

**Centro de Investigación Científica y de Educación  
Superior de Ensenada, Baja California**



---

**Programa de Posgrado en Ciencias  
en Ciencias de la Computación**

---

**Control de computadora basado en gestos con las manos en  
circunstancias de baja iluminación**

Tesis  
para cubrir parcialmente los requisitos necesarios para obtener el grado de  
Maestro en Ciencias

Presenta:  
**América Ivone Mendoza Morales**

Ensenada, Baja California, México  
2015

Tesis defendida por

## **América Ivone Mendoza Morales**

y aprobada por el siguiente Comité

---

Dr. Vitaly Kober

Director del Comité

Dr. Hugo Hidalgo Silva

Dr. Josué Álvarez Borrego



---

Dra. Ana Isabel Martínez García

Coordinador del Programa de Posgrado en Ciencias de la Computación

---

Dr. Jesús Favela Vara

Director de Estudios de Posgrado

Resumen de la tesis que presenta América Ivone Mendoza Morales como requisito parcial para la obtención del grado de Maestro en Ciencias en Ciencias de la Computación.

**Control de computadora basado en gestos con las manos en circunstancias de baja iluminación**

Resumen aprobado por:

---

Dr. Vitaly Kober

El reconocimiento de gestos con las manos ha sido un tema relevante en distintas áreas de las ciencias de la computación, por ejemplo en HCI es importante pues ayuda a crear una interacción natural entre la computadora y el usuario, por lo que se han desarrollado diversos métodos para encontrar el modelo que funcione en tiempo real y en diversas circunstancias. De manera que se pretende crear un modelo que fusione la información proporcionada por el dispositivo Kinect y haga el reconocimiento de gestos estáticos y dinámicos en tiempo real en circunstancias de baja iluminación y cuando existe oclusión. Dicho modelo será aplicado para crear un sistema que sirva como control de una computadora, es decir que los gestos puedan ser utilizados como el cursor de esta.

Palabras Clave: **Gestos con las manos, kinect, baja iluminación, oclusión.**

Abstract of the thesis presented by América Ivone Mendoza Morales as a partial requirement to obtain the Master of Science degree in Master in Computer Science in Computer Science.

### **Computer control based in hand gestures in circumstances of low illumination**

Abstract approved by:

---

Dr. Vitaly Kober

The recognition of hand gestures has been prominent in different areas of computer science, eg. HCI is important because it helps create a natural interaction between the computer and the user, so ~~have developed various methods~~ to find the model that works in real time and in different circumstances. ~~So it is to create a model that merges the information provided by the Kinect device, then the recognition of static and dynamic gestures in real time under conditions of low light and when there is occlusion. This model be applied to create a system that serves as a control computer, is that gestures can be used as the cursor.~~

Keywords: **Hand gestures, kinect, low illumination, occlusion.**

## Dedicatoria

*A mis Padres y Abuela*

## Agradecimientos

A mis padres y hermanas por brindarme el apoyo necesario.

A mis compañeros y mis grandes amigos Darién Miranda y Oscar Peña por ayudarme siempre que lo necesitaba y también por distraerme cuando no lo pedía.

A Julia Diaz y a Daniel Miramontes por ayudarme y sacarme de dudas.

Al Dr. Vitaly Kober por permitirme trabajar con él.

Al Centro de Investigación Científica y de Educación Superior de Ensenada.

Al Consejo Nacional de Ciencia y Tecnología (CONACyT) por brindarme el apoyo económico para realizar mis estudios de maestría.

# Tabla de contenido

	Página
<b>Resumen en español</b>	ii
<b>Resumen en inglés</b>	iii
<b>Dedicatoria</b>	iv
<b>Agradecimientos</b>	v
<b>Lista de figuras</b>	viii
<b>Lista de tablas</b>	x
<b>1. Introducción</b>	1
1.1. Definición del problema . . . . .	2
1.2. Justificación . . . . .	2
1.3. Objetivo general . . . . .	2
1.4. Objetivos específicos . . . . .	2
1.5. Limitaciones y suposiciones . . . . .	3
1.6. Reconocimiento de gestos con la manos . . . . .	3
1.7. Estado del arte . . . . .	4
1.7.1. Modelos de contacto . . . . .	5
1.7.2. Modelos basados en la visión . . . . .	6
1.7.3. Sistemas comerciales . . . . .	8
1.8. Organización de la tesis . . . . .	11
<b>2. Marco teórico</b>	12
2.1. Gestos . . . . .	12
2.2. Reconocimiento de gestos con la manos . . . . .	12
2.2.1. Etapas del reconocimiento . . . . .	13
2.2.1.1. Adquisición de datos . . . . .	13
2.2.1.2. Detección . . . . .	13
2.2.1.3. Extracción de características y seguimiento . . . . .	15
2.2.1.4. Reconocimiento . . . . .	16
2.3. Imagen . . . . .	17
2.4. Oclusión . . . . .	18
<b>3. Sistema de reconocimiento de gestos propuesto</b>	19
3.1. Adquisición de los datos . . . . .	19
3.1.1. Kinect . . . . .	19
3.1.2. Filtro de mediana . . . . .	21
3.2. Detección . . . . .	21
3.2.1. Método detección rápida de objetos usando características simples utilizando el clasificador AdaBoost en forma de cascada .	22
3.2.1.1. Características Haar . . . . .	23
3.2.1.2. Imagen integral . . . . .	23
3.2.1.3. Algoritmo AdaBoost . . . . .	24
3.2.1.4. Clasificador AdaBoost en cascada . . . . .	25
3.2.2. Binarización . . . . .	27
3.2.3. Operaciones Morfológicas . . . . .	28

## Tabla de contenido (continuación)

3.2.3.1. Dilatación . . . . .	29
3.2.3.2. Erosión . . . . .	29
3.2.3.3. Apertura . . . . .	30
3.2.3.4. Cierre . . . . .	30
3.3. Extracción de características . . . . .	31
3.4. Reconocimiento . . . . .	35
<b>4. Implementación del sistema de reconocimiento de gestos propuesto</b>	<b>38</b>
4.1. Adquisición de los datos . . . . .	38
4.2. Detección . . . . .	40
4.3. Extracción de características . . . . .	42
4.4. Reconocimiento . . . . .	43
4.4.1. Reconocimiento de gestos estáticos . . . . .	44
4.4.2. Reconocimiento de gestos dinámicos . . . . .	44
<b>5. Resultados</b>	<b>46</b>
5.1. Experimentos de gestos estáticos . . . . .	46
5.1.1. Experimentos con iluminación . . . . .	47
5.1.2. Experimentos con iluminación media . . . . .	48
5.1.3. Experimentos sin iluminación . . . . .	50
5.2. Experimentos de gestos dinámicos . . . . .	52
5.2.1. Experimentos con iluminación media . . . . .	52
5.2.2. Experimentos sin iluminación . . . . .	55
<b>6. Conclusiones</b>	<b>58</b>
6.1. Limitaciones del sistema . . . . .	58
6.2. Aportaciones . . . . .	58
6.3. Trabajo futuro . . . . .	59
<b>Lista de referencias bibliográficas</b>	<b>60</b>

Figura	<b>Lista de figuras</b>	Página
1.	Dispositivos utilizados para la captura de gestos. . . . .	4
2.	Ejemplo del reconocimiento del gesto usando Leap Motion, mostrando mediante una aplicación donde los gestos son representados en 3D, que es el dispositivo que se encuentra conectado a la Laptop. Imagen recuperada de 9. . . . .	9
3.	Ejemplo del reconocimiento del gesto usando MYO, controlando el volumen de la computadora. El dispositivo es el aparece en el brazo del sujeto. Imagen recuperada de 10. . . . .	10
4.	La imagen anterior representa el funcionamiento del software Flutter. Imagen recuperada de 11 . . . . .	11
5.	El diagrama ejemplifica el procedimiento del reconocimiento de gestos. . . . .	13
<b>18figure.caption.13</b>		
7.	Metodología del sistema propuesto. . . . .	19
8.	Proceso de la etapa de adquisición de datos. . . . .	19
9.	Componentes del sensor Kinect, imagen recuperada de . . . . .	20
10.	Componentes del sensor Kinect, imagen recuperada de . . . . .	20
11.	Proceso de detección de la mano. . . . .	22
12.	Procedimiento del algoritmo de detección rápida de objetos. . . . .	23
13.	Ejemplo de tipos de operadores Haar. . . . .	23
14.	Ejemplo del cálculo de la imagen integral. . . . .	24
15.	Regiones de la imagen integral. . . . .	25
16.	Proceso del clasificador en forma de cascada, donde F representa la tasa de falsos positivos del clasificador de cascada y T . . . . .	27
17.	Ejemplos de elementos estructurales. . . . .	29
<b>30figure.caption.25</b>		
19.	Proceso de la extracción de características. . . . .	31
20.	Ejemplo de un conjunto conexo y un convexo. Image recuperada de 6 . . .	31
21.	En la imagen se aprecia de color rojo la envolvente convexa, de negro el contorno de la figura, y los puntos amarillos son el punto de profundidad de los defectos de convexidad. . . . .	32
<b>33figure.caption.29</b>		
<b>34figure.caption.30</b>		
<b>35figure.caption.31</b>		
25.	Configuración del sistema de reconocimiento de gestos . . . . .	38

## Listas de figuras (continuación)

Figura	Página
26. Representación de los datos capturados por los Kinect . . . . .	39
27. Representación de los datos capturados por los Kinect . . . . .	39
28. Ejemplo de imágenes de poses de nuestra base de datos. . . . .	41
29. Imágenes del fondo de nuestra base de datos. . . . .	41
30. Localización y selección de la mano, en la imagen de entrada del Kinect 2. . . . .	42
31. Binarización de ROI y aplicación de las operaciones morfológicas a la mano localizada en la figura anterior. . . . .	42
32. Resultado del cálculo de las características de la mano. Se muestra en color azul la punta de los dedos, en color verde la posición de la raíz de los dedos y en rojo el centro de la palma de la mano. . . . .	43
33. Ejemplo de imágenes de poses de nuestra base de datos. . . . .	44
34. Secuencia del gesto dinámico de la palma de la mano abierta, la vista es desde el Kinect frontal. . . . .	45
35. Secuencia del gesto dinámico del puño, la vista es desde el Kinect frontal. . . . .	45
36. Laboratorio en condiciones estándar de iluminación. . . . .	47
37. Laboratorio en condiciones con iluminación media. . . . .	49
38. Laboratorio en condiciones con baja iluminación. . . . .	50

## Lista de tablas

Tabla		Página
1.	Matriz de confusión del experimento uno, utilizando ambos Kinect.	47
2.	Matriz de confusión del experimento uno, utilizando solo el Kinect frontal.	47
3.	Matriz de confusión del experimento dos, con dos Kinect	48
4.	Matriz de confusión del experimento dos, utilizando solo el Kinect frontal.	48
5.	Matriz de confusión, los datos se capturaron usando dos Kinect.	48
6.	Matriz de confusión de los gestos 1 y 2. Utilizando un Kinect.	48
7.	Matriz de confusión	48
8.	Matriz de confusión. Se utilizó un Kinect para el análisis del reconocimiento	49
9.	Matriz de confusión	49
10.	Matriz de confusión, utilizando un Kinect	49
11.	Matriz de confusión	50
12.	Matriz de confusión, utilizando un Kinect	50
13.	Matriz de confusión	50
14.	Matriz de confusión del reconocimiento de los dos gestos estáticos.	51
15.	Matriz de confusión	51
16.	Matriz de confusión del reconocimiento de los dos gestos estáticos.	51
17.	Matriz de confusión	51
18.	Matriz de confusión	52

# Capítulo 1. Introducción

---

La interacción entre humanos se lleva a cabo gracias a la comunicación que existe entre ellos, esta puede ser oral o escrita y generalmente viene acompañada de gestos realizados con la cara, manos o otra cualquier parte del cuerpo. Estos gestos sirven como complemento de la comunicación pues ayudan a que el mensaje sea percibido de manera correcta.

