



Desarrollo de un prototipo de un API para la interacción de un usuario con aplicaciones con contenido 3D utilizando Kinect

Andrés Prieto López ⁽¹⁾, Sixto García Aguilar ⁽²⁾
Facultad de Ingeniería en Electricidad y Computación
Escuela Superior Politécnica del litoral (ESPOL)
Campus Gustavo Galindo, Km. 30.5 vía Perimetral
Apartado 09-01-5863. Guayaquil, Ecuador
aprieto@espol.edu.ec ⁽¹⁾, sgarcia@cti.espol.edu.ec ⁽²⁾

Resumen

Con el avance de la tecnología y el desarrollo de nuevas aplicaciones con interfaces en 3 dimensiones, el uso del teclado o del ratón es poco práctico. Por esta razón en el presente artículo se presenta el desarrollo de un prototipo de un conjunto de librerías (API), que permite la interacción con el computador, mediante el uso de las manos y gestos como forma de entradas para generar comandos.

Para desarrollar este proyecto se utilizó la tecnología del Kinect de Microsoft. Este sensor de profundidad permite determinar la distancia de los objetos, en particular la posición de la persona y la de sus manos.

Se analizaron diferentes tipos de interacción, para la utilización de los gestos más comunes y más utilizados, que ya han sido adoptados por las personas en las interfaces táctiles. El API de este proyecto se puede utilizar para manipular objetos tridimensionales como modelos arquitectónicos de edificio, autos, muebles, cuerpo humano, órganos, etc.

Al final de la implementación del proyecto se procedió a realizar las pruebas de usabilidad. Para realizar las pruebas del API, se desarrolló una aplicación que utiliza el API para mover un objeto 3D y se realizó un cuestionario, en donde se pedía a sus usuarios calificar su experiencia, en una escala del 1 al 5 de acuerdo a factores como la comodidad, la facilidad de uso y la exactitud.

El análisis de los resultados demostró que la metodología utilizada fue la correcta, haciendo que la interacción sea natural con este tipo de objetos en 3D.

Palabras claves: sensor Kinect, aplicaciones con contenido 3D, Kinect API, modelo 3D

Abstract

The advancement of technology and the development of new applications in 3D interfaces, using the keyboard or mouse is impractical. For that reason this paper describes the development of a prototype of a set of libraries (API) that allows interaction with the computer, using hands and gestures as a way to generate command inputs is presented.

The Microsoft Kinect technology is used to develop this project. This depth sensor determines the distance of objects, particularly the position of the person and his hands.

Different types of interaction, used by the most common gestures that have been adopted by people in touch interfaces were analyzed. This project's API can be used to manipulate three-dimensional objects as building architectural models, cars, furniture, human body, organs, etc.

At the end of project implementation proceeded to conduct usability testing. For testing API, an application that uses the API to move a 3D object and a questionnaire, where users were asked to rate their experience on a scale from 1 to 5 according to factors such as comfort, was performed developed the ease of use and accuracy.

The analysis of the results showed that the methodology used was correct, making it natural interaction with these objects in 3D.

Key words: Kinect sensor, 3D content applications, Kinect API, 3D model

1. Introducción

Desde la salida al mercado de los primeros computadores personales en la década de los 80's, el teclado y el ratón ha sido el principal medio de comunicación entre el ser humano y la máquina, permitiendo el desplazamiento del cursor en dos dimensiones.

Actualmente la tecnología ha avanzado enormemente y se han desarrollado nuevas aplicaciones con interfaces o entornos en 3 dimensiones (3D), como por ejemplo en realidad virtual, realidad aumentada, navegación en modelos arquitectónicos, etc. donde la utilización del teclado y del ratón tiene sus limitaciones.

Para mejorar la interacción en un entorno 3D se ha sugerido el uso de las manos y de dispositivos sensores de profundidad y movimiento. Uno de esos dispositivos es el Microsoft Kinect. El objetivo de este proyecto es utilizar esta tecnología para implementar un API donde integramos las librerías del Kinect con las de procesamiento digital de imágenes obteniendo nuevas funciones que nos permita el desarrollo de una aplicación 3D con esta nueva forma de interacción.

