

CONTROL VIRTUAL DE UN COMPUTADOR MEDIANTE EL SENSOR KINECT

Cristian Gallardo¹, Jaime Santana¹, Renato Toasa¹, Johana Villacrés¹, Jessica S. Ortiz^{1,2}, Paúl Romero² y Víctor H. Andaluz¹

¹Universidad Técnica de Ambato, Ambato-Ecuador
²Escuela Superior Politécnica de Chimborazo, Riobamba-Ecuador
victorhandaluz@uta.edu.ec

RESUMEN:

Este trabajo presenta la interacción entre el Sensor Kinect y un computador, permitiendo al usuario controlar programas de Windows por medio de gestos, sin necesidad de usar un dispositivo, donde “el control eres tú”. Para el desarrollo del presente trabajo se realiza el rastreo de ciertas partes del cuerpo permitiendo al usuario manipular el ordenador dependiendo de sus necesidades. En esta aplicación se utiliza dos funcionalidades del sensor Kinect; la cámara de profundidad y skeleton tracking, esto se efectuó utilizando una programación orientada a objetos en Visual Studio 2010 (C#, WPF). Se realizó pruebas experimentales en las cuales se comprobó el desempeño de la propuesta.

Palabras clave: Sensor Kinect, skeleton tracking, programación orientada a objetos.

ABSTRACT:

This work presents the interaction between the Kinect Sensor and a computer, allowing the user to control Windows programs through gestures without using any devices, where “you are the control”. For the development of this work certain body parts were tracked which allows the user to manipulate the computer depending on their needs. Two of the functionalities of the Kinect sensor are used in this application; the depth camera and skeleton tracking. This was done using Visual Studio 2010 (C#, WPF) object oriented programming. Experimental tests were done with which the performance of the proposal was validated.

Keywords: Kinect sensor, skeleton tracking, object oriented programming

1. Introducción /

El sensor Kinect es un dispositivo creado por Microsoft para la consola de videojuegos XBOX 360, ha traspasado las fronteras del mundo del ocio y se ha convertido en una potente plataforma de desarrollo, a un coste más que razonable [1]. Kinect hoy en día nos permite controlar e interactuar con la computadora mediante el reconocimiento de gestos, voz, objetos e imágenes. Cuenta con una base giratoria motorizada, cuatro micrófonos que reconocen voces por separado, tres cámaras con sensor de movimiento (cámara de captación de movimiento VGA y doble cámara de profundidad 3D) [2].

La cámara de profundidad captura los datos de profundidad a fondo, sirve para saber cuán lejos se encuentra un píxel del sensor para calcular la distancia a partir de esos datos [3]. La misma se puede usar con la opción de sólo profundidad o profundidad con índice de jugador.



Figura 1. Sensor Kinect

Artículo Recibido: 1 de julio de 2013

Artículo Aceptado: 25 de septiembre de 2013

Para lograr la interacción con el computador, se necesita ciertas librerías que nos ofrece el SDK de Kinect para Windows, este incluye drivers, audio, API para el sensor y seguimiento del movimiento humano [4]. El SDK es un conjunto de herramientas de desarrollo de software que permite al programador crear aplicaciones para un sistema concreto [5].

En la actualidad existe un gran interés en la comunidad científica para la utilización del sensor Kinect, utilizándolo en áreas como: robótica [6, 7,8], procesamiento de imágenes [9, 10, 11], educación y aprendizaje [12, 13], etc.

Con el desarrollo de la computación y la invención de nuevas tecnologías, se inició el estudio de la captura de movimiento de individuos utilizando técnicas de visión por computadora [14]. Es utilizada para un gran número de aplicaciones, entre ellas tenemos: vigilancia inteligente, realidad virtual, interpolación de vistas, análisis de movimiento, reconocimiento.

La manipulación virtual de un computador permite realizar una interacción directa entre el computador y el usuario, ya que se puede manejar un ordenador a distancia como si se estuviera delante del mismo, según sea nuestra necesidad. Los componentes para la manipulación virtual de un computador son: interacción, simulación y percepción; lo que es aplicado en varias áreas como: medicina, educación, defensa, arquitectura, marketing, entretenimiento, robótica [15].

