UNIVERSIDAD DE GRANADA

Nuevos Paradigmas de Interacción

grado en ingeniería informática

Práctica Gestos

Memoria Técnica

Autores

Vladislav Nikolov Vasilev

José María Sánchez Guerrero

Fernando Vallecillos Ruiz

Rama

Computación y Sistemas Inteligentes

Escuela Técnica Superior de Ingenierías Informática y de Telecomunicación

Curso 2019-2020

Índice

1. Introducción

En este proyecto vamos a explicar nuestra *Natural User Interface (NUI)* que hará que nuestra visita a la Alhambra sea más dinámica y productiva. El proyecto constaría de varios dispositivos, como pueden ser principalmente unas gafas de realidad aumentada, un dispositivo *Leap Motion* y un micrófono integrados en las gafas, y un *smartphone* que utilizaremos tanto para manejar el sistema gracias a los múltiples sensores que incorpora como para controlar la aplicación.

En esta segunda versión del proyecto vamos a encargarnos de la interfaz por gestos, es decir, dejaremos a un lado los sensores y el controlador por voz. Vamos a encargarnos de la realidad aumentada y los gestos para manejar el sistema.

2. Interfaz por gestos

Como no disponemos de unas gafas con un Leap Motion incorporado, vamos a realizar nuestra simulación en un programa de `Python` aparte. En nuestra versión, la opción de utilizarlo verticalmente no está disponible, por lo que su funcionamiento se mostrará con el dispositivo sobre la mesa. Se han implementedo los siguientes gestos: *click*, *swipe*, zoom y agarre y movimiento. Un poco más adelante vamos a describir en qué consiste cada gesto y diremos muy brevemente cómo se ha implementado cada uno de ellos.

3. Información y estructuras de datos utilizadas

4. Descripción de los gestos

En esta sección se van a describir todos los gestos que se han implementado. Se va a hacer una breve descripción de en qué consiste y se va a mostrar muy por encima el código asociado en los casos en los que se haya considerado necesario (cuando lo que se quiere mostrar es relevante ya que se trata de algo no trivial).

Antes de continuar, hay que destacar que algunos de los gestos solo están disponibles cuando ambas manos están encima del *Leap Motion*. Por ejemplo, el *click*, el *swipe* y el agarre y movimiento solo pueden ser realizados cuando se detecta una única mano. El zoom solo se puede realizar si se han detectado dos manos y se hace el gesto correspondiente con los dedos. Para comprobar si se usa una mano o dos, podemos hacer los siguiente:

```
1 if(len(frame.hands) == 1):
2     ...
```

```
3
4 f(len(frame.hands) >= 2 ):
5     ...
```

En este caso, comprobamos si se han detectado dos o más manos, ya que puede pasar que otra persona pase la mano por encima del dispositivo. Sin embargo, nos quedamos siempre con las dos primeras manos que hayamos detectado.

4.1. Descripción del *click*

Este gesto sirve para mostrar información extra sobre el objeto. El gesto consiste en cerrar y abrir el puño en la zona sobre la que queremos obtener información. Esta zona se detectaría automáticamente mediante un algoritmo implementado en las gafas, pero como no disponemos de dicha tecnología, utilizaremos una imagen fija con una zona resaltada

Esto mostrará en la interfaz de nuestras gafas de realidad aumentada una imagen de la zona resaltada, junto con información de esta. La información podrá ser modificada mediante otros gestos descritos posteriormente.

Para saber si estamos haciendo o no un *click*, simplemente tenemos que comprobar la siguiente condición:

```
1 if (closed_hand and hands[0].grab_strength == 1.0):
2     ...
```

Después se va a comprobar si la mano está posicionada en la zona adecuada, en cuyo caso se mostrará la imagen que contiene información.

4.2. Descripción del *swipe*

Al igual que mostramos la información con el gesto anterior, necesitaremos otro gesto para ocultarla. Nosotros utilizamos el *swipe*, el cual detectará los movimientos en el eje X. De esta formá, podrá ser usado tanto por personas zurdas como diestras sin ningún tipo de problema. Modificaremos el parámetro de su velocidad para que no se pueda confundir con otros gestos que hacemos naturalmente.

