



UNIVERSIDAD
NACIONAL
DE COLOMBIA

Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción entre el computador y personas con movilidad reducida

Alejandro Marín Cano

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2017

**Diseño de un sistema detector de movimientos
usando técnicas de visión artificial 3D para la
interacción entre el computador y personas con
movilidad reducida**

**Design of a movement detection system using 3D
artificial vision techniques for the interaction
between the computer and people with reduced
mobility**

Alejandro Marín Cano

Tesis de investigación presentada como requisito parcial para optar al título de:
Magíster en Ingeniería – Ingeniería de Sistemas

Director:

Ph.D. Jovani Alberto Jiménez Builes

Línea de Investigación:

Inteligencia Artificial

Universidad Nacional de Colombia

Facultad de Minas, Departamento de Ciencias de la Computación y de la Decisión

Medellín, Colombia

2017

Agradecimientos

Dedico este trabajo principalmente a mi familia que me ha acompañado toda mi vida en cada una de las alegrías y dificultades que se me han presentado hasta ahora, también a todas las personas con las que he compartido mi estancia en la universidad durante el proceso de formación académica, ya que me han brindado su apoyo y guía para afrontar día a día todos los retos que se me presentaron como estudiante, compañero y persona.

Resumen

Hoy en día el desarrollo de nuevos productos enfocados a individuos que presentan alguna limitación es un punto clave en el proceso de inclusión y aprendizaje, ya que gran parte de la información se encuentra en medios virtuales. En la actualidad existen diversos dispositivos alternativos de entrada para computador entre los cuales se encuentran los denominados Free-hand, los cuales permiten interactuar de forma directa sin utilizar los medios tradicionales como lo son el teclado y el ratón.

Dado este contexto, a pesar de que existe una gran cantidad de dispositivos capaces de detectar el movimiento y reconocer gestos, no se cuenta con muchos desarrollos específicos sobre el área de la inclusión social a personas con limitaciones motoras, las cuales permitan el acceso de forma sencilla a los medios de aprendizaje y comunicación. En esta investigación se propone diseñar un sistema de detección de movimientos usando técnicas de visión artificial 3D, con el fin de mejorar la interacción de personas con movilidad reducida.

Palabras clave: Sensor 3D, detección de movimiento, discapacidad, aprendizaje, visión artificial.

Abstract

Nowadays the development of new products focused on individuals that present some limitation is a key point in the process of inclusion and learning, since much of the information is found in virtual media. At present there are several alternative computer input devices among which are the so-called Free-hand, which allow direct interaction without using traditional media such as the keyboard and mouse.

Given this context, although there is a large number of devices capable of detecting movement and recognizing gestures, there are not many specific developments in the area of social inclusion for people with motor limitations, which allow easy access to the means of learning and communication. In this research it is proposed to design a movement detection system using 3D artificial vision techniques, in order to improve the interaction of people with reduced mobility.

Keywords: 3D sensor, motion detection, disability, learning, artificial vision.

Contenido

	Pág.
Resumen	VII
Lista de figuras.....	1
Introducción	2
1. Presentación del Problema.....	5
1.1 Justificación	6
1.2 Objetivo General.....	6
1.3 Objetivos Específicos.....	6
2. Marco Teórico y Antecedentes.....	9
3. Solución Propuesta.....	13
3.1 Subsistema de sensado.....	13
3.1.1 Cámara RGB.	14
3.1.2 Sensor de profundidad	14
3.2 Subsistema computacional	16
3.3 Características y Especificaciones Técnicas Para la Elección del Sensor	20
3.3.1 Características del Kinect, Fabricante Microsoft y PrimeSense.....	20
3.3.2 Características del ASUS Xtion PRO LIVE, Fabricante Asus y PrimeSense.....	22
3.3.3 Elección del sensor	23
3.3.4 Requerimientos de hardware y software:	23
3.4 Esquema De Diseño	24
3.4.1 Posicionamiento del cursor	24
3.4.2 Reconocimiento de gestos	25
3.4.3 Interfaz de Usuario	26
4. Resultados.....	27
4.1 Resultados De Los Test De Funcionamiento De La Integración Del Hardware Y Software.....	28
4.2 Interfaz final	32
4.3 Validación de resultados.....	34
5. Conclusiones y recomendaciones.....	35
5.1 Conclusiones	35
5.2 Recomendaciones	36
A. Anexo: Fragmentos de código utilizados en el desarrollo de la interfaz	37

Bibliografía 43

Lista de figuras

	Pág.
Figura 2-1: Teclado en pantalla [9].	10
Figura 2-2: Array de micrófonos. CrispMic [10].	10
Figura 2-3: Dispositivo de fijación en la cabeza con sensor de posición y distancia [11].	11
Figura 2-4: Configuración del dispositivo de entrada por medio del movimiento de los ojos [12].	12
Figura 3-1: sensor Kinect [15].	14
Figura 3-2: Patrón de puntos emitidos por el sensor Kinect [15].	15
Figura 3-3: Capas del OpenNI [18].	18
Figura 3-4: Coordenadas del sistema usadas por OpenNI SDK.	24
Figura 3-5: Coordenadas del sistema usadas por C++.	25
Figura 4-1: Imagen de profundidad dada por el sensor Kinect.	31
Figura 4-2: Imagen de profundidad integrada con el SkeletonTraking dada por el sensor Kinect.	31
Figura 4-3: Imagen a color integrada con el SkeletonTraking dada por el sensor Kinect.	32
Figura 4-4: Interfaz de usuario desarrollada en C++.	33

Introducción

El desarrollar nuevos productos, que puedan satisfacer las necesidades de individuos, que presentan limitaciones las cuales interfieran con la capacidad para acceder a los computadores, constituye un factor clave en el proceso de inclusión y aprendizaje, ya que gran parte de la información se encuentra a través de medios virtuales y para interactuar con estos medios se debe acceder por lo menos a un computador y tener la capacidad de manipularlo con el fin de realizar las tareas de estudio e investigación que cada individuo desee. Estos dispositivos de entrada alternativos tienen unas características muy específicas de funcionamiento, por lo que a la hora de elegir que dispositivo es más conveniente con respecto al tipo de limitación que se presenta se requiere tener un previo conocimiento acerca de los dispositivos existentes, sus características, y el tipo de tecnología que emplean.

En la actualidad existen diversos dispositivos alternativos de entrada para computador entre los cuales se encuentran los denominados Free-hand, los cuales son controladores no convencionales compactos, que no requieren una conexión mecánica directa entre el operador y el dispositivo de entrada, son convenientes para proporcionar interfaces de manos libres que pueden complementar el control manual [1]. Hoy en día se utiliza tecnología basada en el uso de señales desde el cerebro, los músculos, la voz, los labios, la posición de la mano, posición de los ojos y los gestos para el control de los ordenadores y otros dispositivos [2],[3].

En esta investigación se diseña un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción entre el computador y personas con movilidad reducida. La principal función del desarrollo de este sistema será incentivar la participación y la inclusión de personas con discapacidad, brindando un mayor acceso a la información y la comunicación e impulsar proyectos que promuevan principios de respeto a la diferencia

por medio de estrategias pedagógicas de enseñanza activa en las que el estudiante se sienta parte del conocimiento y pueda interactuar con los diferentes conceptos en las áreas básicas como lo son la comprensión de la materia, el movimiento, la fuerza, el espacio, las estructuras y demás.

Esta investigación está estructurada de la siguiente manera: La primera sección es una revisión sobre los diferentes tipos de dispositivos free-hand existentes, las técnicas empleadas con respecto a la interacción Humano Maquina, y sus características. La siguiente sección describe los elementos que conformaran el dispositivo para la detección de movimientos a partir de visión artificial 3D, estos elementos se dividen principalmente en 2 subsistemas: el de sensado (sensor 3D) y el computacional (Interfaz de usuario). Después de esto, se realiza la interconexión de los sistemas y se continua con la evaluación y verificación de del funcionamiento del sistema ante comportamientos reales del ambiente y del usuario con la interfaz. Por último, se describe el proceso de implementación y los resultados del prototipo propuesto.

1. Presentación del Problema

En la actualidad la visión artificial se utiliza para muchas aplicaciones, entre ellas se pueden destacar la navegación robótica [4], los sistemas de vigilancia, la realidad aumentada [5] y la inclusión social [6]. Hoy en día existen dispositivos basados en la visión artificial que permiten interactuar con el computador a partir de diferentes técnicas, a saber:

- **Movimiento de los ojos:** en esta categoría se presentan varios desarrollos, como por ejemplo el BET2.0 de la empresa IRISBOND, el cual a través de un dispositivo basado en visión artificial y un aplicativo informático le permite al usuario controlar el puntero de un computador. Esta solución permite beneficiar a las personas con limitaciones de movilidad severa, que no pueden utilizar los equipos (mouse y teclado) y software convencional .
- **Detección de gestos y movimientos de las manos:** en esta categoría se presentan tanto gestos realizados por las manos como reconocimiento del movimiento del cuerpo, este tipo de sistema se basa en la detección del movimiento a través de visión estéreo la cual se puede clasificar en dos categorías: detección basada en modelos de la mano y el brazo, y la detección de los gestos de la mano (posiciones de los dedos, orientación, entre otros) [7]. Para este tipo de técnica existen en el mercado varios dispositivos entre los cuales se encuentran el Kinect, iSense™3D scanner, y el Xtion PRO LIVE.

