Interacción gestual con Kinect

Julio Centeno Bejarano y Warren Altamirano Carvajal

Escuela de Ingeniería,
Universidad Latinoamericana de Ciencia y Tecnología,
ULACIT, barrio Tournón, 10235-1000
San José, Costa Rica
waltamiranoc344,jcentenob941]@ulacit.ed.cr
http://www.ulacit.ac.cr

Resumen Desde la aparición de la primera computadora en los años cuarenta, la tecnología se ha reinventado de manera impresionante en varias ocasiones, para lograr mejores resultados, costos más bajos y menor tamaño en sus dispositivos. Desde el trabajo de Samuel Hurst, a quien se le atribuye el primer dispositivo táctil de la historia (Rodríguez, 2010), hasta las pantallas del iPhone y iPod, el ser humano ha seguido un camino de constante búsqueda por cambiar el paradigma de la interacción humano/computadora que, hasta antes de Hurst, era únicamente posible por medio de teclado y el *mouse*. La interacción gestual es en sí misma el siguiente gran paso hacia una interacción más natural con la tecnología, pero tiene un camino lleno de retos que enfrentar y usuarios que convencer. En este documento se expone brevemente qué es y cuáles son esos retos primarios que se deben superar antes de llevar esta tecnología al siguiente nivel.

Palabras clave: Interacción gestual

1. Introducción

La interacción gestual es como se le conoce a la incipiente tecnología que permite capturar movimientos del ser humano y convertirlos en instrucciones que una computadora pueda entender y a partir de ella realizar una determinada tarea, como desplazar la página o arrastrar un objeto en la pantalla. Algunas de las técnicas de interacción persona/computadora más comunes en el contexto de la analítica visual son el clic, doble-clic y arrastrar y soltar, y por lo general son llevadas a cabo mediante el uso del *mouse*. Sin embargo, en entornos en los cuales el usuario no tiene acceso al mouse, como en los paneles de pantalla[1] en centros de seguimiento de eventos, es deseable utilizar la interacción gestual como un método alternativo.

En el año 2006 y de la mano de Nintendo, llegó al mercado la consola de videojuegos conocida como WII, la cual traía consigo una barra de movimiento

[1] Los paneles de pantalla son conocidos en inglés como *screen panels*.

capaz de capturar los movimientos gruesos [2] del cuerpo humano mediante el uso de un control remoto llamado control de sensor de movimiento [3]. Si bien esto fue un gran paso hacia interacción gestual, aún seguía dependiendo de que el usuario del sistema portara un dispositivo en su mano para que la consola de videojuego captara sus movimientos. En el año 2010, Microsoft lanzó al mercado su consola de videojuegos XBOX 360, la cual incorporaba un sensor de movimiento llamado Kinect. Este nuevo hardware además de proveer una experiencia de juego novedosa, también viene acompañado de una librería de código disponible para aquellos que deseen hacer sus propias implementaciones de código utilizando el sensor Kinect para labores más empresariales como el facilitar la interacción gestual humano-ordenador.

En este contexto, esta investigación tiene por objetivo estudiar cómo se realiza la implementación de las técnicas de interacción persona/computadora más comunes utilizando interacción gestual. En consecuencia, este trabajo busca determinar al menos tres técnicas de interacción frecuentes que puedan ser implementadas con computación gestual y mostrar sus ventajas en relación con la interacción tradicional que requiere el uso del *mouse* y el teclado.

2. Estado del arte

Un gesto es comúnmente explicado como cualquier movimiento del cuerpo humano. Esta sencilla definición nos deja entrever que la detección e interpretación de la gesticulación humana es en sí, un área de estudio y entendimiento compleja, pues se dice que el ser humano realiza hasta un 60 % de sus comunicaciones de una manera no verbal (Gu, Do, Ou y Sheng, 2012).

