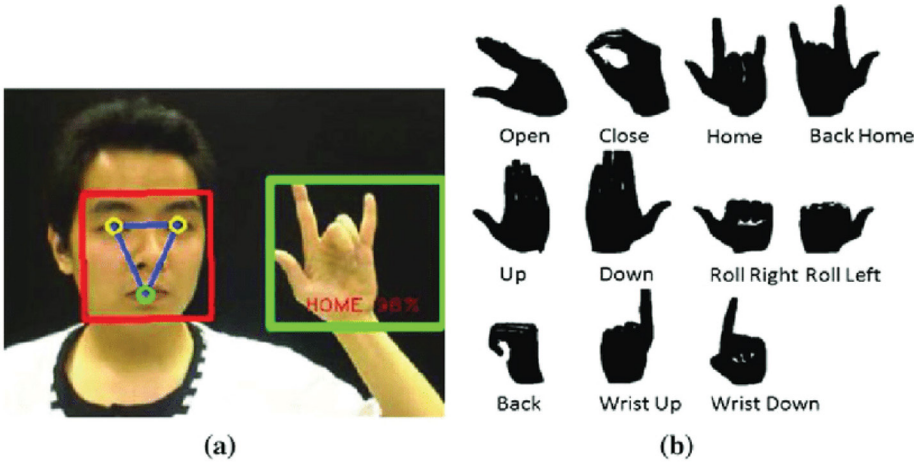
4.2. Gestos de mano

La mano es una de las partes del cuerpo que más se suele utilizar como método de entrada [4], siendo estas las partes del cuerpo con las que interactuamos normalmente. También tienen una gran variabilidad en cuanto a posiciones: palma abierta, puño cerrado, pulgar hacia arriba o hacia abajo, señalar, etc; lo cual permite poder utilizar en una interfaz una amplia variedad de entradas realizadas con la misma extremidad.

Para la detección de este tipo de gesto no es necesaria una cámara con demasiada calidad de imagen, ya que la mayoría de los programas requieren solo la forma general de la mano para detectar qué gesto se está realizando.

Sin embargo, es necesario que si se están utilizando múltiples gestos, la confianza del programa en la detección sea alta, ya que confundir un gesto con otro significaría una interacción errónea, y también es necesario que el usuario separe la mano del cuerpo al momento de realizar el gesto, de forma que no confunda la mano con el resto del cuerpo.

4.3. Seguimiento de mano



La otra forma de utilizar la mano como entrada es utilizarla como una especie de puntero, es decir, hacer un seguimiento de la posición en la

Figura 10

"Melodía Escondida" de Estudio Biopus

que se encuentra la mano en la escena, sin importar la forma que esta tenga, para corresponder esa posición con una ubicación en la pantalla. Este tipo de interacción es más intuitiva, ya que funciona de forma similar a un mouse; no requiere de la detección de una gesto específico, por lo que el programa puede permitirse una confianza un poco menor en la detección; y es también más fácil de aprender para el usuario, ya que solo requiere movimiento. La forma de selección de este gesto no será entonces mediante un gesto específico, sino por el mantenimiento de la mano en un mismo lugar por un espacio determinado de tiempo. Cualquier dispositivo es apto para detectar este tipo de gesto.

Es importante cuando se usa este tipo de interacción que el usuario tenga un feedback constante de la ubicación de la mano en relación a la pantalla y de cuándo la interfaz está tomando su falta de movimiento como confirmación de una acción.

