Universidad de San Carlos de Guatemala
Facultad de Ingeniería
Escuela de Ingeniería Mecánica Eléctrica

GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT

Pedro Pablo Martínez Valenzuela
Asesorado por el Ing. César Augusto Menchú Tumax

Guatemala, septiembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT

TRABAJO DE GRADUACIÓN

PRESENTADO A LA JUNTA DIRECTIVA DE LA
FACULTAD DE INGENIERÍA
POR

PEDRO PABLO MARTÍNEZ VALENZUELA
ASESORADO POR EL ING. CÉSAR AUGUSTO MENCHÚ TUMAX

AL CONFERÍRSELE EL TÍTULO DE

INGENIERO EN ELECTRÓNICA

GUATEMALA, SEPTIEMBRE 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA

FACULTAD DE INGENIERÍA

NÓMINA DE JUNTA DIRECTIVA

DECANO	Ing. Pedro Antonio Aguilar Polanco
VOCAL I	Ing. Angel Roberto Sic García
VOCAL II	Ing. Pablo Christian de León Rodríguez
VOCAL III	Ing. José Milton de León Bran
VOCAL IV	Br. Jurgen Andoni Ramírez Ramírez
VOCAL V	Br. Oscar Humberto Galicia Nuñez
SECRETARIA	Inga. Lesbia Magalí Herrera López

TRIBUNAL QUE PRACTICÓ EL EXAMEN GENERAL PRIVADO

DECANO	Ing. Pedro Antonio Aguilar Polanco
EXAMINADOR	Ing. Julio César Solares Peñate
EXAMINADOR	Ing. Natanael Jonathan Requena Gómez
EXAMINADOR	Ing. Kenneth Issur Estrada Ruiz
SECRETARIA	Inga. Lesbia Magalí Herrera López

HONORABLE TRIBUNAL EXAMINADOR

En cumplimiento con los preceptos que establece la ley de la Universidad de San Carlos de Guatemala, presento a su consideración mi trabajo de graduación titulado:

GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT

Tema que me fuera asignado por la Dirección de la Escuela de Ingeniería Mecánica Eléctrica, con fecha 24 de octubre de 2016.

Pedro Pablo Martínez Valenzuela

UNIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTA DE INGENIERIA

Guatemala, 14 de julio de 2017

Ing. Julio César Solares Peñate
Coordinador de Área de Electrónica
Escuela de Ingeniería Mecánica Eléctrica
Facultad de Ingeniería, USAC.

Ingeniero Solares:

Por este medio me permito dar aprobación al Trabajo de Graduación titulado: **"GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT"**, desarrollado por el estudiante Pedro Pablo Martínez Valenzuela, ya que considero que cumple con los requisitos establecidos.

Por tanto, el autor de este trabajo y yo como asesor, nos hacemos responsables del contenido y conclusiones del mismo.

Sin otro particular, aprovecho la oportunidad para saludarlo.

Atentamente,

"ID Y ENSEÑAD A TODOS"

CESAR AUGUSTO MENCHU TUMAX
ING. ELECTRONICO
COLEGIADO No. 13363

Ing. César Augusto Menchú Tumax
Colegiado No: 13363
Asesor

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

Guatemala, 6 de SEPTIEMBRE 2017.

Señor Director
Ing. Otto Fernando Andrino González
Escuela de Ingenierìa Mecànica Elèctrica
Facultad de Ingenierìa, USAC.

Señor Director:

Me permito dar aprobación al trabajo de Graduación titulado: GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT, del estudiante Pedro Pablo Martínez Valenzuela, que cumple con los requisitos establecidos para tal fin.

Sin otro particular, aprovecho la oportunidad para saludarle.

Atentamente,
ID Y ENSEÑAD A TODOS

Ing. Julio César Solares Peñate
Coordinador de Electrónica

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

NIVERSIDAD DE SAN CARLOS
DE GUATEMALA

FACULTAD DE INGENIERIA

REF. EIME 51. .2017.

El Director de la Escuela de Ingeniería Mecánica Eléctrica, después de conocer el dictamen del Asesor, con el Visto Bueno del Coordinador de Área, al trabajo de Graduación del estudiante; PEDRO PABLO MARTÍNEZ VALENZUELA titulado: GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT , procede a la autorización del mismo.

Ing. Otto Fernando Andrino González

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DIRECCION ESCUELA DE INGENIERIA MECANICA ELECTRICA
FACULTAD DE INGENIERIA

GUATEMALA, 26 DE SEPTIEMBRE 2,017.

Universidad de San Carlos
de Guatemala

Facultad de Ingeniería
Decanato

DTG. 445.2017

El Decano de la Facultad de Ingeniería de la Universidad de San Carlos de Guatemala, luego de conocer la aprobación por parte del Director de la Escuela de Ingeniería Mecánica Eléctrica, al Trabajo de Graduación titulado: **GUÍA DEL FUNCIONAMIENTO DE TECNOLOGÍAS DE RECONOCIMIENTO DE KINECT,** presentado por el estudiante universitario: **Pedro Pablo Martínez Valenzuela,** y después de haber culminado las revisiones previas bajo la responsabilidad de las instancias correspondientes, autoriza la impresión del mismo.

IMPRÍMASE:

Ing. Pedro Antonio Aguilar Polanco
Decano

Guatemala, septiembre de 2017

UNIVERSIDAD DE SAN CARLOS DE GUATEMALA
DECANO
FACULTAD DE INGENIERIA

/gdech

ACTO QUE DEDICO A:

Dios	Por ser una importante influencia en mi carrera, entre otras cosas.
Mis padres	Pedro Obdulio Martínez (q. e. p. d.) y María Hortencia Valenzuela Galicia. Por su amor que siempre será mi inspiración.
Mis hermanos	Elda Maritza y Elmer Aníbal Martínez Valenzuela. Dos personas muy importantes en mi vida.
Mis sobrinos	Yosseline Andrea y Pedro Andrés Hernández Martínez. Por ser dos ángeles en mi vida.
Mi cuñado	Jorge Aníbal Hernández Bran. Por siempre apoyarme en mi superación personal y profesional
Mi abuela	María Chávez. Por siempre apoyarme en todo momento y darme, en todo momento, sabias enseñanzas.
Mi amigo	Nicolae Goga. Por apoyarme en todo momento cuando estuve en Rumania.

AGRADECIMIENTOS A:

Universidad de San Carlos de Guatemala	Por ser el pilar de mi educación superior, en ingeniería y aprendizaje de idiomas.
Proyecto Erasmus Mundus EURICA Action 2	Por haberme seleccionado en la primera convocatoria de proyecto.
Universidad Politécnica de Bucarest	Por sus enseñanzas de ingeniería en inglés con método rumano europeo.
Ing. Carlos Guzmán	Por su apoyo incondicional en todo momento y ser una persona que me guió cuando lo necesite.
Ing. Francisco González	Por su apoyo incondicional y confianza en todo momento.
Ing. César Menchú	Por su apoyo y asesoría para la elaboración de este trabajo de graduación.

ÍNDICE GENERAL

ÍNDICE DE ILUSTRACIONES

FIGURAS

TABLAS

LISTA DE SÍMBOLOS

Símbolo	Significado
API	Interfaz de programación de aplicaciones
CMOS	Semiconductor complementario de óxido metálico
C++	Incremento de C
dB	Decibelio
DLL	Biblioteca de enlace dinámico
FPS	Fotogramas por segundo
GB	*Gigabyte*
GHZ	Gigahercio
GPU	Unidad de procesamiento gráfico
KHZ	Kilohercio
PC	Computadora personal
RAM	Memoria de acceso aleatorio
SoC	Sistema en *chip*
TOF	Tiempo de vuelo
VGA	Adaptador gráfico de video
XML	Lenguaje de marcas extensible

GLOSARIO

Bit Dígito del sistema de numeración binario. La capacidad de almacenamiento de una memoria digital también se mide en bits, pues esta palabra tiene varias acepciones.

CPU Es el hardware dentro de un ordenador u otros dispositivos programables, que interpreta las instrucciones de un programa informático mediante la realización de las operaciones básicas aritméticas, lógicas y de entrada/salida del sistema.

C Sharp Lenguaje de programación orientado a objetos desarrollado y estandarizado por Microsoft como parte de su plataforma .NET,

DirectX Colección de API desarrolladas para facilitar las complejas tareas relacionadas con multimedia, especialmente programación de juegos, en la plataforma Microsoft Windows.

Intel Core i7 Familia de procesadores 4 núcleos de la arquitectura Intel x86-64, lanzados al comercio en 2008.

led	Es una fuente de luz constituida por un material semiconductor dotado de dos terminales. Se trata de un diodo de unión p-n, que emite luz cuando está activado.
.Net Framework	Es un *framework* de Microsoft que enfatiza en la transparencia de redes, con independencia de plataforma de hardware y que permita un rápido desarrollo de aplicaciones. Basado en ella, la empresa intenta desarrollar una estrategia horizontal que integre todos sus productos, desde el sistema operativo hasta las herramientas de mercado.
Píxel	Es la menor unidad homogénea en color que forma parte de una imagen digital.
USB	Es un bus estándar industrial que define los cables, conectores y protocolos usados en un bus para conectar, comunicar y proveer de alimentación eléctrica entre computadoras, periféricos y dispositivos electrónicos.
Visual Studio	Entorno de desarrollo integrado para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET MVC, Django, etc., a lo cual sumarle las nuevas

	capacidades *online* bajo Windows Azure en forma del editor Monaco.
Xbox	Primera videoconsola de sobremesa producida por Microsoft, en colaboración con Intel.
X86	x86-64 (también conocido como x64, x86_64 y AMD64) es la versión de 64 bits del conjunto de instrucciones x86.

RESUMEN

Desde el principio de la electrónica, para la utilización de un dispositivo electrónico, siempre ha sido necesario que se realice un contacto físico con el dispositivo para que funcione. En algunas situaciones puede resultar muy complicado por la complejidad del dispositivo o dificultad de acceso para su utilización.

Recientemente, nuevas tecnologías han creado dispositivos que no necesitan interacción física para su utilización. Estos dispositivos proporcionan herramientas para desarrollar aplicaciones que utilizan reconocimiento de voz y gestos para su utilización.

Con los nuevos avances en dispositivos que utilizan reconocimiento de voz y gestos, se realizará una guía sobre las tecnologías de reconocimiento que utiliza Kinect, el Kinect provee grandes posibilidades para el desarrollo de cualquier tipo de aplicaciones ya que utiliza estas tecnologías.

El Kinect cuenta con una cámara de color, una cámara de profundidad, un proyector infrarrojo, un conjunto de micrófonos, luz LED, un acelerómetro de tres ejes y en especial un procesador especializado que capta el entorno en 3 dimensiones.

Para el desarrollo de cualquier aplicación se debe utilizar el SDK de Kinect, en el cual se incluyen todos los códigos y librerías necesarias para

desarrollar aplicaciones con el Kinect. Estas aplicaciones se desarrollan con el entorno de programación Microsoft Visual Studio.

OBJETIVOS

General

Analizar el hardware y software de Kinect, para implementar una guía donde se detallen todos los aspectos relacionados con su funcionamiento, cuando está realizando el reconocimiento de voz, facial y gestos.

Específicos

1. Analizar el hardware que compone el Kinect.

2. Analizar el procesamiento de voz, imagen y gestos de Kinect.

3. Detallar los conceptos básicos para la programación de Kinect con el entorno de programación Visual Studio.

INTRODUCCIÓN

Kinect, es un dispositivo que permite la interacción con el cuerpo humano mediante una interfaz que capta y reconoce gestos humanos y comandos de voz. Se presentó por primera vez en junio del 2009 con el nombre de Project Natal. El nombre oficial Kinect se conoció el 13 de junio del 2010. En julio del 2015 se lanzó una versión mejorada del Kinect, compatible también para PC. El aumento de aplicaciones para PC hizo que se desarrollara una versión de Kinect para sistemas operativos X86 para su uso en PC con comunicación directa a entornos de programación para computadora.

Hace pocos años las computadoras obtenían información muy básica del mundo que los rodeaba. Las personas tenían básicamente 3 formas muy rudimentarias de comunicarse con las computadoras, las cuales eran, el teclado, mouse o pantallas táctiles. Con el pasar de los años han adquirido cámaras y entradas de audio; las computadoras pueden almacenar y reproducir dicho contenido, pero ha sido muy difícil hacer que las computadoras comprendan la información que se les está proporcionando, y no solo se limiten a guardarla y reproducirla cuando se necesite.

Por ejemplo, cuando una persona escucha un sonido, puede hacer una interpretación sobre la distancia, la dirección e interpretar su significado de la fuente de sonido en relación con su posición; también cuando una persona determina un objeto con su vista, puede analizarlo e interpretarlo a partir de su estructura y colores. Hace pocos años, las computadoras no tenían la

capacidad de analizar e interpretar la información sobre el mundo que las rodeaba.

La información de audio a partir de una serie de micrófonos proporciona considerable información acerca de la distancia y la dirección de la fuente de audio, pero determinar qué sonidos capturaba o determinar la voz de una persona específica era imposible hace pocos años. Del mismo modo, una fotografía digital proporciona una imagen del entorno de la computadora para analizar, pero una computadora con una cámara simple era imposible extraer información de tal fotografía digital.

Con la disponibilidad de dispositivos que captura un mundo tridimensional, como el caso del Kinect cambia todo esto. El Kinect contiene cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos. Internamente incluye el hardware del Kinect, hardware de procesamiento de señal que captura toda la información proporcionada por los dispositivos anteriormente descritos.

En resumen, el Kinect permite a los usuarios controlar e interactuar sin necesidad de tener contacto físico con el Kinect. Esto es posible mediante una interfaz natural de usuario que reconoce la voz y gestos. En diciembre del 2010 se lanzó el primer SDK no oficial para Kinect. Gracias al SDK no oficial de Kinect, numerosos, desarrolladores, investigadores, estudiantes, instituciones y aficionados, comenzaron la investigación y programación de diversas aplicaciones aprovechando todas las características que brinda Kinect más allá de la interacción para entretenimiento.

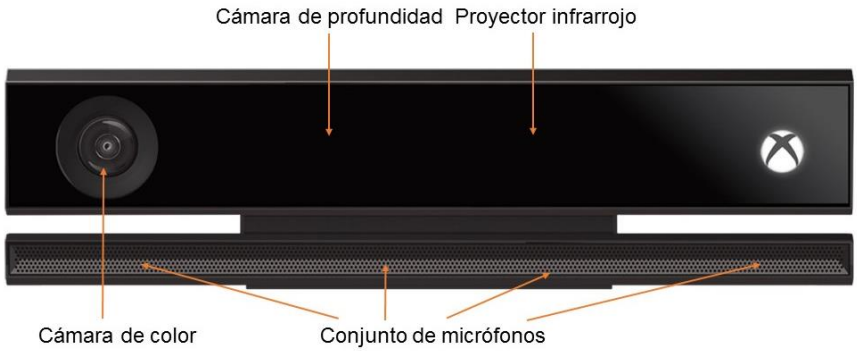
.