A pesar de todos los avances en la materia, aún falta mucho por investigar y desarrollar en el tema de la interacción gestual visual. Actualmente, se ha logrado capturar y comprender de manera efectiva movimientos gruesos de los brazos o los de cabeza siempre y cuando estos sean amplios. Sin embargo, movimientos más finos, como los que se realizan por medio de los dedos, siguen siendo un reto para esta tecnología emergente.

En el mercado existen dos tipos de sensores para captar los gestos humanos: los basados en la visión y los basados en el movimiento. Los sensores basados en la visión son deseables, ya que no requieren que el usuario porte consigo ningún tipo de dispositivo para que se registren sus gestos; sin embargo, este tipo de sensores presentan cierta dificultad cuando se encuentran en entornos con determinada luminosidad. Por su parte, los sensores de movimiento sujetos al cuerpo tienen la ventaja de ser más tolerantes al entorno que rodea al usuario, pero el hecho de tenerlo sujeto al cuerpo es algo que le puede resultar incómodo al individuo. Respecto al reconocimiento de gesto, este se divide en dos partes: (1)la

[2] Los movimientos gruesos implican el movimiento de los brazos, las piernas, los pies o el cuerpo entero. Esto incluye acciones tales como correr, gatear, caminar, nadar y otras actividades que involucran los músculos más grandes.

[3] El control de sensor de movimiento es conocido en el contexto de los videojuegos como *motion sensor controller*

representación del gesto y (2)el entendimiento de este. Sin importar el dispositivo que se utilice para capturar el gesto, siempre será necesario el uso de alguna herramienta o guía para entender su trazo. Dentro del entendimiento del gesto, tenemos dos tipos: el estático y el dinámico. El gesto estático es aquel en que el individuo no se mueve y mantiene su postura a lo largo de la lectura. Este tipo de gestos son los más sencillos de entender, pues solo basta con comparar la imagen capturada para poder comprenderla. Por su parte, los gestos dinámicos poseen siempre un punto de origen, una trayectoria y un punto final que puede o no ser diferente de su punto de origen. Para reconocer los gestos dinámicos existen herramientas como *Hidden Markov Model (HMM)*(Rabiner y Juang, 1986), que se utiliza para modelar la trayectoria del cuerpo a fin de poder compararla posteriormente y así entender el gesto realizado.

Una vez que ya se ha logrado representar el gesto y entenderlo, es necesario comprender la intencionalidad del movimiento (Jang, Elmqvist y Ramani, 2014), que no es más que el estudio del motivo original de un determinado movimiento. Por ejemplo, la trayectoria del brazo puede ser la misma para indicarle al computador que desplace un documento hacia arriba, que para rascarse una ceja. Sin el adecuado entendimiento de la intención del movimiento, esta sencilla acción generaría que el documento se desplace de manera indeseada hacia arriba generando en consecuencia, la necesidad de que el usuario revierta dicho desplazamiento con otra acción, la cual, por ejemplo, podría coincidir nuevamente con otro movimiento tan natural como el frotarse la pierna y así se estaría nuevamente al inicio del problema.

Para poder entender mejor la intención del movimiento, existe una técnica llamada *Gesture Elicitation*[4], que emerge del campo del diseño participativo[5] y propone que para diseñar un buen gesto, este debe cumplir con algunos criterios esenciales como ser sencillo de descubrir, de realizar, de memorizar y ademas confiable (Morris y cols., 2014).

Además de los retos que la implementación de esta tecnología tiene, es importante mencionar que el principal objetivo debe ser poder realizar acciones en el computador sin la necesidad de cables conectados a dispositivos como un guante o dispositivos de entrada más comunes, como el *mouse* y el teclado. Así mismo, la interacción gestual debe soportar la facilidad de movimiento por parte del individuo, al tiempo que facilita la sensación de inmersión cuando sea necesario, como en los ambientes de realidad virtual entre otros (Rodríguez, 2010).

Para el propósito de este documento, se estudiarán los movimientos gruesos de brazos mediante la tecnología de Microsoft Kinect versión 2. Además, se desarrolló un pequeño código informático basado en Kinect para demostrar cómo es la interacción del ser humano con la computadora por medio de la interacción gestual.