El creciente desarrollo de la tecnología, a llevado a crear y estudiar distintas áreas de las ciencias computacionales, particularmente el área de interacción humano computadora (HCI, por sus siglas en inglés Human Computer Interaction), la área encargada del estudio y diseño de la forma en que el humano interactua con la computadora. Uno de los objetivos principales de esta área es que la interacción se lleve acabo de manera natural. No resulta extraño que los investigadores de HCI se hayan interesado en los gestos corporales, en especial los gestos realizados con las manos, para crear un ambiente natural entre el usuario y la computadora. Por lo que es necesario que la computadora pueda identificar la o las manos del usuario y reconocer el gesto que este realiza.

A finales de los años noventa se empezaron a desarrollar técnicas para el reconocimiento de gestos con las manos. Los primeros acercamientos utilizaban como medio de captura sensores como: guantes de datos, marcadores de colores y acelerómetros; los cuales se colocaban en la o las manos para poder capturar la posición e identificar la pose realizada. Las técnicas desarrolladas posteriormente obtienen la información necesaria para reconocer el gesto usando distintos tipos de imágenes o videos, que son obtenidos mediante diversos tipos de cámaras.

Los métodos que utilizan imágenes o video son los más utilizados para realizar el conocimiento de los gestos ya que la interacción entre el usuario y la computadora es más natural, el inconveniente con estos métodos es que es un problema difícil de resolver pues existen distintas variables que entran en juego para obtener una buena precisión en el reconocimiento.

Aunque existe gran variedad de métodos y sistemas que hacen el reconocimiento de gestos de las manos no existe alguno que el reconocimiento tenga un alto grado de precisión en todas las situaciones que se presentan en el mundo real.

Es por eso que se propone crear un sistema que reconozca gestos realizados con las manos, en situaciones que presentan baja iluminación y cuando existe oclusión de los dedos. El sistema se enfoca en atacar estos problemas cuando las manos no se encuentran en movimiento, pero también se abordarán los gestos con las manos que involucran movimiento. El objetivo del sistema es mostrar que se obtiene mayor precisión en el reconocimiento de los gestos utilizando como medio de captura dos sensores Kinect.

### **1.1. Definición del problema**

Existen diversas técnicas que logran obtener buena precisión en el reconocimiento de gestos realizados con las manos, pero no hay técnicas que tengan buena precisión y que al mismo tiempo se adecuen a todo tipo de situaciones que se presentan en la vida real como: amigable con el usuario, invariante a la iluminación, rotación, al fondo, que funcione en tiempo real o cuando exista oclusión.

### **1.2. Justificación**

Debido a la complejidad del problema de reconocimiento de gestos con las manos, las técnicas desarrolladas y actuales se enfocan en aspectos específicos para obtener un buen grado de precisión. De manera que se necesitan nuevos métodos que aborden los aspectos dejados de lado y funcionen no solo en condiciones ideales si no en situaciones que se presentan de manera natural y al mismo tiempo se obtenga un alto grado de precisión.

Una vez logrado lo anterior se pueden desarrollar nuevas aplicaciones y tecnologías que ayuden a interactuar con naturalidad al usuario y la computadora.

### **1.3. Objetivo general**

Desarrollar un sistema que permita controlar la computadora haciendo uso de gestos con las manos, estáticos y dinámicos. El sistema debe ser robusto, funcionar en circunstancias de baja iluminación, cuando exista oclusión en gestos dinámicos.

### **1.4. Objetivos específicos**

- Identificar los métodos actuales de reconocimiento de gestos, estáticos y dinámicos cuando existe baja iluminación y cuando existe oclusión.

- Obtener conocimiento acerca del funcionamiento de sistema Microsoft Kinect.
- Desarrollar un sistema de reconocimiento de gestos estáticos y dinámicos, fusionando la información de los sensores de profundidad de dos dispositivos kinect. El sistema desarrollado deberá funcionar en circunstancias de baja iluminación y también cuando existe oclusión, causada por los dedos.
- Analizar el sistema diseñado, en cuanto a su eficiencia presentada en base al reconocimiento de los gestos, en circunstancias de baja iluminación y oclusión. En el análisis del sistema se usará información real.
- Comparar y analizar el modelo propuesto haciendo uso de uno y dos dispositivos Kinect.

### **1.5. Limitaciones y suposiciones**

Gran porcentaje de los trabajos previos en el área de reconocimiento de gestos con las manos basados en el modelo de la visión utilizan cámaras digitales o cámaras web. Esta investigación utiliza dos dispositivos Kinect, para obtener la información de entrada del sistema.

De manera que las limitaciones del sistema propuesto están dadas por las características de dicho dispositivo, tales como la distancia a la que se encuentran los dispositivos con el usuario y la resolución del sensor.

Otra limitante es el número de gestos que podrá reconocer el sistema.

### **1.6. Reconocimiento de gestos con la manos**

La definición de gestos (Mitra *et al.*, 2007) son movimientos del cuerpo expresivos y significativos que involucran a los dedos, manos, brazos, cabeza, cara o cuerpo con la intención de transmitir información relevante o de interactuar con el ambiente.

Los primeros acercamientos para llevar a cabo el reconocimiento de gestos con las manos fue usando modelos de contacto (Rautaray y Agrawal, 2012) y (Nayakwadi, 2014), como su nombre lo dice utilizan dispositivos que están en contacto físico con la mano del usuario 1(a), para capturar el gesto a reconocer, por ejemplo existen guantes de

datos, marcadores de colores, acelerómetros y pantallas multi-touch, aunque estos no son tan aceptados pues entorpecen la naturalidad entre la interacción del humano y la computadora. Los modelos basados en la visión 1(b) , surgieron como respuesta a esta desventaja, estos utilizan cámaras para extraer la información necesaria para realizar el reconocimiento, los dispositivos van desde cámaras web hasta algunas más sofisticadas por ejemplo cámaras de profundidad.



(a) Dispositivos basados en contacto: en la parte izquierda de la imagen se observan los guantes de datos <sup>1</sup> , en el centro los guantes de colores <sup>2</sup> y a la derecha se encuentra el dispositivo wii <sup>3</sup> .



(b) Dispositivos basados en visión: en la imagen se observan distintos tipos de cámaras en la parte izquierda de la imagen se observa una web <sup>4</sup> , en el centro una digital <sup>5</sup> y a la derecha de la imagen una TOF <sup>6</sup> .

**Figura 1: Dispositivos utilizados para la captura de gestos.**

En este trabajo, se toma el enfoque basado en la visión ya que se quiere obtener un sistema que para el usuario la interacción sea natural y la manera de lograrlo es tomando este enfoque.

### 1.7. Estado del arte

La sección anterior explica los modelos utilizados para llevar acabo el reconocimiento de gestos con las manos, enseguida se presentan los trabajos relevantes de cada uno de

<sup>1</sup><http://www.technologyreview.com/article/414021/open-source-data-glove/>

<sup>2</sup><http://www.digitaltrends.com/computing/the-gloves-that-could-change-the-world/>

<sup>3</sup><https://www.nintendo.es/Wii/Wii-94559.html>

<sup>4</sup><http://es.ccm.net/download/descargar-2562-driver-de-microsoft-lifecam-vx-3000>

<sup>5</sup><http://www.canon.com.mx/ficha.aspx?id=722>

<sup>6</sup><http://us.creative.com/p/web-cameras/creative-senz3d>

estos enfoques y también se mencionan algunos de los sistemas comerciales importantes.

### 1.7.1. Modelos de contacto

Como se menciono en la sección anterior los primeros trabajos de reconocimiento de gestos con las manos utilizaba este modelo, actualmente se sigue utilizando pero en menor grado. En los párrafos siguientes se presentan dos trabajos relevantes en esta área.

El primer trabajo que se presenta es el realizado por (Yoon *et al.*, 2012) el cual propone un sistema de reconocimiento de gestos estáticos usando un guante de datos, el cual reconoce 24 gestos tomados del Lenguaje de Señas Americano, ASL (por sus siglas en inglés, American Sign Lenguaje). Este modelo consta de tres etapas, las cuales se explican enseguida.

La primera etapa del sistema consiste en capturar la información proporcionada por un guante de datos, la cual esta siendo enviada por un protocolo de control de transmisión TCP, (por sus siglas en inglés, Transmission Control Protocol).

Una vez que la información es recibida, los datos son pre-procesados, es decir son normalizados y las características son extraídas, las características son las correlaciones que existe entre los ejes.

La clasificación de gesto se realiza con un modelo de mezclas adaptativo. Para entrenar el modelo de mezclas se toman datos de 5 personas, 300 muestras de cada gestos, 8000 por cada participante. Se realizaron pruebas con estos mismos datos; con un sujeto se alcanzó una precisión de 93.38 % con los demás participantes se obtuvo una precisión de 89.97 %.

La principal desventaja del sistema es que baja la precisión cuando se cambia de usuario, aunque después se adapta y mejora la precisión, otra desventaja para este sistema es que solo reconoce gestos estáticos.

A finales del año 2014 se lanzó el dispositivo MYO<sup>8</sup>, el cual reconoce gestos dinámi-

---

<sup>8</sup><https://www.thalmic.com/en/myo/>

cos, los detalles del dispositivo se encuentran en ultima parte de esta sección.

### **1.7.2. Modelos basados en la visión**

Este modelo es el más popular debido a la variedad de sus aplicaciones y la diversidad de cámaras existentes que proporcionan distinto tipo de información la cual puede hacer que el reconocimiento tenga mayor precisión. Enseguida se presenta tres trabajos relevantes los cuales utilizaron distintos tipos de cámaras y número de ellas.

En trabajo de (Huang *et al.*, 2011), propone un método que reconoce 11 gestos estáticos y dinámicos, la aportación del trabajo es la segmentación de la mano que se lleva acabo usando filtros de Gabor. El sistema propuesto utiliza una cámara CCD para obtener la información de entrada. El sistema es robusto a la iluminación.

Antes de hacer la segmentación de la mano se le aplica a la imagen un preprocesamiento que consiste en aplicar un filtro de Gabor, después se escoge uno de los tres modelos del color; YCbCr, Gaussiano o Soriano, tomando en cuenta un nivel de gris. Una vez que es realizado el preprocesamiento el paso siguiente es segmentar la mano del antebrazo para esto se hace un barrido de la imagen por filas. Se segmenta la mano tomando en cuenta la distancia que existe entre la parte superior de la imagen y el número máximo de pixeles de un solo valor (el valor mayor del histograma).

Una vez realizada la segmentación el siguiente paso es obtener las características necesarias para el reconocimiento, las características son obtenidas utilizando análisis de componentes principales, PCA (por sus siglas en inglés, Principal Component Analysis). La clasificación la hacen usando maquinas de vectores de soporte, SVM (por sus siglas en inglés, Support Vector Machines).

La precisión del reconocimiento varía dependiendo de las imágenes, si son reales o si se les aplica antes un filtro de Gabor, también cambia si el usuario usa manga corta o larga. Las principales ventajas son que el sistema funciona con cambios en la iluminación y es robusto a la rotación y escala. La desventaja es que el problema de oclusión no es tratado.