A pesar de que existe una gran cantidad de dispositivos capaces de detectar el movimiento y reconocer gestos, no se cuenta con muchos desarrollos específicos sobre el área de la inclusión social a personas con limitaciones motoras, las cuales permitan el acceso de forma sencilla a los medios de aprendizaje y comunicación, solo el 10.6% de los colombianos con discapacidad usan los medios escritos, como la prensa, libros, revistas y el 2.2% de las personas discapacitadas en Colombia usan el internet [8], siendo este último

medio de comunicación uno de los más importantes en la última década, para el cual se requiere interactuar de forma física con un computador.

1.1 Justificación

En esta investigación se diseñó un sistema capaz de detectar movimientos usando técnicas de visión artificial 3D para la interacción entre el computador y personas con movilidad reducida. La principal función del desarrollo de este sistema es el de incentivar la participación y la inclusión de personas con discapacidad, brindando un mayor acceso a la información y la comunicación e impulsar proyectos que promuevan principios de respeto a la diferencia por medio de estrategias pedagógicas de enseñanza activa en las que el estudiante se sienta parte del conocimiento y pueda interactuar con los diferentes conceptos en las áreas básicas como lo son la comprensión de la materia, el movimiento, la fuerza, el espacio, las estructuras y demás.

1.2 Objetivo General

Diseñar un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción entre el computador y personas con movilidad reducida.

1.3 Objetivos Específicos

- Identificar los requerimientos del sistema tanto de las técnicas de visión artificial 3D a emplear como las necesarias en la interacción de las personas con movilidad reducida.
- Caracterizar el esquema de diseño del dispositivo de visión 3D orientado a personas con movilidad reducida.
- Diseñar una interfaz de usuario que permita la interacción por medio de movimientos a las personas con movilidad reducida usando el esquema de diseñado.

- Integrar el diseño del dispositivo 3D con la interfaz de usuario, orientado a personas con movilidad reducida.

- Validar el funcionamiento del sistema de acuerdo al comportamiento del usuario.

2.Marco Teórico y Antecedentes

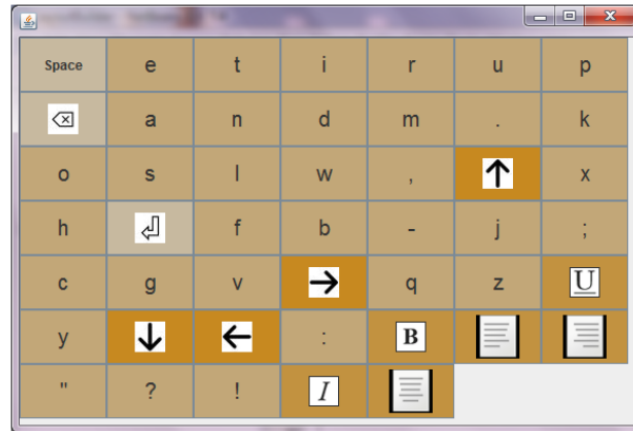
Los dispositivos alternos de entrada pueden ser soluciones de hardware o software que permiten a los usuarios que presentan algún impedimento poder acceder a un ordenador de una manera diferente. Estos dispositivos cuentan con una gran variedad de formas y tamaños con el fin de adaptarse a múltiples limitaciones. Estos dispositivos alternos permiten interactuar con el computador a través del uso de sus pies, la cabeza, los ojos, la boca, o las manos. Algunos de estos dispositivos son activados por el movimiento, mientras que otros pueden ser controlados por medio de los nervios musculares, seguimiento óptico e incluso la actividad cerebral.

En el desarrollo de dispositivos existe una gran cantidad de técnicas empleadas con respecto a la interacción Humano Maquina, los dispositivos más usados cuentan con alguna de las siguientes características:

- Teclados en pantalla: Estos dispositivos se utilizan para controlar teclados en pantalla (Ver Figura 2-1) y barras de herramientas, los cuales emulan el teclado o acciones específicas en el software de destino. La introducción de texto y la edición en este tipo de dispositivos consume gran esfuerzo y es muy lento comparado a los dispositivos tradicionales, en este tipo de tecnología es muy importante optimizar el teclado o la barra de herramientas correspondiente con el fin de maximizar la posible velocidad de mecanografía [9].

10 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador

Figura 2-1: Teclado en pantalla [9].



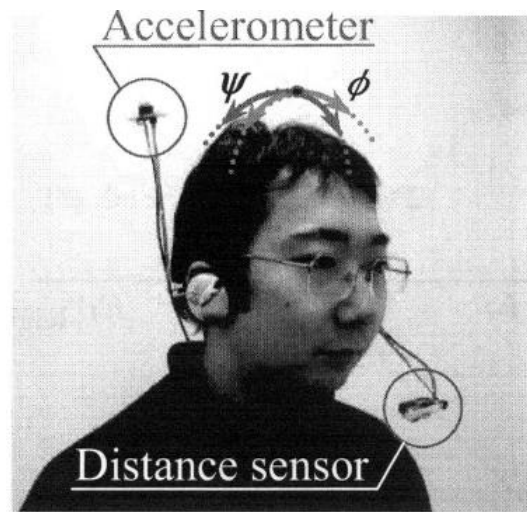
- Dispositivos que usan la voz: El reconocimiento de voz es una tecnología que utiliza un equipo y un servicio deseado el cual puede ser controlado a través de voz, sin el uso de elementos tales como un ratón o un teclado por medio de un micrófono o array de micrófonos (Ver Figura 2-2). Hasta ahora, lo relacionado con las tecnologías de reconocimiento del habla se ha utilizado de forma limitada para propósitos especiales, estas tecnologías han ido evolucionando rápidamente a causa de la proliferación de terminales informáticos portátiles tales como los smartphones. Entre estas tecnologías que usan el reconocimiento de voz están las que proporcionan una forma en la que un individuo ciego o con una discapacidad física puede controlar fácilmente muchas funciones de un ordenador a través del habla [10].

Figura 2-2: Array de micrófonos. CrispMic [10].



- Dispositivos que usan la cabeza y/o la boca: El usuario llevaba un dispositivo de fijación en la cabeza con un sensor de posición en la parte superior y un sensor de distancia delante de la boca (Ver Figura 2-3). El sensor de posición mide los ángulos de balanceo y cabeceo de la cabeza, el sensor de distancia analiza si la boca se abre o no. En las operaciones del ratón, un cursor del ratón en un monitor podría ser trasladado de acuerdo con la dirección y la magnitud del ángulo de la cabeza inclinada, y con el sensor de distancia en la boca se activaría el click del ratón. Esta tecnología es empleada en personas que tienen dificultad en mover los miembros superiores del cuerpo [11].

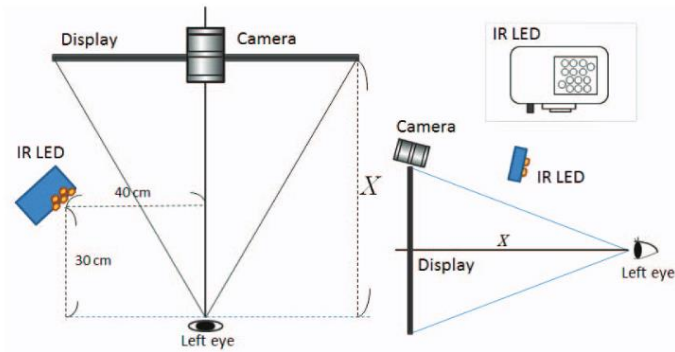
Figura 2-3: Dispositivo de fijación en la cabeza con sensor de posición y distancia [11].



- Dispositivos que usan los ojos: El dispositivo de entrada por el movimiento del ojo es generalmente operado por la dirección de la mirada y en algunos casos por el movimiento de la cabeza. Consta generalmente de una cámara y un led infrarrojo que capta los movimientos de la pupila [12] (Ver Figura 2-4) y convierte estos en un vector de posición con el cual se podría desplazar el puntero del mouse en un computador.

12 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador

Figura 2-4: Configuración del dispositivo de entrada por medio del movimiento de los ojos [12].



- Dispositivos que usan el cuerpo: El seguimiento del cuerpo completo se utiliza comúnmente para interactuar con videojuegos, además, también se puede concentrar en el reconocimiento de gestos y construir una gramática para generar comandos gestuales. Gestos naturales como el escalado, la rotación y traslación de una imagen con dos dedos, los gestos son fundamentales y se han integrado en algunos framework y bibliotecas [13] este tipo de dispositivos funcionan a través de técnicas de visión artificial en la cuales se capta una imagen y se extraen sus características para luego ser procesada. El funcionamiento de estos dispositivos es similar a los usados para la detección de movimientos en los ojos (Ver Figura 2-4) con la diferencia de que requieren 2 cámaras, generalmente una de ellas es infrarroja.