Los trabajos previos que se ejecutaron y de los cuales se partió para realizar el presente trabajo son: Microsoft presenta un prototipo de escritorio interactivo 3D con pantalla transparente OLED y Kinect, aplicación que permite a los usuarios manipular con sus manos el escritorio virtual en tiempo real [16]. Sistema de reconocimiento de los gestos de las manos usando mapa de profundidad, trabajo que presenta un nuevo método para reconocer los gestos de las manos [17].

En este trabajo se propone el diseño de una aplicación orientada al control y monitoreo virtual de un ordenador por medio del sensor Kinect.

El esquema de control multi-capas propuesto está principalmente dividido en cuatro módulos: 1) el módulo superior es el encargado de inicializar el Kinect e identificar al usua-

rio; 2) el siguiente el módulo de detección de movimientos es el responsable capturar los movimientos del usuario y generar los comandos necesarios para el control de las distintas aplicaciones; 3) el módulo de aplicaciones está representados por todas las aplicaciones que se desea controlar y monitorear; 4) finalmente el módulo de interacción virtual es el encargado de controlar en ordenador.

El documento se encuentra dividido en 6 secciones, incluida la introducción. La sección 2 incluye el esquema de programación. La sección 3 comprende skeleton tracking y captura de movimientos. La sección 4 incluye el diseño del trabajo. La sección 5 contiene los resultados experimentales obtenidos con el desarrollo del trabajo y finalmente las conclusiones del trabajo se presentan en la sección 6.

2. Esquema de Programación /

La figura 3, muestra el esquema de control considerado en este trabajo para el monitoreo y control virtual de un computador por medio del sensor Kinect.

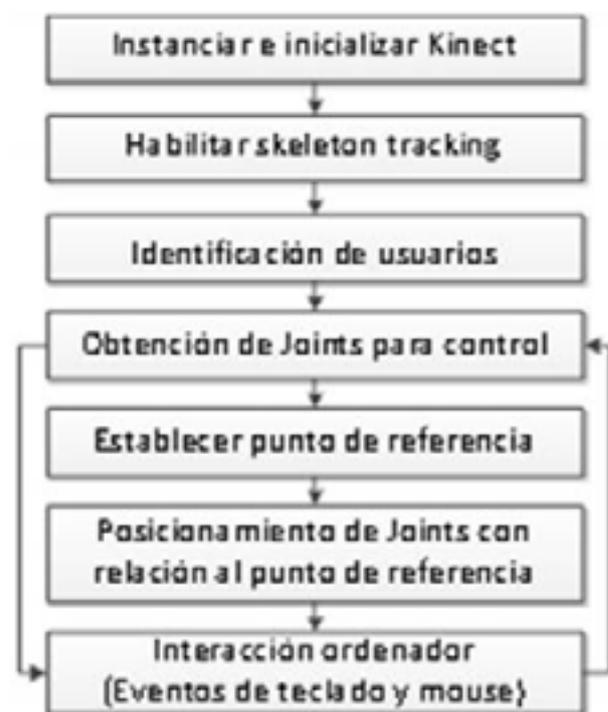


Figura 2. Esquema aplicado

Instanciar e iniciar el Kinect, responsable de asignar al sensor Kinect una variable en la aplicación. En la que se obtiene la información de las 3 cámaras que dispone el sensor.

Habilitar skeleton tracking, responsable de configurar a la variable declarada anteriormente para que regrese información de los Joints del skeleton.

Identificación de usuarios, en esta sección se realiza un control para capturar al individuo que se encuentre más cerca del kinect.

Obtención de Joints para control, responsable de que a partir de la información devuelta en los Joints asignar a variables locales que guardarán las coordenadas de cada punto. Establecer puntos de Referencia, en esta sección se realiza el cálculo con las coordenadas de los Joints para definir el punto desde donde se referenciará los movimientos del cuerpo.