Por otra parte en el trabajo propuesto por (Caputo *et al.*, 2012) gestos dinámicos y estáticos son reconocidos, estos últimos son utilizados para determinar el inicio y el

término de los gestos dinámicos. Se utilizan dos sensores Kinect y una cámara web Logitech C910 de alta definición para capturar los gestos. El trabajo esta compuesto de cuatro etapas, las cuales se explican enseguida.

La primera es la configuración de los dispositivos de captura de datos del sistema. Los dos sensores Kinect son calibrados entre ellos para generar un sistema de coordenadas que esta basado en la ubicación de la manos y la cabeza. La cámara y los dispositivos Kinect no son sincronizados entre si.

La parte de la detección y seguimiento, se lleva acabo utilizando la librería OPENNI, en específico usando la detección del esqueleto proporcionado por esta librería. El esqueleto nos proporciona el punto de la palma de la mano por la cual la región de interés, ROI (por sus siglas en inglés, Region of Interest) es seleccionada, para tener la localización exacta de la mano, se utiliza la cámara RGB. La localización de la mano se realiza convirtiendo la imagen en una imagen binaria, usando un umbral que es determinado por el espacio del color HSV (Matiz, Saturación, Valor); son utilizados guantes neón color rosa o verde para ubicar con mayor facilidad las manos.

Una vez obtenida la imagen binaria se calcula el contorno de la mano usando el algoritmo de Chang y Chen, dicho contorno es extraído como polígono y es simplificado con el algoritmo de Douglas Peuker.

El reconocimiento del gesto se basa en empalamiento de polígonos, basados en la distancia de dos polígonos. Esto usando Distancia de momentos HU (Hu-moments distance) y ángulo de giro (turning angle). Los gestos 3D son calculados usando la diferencia de las posiciones de la mano, en cada cuadro. Las fórmulas para calcular estos gestos depende de que gesto se realice. Para probar la precisión del sistema se crearon dos bases de datos, una con 120 polígonos etiquetados que representan 11 gestos y otra con 144 gestos de 3 personas distintas realizando los 11 gestos. La precisión obtenida usando la distancia de ángulo de giro es de 85 %, usando la distancia de momentos HU la precisión es de 58 %.

Otra aportación importante fue hecha por (Kang *et al.*, 2013) ellos proponen un sistema de reconocimiento de gestos estáticos utilizando el sensor Kinect como dispositivo de captura de los gestos. El sistema reconoce 24 gestos, los cuales pertenecen al ASL, el

reconocimiento es realizado en cuatro etapas, las cuales se explican a continuación.

En la primera etapa la imagen es capturada y la mano junto con el antebrazo son segmentados del fondo. Las imágenes de entrada del sistema son proporcionadas por el sensor de profundidad del Kinect, la mano es detectada usando el SDK (Software Developmet Kit) del Kinect, que proporciona el punto de la palma de la mano, la región de interés es seleccionada usando este punto, donde solo se encuentra la mano y parte del antebrazo.

El siguiente paso es extraer las características, las cuales son extraídas usando Histogramas Orientados a Gradientes, HOG (Histogram of Oriented Gredient).

El paso siguiente es clasificar los gestos, se utiliza el algoritmo de aprendizaje de máquina, máquinas de soporte vectorial.

Para el entrenamiento del se utilizaron 2400 imágenes, 100 por cada letra del alfabeto. Se encontró que existe gesto ambiguos, es decir que no se pueden clasificar correctamente, estos son los gestos que representan la letras A, E, M, N, S, T. Se realizo una prueba en linea, donde los gestos aparecían aleatoriamente para ser clasificados. La precisión de todos los gestos se encuentra alrededor de 92.8 %, pero el de los gestos ambiguos es 72.9 %

Por ultimo una interfaz gráfica es mostrada donde se aprecia el reconocimiento de los gestos en tiempo real.

### **1.7.3. Sistemas comerciales**

Existen dispositivos como: Leap Motion <sup>9</sup>, MYO <sup>10</sup>, y software como: Flutter <sup>11</sup>, que realizan el reconocimiento de gestos, y este reconocimiento es aplicado para controlar la computadora. Algunos de estos dispositivos comerciales tienen buen rendimiento en cuanto a la precisión y a sobrellevar los problemas del reconocimiento de gestos, el inconveniente es que los desarrolladores de los dispositivos o software no dan a conocer los detalles de como solucionan algunos de los problemas o como mejoran la precisión. Enseguida se describen los sistemas mencionados anteriormente.

---

<sup>9</sup> <https://www.leapmotion.com/>

<sup>10</sup> <https://www.myo.com/>

<sup>11</sup> <https://flutterapp.com/>

El dispositivo Leap Motion, fig 2 fue creado para el seguimiento de manos y dedos, este también hace el reconocimiento de ciertos gestos estáticos y dinámicos. El dispositivo consta de tres emisores y dos cámaras infrarrojas, estos sensores capturan los datos crudos en un rango de  $60 \times 60 \times 60\text{ cm}$ . y con la información capturada se construye un modelo 3D de las manos (Weichert *et al.*, 2013).



**Figura 2: Ejemplo del reconocimiento del gesto usando Leap Motion, mostrando mediante una aplicación donde los gestos son representados en 3D, que es el dispositivo que se encuentra conectado a la Laptop. Imagen recuperada de 9.**

El proceso de como se capturan los datos, la segmentación, la extracción de características, el seguimiento y el reconocimiento del dispositivo no se conoce a detalle.

Solo se conoce <sup>12</sup> que se utilizan tres cámaras infrarrojas, con la imágenes obtenidas se hace una representación 3D de las manos, antes de realizar el modelo las imágenes son segmentadas del fondo para eliminar el ruido generado por la iluminación u otros objetos que causen ruido en la imágenes.

Para realizar el seguimiento se extraen la características, una de ellas son el dedos, el algoritmo de seguimiento interpreta la información 3D e infiere la posición de los objetos ocluidos. Se aplican filtros para suavizar los datos.

Un dispositivo de reconocimiento de gestos basado en el modelo de contacto, es el MYO, este aparato es un brazalete que reconoce 5 gestos dinámicos. Leyendo la actividad de los músculos del antebrazo y mandando estas señales vía Bluetooth a la computadora donde estas señales son procesadas. <sup>13</sup>

No se cuenta con la información detallada del funcionamiento de MYO, lo único que se conoce es que el reconocimiento consta de tres etapas <sup>14</sup>. La primera es la adquisición de

<sup>12</sup><http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>

<sup>13</sup><http://www.digitaltrends.com/pc-accessory-reviews/myo-gesture-control-armband-review/>

<sup>14</sup><https://www.quora.com/How-does-MYO-wearable-gesture-control-work>

la señales eléctricas que producen los músculos del antebrazo, las cuales son capturadas mediante sensores EMG (estos detectan la actividad eléctrica), giroscopio, acelerómetro y magnetómetro; en la segunda etapa se amplifica la señal y se aplica un filtro pasa banda. Por último se realiza el procesamiento de la señal donde se reconoce el gesto usando un algoritmo de aprendizaje de máquina desarrollado por la compañía.



**Figura 3: Ejemplo del reconocimiento del gesto usando MYO, controlando el volumen de la computadora. El dispositivo es el aparece en el brazo del sujeto. Imagen recuperada de 10.**

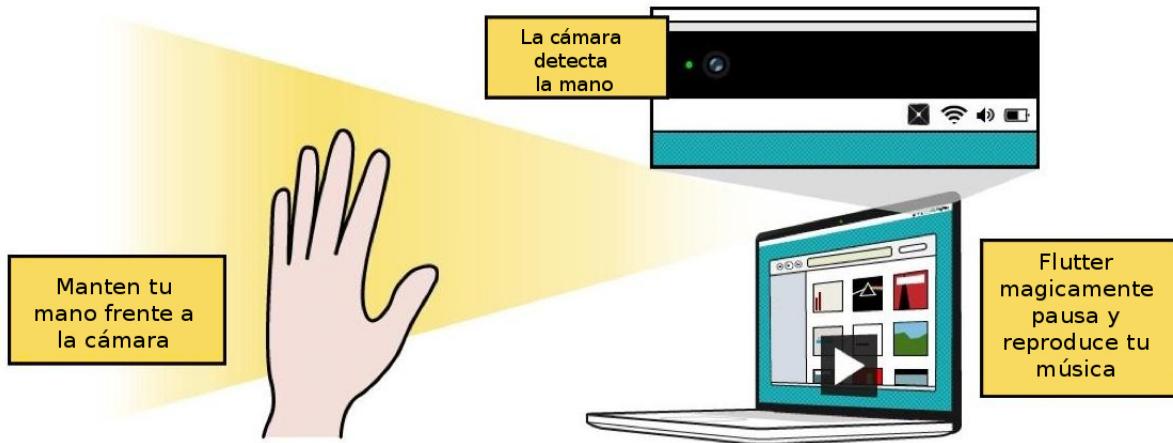
MYO funciona en cualquier ambiente donde haya variaciones en la iluminación y es invariante a rotación. La desventaja que tiene la calibración pues esta puede ser tediosa ya que requiere realizar repeticiones de algunos gestos y esta requiere de varios intentos; el uso del dispositivo requiere uso de manga corta; otra desventaja es que tiene una cantidad considerable de falsos positivos.<sup>15</sup>

Enseguida se explica el software de reconocimiento de gestos estáticos Flutter, fig:4 el cual reconoce cuatro gestos estáticos usando la cámara web como dispositivo de entrada.

Se conoce muy superficialmente como funciona el software, pues solo se sabe que la mano es detectada por la cámara, para que la detección sea correcta la mano tiene que estar totalmente frente a la cámara web. Los algoritmos utilizados para el reconocimiento no se conocen.

---

<sup>15</sup>[http://myogroupfive.blogspot.mx/2013/11/benefits-disadvantages-for-business\\_24.html](http://myogroupfive.blogspot.mx/2013/11/benefits-disadvantages-for-business_24.html)



**Figura 4:** La imagen anterior representa el funcionamiento del software Flutter. Imagen recuperada de 11 .

Flutter permite controlar aplicaciones multimedia como: YouTube<sup>16</sup>, VLC<sup>17</sup>, Spotify<sup>18</sup>, Netflix<sup>19</sup>. Las limitaciones del software son que solo reconoce gestos estáticos, realiza acciones no deseadas al hacer gestos involuntarios y no siempre reconoce los gestos.

Aunque estos dispositivos y software para reconocer gestos solucionan algunos problemas importantes en el área, sigue existiendo el problema de occlusiones e iluminación. De allí la importancia que existan nuevos modelos que ataquen estos problemas que se presentan frecuentemente en el reconocimiento de los gestos.

## 1.8. Organización de la tesis

La tesis se encuentra distribuida de la siguiente manera: la segunda sección presenta los fundamentos teóricos como base para la comprensión del tema. La tercera sección presenta la metodología utilizada en el sistema propuesto. En la cuarta sección se encuentran los detalles de la implementación del sistema. En la quinta sección están las pruebas realizadas al sistema junto con los resultados y las discusiones de estos. Finalmente la sexta sección presenta las conclusiones generales del sistema y el trabajo futuro.

<sup>16</sup><https://www.youtube.com/>

<sup>17</sup><http://www.videolan.org/vlc/>

<sup>18</sup><https://www.spotify.com/>

<sup>19</sup><https://www.netflix.com/>

## Capítulo 2. Marco teórico

---

En este capítulo se definen una serie de conceptos importantes del área de procesamiento de imágenes y reconocimiento de patrones, estas definiciones son importantes para la comprensión del tema.