3. Solución Propuesta

El sistema para la detección de movimientos a partir de visión artificial 3D se divide en dos subsistemas principalmente: el de sensado (sensor 3D), el computacional (Interfaz de usuario enfocada a personas con movilidad reducida).

El primero se encargará de recolectar la información del sensor de visión 3D y luego la enviará al computador, en el cual la interfaz de usuario tomará estos datos y aplicará técnicas de visión artificial con el fin de elaborar el modelo del usuario y extraer sus características para luego ser procesadas como ordenes hacia el computador.

3.1 Subsistema de sensado

Este sistema de visión estéreo está conformado por dos cámaras (una cámara a color y otra infrarrojo) y un proyector infrarrojo con las que se realizara el mapeo 3d del ambiente obteniendo los mapas de profundidad (Depth map) [14] y las nubes de puntos necesarias para la reconstrucción del modelo en 3D y al mismo tiempo obtendrá información acerca de la ubicación de la persona dentro del medio, del cual podrá obtener la velocidad a la cual se desplaza y la posición en la que se encuentra en cada instante de tiempo a partir de técnicas de visión artificial.

Para este subsistema se realizaron pruebas de dos tipos:

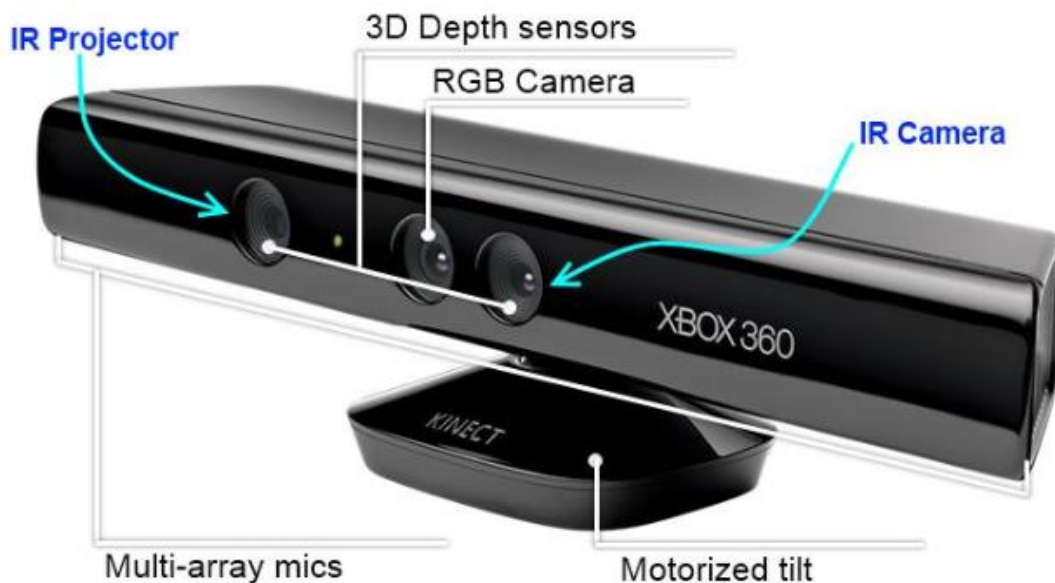
Para objetos fijos, en las cuales se estudiarán los límites de tamaño, resolución y calidad, entre estos objetos físicos se pretende también que logre modelar con buena precisión rostros humanos de diferentes tipos y a diferentes distancias, todo en tiempo real permitiendo visualizar gestos realizados de forma precisa.

Para objetos en movimiento: en estas pruebas el dispositivo debe ser capaz de recrear el modelo 3d del lugar por donde se vaya desplazando el usuario e ir guardando estos datos para su posterior análisis y procesamiento.

- 14 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador
-

Este sistema cuenta también con un sistema de comunicación encargado de transferir los datos por medio de una conexión USB, en la figura 3-1 se puede ver cómo están distribuidos en el sensor estos elementos.

Figura 3-1: sensor Kinect [15]



3.1.1 Cámara RGB.

La cámara RGB, puede trabajar con dos formatos de imagen principalmente: formato RGB y formato YUV. En el formato RGB la cámara es capaz de capturar video a treinta frames por segundo (30fps). Las imágenes obtenidas en el formato YUV solo se pueden obtener en una resolución de 640x480 píxeles y a solo 15fps. [16]

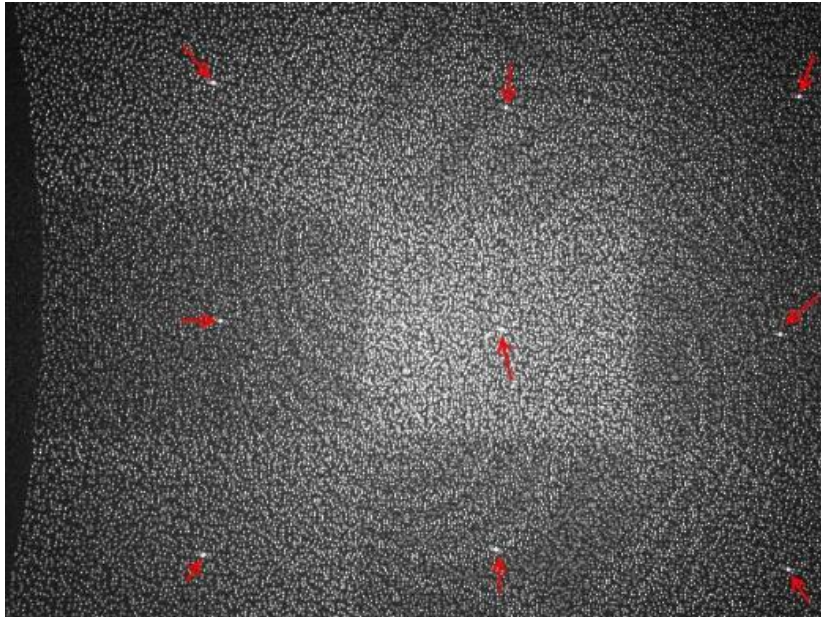
3.1.2 Sensor de profundidad

El sensor de profundidad es un dispositivo que está compuesto por un diodo laser infrarrojo el cual pasa por una rendija de difracción, lo cual es necesario para su correcto funcionamiento.

La fuente de luz infrarroja, proyecta un patrón de puntos sobre el escenario el cual es captado e interpretado por un sensor infrarrojo. El sensor detecta los segmentos de puntos reflejados y estima la profundidad a partir de la intensidad y la distorsión de los mismos.

La figura 3-2 muestra un patrón de puntos claros y oscuros proyectados por el sensor Kinect. Según ROS (Robot Operating System), el algoritmo comienza calculando la profundidad de un plano de referencia a partir de los nueve puntos que aparecen muy marcados y guarda el patrón para ese plano. Posteriormente, la profundidad para cada píxel se calcula eligiendo una ventana de correlación pequeña (9x9 ó 9x7) y se compara el patrón local en ese píxel con el patrón memorizado en ese píxel y los 64 píxeles vecinos en una ventana horizontal [17].

Figura 3-2: Patrón de puntos emitidos por el sensor Kinect [15]



Dadas la profundidad del plano y la disparidad, se estima una profundidad para cada píxel la cual puede ser calculada por triangulación, de acuerdo a la expresión utilizada por un sistema estéreo normal. Por otro lado, la resolución en la dimensión z es de aproximadamente un centímetro, mientras que la resolución espacial (ejes x e y) es del orden de milímetros. La información de profundidad se devuelve en un mapa de píxeles a

16 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador

una frecuencia de 30 frames por segundo. Cada pixel está representado por dos bytes (16 bits), cuyo valor representa teóricamente la distancia del objeto al sensor.

Para un funcionamiento correcto, deben satisfacerse las siguientes condiciones:

- Máxima (mínima) distancia del objeto al sensor: 3.5 (1.2) metros estos valores varían dependiendo del tipo de sensor seleccionado.
- Los objetos deben estar dentro del ángulo de visión: $\pm 43^\circ$ verticalmente y de $\pm 57^\circ$ horizontalmente.
- Características relacionadas con el tipo de objeto (translucido, especular, cóncavo, etc.).
- Características relacionadas con la iluminación (luz solar fuerte).

Si el valor de un píxel del mapa es cero, significa que el sensor no pudo estimar la profundidad para esa región por no cumplirse alguna de estas condiciones [17][8].

3.2 Subsistema computacional

Este subsistema cuenta con el desarrollo de una interfaz de usuario en C++ la cual se encarga de extraer las características de las imágenes tales como: procesadas y tomar las decisiones con respecto a cómo debe comportarse el cursor del monitor.

El proceso completo de visión estéreo está compuesto por 6 partes fundamentales [16]:

1. Adquisición de imágenes
2. Modelado de la cámara (geometría del sistema)
3. Extracción de las características
4. Correspondencia de las imágenes (características)
5. Determinación de la distancia (profundidad)
6. Interpolación

Las dos primeras pertenecen al dispositivo de sensado 3D: la primera adquiere la imagen y la envía al computador y la segunda corresponde a los parámetros intrínsecos de la cámara como distancia focal y modelo de distorsión.