Posicionamiento de Joints en relación al punto de Referencia, es responsable de obtener los puntos del cuerpo con los que se trabaja y se obtiene las coordenadas en las que se acotan en los planos x, y, z en relación al punto anterior con eso se calcula el punto de referencia.

Interacción Ordenador (Eventos teclado y mouse,) se envía la información al ordenador sobre los movimientos que realiza el usuario.

3. Captura de Movimientos y Skeleton Tracking /

A. Captura de Movimientos

En [18] se define como: proceso que consiste en traducir movimientos reales a representaciones digitales y se logra mediante el seguimiento de un conjunto de puntos de interés en una escena por un tiempo determinado.

La captura de movimiento se puede realizar sobre cualquier actor que posea movimiento.

Para realizar el seguimiento del actor, se define un conjunto de puntos de interés sobre el mismo, los cuales se localizan en áreas de éste que posean mayor información sobre sus movimientos. Estos puntos corresponden a articulaciones del cuerpo [19].

La siguiente imagen muestra la forma en que reconoce y captura el sensor Kinect a la persona.



Figura 3. Forma de capturar la imagen con Kinect

A continuación se explicará la imagen anterior:

1. Un foco de luz ilumina al sujeto que refleja el haz al sensor. El rebote de un haz de láser permite que la cámara capte cuanto se separa, a esto se le llama "campo de profundidad".
 2. El sensor del chip calcula la distancia a partir del tiempo de salida y llegada a la luz en cada pixel. El Kinect recibe el haz de luz como infrarrojos que varía en mayor o menor grado de color. De esta manera los cuerpos aparecen de color rojo, verde y los cuerpos más lejanos aparecen en gris.
 3. Un software de imagen basado en un mapa de profundidad percibe e identifica los objetos en tiempo real; debe tomar la imagen y lo ejecuta a través de una serie de filtros para que reconozca si es o no una persona.
 4. El dispositivo final reacciona a la señal. Ya que está ordenada la información, convierte la identificación de las partes del cuerpo en un esqueleto en movimiento.
- B. Skeleton Tracking

Es una colección de Joints del cuerpo humano; para obtener las diferentes partes del cuerpo se aplica un algoritmo de búsqueda. Un Joint es un tipo de dato que devuelve el sensor Kinect con la información de los puntos de cuerpo.

Para el desarrollo de esta aplicación se realizaron las siguientes tareas:

Medir la longitud de los puntos del brazo, esto se hace mediante el cálculo de la distancia desde el hombro hasta el codo, a continuación a la muñeca, y, finalmente, a la mano. También se hace un cálculo desde el hombro hacia la cintura.

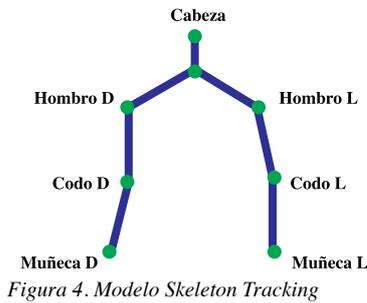


Figura 4. Modelo Skeleton Tracking

- Tercero, se comprueba que la altura de la mano derecha esté en el intervalo de 30 a 50 cm por debajo de la cabeza en el eje Y
- Por último, si las 3 condiciones anteriores se cumplen significa que la mano está en la posición esperada y realizamos la acción deseada.