### 2.1. Gestos

Los gestos (Mitra *et al.*, 2007) son movimientos del cuerpo expresivos y significativos que involucran dedos, manos, brazos, cabeza, cara o cuerpo con la intención de transmitir información relevante o interactuar con el ambiente.

De acuerdo con la literatura (Mitra *et al.*, 2007) los gestos con las manos se clasifican en estáticos y dinámicos, los primeros están definidos como la posición y orientación de la mano en el espacio manteniendo esta pose durante cierto tiempo, por ejemplo para hacer una señal de aventón, a diferencia de los gestos dinámicos donde hay movimiento de la pose, un ejemplo es cuando mueves la mano en señal de adiós.

### 2.2. Reconocimiento de gestos con la manos

El reconocimiento de gestos con las manos consiste no solo en el seguimiento del movimiento de la o las manos realizados por un emisor, también en la interpretación de este movimiento por un receptor, (Mitra *et al.*, 2007), (Murthy y Jadon, 2009).

De aquí en adelante entiéndase el término gestos con las manos, como gestos.

En el capítulo 1 sección 1.6, se explicó que existen dos modelos utilizados para el reconocimiento de gestos dependiendo del dispositivo de captura, los basados en contacto y en la visión.

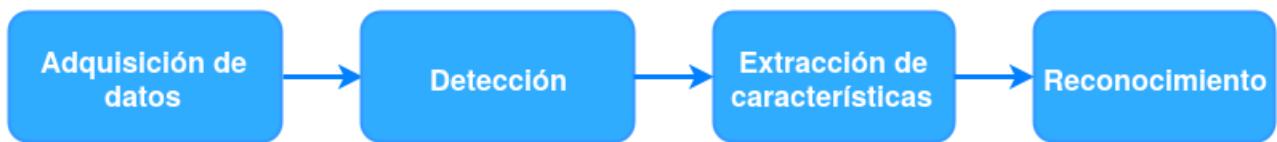
Los métodos basados en la visión realizan la representación del gesto con diferentes técnicas las cuales se separan en dos categorías (Rautaray y Agrawal, 2012): basados en apariencia y basados en un modelo 3D. Los basados en un modelo 3D convierten los datos en entrada en una forma espacial y los basados en apariencia utilizan los datos 2D de la imagen de entrada.

De acuerdo con la literatura el proceso de reconocimiento de gestos basados en la visión se dividen en tres (Rautaray y Agrawal, 2012) fases que son: detección; extracción

de características o seguimiento, dependiendo si los gestos son dinámicos; por último el reconocimiento del gesto. Otros autores (Hasan y Mishra, 2012) incluyen la etapa de adquisición de datos. Las etapas se abordaran en la sección siguiente.

### **2.2.1. Etapas del reconocimiento**

En la sección se abordaron las estas etapas del reconocimiento y se mencionan los principales algoritmos utilizados en cada una de estas etapas. El diagrama de la figura 5 muestra las etapas del reconocimiento.



**Figura 5:** El diagrama ejemplifica el procedimiento del reconocimiento de gestos.

El proceso de reconocimiento varia un poco dependiendo del tipo de gesto, si es estático o dinámico. Por ejemplo el diagrama 5 ejemplifica perfectamente el proceso de reconocimiento de un gesto estático, para los gestos dinámicos se necesita una fase extra, el seguimiento el cual se realiza una vez detectada la mano, esta puede estar englobada en la fase de extracción de características o viceversa.

#### **2.2.1.1. Adquisición de datos**

Es la primera etapa del reconocimiento en la cual los datos son capturados. En el modelo basado en la visión se utilizan cámaras como:

La información obtenida es representada como imágenes.

#### **2.2.1.2. Detección**

En esta etapa se localiza y segmenta la mano del fondo de la imagen para obtener las características necesarias para identificar el gesto.

Existen distintos métodos para obtener dichas características como la de color de la piel, forma, movimiento, entre otras que generalmente son combinaciones de alguna de estas, para obtener un mejor resultado. Enseguida se describe brevemente cada una de estas.

- Color de la piel: Se basa principalmente en escoger un espacio del color, es una organización de colores específica; como; RGB (rojo, verde, azul), RG (rojo, green), YCrCb (brillo, la diferencia entre el brillo y el rojo, la diferencia entre el brillo y el azul), etc. La desventaja es que si es color de la piel es similar al fondo, la segmentación no es buena, la forma de corregir esta segmentación es suponiendo que el fondo no se mueve con respecto a la cámara.
- Forma: Extrae el contorno de las imágenes, si se realiza correctamente se obtiene el contorno de la mano. Aunque si se toman las yemas de los dedos como características, estas pueden ser ocluidas por el resto de la mano, una posible solución es usar más de una cámara.
- Valor de píxeles: Usar imágenes en tonos de gris para detectar la mano en base a la apariencia y textura, esto se logra entrenando un clasificador con un conjunto de imágenes.
- Modelo 3D: Depende de cual modelo se utilice, son las características de la mano requeridas.
- Movimiento: Generalmente esta se usa con otras formas de detección ya que para utilizarse por sí sola hay que asumir que el único objeto con movimiento es la mano.

La segmentación es la partición o separación de la imagen en regiones representativas, es decir separar la mano del fondo de la imagen. Existen diversos métodos para llevar acabo la segmentación de la mano los cuales se clasifican en cuatro clases, basados en píxeles los cuales hacen la separación usando el valor del nivel de gris en la imagen; en el borde estos métodos utilizan los píxeles que representan las orillas del objeto y encuentran el correspondiente contorno; en regiones los cuales van agrupando vecindarios de la imagen de acuerdo a ciertas propiedades; por ultimo la segmentación basada en un modelo la cual hacen uso de algún modelo definido, estos requieren imágenes de entrenamiento para representar la probabilidad de las muestras registradas y finalmente hace inferencias en la imagen (Ibraheem, 2013). Enseguida se presentan algunas técnicas para realizar la segmentación.

- Color de la piel: Se basa principalmente en escoger un espacio del color, es una

organización de colores especifica; como; RGB (rojo, verde, azul), RG (rojo, green), YCrCb (brillo, la diferencia entre el brillo y el rojo, la diferencia entre el brillo y el azul), etc. La desventaja es que si el color de la piel es similar al fondo, la segmentación no es buena, la forma de corregir esta segmentación es suponiendo que el fondo no se mueve con respecto a la cámara.

### **2.2.1.3. Extracción de características y seguimiento**

La extracción de características consiste en obtener ciertas entradas medibles de la imagen de la mano, generalmente segmentada, las cuales son utilizadas para reconocer el gesto realizado, (Premaratne, 2013), (Nayakwadi, 2014).

Existen dos tipos de características geométricas tales como: las yemas de los dedos, la dirección de los dedos, el contorno de la mano y entre otras características; y las características no geométricas como el color, siluetas y texturas. (Murthy y Jadon, 2009).

Enseguida se mencionan algunas de los métodos para la obtención de características (Premaratne, 2013).

- Descriptores de Fourier los cuales describen formas en la imagen, haciendo uso de la serie de Fourier. Estas características son invariantes a escala.
- Descriptores de Contorno nos dan el contorno o el límite del objeto con invariancia a traslación, escala y reflexión.
- Características descritas por histogramas. Histogramas de gradientes orientados (HOG) es un descriptor de características. Se trata de contar la orientación de los gradientes en cierta porción de la imagen.

Consiste en localizar la mano en cada cuadro (imagen). Se lleva a cabo usando los métodos de detección si estos son lo suficientemente rápidos para detectar la mano cuadro por cuadro. Se explica brevemente los métodos para llevar a cabo el seguimiento.

- Basado en plantillas: Este se divide en dos categorías (Características basadas en su correlación y basadas en contorno), que son similares a los métodos de detección, aunque supone que las imágenes son adquiridas con la frecuencia suficiente

para llevar acabo el seguimiento. Características basadas en su correlación, sigue las características a través de cada cuadro, se asume que las características aparecen en mismo vecindario. Basadas en contorno, se basa en contornos deformables, consiste en colocar el contorno cerca de la región de interés e ir deformando este hasta encontrar la mano.

- Estimación óptima: Consiste en usar filtros Kalman, un conjunto de ecuaciones matemáticas que proporciona una forma computacionalmente eficiente y recursiva de estimar el estado de un proceso, de una manera que minimiza la media de un error cuadrático, el filtro soporta estimaciones del pasado, presente y futuros estados, y puede hacerlo incluso cuando la naturaleza precisa del modelo del sistema es desconocida; para hacer la detección de características en la trayectoria.
- Filtrado de partículas: Un método de estimación del estado de un sistema que cambia a lo largo del tiempo, este se compone de un conjunto de partículas (muestras) con pesos asignados, las partículas son estados posibles del proceso. Es utilizado cuando no se distingue bien la mano en la imagen. Por medio de partículas localiza la mano la desventaja es que se requieren demasiadas partículas, y el seguimiento se vuelve imposible.
- Camshift: Busca el objetivo, en este caso la mano, encuentra el patrón de distribución mas similar en una secuencia de imágenes, la distribución puede basada en el color.

#### 2.2.1.4. Reconocimiento

Es reconocimiento es la etapa final de este proceso, la cual consiste en identificar el gesto utilizando alguna técnica de clasificación.

El método de clasificación a utilizar se elige dependiendo del tipo de gesto a reconocer, por ejemplo para los gestos estáticos se realiza el empatamiento del gesto con una plantilla previamente calculada; en los gestos dinámicos generalmente se usan algoritmos de aprendizaje de máquina. Aunque los más utilizados son los algoritmos de redes neuronales, maquina de soporte vectorial y modelo oculto de Markov 

A continuación se encuentran los principales métodos para llevar acabo el reconocimiento del gestos (Rautaray y Agrawal, 2012).

- K-medias: Es un método de agrupamiento el cual consiste en determinar los  $k$  puntos llamados centros para minimizar el error de agrupamiento, que es la suma de las distancias de todo los puntos al centro de cada grupo. El algoritmo empieza localizando aleatoriamente  $k$  grupos en el espacio espectral. Cada píxel en la imagen de entrada es entonces asignadas al centro del grupo mas cercano
- Desplazamiento de medias: Es un método iterativo que encuentra el máximo en una función de densidad dada una muestra estadística de los datos.
- Máquinas de soporte vectorial: Consiste en un mapeo no lineal de los datos de entrada a un espacio de dimensión más grande, donde los datos pueden ser separados de forma lineal.
- Modelo oculto de Markov: Es definido como un conjunto de estados donde, un estado inicial, un conjunto de símbolos de salida y un conjunto de estados de transición. En el reconocimiento de gestos se puede caracterizar a los estados como un conjunto de las posiciones de la mano; las transiciones de los estados como la probabilidad de transición de cierta posición de la mano a otra; el símbolo de salida como una postura específica y la secuencia de los símbolos de salida como el gesto de la mano.
- Redes neuronales con retraso: Son una clase de redes neuronales artificiales que se enfocan en datos continuos, haciendo que el sistema sea adaptable para redes en linea y les da ventajas sobre aplicaciones en tiempo real.

### 2.3. Imagen

Una imagen se puede definir como una función bidimensional,  $S(x, y)$  donde  $x, y$  representan las coordenadas en el plano y el valor de la función es la intensidad o nivel de gris en el punto  $(x, y)$ . Si el valor de la función y los puntos de la imagen son finitos, esta es una imagen digital, la cual se puede representar en una matriz donde cada valor o pixel es el nivel de gris de la imagen 6, y los indices de esta indican la posición, (Gonzalez y Woods, 2002).