El resto de partes del proceso de la visión estéreo están conformadas por el software el cual realiza las tareas de extracción de características como: bordes, áreas, conectividad, entre otros, luego compara las imágenes de ambas cámaras y calcula las diferencias entre una y otra, luego de tener estos valores y los parámetros de la cámara determina el mapa de profundidad de cada pixel a través de triangulación y por último realiza una interpolación para corregir la imagen si es necesario [16].

Para lograr esta extracción de características en la interfaz de usuario se utilizaron los siguientes componentes y frameworks [18]:

Kinect SDK

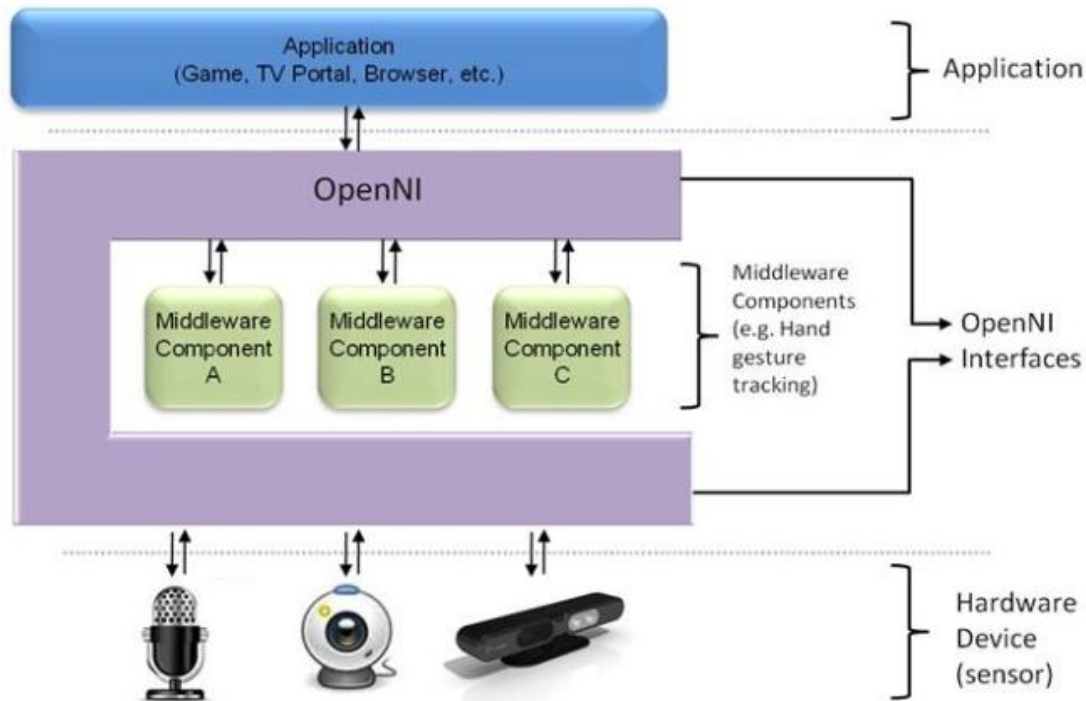
SDK, Framework para sensores o módulos que proveen depth maps, color maps, scene maps, gesture recognition, user pose (skeleton). La figura 3-3 muestra una vista de tres capas del concepto de OpenNI el cual es el mismo implementado en el SDK de Kinect, con cada capa representando un elemento integral:

Arriba: Representa el software que implementa aplicaciones de interacción naturales

Medio: Representa el Framework OpenNI, proporcionando interfaces de comunicación que interactúan tanto con los sensores y los componentes de middleware, que analizan los datos de los sensores.

Abajo: Muestra los dispositivos de hardware que capturan los elementos visuales y de audio de la escena.

Figura 3-3: Capas del OpenNI [18].



La API permite que varios componentes puedan ser registrados en el Framework del OpenNI, estos componentes se conocen como módulos, y se utilizan para producir y procesar los datos de los sensores.

Los módulos que están soportados actualmente son:

Módulos de sensor

- sensor 3D
- cámara RGB
- cámara IR
- dispositivo de audio (micrófono o una matriz de micrófonos)

Componentes de middleware

- análisis corporal completo: un componente de software que procesa los datos sensoriales y genera la información relacionada con el cuerpo (por lo general la estructura de datos que describe las articulaciones, la orientación, el centro de masa, y así sucesivamente)
- análisis del punto de la mano: un componente de software que procesa los datos sensoriales y genera la ubicación de un punto de la mano
- detección de gestos: un componente de software que identifica gestos predefinidos (por ejemplo, una mano moviéndose) y alertas de la aplicación.
- Analizador de escena: un componente de software que analiza la imagen de la escena con el fin de producir información, tales como:
 - La separación entre el primer plano de la escena (es decir, las figuras) y el fondo.
 - Las coordenadas del plano del suelo.
 - Identificación individual de las figuras en la escena.

Estos son los aspectos más relevantes del Framework, para utilizar este framework se requieren los drivers para el Kinect; creado por PrimeSensor driver (open source) se obtiene en: github.com/avin2/SensorKinect.

Para cumplir con estos lineamientos se desarrollan las siguientes actividades:

De forma resumida se definen las características y especificaciones técnicas para el desarrollo del sistema de visión 3D, estableciendo los soportes necesarios sobre el diseño y construcción del sistema, se diseñó una interfaz de usuario que permite el uso a personas con movilidad reducida, luego se realizó una integración del dispositivo de visión 3D con la interfaz de usuario, posteriormente se evaluó y verifico el desempeño del sistema en un ambiente real.

Esta etapa de desarrollo de la investigación se presenta de la siguiente manera:

Primero se definieron las características del sistema tanto de las técnicas de visión artificial 3D a emplear como las necesarias en la interacción de las personas con movilidad reducida, Segundo se elaboró un esquema de diseño del dispositivo de visión 3D, Tercero se realizó el diseño de una interfaz de usuario, la cual permite interactuar a las personas a través del dispositivo portátil desarrollado con el computador por medio de movimientos, Cuarto se integró el dispositivo 3D con la interfaz de usuario, permitiendo el uso de este a personas con movilidad reducida, Quinto se validó el funcionamiento del sistema de acuerdo al comportamiento del usuario.

3.3 Características y Especificaciones Técnicas Para la Elección del Sensor

3.3.1 Características del Kinect, Fabricante Microsoft y PrimeSense.

El sensor Kinect está desarrollado por PrimeSense en colaboración con Microsoft para la consola Xbox, este dispositivo ofrece la capacidad de obtener imágenes de profundidad gracias a un sensor infrarrojo. Este dispositivo también incluye una cámara RGB que ofrece imágenes a color además de una serie de micrófonos colocados de tal manera que son capaces de saber de qué ángulo proviene el sonido. En un principio Kinect se creó para el ocio doméstico, pero su bajo precio (en comparación con otras cámaras de iguales características) y la aceptación por parte de desarrolladores han explotado sus posibilidades [15].

Especificaciones técnicas [19]:

Sensores

- Lentes de color y sensación de profundidad
- Microfono multi-arreglo
- Ajuste de sensor con su motor de inclinación
- Totalmente compatible con las consolas existentes de Xbox 360

Campo de visión

- Campo de visión horizontal: 57 grados
- Campo de visión vertical: 43 grados
- Rango de inclinación física: ± 27 grados
- Rango de profundidad del sensor: 1,2 - 3,5 metros

Data Streams (Flujo de datos)

- 320 x 240 a 16 bits de profundidad @ 30fps
- 640 x 480 32-bit de color @30fps
- Audio de 16-bit @ 16 kHz

Sistema de Seguimiento

- Rastrea hasta 6 personas, incluyendo 2 jugadores activos
- Rastrea 20 articulaciones por jugador activo

Sistema de audio

- Sistema de cancelación de eco que aumenta la entrada de voz
- Reconocimiento de voz múltiple

OS Support

- Win 32/64: XP, Vista, 7, 8
- Linux Ubuntu 10.10: X86,32/64 bit

Software

- Software development kits (OpenNI SDK bundled)

Programming Language

- C++/C# (Windows)
- C++(Linux)
- JAVA

- 22 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador
-

3.3.2 Características del ASUS Xtion PRO LIVE, Fabricante Asus y PrimeSense.

El ASUS Xtion PRO LIVE está basado en la tecnología de proyección estéreo desarrollada por PrimeSense. Este provee imágenes en RGB e imágenes de profundidad con resoluciones VGA (640x480 pixeles) a una tasa de 30 frames por segundo, una alta tasa de frame de 60 fps puede ser obtenida utilizando una resolución QVGA (320x240 pixeles). Los valores de profundidad se codifican como un valor de 16bit sin signo los cuales representan la distancia en milímetros. Acorde a las especificaciones técnicas el rango de funcionamiento esta entre los 0.8m y los 3.5m [20].