[4] Gesture Elicitation se refiere a la técnica para el entendimiento de la intención del movimiento.

[5] Diseño participativo se refiere a la inclusión del usuario final para así poder definir cuál y cómo será el gesto

3. Desarrollo

Para la realización de este ejemplo se cuenta con la siguiente configuración de *software* y *hardware* en observancia con los requerimientos mínimos dados por Microsoft (Microsoft, 2017b).

3.1. Consideraciones de *hardware*

A continuación se detalla el *hardware* mínimo y el utilizado durante esta investigación:

- Memoria RAM mínimo de 4GB o superior.
- Procesador I7 de 3.1 GHz.
- USB 3.0 que sea de controlador Intel únicamente.
- Tarjeta gráfica con tecnología DX11: Intel HD 4400 integrated display adapter, ATI Radeon HD 5400 series, ATI Radeon HD 6570, ATI Radeon HD 7800 (256-bit GDDR5 2GB/1000Mhz), NVidia Quadro 600, NVidia GeForce GT 640, NVidia GeForce GTX 660, NVidia Quadro K1000M.
- Sensor Kinect V2 con el adaptador que contiene el cable USB y fuente de poder.

Es importante notar que la configuración anterior únicamente funciona con Chipset Intel. Si el computador utiliza Chipset AMD, el Kinect y el SDK no se conectaran correctamente con su computador. Respecto a la tarjeta gráfica, esta sí puede ser de cualquier marca siempre y cuando cumpla con las especificaciones arriba mencionadas.

3.2. Software

El software utilizado en este trabajo se detalla a continuación:

- Visual Studio 2017 como entorno de desarrollo. Mínimo recomendado Visual Studio 2012.
- Microsoft Windows 10 con arquitectura de 64bits como sistema operativo. Mínimo recomendado Windows 8.
- C-# como lenguaje de programación con interfaz gráfica XAML (Nathan, 2014).

En la figura 1 se observa una captura del validador que trae consigo el entorno de desarrollo. Este debe ser utilizado para eliminar la posibilidad de cualquier descuido en el momento de completar los requisitos.

Tras realizar una revisión bibliográfica previa al inicio de este documento, se decidió utilizar Kinect versión 2, apoyada en la referencia de código javascript. En esta línea, se trabajó con la generación y almacenamiento de gestos, los cuales se pretendía utilizar para acciones sencillas como hacer clic, doble clic y arrastrar y soltar; sin embargo, surgió el problema de que Microsoft retiró las referencias a javascript luego de la versión de Kinect 1.8, sin que esto fuera

