Practica 1:

Detección de movimientos básicos con Kinect

Jose Antonio España Valdivia

Índice

Información

Autor: Jose Antonio España Valdivia

Correo: jaespana@correo.ugr.es

Web: https://github.com/Yujadeh

Descripción del problema

El problema trata sobre como detectar una posición inicial y la realización de un movimiento en concreto que será captado por el dispositivo Kinect.

La posición inicial será estar recto con los brazos pegados al cuerpo y las piernas con una pequeña apertura.

El movimiento a realizar será la composición de dos movimientos: agacharse y levantar los brazos. Dichos movimientos deben de realizarse de forma consecutiva, de cualquier otro modo se considerará que el movimiento no es valido.

Descripción de la solución

Para poder abordar el problema que tenemos entre manos la solución que se da es la siguiente.

En primer lugar partimos de los ejemplos de reconocer el cuerpo y los colores que nos proporciona el kit de desarrollo de Kinect.

Para ayudarnos a la detección de posturas, se crea el enumerador siguiente:

```
public enum posturas { Mal, Postura_Inicial, Sigue_Bajando, Agachado, Final };
```

Para saber en que posición estamos en el momento se usará la variable:

```
posturas posturaActual = posturas.Postura_Inicial;
```

A continuación creamos dos métodos principales que serán ejecutados cada vez que el Kinect proporcione información, es decir, cada vez que se capta un frame. Estas funciones son:

```
public void CompruebaPostura(Skeleton esqueleto){
                if (posturaActual == posturas.Postura_Inicial){
                    altura_cabeza = esqueleto.Joints[JointType.Head].Position.Y;
                    captarDistancias(esqueleto);
                }
                else if (posturaActual == posturas.Sigue_Bajando){
                    this.Mensaje.Text = "\t Baja hasta la posición indicada";
                    Agachado(esqueleto);
                }
                else if (posturaActual == posturas.Agachado){
                    this.Mensaje.Text = "\t Coloque los brazos hacia arriba";
                    comprobarSubida(esqueleto);
                    ManosArriba(esqueleto);
                }
                else if (posturaActual == posturas.Final){
                    this.Mensaje.Text = "\t Se acabo el ejercicio";
```

```
                posturaActual = posturas.Postura_Inicial;
            }
            else if (posturaActual == posturas.Mal) {
                this.Mensaje.Text = "\t ERROR: Empieza de nuevo";
                posturaActual = posturas.Postura_Inicial;
            }
        }
```

En esta función haremos uso del enumerador ya comentado para saber la postura en la que estamos y poder realizar el movimiento correctamente. En primer lugar, vemos si el usuario está en la posición inicial, en caso de que lo esté tomaremos los datos y haremos los cálculos pertinentes para el desarrollo del movimiento.

Una vez realizado lo anterior, indicamos al usuario que se agache hasta cierto punto para posteriormente realizada la sentadilla que suba los brazos hasta el cielo.

Por otro lado tenemos un caso de error en caso de que el movimiento no se realice de forma fluida

Por último finalizamos el ejercicio e inicializamos las variables de nuevo.

La segunda de las funciones principales es la siguiente:

```
public void Indicaciones(Skeleton esqueleto, DrawingContext drawingContext){

            if (posturaActual == posturas.Sigue_Bajando){
                posicionamientoAgache(esqueleto,drawingContext);
            }
            else if (posturaActual == posturas.Agachado)
            {
                posicionamientoManos(esqueleto, drawingContext);
            }
        }
```

En esta función proporcionaremos ayuda al usuario con unas esferas que cambiarán de color al pasar por la zona correcta.

En primer lugar como el primer movimiento consiste en agacharse hasta cierto punto lo indicaremos de modo que el usuario baje hasta que su cabeza se sitúe en una esfera de color rojo.

En segundo lugar indicamos al usuario que suba los brazos hasta cierto punto, realizando esto con un par de esferas en las cuales el usuario ha de "introducir" el brazo quedando sus codos y manos en dichas esferas.