B. Código del programa

El siguiente segmento muestra el código básico para empezar a trabajar con el sensor Kinect desde C#.

```
//Se importa la clase a nuestra aplicación;
using Microsoft.Kinect;
//Se agrega a la clase un método que inicialice el sensor
public void inicializarKinect()
{
    //Instancia el primer sensor Kinect conectado
    sensor = KinectSensor.KinectSensors.FirstOrDefault();
    // Método que Inicializa y configura el seguimiento de esqueleto
    configTraqueoSkeleton();
    //Iniciar sensor Kinect
    sensor.Start();
}
```

En el siguiente fragmento de código se realiza skeleton tracking mediante la función EventHandler de Windows.

```
void configTraqueoSkeleton()
{
    //Manejador para evento de seguimiento de esqueleto
    sensor.SkeletonFrameReady += new EventHandler<SkeletonFrameReadyEventArgs>(sensor_SkeletonFrameReady);
    //Activar seguimiento de esqueleto
    sensor.SkeletonStream.Enable();
}
void sensor_SkeletonFrameReady(object sender, SkeletonFrameReadyEventArgs e)
{
    //Obtención del skeleton con el cual se quiera trabajar
    if (skeleton.TrackingState == SkeletonTrackingState.Tracked)
    {
        var head = skeleton.Joints[JointType.Head];
        var rightHand = skeleton.Joints[JointType.HandRight];
        //Enviamos a la función que procese los datos de los puntos obtenidos
        operacionesControl(head, rightHand);
    }
}
```

También se utiliza la función SendKeys para enviar pulsaciones y combinaciones de teclas a una aplicación activa, se puede especificar con una sola tecla o una combinación de teclas por ejemplo: ALT, CTRL o MAYÚS (o cualquier

4. Diseño de Control

En esta sección se lo ha dividido en dos subsecciones: a) flujograma para el proceso de reconocimiento del movimiento del cuerpo humano, b) código del programa, donde se muestra los métodos más relevantes que se utiliza en el desarrollo de esta aplicación.

A. Flujograma de proceso de Reconocimiento

Para esta subsección se ha empleado el siguiente algoritmo de control, basado en el Skeleton Tracking.

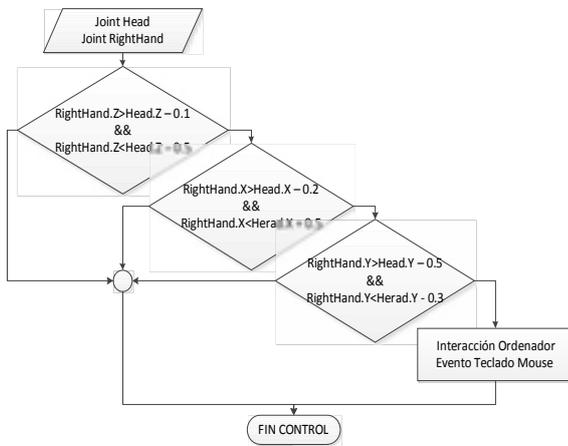


Figura 5. Proceso de reconocimiento

La gráfica expuesta anteriormente básicamente lo que hace es:

- Recibir los Joints de la cabeza y de la mano derecha y luego proceder a comprobar en los ejes X, Y, y Z la distancia de la mano en relación a la cabeza.
- Primero se comprueba en Z la profundidad en la que debe rastrearse, dentro del intervalo de 10 a 50 centímetros delante.
- Segundo, se comprueba la ubicación en el eje X que la mano derecha está dentro del intervalo de 20 a 50 cm a la derecha de la cabeza.

combinación de las teclas). Cada tecla está representada por uno o más caracteres.

```
//Función de la API de Windows que permite interactuar con el teclado en el entorno de Windows
[DllImport("user32.dll", SetLastError = true)]

void keybd_event(byte bVk, byte bScan, uint dwFlags, UIntPtr dwExtraInfo);
//Constantes para el envío de teclas
const int KEYEVENTF_KEYDOWN = 0x0;
const int KEYEVENTF_KEYUP = 0x2;
const int KEYEVENTF_EXTENDEDKEY = 0x1;

//Envío de tecla hacia una aplicación
void enviarTecla(String letraHexa )
{
    keybd_event(letraHexa, 0x45, KEYEVENTF_EXTENDEDKEY, (UIntPtr)0);
}
```