1. KINECT

1.1. Introducción a Kinect

El Kinect posee un gran campo de visión en forma horizontal y vertical para la cámara de color y cámara de profundidad, incluye un zoom de 3x, la cámara de color es de alta definición con resolución de 1 920 por 1 080 pixeles a 15 y 30 FPS, la cámara de profundidad con resolución de 512 por 424 pixeles a 30 FPS y un conjunto de micrófonos. El hardware principal de Kinect se describe a continuación para observar cómo se encuentra distribuida la cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos. La figura 1 muestra un Kinect cuando está con su cubierta de plástico frontal.

Figura 1. **Kinect**

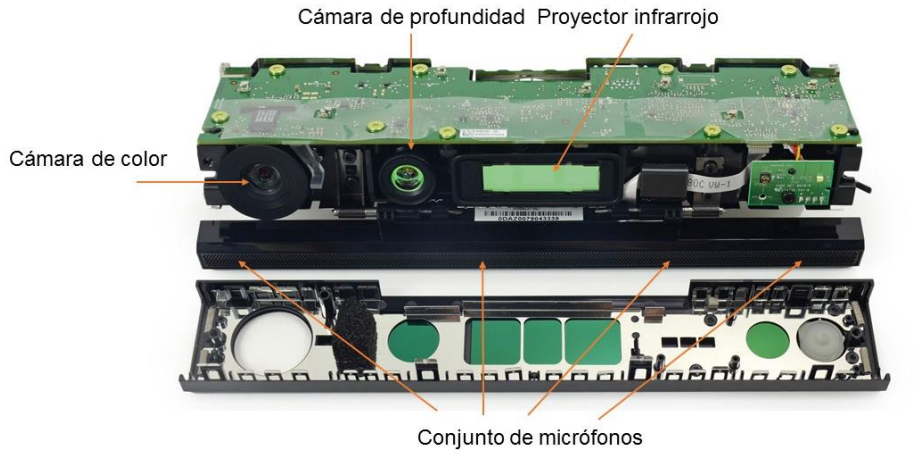


Fuente: http://www.xbox.com/en-US/xbox-one/accessories/kinect. Consulta: noviembre de 2016.

La figura 2 muestra el Kinect sin la cubierta. Se puede ver la cámara de color en el lado izquierdo, la cámara de profundidad y proyector infrarrojo se

encuentra en el centro del Kinect que, en conjunto, proporcionan una vista de reconocimiento al Kinect sobre el mundo que lo rodea en tiempo real. Por último, el conjunto de micrófonos está colocado a lo largo de la parte inferior. Se pueden observar todos los dispositivos que realizan la función principal del Kinect y que, gracias a ellos se puede capturar toda la información del ambiente que lo rodean.

Figura 2. **Kinect sin cubierta**



Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

1.2. Cámara de color

La cámara de color de Kinect está formada por un sensor de imagen CMOS con la que captura la información a color. Esta puede capturar imágenes a color con una resolución de 1 920 por 1 080 pixeles, por lo cual se puede decir también que tiene resolución Full HD a 15 y 30 FPS.

Figura 3. **Cámara de color de Kinect**

Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

1.3. Cámara de profundidad

El Kinect tiene la capacidad única de capturar datos de profundidad. A diferencia de la mayoría de los otros sistemas de visión por computadora, el Kinect es capaz de construir una imagen de profundidad sobre lo que está por delante de él. Esta imagen se toma, utilizando la cámara de profundidad y proyector infrarrojo, esta se transmite de la misma manera que unas imágenes a color de una cámara típica serían transferidos, excepto que, en lugar de la información de color de cada píxel de una imagen, se transmiten los valores de distancia, los cuales son denominados pixeles de distancia.

Figura 4. **Cámara de profundidad de Kinect**

Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

Se podría pensar que la cámara de profundidad utiliza algún tipo de radar o transmisor de sonido ultrasónico para medir la distancia del Kinect hasta un determinado objeto, pero en realidad, no es así. Sería difícil medir una distancia corta, en cambio, el Kinect utiliza una técnica inteligente que consiste en un proyector de infrarrojos y una cámara que puede ver los pequeños puntos que produce el proyector infrarrojo. Separados pueden parecer sencillos, pero la cámara de profundidad es definitivamente diferente si se compara con una cámara de color estándar.

1.4. Proyector Infrarrojo

El proyector infrarrojo de Kinect es el encargado de proyectar haces de luz infrarrojos hacia todo el espacio físico que se encuentra enfrente de Kinect, para que posteriormente la cámara de profundidad analice cada punto que sale del proyector infrarrojo. El proyector infrarrojo de Kinect es un conjunto de tres proyectores de luz infrarroja como se observa en la figura 5.

Figura 5. **Proyecto infrarrojo de Kinect**

Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

1.5. Conjunto de micrófonos

Este es el conjunto de micrófonos de Kinect con el cual se realiza el reconocimiento de voz. El conjunto de micrófonos posee cuatro micrófonos, en una configuración de tres más uno. En la placa hay dos convertidores analógico digital de tipo estéreo. Esta forma es la mejor para obtener la voz del usuario desde determinada distancia. El ruido del ambiente es cancelado por la unidad de procesamiento y por medio de software se calcula dónde proviene el sonido, y así crear un entorno cerrado de sonido alrededor, para el usuario. De esta manera, se separa el sonido de voz del usuario principal y se hace caso omiso a las otras personas que se encuentren alrededor de él.

Figura 6. **Conjunto de micrófonos de Kinect**

Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

Es de suponer que hay un complejo tratamiento de anulación de sonido que se realiza cuando se captura el sonido con los cuatro micrófonos antes de ser enviado a la computadora para su interpretación. Cada canal procesa 16 bits de información en un rango de muestreo de 16 kHz. Estos micrófonos son la única razón por la cual el sensor es tan ancho.

1.6. Microprocesador X871141-001

El microprocesador X871141-001 es el encargado de procesar toda la información que recibe de la cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos del Kinect. El microprocesador X871141-001, sustituye al microprocesador de la primera versión de Kinect que fue desarrollado para la consola Xbox 360. Este tiene un empaquetado tipo SoC, para su montaje en la placa sin ocupar mucho espacio en la misma.

Figura 7. **Microprocesador X871141-001**

Fuente: https://es.ifixit.com/Teardown/Xbox+One+Kinect+Teardown/19725. Consulta: noviembre de 2016.

1.7. Versiones de Kinect

1.7.1. Kinect v1

La primera versión de Kinect con nombre Project Natal fue presentado en junio del 2009 como un dispositivo en forma de barra. Usaba un proyector infrarrojo y una cámara que escaneaban la escena y enviaban la información a un microprocesador preparado para capturar imágenes en tres dimensiones a los objetos y personas en movimiento. Adicionalmente, tenía un conjunto de micrófonos capaz de reconocer la voz del usuario. Para tales tareas, las características de Kinect no suponían nada extraordinario. La cámara de color

de tipo VGA tenía una resolución que capturaba información 640 por 480 pixeles por defecto, aunque era capaz de trabajar a 1 280 por 1 024 píxeles a costa de una menor tasa de transferencia. El microprocesador incluido, apenas realizaba parte del trabajo de procesar la información, dejaba buena parte de la tarea a la propia computadora. Una de las claves de todo el sistema residía en el software patentado para interpretar toda la información recogida por todos los dispositivos de Kinect.

Figura 8. **Kinect v1**

Fuente: http://www.xbox.com/en-US/xbox-360/accessories/kinect. Consulta: noviembre de 2016.

El Kinect utilizaba, adicionalmente, un motor que inclinaba el Kinect de esta manera daba mayor rango vertical. Esta última característica fue eliminada en la versión siguiente de Kinect.

1.7.2. Kinect v2

La gran diferencia del nuevo Kinect respecto a su predecesor reside en la nueva cámara de color. La segunda generación del Kinect, incorpora una

cámara TOF de alta resolución que permite al Kinect capturar más detalles con mayor precisión y mayor resolución. Por ello, el Kinect v2 permite capturar una escena con tres veces más fidelidad que el Kinect v1. No es la única ventaja de utilizar este tipo de cámara. Con ella, además, se logra un campo de visión un 60 % más grande, lo que permite registrar un espacio mayor y posibilita que más personas puedan ser registradas al mismo tiempo y a una menor distancia de Kinect. Con el nuevo Kinect pueden aparecer hasta 6 personas en escena, reconociendo y distinguiendo sus movimientos. Es un avance importante respecto a su predecesora que solo era capaz de registrar el movimiento de 2 personas en total.

Figura 9. **Kinect v2**

Fuente: http://www.xbox.com/en-US/xbox-one/accessories/kinect. Consulta: noviembre de 2016.

El segundo gran cambio en la nueva generación de Kinect viene de la mano de la nueva cámara de profundidad que logra reconocer objetos y personas en condiciones de muy poca luz. La cámara de profundidad es ahora tan potente y precisa que puede reconocer objetos y personas en una habitación totalmente a oscuras. Incluso, con poca luz, puede reconocer la posición de la mano hasta a cuatro metros de distancia, distinguiendo cada uno de los dedos.

El Kinect puede reconocer el esqueleto completo del usuario, la orientación de sus miembros, los músculos del cuerpo con la fuerza y distribución del peso ejercida sobre ellos, y hasta el latido de su corazón. El reconocimiento facial también se ve ampliamente mejorado, detectando hasta el mínimo detalle en los gestos. Esto permite una identificación más precisa de cada uno de los usuarios. Esta nueva tecnología cuenta, además, con una mejora en el microprocesador de Kinect, el microprocesador X871141-001, que procesa gran cantidad de información que obtienen todos los nuevos dispositivos del nuevo Kinect. 2 GB de información por segundo son recogidos por el Kinect para leer el ambiente. Pero cambiar los dispositivos no ha sido suficiente. El Kinect en que se ha convertido necesita de un software capaz de interpretar todo lo que ve y, para ello, ha sido necesario llevar a cabo una importante evolución en el código que lo dirige.

1.7.3. Kinect v2 para PC

Se pueden escribir programas que utilizan el Kinect v1 o en el Kinect v2 en cualquier sistema operativo. El Kinect para PC ha sido creado para que sea más eficaz en el seguimiento de los usuarios. Esto significa que se puede realizar un seguimiento de los usuarios que están hasta 4,0 metros desde el Kinect, pero no puede realizar un seguimiento de todos los objetos que están más cerca de 0,8 metros. El Kinect v2 para PC se debe usar para que realice un seguimiento de un único usuario en una computadora, y rinda mejor en una distancia corta.

Figura 10. **Kinect para PC**

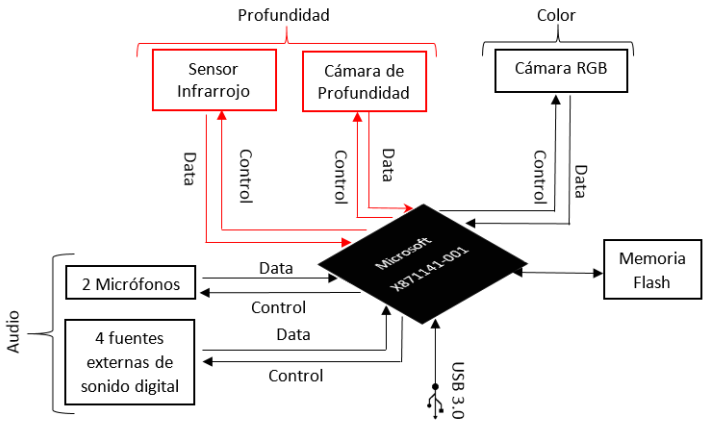
Fuente: https://blogs.msdn.microsoft.com/kinectforwindows/2014/06/05/pre-order-your-kinect-for-windows-v2-sensor-starting-today/. Consulta: noviembre de 2016.

El Kinect v2 para PC fue creado, principalmente, para usarlo con sistemas operativos, pero también puede trabajar con consolas de entretenimiento.

1.8. Interacción de componentes de Kinect

En el siguiente diagrama se muestra la conexión de todos los dispositivos de Kinect. Donde se expone que las imágenes a color son capturadas por la cámara de color, las imágenes de profundidad son proporcionadas en conjunto con la cámara de profundidad y proyector infrarrojo. Estos son indispensables para el funcionamiento de Kinect. El audio está conformado por el conjunto de micrófonos que capturan la voz del usuario. Los dispositivos fundamentales en el Kinect, que son el color, profundidad y audio, se transfieren mediante un bus de control y un bus de datos para utilizarlos. Adicionalmente, se tiene un bus para almacenamiento de datos temporales en la memoria *flash* y el puerto USB 3,0 que se encarga de la transmisión de datos y alimentación de energía del Kinect.

Figura 11. **Interacción de componentes de Kinect**



Fuente: elaboración propia, empleado Visio.

1.9. Capacidades y límites de Kinect

Con el Kinect, se puede capturar el movimiento de todo el cuerpo de los usuarios, así como pequeños gestos con las manos y hasta seis usuarios pueden ser reconocidos al mismo tiempo. Ofrece una amplia variedad de interacciones, pero posee limitaciones y puntos clave. Esto define los límites físicos y puntos óptimos. Los límites físicos van de 0,8 m hasta 4,0 m que son los valores por defecto. El ángulo de visión del Kinect va de los 70 grados en sentido horizontal y 60 grados en sentido vertical. Los puntos óptimos son las zonas donde las personas experimentan interacciones óptimas con el Kinect, dado que, con frecuencia, los usuarios necesitan gran espacio físico para movimiento y de ser rastreados con sus brazos o piernas extendidas.

El Kinect tiene cuatro micrófonos para que cualquier aplicación responda a comandos verbales de entrada y al movimiento. Detecta la entrada de audio de más 50 grados y menos 50 grados delante del mismo. El conjunto de micrófonos puede ser señalado en incrementos de 5 grados dentro del rango de

180 grados. Esto se puede utilizar para especificar la dirección de sonidos importantes, como un usuario que habla. El conjunto de micrófonos puede cancelar de 20 dB del ruido ambiental, lo que mejora la calidad del audio, eso es alrededor el nivel de sonido de un murmullo humano. El sonido que viene de detrás del Kinect recibe 6 dB adicional de supresión de sonido basado en el diseño de la carcasa del conjunto de micrófonos de Kinect. El Kinect es compatible con el sonido monofónico, pero no estereofónico.