2. Estado del arte

Alex Kipman, creó en el 2008 un dispositivo que capturaba el movimiento de los objetos. Este fue el prototipo de Kinect que posteriormente lo desarrolló Microsoft. Andrew Blake, director general de Microsoft Research en Cambridge, lideró el equipo que desarrolló el software que utiliza el kinect para el Xbox 360 (tracking del esqueleto humano).

Algunas aplicaciones que se han realizado utilizando el Kinect, relacionadas con el presente trabajo se muestran a continuación. En la figura 1 un sistema que reconoce gestos de números, desarrollado por Harishankar Narayana[7]. Para esta aplicación se utilizaron los controladores del Kinect para Linux y librerías libres libfreeenct [8] o OpenNI [9].

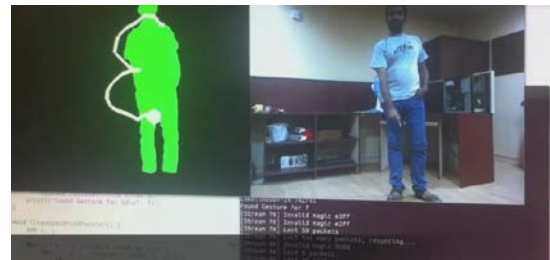


Figura 1: Muestra del reconocimiento del número 3 en este proyecto.

La empresa alemana Evoluce [10] que se especializa en tecnología en pantallas multi – táctiles, desarrolló una aplicación donde se puede jugar solitario y mover imágenes en el sistema operativo Windows 7 (ver Figura 2), utilizando kinect en conjunto con las librerías OpenNi y Windows OS.



Figura 2: Muestra de la interacción con imágenes en Windows 7.

En el Instituto de tecnología de Stockolmo [11], se utilizó el Kinect de Microsoft para navegar en Google Earth. El usuario puede usar su cuerpo para mover la vista del Google Earth (Ver Figura 3). Para este proyecto se utilizó el SDK de Kinect y Microsoft Visual Studio Express C# 2010.



Figura 3: Muestra del usuario utilizando Google Earth.

Las aplicaciones desarrolladas utilizan las librerías para comunicarse con el Kinect, implementando desde cero las funciones que se necesiten. Estas aplicaciones se las puede desarrollar partiendo desde un nivel mayor de abstracción. Ninguna de las aplicaciones desarrolladas que se muestran ofrece esta característica.

Mediante el uso de un API, desarrollar aplicaciones similares a las mostradas se las podría realizar en menos tiempo, con el uso de funciones implementadas y listas para utilizar, además de permitir añadir nuevas funcionalidades.

3. Análisis y Diseño del sistema

Luego de revisar las herramientas existentes y todo lo desarrollado hasta el momento utilizando el Kinect, en esta sección se describen los requerimientos funcionales y no funcionales para el desarrollo de nuestra API.

Requerimientos funcionales

El conjunto de librerías (API) a ser desarrollada deberá cumplir con las siguientes funcionalidades:

- a) Detectar con una precisión de 4cm en profundidad los movimientos lentos y suaves de las manos del usuario.
- b) Permitir al usuario el control los movimientos de rotación del objeto 3D alrededor del eje Y; y el acercamiento y alejamiento de la cámara a través de los movimientos de las manos.

- c) Detectar como gestos de comandos el desplazar la mano a la derecha o izquierda, el extender y recoger brazos.
- d) Permitir el acercamiento o alejamiento del objeto en 3D, separando las manos para acercar y juntándolas para alejar el objeto.
- e) Permitir que una persona utilice la aplicación entre 100 cm y 200 cm de distancia con respecto al Kinect.
- f) Aceptar la carga de modelos 3D con extensión .3ds (es un formato abierto).
- g) Permitir la ejecución de comandos de Microsoft Windows usando las librerías nativas de Windows, como mover el puntero del ratón o la acción de presionar una tecla.

Requerimientos no funcionales

Como requerimientos no funcionales se deben considerar:

- a) El API desarrollado deberá funcionar y hacer uso de los recursos de hardware de un computador con al menos un procesador de doble núcleo y con una tarjeta gráfica con GPU, para un mejor rendimiento y con un tiempo de respuesta para el análisis de al menos 30 imágenes por segundo.
- b) El sistema operativo deberá ser Windows 7 32/64 bits, para ejecutar la aplicación.
- c) Requerirá que se conecte al computador un Kinect de Microsoft para Xbox 360.
- d) Seleccionar el plan de energía de Windows 7, la opción de Alto rendimiento.
- e) El área desde el Kinect hasta el usuario debe estar despejada, sin obstrucción de algún tipo.