Especificaciones técnicas [21]:

Sensores

- RGB
- Depth
- Microphone*2

Data Streams (Flujo de datos)

- VGA (640x480): 30 fps
- QVGA (320x240): 60 fps

Campo de visión

- 58° H, 45° V, 70° D (Horizontal, Vertical, Diagonal)
- Campo de visión horizontal: 58 grados
- Campo de visión vertical: 45 grados
- Campo de visión Diagonal: 70 grados
- Rango de profundidad del sensor: 0,8 - 3,5 metros

Interfaz

- USB2.0/ 3.0

OS Support

- Win 32/64: XP, Vista, 7, 8
- Linux Ubuntu 10.10: X86,32/64 bit
- Android

Software

- Kit de desarrollo de software (OpenNI SDK bundled)

Lenguaje de programación

- C++/C# (Windows)
- C++(Linux)
- JAVA.

3.3.3 Elección del sensor

Debido a que ambos sensores tanto el Kinect como el Xtion PRO LIVE, cuentan con características similares se eligió el Kinect debido a que es más común encontrarlo en el mercado ya que este dispositivo es usado en las consolas de videojuego Xbox, por lo que es muy fácil encontrarlo en gran cantidad de hogares, por lo cual sería un costo que no se tendría que asumir, aumentando la facilidad de adquisición e implementación del sistema a gran cantidad de usuarios.

3.3.4 Requerimientos de hardware y software:

El sistema para la detección de movimientos a partir de visión artificial 3D se divide en 2 subsistemas: el de sensado (sensor 3D) y el computacional (Interfaz de usuario).

El primero se encargará de recolectar la información de las 2 cámaras y luego las enviará al computador, en el cual la interfaz de usuario tomará estos datos y aplicará técnicas de visión artificial con el fin de elaborar el modelo del usuario y extraer sus características para luego ser procesadas como ordenes hacia el computador.

3.4 Esquema De Diseño

Para el esquema de diseño se tuvo en cuenta tres partes fundamentales, la primera es el posicionamiento del cursor la cual está encargada de obtener la información de las coordenadas dadas por el SDK y pasarlas al sistema de coordenadas de C++, la segunda parte es la encargada de reconocer los gestos de los brazos del usuario con el cual se ejecutaran las acciones asociadas a cada gesto y por último se encuentra el diseño de la interfaz de usuario la cual cumple ciertas características, que permiten el correcto funcionamiento y orienten al usuario sobre que acción están realizando en cada momento.

3.4.1 Posicionamiento del cursor

Con el fin de interactuar con un computador por medio de detección de movimientos la idea es usar el dispositivo de entrada más común como lo es un ratón, donde el movimiento de la mano del usuario se asigna directamente a la posición del cursor. Si el usuario mueve su mano a la izquierda, el cursor debe moverse hacia el lado izquierdo de la pantalla y así respectivamente para el resto de movimientos.

Al detectar el movimiento de la mano utilizando el Kinect el OpenNI SDK presenta los datos de la mano como un punto en un sistema de coordenadas con el tamaño de 640 x 480. El sistema de coordenadas tiene un eje X desde -320 a 320 píxeles y un eje Y de -240 a 240 píxeles como se muestra en la Figura 3-4.

Figura 3-4: Coordenadas del sistema usadas por OpenNI SDK.

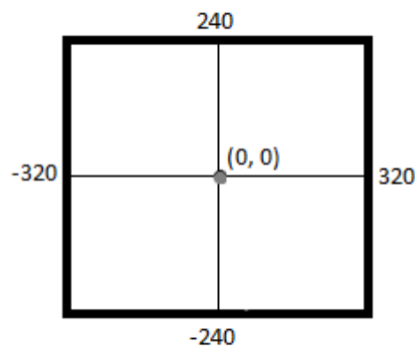


Figura 3-5: Coordenadas del sistema usadas por C++.



Para utilizar los datos del SDK para mover el cursor en la pantalla, los datos deben convertirse ya que la interfaz diseñada en C++ sólo es capaz de trabajar con un sistema de coordenadas, donde el centro del sistema de coordenadas está situado en la parte superior izquierda de la pantalla. Una ilustración del sistema de coordenadas utilizado por C++ para un 1920 x 1080 Se muestra en la Figura 3-5.

Cuando se utiliza una pantalla de 1920 x 1080, el sistema de coordenadas utilizado en C++ tiene un eje x que va de 0 a 1920 y un eje y de 0 a 1080. Los datos deben ser convertidos desde un rectángulo con medidas de 640 x 480 píxeles con centro en el medio a un rectángulo de 1920 x 1080 que tiene centro en la parte superior izquierda.

3.4.2 Reconocimiento de gestos

Con el fin de interactuar con un computador por medio de detección de movimientos la idea es usar el dispositivo de entrada más común como lo es un ratón, donde el movimiento de la mano del usuario se asigna directamente a la posición del cursor. Si el usuario mueve su mano a la izquierda, el cursor debe moverse hacia el lado izquierdo de la pantalla y así respectivamente para el resto de movimientos. Para hacer clic solo basta con acercarla la mano a la pantalla y alejarla, el clic izquierdo es acercando la mano y devolviéndolo al lugar central, y el clic derecho alejando el brazo de la pantalla.

3.4.3 Interfaz de Usuario

Se requiere una aplicación que funcione como una herramienta de accesibilidad similares a las de Windows como lo son la lupa y el teclado sobre la pantalla, esta aplicación podrá ejecutarse en segundo plano he integrarse y funcionar en conjunto con otras aplicaciones que se estén usando en el sistema.

El usuario también debe ser capaz de ver, cuando el sistema está reconociendo un gesto. Si no, hay indicación de si el sistema ha reconocido un gesto, esto puede aumentar la dificultad de usar el sistema. Si, por ejemplo, el usuario intenta hacer clic en un enlace y ejecuta el gesto correcto, pero falla, el usuario no puede ver si perdió el enlace o hizo un gesto incorrecto.

Por lo tanto, para informar al usuario de que ha realizado correctamente el gesto, el sistema escribe el gesto reconocido, por ejemplo: "Clic" en un cuadro de texto.

Para facilitar el aprendizaje de los gestos se muestra la imagen RGB del Kinect en la interfaz como método de ayuda al usuario. Al tener la imagen RGB a su disposición los usuarios pueden ver los límites de la Kinect y es capaz de verse ellos mismos realizando los gestos.

4. Resultados

Para la obtención de resultados de la aplicación se tuvieron en cuenta las partes principales que la componen, las cuales están dadas por la arquitectura del SDK que se muestra en la Figura 3. Para este caso la capa de aplicación se ha dividido en varias partes. Las partes en las que se dividió se describen a continuación.

- Seguimiento de usuarios

Seguimiento de un usuario para poder recuperar datos del usuario ubicado en el Vista del Kinect. Esto se hace a través de un componente en la capa de OpenNI, a este objeto lo llamaremos UserGenerator.

- Seguimiento de la mano

Seguimiento de la mano derecha del usuario para recuperar coordenadas de la posición de las manos. Esto es necesario para poder mover el cursor en la pantalla de acuerdo con la posición actual de la mano derecha. Esto se hace usando componente en la capa de OpenNI, a este objeto lo llamaremos HandGenerator.

- Filtrado de los datos de la mano derecha

El filtrado se realiza para eliminar la fluctuación de fase en los datos utilizados para controlar el cursor. Esto se hace usando el elemento Filter, el cual pertenecerá a la capa de aplicación.

4.1 Resultados De Los Test De Funcionamiento De La Integración Del Hardware Y Software

Luego de realizar la integración del sistema, se continua con la evaluación y verificación de del funcionamiento del sistema ante comportamientos reales del ambiente como objetos en el campo de visión y cambio de posición del usuario.

En estas pruebas se obtuvieron varios tipos de resultados:

En una se realizó un programa en la cual se pudieran obtener las imágenes de profundidad con las cuales se podrá hacer la identificación de los movimientos y gestos que realice el usuario, y en la otra se obtuvo la imagen de profundidad y de color a las cuales se les añadió un seguimiento principalmente de las manos por el método SkeletonTracking del SDK, con el fin de identificar la posición en la que cada extremidad del cuerpo se encontraba en cada momento.

En la realización de este programa se utilizaron las funciones y métodos que están incluidas en el SDK para el Kinect. Las funciones principales que se utilizaron son las siguientes:

[NuiCreateSensorByIndex](#): Crea una instancia del sensor Kinect con un especifico index para que una aplicación pueda abrir y usar

```
HRESULT NuiCreateSensorByIndex(  
    int index,  
    INuiSensor **ppNuiSensor  
)
```

- **Parámetros**

Index Tipo: int [in]

ppNuiSensor Tipo: INuiSensor [out] un pointer que recibe una referencia para crear un

[INuiSensor](#): interfaz.

- **Valores que regresa**

Tipo: HRESULT

Retorna S_OK si fue exitoso; en otro caso retorna un código de error

[NuImageResolutionToSize](#): con esta función se define la resolución de la imagen que se va a obtener

```
void NuImageResolutionToSize(  
    NUI_IMAGE_RESOLUTION res,  
    DWORD &refWidth,  
    DWORD &refHeight  
)
```

Parametros

res Tipo: NUI_IMAGE_RESOLUTION [in] La resolución de la imagen.

refWidth Tipo: DWORD [out] Ancho de la imagen.

refHeight Tipo: DWORD [out] Alto de la imagen.

