Revista Ingeniería y Región
2013(10):67-76

Software que captura, por medio de Kinect, los datos de señas manuales y los traduce a texto

Software that seizes through Kinect, the data of manual systems and it translates to text

Claudia Patricia Rodríguez[1], Jhon Alexander Pineda[2] y Diego Fabián Sánchez[3]

Resumen

A continuación se presentan los resultados del proyecto que tiene como objetivo desarrollar un software, que permite, por medio de la cámara de profundidad de Kinect, capturar la información de las señas manuales, que una persona realice frente a Kinect, y mostrar el texto equivalente a esta seña. El software presentado aún está en la fase de prototipo, son varias las mejoras que en él se debe y se pueden implementar con el fin de obtener mejores resultados en el reconocimiento de señas.

El prototipo desarrollado es capaz de tomar los puntos de una escena captada por medio del sensor de profundidad de Kinect, aplicar un filtro para eliminar datos no necesarios y posteriormente mostrar el Mesh o maya que reconstruye la imagen de la escena final en 2D, estableciendo la diferencia de distancias con el cambio de color. El aplicativo detecta la localización de la o las manos en el Mesh, luego toma imágenes del gesto manual y las compara con imágenes almacenadas en archivo, estas cuentan con la traducción respectiva, si encuentra una imagen con alto grado de coincidencia se devuelve el texto correspondiente. Los resultados de la investigación permiten reconocer las fases, librerías y plataformas óptimas para el procesamiento de imágenes implementado en el prototipo.

Palabras clave: Kinect; nubes de puntos; OpenCV; procesamiento de imágenes; señas manuales; PLC.

Abstract

Proceed, are presented the results of the Project that has as a purpose developing a Software, that allows, through the depth camera of Kinect, seize the information of the manual signals, that a person carries out in front of Kinect, and shows the text equivalent of this sign. The software presented it is still en in the stage of prototype, are various the improvements that we should and can implement with the purpose for obtaining better results in signs recognition.

The prototype developed it is able for taking the scene points captured through the depth Device of Kinect, apply a screening for delete unnecessary piece of information and after, shows the mesh (something that rebuilds the picture of the final scene in 2D), establishing the difference of distances with the color change. This detects the location of the hand or hands in Mesh, then takes pictures of manual signals and compares it with the pictures stored in the file, these have the respective translation, if it finds a picture with a higher level of coincidence it returns with the correct text. The results of the investigation allow recognize the stages, and ideal platforms processing the implemented pictures in the prototype.

Key words: Kinect; OpenCv, pictures processing, manual signals, PPC.

1 Especialista en Bases de Datos y Gerencia educativa. Docente UNISANGIL sede Chiquinquirá Calle 18 # 12 18.E-mail: crodriguez2@unisangil.edu.co.
2 Estudiante Ingeniería de Sistemas de UNISANGIL. Sede Chiquinquirá. E-mail: jhonalex196@unisangil.edu.co
3 Estudiante Ingeniería de Sistemas de UNISANGIL. Sede Chiquinquirá. E-mail: diegofabian@unisangil.edu.co

1. Introducción

El desarrollo de herramientas tecnológicas lleva consigo el desarrollo de las regiones, las orientaciones y diseños propuestos han permitido que personas con discapacidades puedan desenvolverse mejor en un medio en el cual no está preparado para ellos; sin embargo en algunos campos el desarrollo ha sido poco; por ejemplo, las personas no oyentes sufren las consecuencias de manejar un lenguaje diferente al oralismo, ellos deben aprender la dactilología, lenguaje basado en las posiciones de la mano, no obstante, mucha gente de la comunidad en general, lo desconoce, y es ahí donde la comunicación interpersonal se ve truncada y las personas no oyentes alejadas de la sociedad. La tecnología desarrollada para esta población se ha llevado por un solo sentido, es decir, se han diseñado TIC para enseñarles un sistema de comunicación, dactilología, utilizando diferentes didácticas, pero son muy pocos los estudios realizados y enfocados en traducir a texto estas posiciones de la mano en tiempo real, con el fin de hacerlo comprensible para quienes lo desconocen.