4. Diseño del API

La arquitectura del API está diseñada para utilizar sus propias funciones, además de poder usar otras librerías directamente y así implementar nuevas funcionalidades.

En base a los requerimientos funcionales y no funcionales, y habiendo revisado previamente las librerías existentes que pudieran trabajar con el Kinect de Microsoft, se pudo establecer una arquitectura para nuestro API. Esta arquitectura está orientada al control de la aplicación y al uso de comandos (como mover el puntero del ratón o presionar una tecla) de Microsoft Windows mediante la detección de gestos y movimientos de las manos tal como se indica en la Figura 4.

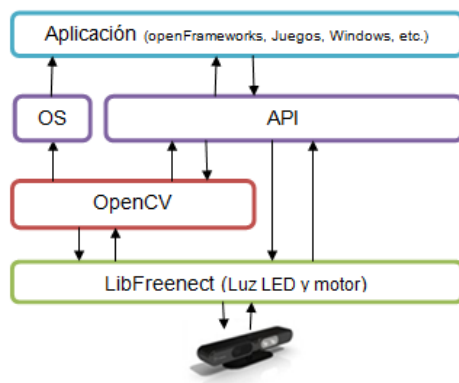


Figura 4: Arquitectura del API.

Cada función del API llama o utiliza las librerías de: procesamiento digital de imágenes, control de las funcionalidades del Kinect de Microsoft, visualización de modelos 3D y control de acciones de Microsoft Windows por medio de código.

También hace uso de las siguientes librerías: libfreenect, OpenCV y OpenFrameworks, por lo que estas deben ser previamente instaladas.

Un factor importante del API, es la utilización de librerías de código abierto y multiplataforma, para que puedan ser utilizados en diferentes sistemas operativos. Las siguientes librerías fueron seleccionadas:

- Libfreenect
- OpenFrameworks
- OpenCV

Estas librerías son compatibles al ser integradas y pueden ser utilizadas en Linux, Mac OS y Microsoft Windows.

En nuestra implementación todas las funciones de la API comienzan con las letras “kl” como prefijo haciendo referencia a las iniciales en inglés de Kinect library.

Para la demostración de integración del API en un programa, se desarrolló una aplicación que permite visualizar y manipular un modelo 3D en pantalla con extensión .3ds (ver Figura 5), los círculos de color verde aparecen cuando se detectan las manos y sirven para retroalimentación del usuario que la está utilizando.



Figura 5: Vista de la pantalla cuando se detectan las 2 manos.

Para determinar la profundidad de los objetos, se utilizó un rango de color en RGB que va desde 0 al 255 como se muestra en la Figura 6, siendo lo más blanco lo más cerca y lo más rojo lo más lejos. Se trabajó con la librería libfreenect que permite determinar la distancia en profundidad por medio de rangos de precisión.



Figura 6: Muestra del rango de color en Rojo en el Kinect.

En este estudio la precisión está determinada por el error que hay para determinar la profundidad de los objetos del kinect en cada caso.

Caso 0: De 49 hasta 50 cm – precisión de 5 mm

Caso 1: De 50 hasta 60 cm – precisión de 1 cm

Caso 2: De 60 hasta 100 cm – precisión de 1 cm

Caso 3: De 100 cm hasta 500 cm

De 100 cm a 150 cm – precisión de 2 cm

De 150 cm a 190 cm – precisión de 4 cm

De 190 cm a 240 cm – precisión de 5 cm

De 240 cm a 330 cm – precisión de 9 cm

De 330 cm a 430 cm – precisión de 18 cm

De 430 cm a 500 cm – precisión de 20 cm

Caso 4: De 500 hasta 1000 cm – precisión es mayor a 20 cm y aumenta a mayor distancia.

Cada caso tiene su rango de precisión, para la utilización de este proyecto se utilizó como precisión de profundidad de 4 cm, dejando lineal el rango de precisión entre 60 cm y 200 cm.

4.1 Diseño de plan de pruebas

Para comprobar el grado de precisión, y determinar el grado de sensibilidad y control que se puede tener con diferentes usuarios, se diseñó un conjunto de pruebas de usabilidad de la aplicación.