Figura 6: Representación de un imagen digital. Recuperada de (Shin, 2013)

## 2.4. Oclusión

Se puede definir una oclusión como discontinuidades del movimiento y profundidad que se es percibida por un observador que se encuentra en movimiento en un ambiente estático.

Los puntos de oclusión en una imagen o cuadro son pixeles que aparecen o desaparecen en dos cuadros consecutivos, estos son llamados puntos de oclusión o punto de no oclusión, (Silva y Santos-Victor, 2001).

Existen tres tipos distintos de oclusiones la cuales depende de la forma en que es causada. Estas son: oclusión por el mismo objeto, entre objetos y por el fondo. La oclusión por el mismo objeto se presenta cuando parte del objeto ocluye a otra. La oclusión entre objetos es cuando dos objetos que se siguen se ocluyen entre ellos mismos. La oclusión por el fondo es cuando parte del fondo ocluye al objeto que se sigue, (Yilmaz *et al.*, 2006)

## Capítulo 3. Sistema de reconocimiento de gestos propuesto

---

En este capítulo se describen las etapas del sistema propuesto junto con los métodos o algoritmos que son utilizados en cada una de las fases.

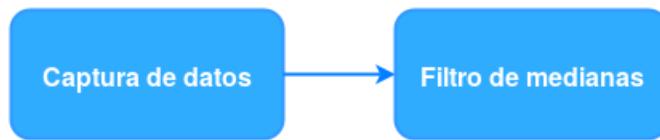
El sistema de reconocimiento de gestos propuesto consta de cuatro etapas principales. La primera etapa es la adquisición de los datos, en la cual se capturan las imágenes de entrada del sistema; es la detección, aquí la mano es localizada y segmentada del fondo; en la etapa tres se extraen las características de la mano para ser procesadas; en la etapa final el gesto realizado es reconocido.



**Figura 7: Metodología del sistema propuesto.**

### 3.1. Adquisición de los datos

Es la primera etapa del sistema, donde los datos de entrada del sistema son capturados y preprocesados para eliminar ruido existente en la imagen.



**Figura 8: Proceso de la etapa de adquisición de datos.**

En este trabajo la adquisición de los datos se divide en dos partes, ya que los datos provienen de los sensores de profundidad de dos dispositivos Kinect, debido a la naturaleza del sensor las imágenes capturadas tienen ruido tipo de [poner referenciar], este puede ser reducido utilizando filtro de medianas.

#### 3.1.1. Kinect

En noviembre del 2010 la compañía Microsoft lanzó el sensor Kinect para consolas de video juego Xbox 360 y en febrero del 2011 lanzó la versión para Windows, que se

muestra en la figura 9.

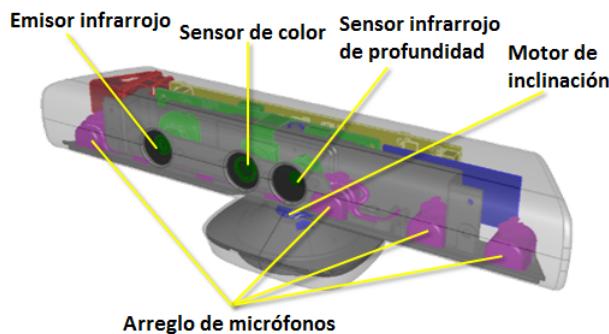
El dispositivo Kinect esta equipado con una serie de sensores que permiten obtener imágenes a color y de profundidad (las cuales indican la distancia a la que esta ubicada un objeto del sensor). Los sensores permiten hacer detección y seguimiento de personas. El dispositivo tiene la capacidad de detectar 6 personas y hacer el seguimiento de 2 personas<sup>1</sup>.



**Figura 9: Componentes del sensor Kinect, imagen recuperada de <sup>2</sup>.**

El sensor esta equipado con los siguientes componentes: un cámara de color o sensor de color, un emisor infrarrojo, un sensor infrarrojo de profundidad, un motor que controla la inclinación, un arreglo de cuatro micrófonos y un LED 10.

Enseguida se describen brevemente cada uno de los componentes del sensor Kinect, (Jana, 2013).



**Figura 10: Componentes del sensor Kinect, imagen recuperada de <sup>3</sup>.**

- La cámara de color captura y transmite datos de vídeo a color, detectando los colores rojo, verde y azul (RGB, por sus siglas en inglés, red, green and blue). La

<sup>1</sup><https://msdn.microsoft.com/en-us/library/hh973074.aspx>

<sup>2</sup><http://blogs.msdn.com/b/eternalcoding/archive/2012/02/01/official-kinect-for-windows-sdk-and-kinect-toolbox-1-1-1-are-out.aspx>

<sup>3</sup><https://msdn.microsoft.com/en-us/library/jj131033.aspx>

transmisión de datos que brinda la cámara es una secuencia de imágenes (cuadros), a una velocidad de hasta 30 cuadros por segundo con una resolución de hasta  $1280 \times 960$  píxeles. La velocidad de los cuadros por segundo varía según la resolución de la imagen.

- El emisor infrarrojo proyecta puntos de luz infrarroja, estos puntos son proyectados frente al sensor. Estos puntos junto con el sensor de profundidad es posible medir la distancia que existe del sensor a algún objeto que este frente a él.
- El sensor infrarrojo lee los puntos infrarrojos proyectados por el emisor infrarrojo y calcula la distancia que existe entre el objeto y el sensor. El sensor transmite los datos de profundidad con una velocidad de 30 cuadros por segundo con una resolución de hasta  $640 \times 480$  pixeles.
- El motor de inclinación controla el ángulo de la posición vertical de los sensores del dispositivo. El motor puede moverse desde el ángulo de  $-27^\circ$  a  $+27^\circ$ .
- El arreglo de micrófonos, consta de 4 micrófonos. También se captura el sonido y localiza la dirección de la cual proviene.
- LED este indica el estado del sensor.

### 3.1.2. Filtro de mediana

Existen distintos métodos para eliminar el ruido proveniente del Kinect, algunos de estos métodos son invasivos pues el hardware del Kinect es modificado. Una opción es utilizar filtro de mediana, pues rellena los valores faltantes en la imagen proveniente del Kinect,

El filtro de mediana reemplaza el valor del pixel usando la mediana de las intensidades del vecindario del pixel, el valor del pixel en la posición  $(x,y)$  es incluido en el cálculo.

## 3.2. Detección

En esta etapa del sistema el objetivo es localizar y segmentar la mano para extraer las características necesarias para el reconocimiento.

Este procedimiento se lleva acabo de la siguiente manera, figura 11. El primer paso es localizar la mano, en este trabajo se utiliza el método de detección rápida de objetos; el siguiente paso es segmentar la mano del fondo, binarizando la imagen de la mano usando el algoritmo propuesto por [Otsu]; finalmente se aplican las operaciones morfológicas apertura y cierre, para mejorar la segmentación, es decir eliminar ruido existente.



**Figura 11: Proceso de detección de la mano.**

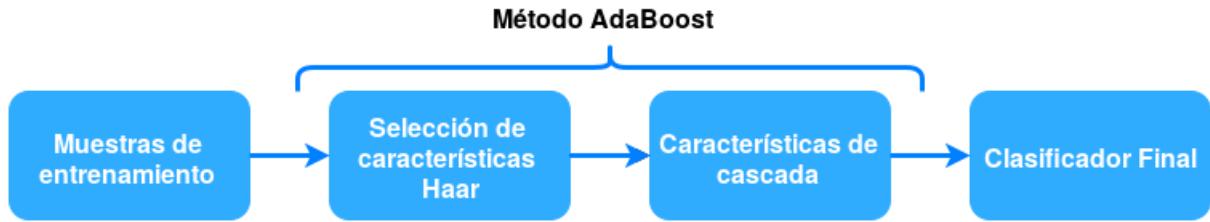
### **3.2.1. Método detección rápida de objetos usando características simples utilizando el clasificador AdaBoost en forma de cascada**

En este trabajo se utiliza el método detección rápida de objetos usando características simples utilizando el clasificador AdaBoost en forma de cascada, (Viola y Jones, 2001), el cual fue creado originalmente para atacar el problema de detección de rostros, este puede ser usado para detectar cualquier objeto, debido a la forma en que este fue creado, pues detecta un objeto clasificando imágenes basándose en el valor de características simples.

La técnica clasifica si el objeto se encuentra en la escena, usando una versión modificada del clasificador AdaBoost (Freund y Schapire, 1995) en forma de cascada, y discrimina el objeto tomando en cuenta el valor de las características Haar (Viola y Jones, 2001), las características son seleccionadas usando también el clasificador AdaBoost y el valor de estas es calculado mediante el uso de una imagen integral (Viola y Jones, 2001).

La figura 12 muestra un diagrama del proceso del método de detección, el primer paso es obtener la muestras de entrenamiento con las cuales se construirá el clasificador; el siguiente paso es seleccionar las características que formaran el clasificador, estas se escogen mediante el algoritmo de AdaBoost y su valor es calculado usando la imagen integral; el paso final que es construir el clasificador mediante el uso de Adaboost, en forma de cascada.

Enseguida se explica a detalle cada etapa del método (Viola y Jones, 2001).

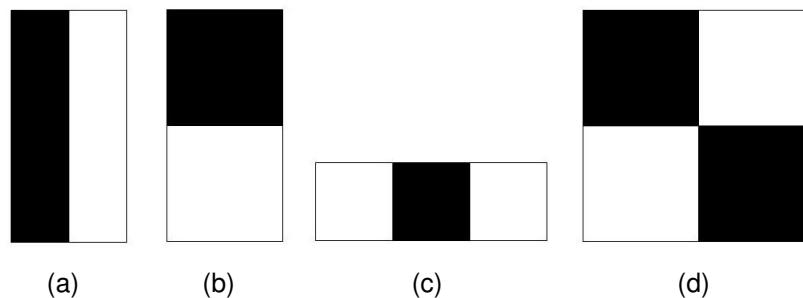


**Figura 12: Procedimiento del algoritmo de detección rápida de objetos.**

### 3.2.1.1. Características Haar

Las características Haar, son operadores rectangulares como los que se muestran en la figura 13. A continuación se explicarán los operadores Haar básicos:

- Las características con dos rectángulos 13(a), 13(b), contienen dos regiones rectangulares adyacentes, y el valor de la característica se calcula tomando la diferencia de la suma de ambas regiones.
- Las características con tres rectángulos 13(c), contienen tres regiones rectangulares adyacentes, y el valor de la característica se calcula sumando las regiones exteriores y restando la suma de la región interior.
- Las características con cuatro rectángulos 13(d), contienen cuatro regiones rectangulares adyacentes, y el valor de la característica se obtiene con la diferencia entre la suma de las regiones pares diagonales.



**Figura 13: Ejemplo de tipos de operadores Haar.**

### 3.2.1.2. Imagen integral

Uno de los aportes del método desarrollado por Viola y Jones es el concepto de imagen integral con la cual se calcula el valor de las características de manera rápida, es

decir en tiempo constante.

La imagen integral,  $SI$ , de un imagen,  $S(x, y)$ , es calculada como la suma del valor de los pixeles que se encuentran arriba y a la izquierda de cierta posición de la imagen a la cual se le quiere hacer el cálculo. Lo anterior se puede escribir como:<sup>4</sup>

$$SI(x, y) = S(x, y) + S(x - 1, y) + SI(x, y - 1) - SI(x - 1, y - 1)$$

La figura 14 muestra un ejemplo donde se calcula la imagen integral, fig. 14(b), de la imagen original 14(a).