La implementación es la siguiente:

```
1 if (len(frame.gestures()) != 0):
2     # Check if the hand is completely open and the gesture is swiping
3     if open_hand and (frame.gestures()[0].type == Leap.Gesture.TYPE_SWIPE
      ):
4         swipe = SwipeGesture(frame.gestures()[0])
```

```
        # Check that the main direction of the swipe is the x-axis (left
    or right)
        if(abs (swipe.direction[0]) > 0.7):
            swiping = True
```

Simplemente estamos comprobando si se ha detectado algún gesto y si ese gesto se ha realizado con la mano abierta y es un *swipe* en cualquier sentido en el eje X.

Como se puede ver, estamos utilizando un gesto que ya viene en *Leap*, aunque lo estamos adaptando a nuestras necesidades, ya que hemos limitado su funcionamiento al eje X.

4.3. Descripción del zoom

Este gesto sirve para disminuir o aumentar el tamaño de la información mostrada. Para utilizar este gesto tendremos que tener las dos manos sobre la zona de detección del *Leap Motion* y, posteriormente, extender los dedos pulgar e índice con una apertura mínima entre ellos de 40º. El zoom que se le aplicará a la imagen será directamente proporcional a la distancia en el eje X entre la mano derecha y la mano izquierda. La implementación de dicho gesto es la siguiente:

```
# Let's check if both hands have only the thumb and index extended
two_up_left = True
two_up_right = True

for i in range(0, 2):
    two_up_left = two_up_left and left_hand.fingers[i].is_extended
    two_up_right = two_up_right and right_hand.fingers[i].is_extended

for i in range(2, 5):
    two_up_left = two_up_left and not left_hand.fingers[i].is_extended
    two_up_right = two_up_right and not right_hand.fingers[i].is_extended

# Set the minimum angle between thumb and index to be considered zooming
threshold_angle = 40

# If they have the left index and thumb extended, we calculate the angle
if two_up_left:
    vec1 = left_hand.fingers[0].direction    # Thumb
    vec2 = left_hand.fingers[1].direction    # Index
    angle1 = acos(max(-1.0, min(1.0, vec1.dot(vec2)))) * Leap.RAD_TO_DEG

    # Check if the angle is open enough
    if (angle1 > threshold_angle):
            left_hand_zoom = True
    else:
            left_hand_zoom = False
else:
    left_hand_zoom = False
```

```
29
30 # If they have the right index and thumb extended, we calculate the angle
31 if two_up_right:
32     vec1 = right_hand.fingers[0].direction    # Thumb
33     vec2 = right_hand.fingers[1].direction    # Index
34     angle2 = acos(max(-1.0, min(1.0, vec1.dot(vec2)))) * Leap.RAD_TO_DEG
35
36     # Check if the angle is open enough
37     if (angle2 > threshold_angle):
38             right_hand_zoom = True
39     else:
40         right_hand_zoom = False
41 else:
42     right_hand_zoom = False
```

Básicamente, lo que se hace es comprobar si los dedos índice y pulgar de las dos manos están extendidos y el resto cerrados. Después, se comprueba si en cada caso el ángulo que forman es mayor que 40º, en caso de que la condición anterior se haya satisfecho.

Además de hacer zoom, también está implementada una función que, si llegamos a un umbral de zoom en algunas zonas determinadas (por ejemplo, a la fuente de la funete de los leones, o una habitación del Palacio de Carlos V), mostraremos información más detallada sobre ésta. La implementación es la siguiente?? o ya la hemos puesto antes???

4.4. Descripción del agarre y movimiento

Para mover la información sobre toda la superficie de las gafas de realidad aumentada, utilizaremos el siguiente gesto. Cerraremos el puño (tendrá que ser en las mismas coordenadas sobre las que está la información, para así evitar moverla sin querer con algún gesto natural) y posteriormente la moveremos al lugar que queramos. Para dejar de moverla, simplemente volvemos a abrir la mano.

Si estamos arrastrando la información fuera de nuestro campo de visión, hemos puesto unos límites tanto vertical como horizontalmente, para que el programa no nos permita hacerlo.

Arrastrada la información, también guardaremos las propiedades que conservaba previamente, como por ejemplo, si le habíamos hecho zoom, o la habíamos arrastrado en otro momento.

Para comprobar que estamos haciendo un agarre con la mano, tenemos que comprobar la misma condición que con el *click*, aunque lo que se va a ejecutar va a ser diferente, ya que se va a actualizar la posición de la imagen en función del movimiento realizado, respetando siempre los bordes que se hayan impuesto.

Referencias