La tecnología útil y que está dispuesta para tal fin, es el análisis de imágenes, el cual está definido en el artículo "Procesamiento de imágenes digitales", de la Universidad Autónoma de Puebla, en México; como un proceso que consiste en la extracción de características y propiedades de las imágenes, así como la clasificación, identificación y reconocimiento de patrones; indicando la importancia científico-técnica de este campo no solo en las ciencias sino principalmente en la sociedad; así mismo, las empresas y desarrolladores de software han enfocado sus estudios en el reconocimiento facial, la principal razón es la necesidad de aplicaciones de seguridad y vigilancia utilizadas en diferentes contextos. Los sistemas de reconocimiento facial pertenecen a las técnicas FRT (Face Recognition Thecniques), las cuales pueden clasificarse en dos categorías según el tipo de aproximación, holística o analítica. La aproximación holística (método de las EigenFaces) considera las propiedades globales del patrón, mientras que la segunda (eingenfeautres) considera un conjunto de características geométricas de la cara. El proceso utilizado en este tipo de identificación se clasifica en:

- Detección: el cual localiza la cara humana dentro de una imagen capturada por una cámara, toma esa cara y la aísla de los otros objetos en la imagen.

- Reconocimiento: compara la imagen facial capturada con imágenes que han sido guardadas en una base de datos.

La tecnología de reconocimiento básico involucra tanto a los 'eigenfeatures' (métrica facial, técnica analítica) como a los 'EigenFaces'(técnica holística).

Y aunque, dentro del campo del análisis de imágenes, los adelantos para el análisis de la mano son muy pocos, en el proyecto se reenfocaron e implementaron los métodos y técnicas propias del reconocimiento facial para el análisis de las posiciones de la mano. Como dispositivo de captura se utilizó Kinect, este es un sensor de profundidad de bajo costo, utilizado en muchas aplicaciones cada vez más innovadoras, gracias a la esencia de permitir la visión por computadora; además brinda mayor cantidad de información procesable para el ordenador en relación con las imágenes típicas como las RGB.

2. Metodología

El primer paso fue identificar las características técnicas, programación y configuración de Kinect, con el fin de establecer su correcto funcionamiento; se hizo necesario utilizar los controladores Primesense (Primesense, 2013) y OpenNi (Openni, 2013) para la configuración del dispositivo en el equipo de cómputo. Una vez configurado el dispositivo se procedió a probarlo para ello se desarrolló un aplicativo.

La interfaz de la aplicación contaba con botones que permitían inicializar el sensor, activar el flujo de la cámara de profundidad a una resolución de 640x480 y una frecuencia de 30 fps (imágenes por segundo), utilizar la función Skeleton (Kinectfordevelopers, 2013) para detectar las articulaciones de un cuerpo humanoide, controlar la inclinación que abarca de -27 a 27 grados.

Como resultado de las pruebas, el aplicativo muestra tres imágenes procesadas por el sensor de Kinect (Figura 1) , las cuales corresponden a:

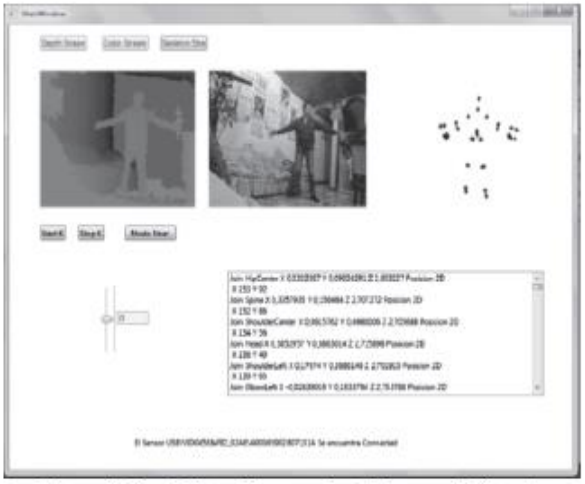

Figura 1. Resultados de las pruebas del sensor de Kinect.

- DepthStream o flujo de profundidad
- Flujo RGB
- Función Skeleton

Una vez probado el sensor se inicia la programación; Point Cloud Library (PCL) (Pointclouds, 2012) y OpenCV con el wrapper Emgu (EMGU, 2012) fueron las herramientas de desarrollo seleccionadas, debido a la amplia gama de funciones implementadas para el procesamiento de imágenes.