2. CARACTERÍSTICAS Y UTILIZACIÓN DE KINECT

2.1. El kit de desarrollo de software de Kinect

Con el éxito del Kinect, se desarrolló un SDK oficial y gratuito que es compatible con los sistemas operativos actuales. Han existido seis versiones del SDK y cada una ha sido mejor que la anterior. Las versiones de SDK de Kinect han sido:

- SDK de Kinect 1.0 lanzado en febrero del 2012
- SDK de Kinect 1.5 lanzado en mayo del 2012
- SDK de Kinect 1.6 lanzado en octubre del 2012
- SDK de Kinect 1.7 lanzado en marzo del 2013
- SDK de Kinect 1.8 lanzado en noviembre del 2013
- SDK de Kinect 2.0 lanzado en julio del 2015

La empresa desarrolladora del Kinect ha facilitado sus códigos fuente que sirven de base para aprender a utilizar sus características, como la cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos. El SDK es un *kit* de herramientas para el desarrollo de aplicaciones para Kinect. El desarrollo de aplicaciones mediante el SDK de Kinect es bastante fácil y sencillo, proporciona una interfaz para interactuar con Kinect a través de controladores para computadora. El SDK de Kinect incluye controladores, que interactúan con el dispositivo a través de un sistema operativo para la utilización de determinada aplicación. En general, el SDK de Kinect proporciona la

oportunidad para crear una aplicación utilizando código administrado ya sea C# y VB.NET o código no administrado como lo es el C++.

Durante el desarrollo de aplicaciones para cualquier dispositivo que use un SDK, la compatibilidad en el sistema operativo juega papel fundamental y es importante que las aplicaciones que se desarrollarán cumplan con los algunos requisitos de hardware antes de comenzar a trabajar con el Kinect, en este caso. Los requisitos de hardware son menos estrictos que los requisitos de software, las aplicaciones se pueden ejecutar en la mayor parte del hardware disponible en el mercado. Las siguientes son las configuraciones de mínimas de hardware requeridas para el desarrollo de aplicaciones con el SDK de Kinect para un correcto funcionamiento de las aplicaciones que se desarrollarán.

- CPU Intel Core i7 a 3,10 GHZ
- 4GB de RAM
- GPU con DirectX 11
- USB 3,0

El SDK de Kinect está disponible gratuitamente y puede ser descargado de la página oficial del fabricante. Se instalan automáticamente las versiones de 64 o 32 bits del SDK en función del sistema operativo donde se está instalando.

2.2. Dentro del SDK de Kinect

El SDK de Kinect ofrece dos bibliotecas: .NET Kinect Runtime y Native Kinect Runtime que se utilizan para desarrollar aplicaciones con lenguajes de programación administrados y no administrados. Si se desarrolla una aplicación mediante C # o VB.NET, se tiene que utilizar la API llamada .NET Kinect Runtime; y para aplicaciones C++, se tiene que utilizar la API llamada Native

Kinect Runtime, estas API pueden interactuar con los controladores de Kinect que se instalaron como parte de la instalación del SDK de Kinect. Para código administrado, el SDK de Kinect proporciona una biblioteca de enlace dinámico llamado Microsoft.Kinect.dll, que se puede añadir a cualquier aplicación que se quiera utilizar el Kinect. Este archivo con extensión DLL, se puede encontrar en el directorio de instalación del SDK de Kinect. El SDK de Kinect puede controlar la cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos. La información capturada pasa desde el Kinect y la aplicación en tres tipos de datos, catalogados de la siguiente manera:

- Datos de color
- Datos de profundidad
- Datos de audio

El SDK de Kinect almacena la información capturada por la cámara de color en la base de datos de color. Los datos capturados por el proyector infrarrojo y cámara de profundidad son almacenados en la base de datos de profundidad. El sonido almacenado por el conjunto de micrófonos se almacena en la base de datos de audio.

2.3. Características del SDK de Kinect

Una vez discutido cómo se conforma el SDK de Kinect, los requisitos del sistema y la implementación de servicios en dispositivos, se observan las funciones de nivel superior del SDK de Kinect. El SDK de Kinect proporciona la librería Microsoft.Kinect.dll para interactuar directamente con la cámara de color, cámara de profundidad, proyector infrarrojo y conjunto de micrófonos. Incluso se puede desarrollar una aplicación acerca de reconocimiento de gestos utilizando el movimiento del cuerpo del usuario y también permitir que una

aplicación reconozca la voz. La siguiente lista de operaciones se pueden realizar con el SDK de Kinect:

- Capturar y procesar datos de imágenes en color.
- El procesamiento de la secuencia de imágenes de profundidad.
- La captura de la proyección de infrarrojos.
- Seguimiento de esqueleto humano y el movimiento de las articulaciones.
- Reconocimiento de gestos humanos.
- Captura de audio y reconocimiento de voz.
- Obtención de datos del acelerómetro.

2.4. Captura de imágenes de color

La cámara de color devuelve las imágenes a color de 32 bits a una resolución que va desde 640 por 480 píxeles hasta 1 920 por 1 080 píxeles. El Kinect tiene una velocidad de captura desde 15 FPS hasta 30 FPS dependiendo de las condiciones de luz en las que se encuentre. En otras palabras, utilizando el SDK, puede capturar imágenes en vivo a diferentes resoluciones. Aunque los datos de color son referidos como secuencia de imágenes. La misma técnica se utiliza para toma de un video.

Figura 12. **Imagen de color de Kinect**

Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: diciembre de 2016.

2.5. Captura de imágenes de profundidad

La cámara de profundidad devuelve imágenes de profundidad de 16 bits con una resolución de 521 por 424 pixeles a una velocidad de captura de 30 FPS. Cada uno de los píxeles en la imagen de profundidad representa la distancia entre el objeto y el Kinect cuando se realizó la captura de esta imagen.

Figura 13. **Imagen de profundidad de Kinect**

Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: diciembre de 2016.

2.6. Captura de imágenes de infrarrojo

El Kinect también puede capturar imágenes en condiciones de poca luz, es decir, imágenes infrarrojo. Esto es posible porque se capturan mediante la lectura de infrarrojos desde el Kinect. El Kinect devuelve imágenes de infrarrojo de 16 bits con una resolución de 521 x 424 a una velocidad de captura de 30 FPS. No es posible leer imágenes de color e infrarrojo en la transmisión de información simultánea, pero se puede leer información de profundidad e infrarrojos de forma simultánea. La razón es que los datos de infrarrojos se capturan como parte de la imagen de profundidad.

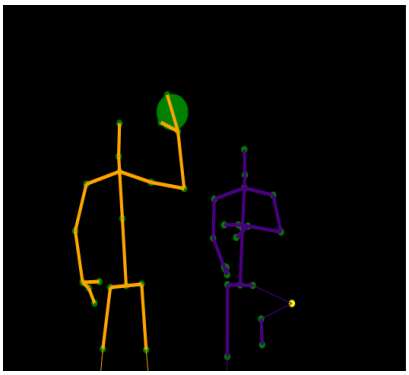
Figura 14. **Imagen de infrarrojos de Kinect**

Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: diciembre de 2016.

2.7. Seguimiento de movimientos y esqueleto humano

Una de las partes más interesantes del SDK de Kinect es su soporte para el seguimiento del esqueleto humano. Se puede detectar el movimiento del pie del esqueleto humano en un Kinect.

Figura 15. **Imagen de seguimiento de esqueleto humano de Kinect**



Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: diciembre de 2016.

El Kinect puede realizar un seguimiento de hasta 20 puntos en un solo esqueleto. Se puede realizar un seguimiento de hasta seis esqueletos, lo que significa que puede detectar hasta seis personas de forma simultánea si están de pie delante del Kinect.

2.8. Captura de sonidos de Kinect

El Kinect tiene cuatro micrófonos en una configuración lineal. El SDK de Kinect proporciona capacidades de procesamiento de audio de alta calidad mediante el uso de su propio audio interno en canalización de procesamiento. El SDK le permite capturar datos de audio puros y el procesamiento de audio de alta calidad. Esto es posible porque permite la supresión de ruido y las características de cancelación de eco que se pueden presentar en el ambiente donde se encuentra el Kinect. También puede determinar la dirección del sonido que captura el conjunto de micrófonos con la ayuda del SDK.

2.9. Reconocimiento de voz

Se puede lograr del conjunto de micrófonos de Kinect y el API del SDK de Kinect el reconocimiento de voz. Para desarrollar aplicaciones pertinentes, se puede construir su propio vocabulario, pasarlo al motor de reconocimiento de voz y diseñar su propio conjunto de comandos de voz para controlar la aplicación. Si un usuario decide realizar un gesto y hablar al mismo tiempo, se puede desarrollar una aplicación para llevar a cabo algún trabajo que realice en función de los gestos y voz del usuario.

2.10. Reconocimiento de gestos

El reconocimiento de gestos ha sido un área de investigación principal durante largo tiempo. Sin embargo, en la última década, una enorme cantidad de tiempo, esfuerzo y recursos se han dedicado a este campo como consecuencia del desarrollo de dispositivos con esta característica. El reconocimiento de gestos permite a las personas interactuar con un dispositivo de forma natural con el cuerpo. En el SDK de Kinect, no hay apoyo directo para una API, para reconocer y tratar con los gestos. Sin embargo, mediante el uso de seguimiento del esqueleto y el procesamiento de datos de profundidad, se puede construir una API de gestos, que puede interactuar con una aplicación.

2.11. Acelerómetro de Kinect

El Kinect trata el ángulo de elevación en relación con la gravedad y no desde su base, como se utiliza sus acelerómetros para controlar la rotación, El SDK de Kinect expone las API para leer los datos del acelerómetro directamente desde el Kinect y puede detectar la orientación del Kinect mediante la lectura de los datos de acelerómetro.

2.12. Seguimiento de esqueleto humano

Es la característica principal de Kinect, ya que detecta el esqueleto humano, cuando está en moviendo. Los movimientos son asociados a través de puntos, los cuales grafican el esqueleto humano. Para la detección de este esqueleto, el Kinect toma 20 articulaciones. Esto es posible, porque el seguimiento de esqueleto humano tiene almacenado los datos de la cámara de color y de la cámara de profundidad, lo cual permite reconocer la diferencia entre objetos y humanos. Para ello, asocia los parámetros del esqueleto humano, como extremidades superiores e inferiores, articulaciones y gestos, para reconocerlo y detectar el movimiento de todas las partes que lo conforman, por ejemplo, la cabeza, brazos, manos, cintura, rodillas, pies, entre otros.

El seguimiento de esqueleto humano consta de una clase llamada SkeletonFrame y este a su vez está compuesto por un vector de SkeletonData, que es un vector que contiene los datos del esqueleto humano, uno por cada esqueleto humano que se reconoce. De esta forma, cuando el usuario realiza un movimiento, el Kinect lo captura y lo trasladará a la consola o PC para que él asocie esta información y lo muestre en la aplicación. El seguimiento de esqueleto humano es el método por el que Kinect es capaz de detectar al usuario que se sitúa delante. El mismo consta de 6 pasos:

- Kinect lanza una serie de puntos.
- Kinect crea el mapa de profundidad a partir de los puntos detectados.
- Detecta el suelo y separa los objetos del fondo para encontrar al usuario.
- Asocia las partes detectadas para hacer una clasificación de las partes humanas.
- Identifica las articulaciones del usuario.
- Simula el esqueleto humano.

A cada usuario le corresponde un único ID mientras se mueva dentro del rango de visión del Kinect. Si sale del campo de visión del Kinect pierde su ID. El valor de la posición del usuario es el único valor que tienen los jugadores con el seguimiento de esqueleto humano y la posición de cada punto viene determinada por coordenadas X, Y y Z. A diferencia de los datos de la imagen de profundidad, las medidas están expresadas en metros. Este se trata de un sistema de coordenadas que coloca el Kinect en el punto de origen con el lado positivo del eje *z* extendiéndose en la dirección en que apunta el Kinect, el lado positivo del eje *y* extendiéndose hacia arriba y el lado positivo del eje *X* extendiéndose hacia la izquierda respecto del Kinect. Cada eje indica: eje X, posición horizontal, viene dada por la distancia en metros del Kinect a lo largo del eje X. Eje Y, posición vertical, viene dada por la distancia en metros del Kinect a lo largo del eje Y. Eje Z, distancia desde el Kinect medida en metros.

El SDK de Kinect también soporta el seguimiento del esqueleto de un cuerpo humano cuando está sentado. El Kinect puede realizar un seguimiento de sus articulaciones, incluso si el usuario está sentado, sin embargo, solo presenta 10 puntos enfocados en la parte superior del cuerpo. El Kinect reconoce otros puntos del esqueleto humano y se enfoca en las partes del nivel superior del usuario, de la cintura hasta la cabeza.

2.13. Reconocimiento facial

El reconocimiento facial es una parte del SDK de Kinect que contiene un par de conjuntos de APIs que se pueden utilizar para realizar un reconocimiento facial. El SDK de Kinect detecta y rastrea las posiciones y orientaciones de los rostros humanos y también puede animar a las posiciones de la ceja y la forma de boca en tiempo real. Las capacidades del reconocimiento facial de Kinect son como el reconocimiento de expresiones faciales, la interacción facial con

aplicaciones con interfaces naturales de usuario, y tareas que están relacionadas con el rostro humano.

Figura 16. **Reconocimiento facial de Kinect**

Fuente: http://www.codeproject.com/Articles/394975/How-To-Use-Kinect-Face-Tracking-SDK. Consulta: diciembre de 2016.

3. PROCESAMIENTO DE DATOS COLOR Y PROFUNDIDAD

Para el desarrollo de todas las aplicaciones de Kinect, hay ciertas operaciones que se necesitan realizar, las cuales son:

- La aplicación debe detectar el Kinect conectado y necesita iniciarlo.
- Una vez que se inicia el Kinect, la aplicación tiene que inicializar y escribir el tipo de datos necesarios desde el Kinect.
- Durante el ciclo de ejecución de la aplicación, el Kinect puede cambiar su estado. La aplicación debe monitorear los cambios en el estado del Kinect y manejarlos adecuadamente.
- Cuando la aplicación se cierre, debe apagar el Kinect correctamente.

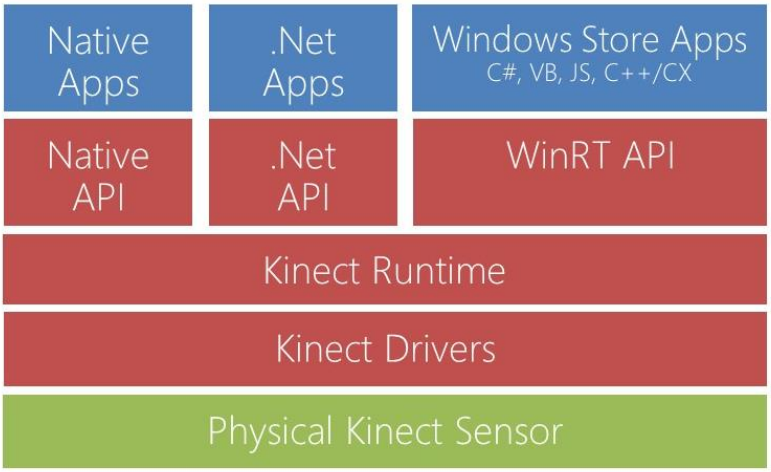
3.1. Cómo las aplicaciones interactúan con Kinect

El SDK de Kinect funciona como una interfaz entre el Kinect y su aplicación. Cuando necesite acceder al Kinect, la aplicación envía una llamada a la API hacia el controlador del Kinect. El controlador del Kinect tiene acceso a los datos del Kinect, los controladores instalados para el Kinect permiten interactuar entre los diferentes componentes que lo conforman.