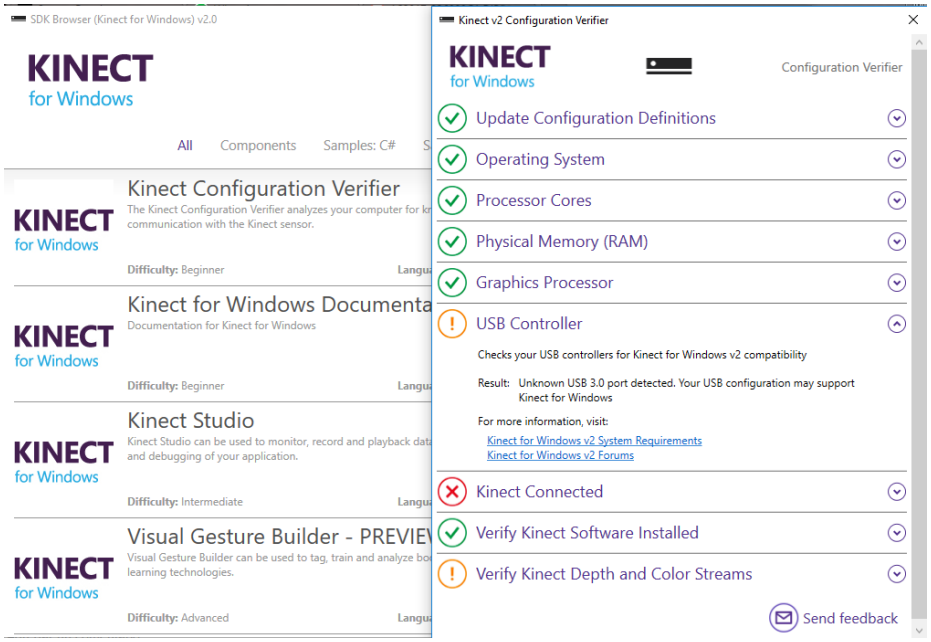

Figura 1: Validador de Kinect SDK

adecuadamente informado a la comunidad de informática. Ahondando más en este hallazgo, se encontró que en las versiones 1.x de kinect (1.5, 1.6, 1.7 y 1.8) se incluyen referencias a *C++, Managed Code y JavaScript*(Jana, 2012). No obstante, para la versión 2.0 de Kinect, Microsoft removió la referencia para javascript y con esta, la librería *WebServerBasics-WPF Sample* comúnmente utilizada para realizar implementaciones de interacción gestual usando Kinect (Microsoft, 2017a). Esto fue descubierto gracias a que la presente investigación se encontraba en una etapa comparativa sobre los beneficios de utilizar JavaScript o C#. Si bien Microsoft deliberadamente removió la referencia citada, es por medio de su foro oficial de *Microsoft Developer Network* [6] que se encontró gran número de usuarios reportando problemas para implementar sus soluciones en Kinect versión 2, utilizando JavaScript (Catuhe, 2012).

Con el objetivo de continuar con la investigación, se decidió cambiar al lenguaje de programación C# junto con archivos de extensión XAML para la interfaz gráfica. A nivel de librerías, se utilizó únicamente la librería Kinect tal y como se muestra en la figura 2, la cual está disponible a partir de la instalación del SDK de Kinect.

La librería Kinect incluye el código para interpretar los movimientos más comunes, como el trazo de extremidades; y movimientos gruesos del tronco del cuerpo, como el ponerse de pie. Una vez llamado el evento correcto para que

[6] Microsoft Developer Network conocido en el mundo informático como MSDN es el foro oficial de microsoft para programadores de su plataforma .Net

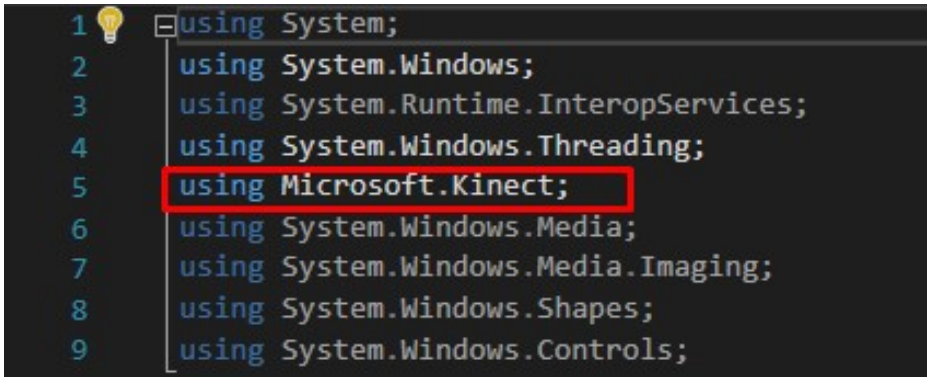

Figura 2: Libreria de Kinect

este capture el movimiento, se debe programar la acción computada que se desea llevar a cabo a partir de dicho movimiento. Durante las pruebas realizadas, se pudo reconocer el desplazamiento de los brazos así como la acción de cerrar y abrir el puño. Con esto se programaron las acciones de clic derecho, clic izquierdo y arrastrar y soltar. A su vez, se notó que la luz de fondo al individuo que realiza la acción dificulta el reconocimiento de los movimientos, por lo que se debe considerar como un factor primario para el éxito en la interacción gestual. Entre otros aspectos, se deben considerar los siguientes:

- Que esta aún no puede darse en ambientes con mucha luz natural.
- Que en entornos con presencia de otras personas dentro del rango de lectura del sensor, como por ejemplo un centro de monitoreo, la detección del movimiento se ve afectada, pues si bien Kinect es capaz de recordar el individuo que fue reconocido primero, sí se observa una disminución en la velocidad de lectura en el momento de detectar a una segunda persona.
- Que el sensor Kinect no debe estar a menos de 1.5 metros de distancia del individuo que realiza la gesticulación.