2.1 Point cloud library o nubes de puntos. Es un proceso compuesto por dos fases: Adquisición de puntos y Filtrado Pass Through.

Adquisición de los puntos: Representa la información de la escena como una constelación de puntos; datos que necesariamente son capturados por el sensor de profundidad, creando un patrón de puntos que corresponden a la profundidad de los objetos (Figura 2).

Con el fin de conocer la distancia óptima de captura de datos; se realizaron experimentos en los cuales se capturaban 3 escenas consecutivas de la palma de la mano, a distancias de 30 cm, 70 cm y 100 cm con respecto a la cámara de Kinect.

Se concluyó que a distancias menores de 30 cm no es posible captar los puntos propios de la mano, ya que a esta distancia no hay convergencia entre la cámara de profundidad y el láser, quienes tienen un umbral mínimo de 70cm, si la mano se sitúa antes del umbral genera una sombra que no se puede procesar; si se aumenta la

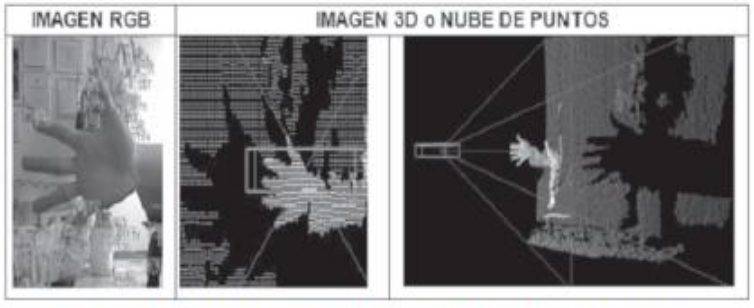

Figura 2. Comparación entre una imagen RGB y una Nube de Puntos.

Figura 4. Imagen Filtrada.

131387	1.013557	-0.238274	-0.405156
131388	0.996055	-0.245894	-0.398160
131389	0.976374	-0.252537	-0.390293
131390	0.952167	-0.260698	-0.380617
131391	0.931621	-0.266047	-0.372404
131392	0.931621	-0.286428	-0.372404
131393	0.926619	-0.306722	-0.370404
131394	0.929114	-0.327875	-0.371401
131395	0.916774	-0.343577	-0.366469
131396	0.904752	-0.358865	-0.361663
131397	0.900029	-0.379712	-0.359775
131398			

Figura 3. Información de los puntos capturados de una imagen..

distancia, el sensor captura una mayor cantidad de puntos de la escena, sin embargo la densidad de puntos sobre la mano disminuye, por lo tanto se establece que la distancia óptima de captura es 70 cm respecto a la cámara de Kinect, ya que en ese punto se tiene menos grado de dispersión, y la densidad de puntos (Figura 3) es significativa para definir la silueta morfológica de una mano.

Los archivos con la información de los puntos contienen aproximadamente 1'300.000 mil registros con números reales distribuidos en tres columnas, correspondientes a las coordenadas de los puntos de la escena, llegando a pesar en promedio 50 Megabytes, esta información corresponde al total de la escena captada, es decir, no solo aparecerá la información de la mano sino también puntos del medio que rodea la misma, por tal razón se hizo necesario implementar un mecanismo para la reducción de la nube de puntos, este proceso se conoce como filtrado. Se utilizó el filtro Pass Through (Universidad de Málag, 2010) el cual permite desechar los puntos que están por encima de una distancia establecida, se hizo una reducción de 750 mm sobre el eje Z, el resultado permitió definir la silueta morfológica de la mano que se presenta en la figura 4.

Inicialmente se implementó el algoritmo RANSAC (Zuliani, 2012), el cual escoge un punto y busca los puntos más cercanos que están dentro de un rango establecido, los que no cumplen con el criterio son eliminados, sin embargo no resulto optimo porque toma puntos que no hacen parte de la mano pero están por debajo del umbral del algoritmo, además, el consumo de recursos computacionales y tiempo es alto para su procesamiento en tiempo real. También se hizo uso de Matlab, pero no fue posible implementar el proceso en tiempo real.