Las pruebas consistieron en determinar la experiencia del usuario a 3 diferentes distancias con respecto al Kinect (110 cm, 150 cm y 190 cm).

La altura a la que se trabajó con el Kinect fue de 70 cm con respecto al piso.

Las pruebas constaron de 3 partes:

1.- Un cuestionario para determinar qué gestos realizaría el usuario para mover y realizar un acercamiento o alejamiento de un objeto 3D, pero sin ninguna previa instrucción.

2.- Un conjunto de pruebas practicas para analizar la precisión, sensibilidad y control de los gestos diseñados y utilizados en la aplicación.

3.- Un cuestionario al final de las pruebas practicas, para recoger datos sobre el modo de trabajar con la aplicación y poder determinar si los gestos utilizados fueron los adecuados, si los recordaría con facilidad y si el usuario se sintió bien (bajo tensión, satisfecho, molesto, impaciente, frustrado, bien) al terminar de usar la aplicación.

Para el conjunto de pruebas cada acción implementada por el API que puede realizar el usuario se la identificó con una letra, para poder distinguirla en el cuestionario como se muestra a continuación:

a: Para rotar a la derecha el objeto 3D, usa la mano (izquierda o derecha). Coloca la mano en el pecho, extiende el brazo hacia adelante, muévelo a la derecha lentamente y regresa la mano junto al pecho.

b: Para rotar a la izquierda el objeto 3D, usa la mano (izquierda o derecha). Coloca la mano en el pecho, extiende el brazo hacia adelante, muévelo a la izquierda lentamente y regresa la mano junto al pecho.

c: Para acercar el objeto 3D (zoom in), junta las dos manos junto al pecho, extiéndelas hacia adelante, sepáralas y regresa las manos junto al pecho.

d: Para alejar el objeto 3D (zoom out), extiende los brazos hacia los lados, muévelos extendidos hacia adelante y regresa las manos junto al pecho.

Las pruebas se aplicaron a un grupo de menores de 18 años (niños y jóvenes) y otro de mayores de 18 años (adultos).

5. Pruebas y Análisis de resultados

Con los datos obtenidos de la primera prueba, para determinar que gestos utilizarían las personas para mover, acercar o alejar un objeto en 3D en la aplicación, se pudo comprobar que las personas

utilizaban gestos similares a los utilizados en las pantallas táctiles, como el ampliar una imagen con dos dedos separándolos y reducir la imagen a unir los dos dedos. El mover los dedos hacia los lados para rotar un objeto 3D en los dispositivos táctiles, fue un gesto similar para rotar el objeto utilizando la mano.

Las pruebas de precisión, sensibilidad y control de los gestos que se realizaron, dieron como resultado una distancia óptima de 190 cm.

Para la precisión (ver figura 7.1, 7.2, 7.3 y 7.4) se puede apreciar, para las 4 encuestas, que las personas que utilizaron la aplicación les parecía que tuvieron una mejor experiencia a una distancia de 190 cm desde la posición del kinect.

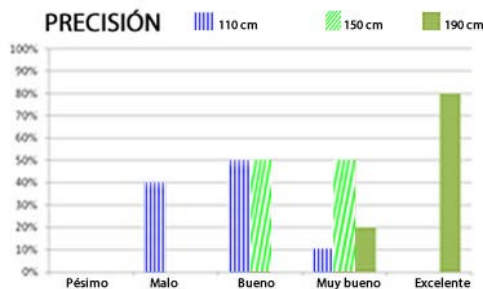


Figura 7.1: Datos obtenidos de los usuarios sobre la precisión de la aplicación. a: Para rotar a la derecha el objeto 3D, usa la mano (izquierda o derecha).

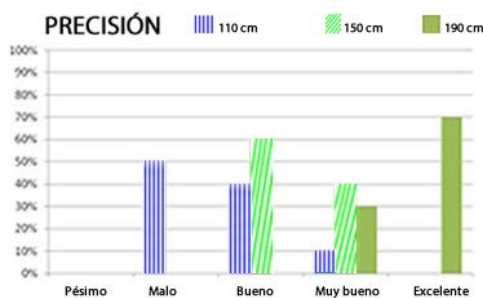


Figura 7.2: Datos obtenidos de los usuarios sobre la precisión de la aplicación. b: Para rotar a la izquierda el objeto 3D, usa la mano (izquierda o derecha).