Para desarrollar aplicaciones con las API del SDK de Kinect mediante el acceso a la cámara color y profundidad de Kinect e interactuar con los datos capturados se debe seguir un proceso desde los datos capturados por el Kinect hasta llegar a la aplicación. Como parte del Kinect, la cámara ayuda en la captura de imágenes de color, habilita el seguimiento facial y otros más. La

primera aplicación que se puede desarrollar en el Kinect debe utilizar la cámara de color de Kinect para, posteriormente, ampliar las capacidades y opciones. En el siguiente diagrama se puede observar cómo las aplicaciones interactúan con el Kinect:

Figura 17. **Interacción de las aplicaciones con Kinect**



Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: diciembre de 2016.

Un video no es más que una sucesión de cuadros de imágenes. El Kinect puede entregar un marco de imagen fija dentro de un rango de velocidad de 15 y 30 FPS. Esta tasa de transferencia puede cambiar según el tipo y la resolución que se necesite en la aplicación. El SDK de Kinect es compatible con dos siguientes tipos de formatos imagen:

- Imágenes de color
- Imágenes de profundidad

Estas imágenes son transferidas de diferente manera, por lo que se deben habilitar diferentes métodos para obtener los datos de imagen desde el Kinect.

El tipo de imagen dependerá de los parámetros de entrada, como la resolución, tipo de imagen, y la tasa de transferencia. Con base en sus entradas, el Kinect inicia la transferencia en los canales de datos. Si no hay especificaciones, la aplicación tomará imágenes de manera predeterminada.

3.2. Captura de imágenes de color en Kinect

Los cuadros de imagen son tomadas bajo la clase *ColorImageFrame*. Dependiendo del tipo de imágenes que está utilizando, el Kinect volverá el cuadro de imagen correspondiente. Hay dos formas de capturar imágenes en una aplicación:

- Modelo de eventos
- Modelo de sondeo

3.2.1. Modelo de eventos

Usando el modelo de eventos, el Kinect enviará una imagen a la aplicación siempre que una nueva imagen es capturada por el Kinect. Por ello, se necesitará activar el controlador de modelo de eventos en el código de la aplicación, donde se procesará la entrada de imágenes. Antes de activar al modelo de eventos, deberá informar al SDK del tipo y resolución de la imagen que se necesita. Una vez que se active el modelo de eventos, el Kinect enviará los datos de forma continua a menos que deshabilite, se detenga la transferencia de datos o se detenga el Kinect.

3.2.2. Modelo de sondeo

El modelo de sondeo es a donde se debe enviar una solicitud al Kinect cada vez que se necesite obtener una imagen desde la aplicación. Para el modelo de sondeo, se debe esperar cierto intervalo de tiempo para que el Kinect devuelva una imagen a color, después de que se haya hecho la solicitud para capturar tal imagen.

3.3. Cámara de profundidad

Hasta el momento se ha explicado cómo funciona la cámara de color de Kinect, es decir, cómo se capturan datos de imagen de la cámara de color con los tres componentes de color principales: el rojo, verde y azul. De forma parecida, se pueden capturar los datos desde la cámara de profundidad del Kinect. Sin embargo, el principio de la cámara de color y la cámara de profundidad son totalmente diferentes. En la cámara de color, cada cuadro de color se compone de un número de valores de píxeles, que dan los valores de componentes de color rojo, verde y azul; mientras que cada píxel de la información de la cámara de profundidad representa la distancia de un objeto desde el Kinect. Los datos obtenidos con la cámara de profundidad son los aspectos más importantes del Kinect. Los datos obtenidos en la cámara de profundidad son muy importantes para la construcción de cualquier aplicación en Kinect.

Figura 18. **Puntos infrarrojos de Kinect con visión nocturna**

Fuente: http://www.silenzine.com/wp-content/uploads/2014/01/kinect-toby-paranormal-4.png. Consulta: diciembre de 2016.

Tanto la cámara de profundidad y el proyector infrarrojo trabajan en conjunto para producir la información de la profundidad a que se encuentran los objetos que se tienen enfrente al Kinect. Por lo que se deben detallar los conceptos básicos de procesamiento de datos de profundidad y su uso: también sobre cómo identificar la distancia de un objeto o usuario que esté al frente del Kinect.

3.4. Análisis de los datos de profundidad

El Kinect devuelve los datos de profundidad como una sucesión datos capturados de los valores de profundidad en un marco determinado. El Kinect vuelve los datos de profundidad en un formato de escala de grises de 16 bits con un alcance de visión de 43 grados en vertical y 57 grados en horizontal. Ya que los datos de profundidad no son solamente una imagen, detrás de los datos capturados por el Kinect se ejecuta una serie de algoritmos en cada captura de datos, que indican la distancia que cada píxel de profundidad posee. El píxel de

profundidad contiene la distancia entre el objeto o usuario y el Kinect expresada en milímetros. Los datos se basan en coordenadas X e Y en la vista del Kinect. Por ejemplo, si una coordenada de píxel es 240 x 320, los datos de la profundidad de ese punto de píxel contienen la distancia en milímetros desde el Kinect al objeto o usuario.

El rango del ángulo de visible seguirá siendo el mismo con el cambio de ángulo de elevación del Kinect, ya que este cambio se considera como un cambio en la base del Kinect. La resolución compatible con los datos de profundidad es de 424 por 512 píxeles. El Kinect puede capturar los datos de aproximadamente hasta una distancia de 4,0 metros, o distancias muy cercanas a 0,8 metros, sin la pérdida de exactitud o precisión. El Kinect también puede tomar datos de profundidad a objetos más allá de 4,0 metros, pero en tales casos la calidad y exactitud se ve comprometida por el ruido que se puede generar en distancia muy largas.

El Kinect se compone de dos dispositivos que utilizan tecnología infrarroja, los cuales son la cámara de profundidad y el proyector infrarrojo. La cámara tiene como finalidad detectar la distancia de rayos infrarrojos que el proyector emite. Ambos componentes tienen que trabajar en conjunto para producir la salida deseada. El proyector infrarrojo emite luz infrarroja en un patrón de puntos pseudoaleatorio constantemente en la parte frontal del Kinect. Estos puntos son invisibles para el ojo humano, ya que la longitud de onda infrarroja es más larga que la longitud de onda de la luz visible para los ojos humanos. La cámara de profundidad se encarga de leer los puntos en la escena, y envía la información de los pixeles de profundidad de la que fueron reflejados. Para lograr esto, el Kinect recure a una técnica llamada triangulación estéreo.

3.5. Triangulación estéreo

La triangulación estéreo es un algoritmo de análisis para el cálculo de la posición 3D de los puntos en un cuadro de imagen. En la triangulación estéreo, en general, se utilizan para obtener dos vistas diferentes en una escena, de una forma similar a la visión binocular humana. Por la comparación de estas dos imágenes, se calcula la información de profundidad relativa. Cuando se trata de Kinect, para los datos de profundidad solo hay una imagen, que es capturada por la cámara de profundidad y proyector infrarrojo. Una parte es capturada por la cámara de profundidad con los valores de distancia y la otra es invisible, ya que es un patrón de emisión de infrarrojos que es enviado por el proyector infrarrojo. Todo lo que el proyector infrarrojo hace es proyectar un patrón pseudoaleatorio según las especificaciones que posee el Kinect.

Estas partes de imagen no son equivalentes, ya que hay cierta distancia entre la cámara de profundidad y proyector de infrarrojos. Esta imagen es considerada como diferente a la imagen de la cámara de color. Los datos de profundidad dependen de los rayos infrarrojos, por lo que en la medición de la profundidad los datos pueden verse afectados si se coloca el Kinect en directo con la luz solar o con cualquier otro dispositivo que interfiere con los rayos infrarrojos. Para capturar los datos de profundidad del Kinect, el programa debe utilizar exactamente las mismas escenas que utilizamos para leer imágenes a color. Si se acerca el usuario o hay un objeto cercano al Kinect en la captura de datos de profundidad, este se muestra de color blanco en la imagen de datos de profundidad. En caso contrario si se aleja se tornará de color negro en la imagen de profundidad, de esto se trata la captura de los datos básicos de profundidad en bruto del Kinect.

3.6. Los datos de profundidad y distancia

El rango de visión de la cámara de profundidad para el Kinect va de 0,8 m a 4,0 m, que es el intervalo predeterminado para la captura de datos de profundidad. Sin embargo, el Kinect puede captar la información más allá de 4,0 m, en tales casos se ve comprometida la calidad de los datos que el Kinect toma. Las clases DepthImageStream y DepthImageFrame tienen las propiedades MaxDepth y MinDepth, que devuelven el máximo y mínimo de la distancia de profundidad que puede ser tomado por el Kinect. Este rango de valor devuelve mejor el cálculo de distancia que se puede medir correctamente. El Kinect puede realizar un seguimiento de los objetos y medir las distancias que pueden tener estas propiedades. Cuando hay un píxel de distancia en cualquier punto del Kinect, internamente dibuja una línea que es perpendicular al Kinect y luego calcula la distancia directamente desde allí.

3.7. Calculo de distancia

El Kinect devuelve los datos de profundidad de 16 bits. Se utilizan los tres primeros bits para representar a los jugadores identificados y los 13 bits restantes tienen la distancia en milímetros. Por lo tanto, primero se necesita para llevarlo a cabo es una operación de desplazamiento para mover los bits a su posición correcta, el ID Usuario representa el código único en binario cuando se encuentran hasta un máximo de 6 usuarios.

Tabla I. **Ilustración de los 16 bits del pixel de profundidad**

													ID Usuario		
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Bits de profundidad															

Fuente: https://msdn.microsoft.com/en-us/library/jj131029.aspx. Consulta: diciembre de 2016.

Para obtener la distancia de un píxel en particular, en primer lugar, es necesario calcular el pixelIndex desde la posición seleccionada actual. El Kinect captura datos dentro de un cierto rango de distancia, que se especifica por la propiedad Range de la clase DepthStream. La propiedad Range establece el rango visible para el Kinect, la clase DepthRange puede tener los dos valores siguientes:

- Rango por defecto
- Rango cercano

De forma predeterminada, DepthRange se establece el modo en defecto y el rango varía desde 0,8 m hasta 4,0 m. En la función modo cerca, ayuda a seguir un cuerpo humano dentro de un rango muy cerca y es útil cuando el Kinect se usa en una consola, ya que el usuario se encuentra a una distancia considerable para realizar movimientos con su cuerpo. En el rango cercano, este se establece desde 0,4 m hasta 3,5 m, con ello se tiene un seguimiento más cercano del Kinect, ideal para trabajar en entornos con computadora.

3.8. Valores de rango de profundidad

DepthImageStream define tres propiedades de solo lectura que son *TooFarDepth*, *TooNearDepth*, y *UnknownDepth*, que ayudan mejorar el control

sobre la distancia, proporcionando un rango, donde es imposible conseguir los valores de profundidad:

- TooFarDepth: Proporciona el punto de profundidad donde el objeto está más allá del rango del Kinect.
- TooNearDepth: Proporciona el punto de profundidad donde el objeto está demasiado cerca del Kinect.
- UnknownDepth: Hay casos en que una profundidad es totalmente indeterminada; esto puede ser considerado como una profundidad desconocida, y el valor será cero.

3.9. Distribución de los datos de profundidad

Una de las mejores maneras de representar gráficamente la distribución de los datos es el histograma. Los histogramas cuentan, visualmente cómo se distribuyen los valores de los datos actuales para un determinado conjunto de datos. A partir de un histograma, podemos identificar con qué frecuencia y cómo los datos se distribuyen. Los valores de los datos de profundidad contienen distancias para los diferentes rangos posibles, sombras, o incluso información de profundidad desconocida. Con la representación de histograma, se hace posible:

- Identificar el rango de datos para filtrar los datos de profundidad
- La aparición de un cierto rango de valores
- Distribución de probabilidad para un rango específico de datos
- Definición del rango de captura de datos

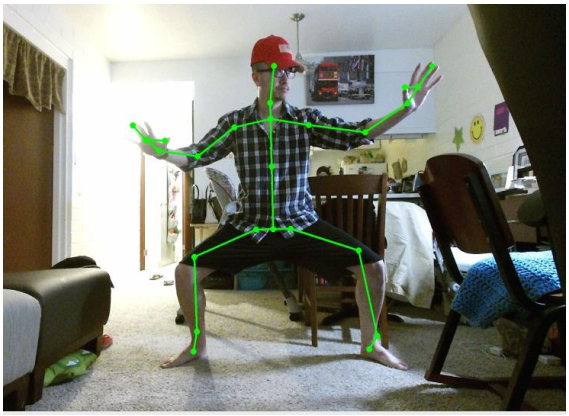
Observar los valores de profundidad generando un histograma es muy fácil, ya que la representación es tan simple como un gráfico de barras. Se ha

visto que, para un pixel de profundidad son 16 bits en bruto, los tres primeros bits representan el índice de jugador y los 13 bits superiores representan la distancia. El seguimiento de usuarios requiere que el seguimiento de esqueleto humano esté habilitado.

3.10. Seguimiento de esqueleto humano

El Kinect devuelve los datos de profundidad en bruto, donde cada pixel contiene un valor que representa la distancia entre el Kinect y el objeto. Anteriormente, se ha explicado cómo funcionan las técnicas de procesamiento de imágenes, cámara de profundidad, la forma en que se puede medir las distancias, y cómo cada valor de píxel representa la información capturada del usuario.

Figura 19. **Seguimiento de esqueleto humano en acción**



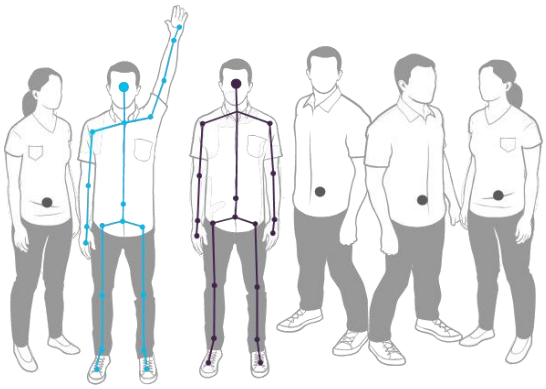
Fuente: https://matrivian.files.wordpress.com/2015/08/rgb.png. Consulta: diciembre de 2016.