Para la interacción de la aplicación con el mouse se utiliza la función de la API de Windows.

```
[DllImport("user32.dll")]
void mouse_event(UInt32 dwFlags, UInt32 dx, UInt32 dy, UInt32 dwData, IntPtr dwExtraInfo);
//Constantes que identifican el evento del mouse
const UInt32 MOUSEEVENTF_LEFTDOWN = 0x0002;
const UInt32 MOUSEEVENTF_LEFTUP = 0x0004;
const UInt32 MOUSEEVENTF_RIGHTDOWN = 0x0008;
const UInt32 MOUSEEVENTF_RIGHTUP = 0x0010;
//Envío de evento del mouse hacia la aplicación
void SendEventClick()
{
    mouse_event(MOUSEEVENTF_LEFTDOWN, 0, 0, 0, new System.IntPtr());
}
//Movimiento del mouse, tomando como entrada las coordenadas devueltas por el Joint de la mano derecha
void MoverMouse(int x, int y)
{
    Cursor.Position = new Point(Cursor.Position.X+(x), Cursor.Position.Y + (y));
}
```

5. Resultados Experimentales

En esta sección se describen las aplicaciones controladas con el sensor Kinect.

A. PicLighter

La aplicación desarrollada ayuda a mover y realizar el zoom de una imagen seleccionada con el movimiento de las manos y brazos del usuario.

Con la mano derecha al frente se selecciona la imagen y se la mueve. Para realizar el Zoom de la imagen, se extiende las manos y se las separa, para alejar y para disminuir la imagen las manos se deben unir.

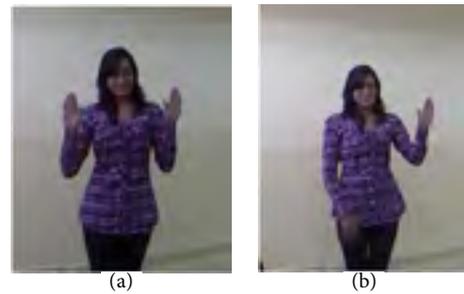


Figura 6. Zoom a la imagen (a) y (b) Mover imagen

B. Control de PowerPoint

Se realiza una aplicación para obtener el control total de PowerPoint. El usuario puede movilizarse en todas las diapositivas con el movimiento de las manos; para iniciar la presentación se levanta la mano derecha en su totalidad, para ir a la siguiente diapositiva se realiza un movimiento de la mano hacia la derecha, con el movimiento de la mano hacia la izquierda regresa una diapositiva y para finalizar la presentación se debe de levantar la mano izquierda en su totalidad. Para abrir otra presentación se coloca las manos al frente a nivel de la cintura deslizándolas hacia la izquierda, para tabular mano izquierda hacia adelante, para el control de las flechas realizamos movimientos de la mano derecha de arriba hacia abajo. El backspace (regresar) se realiza con la mano derecha hacia atrás.

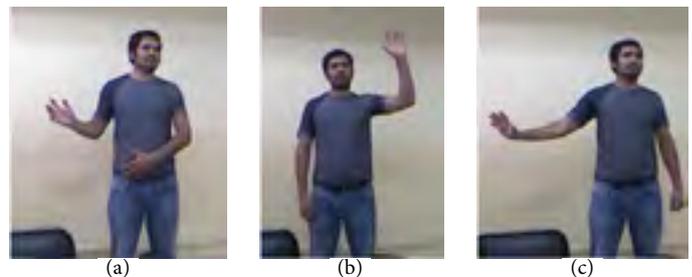


Figura 8. Nueva presentación, Iniciar presentación, Siguiente diapositiva



Figura 9. Resultados reales iniciando una presentación

C. NeedforSpeed

Se realizaron los controles necesarios para este juego, de tal modo que el usuario pueda jugar sin necesidad de utilizar el teclado, con el fin de simular que se está manejando un carro.

Paramovilzarse en el menú de opciones se lo realiza con el movimiento de la mano derecha de arriba-abajo, izquierda-derecha, para seleccionar una opción se coloca la mano derecha al nivel de la cintura hacia el frente. Si se desea acelerar se coloca el pie derecho adelante en manera de un paso, para retroceder el pie derecho atrás, para controlar el volante se hace una simulación como si estuviese manejando un vehículo en la realidad. Para salir del juego se levanta la mano izquierda, para regresar a un menú la mano derecha levantada.