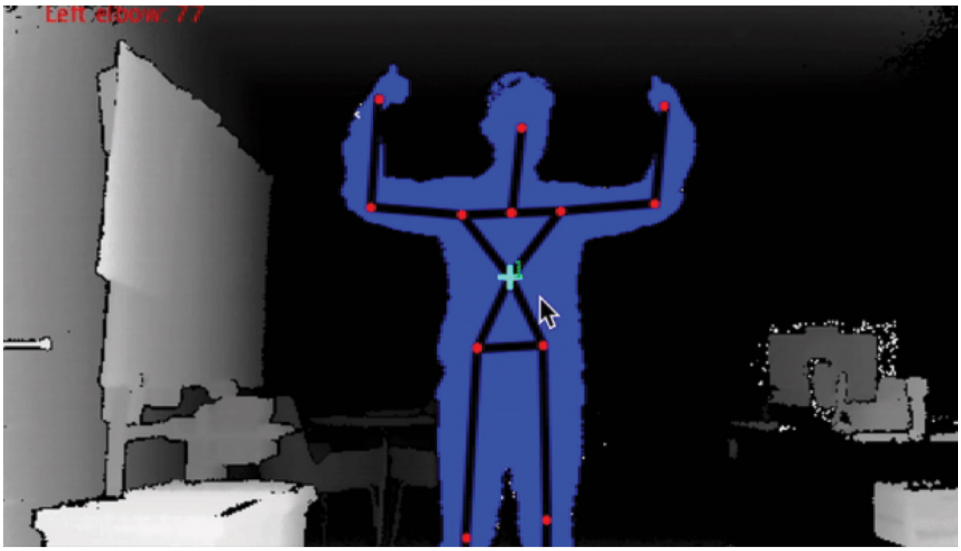
4.4. Cuerpo entero

También es posible utilizar al cuerpo entero como método de interacción con las interfaces. Esto nos permite aprovechar diversos parámetros que no aparecen en los demás gestos, como pueden ser la distancia del cuerpo al sensor, la altura (midiendo si el usuario está en una posición agachada o parada), la distancia entre extremidades, etc. Su principal desventaja se presenta en la distancia que debe haber entre el usuario y sensor para captar el cuerpo entero, lo cual obliga a un campo de visión muy amplio y a un espacio grande para situar a la interfaz.

Hay dispositivos como Kinect, que tienen funcionalidades específicas para la detección de esqueletos, pero también esto puede hacerse mediante programas específicos. De lo contrario, también podría usarse una simple detección de cuerpo, sin esqueleto, lo cual permitiría solamente detectar la posición y la distancia del usuario.

Figura 11

Detección de cuerpo entero a traves de SimpleOpenNI



5. Software

El software es la herramienta que analiza la información que entra a través del dispositivo de detección, reconoce en ella los gestos definidos y ejecuta las respuestas de la interfaz correspondientes. Al igual que los dispositivos de captura, cada herramienta tiene situaciones en las cuales se desenvuelve mejor y peor. La elección no dependerá solo de esto, sino también del tipo de sistema operativo en el que se vaya a ejecutar la interfaz, el programa o programas con los que vaya a conectarse y, como mencionamos anteriormente, del dispositivo de detección y el gesto a detectar.

A grandes rasgos, el funcionamiento de cada programa o software es similar. Explicado de forma sencilla, existen dos eventos que suceden de forma simultánea: primero, el software detecta la parte del cuerpo que ejecuta el gesto. La manera para llevar a cabo la detección puede variar de programa a programa. Seguidamente, una vez identificada la parte del cuerpo que ejecuta el gesto, se realiza un seguimiento de la misma para entender los cambios que sufre en el tiempo.

Durante los siguientes párrafos exploraremos diversos softwares, un poco de su historia y diversos casos de aplicación en los cuales pueden ser más o menos útiles.

5.1. OpenCV

OpenCV (Open Source Computer Vision Library) [5] es una librería de visión computacional[4] y machine learning[5] orientada especialmente a aplicaciones de visión en tiempo real. Cuenta con más de 2500 algoritmos optimizados que pueden utilizarse para la detección de caras, reconocimiento de gestos, acciones o movimientos, extracción de modelos 3D a partir de imágenes, entre otras cosas y es una de las librerías más amplias actualmente. Está escrita nativamente en C++ pero tiene contenedores para Python, Java, MatLab, Octave y Javascript. Las principales ventajas de esta librería son que tiene soporte para Android, Linux, MacOS y Windows; y el tener una documentación extensa y

organizada.

5.1.1. OpenCV for Processing

OpenCV for Processing [6] es la librería de OpenCV desarrollada específicamente para ser utilizada con el entorno de programación Processing. Está basada en la versión de OpenCV para Java y tiene soporte para Linux, MacOS y Windows, pero no Android. Las funcionalidades de esta librería son mucho más reducidas que la versión original en C++: puede reconocer diferencias entre dos imágenes para eliminar fondos o detectar nuevos objetos, encontrar líneas, contornos o bordes de una imágen, trabajar con los canales de color de una imágen o detectar algún objeto o parte del cuerpo específica mediante las cascadas de Haar[6] que vienen con la librería o cascadas importadas.

Puede utilizarse tanto con cámaras RGB, como estereoscópicas e infrarrojas.

5.2. OpenNI

OpenNI (Open Natural Interaction) [7] es un software desarrollado por una organización del mismo nombre que consiste en detección de movimiento, principalmente detección de profundidad y esqueleto del cuerpo humano. A partir de la detección del esqueleto se puede obtener la posición de una parte del cuerpo específico, como de las manos, la cabeza, etc. Actualmente el software está siendo desarrollado en una variedad de proyectos de código abierto dentro del mundo académico y la comunidad de aficionados. Los sensores de movimiento compatibles con este software son: Microsoft Kinect v1 (XBOX 360), Microsoft Kinect v2 (XBOX ONE) y Asus Wavi Xtion (PC).