Kinect proporciona la posibilidad de discriminar la porción del cuerpo que lee en busca de movimientos. De esta forma es posible hacer que el sensor se enfoque solo en los brazos, en los pies o en el cuerpo entero para capturar movimientos más completos que involucren a todo el individuo.

En este documento se ha desarrollado un ejemplo de interacción gestual utilizando Kinect versión 2 y código informático en C-#, que incluye los gestos más comunes de manera nativa en su biblioteca, lo cual simplifica enormemente la codificación de interacciones gestuales. Esta característica es una enorme limitación para esta tecnología, pues impide que los usuarios creen sus propios gestos, aquellos que les son sencillos de descubrir, de realizar y de memorizar, y que son además confiables.

4. Resultados

Para iniciar con la etapa de pruebas, se diseñó una pantalla de inicio que permitiera ajustar de manera sencilla algunas variables como la sensibilidad del *mouse*, tiempo máximo en pausa y la suavidad de movimiento del cursor. En la figura 3 se observan los controles básicos para la interacción gestual de esta investigación.

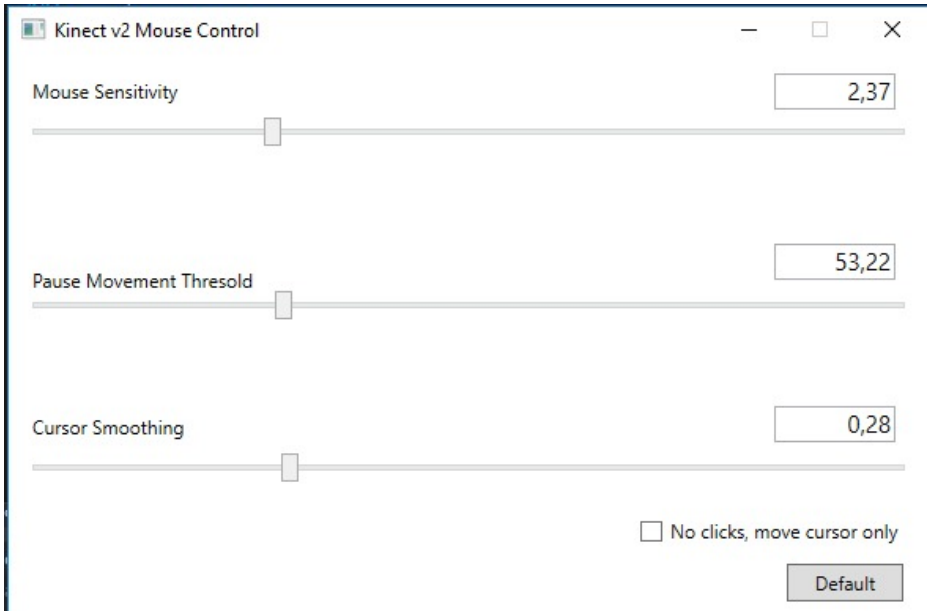

Figura 3: Pantalla principal

El sensor inicia detectando el cuerpo completo del individuo tal y como se muestra en la figura 4, y a partir de ese momento detectará los movimientos de las extremidades para las cuales fue programado. En nuestro caso, extremidades superiores.