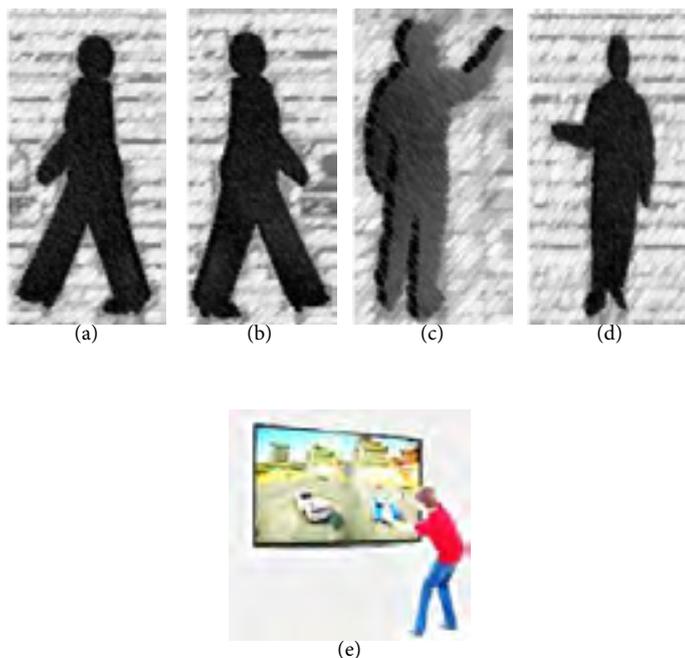


Figura 10. (a) Acelerar pie adelante, (b) Retroceder pie atrás, (c) Escape, (d) Seleccionar opción, (e) Control del volante.

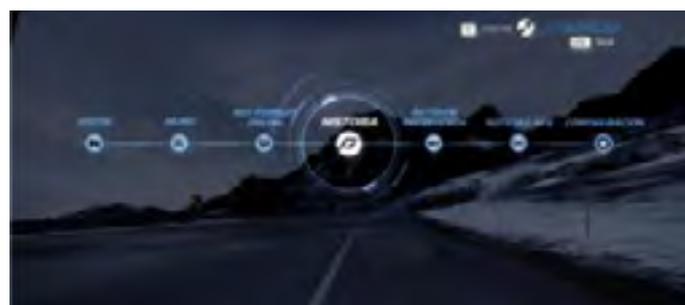


Figura 11. Desplazamiento por el menú del juego (a) y (b) Jugando

D. Paint

Aplicación simple que nos permite dibujar, abrir una imagen y editarla.

Para controlar esta aplicación se debe de mover la mano derecha simulando que es el puntero de la aplicación. Para simular el clic izquierdo lo realizamos con la mano izquierda sosteniéndola a la altura del hombro y movilizándola para dibujar.