[INuiSensor::NuImageStreamOpen Method](#): crea un stream que recibe la imagen de profundidad del sensor.

public:

```
HRESULT NuImageStreamOpen(  
    NUI_IMAGE_TYPE eImageType,  
    NUI_IMAGE_RESOLUTION eResolution,  
    DWORD dwImageFrameFlags,  
    DWORD dwFrameLimit,  
)
```

▪ Parámetros

[eImageType](#) Tipo: NUI_IMAGE_TYPE [in] un [NUI_IMAGE_TYPE](#) valor que especifica el tipo de imagen que va a recibir los valores dependen del *dwFlags* parámetro del [INuiSensor::NuInitialize](#) método.

[eResolution](#) Tipo: NUI_IMAGE_RESOLUTION [in] un [NUI_IMAGE_RESOLUTION](#) es un valor que especifica cual es la resolución de la imagen los valores dependen del *dwFlags* parámetro del [INuiSensor::NuiInitialize](#) metodo.

[dwImageFrameFlags](#) Tipo: DWORD [in] Especifica las opciones del frame

dwFrameLimit Tipo: DWORD [in] El numero de frame que el Kinect mantendrá en el buffer

- **Valores que regresa**

Tipo: HRESULT

Retorna S_OK si fue exitoso; en otro caso retorna un código de error

INuiSensor::NuiSkeletonGetNextFrame Method

public:

```
HRESULT NuiSkeletonGetNextFrame(  
    DWORD dwMillisecondsToWait,  
    NUI_SKELETON_FRAME *pSkeletonFrame  
)
```

- **Parameters**

[dwMillisecondsToWait](#) Tipo: DWORD [in] El tiempo en milisegundos que espera antes de regresar sin un frame.

[pSkeletonFrame](#) Tipo: NUI_SKELETON_FRAME un puntero a [NUI_SKELETON_FRAME](#) estructura que contiene los siguientes frame en el skeleton stream.

- **Valores que regresa**

Tipo: HRESULT

Retorna S_OK si fue exitoso; en otro caso retorna un código de error

Los resultados de las pruebas se pueden observar en las figuras 4-1, 4-2 y 4-3.

Figura 4-1: Imagen de profundidad dada por el sensor Kinect.

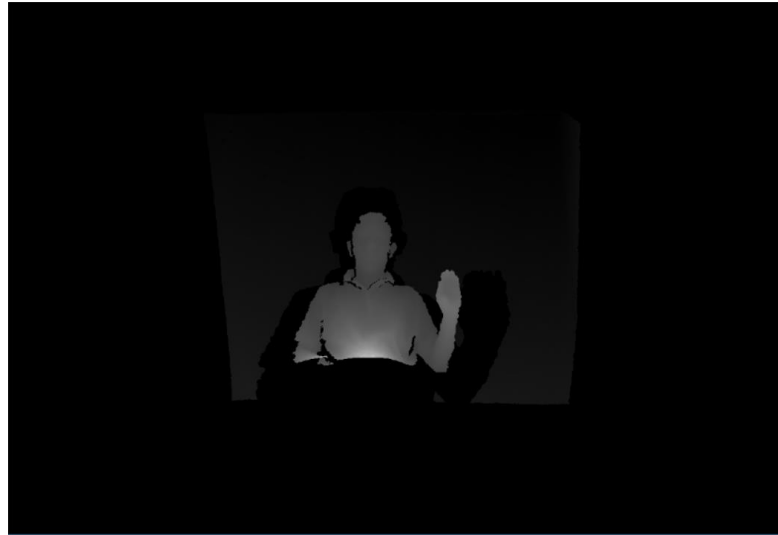


Figura 4-2: Imagen de profundidad integrada con el SkeletonTraking dada por el sensor Kinect.

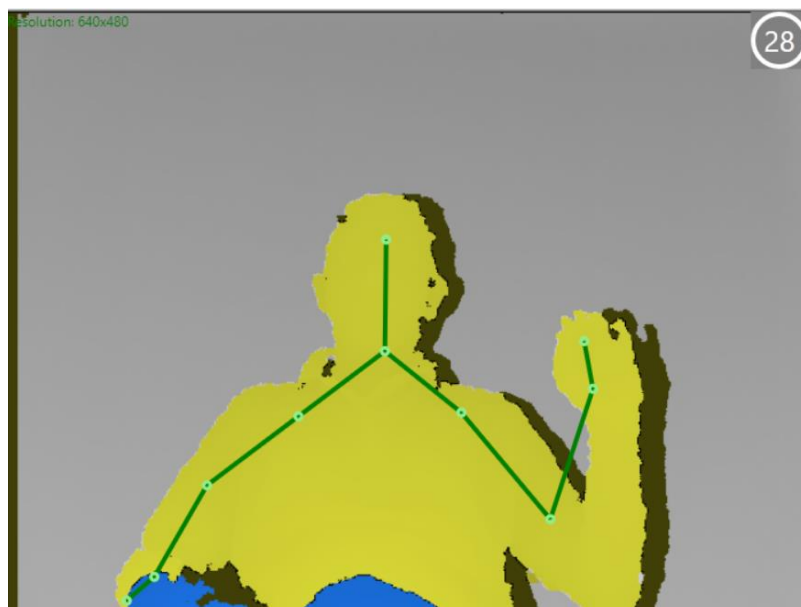
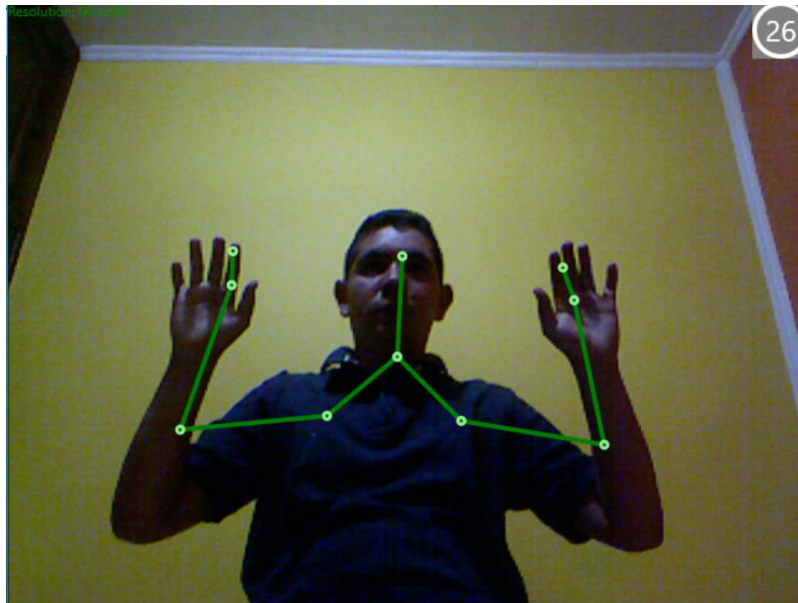


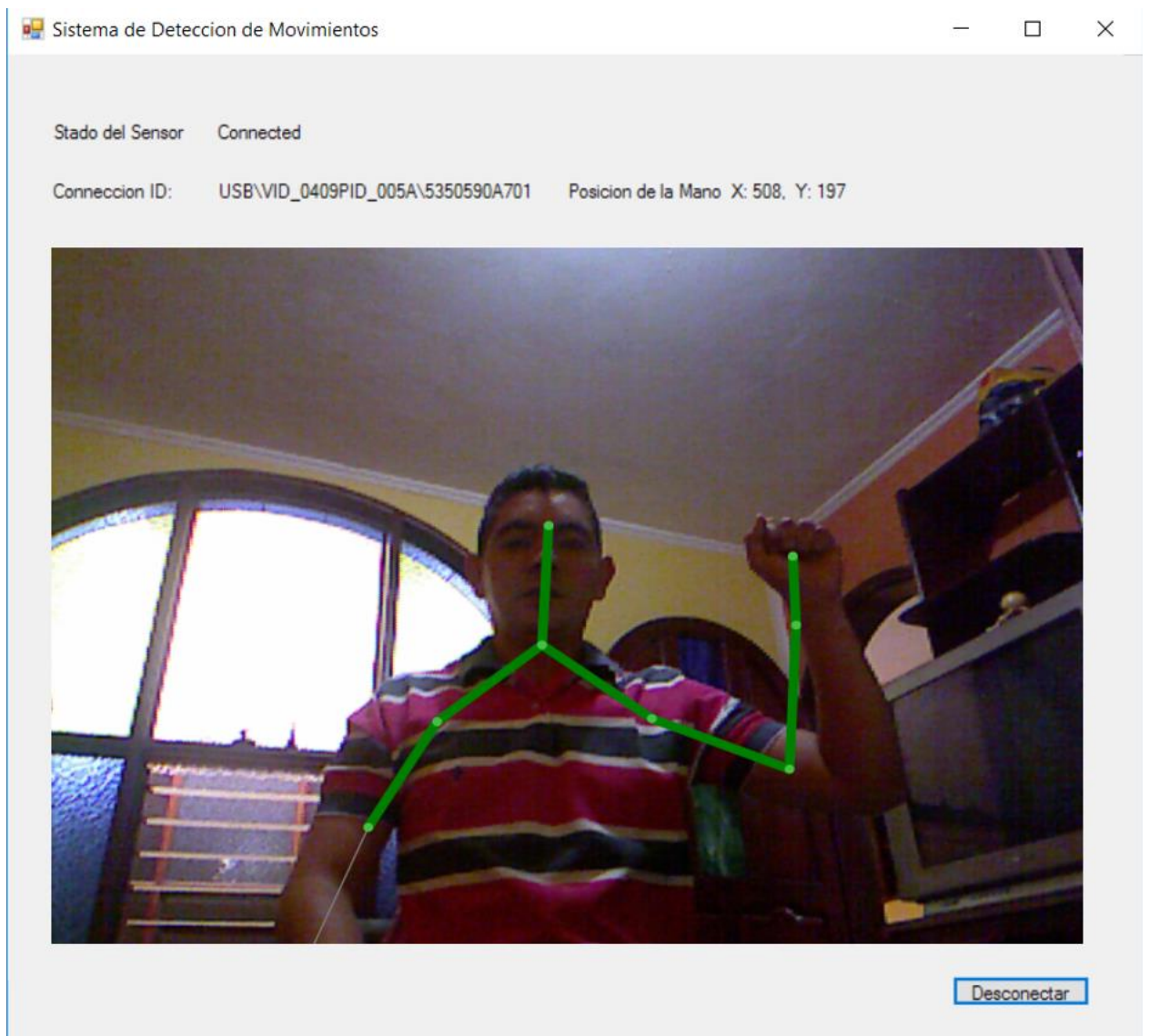
Figura 4-3: Imagen a color integrada con el SkeletonTraking dada por el sensor Kinect.