El procesamiento de imágenes con nubes de puntos fue abandonado porque al presentar un mayor grado de precisión en la comparación de imágenes, exige que los puntos y la cantidad de los mismos sean iguales, en la imagen dada en tiempo real y la almacenada en la base de datos, lo cual no es posible debido a la diferencia en el tamaño de las manos de los usuarios y la información de la escena o medio en donde este el usuario; estos aspectos hacen que la cantidad de puntos, así se trate de una misma seña, sean diferentes. Además el tratamiento de nubes de puntos trae consigo un alto costo computacional, ya que amerita el manejo de programación paralela, la cual consiste en la explotación del poder computacional de las tarjetas aceleradoras de video (GPU) en el procesamiento de los algoritmos.

2.2. OpenCV (Open Source Computer Vision). Esta herramienta fue seleccionada por el paquete de librerías que contiene para el procesamiento de imágenes.

Se plantearon las siguientes fases para el desarrollo del intérprete (Figura 5).

A. Seña sin identificar. Fase enfocada en el reconocimiento de la estructura de la mano, la función principal de este proceso fue HandDataSource; la distancia de toma es de 75cm, ya que de acuerdo a las especificaciones técnicas de Kinect, el umbral mínimo de reconocimiento de profundidad es 70cm; como resultado se visualiza el clustering de la mano (Figura 6), proceso que consiste en trazar líneas sobre los bordes de la mano, como aparece en seguida:

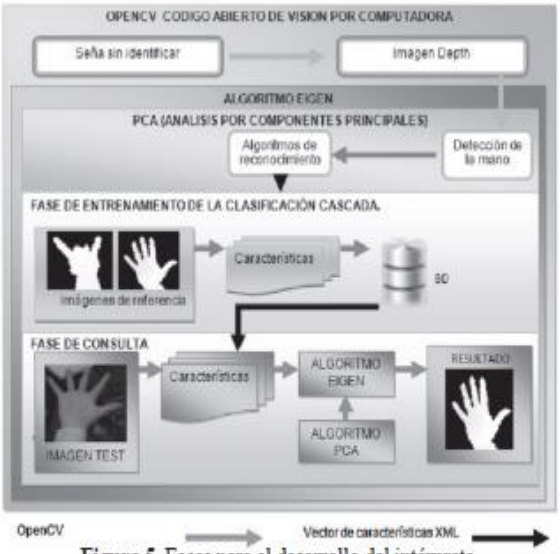

Figura 5. Fases para el desarrollo del intérprete.

Figura 6. Clustering de la mano.

B. Imagen Depth. Representa los objetos en escala de grises donde las zonas más oscuras equivalen a las mayores distancias con respecto al sensor; esta escala permite un proceso de comparación rápido y preciso ya que tendrá menor cantidad de datos, en contraste con las imágenes RGB, las cuales contienen demasiadas características en el color (pixeles) y la información varía según el ambiente.

C. Algoritmo EIGEN

1. *Detección de la mano:* localiza en la imagen Depth la/las manos abiertas o cerradas, con el fin de devolver características como ubicación y tamaño. En el proyecto se hizo uso de los métodos basados en el aspecto, con algoritmos como el Eigenfaces (Turk, 2001), Fisherfaces y Local Binary Patterns Histograms, estas clases detectan y extraen imágenes de la pose de la mano haciendo uso del método PCA (Principal Components Analysis) (Viola and Jones, 2001) el cual se describe a continuación:

2. *Análisis por componentes principales (PCA):* busca exhaustivamente en la imagen de entrada una mano, reduciendo su escala según características de la mano; descarta las regiones donde no existe la posibilidad de encontrarla, esto se realiza mediante la aplicación de heurísticas sencillas que eliminan las regiones no uniformes; el resultado de este proceso es la ubicación de la mano, estos datos alimentaran la siguiente fase, denominada Haarcascade (Universidad del Quindío, 2010).

D. HaarCascade y Base de Datos. Las redes neuronales están definidas como una estructura formada por muchos procesadores simples llamados nodos o neuronas, conectados por medio de canales de comunicación o conexiones; estas redes deben tener asociadas reglas de aprendizaje o entrenamiento (Romero y Caro, 2011); estas redes aprenden a partir de ejemplos (Vidal, 2001).