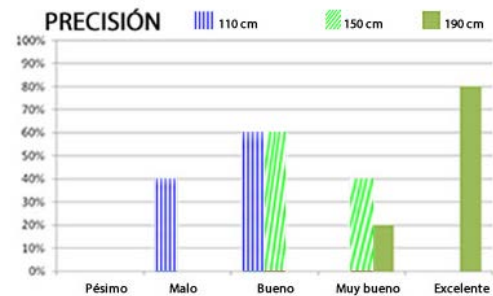


Figura 7.3: Datos obtenidos de los usuarios sobre la precisión de la aplicación. c: Para acercar el objeto 3D (zoom in).

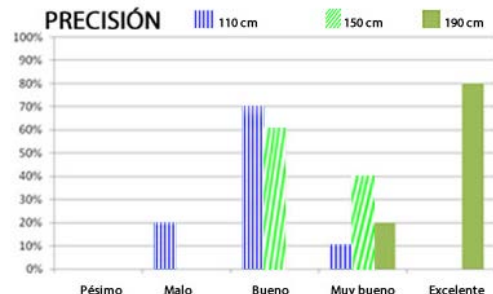


Figura 7.4: Datos obtenidos de los usuarios sobre la precisión de la aplicación. d: Para alejar el objeto 3D (zoom out).

En los gráficos de las figuras 8.1, 8.2, 8.3 y 8.4 observamos la sensibilidad de reacción a los diversos gestos de la aplicación. Se puede apreciar que a una distancia cerca de 110 cm es muy sensible y a las personas no les gustaba mucho, excepto cuando había que alejar el objeto les parecía bueno que se alejara rápidamente.

La sensibilidad a una distancia de 150 cm les pareció buena para todas las acciones. En las pruebas a una distancia mayor de 150 cm las acciones de rotar y acercar el objeto les parecía de muy bueno a excelente, pero cuando tenían que alejar, les parecía mejor a una distancia menor porque la función de alejar la hacía más rápida.

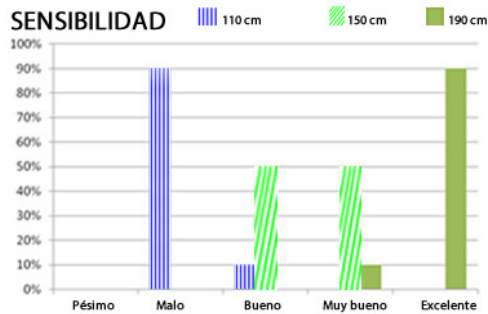


Figura 8.1: Datos obtenidos de los usuarios sobre la sensibilidad de la aplicación. a: Para rotar a la derecha el objeto 3D, usa la mano (izquierda o derecha).

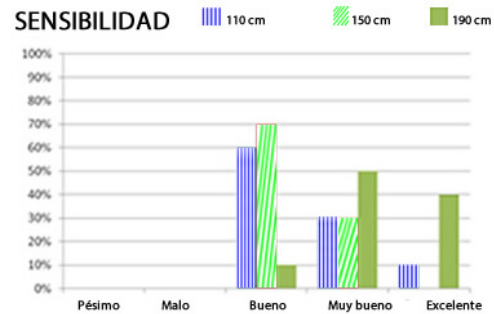


Figura 8.4: Datos obtenidos de los usuarios sobre la sensibilidad de la aplicación. d: Para alejar el objeto 3D (zoom out).

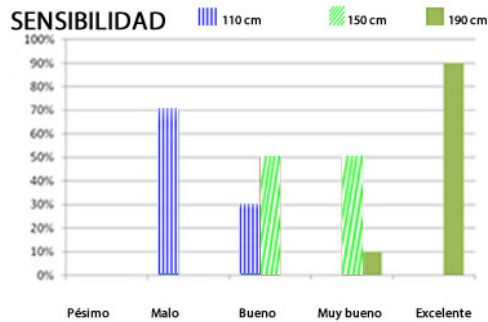


Figura 8.2: Datos obtenidos de los usuarios sobre la sensibilidad de la aplicación. b: Para rotar a la izquierda el objeto 3D, usa la mano (izquierda o derecha).