Como podemos observar estos métodos son los más generales y necesitan

la ayuda de otros métodos para realizar lo indicado. En primer lugar veamos todos los métodos que consisten en la realización de cálculos internos para posteriormente ver las funciones que expondrán dichos cálculos al usuario mediante señales visuales para poder realizar el movimiento correctamente.

Métodos de cálculo

En este apartado destacamos tres grandes métodos:

```
public void Agachado(Skeleton esqueleto){

            Joint cabeza = esqueleto.Joints[JointType.Head];

            /*Pasos de comprobación:
             *  - Compruebo que la posición de la cabeza está a una determinada altura
             */
            if (altura_cabeza - cabeza.Position.Y > 0.30 ) {//then
                if (pinta_correcto == false) {
                    altura_cabeza_agachado = cabeza.Position.Y;
                    pinta_correcto = true;
                    frame_aux = numero_frame;
                }
                else if (frame_aux + 30 < numero_frame){
                    posturaActual = posturas.Agachado;
                    pinta_correcto = false;
                }
            }
            else if (pinta_correcto == true)
                pinta_correcto = false;

        }
```

En este primer método comprobamos si el usuario se ha agachado hasta cierto punto. En cuanto esto sea realizado, proporcionamos al usuario una salida visual para indicarle que el movimiento de agacharse se ha realizado correctamente.

```
        public void ManosArriba(Skeleton esqueleto){

            Joint muñecaDerecha = esqueleto.Joints[JointType.WristRight];
            Joint hombroDerecho = esqueleto.Joints[JointType.ShoulderRight];
            Joint codoDerecho = esqueleto.Joints[JointType.ElbowRight];
            Joint muñecaIzquierda = esqueleto.Joints[JointType.WristLeft];
            Joint hombroIzquierdo = esqueleto.Joints[JointType.ShoulderLeft];
            Joint codoIzquierdo = esqueleto.Joints[JointType.ElbowLeft];

             /*   Pasos de comprobación:
             *  - Compruebo que la muñeca y el codo estén en la misma recta vertical
             *  - Compruebo que el hombro y el codo estén en la misma recta vertical
             *  - Compruebo que la muñeca esté por encima del codo
             *  - Compruebo que el codo esté por encima del hombro
             *  - Compruebo que no tengo la muñeca inclinada(en profundidad) respecto al
                    codo(o viceversa)
             *  - Compruebo que no tengo el codo inclinado(en profundidad) respecto al
                    hombro (o viceversa)
             */

      if (   muñecaDerecha.Position.Y > codoDerecho.Position.Y &&
```

```
            codoDerecho.Position.Y > hombroDerecho.Position.Y &&
        Math.Abs(muñecaDerecha.Position.X - codoDerecho.Position.X) < this.tolerancia &&
        Math.Abs(hombroDerecho.Position.X - codoDerecho.Position.X) < (this.tolerancia) &&
        Math.Abs(muñecaDerecha.Position.Z - codoDerecho.Position.Z) < this.tolerancia*3 &&
        Math.Abs(hombroDerecho.Position.Z - codoDerecho.Position.Z) < (this.tolerancia) &&

        muñecaIzquierda.Position.Y > codoIzquierdo.Position.Y &&
         codoIzquierdo.Position.Y > hombroIzquierdo.Position.Y &&
         Math.Abs(muñecaIzquierda.Position.X - codoIzquierdo.Position.X) < this.tolerancia &&
         Math.Abs(muñecaIzquierda.Position.Z - codoIzquierdo.Position.Z) < this.tolerancia &&
         Math.Abs(hombroIzquierdo.Position.X - codoIzquierdo.Position.X) < this.tolerancia &&
         Math.Abs(hombroIzquierdo.Position.Z - codoIzquierdo.Position.Z) < this.tolerancia*3)
         {//then
                 if (pinta_correcto == false){
                     pinta_correcto = true;
                     frame_aux = numero_frame;
                 }
                 else if (frame_aux + 30 < numero_frame){
                         posturaActual = posturas.Final;
                         pinta_correcto = false;

                 }
             }else if (pinta_correcto == true)
                     pinta_correcto = false;
          }
```