5.2.1. SimpleOpenNI

SimpleOpenNI [8] es la librería de OpenNI para ser utilizada con el entorno de programación Processing. A diferencia de OpenNI, esta librería funciona únicamente con dispositivos de captura de movimiento Microsoft Kinect, centrándose en detección de esqueleto, lo cual lo hace óptimo para detección de cuerpos o seguimiento de mano. Aunque tiene documentación disponible, este programa ya no está siendo actualizado.

5.2.2. FingerTracker

Dos años después del lanzamiento de OpenNI, Greg Borenstein desarrolló un software llamado FingerTracker [9] que consiste en tomar las propiedades de OpenNI y agregar el reconocimiento de dedos de las manos, cosa que OpenNI no es capaz de detectar.

5.3. Tensor Flow

TensorFlow [10] es una plataforma creada por Google para el desarrollo de proyectos con machine learning. Tiene un ecosistema completo y

flexible de herramientas, librerías y recursos de la comunidad que permite a los desarrolladores construir y desplegar fácilmente aplicaciones potenciadas por esta tecnología.

5.3.1. PoseNet

PoseNet [11] es un modelo de visión computacional que usa una red neuronal convolucional para determinar la pose de una persona en una imagen o video, estimando dónde están las articulaciones o puntos claves del cuerpo. A diferencia de otros sistemas de detección, que requieren hardware y/o cámaras especializadas, PoseNet, ejecutándose a través de TensorFlow.js o Runway, permite conectar cualquier tipo de cámara con el modelo.

Puede utilizarse para estimar una sola pose o múltiples poses, es decir, hay una versión del algoritmo que puede detectar sólo una persona en una imagen/vídeo y otra que puede detectar múltiples personas. Las articulaciones clave que detecta son: hombro, codo, muñeca, cadera, rodilla y tobillo, además de nariz, ojo y oreja. Para cada articulación u característica guarda la posición derecha e izquierda como dos objetos distintos (a excepción de la nariz). La forma en la que funciona este modelo es primero reconociendo un sector de la imagen en el que crea que hay una persona y luego intentar definir donde están cada uno de los puntos mencionados anteriormente, guardando un número entre 0 y 1 relativo a la confianza que tiene en cada detección, siendo 1 el más alto. Al siempre intentar definir todas las articulaciones, funciona mejor cuando la cámara está ubicada a suficiente distancia del usuario para tener en el campo de visión el cuerpo entero.

Puede identificar características de la cara, pero no gestos particulares y en lugar de detectar la posición de la mano puede detectar la de la muñeca, así que no es posible utilizarlo para reconocimiento de gestos o seguimiento.

Otra de las particularidades de esta herramienta es que para ser conectada con Processing se necesita una conexión a Internet y obligatoriamente deben comunicarse los programas mediante algún protocolo de comunicación, como OSC o WebSockets.

5.3.2 PoseOSC

Una forma de utilizar PosNet sin necesidad de una conexión a Internet es mediante la aplicación PoseOSC [12]. Esta cuenta del modelo y permite desde una interfaz gráfica controlar algunos de los parámetros del mismo, tanto como el formato y el puerto al que se quiere enviar la data de captura por OSC. Para modificar más detalladamente los parámetros, también se puede modificar el archivo json de configuración [13].

5.4. TSPS

TSPS [14] (Toolkit for Sensing People in Spaces) es un software desarrollado en OpenFrameworks para crear aplicaciones interactivas basadas en la interacción natural del usuario; envuelve los algoritmos de

visión por computadora en una interfaz simple y se enfoca en la creación rápida de prototipos y talleres educativos.

El método de captura funciona mediante reconocimientos de contornos y reconocimiento facial, proporcionando también configuraciones para calibrar la cámara y hacer más precisa la captura. Para vincularse con otros lenguajes de programación o softwares, se necesita usar algún sistema como OSC, TUIO, Web Sockets, Spacebrew o TCP. Es compatible con cualquier tipo de cámara.

5.4.1. TSPS para Processing

TSPS cuenta con su propia librería para el entorno de programación Processing, pero para poder realizar tracking se requiere tener abierta la interfaz y tener importada alguna de las librerías de comunicación mencionadas anteriormente. La ventaja que tiene esta librería es que puede obtener datos específicos de la interfaz, como el tiempo que la persona está siendo capturada por el sistema, las posiciones del rostro, o la posición del centro del contorno de la persona detectada.