Los datos de profundidad ofrecen posibilidades ilimitadas con el Kinect. Para construir una aplicación interactiva y lograr que el usuario tenga una experiencia enriquecedora, se deben agregar características donde se pueda

utilizar movimientos del cuerpo. Cuando se habla de cómo construir una aplicación que interactúa con el movimiento del cuerpo humano, en primer lugar, se debe capturar la información sobre los usuarios que se colocan delante del Kinect y, a partir de ese punto, el seguimiento del esqueleto humano entra en acción. La función de seguimiento del esqueleto humano está construida sobre el procesamiento de datos de profundidad, auto aprendizaje, y algoritmos de imágenes a color.

Usando el SDK de Kinect, se puede realizar un seguimiento de hasta seis jugadores y un máximo de 20 articulaciones para cada esqueleto. Solo dos usuarios se pueden rastrear en detalle, lo que significa que el Kinect puede devolver los veinte puntos seguimiento en forma simultánea de los dos usuarios. Mientras que, los cuatro usuarios restantes, el Kinect solo da la posición general de los mismos sin mencionar las articulaciones, como se representa a continuación:

Figura 20. **Ilustración de seis usuarios en frente de Kinect**



Fuente: https://i-msdn.sec.s-msft.com/dynimg/IC584841.png. Consulta: diciembre de 2016.

Esto se debe a que requeriría un gran poder de procesamiento para realizar un seguimiento de todas las articulaciones de los seis usuarios. El

Kinect solo puede obtener las 20 articulaciones completas de los dos usuarios principales. Para las otras cuatro personas, se obtiene información solo de la articulación de la cadera central. En los dos esqueletos humanos rastreados, uno de ellos estará activo y el otro será tratado como pasivo, en función de cómo se están utilizando los datos del esqueleto. Si en un esqueleto humano se realiza un seguimiento completo, en la siguiente captura de datos se volverán a capturar los datos del esqueleto humano completo, mientras que, para los esqueletos de forma pasiva, se obtendrán solo las posiciones propuestas nuevamente. El SDK de Kinect proporciona un conjunto de APIs que permiten acceso fácil a las articulaciones del esqueleto humano. El SDK de Kinect es compatible hasta 20 puntos en conjuntos. Cada posición de la articulación se identifica por su nombre, por ejemplo: cabeza, hombros, codos, muñecas, brazos, columna vertebral, caderas, rodillas, tobillos, etc. El estado de esqueleto de seguimiento es determinado por cualquiera de los estados: seguimiento, no seguimiento o solo posicionamiento.

. El SDK de Kinect también soporta el seguimiento de un esqueleto humano sentado. Se puede cambiar el modo de seguimiento para detectar un cuerpo humano sentado, que este devuelve hasta las 10 articulaciones de parte superior de usuario.

3.11. Seguimiento de articulaciones

El Kinect devuelve datos esqueleto en forma de la clase SkeletonStream. Se puede establecer el modo seguimiento por defecto o el seguimiento sentado con la propiedad SkeletonTrackingMode durante la inicialización del seguimiento del esqueleto humano. El proceso para la captura de datos del esqueleto humano seguirá siendo la misma que la que se utilizó para la captura de datos en color y datos de profundidad. Se pueden capturar los datos

mediante el uso de cualquiera de los modelos, tanto el modelo de eventos o modelo de sondeo.

El Kinect tiene un evento denominado SkeletonFrameReady, que captura cada vez que los nuevos datos del esqueleto están disponibles en el Kinect. Cada paquete de información que se encuentra en el SkeletonStream produce un nuevo paquete de posiciones del esqueleto humano capturado por el Kinect. Cada paquete de posiciones del esqueleto humano contiene los datos de cada conjunto de puntos, que este se ajusta dentro de la clase JointCollection.

4. PROCESAMIENTO DE AUDIO DE KINECT

El Kinect consiste en un conjunto de micrófonos que dan soporte el audio que captura el Kinect. Este tiene cuatro micrófonos que miran hacia el frente y están separados en la parte inferior del Kinect de forma lineal. El conjunto de micrófonos permite lo siguiente al Kinect.

La captura de la mejor calidad de sonido y la disponibilidad para el procesamiento de señales de audio, incluyendo la supresión de ruido, cancelación de eco y la identificación de la dirección de la fuente del sonido entrante, con base en el sonido que captura individualmente cada micrófono, se puede, automáticamente, identificar la dirección en la que el sonido y escuchar, específicamente, el sonido que captura ese micrófono mediante la supresión de los otros ruidos.

Una vez que se establece la dirección de la fuente de sonido, el Kinect es suficientemente inteligente como para cambiar la dirección de cómo y cuándo la fuente del sonido se mueve. Uno de los ejemplos más comunes de tal escenario es cuando un usuario está usando comandos de voz. Si el usuario se mueve, la dirección de la fuente de sonido se mueve automáticamente. El Kinect tiene un canal de procesamiento de audio incorporado que se encarga de las capacidades completas de procesamiento de audio. Otro aspecto importante del conjunto de micrófonos de Kinect es el reconocimiento de voz. El conjunto de micrófonos ayuda a reconocer la voz humana con toda claridad, centrándose solo en una dirección particular y cancelando ruidos en el medio ambiente.

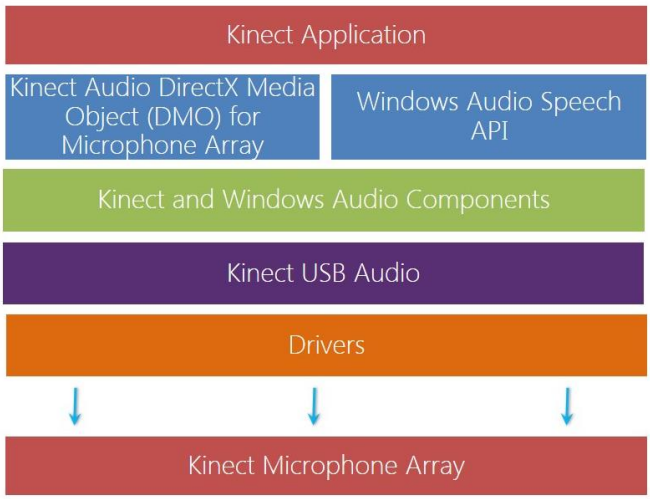
En este capítulo se profundiza en la comprensión de procesamiento de datos de audio de Kinect usando el conjunto de micrófonos e interactuando con la API de audio del SDK de Kinect. Incluye las principales áreas que se pueden tratar con el procesamiento de audio con Kinect, los cuales son la verificación de la configuración de audio de Kinect, arquitectura SDK de Kinect para el audio, cómo procesa las señales de audio el Kinect, el interior del conjunto de micrófonos de Kinect, captura y reproducción de audio, procesamiento de audio mediante la supresión de ruido y cancelación, comprensión de sonido de la localización de la fuente y la formación de haz de audio.

4.1. Arquitectura del SDK de audio de Kinect

El SDK de Kinect instala los componentes de audio de Kinect que interactúan con el conjunto de micrófonos del Kinect y los componentes para el reconocimiento de voz, el Kinect utiliza la API de voz subyacente de la operación del sistema operativo. El Kinect tiene su propio canal que procesa los datos de audio capturados; sin embargo, cuando se trata bajo el nivel del sistema operativo, la API de audio se basa en componentes del marco de audio existente.

Junto con los controladores de Kinect, los siguientes son los componentes principales de captura de audio: DirectX Media Object y Windows Speech Recognition API. La mayoría del procesamiento de audio, por ejemplo Noise Suppression, Acoustic Echo Cancellation y Automatic Gain Control es controlado por el DirectX Media Object. El SDK de Kinect expone un conjunto de API’s que pueden controlar las características mencionadas anteriormente a través de la aplicación de Kinect. Desde el siguiente diagrama, se puede ver que el audio capturado del conjunto de micrófonos del Kinect pasa a la aplicación a través los componentes de Kinect y de audio del sistema operativo.

Figura 21. **Diagrama de procesamiento de audio de Kinect**



Fuente: https://msdn.microsoft.com/en-us/library/jj883682.aspx Consulta: diciembre de 2016.

Cuando hay necesidad de procesar algunos datos mediante la aplicación de Kinect, es necesario utilizar estos conjuntos de métodos desde el SDK de Kinect, internamente llamar al método DirectX Media Object para realizar la acción. Por otro lado, el reconocimiento completo de voz de Kinect se basa en la Windows Speech API.

4.2. El área de enfoque del audio de Kinect

El área de enfoque principal de procesamiento de audio de Kinect fue el reconocimiento de voz de los usuarios cuando se mueven alrededor y están en diferentes posiciones, por lo que se enfrentaron dos retos. El primero fue la identificación de audio con ruido adicional. Se trata de una situación en la que está un usuario y un ruido llega desde una fuente diferente, como un televisor. Eso dificulta el reconocimiento de la voz del usuario debido al fuerte ruido, así como el eco y otros posibles ruidos en la habitación. El segundo reto fue

identificar el discurso dentro de un área. Durante la interacción del Kinect, el usuario podría cambiar su posición, o varios usuarios podrían estar hablando desde diferentes direcciones. Para superar todos estos problemas y ofrecer una mejor solución para el reconocimiento de voz, el Kinect tiene un conjunto de micrófonos que capturan la voz y hacen frente a los requerimientos de sonidos de alta calidad. El Kinect tiene cuatro micrófonos; tres de ellos están en el lado derecho y el otro a la izquierda, la lógica detrás de la colocación de micrófonos en diferentes lugares es identificar el origen y la dirección del sonido.

Como todos los micrófonos se colocan en diferentes posiciones, el sonido llegará a cada uno de los micrófonos en diferentes intervalos de tiempo, lo que significa que debería haber algún retraso en la recepción del sonido entre cada micrófono. Por esto, el Kinect puede entender la dirección de la que proviene el sonido y calcula la distancia aproximada con base en la onda de sonido y el tiempo de diferencia del sonido desde la fuente real hasta el Kinect. Esta acción es similar al funcionamiento de los oídos y el cerebro. Al tener una distancia de 149 mm entre el primer micrófono y el segundo, el Kinect puede ver una diferencia entre retardo de sonido y el cálculo del mismo y su dirección de origen. Hay un retraso de hasta siete muestras entre el micrófono en el extremo derecho y el primer micrófono que está a la izquierda, considerando que hay menos retardo para los otros tres micrófonos colocados más cerca unos de otros.

4.3. Procesamiento de señales de audio en Kinect

El Kinect tiene su propio canal de procesamiento de audio incorporado para estos obtener estos datos. Una vez que se calcula la fuente y la posición del sonido, el canal de procesamiento de audio fusiona las señales de todos los

micrófonos y produce una señal de alta calidad. El Kinect puede identificar el sonido dentro de un rango de 50 grados frente a él.

El ángulo en el que Kinect puede escuchar es llamado ángulo de fuente de audio. Como Kinect es responsable del reconocimiento de voz, la canalización de procesamiento de audio también aplica un filtro en la frecuencia de la onda para suprimir las que están fuera de las frecuencias de la voz humana, la cual va entre 300 Hz y 3 800 Hz. El canal de procesamiento de voz también filtra cualquier otro tipo de ruido, la eliminación de los ecos, y la producción de una voz amplificada.

Los canales de procesamiento de audio del Kinect utilizan varios procesadores de señal digital que contienen los complejos algoritmos para producir un mejor reconocimiento de voz, independientemente de las circunstancias. La combinación de diferentes señales de sonido, mediante la identificación de la fuente de sonido y escuchando a una dirección en particular, se llama formación de haz. Los siguientes son algunos componentes clave del procesamiento de voz:

- Acoustic Echo Cancellation
- Beam Former
- Sound Source Localizer
- Noise Suppression
- Automatic Gain Controller

La combinación de Beam Former y Sound Source Localizer hace que el Kinect tenga un micrófono altamente direccional de modo que pueda escuchar a una dirección particular. El Beam Former cambia el ángulo de la formación de haz con base en la respuesta del Sound Source Localizer. Kinect está

escuchando el ángulo devuelto por el formador de haz. Igual que otros micrófonos, el Kinect utiliza Acoustic Echo Cancellation para eliminar los sonidos que vienen desde la fuente que produce el sonido. Noise Suppression se utiliza para suprimir los sonidos no deseados. El Automatic Gain Controller se encuentra al final del canal y se utiliza como un amplificador. Esto le permite al Kinect controlar la voz del usuario y mantener el audio capturado a alto nivel, mientras que el usuario se mueve alrededor del Kinect.

4.4. Control sobre el conjunto de micrófonos de Kinect

El SDK de Kinect expone un conjunto de propiedades y métodos que ayudan a capturar el audio desde el conjunto de micrófonos Kinect y procesar los datos de audio. El SDK tiene una clase de llamada KinectAudioSource que es la principal responsable de la captura y la manipulación de los datos de audio. El conjunto de micrófonos puede suministrar cuatro canales de audio de 32 bits a 16 kHz. El canal de audio debe estar habilitado antes de que los micrófonos empiecen a enviar datos de audio. La clase KinectAudioSource expone un método llamado *start*() para activar el canal de audio e iniciarlo para su captura.

El Kinect debe estar en el estado de ejecución para iniciar la captura de audio, si se intenta iniciar la captura de audio y el Kinect no se está ejecutando, la aplicación generará un InvalidOperationException. Por lo que siempre es bueno comprobar la característica IsRunning de KinectSensor antes de iniciar el canal de audio. Si IsRunning vuelve falsa, inicie el Kinect antes de que el canal de audio.

A partir de la transmisión de audio después de un intervalo de tiempo la clase KinectAudioSource tiene un método sobrecargado para iniciar el audio. La

única diferencia en este método es que se necesita un período de espera para asegurarse de que el Kinect está listo para la captura de audio.

Si usted no lee el valor dentro del período de tiempo especificado, el método DirectX Media Object descarta los datos de amortiguamiento y serán capturados nuevos datos. Una vez que se inicia la captura de audio, el Kinect comienza a enviar los datos de audio, entonces se pueden utilizar fácilmente los datos de audio. Puede tener un solo canal de audio en marcha a la vez de un Kinect. Si se iniciará un canal de audio y ya se ha abierto otro, el SDK de Kinect primero detendrá el vigente antes de comenzar el nuevo canal de audio.