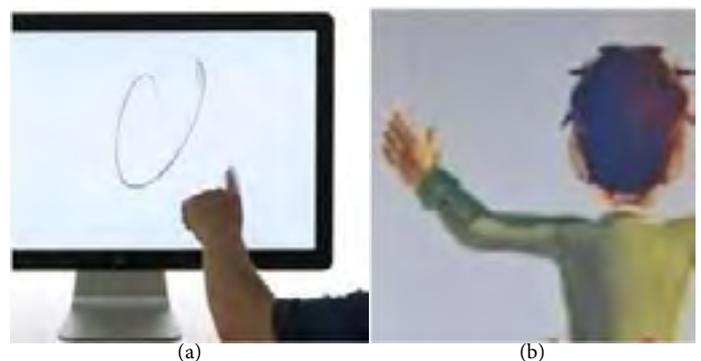


Figura 12. Simulando el movimiento del mouse (a) y (b) Clic izquierdo

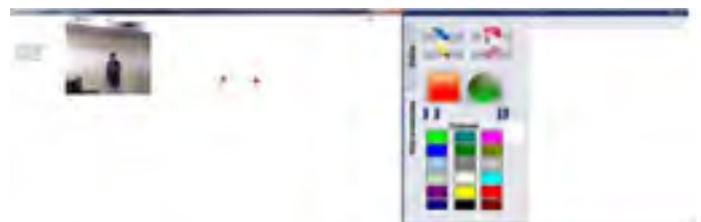


Figura 13. Manipulación de Paint

E. Control Media Center

Se realizó el control total de esta aplicación. Los controles que efectuados son: mover la mano derecha arriba-abajo y de derecha a izquierda para desplazarse entre las diferentes opciones del Media Center, para seleccionar una opción se coloca la mano a nivel de la cintura hacia el frente.



Figura 14. Movimiento de las manos arriba, abajo, izquierda, derecha (a) y (b) Selección opción



Figura 15. Desplazándose por Windows Media Center

6. Conclusión

En este trabajo se consiguió el control y monitoreo de un computador con el sensor Kinect. Los resultados cualitativos de los experimentos preliminares demostraron que mediante gestos del cuerpo se logró controlar Windows.

Referencias

- [1] Microsoft. Introducción Kinect para Xbox 360. <http://www.xbox.com/kinect>, 2012.
- [2] Kinect, <http://www.vidaextra.com/juegos/accesorios/kinect>
- [3] Usar las cámaras de sensor Kinect, <http://blogs.msdn.com/b/esmsdn/archive/2011/07/20/reto-kinect-usar-las-c-225-maras-del-sensor.aspx>
- [4] SDK de Kinect para Windows, [http://www.genbeta.com/herramientas/el-sdk-de-kinect-para-win-](http://www.genbeta.com/herramientas/el-sdk-de-kinect-para-win)

dows-listo-para-descargar

- [5] Kit de desarrollo de software. http://es.wikipedia.org/wiki/Kit_de_desarrollo_de_software, 2012
- [6] John Stowers, Michael Hayes and Andrew Bainbridge-Smith, "Altitude Control of a Quadrotor Helicopter Using Depth Map from Microsoft Kinect Sensor", Proceedings of the 2011 IEEE International Conference on Mechatronics, pp. 358-362, 2011, Turkey.
- [7] Patrick Benavidez, Mo Jamshidi, Lutch Brown, Endowed Chair, "Mobile Robot Navigation and Target Tracking System", Proc. Of the 2011 6th International Conference on System of Systems Engineering, app. 299-304, 2011
- [8] N. Ganganath and H. Leung, "Mobile robot localization using odometry and kinect sensor", Department of Electrical and Computer Engineering Schulich School of Engineering, app. 91-94, 2012
- [9] Valentino Frat, Domenico Prattichizzo, "Using Kinect for hand tracking and rendering in wearable haptics", app. 317-321, 2011
- [10] K. K. Biswas, Saurav Kumar Basu, "Gesture Recognition using Microsoft Kinect", Proceedings of the 5th International Conference on Automation, Robotics and Application, app. 100-103, 2011
- [11] Tarık Arıcı, İstanbul Şehir Üniversitesi, "Introduction to Programming with Kinect: Understanding Hand / Arm / Head Motion and Spoken Commands"
- [12] Empresa "Virtualware". <http://virtualwaregroup.com/unidades-de-negocio/educacion/educacion-con-kinect/>
- [13] Eadweard Muybridge. Muybridge's Complete human and animal locomotion: all 781 plates from the 1887 Animal locomotion / by Eadweard Muybridge ; introd. to the Dover edition by Anita Ventura Mozley. Dover Publications, New York, 1979.
- [14] Hui-mei Justina Hsu, "The Potential of Kinect as Interactive Educational Technology", pp. 334-338, 2011, Singapore
- [15] Digitalavmagazine, Microsoft presenta un prototipo de escritorio interactivo en 3D con pantalla transparente.