HaarCascade es considerado una red neuronal porque clasifica la presencia o ausencia de la mano; la fase previa de entrenamiento se basa en la implementación de imágenes positivas y negativas, para posteriormente realizar el reconocimiento del gesto manual. Las imágenes positivas son aquellas que contienen la información de los gestos de la mano, se utilizan en el entrenamiento de la red neuronal como referencia para la extracción de características o descriptores; mientras que las negativas no tienen relación con la silueta morfológica de la mano, por lo tanto son consideradas en el proceso de ubicación de la misma, ya que se tiene una referencia de lo que no es una mano. Las etapas del HaarCascade pueden verse en la figura 7.

Figura 7. Etapas HaarCascade.

- Imagen de Entrada o Preprocesado: imagen en tiempo real tras aplicarle un filtro de distancia para eliminar ruido.

- Detección: Recibe como entrada la imagen preprocesada y devuelve la seña detectada en la imagen. Aplica el algoritmo de detección. Como resultado de este proceso se obtiene la imagen integral, que es una extracción a diferentes escalas de la zona donde se encuentra la mano.

- Extracción de características: Recibe como entrada la imagen integral de la mano o seña manual detectada y devuelve un vector de características. Consiste en obtener la información necesaria para identificar la seña que aparece en la imagen capturada en tiempo real.

- Clasificación: Recibe como entrada el vector de características de la imagen y devuelve la seña de la base de datos a la que más se parece. Para el proyecto se implementaron dos clasificadores, uno para la mano abierta y cerrada y el otro para el reconocimiento de la seña.

La red neuronal implementada está compuesta por cuatro nodos, y fue entrenada con 26 imágenes positivas y 6 negativas, este proceso tardo 11,42seg.

E. Fase de Entrenamiento o Aprendizaje. Se entrena la red neuronal para que funcione como un sistema de reconocimiento; para los clasificadores se crean dos carpetas una correspondiente a las imágenes positivas y la otra a las negativas (Figura 8); estas imágenes son cargadas desde la interfaz del sistema.

Teniendo las imágenes negativas y positivas se procede a generar el archivo con las direcciones de las mismas, que luego son agregadas como parte final del entrenamiento del sistema.

En esta fase también se adicionan señas y su interpretación en texto; las imágenes son captadas por la aplicación que las almacena en un carpeta, en donde para cada seña se realizan 10 capturas, además registra en un

Imágenes positivas	Imágenes negativas
	
Son las señas o imágenes que el sistema debe llamar desde archivo y con las cuales comparara las obtenidas en tiempo real.	Corresponden a imágenes donde no existe presencia de la mano.

Figura 8. Ejemplos de imágenes positivas y negativas.

archivo Xaml etiquetas con el nombre de la seña y el nombre del archivo para generar una estructura que usara el training. Para agregar estas señas el aplicativo cuenta con una ventana que permite adicionar y digitar su significado.

3. Resultados

El prototipo está conformado por imágenes de la mano de dos personas que muestran 9 señas básicas definidas en el diccionario básico de lenguaje de Señas Colombiano, elaborado por INSOR, cada foto se encuentra en formato JPG a 8 bits con un tamaño de 100x100 píxeles y una resolución de 96ppp y un peso promedio de 2,05kb.

La interfaz de usuario se muestra en la Figura 9.

Figura 9. Interfaz de usuario del intérprete de señas.

En ella se presenta tres opciones de menú y tres botones de control:

- Opciones de Menú:
  - File: Permite salir del sistema de reconocimiento, detener todos los módulos de Kinect.
  - Train: Carga el formulario para el entrenamiento del reconocedor de señas.
  - PCL: Carga el formulario que permite visualizar la imagen en nubes de puntos.

- Botones de Control:
  - Iniciar: Inicia el dispositivo Kinect, permite visualizar la imagen Depth.
  - Cap. Positivas: Permite el entrenamiento del reconocedor de la mano, agrega imágenes positivas.
  - Cap. Negativas: Permite el entrenamiento del reconocedor de la mano, agrega imágenes negativas.
  - Salir: Permite salir del sistema de reconocimiento.