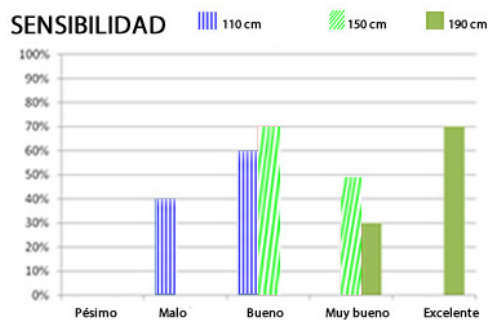


Figura 8.3: Datos obtenidos de los usuarios sobre la sensibilidad de la aplicación. c: Para acercar el objeto 3D (zoom in).

El control en todas las pruebas estuvo bien. Donde hay mayor contundencia es a una distancia mayor a 190 cm y menor a 200 cm, donde el control de los gestos es excelente según los encuestados (ver figura 9.1, 9.2, 9.3 y 9.4). El control a una menor distancia estaba bien pero a una mayor distancia es un poco más estable y agradable el manejo.

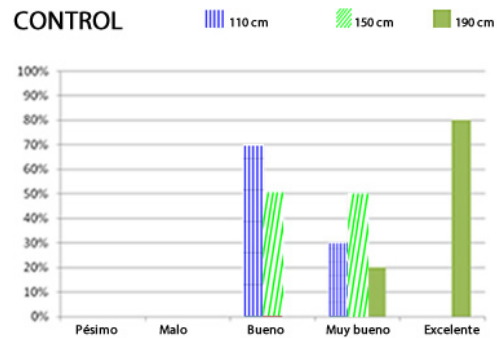


Figura 9.1: Datos obtenidos de los usuarios sobre la control de la aplicación. a: Para rotar a la derecha el objeto 3D, usa la mano (izquierda o derecha).

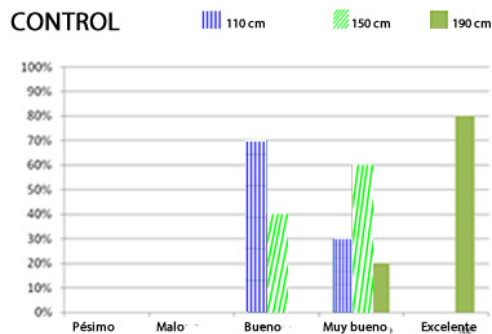


Figura 9.2: Datos obtenidos de los usuarios sobre la control de la aplicación. b: Para rotar a la izquierda el objeto 3D, usa la mano (izquierda o derecha).

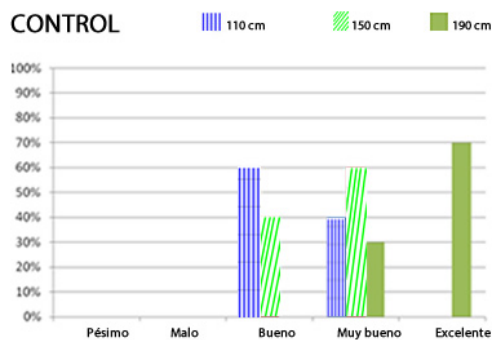


Figura 9.3: Datos obtenidos de los usuarios sobre la control de la aplicación. c: Para acercar el objeto 3D (zoom in).

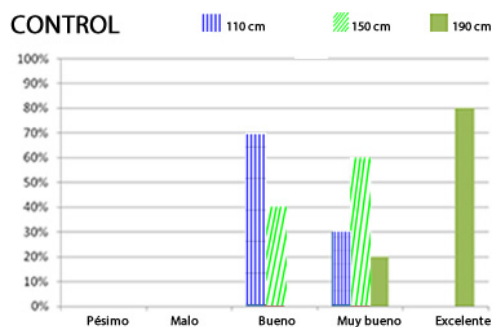


Figura 9.4: Datos obtenidos de los usuarios sobre la control de la aplicación. d: Para alejar el objeto 3D (zoom out).

El total de las personas que utilizaron la aplicación mencionaron que recordarían los gestos utilizados, porque se asemejan a los empleados en las tabletas que utilizan tecnología táctil y que ya tenían previa experiencia de utilización. De igual manera a las personas que no ha utilizado una tableta también se les hacía fácil recordar.

Con respecto al control de la aplicación el 90% de las personas se encuentran bien o satisfechas al usar la aplicación y un 10% de las personas frustradas.