4.5. Reconocimiento de voz

Uno de los aspectos clave de la interfaz de usuario natural es el reconocimiento de voz. Permite que los usuarios realicen cualquier acción con cualquier comando frente al conjunto de micrófonos de Kinect. En el otro lado el ordenador ejecuta una acción determinada en función del comando reconocido. El conjunto de micrófonos de Kinect funciona como un dispositivo de entrada excelente para aplicaciones de voz. La captura de audio es mejor que cuando se realiza con un solo micrófono. Suprime el ruido, el eco y se escucha en una dirección específica con la ayuda de localización de la fuente de sonido. La clase KinectAudioSource sirve para la transmisión y procesamiento digital de audio. Si se le combina con la clase SpeechRecognitionEngine demuestra el poder de utilizar el conjunto de micrófonos de Kinect. Con ello se puede aprender acerca de reconocimiento de voz usando el conjunto de micrófonos de Kinect y cómo construir algunas aplicaciones habilitando el uso de comandos de voz.

Una aplicación puede tener diferentes tipos de interfaz de usuario. Uno de los enfoques de interacción es el control de la interfaz de usuario utilizando la voz. Usando el sistema de reconocimiento de voz, los usuarios dicen lo que quieren y el ordenador ejecuta el mando. Los resultados se reflejan en la interfaz de usuario, por lo cual los patrones de reconocimiento de voz se clasifican en las formas siguientes.

Modo comando: en este modo se dice un comando y el reconocimiento de voz lo reconoce. Por ejemplo, es posible que desee iniciar y detener una aplicación con sólo decir "Iniciar" y "Detener".

Modo sentencia: en este modo se puede decir una frase para realizar una operación. Por ejemplo, para girar una línea, se puede decir "Rotar la línea". A primera vista, el reconocimiento de voz se parece a una lógica de simple coincidencia, pero de hecho no lo es. El reconocimiento de voz se compone de los siguientes dos modelos principales:

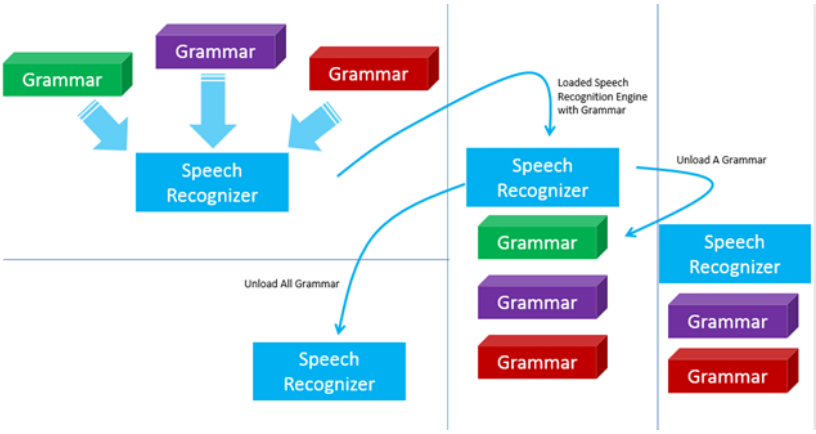
- Modelo de acústico
- Modelo de lenguaje

Cada uno de los modelos tiene una responsabilidad exclusiva para el reconocimiento de voz. Esto es una tarea de reconocimiento de patrones que se realiza en diferentes etapas con el reconocimiento de voz. La operación para el reconocimiento de voz del usuario se realiza de la siguiente manera: los micrófonos capturan el flujo de audio y convierten los datos de audio analógico en digital para que los entienda la computadora. Esta operación se hace en la primera etapa del procesamiento de audio porque el audio necesita que una mejor representación acústica sea adicionalmente procesada por el motor de reconocimiento de voz.

Las señales de sonido de audio se envían al motor de reconocimiento de voz. El modelo acústico del motor de reconocimiento de voz analiza el audio y convierte el sonido en un número de elementos básicos del habla llamados fonemas. El modelo acústico es uno de los principales componentes del procesamiento de voz. El motor de reconocimiento también incluye algoritmos de aprendizaje internos. Los fonemas son las unidades del habla, que se utilizan para coincidir con la voz y la capture el conjunto de micrófonos de Kinect. El modelo de lenguaje es el segundo componente principal del motor de reconocimiento de voz. En el modelo de lenguaje se analiza el contenido del discurso y trata de que coincida con la palabra mediante la combinación de los fonemas dentro de un diccionario digital incorporado. Combina los fonemas creados por el módulo de acústica con cada palabra y los compara con el diccionario digital incorporado.

Si la palabra existe en el diccionario, el motor de reconocimiento de voz reconoce lo que el usuario ha dicho. El motor de reconocimiento de voz también utiliza listas de lenguaje léxico para un gran número de palabras en el reconocimiento de voz, y proporciona información sobre cómo pronunciar cada palabra. Esto ayuda a un mejor reconocimiento de voz, de acuerdo con el factor de la pronunciación.

Figura 22. **Funcionamiento de reconocimiento de voz**



Fuente: http://dailydotnettips.com/2014/01/06/dynamically-loadingunloading-grammar-kinect-speech-recognition-engine/. Consulta: diciembre de 2016.

El motor de reconocimiento de voz coincide con los patrones sensibles al contexto, que es de gran ayuda para hacer coincidir las palabras muy similares, como "su" y "tu". Aunque estas dos palabras suenan muy similares, los significados son totalmente diferentes.

4.6. Módulos de conocimiento de voz

Por ejemplo, la palabra "bienvenida", es captada por el conjunto de micrófonos y la convierte en una señal digital. El modelo acústico del motor de reconocimiento de voz, inicia separando la palabra en los fonemas que pronuncian "bien", "veni", y "da". A continuación, el modelo de lenguaje combina cada uno de ellos y trata de que coincidan con el diccionario incorporado. En el diccionario, hay varias palabras que se incluyen en una combinación de la misma serie de fonemas. Como ya se ha definido la palabra " bienvenida", el motor de reconocimiento de voz devolverá un valor con la palabra coincidente.

El SDK de Kinect utiliza la biblioteca de voz del sistema operativo para el reconocimiento de voz. El SDK de Kinect instala el paquete de reconocimiento de voz en el sistema operativo junto con todos los demás componentes necesarios. Microsoft Speech API (SAPI) funciona como un mediador y proporciona una interfaz entre la aplicación y el motor de reconocimiento de voz. Al instalar el SDK de Kinect, debe asegurarse de que no tiene instalado ninguna versión anterior de un paquete de idioma de voz. Si es así, primero se debe desinstalar, de lo contrario la instalación no continuará. Para desarrollar cualquier aplicación habilitada para el reconocimiento de voz, es necesario desarrollar los siguientes pasos:

- Activar la fuente de audio de Kinect.
- Identificar el dispositivo que realizara el reconocimiento de voz.
- Iniciar el dispositivo de reconocimiento de voz.
- Comenzar a capturar datos de audio.
- Definir la gramática.
- Conectar la fuente de audio de voz al dispositivo de reconocimiento de voz.
- Registrar el controlador de eventos para el reconocimiento de voz.
- Gestionar los diferentes eventos invocados por el motor de reconocimiento de voz.

Microsoft Speech API proporciona la clase SpeechRecognitionEngine que trabaja como una columna vertebral para la aplicación habilitada para el reconocimiento de voz. El sistema de reconocimiento no es más que el sistema de ejecución instalado para el reconocimiento de voz. Incluye el modelo acústico, modelo de lenguaje, y toda información esencial para el reconocimiento de voz.

Para empezar a utilizar el SpeechRecognitionEngine primero se debe identificar el Kinect. La clase SpeechRecognitionEngine acepta un canal de audio desde el Kinect y lo procesa. En primer lugar, la clase SpeechDetected se activa si el audio parece ser una conversación. La clase SpeechHypothesized, analiza varias veces el audio cuando las palabras que detecta aparecen tentativamente. Esto sucede cuando intenta emparejarlos con el diccionario incorporado. Por último, SpeechRecognized se activa cuando encuentra la conversación capturada en el diccionario. El Kinect también proporciona la información basada en la confianza del nivel de la fuente de sonido en el habla que fue identificada. Si la conversación es detectada, pero no coincide correctamente o tiene un nivel de confianza muy bajo, el controlador de eventos SpeechRecognitionRejected se activará.

La forma sencilla de utilizar el reconocimiento de voz es crear una instancia de la clase SpeechRecognitionEngine. Para ello, primero se debe agregar el archivo Microsoft.Speech.dll como un conjunto de referencia y luego añadir la clase Microsoft.Speech.Recognition. A continuación, se puede adjuntar un controlador de eventos SpeechRecognized, que abrirá cada vez que el audio es detectado.

El controlador de eventos SpeechRecognizer tiene una propiedad de tipo resultado, la cual es la clase RecognitionResult. La propiedad de RecognitionResult contiene lo que se ha identificado incluyendo la calidad en términos de nivel de confianza. A continuación, se muestra algunos de los nombres de propiedad y sus usos:

- Texto: devuelve el texto identificado por el motor de reconocimiento de voz.
- Confianza: este es el valor asignado por el reconocimiento de voz al aceptar un comando. Es un valor flotante que oscila entre 0 y 1.

- Alternativas: esto devuelve las coincidencias alternativas para el sonido de entrada. El regreso es el valor único de la opción RecognizedPhrase.
- Audio: esta propiedad devuelve el audio original que se está utilizando para reconocer los comandos. Se puede obtener este audio con la ayuda de la clase RecognizedAudio.
- Palabras: esto devuelve las palabras reconocidas por el reconocimiento de voz. Palabras es una colección de la clase de tipo RecognizedWordUnit. Esto útil cuando se trata de que una oración coincida con el diccionario.

4.7. Construyendo la gramática

Se ha abordado el funcionamiento del reconocimiento de voz y los eventos que suceden cuando es emparejado o rechazado. Se debe saber qué reconocer cuando se desea realizar una operación de reconocimiento de habla. De tal manera que se debe saber cuáles son los discursos o mandatos y con quién se deben combina. Es pertinente también definir dichos comandos en la aplicación. Finalmente, también se debe saber cómo deberá reconocer el Kinect determinado comando de voz.

Esto se puede hacer mediante la especificación de la gramática basada en el reconocimiento de voz de especificación de la gramática. La gramática es el conjunto de comandos que indican al motor de reconocimiento de voz lo que debe coincidir. Podría ser una palabra o una frase. Por ejemplo, si desea iniciar o detener un programa simplemente diciendo "iniciar" o "detener", es necesario definirlos como gramática para que el motor de reconocimiento del conjunto de gramáticas devuelve la coincidencia que estaba buscando. Utilizando la biblioteca de Microsoft Speech API, hay dos maneras de construir comandos de gramática explicados anteriormente.

4.8. Uso de *Choice* y *GrammarBuilder*

La clase GrammarBuilder utiliza la clase Choice para construir las bases gramaticales de una manera declarativa. La clase Choice proporciona un conjunto de alternativas para gramática de reconocimiento de voz. Por ejemplo, cuando se desea dibujar un objeto con un conjunto específico de colores mediante un discurso de mando, primero se definen las opciones que se pueden escoger. Es posible interactuar con la clase GrammarBuilder para crear un tipo especial de secuencia de la gramática como requisito.

Con la declaración de la gramática con el objeto de GrammarBuilder, se puede construir una gramática declarativa que acepta una entrada de dos palabras, donde la primera palabra tiene cuatro posibilidades de colores: rojo, verde, azul y amarillo. La segunda palabra tiene cuatro tipos de objetos: circulo, triangulo, rectángulo y cuadrado. En estos casos, se reconocerán los comandos "círculo rojo" y "cuadrado verde", pero "rectángulo rojo" no será reconocido, si se elige que las dos primeras frases sean reconocidas para ejecutar determinado comando.

La gramática también puede seguir las reglas de reconocimiento de voz adicional, donde el elemento de gramática coincide con cualquier entrada a la secuencia actual. Co estas reglas, se puede decir algo para el elemento actual, como, "dibujar círculo rojo" y "necesidad cuadrado azul" se reconocerá así la primera palabra como una regla de reconocimiento de voz adicional, ya que las palabras anteriores fueron declaradas para determinado uso. De forma alternativa, la gramática se puede construir cargándola desde un documento llamado Especificación Gramática de Reconocimiento. Este es un documento XML con un conjunto de reglas para especificar la gramática.

El objeto GrammarBuilder simplemente construye la gramática; para usar la gramática que necesitamos crear una instancia de la clase de gramática con la instancia creada de GrammarBuilder y del mismo modo, puede crear la gramática de un documento XML. Una vez se ha definido la gramática, se carga al Kinect. Se debe cargar el ejemplo de la gramática con la instancia creada del GrammarBuilder en el Kinect. Se puede crear diferentes constructores de gramática y cargarlos en el motor de reconocimiento de voz uno a uno, esto ayuda para crear un conjunto versátil de comandos.

La clase SpeechRecognitionEngine posee el método llamado UnloadGrammar. Utilizando este método se pueden descargar las gramáticas cuando sea necesario. Incluso se puede usar el método UnloadAllGrammars para descargar todas las gramáticas. Una vez que se descarga la gramática, hay que volver a cargarla para su uso posterior.

5. RECONOCIMIENTO DE GESTOS

Esta es una de las características de Kinect. La cantidad de innovación e investigación que se ha desarrollado en esta tecnología de reconocimiento de gestos es incomparable. Los gestos pueden ser descritos como acciones corporales que transmiten un mensaje. Podría ser la agitación de las manos, mover las muñecas, o una acción que incluya múltiples partes del cuerpo. El reconocimiento de gestos es la tecnología utilizada para identificar gestos y convertirlos en una forma que puede ser reconocida por los dispositivos basados en ella. El reconocimiento de gestos se basa en varios algoritmos matemáticos y el seguimiento del esqueleto humano para reconocer y clasificar los gestos. El reconocimiento de gestos trabaja principalmente en una interfaz natural de usuario.

El reconocimiento de gestos proporciona una integración perfecta entre la interfaz natural de usuario y el Kinect. Se explorarán diferentes conceptos de reconocimiento de gestos con el SDK de Kinect, incluyendo los diferentes enfoques del reconocimiento de gestos y emplearlos en una serie de aplicaciones. Se deben cubrir los siguientes temas para el análisis del reconocimiento de gestos:

- Fundamentos del reconocimiento de gestos.
- Diferentes enfoques para reconocer gestos.
- Comprender el reconocimiento de gestos básicos y algorítmicos.
- Introducción rápida a la red ponderada y basada en plantillas de enfoques de reconocimiento.

- Implementación de aplicaciones y controles habilitados para gestos.

Un gesto es un movimiento del cuerpo humano o la acción que pretende comunicar un mensaje. Estos gestos permiten que la aplicación sepa lo que se desea hacer. En el contexto del SDK de Kinect, un gesto puede interpretarse como una acción por la cual el usuario transmite algunos mensajes o información a la aplicación. Es similar al concepto de escribir en un teclado, dibujar con la ayuda de papel o mediante el uso de un dispositivo táctil. En todos estos casos, la entrada estaba destinado a un propósito particular, que el dispositivo necesita entender y luego proporcionan la salida deseada interactuando con la aplicación. Del mismo modo, un gesto actúa como una entrada para el Kinect. Con base en esta entrada, la aplicación necesita realizar ciertas funciones. En realidad, no hay conexión entre los usuarios y el dispositivo, por lo tanto, esta tecnología forma la columna vertebral de la interfaz natural de usuario para Kinect.