1	1	1
1	1	1
1	1	1

(a) Imagen original

1	2	3
2	4	6
3	6	9

(b) Imagen integral

**Figura 14: Ejemplo del cálculo de la imagen integral.**

La imagen integral permite calcular la suma de los pixeles de cierta región usando solo los valores de las esquinas de la imagen integral de dicha región, la cual se obtiene como:<sup>5</sup>

$$REG(\alpha) = SI(A) + SI(D) - SI(B) - SI(C)$$

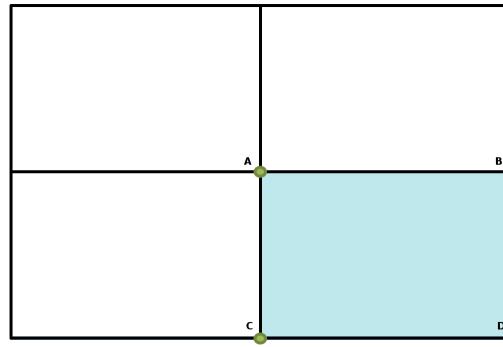
donde  $REG(\alpha)$  es la región a la cual se quiere calcular el valor de la suma de sus pixeles;  $A, B, C, D$  son las esquinas de dicha región, como se muestra en la figura 15, la región  $\alpha$  se encuentra resaltada en color azul.

### 3.2.1.3. Algoritmo AdaBoost

En el método de detección el clasificador AdaBoost es utilizado para seleccionar las características relevantes, con las cuales se podrá detectar el objeto. También es utilizado para construir el clasificador final pero en forma de cascada, el cual se explicará en la sección 3.2.1.4.

<sup>4</sup><https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>

<sup>5</sup><https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/>



**Figura 15: Regiones de la imagen integral.**

El algoritmo AdaBoost realiza la discriminación de objetos construyendo un clasificador fuerte,  $h(x)$ , llamado así debido a que tiene una precisión mayor en comparación con los clasificadores con los que es construido, clasificadores débiles,  $h_i(x)$ . Los clasificadores débiles son calculados de la siguiente manera:

$$h_i(x) = \begin{cases} 1, & \text{si } p_i f_i(x) < p_i \theta_i \\ 0, & \text{de otra forma.} \end{cases},$$

donde  $x$  es una sub-ventana de la imagen,  $f_i(x)$  es una característica,  $\theta$  es un umbral, y  $p_i(x)$  representa el signo de la desigualdad.

El clasificador fuerte es una combinación lineal de los clasificadores débiles, y se define de la siguiente forma:

$$h(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \dots + \alpha_n h_n(x),$$

donde  $n$  es el número de características,  $\alpha_i$  es el valor asociado a cada característica, el cual va entre 0 y 1.

Enseguida se presenta el algoritmo AdaBoost:

#### 3.2.1.4. Clasificador AdaBoost en cascada

El objetivo de realizar la detección utilizando un clasificador en forma de cascada es descartar de manera rápida las regiones donde no se encuentra el objeto.

---

**Algoritmo 1**

**Entrada:** El conjunto  $\{(x_1, y_1), \dots, (x_n, y_n)\}$  donde  $x_i$  representa las imágenes de entrenamiento,  $y_i = 0, 1$ , representa las imágenes negativas y positivas respectivamente.

**Salida:** El clasificador fuerte  $h(x)$ .

1: Se inicializan los pesos  $w_{1,i} = \frac{1}{2m}, \frac{1}{2l}$ , para  $y_i = 0, 1$  respectivamente, donde  $m$  y  $l$  son el número de imágenes negativas y positivas respectivamente.

2: **para**  $t = 1$  hasta  $T$  **hacer**

3: Se normalizan los pesos

$$w_{t,i} = \frac{w_{t,i}}{\sum_{j=1}^n w_{t,j}},$$

para que  $w_t$  sea una distribución de probabilidad.

4: **para** cada características  $j$  **hacer**

Entrenar un clasificador  $h_j$ , donde se utiliza una sola característica. El error  $\epsilon$  es evaluado con respecto a  $w_t$ ,

$$\epsilon = \sum_i w_i |h_i(x_i) - y_i|.$$

5: **fin para**

6: Escoger el clasificador  $h_i$  con el error más pequeño.

7: Se actualizan los pesos

$$w_{t+1,i} = w_{t,i} \beta_t^{1-e_i},$$

donde  $\beta_t = \frac{e_t}{1-e_t}$ , el valor de  $e_i = 0$  si  $x_i$  es clasificado correctamente de otra forma  $e_i = 1$ .

8: **fin para**

9: El clasificador final o clasificador fuerte es:

$$h(x) = \begin{cases} 1, & \sum_{t=1}^T \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^T \alpha_t \\ 0, & \text{de otra forma.} \end{cases},$$

donde  $\alpha_t = \log \frac{1}{\beta_t}$ .

---

El clasificador en cascada esta compuesto por etapas 16, cada una de estas es un clasificador fuerte. Este clasificador es entrenado por medio de AdaBoost. El cual se encarga de encontrar el orden de evaluación de las características relevantes.

La selección se realiza como se muestra en el algoritmo 2 , cumpliendo cierta precisión en la detección  $D$ , ver ecuacion 1 , y cierta tasa de falsos positivos  $F$ , ver ecuación 2 .

El proceso de detección funciona de la siguiente manera: la sub-ventana es evaluada en el primer clasificador; un resultado positivo desencadena la evaluación de un segundo clasificador, el cual también ha sido ajustado para alcanzar cierta precision; un resultado positivo en el segundo clasificador desencadena una tercera evaluación en el siguiente

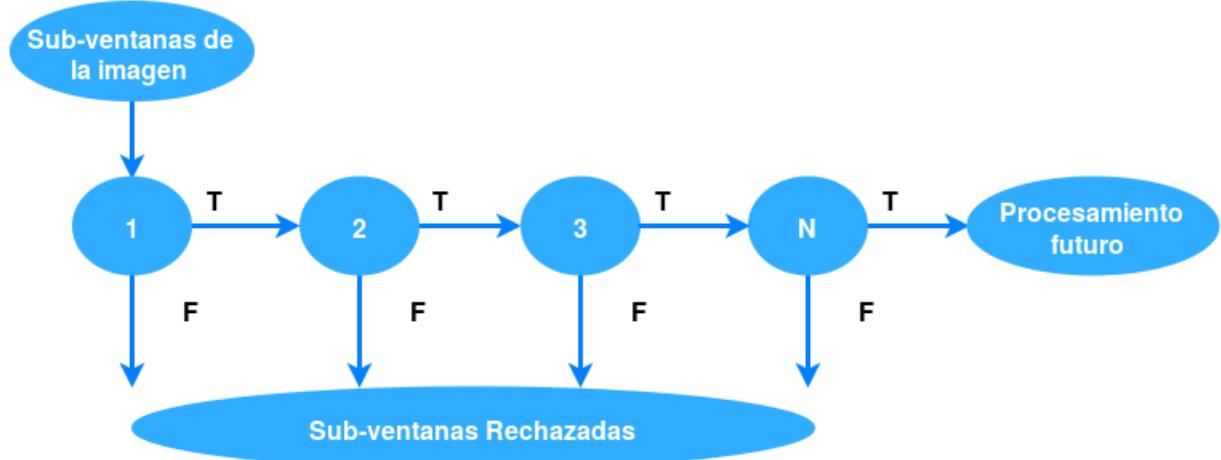


Figura 16: Proceso del clasificador en forma de cascada, donde  $F$  representa la tasa de falsos positivos del clasificador de cascada y  $T$ .

clasificador, y así sucesivamente. Un resultado negativo en cualquier punto del proceso de evaluación conduce al rechazo inmediato de la sub-ventana.

La tasa de detección del clasificador en forma de cascada es:

$$D = \prod_{i=1}^K d_i, \quad (1)$$

donde  $d_i$  es la tasa de precisión de detección del  $i$ -ésimo clasificador fuerte.

La tasa de precisión de falsos positivos,  $F$  del clasificador de cascada es:

$$F = \prod_{i=1}^K f_i, \quad (2)$$

donde  $K$  es el número de clasificadores fuertes y  $f_i$  es la tasa de precisión del  $i$ -ésimo clasificador.

### 3.2.2. Binarización

La binarización es una técnica de procesamiento de imágenes, la cual se encarga de transformar una imagen en escala de grises  $S(x, y)$  en una imagen binaria  $B(x, y)$  es decir, los pixeles de la imagen toman un valor de 0 ó 1. Para formar la imagen binaria un valor o umbral de la imagen en escala de grises es seleccionado.

Una vez seleccionado el umbral,  $T$ , los pixeles de la imagen son discriminados. Si el valor

---

**Algoritmo 2**

**Entrada:** Imágenes positivas  $P$ , negativas  $N$ ,  $f$  el valor máximo de precisión de falsos positivos por etapa,  $d$  es el valor mínimo de precisión en la detección por etapa.

**Salida:** El clasificador en forma de cascada.

```

 $F_0 = 1, D_0 = 1.$ 
2:  $i = 0.$ 
mientras  $F_i > F_{\text{Tarjet}}$  hacer
4:    $i = i + 1.$ 
     $n_i = 0, F_i = F_{i-1}.$ 
6:   mientras  $F_I > F \times fp_{i-1}$  hacer
         $n_i = n_i + 1.$ 
8:   Entrenar un clasificador usando AdaBoost con  $P, N$  y  $n_i$  características.
       Evaluar el clasificador de cascada para determinar  $F_i$  y  $D_i$  en el conjunto de validación.
10:  Decrementar el umbral para el  $i$ -ésimo clasificador hasta que el actual clasificador en cascada tenga un grado de detección de por lo menos  $d \times D_i - 1.$ 
fin mientras
12:   $N = 0.$ 
si  $F_i > F_{\text{Tarjet}}$  entonces
14:   Evaluar el actual clasificador en cascada en el conjunto de imágenes negativas y
       poner cualquier detección falsa en el conjunto  $N.$ 
fin si
16: fin mientras

```

---

de los pixeles de la imagen es mayor o igual al umbral entonces el valor de los pixeles de la imagen binaria es 1, si no toma el valor de 0. Es decir:

$$B(x, y) = \begin{cases} 1, & \text{Si } S(x, y) \geq T \\ 0, & \text{de otra forma} \end{cases}.$$

Existen diversas técnicas para binarizar una imagen, estas se pueden clasificar en dos grupos: global y local. Los métodos globales calculan un umbral, el cual es utilizado para todos los pixeles de la imagen y los métodos locales que calculan varios umbrales para ciertas regiones de la imagen (Chaki *et al.*, 2014). En este trabajo se utiliza el método desarrollado por (Otsu, 1979).

### 3.2.3. Operaciones Morfológicas

Otra técnica muy utilizada en procesamiento de imágenes son las operaciones morfológicas que son un conjunto de operaciones no lineales, la idea es que al aplicar alguna de estas operaciones el ruido sea removido tomando en cuenta la forma y estructura de

la imagen.

Las operaciones morfológicas (Premaratne, 2013) utilizan un elemento estructural el cual se aplica por toda la imagen, los elementos estructurales pueden ser de distintas formas como las que se muestran en la figura 17.

1	1	1
1	1	1
1	1	1

(a) Rectángulo de  $3 \times 3$ .

0	1
1	1
0	1

(b) Figura de  $3 \times 2$ .