6. Conclusión

A lo largo de este trabajo desarrollamos tres diferentes elementos a tener en cuenta al momento de crear una interfaz natural gestual: dispositivos de captura, gestos de entrada y programas de detección, mostrando como distintas iteraciones de estos elementos presentan diferentes características, ventajas y desventajas. A través de este desarrollo buscamos no solo que el lector pueda conocer y comprender las diferentes opciones disponibles a la hora de crear su propia interfaz, sino que pueda hacerlo teniendo en cuenta las especificidades propias a su proyecto como son el tamaño y luz en el espacio de desarrollo, los destinatarios, la complejidad de la interfaz, la cantidad de acciones o entradas necesarias, entre otras cosas; y que pueda a partir de estas consideraciones tomar una decisión educada con respecto a la combinación de elementos que más se adecua a su proyecto, a sus necesidades y objetivos específicos.

7. Referencias

1. Echeverri, O. J. G. (2014). *Consideraciones en el desarrollo de in-*

terfaces naturales gestuales. Revista CINTEX (Vol. 19, pp. 183-193). https://revistas.pascualbravo.edu.co/index.php/cintex/article/view/46/48

2. Suarez, J., & Murphy, R. R. (2012). *Hand gesture recognition with depth images: A review*. In 2012 IEEE RO-MAN: the 21st IEEE international symposium on robot and human interactive communication (pp. 411-417). https://ieeexplore.ieee.org/abstract/document/6343787
3. Kinect: desarrollo de aplicaciones de Windows. (s. f.). Microsoft. Com. https://developer.microsoft.com/es-es/windows/kinect/
4. Vuletic, T. (2019). *Systematic literature review of hand gestures used in human computer interaction interfaces*. International Journal of Human-Computer Studies. https://doi.org/10.1016/j.ijhcs.2019.03.011
5. *OpenCV*. (2021) (Versión 2.4.2) [Librería] https://opencv.org/
6. atduskgreg. (2017). OpenCV for Processing (Versión 0.5.4) [Librería de Processing]. Descargado de: https://github.com/atduskgreg/opencv-processing/releases
7. OpenNI (2013) (Version 1.5.4.0) [Librería] Descargado de: https://github.com/OpenNI/OpenNI
8. Totovr.(2019)SimpleOpenNI [Librería de Processing]. (Versión 3.5.3)
9. Descargado de: https://github.com/totovr/SimpleOpenni/tree/Processing_3.5.3
10. Borenstein,G.(2012).FingerTracker. GitHub. [Librería de Processing]. (Versión 3.5.2) Descargado de: https://github.com/atduskgreg/FingerTracker
11. Google. (2015). *Tensor Flow.* Disponible en: https://www.tensorflow.org/?hl=es-419
12. Oved, D. (2018) *Real-time human pose estimation in the browser with tensorflow. js*. TensorFlow Medium. Disponible en: https://medium.com/tensorflow/real-time-human-pose-estimation-in-the-browser-with-tensorflow-js-7dd0bc881cd5
13. LingDong. (2020). PoseOSC (Versión 0.0.3) [Aplicación]. Descargado de: https://github.com/LingDong-/PoseOSC/releases/tag/0.0.3
14. Saenz, J. Schön A. , Espinosa L. N. (2021) *Desarrollo de interfaces naturales gestuales con Kinect*. Invasión Generativa IV
15. LAB at Rockwell, IDEO Labs. (2017). TSPS (Versión 1.3.6) [Aplicación]. Descargado de: https://www.tsps.cc/

Notas

1. Campo de visión (FOV): El campo de visión, más comúnmente llamado por sus siglas en inglés, el FOV (Field of View) se refiere al área total que puede detectar el sensor. Depende del tamaño del sensor, la distancia focal y la distancia a la que se esté enfocando. Puede medirse el FOV horizontal, vertical y diagonal calculando la longitud de la imagen a una distancia dada del lente.

2. Cámaras de profundidad: Una cámara de profundidad (o depth camera) se refiere a cualquier cámara que pueda obtener información sobre la distancia de los objetos presentes en la escena que esté viendo.

3. RGB-D: Una cámara que además de detectar profundidad puede detectar información de color.

4. Visión Computacional: Área de la ciencia que busca desarrollar técnicas y algoritmos para el análisis y la interpretación de imágenes por computadora.

5. Machine Learning: Un subcampo de las ciencias de la computación y una rama de la inteligencia artificial, cuyo objetivo es desarrollar técnicas que permitan que las computadoras aprendan con la experiencia.

6. Cascadas de Haar: Método de detección de objetos basado en imágenes. Recibe imágenes que tienen aquello que se quiere detectar (imágenes positivas) e imágenes que no lo tengan (imágenes negativas) y a partir ellas se extraen y agrupan las características que hacen al objeto que se quiere reconocer. Para realizar la detección el algoritmo recorre la imagen y evalúa si cumple con todas las características encontradas, yendo de la más general a la más particular.