5.1. Enfoques para el reconocimiento de gestos

El reconocimiento de gestos es uno de los procesos más interesantes, y debido a los diferentes cálculos, algoritmos, enfoques y metodologías. El SDK de Kinect no proporciona ninguna API integrada para reconocimiento de gestos. Por lo tanto, depende totalmente de los desarrolladores definir sus propios enfoques y escribir su propia lógica para reconocer e interactuar con los gestos. Los enfoques pueden variar dependiendo de los gestos que se elijan y cómo se utilizan en la aplicación, de esta manera se convierten los gestos de simples a complejos. Los enfoques para el reconocimiento de gestos se pueden clasificar de la siguiente forma:

- Reconocimiento básico de gestos

- Enfoque algorítmico
- Enfoque de redes ponderadas
- Enfoque basado en plantillas

La elección entre los enfoques de reconocimiento de gestos depende, totalmente, del desarrollador y los requisitos de la aplicación. A veces, un gesto para una aplicación podría ser simple, por ejemplo, levantar ambas manos, midiendo distancias entre las articulaciones. Pueden ser gestos avanzados, como una rotación de movimiento con ambas manos. También pueden ser tan complejos como hacer ejercicios de salto o balanceo de una raqueta. Antes de desarrollar cualquier aplicación, se deben conocer los requisitos, implicaciones del gesto, precisión requerida, aceptación del proceso, retraso, y el marco de tiempo dado para el desarrollo del gesto.

El usuario interactúa con el Kinect, que capta las acciones del usuario y los pasa a la aplicación como un canal de datos del esqueleto humano. Las aplicaciones basadas en el gesto incluirán un componente llamado el motor de reconocimiento de gestos, que reconocerá los gestos basados en las acciones del usuario y el enfoque definido en la aplicación. Al reconocer los gestos, la aplicación puede realizar la acción necesaria y notificar al usuario. El motor de reconocimiento típicamente realiza las siguientes tareas:

- Acepta las acciones del usuario en forma de datos esqueleto.
- Combina los puntos de datos con una lógica predefinida para un gesto específico.
- Ejecuta acciones si el gesto es reconocido.
- Responde a los usuarios.

Es esencial entender el seguimiento de esqueleto con el SDK de Kinect para trabajar con el motor de reconocimiento de gestos.

5.2. Reconocimiento básico de gestos

El enfoque fundamental del reconocimiento de los gestos es interactuar con el esqueleto humano y aplicar la lógica básica para realizar alguna acción. En el gesto básico la detección del mismo depende de un conjunto predefinido de condiciones, conocidas como el conjunto de resultados. Si la acción realizada coincide con el conjunto de resultados, se puede decir que el usuario realizó un gesto, de lo contrario, no. La implementación de los gestos básicos es relativamente fácil y directa. Sin embargo, este enfoque es la base fundamental de la construcción de cualquier tipo de gesto controlado.

La detección de gestos depende de las articulaciones de seguimiento del esqueleto humano porque se basa en definir la condición de los gestos basados en las articulaciones. A continuación, se describe, brevemente, la representación de las articulaciones del esqueleto.

La unión de cada uno de los puntos del esqueleto humano se mide en un espacio tridimensional (X, Y, Z). Las coordenadas X y Y especifican la ubicación de la articulación en el plano, cuando el Kinect está en la dirección Z. Cuando una articulación está representada con coordenadas X, Y y Z en una dimensión tridimensional, las coordenadas X e Y, realmente, indican la ubicación conjunta en el plano, y Z indica la distancia del punto del objeto hasta el Kinect.

Si las articulaciones se mueven desde la derecha a la izquierda o viceversa, el eje X de los puntos cambiará. De forma similar, para mover las articulaciones en dirección ascendente o descendente, el eje Y cambiará. Los

cambios en el eje Z reflejarán si las articulaciones avanzan o retroceden respecto al Kinect. Los cálculos para los gestos básicos se pueden hacer con cualquier de los siguientes métodos:

- Calcular la distancia entre diferentes juntas.
- Comparar las posiciones de las articulaciones y la desviación entre las articulaciones.

5.3. Cálculo de la distancia entre dos articulaciones

La representación del esqueleto humano es tridimensional; sin embargo, antes de analizar el espacio de coordenadas 3D, primero se consideran los puntos en un plano de coordenadas 2D con X y Y, para ver cómo calcular la distancia entre dos puntos. En matemáticas generales, para calcular la distancia entre dos puntos, se utiliza el Teorema de Pitágoras cuyo enunciado es que, para un triángulo rectángulo, el cuadrado de la hipotenusa es igual a la suma de los cuadrados de los otros dos lados.

Con el teorema de Pitágoras se calcula la distancia entre dos puntos, por ejemplo, se tiene un punto A (X1, Y1) y un punto B (X2, Y2) en un plano bidimensional de un plano coordinado. Si se desea calcular la distancia "d" entre el punto A y el punto B, el teorema de Pitágoras, indica que primero se debe trazar una línea paralela al eje X del punto A y otra línea desde el punto B, que es paralela al eje Y. Se consideran las líneas reunidas en el punto C (X2, Y1). Los ejes X e Y son perpendiculares a cada uno; el triángulo formado por los puntos A, B, y C es un ángulo recto.

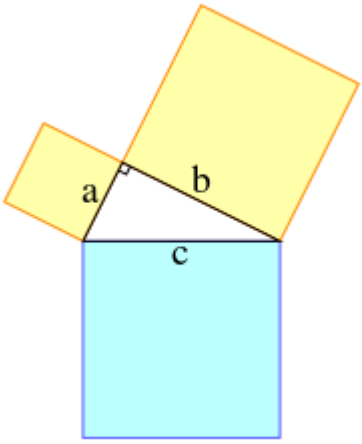
El valor de "d", la distancia entre los puntos A y B, será la hipotenusa del triángulo rectángulo formada por los puntos A, B, y C. La distancia entre A y B

se puede calcular por medio de la fórmula: El teorema de Pitágoras funciona bien para espacios tridimensionales y la distancia entre los puntos (X1, Y1, Z1) y (X2, Y2, Z2) puede ser calculada por la siguiente fórmula:

$$c = \sqrt{x^2 + y^2 + z^2}$$

El teorema de Pitágoras muestra dónde se realiza el cálculo de la distancia en los puntos deseados.

Figura 23. **Teorema de Pitágoras**



Fuente: https://es.wikipedia.org/wiki/Teorema_de_Pitágoras. Consulta: febrero de 2017.

El método GetJointDistance() acepta dos puntos del plano como argumentos. Mide la distancia de los puntos para cada eje en el sistema de coordenadas y, finalmente, regresa la distancia entre los puntos que representa las articulaciones. Math.Sqrt devuelve el valor de la raíz cuadrada de un número determinado, y Math.pow devuelve el valor de la potencia cuadrada de la distancia determinada. La propiedad TrackingState de las articulaciones se comprueba antes de llamar al método GetJointDistance() para asegurarse de

realizar un seguimiento de las articulaciones y tiene posiciones válidas para calcular la distancia.

Es posible que Kinect no pueda ver las articulaciones y, por lo tanto, TrackingState no será rastreado. Sin embargo, es posible que el Kinect puede inferir la posición. Por lo tanto, también puede utilizar el código basado en TrackingState == JointTrackingState.Inferred y calcular la distancia con una de las articulaciones. El reconocimiento de gestos basado en un cálculo entre las articulaciones es un patrón que se realiza mediante cálculo.

5.4. Reconocimiento de gestos algorítmicos

El reconocimiento del gesto algorítmico se construye sobre el reconocimiento básico de gestos, por lo tanto, la comprensión del reconocimiento básico del gesto es esencial para seguir adelante con el enfoque algorítmico. El enfoque algorítmico utiliza un conjunto de condiciones predefinidas y parámetros para detectar y validar un gesto contra cada uno de ellos. Con el enfoque algorítmico, básicamente, se valida un gesto como se genera, asegurando que los puntos de inicio, las restricciones, los parámetros y los puntos finales son siempre válidos.

Se puede considerar que los gestos se miden mediante un enfoque algorítmico cuando los gestos necesitan ser validados contra múltiples condiciones, las articulaciones múltiples involucradas, y donde se necesitan medir los múltiples estados de gesto. Si la aplicación necesita todas las condiciones para ser validada y medida con base en el tiempo o marcos, se debe seguir con el enfoque algorítmico. Primero se deben considerar las condiciones de frontera, los criterios de entrada y salida para los gestos y los estados de validación. En general, la mayoría de las aplicaciones siguen el

algoritmo para la implementación de gestos regulares. Los siguientes son algunos ejemplos de gestos que pueden ser considerados como algorítmicos:

- Movimiento en la misma dirección
- Un desplazamiento a la derecha o hacia la izquierda
- Acercamiento o alejamiento
- Manos agitando

De los tipos de gesto mencionados, se puede entender cómo el enfoque podría ser diferente de la detección de gestos básicos. El enfoque algorítmico reconoce los gestos y rastrea si el gesto se realiza correctamente o no. Aunque están estrechamente relacionados entre sí, se puede asegurar que todos los enfoques son un conjunto más pequeño de enfoques algorítmicos. Los gestos algorítmicos son más avanzados y se calculan con diversas condiciones y parámetros. En resumen, se elige el enfoque algorítmico cuando la aplicación necesita jugar alrededor de una serie de puntos. Esto implica un conjunto de cálculos para cada uno de los marcos con criterios de inicio, validación de diferentes estados y criterios finales.

Comprender el gesto algorítmico en enfoque de detección, para comprender aún mejor, se articularán los componentes que se requieren para el reconocimiento de gestos algorítmicos en una lista:

- Comienzo
- Condición
- Validación
- Terminación

Para empezar con cualquier gesto, siempre habrá una posición inicial. Este es el punto de entrada para cualquier gesto y tiene que ser validado antes de otras posiciones. Una vez validada la posición de inicio y el gesto por el usuario final, cada uno de los cuadros tiene que ser validado bajo el predefinido "condición" para los tipos de gestos en particular. Si cualquiera de estas condiciones falla durante el ciclo de ejecución completo, se puede detener el seguimiento de gestos y esperar a que se inicie de nuevo. Por último, debe haber una condición que desencadene el final del gesto y "valida" la posición final, lo que indica que el gesto reconocimiento está "terminado".

Por ejemplo, si se necesita realizar un movimiento hacia la izquierda usando su mano derecha:

- Antes de comenzar, la articulación de la mano izquierda debe estar por debajo del codo izquierdo y la columna vertebral. Además, la articulación derecha debe estar por debajo de la articulación del hombro derecho y por encima de la articulación del codo derecho.
- El usuario debe mover la mano de derecha a izquierda mientras mantiene la derecha la mano izquierda sin movimiento.
- Esta condición debe ser validada para un par de marcos predefinidos, y debe resultar exitosa cada vez que comprueba si la mano se está moviendo a la izquierda.
- Después de un número específico de cuadros cuando el gesto alcanza para validar la última condición, verificará si la distancia entre la articulación de la mano derecha El hombro izquierdo se ha reducido desde el punto de partida.

Al tener en cuenta los puntos anteriores para cada cuadro esqueleto, se puede determinar si un movimiento hacia la izquierda se ha detectado o no. La

validación se debe realizar para todos y cada cuadro, y se puede añadir un tiempo o número de fotogramas para validar. En esta sección se explicará cómo implementar el motor de reconocimiento de gestos siguiendo los enfoques anteriores, adicionalmente, se ampliará la clase GestureRecognizer desarrollada anteriormente.

Para tomar otro aspecto simple en el enfoque algorítmico reconocimiento de gestos, y para simplificar la implementación, se puede desfragmentar el proceso en varios bloques de estado en varios módulos más pequeños, y se les denominará "fases". Cada fase de un gesto tendrá su propio conjunto de resultados o condiciones que medir el éxito o fracaso de la fase. El resultado de las fases dependerá unos de los otros. Esto significa que el reconocimiento se moverá a la siguiente fase si el resultado fase previa fue aprobada. También podría suceder que todas las condiciones de una fase no están satisfechas.

Esto no siempre significa que la fase ha fallado, podría ser que el usuario se encuentra "en pausa" o "en curso" en esa posición particular por algún tiempo. Se puede marcar el estado de la fase como la pausa y esperar a la próxima acción de unos cuantos fotogramas. Estas fases pueden comunicarse entre sí se utiliza una interface de comunicación, para compartir la información, resultados, y los datos entre sí.

5.5. Reconocimiento de gestos en red ponderada

La detección de las posturas utilizando redes ponderadas es uno de los enfoques avanzados en la detección de las posturas. En esta sección se conocerá este enfoque y su funcionamiento. Se ha explicado que la mayoría del reconocimiento de los gestos se realiza con base en movimientos de las articulaciones del esqueleto humano, y para todos los casos anteriores, las

articulaciones eran claramente visibles para el Kinect (que se ha verificado con el estado de los puntos de seguimiento). Algunos movimientos del cuerpo son muy dinámicos en la vida cotidiana. Por ejemplo, los ejercicios de flexión hacia delante, hacia los lados, que se inician con una posición recta, luego subir lentamente las manos hacia delante, y tratar de tocar los dedos de los pies, flexión hacia adelante, o de pie con las piernas separadas, manteniendo una mano en la cintura y elevar la otra mano. Por otro lado, intentando realizar ejercicios de flexión con el cuerpo hacia los lados, en estos casos, los movimientos del cuerpo son dinámicos ya que cada vez que un usuario no pueda llegar a la misma posición, exactamente, dos veces.

Por otra parte, un usuario no puede realizar movimientos y posiciones perfectamente con las manos y piernas. También hay una buena probabilidad de puntos superpuestos o también podría suceder que algunas de las articulaciones se van fuera del rango visible de Kinect. El ejercicio de salto, a primera vista, parece ser uno de los más simples gestos de reconocimiento, pero ese no es el caso. Similar a ejercicios de flexión, un ejercicio de salto se puede variar con base en la forma como un usuario salta, por ejemplo, saltos normales, entrenamiento de la cuerda, salto de altura, y otros, y se basa en el movimiento que, a su vez, podría causar un problema de visibilidad de las articulaciones ya que cada usuario puede hacerlos de diferentes maneras.