Una vez detectado el cuerpo humano, se realizaron pruebas básicas de movimiento, apertura y cierre de la palma de la mano, las cuales fueron exitosas según se puede visualizar en la 5

Tras avanzar con las pruebas, se pudieron constatar algunas limitaciones sencillas de inferir, como por ejemplo que la luz natural en la habitación en donde se realizan los gestos afecta significativamente el funcionamiento del sensor Kinect. Un excesivo movimiento del puntero en la pantalla cuando la mano del individuo está en una posición fija dificultó la capacidad de dar clic derecho y clic izquierdo, pero no se dio ningún problema con el gesto de arrastrar y soltar.

Figura 4: Detección de cuerpo completo

El mismo comportamiento errático fue detectado al tener al individuo frente a un proyector [7], cuyo destello de luz afectó el funcionamiento del sensor.

Otras pruebas que se realizaron fueron el colocar al individuo al lado de otro objeto móvil (como un ventilador) y poner a dos individuos dentro del rango de detección del sensor para probar su capacidad de discriminación y entendimiento de gestos que le competen.

En este caso, el sensor fue capaz de mantener el foco sobre el individuo primeramente detectado (en color rojo), pero sí se observaron algunas pérdidas del foco hacia el segundo individuo en su rango de detección. Esto no significó en ningún momento una pérdida de capacidad de detección de los gestos; sin embargo, sí ralentizó el proceso.

Finalmente, a pesar de que el *hardware* utilizado sobrepasa levemente los requerimientos mínimos según Microsoft, en repetidas ocasiones fue necesario reiniciar el sensor debido a que este se çongeló"[8].

[7] Comúnmente conocido como *video beam*

[8] En el entorno tecnológico, decir que un equipo se *congeló* se refiere a que este deja de responder a cualquier interacción dada por el usuario.

Figura 5: Detección del movimiento, apertura y cierre de la mano

En el siguiente enlace se puede encontrar un video demostrativo sobre cómo funciona la interacción gestual a partir del uso de Kinect versión 2 con código C#: https://goo.gl/ByqVC2.

5. Conclusiones

La tecnología de computación gestual tiene gran potencial para ser utilizada en diferentes tipos de aplicaciones, pues permite que el individuo pueda realizar operaciones con libertad de movimiento. Sin embargo, el desarrollo de este trabajo de investigación permitió determinar que Kinect v.2 no cuenta con la madurez y confiabilidad suficiente para realizar funciones en un entorno real, que requieran un nivel intermedio/alto de precisión. Las pruebas realizadas permitieron determinar que es muy sensible a la luz y que tiene dificultad para captar movimientos finos del cuerpo humano. Además, cuenta con un rango de detección muy corto, que puede afectar la realización de determinadas tareas y además dificultar la interacción humano/computadora. A esto se debe agregar que el fabricante no tiene definidos de forma clara los ciclos de vida del produc-