0	0	1	0	0
0	0	1	0	0
1	1	1	1	1
0	0	1	0	0
0	0	1	0	0

(c) Cruz de  $5 \times 5$ .

**Figura 17: Ejemplos de elementos estructurales.**

Existen distintas operaciones morfológicas, las básicas o principales son la dilatación y erosión las cuales se explican enseguida junto con la apertura y cierre.

### 3.2.3.1. Dilatación

La dilatación es una operación que añade pixeles a la orilla de los objetos que se encuentran en la imagen. En la figura 18(b) se aplica esta operación a la figura 18(a).

La dilatación se define como:

$$S \oplus EX = \{S | EX_S \subseteq S\},$$

donde  $EX_S$  es el elemento estructural trasladado con la imagen.

### 3.2.3.2. Erosión

La erosión remueve pixeles a la orilla de los objetos que se encuentran en la imagen. En la figura 18(c) se muestra el resultado de aplicar la operación a la figura 18(a).

La erosión se define como:

$$S \ominus EX = \{S | EX_S \subseteq S\},$$

donde  $EX_S$  es el elemento estructural trasladado con la imagen.

### 3.2.3.3. Apertura

La operación apertura abre huecos entre objetos conectados por un enlace delgado de pixeles, también suaviza los contornos del objeto. Esta operación se calculada realizando dos operaciones básicas una erosión seguida de una dilatación

La apertura se define como:

$$S \circ EX = (S \ominus EX) \oplus EX.$$

La figura 18(d) muestra el resultado de aplicar la operación apertura a la figura 18(a).

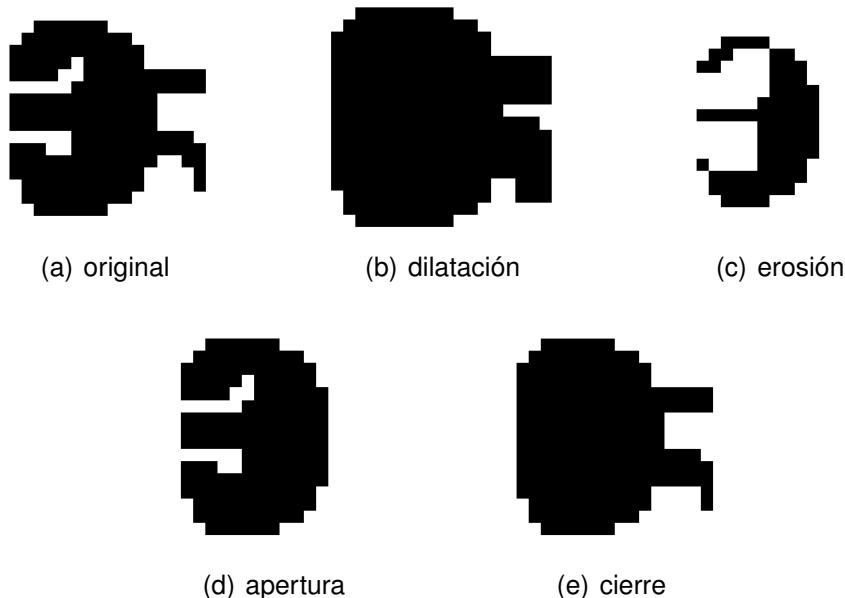
### 3.2.3.4. Cierre

La operación cierre elimina huecos pequeños y rellena huecos en los contornos. El cierre es calculado realizando las operación de dilatación seguida de la erosión.

El cierre se define como:

$$S \bullet EX = (S \oplus EX) \ominus EX.$$

La figura 18(e) muestra el resultado de aplicar la operación a la figura 18(a).

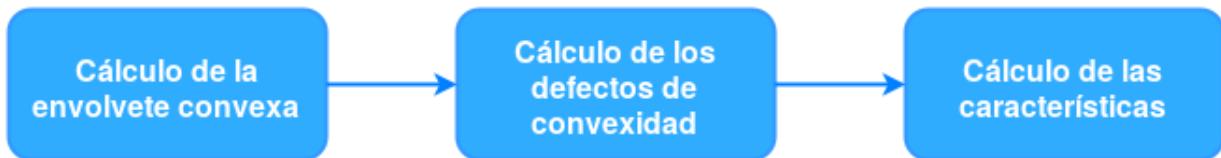


**Figura 18: Aplicación de las principales operaciones morfológicas a la imagen que se encuentra en el inciso a, (Smith, 1999).**

### 3.3. Extracción de características

La idea de esta etapa es obtener las características de la imagen que sean capaces de describir la mano, de manera que con estas, se pueda reconocer los gestos realizados.

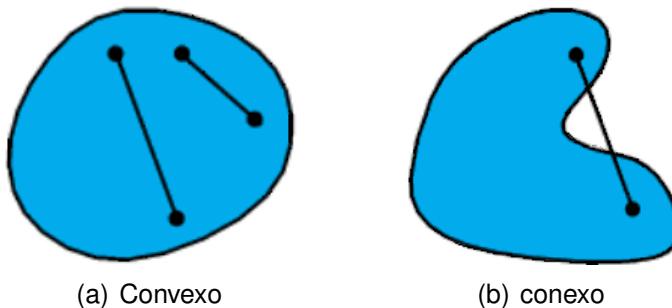
En este trabajo se extraen características geométricas, las cuales son extraídas de la siguiente forma 19: el primer paso es encontrar la envolvente convexa de la mano para posteriormente calcular los defectos de convexidad, una vez aplicados estos algoritmos se calcula el número de dedos de la mano entre otras características; finalmente las características calculadas se guardan en un vector.



**Figura 19:** Proceso de la extracción de características.

A continuación se definen los conceptos anteriores y el de conjunto convexo.

Sea  $A$  un conjunto en el espacio euclidiano  $\mathbb{R}^d$ , donde  $d$  es la dimensión del espacio euclidiano.  $A$  es un conjunto convexo<sup>6</sup> si contiene todos los segmentos de línea que unen a cualquier par de puntos pertenecientes al conjunto.



**Figura 20:** Ejemplo de un conjunto conexo y un convexo. Image recuperada de 6

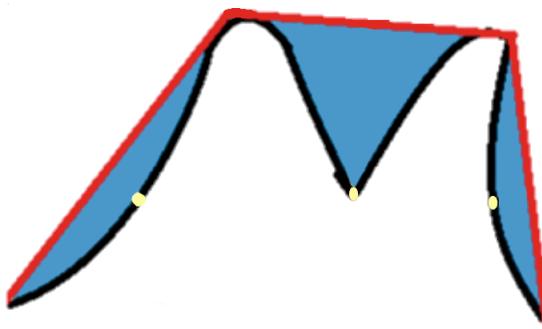
Sea  $B$  un conjunto de puntos en el plano Eucliano, la envolvente convexa de  $B$  es el conjunto convexo más pequeño que contiene a todos los puntos en  $B$ . En la imagen 21

<sup>6</sup> Weisstein, Eric W. "Convex." From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/Convex.html>

se muestra de color rojo la envolvente convexa de la figura cuyo contorno se encuentra de color negro.

Los defectos de convexidad de la envolvente convexa, son el conjunto de puntos que no pertenecen al casco convexo. El defecto es el espacio que existe entre el contorno de la envolvente convexa y del objeto.

Sea  $CD = \{cd_1, cd_2, \dots, cd_n\}$  el conjunto de defectos de convexidad de una envolvente convexa. Cada defecto esta compuesto por tres elementos: el punto de inicio del defecto  $s_i(x, y)$ , el punto con mayor distancia de la envolvente al objeto,  $d_i(x, y)$  y el punto final del defecto,  $e_i(x, y)$ . En la imagen 21 los puntos amarillos representan los puntos de profundidad de los defectos de convexidad.



**Figura 21:** En la imagen se aprecia de color rojo la envolvente convexa, de negro el contorno de la figura, y los puntos amarillos son el punto de profundidad de los defectos de convexidad.

Usando las técnicas anteriores podemos extraer características importantes como el número de dedos, la posición del centro de la palma de mano, la posición de la punta de los dedos, la posición del inicio o raíz de los dedos, el ángulo del centro de la palma de la mano a la punta de los dedos, ángulo  $TC$ , el ángulo del centro de la palma de la mano a al inicio de los dedos, ángulo  $RC$  y la distancia vertical del centro de la palma de la mano al inicio de los dedos. Enseguida se explica como son obtenidas las características mencionadas.

El número de dedos que se encuentran levantados es calculado con el algoritmo 3 desarrollado por (Kathuria, 2011), el cual utiliza los defectos de convexidad en específico los conjuntos de puntos de inicio,  $\mathcal{S} = \{s_1(x, y), s_2(x, y), \dots, s_n(x, y)\}$ , los puntos de mayor distancia,  $\mathcal{D} = \{d_1(x, y), d_2(x, y), \dots, d_n(x, y)\}$ , las distancias del punto de inicio al de mayor distancia,  $\delta = \{\delta_1(x, y), \delta_2(x, y), \dots, \delta_n(x, y)\}$ , donde  $n$  es el número total de defectos de la envolvente convexa .

Sea  $C_r(x, y)$ , el punto que representa el centroide del rectángulo más pequeño que encierra a la mano,  $L_r$ , la altura del rectángulo y  $k$  una constante.

---

**Algoritmo 3** Cálculo del número de dedos levantados de la mano.

---

**Entrada:** Los conjuntos  $\mathcal{S}$ ,  $\mathcal{D}$ , el punto  $C_r$  y el valor  $L_r$ .

**Salida:** Número de dedos levantados,  $Nf$ .

```

1: para  $i = 1$  hasta  $n$  hacer
2:    $k = 6$ .
3:   si  $[s_i(x, y) < C_r(x, y) \text{ O } d_i(s, y) < C_r(x, y)] \text{ Y } s_i(x, y) < d_i(x, y) \text{ Y } \delta_i > \frac{L_r}{k}$  entonces
4:      $Nf = Nf + 1$ 
5:   fin si
6: fin para
```

---

La posición de la raíz de los dedos,  $Fr(s, y)$  puede ser calculada usando los defectos de convexidad (Hummel *et al.*, 2014), en específico los puntos de profundidad,  $d(x, y)$ . Se calcula tomando el punto medio de los puntos de profundidad consecutivos encontrados en medio de los dedos, como se muestra en la figura 22 . Es decir:

$$Fr_i(x, y) = \left( \frac{x_{d_i} + x_{d_{i-1}}}{2}, \frac{y_{d_i} + y_{d_{i-1}}}{2} \right),$$

donde  $i$  representa el número de dedo, cuando  $i = 0$  se toma el punto de profundidad anterior al dedo, si  $i = 6$  se toma el punto de profundidad posterior al dedo.

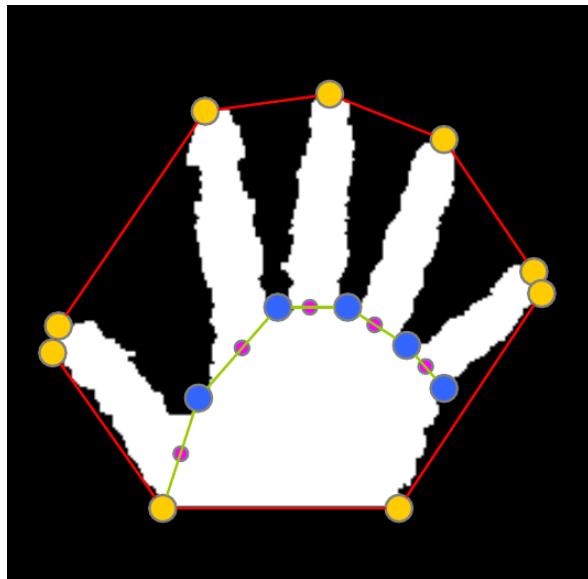


Figura 22: La figura muestra parte de la mano y en ella se aprecia los siguientes elementos: en color rojo la envolvente convexa, en amarillo los puntos de inicio y final de los defectos de convexidad, en color azul los puntos de profundidad de los defectos, en verde la linea que une a los puntos de profundidad consecutivos y finalmente en morado los puntos medios, (Hummel *et al.*, 2014)

El centro de la palma de mano también es calculado, con los puntos de profundidad de los defectos de convexidad, para ello se toma como centro de la palma el centro del rectángulo mas chico que une rodea a los puntos de convexidad.