Se realizaron pruebas con dos personas (Figura 10), cada persona mostraba una serie de señas en donde se midió el tiempo de respuesta del sistema y el porcentaje de error en el reconocimiento de la misma, en el

Figura 10. Esquema del entrenamiento del HaarCascade con su correspondiente resultado.

caso de presentar señas que no estuvieran registradas el sistema no muestra respuesta alguna. Los resultados se muestran en la Tabla 1.

Tabla 1. Resultados de la detección de la señal

	Persona 1	Persona 2
No Imágenes	9	9
No de detecciones	9	8
No de Fallos	0	1
% de Detección	100%	89%
Tiempo de Detección Promedio	0.0016s	0.0020s

Además de la identificación de señas el prototipo cuenta con la opción TRAIN, la cual permite agregar nuevas señas al sistema.

En la imagen se muestra un ejemplo en el cual se agrega la seña correspondiente a la letra P. Es necesario colocarse a una distancia de 75 cm, para que Kinect pueda reconocer la seña; la opción Agregar seña permite reforzar el reconocimiento de una seña previamente almacenada en el sistema, permitiendo agregar una nueva imagen con la seña, se debe recordar que el reconocimiento de una seña es óptimo si se almacenan por lo menos 10 imágenes de la misma seña, si se utiliza esta opción para agregar una nueva seña, solo almacenara una imagen por lo tanto el reconocimiento no será optimo, es decir, no reconocerá la seña cuando el usuario la esté realizando. Capturar 10, es la opción apropiada para agregar una nueva seña, toma diez imágenes de la seña y las almacena para luego utilizarlas en el proceso de reconocimiento.

Adicional existe la opción de eliminar todas las señas que se encuentran en el sistema y así comenzar desde cero.

La opción PCL, el manejo de imágenes con nubes de puntos RGB, es un campo que se deja inicializado para posteriores investigaciones, en el mismo se puede:

- Capturar nube de puntos RGB
- Capturar nube de puntos sin RGB

Visualizar nube RGB - Visualizar nube.

Las nubes visualizadas ya han sido almacenadas en el sistema por medio de la opción capturar, luego son cargadas desde el lugar de almacenamiento (Figura 11).

Las dos últimas opciones permiten agregar imágenes positivas y negativas. En la Figura 12 se muestra la captura de una Imagen negativa en la cual no debe existir una mano.

Las imágenes positivas permite el reconocimiento de una mano en un espacio, no el reconocimiento de una seña.

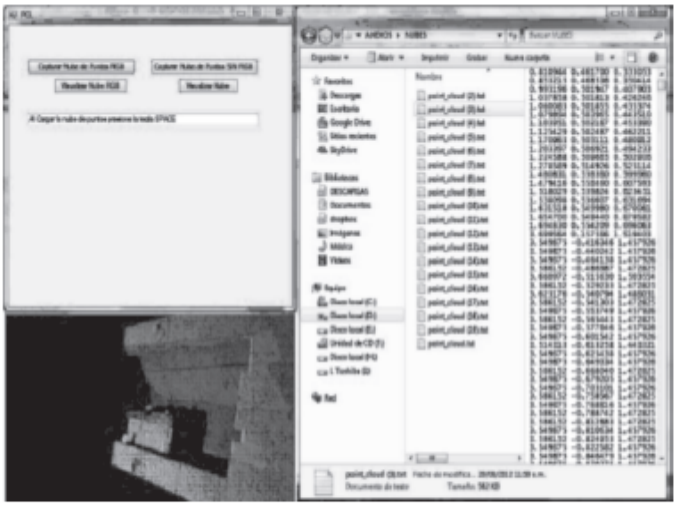

Figura 11. Proceso de captura, almacenamiento y visualización de una nube de puntos.

Figura 12. Proceso captura imagen negativa.