Ya teniendo las coordenadas de los puntos de las manos solo se necesita hacer el escalamiento de las coordenadas del Kinect a las coordenadas de la interfaz que controlara el mouse en C++, como se observa en las figuras 3-4 y 3-5.

4.2 Interfaz final

En la interfaz (Ver Figura 4-4) aparecen 3 indicadores, y un botón, los indicadores son: el label de estado del sensor, el cual Indica si el sensor está conectado, desconectado o iniciando. El label de Coneccion ID que indica el ID del sensor conectado, y el label que indica la posición X y Y de la mano en las coordenadas del sensor.

Figura 4-4: Interfaz de usuario desarrollada en C++.

4.3 Validación de resultados

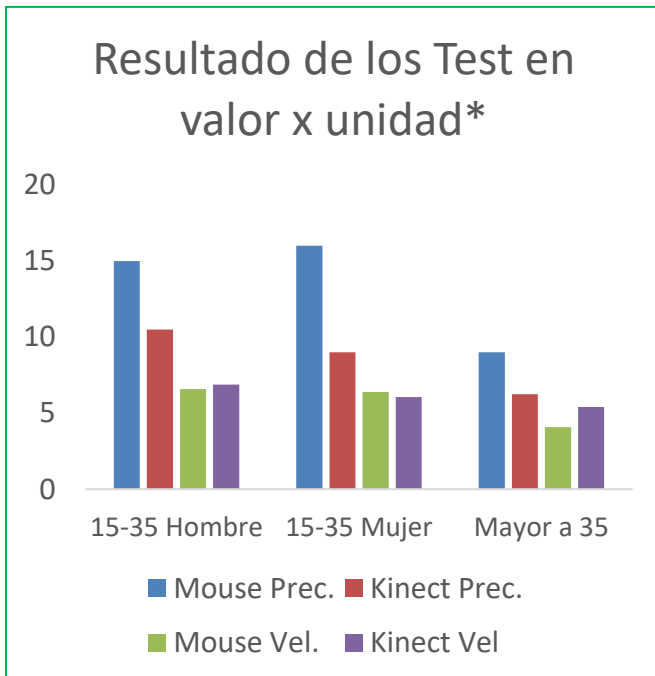
La validación de resultados se realizó por medio de dos test muy sencillos:

En el primer test se evaluó la velocidad a la que una persona es capaz de presionar el botón izquierdo del mouse vs la velocidad a la que realiza el evento de clic con el sensor Kinect, para medir esta velocidad se utilizó un juego disponible en línea llamado Super click tester [22].

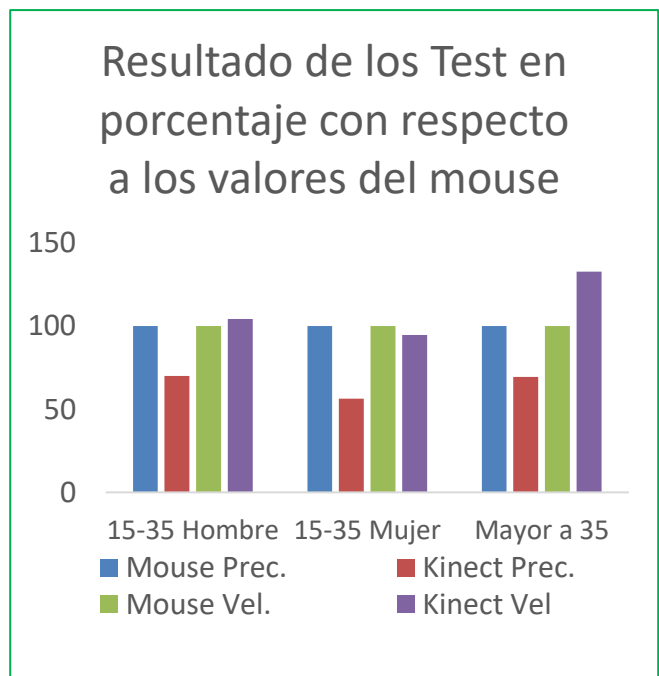
En el segundo test se evaluó la velocidad y precisión de una persona a lo hora de seleccionar un objeto con el mouse vs la velocidad a la que realiza el movimiento y el evento de clic con el sensor Kinect, para medir este resultado se utilizó un juego disponible en línea llamado Guerra de globos de agua [23].

Los Resultados se pueden apreciar en las Gráficas 1 y 2 siguientes.

Grafica 1. Resultados Test. Fuente: Elaboración propia



Grafica 2. Resultados Test en %. Fuente: Elaboración propia



* En el Test de precisión la unidad es # de Aciertos
* En el Test de velocidad la unidad es # de clic x seg.

5. Conclusiones y recomendaciones

5.1 Conclusiones

Como resultado de todo el proceso de aprendizaje que genera un proyecto de investigación, queda la sensibilidad de identificar una necesidad en un grupo poblacional con discapacidad en miembros superiores, la Adquisición de conocimientos durante la consulta de Bibliografía y la vigilancia tecnológica, la satisfacción de aplicar los conocimientos adquiridos durante el desarrollo del proyecto con el fin de satisfacer las actividades y objetivos planteados

En cuanto a los objetivos y actividades planteadas inicialmente, podemos concluir que: Se logró obtener un sistema Para La Detección De Movimientos A Partir De Visión Artificial 3D, Como Herramienta De Inclusión Y Aprendizaje, que permite controlar el cursor del computador mediante movimientos de la mano utilizando las funciones del SDK del Kinect de Microsoft.

Se seleccionó una plataforma óptima de desarrollo, compuesta por un sensor Kinect de Microsoft y un computador el cual realiza la comunicación con la interfaz de usuario por medio de una conexión USB.

Se validó el funcionamiento del sistema con diferentes usuarios dando como resultados valores de desempeño aceptables

El lenguaje de programación utilizado para el desarrollo de la interfaz fue C++, utilizando el IDE de Microsoft Visual Studio para su implementación y desarrollo, también se realizó la integración del hardware y el software emulando un dispositivo como un ratón con interfaz USB.

5.2 Recomendaciones

Como trabajo futuro a todo aquel que este interesado en este tema, se recomienda primero: hacer una revisión de la tecnología existente en el momento, ya que día a día se presentan nuevos desarrollos en este tipo de tecnologías debido al gran avance tecnológico que se presenta en esta era.

Para todo aquel que desee trabajar con el Kinect se recomienda utilizar la versión de Kinect optimizada para Windows, aunque es un poco mas costoso brinda una gran cantidad de herramientas que podrían facilitar el desarrollo de trabajos futuros o mejoras del mismo propuesto en esta investigación.