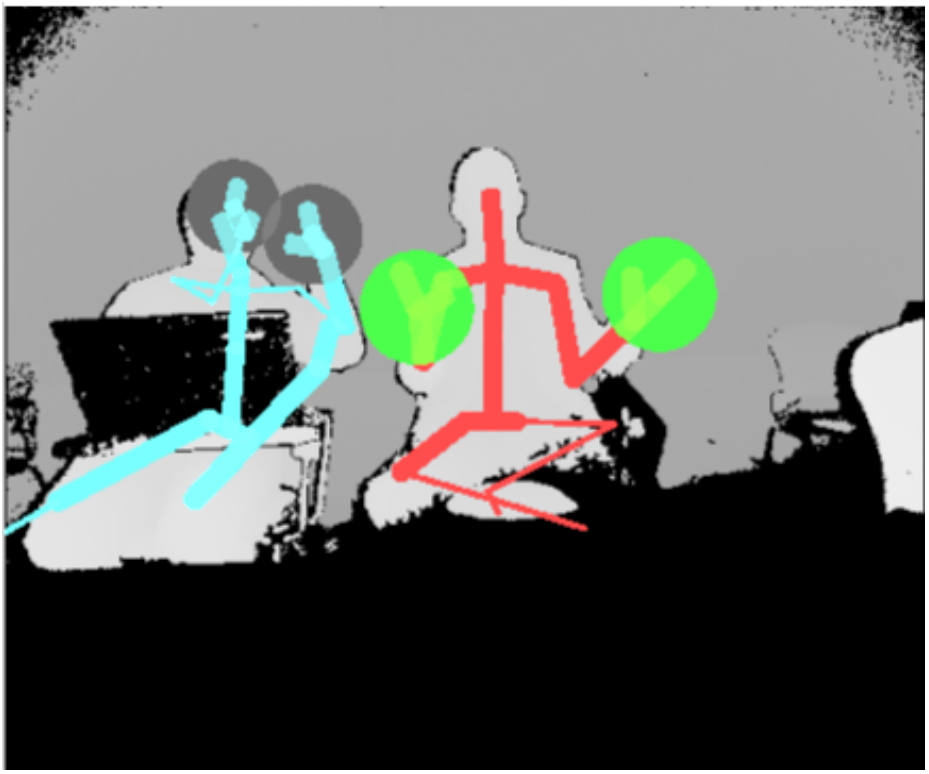

Figura 6: Prueba de de detección de múltiples individuos de manera simultánea

to para permitir que los desarrolladores y usuarios de esta tecnología puedan planificar sus proyectos con suficiente antelación.

Como trabajo futuro se recomienda investigar sobre otras tecnologías como Leap Motion[9] y RealSense[10], las cuales cuentan con capacidad de detección de gestos y herramientas para el desarrollo de aplicaciones. Por lo que sería conveniente someterlas a pruebas para determinar su capacidad de adaptación a problemas del mundo real.

Referencias

Catuhe, D. (2012). *Programming with the Kinect for Windows software development kit*. Pearson Education.

[9] El sitio web oficial de Leap Motion se encuentra en la siguiente URL: `https://www.leapmotion.com`

[10] El sitio oficial de RealSense se encuentra en la siguiente dirección: `http://www.intel.com/content/www/us/en/architecture-and-technology/realsense-overview.html`

Gu, Y., Do, H., Ou, Y. y Sheng, W. (2012). Human gesture recognition through a kinect sensor. En *Robotics and Biomimetics (ROBIO), 2012 IEEE International Conference on* (pp. 1379–1384).

Jana, A. (2012). *Kinect for windows SDK programming guide*. Packt Publishing Ltd. Descargado de `http://amzn.to/2fiqRZH`

Jang, S., Elmqvist, N. y Ramani, K. (2014). GestureAnalyzer: visual analytics for pattern analysis of mid-air hand gestures. En *Proceedings of the 2nd ACM Symposium on Spatial User Interaction* (pp. 30–39). New York, NY, USA: ACM.

Microsoft. (2017a). *Reference: The Kinect for Windows SDK 2.0 provides the following reference documentation.* Descargado de `http://bit.ly/2vCD1DE`

Microsoft. (2017b). *Set up Kinect for Windows v2 or an Xbox Kinect sensor with Kinect Adapter for Windows.* Descargado de `http://bit.ly/2ueuSVx`

Morris, M. R., Danielescu, A., Drucker, S., Fisher, D., Lee, B., Schraefel, M. y Wobbrock, J. O. (2014, mayo). Reducing legacy bias in gesture elicitation studies. *Interactions*, *21*(3), 40–45.

Nathan, A. (2014). *XAML Unleashed* (1st ed.).

Rabiner, L., y Juang, B. (1986). An introduction to hidden markov models. *IEEE ASSP Magazine*, *3*(1), 4–16. Descargado de `http://bit.ly/2wv8lAu`

Rodríguez, A. (2010). El gesto como mecanismo de interacción corporizada con computadoras: posibilidades y desafios. En Laura Inés Fillottrani y Adalberto Patricio Mansilla (Ed.), *Tradición y Diversidad en los aspectos psicológicos, socioculturales y musicológicos de la formación musical. Actas de la IX Reunión de SACCoM* (p. 116-120). Sociedad Argentina para las Ciencias Cognitivas de la Música.