4. Conclusiones

Con la implementación de las librerías que ofrece OpenCV se obtuvieron mejores resultados en el análisis de imágenes debido al gran número de funciones que maneja en lo relacionado a visión por computador, siendo la ejecución menos costosa en cuanto a tiempo y al consumo de recursos computacionales. Mientras el proceso de nubes de puntos trabaja archivos con las coordenadas de cada punto en un espacio 3D, OpenCV permite trabajar con imágenes en escala de grises, con la ventaja de que una imagen de este tipo siempre permanece con la misma información, al contrario de una imagen manejada a color en donde la información depende de la luz reflejada en el objeto. OpenCV realiza el reconocimiento con archivos de clasificación, lo cual amerita el entrenamiento de una red neuronal, por medio de la cual el proceso de reconocimiento es más rápido y sencillo.

El análisis de las posiciones de la mano está basado en los estudios y avances realizados para el reconocimiento de rostros, analizando estos métodos, técnicas y algoritmos se consiguió el diseño final del prototipo.

Kinect es un dispositivo que cumple con los requerimientos necesarios para el diseño del aplicativo, a un costo de adquisición moderado, la desventaja que principalmente se vio fue en el manejo de distancias ya que no captura imágenes por debajo de los 70cm ni por encima de los 4m. Para el aplicativo se estableció una distancia de captura de 75cm.

Los sistemas de reconocimiento tienen hoy en día varias dificultades por superar ya que la información de una imagen de un mismo objeto puede variar de acuerdo al ambiente, a la distancia e inclusive al ángulo de percepción lo que marca sustancialmente los resultados dados por el reconocedor, limitaciones como estas deben ser analizadas para determinar mejoras en la implementación de este tipo de sistemas.

Este proyecto es la puerta abierta a partir del cual se puede profundizar en campos diversos como:

- Traductores de Lenguajes de señas para la población no oyente.
- Sistemas de reconocimiento, análisis y procesamiento de imágenes.
- En la seguridad se podría implementar un software que realizara un reconocimiento de la mano para la identificación de rasgos biométricos.
- Reconocimiento de los dedos e implementación en pantallas y teclados holográficos.
- Soluciones de navegación con visión artificial en vehículos aéreos, terrestres o acuáticos no tripulados.
- Rehabilitación física en la cual la persona tenga que realizar una serie de ejercicios y el Kinect detecte si los está realizando de manera correcta desde la comodidad del hogar o la oficina.

5. Referencias bibliográficas

1. EMGU. 2012. Main Page. Consultado el 3 de Abril de 2012 http://www.emgu.com/wiki/index.php/Main_Page.
2. KINECTFORDEVELOPERS. 2013. Skeleton Tracking – MapSkeletonPointToDepth Deprecated. Consultado el 19 de Febrero de 2012 http://www.kinectfordevelopers.com/2013/01/31/skeleton-tracking-mapskeletonpointtodepth-deprecated/.
3. OPENNI. 2013. What is openni?. Consultado el 28 de marzo de 2012 http://www.openni.org/.
4. POINTCLOUDS. 2012. PCL. Consultado el 19 de marzo de 2012 http://pointclouds.org/documentation/ /.
5. PRIMESENSE. 2013. PrimeSense Documentación. Consultado el 21 de Marzo de 2012. http://www.primesense.com.
6. Romero L. A. y Caro T. C. 2011. Redes Neuronales y Reconocimiento de Patrones.
7. Turk, M., 2001. Eigenfaces for Recognition: Journal of Cognitive Neurosicence, Vol. 3, No. 1, pp. 71-86.
8. Universidad de Málaga, (2010). Pass-Through del tipo de cambio. Consultado el 12 de Mayo de 2012 http://externos.uma.es/cuadernos/pdfs/papeles38.pdf .
9. Universidad del Quindio, 2010. Técnicas para la Detección de Rostros en Secuencias de Imágenes basadas en enfoques Holísticos. Consultado el 4 de Junio de 2012 http://www.uniquindio.edu.co/uniquindio/revistadyp20111013_2/Articulos/5ta%20Edicion/articulo_final.pdf
10. Vidal E. 2001. Aprendizaje y Percepción: Preproceso y Extracción de Características.
11. Viola P. and Jones M. 2001. Rapid Object Detection using a Boosted Cascade of Simple Features. IEEE CVPR.
12. Zuliani M. 2012. RANSAC for Dummies. Consultado el 14 de Agosto de 2012 http://vision.ece.ucsb.edu/~zuliani/Research/RANSAC/docs/RANSAC4Dummies.pdf .