En este segundo método comprobamos si las manos están arriba para posteriormente comunicarle al usuario mediante señales visuales en modo de esferas azules que el movimiento realizado es correcto.

```
        public void comprobarSubida(Skeleton esqueleto){
            Joint cabeza = esqueleto.Joints[JointType.Head];
            /*Pasos de comprobación:
             *  - Compruebo que la posición de la cabeza está a una determinada altura
             */
            if (cabeza.Position.Y > altura_cabeza_agachado+0.1) {//then
                posturaActual = posturas.Mal;
            }
        }
```

Por último, este método comprobará que el usuario una vez agachado no suba una cierta distancia. En caso de subirla el movimiento quedará anulado y se tendrá que realizar desde el principio.

Métodos de visualización

A continuación se muestran los métodos que guiarán al usuario por la pantalla del PC a situarse correctamente.

En primer lugar tenemos el método que nos indicará hasta que punto el usuario ha de agacharse:

```
private void posicionamientoAgache(Skeleton esqueleto, DrawingContext drawingContext){
            //Puntos que indican donde se deben situar las manos y los codos.
            SkeletonPoint cabeza = new SkeletonPoint();
            cabeza.X = esqueleto.Joints[JointType.Head].Position.X;
            cabeza.Y = altura_cabeza - dist_agache;
            cabeza.Z = esqueleto.Joints[JointType.Head].Position.Z;

            if (pinta_correcto == false){
                Point pos1 = this.SkeletonPointToScreen(cabeza);
                drawingContext.DrawEllipse(inferredJointBrush, null, pos1, 15, 15);

            }
            else{
                Point pos1 = this.SkeletonPointToScreen(cabeza);
                drawingContext.DrawEllipse(centerPointBrush, null, pos1, 15, 15);
            }

        }
```

Este método fija un punto según la altura del usuario para que dicho usuario baje hasta colocar su cabeza a dicha altura. Si el usuario no está a la altura correspondiente el punto será de color rojo. En caso contrario, el punto será de color azul.

```
private void posicionamientoManos(Skeleton esqueleto, DrawingContext drawingContext){
            //Puntos que indican donde se deben situar las manos y los codos.
            SkeletonPoint manoDerecha = new SkeletonPoint();
            SkeletonPoint codoDerecha = new SkeletonPoint();
            SkeletonPoint manoIzquierda = new SkeletonPoint();
            SkeletonPoint codoIzquierda = new SkeletonPoint();

            // X sera la distancia euclidea entre la muñeca y el hombro
            manoDerecha.X = esqueleto.Joints[JointType.ShoulderLeft].Position.X;
            manoDerecha.Y = esqueleto.Joints[JointType.ShoulderLeft].Position.Y +
distancia_hombro_muñeca;
            manoDerecha.Z = esqueleto.Joints[JointType.ShoulderLeft].Position.Z;

            // X sera la distancia euclidea entre el codo y el hombro
            codoDerecha.X = esqueleto.Joints[JointType.ShoulderLeft].Position.X;
            codoDerecha.Y = esqueleto.Joints[JointType.ShoulderLeft].Position.Y +
distancia_hombro_codo;
            codoDerecha.Z = esqueleto.Joints[JointType.ShoulderLeft].Position.Z;


            //La X sera la distancia euclidea entre la muñeca y el hombro
            manoIzquierda.X = esqueleto.Joints[JointType.ShoulderRight].Position.X;
            manoIzquierda.Y = esqueleto.Joints[JointType.ShoulderRight].Position.Y +
distancia_hombro_muñeca;
            manoIzquierda.Z = esqueleto.Joints[JointType.ShoulderRight].Position.Z;

            // X sera la distancia euclidea entre el codo y el hombro
            codoIzquierda.X = esqueleto.Joints[JointType.ShoulderRight].Position.X;
            codoIzquierda.Y = esqueleto.Joints[JointType.ShoulderRight].Position.Y +
distancia_hombro_codo;
            codoIzquierda.Z = esqueleto.Joints[JointType.ShoulderRight].Position.Z;

            if (pinta_correcto == false){
                Point pos1 = this.SkeletonPointToScreen(manoDerecha);
                drawingContext.DrawEllipse(inferredJointBrush, null, pos1, 15, 15);
                Point pos2 = this.SkeletonPointToScreen(codoDerecha);
                drawingContext.DrawEllipse(inferredJointBrush, null, pos2, 15, 15);
                Point pos3 = this.SkeletonPointToScreen(manoIzquierda);
```

```
                drawingContext.DrawEllipse(inferredJointBrush, null, pos3, 15, 15);
                Point pos4 = this.SkeletonPointToScreen(codoIzquierda);
                drawingContext.DrawEllipse(inferredJointBrush, null, pos4, 15, 15);
            }
            else{
                Point pos1 = this.SkeletonPointToScreen(manoDerecha);
                drawingContext.DrawEllipse(centerPointBrush, null, pos1, 15, 15);
                Point pos2 = this.SkeletonPointToScreen(codoDerecha);
                drawingContext.DrawEllipse(centerPointBrush, null, pos2, 15, 15);
                Point pos3 = this.SkeletonPointToScreen(manoIzquierda);
                drawingContext.DrawEllipse(centerPointBrush, null, pos3, 15, 15);
                Point pos4 = this.SkeletonPointToScreen(codoIzquierda);
                drawingContext.DrawEllipse(centerPointBrush, null, pos4, 15, 15);
            }

        }
```