Pruebas de funcionalidad de la aplicación

Todas las pruebas fueron realizadas en un ordenador con procesador Intel Core i7, 4GB de memoria RAM, tarjeta gráfica GeForce GT 650M, Sistema Operativo Windows 7 de 64 bits. El número de imágenes que procesaba la aplicación era de 30 imágenes por segundo.

La aplicación funcionaba correctamente después de que cada persona lo utilizaba, se hicieron validaciones, para que no haya exceso de consumo de memoria, además cargar archivos de hasta 10 megabytes de tamaño, mayores a esto se produce un error, por parte de la librería de OpenFrameworks.

6. Conclusiones

- Con los datos obtenidos de las pruebas de usabilidad, podemos sacar como conclusiones que la mejor precisión, sensibilidad y control de la aplicación se obtiene dentro de un rango de 180 a 190 cm con respecto al Kinect.
- Los gestos implementados fueron los adecuados para la aplicación porque todos los que realizaron las pruebas estuvieron de acuerdo que recordarían fácilmente los movimientos realizados si volvieran a interactuar con la aplicación.
- Este API permite implementar diversas aplicaciones utilizando sus funciones.
- La interacción usando los brazos muchos minutos cansa a la persona.
- La elección de las librerías utilizadas permiten a este API ser multiplataforma.



7. Recomendaciones

- Para futuros proyectos se pueden implementar nuevos algoritmos, que utilicen técnicas de reconocimiento de imágenes para nuevos gestos.
- Se podría también utilizar un método que incluya el reconocimiento en la imagen con la cámara RGB para determinar la posición de los objetos, mediante la comparación de la posición del pixel en la imagen, con la imagen de la cámara de profundidad (Depth camera).
- Hacer las respectivas instalaciones de OpenCV, OpenFrameworks y Libfreenect, en Linux y Mac OS, para utilizar el API en estas plataformas.
- Agregar las funciones para controlar el puntero y las teclas del sistema operativo de Linux o Mac OS, utilizando el IDE nativo para su integración con el API.

8. Referencias

[1] **Qurren**, Imagen ratón, Un ratón de computador con una rueda en medio de los botones, Abril 11 2006. Disponible en: http://commons.wikimedia.org/wiki/File:Wheel_mouse.JPG?uselang=es

[2] **Wtshymanski**, imagen teclado, Teclado de un computador, un Televideo 925, Enero 3 2006. Disponible en: <http://commons.wikimedia.org/wiki/File:Televideo925Terminal.jpg?uselang=es>

[3] **Microsoft**, Kinect dispositivo de entrada para la Xbox 360, Noviembre 4 2010. Disponible en: <http://www.xbox.com/en-US/KINECT>

[4] **Wikipedia**, Frustum definición en gráficos por computadora y en geometría, Junio 30 2012. Disponible en: <http://es.wikipedia.org/wiki/Frustum>

[5] **Wikipedia**, Frame definición para este estudio, Mayo 14 2012. Disponible en: <http://es.wikipedia.org/wiki/Frame>.

[6] **Esqueda Elizondo José Jaime y Palafox Maestre Luis Enrique**, Fundamentos de procesamiento de imágenes, Motion Detection Libro de Procesamiento digital de imágenes,

Universidad Autónoma de Baja California, Edición 2005. Disponible en: <http://books.google.com.ec/books?id=h4Gj8GuwPVkC&lpg=PA1&pg=PA1#v=onepage&q&f=false>

[7] **Harishankar Narayanan**, Reconocimiento de gestos, desarrollador de este proyecto, Julio 9 2012. Disponible en: <http://www.youtube.com/user/harishankarn>

[8] **Comunidad OpenKinect**, librerías de código abierto para Kinect, Abril 7 2012. Disponible en: http://openKinect.org/wiki/Main_Page

[9] **Code Laboratories**, CL NUI, Librería para usar el Kinect de Microsoft en Windows 7, Diciembre 8 2010. Disponible en: <http://codelaboratories.com/kb/nui>

[10] **Evoluce GmbH**, Empresa alemana dedicada a tecnología multi – touch e interfaces naturales de usuario, Septiembre 7 2012. Disponible en: <http://www.evolute.com>

[11] **Royal Institute of Technology**, Kinect + Google Earth, Proyecto de curso de otoño, Stockholm, Diciembre 11 2011. Disponible en: <http://www.youtube.com/watch?v=5vYnwiCkk-g>