A. Anexo: Fragmentos de código utilizados en el desarrollo de la interfaz

Parte principal del código de la interfaz:

```
namespace kinectv2
{
    public partial class frmKinect : Form
    {
        private KinectSensor kSensor;

        public frmKinect()
        {
            InitializeComponent();
        }

        //Metodo que se ejecuta cuando dan click al boton conectar

        private void btnStream_Click(object sender, EventArgs e)
        {

            if (btnStream.Text == "Conectar")
            {
                this.btnStream.Text = "Desconectar";
            }
        }
    }
}
```

38 Diseño de un sistema detector de movimientos usando técnicas de visión artificial 3D para la interacción de personas con movilidad reducida con el computador

```
        if (KinectSensor.KinectSensors.Count > 0)
        {
            kSensor = KinectSensor.KinectSensors[0];

            KinectSensor.KinectSensors.StatusChanged +=
KinectSensors_StatusChanged;

// Llama a un metodo del Kinect encargado de decir el estado del sensor

        }

        kSensor.Start(); // inicia el kinect

        this.lblConnectionID.Text = kSensor.DeviceConnectionId;

kSensor.ColorStream.Enable(ColorImageFormat.RgbResolution640x480Fps30);

//Se habilita el Stream de la camara rgb a 30 fps

        var parameter = new TransformSmoothParameters
        {

//Con estos valores se calibra la velocidad de detección y predicción del kinect

            Smoothing = 0.75f,

            Correction = 0.0f,

            JitterRadius = 0.05f,

            MaxDeviationRadius = 0.04f,

            Prediction = 0.0f

        };

        kSensor.DepthStream.Enable();

//Se habilita el Stream de la camara infraroja
```



```
kSensor.DepthStream.Range = DepthRange.Default;
kSensor.AllFramesReady += KSensor_AllFramesReady;
//Se llama al método que captura todos los frames del kinect

kSensor.SkeletonStream.Enable(parameter);
kSensor.SkeletonStream.TrackingMode = SkeletonTrackingMode.Seated;
//Se habilita el Stream de los nodos del cuerpo en la parte superior
}
else
{
    if(kSensor != null && kSensor.IsRunning)
    {
        kSensor.Stop(); //Detiene el kinect
        this.btnStream.Text = "Conectar";
        this.pbStream.Image = null;
    }
}

}

//método que captura todos los frames del kinect

private void KSensor_AllFramesReady(object sender, AllFramesReadyEventArgs
e)
{
    using (var frame = e.OpenColorImageFrame())
```

```
{
    if (frame != null)

        pbStream.Image = CreateBitmapFromSensor(frame);
//Llama a la función CreateBitmapFromSensor()

}

using (var frame = e.OpenSkeletonFrame())
{
    if (frame != null)
    {
        var skeletons = new Skeleton[frame.SkeletonArrayLength];
        frame.CopySkeletonDataTo(skeletons);

        var TrackedSkeleton = skeletons.FirstOrDefault(s =>
s.TrackingState == SkeletonTrackingState.Tracked);
        if (TrackedSkeleton != null)
        {
            var position =
TrackedSkeleton.Joints[JointType.HandRight].Position;

            var coordinateMapper = new CoordinateMapper(kSensor);

            var colorPoint =
coordinateMapper.MapSkeletonPointToColorPoint(position,
ColorImageFormat.InfraredResolution640x480Fps30);

//Se obtienen las coordenadas de la mano derecha del cuerpo detectado
```

```
        this.lblPosition.Text = string.Format("Posicion de la Mano  
X: {0}, Y: {1}", colorPoint.X, colorPoint.Y);
```

```
        this.Cursor = new Cursor(Cursor.Current.Handle);
```

```
        Cursor.Position = new Point(colorPoint.X, colorPoint.Y);
```

```
//Mueve el cursor a las posiciones X y Y.
```

```
    }
```

```
  }
```

```
}
```

```
}
```

```
//Metodo que monitorea el estado del sensor
```

```
private void KinectSensors_StatusChanged(object sender, StatusChangedEventArgs e)
```

```
{
```

```
    this.lblStatus.Text = kSensor.Status.ToString();
```

```
}
```

```
//Funcion que crea un mapa de bits con los frame del sensor y los pone en la imagen  
del formulario
```

```
private Bitmap CreateBitmapFromSensor(ColorImageFrame frame)
```

```
{
```

```
    var pixelData = new byte[frame.PixelDataLength];
```

```
    frame.CopyPixelDataTo(pixelData);
```

```
        return pixelData.ToBitmap(frame.Width, frame.Height);  
  
    }  
  
}  
  
}
```

Bibliografía

- [1] Calhoun G. L.; G. R. McMillan. "Hands-free input devices for wearable computers", Human Interaction with Complex Systems, 1998. Proceedings, Fourth Annual Symposium on, page(s): 118 – 123.
- [2] Reney, D., Tripathi, N. "An Efficient Method to Face and Emotion Detection", Communication Systems and Network Technologies (CSNT), 2015 Fifth International Conference on, page(s): 493 – 497.
- [3] Rajesh P. N. Rao; Reinhold Scherer. "Brain-Computer Interfacing", IEEE Signal Processing Magazine, Julio 2010, page(s): 152-150, Vol 27.
- [4] Villaverde de la Nava I., "On Computational Intelligence Tools for Vision Based Navigation of Mobile Robots", University of the Basque Country, Department of Computer Science and Artificial Intelligence, 2009.
- [5] Bautista L., Archila J., "Visión Artificial aplicada en sistemas de Realidad Aumentada", 3er Congreso Internacional de Ingeniería Mecatrónica – UNAB, 2011, Vol 2, No 1.
- [6] Mauri C., "Interacción Persona-Ordenador Mediante Cámaras Webcam". Interacción, Lleida, 2004.
- [7] Song Y., Demirdjian D., "Continuous Body and Hand Gesture Recognition for Natural Human-Computer Interaction", Massachusetts Institute of Technology, ACM

Transactions on Interactive Intelligent Systems, Diciembre 2010, Pages 111-148, Vol. 1, No. 1.

[8] Gómez J., "Discapacidad en Colombia: Reto para la Inclusión en Capital Humano", Colombia Líder, Fundación Saldarriaga Concha, Bogotá, Abril 2010.

[9] Šimon Rudolf; Adam J. Sporka; Ondřej Poláček. "Constructing on-screen keyboards in word processing environment for people with motor difficulties", Cognitive Infocommunications (CogInfoCom), 2014 5th IEEE Conference on, page(s): 85-89.

[10] Hae-Duck J. Jeong; Sang-Kug Ye; Jiyoung Lim; Ilsun You; Wooseok Hyun; Hee-Kyoung Song. "A Remote Computer Control System Using Speech Recognition Technologies of Mobile Devices", Innovative Mobile and Internet Services in Ubiquitous Computing (IMIS), 2013 Seventh International Conference on, page(s): 595-600.

[11] Nakazawa N.; Matsui T.; Yamada K. "Welfare Support-equipment for Operations of Personal Computer with Head Tilting and Mouth Open-Close Motions", Systems, Man and Cybernetics, 2006. SMC '06. IEEE International Conference on, page(s): 4046 - 4051, Vol 5.

[12] Oyabu Y; Takano H; Nakamura K. " Development of the eye input device using eye movement obtained by measuring the center position of the pupil", Systems, Man, and Cybernetics (SMC), 2012 IEEE International Conference on, page(s): 2948-2952.

[13] Hongyong T; Youling Y. "Finger Tracking and Gesture Recognition with Kinect", Computer and Information Technology (CIT), 2012 IEEE 12th International Conference on, page(s): 214-218.

[14] Zhengyou Z., "Microsoft Kinect Sensor and Its Effect", Multimedia at Work, IEEE Computer Society, pag 4-10 2012.

- [15] Daniel Ramos Gutiérrez, Estudio Cinemático Del Cuerpo Humano Mediante Kinect, Trabajo De Grado en Telecomunicaciones, Universidad Politécnica De Madrid, Escuela Técnica De Telecomunicaciones, 2013.
- [16] Montalvo M., “Técnicas de visión estereoscópica para determinar la estructura tridimensional de la escena”, Universidad Complutense de Madrid, 2010.
- [17] Martin Morales, Jorge Osio, Juan Eduardo Salvatore, Detección de objetos utilizando el sensor Kinect, Twelfth LACCEI Latin American and Caribbean Conference for Engineering and Technology (LACCEI'2014). “Excellence in Engineering To Enhance a Country’s Productivity”, Guayaquil, Ecuador, July 22 - 24, 2014.
- [18] OpenNI. (18/10/2016). OpenNI User Guide. [Online]. Disponible en: https://github.com/OpenNI/OpenNI/blob/master/Documentation/OpenNI_UserGuide.pdf
- [19] Microsoft. (18/08/2016). Kinect for Windows SDK. [Online]. Disponible en: <http://msdn.microsoft.com/en-us/library/hh855355.aspx>
- [20] Christian Kerl, Odometry from RGB-D Cameras for Autonomous Quadrocopters, Tesis de Maestria en Robotica, Der Technischen Universitat Munchen, Fakultat Fur Informatik, 2012.
- [21] Asus. (18/08/2016). Asus Xtion Pro Live. [Online]. Disponible en: http://www.asus.com/Multimedia/Motion_Sensor/Xtion_PRO_LIVE/#specifications
- [22] Minijuegos. (11/01/2018). Minijuegos en linea. [Online]. Disponible en: <http://www.minijuegos.com/juego/super-click-tester>
- [23] Juegosinfantiles.com. (11/01/2018). Juegos infantiles en linea. [Online]. Disponible en: <http://www.juegosinfantiles.com/locos/guerradeglobosdeagua.html>