Este segundo método situará dos esferas por brazo a una altura superior al usuario, indicando así que este ha de subir los brazos hasta los puntos mencionados. Esto se manifiesta de forma que hasta que el usuario no tenga su codo y muñeca situados en las esferas estas indicarán de forma negativa que el movimiento no se ha realizado gracias al color rojo. Cuando el usuario haya colocado correctamente los brazos, las esferas tornarán a color azul.

Sección de errores

Los errores encontrados al realizar la práctica son:

- A la hora de detectar el movimiento, debido a la gran falta de precisión de Kinect versión 1.0, es necesario introducir un pequeño bucle entre postura y postura para poder detectar y analizar correctamente el movimiento correspondiente.
- En la realización del método de detectar la subida del usuario, el margen que se le da ha de ser mínimo de 0.1 ya que para distancia menor suele dar como resultado una subida del usuario cuando este a vista humana no ha movido su cuerpo.
- Al intentar añadir profundidad en la detección del movimiento de agacharse el Kinect no respondía bien con la tolerancia pasada por lo cuál se ha suprimido ese aspecto. Se comproborá y añadirá correctamente en la segunda práctica.
- Si se inicia el movimiento inclinado hacia delante si se tienen los brazos y piernas correctamente situados la distancia a agacharse será mayor.

Sección de lectura

Leer las wikis/blogs de gente que haya realizado la asignatura. La información disponible en internet es algo liosa y no suele ir muy enfocada a lo que se quiere en la asignatura.

Referencias

Como se ha comentado en el apartado anterior, me he basado en personas que ya han cursado esta asignatura para poder entender el comportamiento de Kinect y como poder realizar mis movimientos.

En primer lugar, para entender como Kinect capta los puntos del esqueleto y como se trabaja con dichos puntos me he basado en las prácticas de Hugo Mario Lupión Fernández:

- Web: https://github.com/Katovit

Posteriormente para la visualización por pantalla no solo del esqueleto sino de la imagen en tiempo real me he basado en lo desarrollado por:

- Alumno: Jose Arcos Aneas
- Web: https://github.com/josearcosaneas
- Correo: joseaa@correo.ugr.es