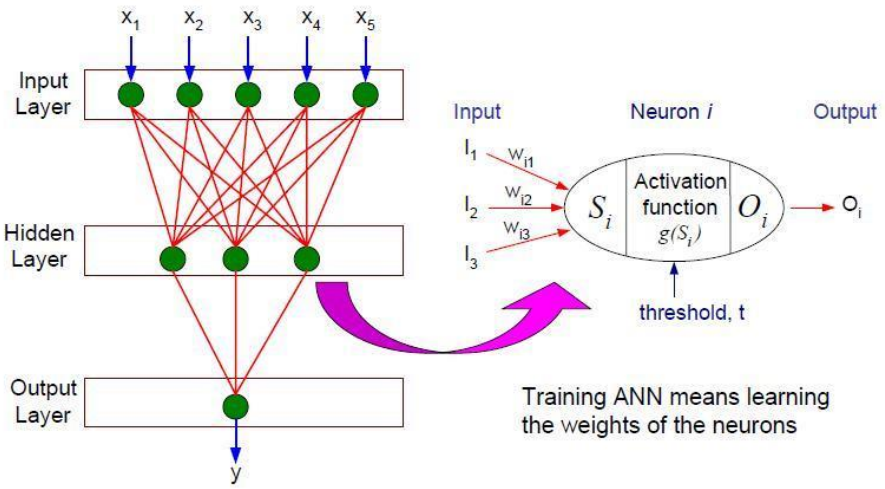
Si desea implementar los ejercicios mencionados con gestos realmente se necesita pensar mucho en estos ejercicios. Esto se debe a que son difíciles y no solo se pueden proporcionar algunos cálculos sobre las articulaciones. Por lo tanto, se necesita alguna alternativa que puede decir en qué medida los usuarios están realizando el ejercicio correctamente, en lugar de solo decir sí o no. En términos técnicos, se necesita una estructura de datos flexible que puede calcular las probabilidades y puede llegar a una decisión basada en las

entradas del usuario. Uno de los mejores enfoques para resolver este tipo de problema ingeniería informática es una red neuronal.

5.6. Red neuronal

Las redes neuronales son un conjunto de nodos (llamados neuronas) conectados entre sí como un grupo. Esto es similar a la del cerebro humano y la sinopsis que forman. El concepto de la red neuronal procede de las neuronas humanas y se utiliza sobre todo para el campo de la biología de redes neuronales de los seres humanos. En las ciencias de la computación, se usan como una construcción de máquina de aprendizaje que tiene una capa de entrada conjunto de nodos de entrada / neuronas y una capa de salida conjunto de nodos de salida / neuronas. Las capas de entrada y de salida están conectados en alguna parte entre ellas. Esta capa se denomina oculta o abstracta, que tiene otro conjunto de toma de decisiones de nodos. La capa abstracta consta de otro conjunto de nodos decisión de fabricante que están conectados entre sí, y en la combinación de sus decisiones resultado se puede dar de la siguiente manera.

Figura 24. **Red neural de toma de decisiones**



Fuente: http://mines.humanoriented.com/classes/2010/fall/csci568/. Consulta: febrero de 2017.

En el diagrama anterior se muestra una red neuronal muy básica, con dos nodos de entrada (x1 al x5) y un nodo de salida (y). La capa intermedia es una capa abstracta con un conjunto de nodos más pequeños que están conectados entre sí. Estos nodos más pequeños funcionan como los tomadores de decisiones; basándose en la entrada proporcionada por x1 al x5, la salida y este será recibido por la aplicación. Esto es similar a la naturaleza de las neuronas en el cerebro humano.

5.7. El reconocimiento de gestos con las redes neuronales

La detección de las posturas en una red neural siempre se basa en la relación de probabilidad en vez de los valores exactos. La salida sería decir que el usuario ha realizado determinado porcentaje del gesto correctamente, y cierto porciento no realizado correctamente. Con base en esta relación, se toma una decisión.

Cada nodo dentro de la red es un algoritmo para evaluar pequeños elementos de un gesto o movimiento. Salida de un nodo (resultado de la acción) decidirá qué nodos mover. Por lo tanto, el flujo de secuencia en este enfoque nunca será la misma. Los nodos están conectados a través de un enlace, y cada eslabón tiene un valor llamado asociado peso. Este valor define el peso que algunos enlaces son más importantes que los demás. El propósito de tener pesos a cada uno de los nodos es tomar decisiones más precisas. Con base en el valor del peso medido, los gestos van al siguiente paso, los pesos superiores siempre gozan de preferencia. Al final de la red completa, se tiene un cálculo resultante llamado de salida calculado por la acción de entrada del usuario.

Esto corresponde con la salida esperada predefinida o la salida de los mejores valores por el gesto detectado. A medida que estos resultados se calculan en función de la probabilidad, que es muy común que el resultado nunca puede ser exactamente el mismo y tendrá por lo menos alguna diferencia o error. Si la diferencia está dentro de un límite predefinido llamado umbral, se aceptará el resultado, de lo contrario la red llevará a cabo la operación varias veces para producir resultados gesto de acción correctas o muy cerca.

5.8. El reconocimiento de gestos basado en plantillas

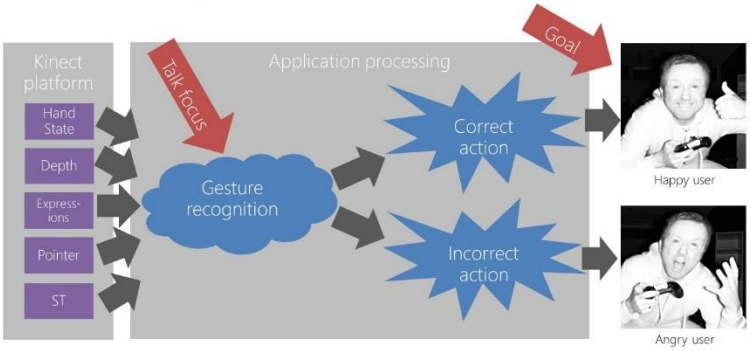
El reconocimiento de gestos basados en plantillas también se conoce como gestos basadas en patrones. Este gesto puede, potencialmente, ser utilizado cuando no estamos seguros de la solidez del reconocimiento de la red neuronal. Con este sistema de reconocimiento, el motor de reconocimiento de gestos coincide los movimientos del usuario con gestos predefinidos, y sirve para medir correctamente qué será capturado. En este enfoque, los gestos se registran y se almacenan primero en una ubicación de la manera normal. Mientras que empareja los gestos, el mismo conjunto de acciones de los

usuarios se toman como parámetros de entrada y validado frente a los datos almacenados. El resultado final es accionado a partir de una relación de probabilidad, haciendo coincidir los datos entre los datos existentes y el conjunto de datos que se está realizando actualmente. En general, la plantilla basada sistema de reconocimiento implica las siguientes fases:

- Creación de plantillas
- Seguimiento del gesto
- Comparación de plantillas

La primera fase del sistema de reconocimiento basado en la plantilla es la creación de plantillas. En esta fase, los gestos se registran y almacenan información de articulaciones junto con los metadatos. Los metadatos contienen el nombre del gesto, tipo de gestos, intervalo de tiempo para los gestos, duración mínima y máxima. La segunda fase es el seguimiento del gesto cuando el usuario lleva a cabo las acciones frente a Kinect y la aplicación pasa la información al reconocimiento de gesto. Cuando el gesto termina, se compra con un conjunto predefinido de plantillas para averiguar si el gesto es una de esas plantillas almacenadas, esta fase se conoce como la comparación de plantillas y se representan en el diagrama siguiente:

Figura 25. **Fases de reconocimiento de gestos**



Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: febrero de 2017.

Como se muestra en el diagrama anterior, el seguimiento del gesto y la comparación de plantillas se trabajan simultáneamente. Mientras el usuario está realmente realizando la acción, los datos se pueden emparejar varias veces para ver si se correspondan correctamente. Para almacenar datos se puede utilizar cualquier base de datos, XML o archivos. También puede elegir su propio algoritmo para que coincida con los datos de la ubicación de la tienda de Microsoft; porque la mayoría de las veces se obtiene el resultado exacto en concordancia de todos los puntos. Si no se obtienen los resultados que coinciden se prefieren los valores más cercanos para seleccionar los correctos.

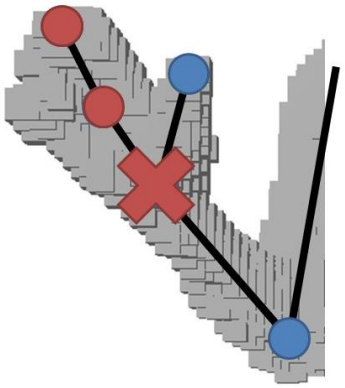
La detección de posturas suena muy inspirador y motivador, y puede impulsar a los desarrolladores a utilizar su imaginación y creatividad, sin embargo, hay algunas opciones que se necesitan considerar en la utilización de una aplicación de gesto habilitado:

- Las acciones de usuario
- Desarrollo
- Datos coincidentes

- Pruebas

Las acciones del usuario o entradas son los elementos clave para cualquier aplicación de gesto habilitado. Las entradas erróneas pueden inducir a error a la aplicación si no se manejan correctamente. Por lo tanto, el usuario debe saber qué acción realizar. Esto puede lograrse mediante la capacitación o proporcionando retroalimentación en vivo en la interfaz de usuario de la aplicación. Durante el desarrollo, se deben capturar todas las condiciones entrada que vienen del usuario. Además, debe validar las condiciones de contorno para los criterios de entrada y salida de los gestos.

Figura 26. **Captura de gesto con una mano**



Fuente: https://channel9.msdn.com/coding4fun/kinect/Programming-Kinect-for-Windows-v2-Jumpstart-on-Demand. Consulta: febrero de 2017.

Si la entrada del usuario coincide con el conjunto de datos, se debe recurrir a medidas de gesto reconocido, de lo contrario, enviar el mensaje de error al usuario. Esto hace que su desarrollo sea un poco complejo ya que se debe comprobar si el gesto es reconocido exitosamente o no. Antes de empezar el desarrollo, se debe estar al tanto de la línea de tiempo disponible para la ejecución, ya que, incluso, un gesto fácil puede tomar una buena cantidad de tiempo en función de los criterios y la validación de los escenarios.

La depuración juega un papel importante durante el desarrollo de aplicaciones con gestos, siempre es mejor grabar los gestos utilizando Kinect si los necesita. Asimismo, el uso del condicional de los puntos de interrupción puede hacer la vida aún más fácil, en la depuración de la aplicación.

Si la aplicación es todo sobre la correspondencia de los datos y la comparación de patrones con gestos, la recopilación de datos juega un papel crucial. Por lo tanto, se debe estar seguro que los datos de la sentencia se normalizan con todos los factores en los que el usuario puede realizar. Además, puede existir una discrepancia entre los datos almacenados y los que se toman en tiempo real, afectando la calidad de reconocimiento. Así que asegúrese de que está utilizando el mejor algoritmo posible para hacer coincidir los datos. Se trata de cómo se puede hacer que su aplicación perfecta. Probar aplicaciones gesto habilitado requiere una buena cantidad de tiempo y esfuerzo.

CONCLUSIONES

1. Se analizó a detalle cada componente fundamental en el funcionamiento de Kinect, como la cámara de color, cámara de profundidad, proyector de infrarrojos y conjunto de micrófonos que, capturan la información que Kinect necesita para realizar los diferentes tipos de reconocimiento que puede realizar.

2. Se detallaron las versiones de Kinect, así como sus características en hardware, capacidades y límites. Tanto cuando está en el reconocimiento del Kinect v1 y Kinect v2. Adicionalmente, se detallaron sobre como son los datos obtenidos por el Kinect en el reconocimiento en la captura de imágenes de color, profundidad reconocimiento facial y de gestos.

3. El procesamiento de datos de color y profundidad están estrechamente relacionadas entre sí porque para realizar el reconocimiento de gestos, esqueleto humano y facial, el Kinect debe utilizar ambos en todo momento, realizando el cálculo de distancia y seguimiento de esqueleto humano

4. El procesamiento de audio en el Kinect es una parte fundamental cuando se reconoce un usuario mediante voz y cuando este da instrucciones al Kinect para realizar determinadas, el procesamiento de voz realiza mediante el conjunto de micrófonos instalados en el Kinect, cada uno de ellos captura en diferente tiempo el sonido. Así se logra ubicar la fuente de sonido, para la conversión analógica - digital de Kinect utiliza

convertidores de alta calidad que brindan una gran fiabilidad del sonido capturado para su interpretación en el programa donde se esté utilizando.

5. El reconocimiento de gestos es una de las partes más importantes del Kinect ya que por ella se debe la última etapa de reconocimiento del Kinect en la cual se debe tener el resultado del reconocimiento de imagen y voz, para, posteriormente, Kinect pueda interpretar cierto movimiento o acción que el usuario realice.

RECOMENDACIONES

1. Trabajar con la resolución de 640 por 940 pixeles cuando se utiliza el Kinect V1, ya que la misma indica que es la ideal para el mismo, si se trabaja con la resolución de 1 280 por 720 pixeles se ve comprometido el rendimiento, a diferencia del Kinect V2 que puede trabajar con la resolución 1 280 por 720 pixeles sin inconvenientes.

2. Trabajar con el entorno de programación de la empresa desarrolladora y no utilizar software de terceros, como los software libres, los cuales pueden ocasionar un daño o mal funcionamiento de Kinect, de preferencia utilizar el Kinect Studio donde se pueden utilizar todas las funcionalidades de Kinect sin necesidad de programar.

3. El Kinect se debe encontrar en el mayor espacio libre de objetos para realizar de la mejor manera el reconocimiento, ya que, al detectar varias personas y objetos, el Kinect se confundirá si encuentra muchos obstáculos en su rango de visión.

4. Considerar a la hora de programar, actualizaciones en los entornos de programación dedicados para el Kinect, ya que los mismos no deben dejar de lado clases o librerías a la hora de programar el Kinect.

5. Tener precaución cuando Kinect realiza el reconocimiento de cuerpo, no realizar movimientos bruscos, ya que estos tienden a confundir al Kinect al no presentarse como un movimiento normal.

BIBLIOGRAFÍA

1. ABHIJIT, Jana. *Kinect for Windows SDK Programming* Guide. 1a ed. Reino Unido: Packt Publishing Ltd, 2012. 357p.

2. CLEMENTE, Giorio. *Kinect in Motion – Audio and Visual Tracking by Example*. 1a ed. Reino Unido: Packt Publishing Ltd, 2013. 95p.

3. BORENSTEIN, Greg. *Making Things See*. 1a ed. Canada: O'Reilly Media, Inc, 2012. 411p.

4. CATUHE, David. *Programming with the Kinect for Windows Software Development* Kit. 1a ed. Estados Unidos de América: Microsoft Press, 2012. 201p.

5. MILES, Rob. *Start Here! Learn the Kinect API*. 1a ed. Estados Unidos de America: O'Reilly Media, Inc, 2012. 251p.