El ángulo  $RC$  es el formado por el eje  $y$  y la línea que une al punto que representa la posición de la raíz de los dedos,  $Fr(x, y)$  con el centro de la palma de la mano,  $Ch(x, y)$ , (Sgouropoulos *et al.*, 2014).

$$\angle RC = 90^\circ - \tan^{-1} \left( \frac{y_{Fr} - y_{Ch}}{x_{Fr} - x_{Ch}} \right).$$

El ángulo  $TC$  es el formado por el eje  $y$  y la línea que une al centro de la mano,  $Ch(x, y)$  y con el de la punta de los dedos,  $Ft(x, y)$ , (Sgouropoulos *et al.*, 2014). El ángulo anterior se representa como:

$$\angle TC = 90^\circ - \tan^{-1} \left( \frac{y_{Ft} - y_{Ch}}{x_{Ft} - x_{Ch}} \right).$$

La distancia  $PC$  es la distancia vertical de la raíz del dedo al centro de la palma de la mano. Esta distancia es invariante al tamaño de la mano ya que es dividida por el tamaño de la palma, que se toma como el ancho del rectángulo que encierra la palma.

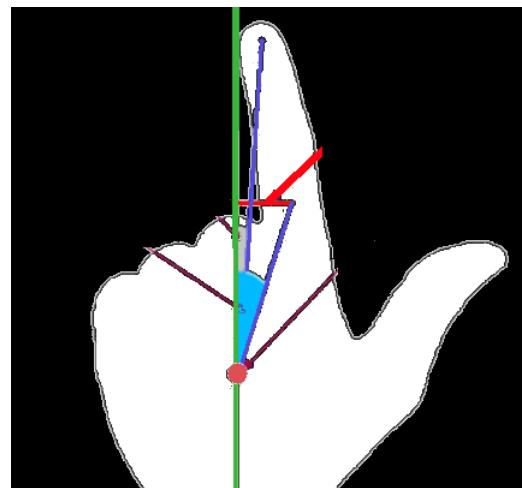


Figura 23: En la imagen se representan los siguientes elementos, el eje vertical con respecto a la mano se encuentra como una línea de color verde; la línea roja representa la distancia del eje vertical a la raíz de los dedos; el punto rosa representa el centro de la palma de la mano, el área azul representa el ángulo de que existe de la línea que une al centro con la raíz de los dedos y finalmente el área amarilla representa el ángulo que forma la línea del centro a la punta de los dedos, (Sgouropoulos *et al.*, 2014).

Una vez que todas las características son calculadas estas son guardadas en un vector, llamado vector de características. La dimensión del vector es el número de características que este contiene.

### 3.4. Reconocimiento

Es la etapa final del reconocimiento, es donde finalmente el gesto puede ser interpretado por la computadora.

En este trabajo el reconocimiento se realiza utilizando el algoritmo de máquinas de soporte vectorial SVM, (Cortes y Vapnik, 1995), un método de aprendizaje de máquina supervisado el cual es utilizado para resolver problemas de clasificación y regresión. SVM tiene como objetivo crear un modelo basado en datos conocidos, datos de entrenamiento, donde este modelo es capaz de predecir a que clase pertenecen datos nuevos.

SVM realiza la clasificación separando las clases calculando el hiperplano que tengan el margen de separación más grande 24 .

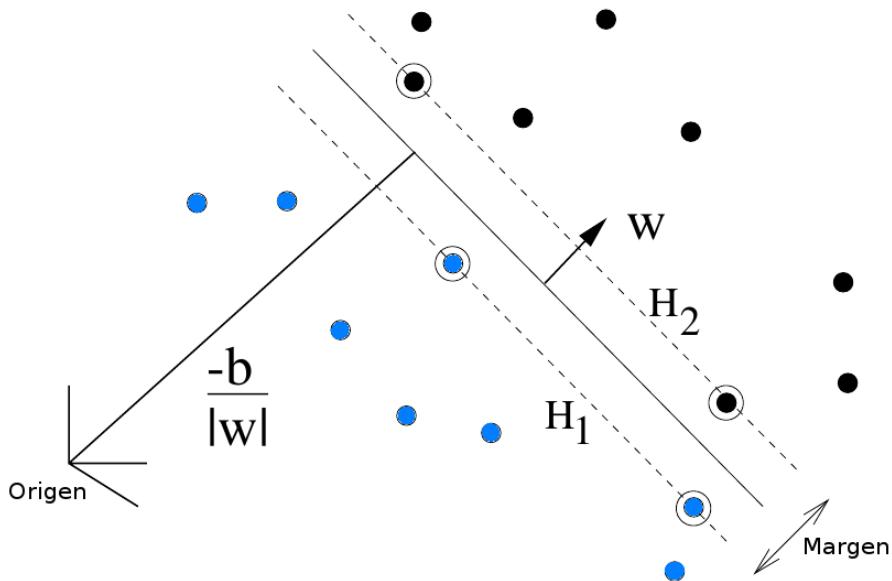


Figura 24: La imagen muestra la separación de dos clases, (los círculos en color azul y negro), mediante un hiperplano óptimo; donde  $w$  representa la normal al hiperplano,  $\frac{-b}{w}$  la distancia el hiperplano al origen. (Burges, 1998).

Enseguida se explica el caso cuando las clases son linealmente separables.

Dado  $N$  muestras de entrenamiento  $x_i$ , de dimensión  $D$ , dos clases distintas  $y_i = -1$  ó  $+1$

es decir:

$$\{x_i, y_i\} \quad \text{donde} \quad i = 1, \dots, N \quad y \in \{-1, 1\} \quad x \in \Re^D.$$

Sea

$$w \cdot x + b = 0, \quad (3)$$

el hiperplano óptimo que separa a las clases, donde  $w$  es la normal al hiperplano,  $\frac{b}{\|w\|}$  es la distancia perpendicular desde el hiperplano al origen.

Sea  $d_+$ , la menor distancia del hiperplano que separa a las muestras positivas de las negativas y  $d_-$ , la menor distancia del hiperplano que separa a las muestras negativas de las positivas. Se define el margen del hiperplano como la suma de estas distancias, es decir:  $d_+ + d_-$ .

Para el caso cuando las clases son linealmente separables basta con encontrar el hiperplano con el margen mayor. Es decir que el hiperplano puede ser calculado seleccionando  $w$  y  $b$  de manera que los datos de entrenamiento cumplan con:

$$w \cdot x_i + b \geq +1 \quad \text{para} \quad y_i = +1 \quad (4)$$

$$w \cdot x_i + b \leq -1 \quad \text{para} \quad y_i = -1 \quad (5)$$

Combinando las desigualdades anteriores, se obtiene:

$$y_i(x_i \cdot w + b) - 1 \geq 0 \quad \forall i \quad (6)$$

Tomando en cuenta los puntos en donde se cumple la igualdad de 4. Estos puntos se encuentran sobre el hiperplano  $H_1$ , el cual se escribe como:

$$w \cdot x_i + b = +1, \quad (7)$$

con normal  $w$  y una distancia perpendicular desde el origen de  $\frac{|1-b|}{\|w\|}$ . Similarmente para la ecuación 5, entonces el hiperplano  $H_2$  se describe como:

$$w \cdot x_i + b = -1, \quad (8)$$

con normal  $w$  y una distancia perpendicular desde el origen de  $\frac{|-1-b|}{\|w\|}$ .

Como  $d_+ = d_- = \frac{1}{\|w\|}$ , el margen es  $\frac{2}{\|w\|}$ . Los hiperplanos son paralelos pues tienen la misma normal, también ninguna muestra de entrenamiento cae entre ellos. Entonces se puede encontrar un par de hiperplanos que tengan un margen máximo minimizando  $\|w\|^2$ , es decir:

$$\min \quad \frac{1}{2} \|w\|^2 \quad \text{tal que} \quad y_i(w \cdot x_i + b) - 1 \geq 0. \quad (9)$$

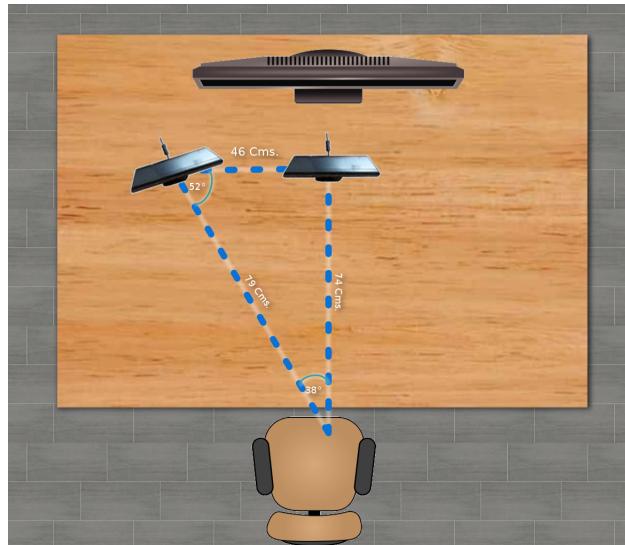
## Capítulo 4. Implementación del sistema de reconocimiento de gestos propuesto

---

En este capítulo se describen los detalles de implementación de cada etapa del sistema.

### 4.1. Adquisición de los datos

Como se vio en el capítulo 3 sección 3.1 los datos provienen de los sensores de profundidad de dos dispositivos Kinect, estos se encuentran ubicados uno frente al usuario (Kinect frontal), y otro al lado izquierdo (Kinect lateral), con una distancia de 74 y 79 cm. respectivamente; y entre ellos de 46 cm. como se muestra en la figura 25.



**Figura 25: Configuración del sistema de reconocimiento de gestos**

Una vez que el flujo de datos de los sensores de profundidad es capturado este es representado como una imagen en escala de grises de 8 bits de 640 píxeles de ancho por 480 píxeles de largo. En las imágenes se puede apreciar detalles pequeños, es decir cambios en la profundidad de hasta 1 mm. esto debido a que la escala de grises inicia cada 26 cm. En la siguiente imagen se puede apreciar un ejemplo de las imágenes de profundidad. 26

Por la naturaleza del Kinect, las imágenes obtenidas de ambos sensores contiene ruido, como el se muestra en la figura 27; el ruido es reducido usando un filtro de mediana,



**Figura 26:** Representación de los datos capturados por los Kinect

este es aplicado en toda la imagen usando una ventada de tamaño 13. La imagen resultante  $S(x, y)$  es como la que se muestra en la figura 27.

Se aprecia en la imagen siguiente que gran parte del ruido es reducido obteniendo una



**Figura 27:** Representación de los datos capturados por los Kinect

mejora en la imagen, desafortunadamente todavía existe ruido en la imagen, este puede ser eliminado casi en su mayoría si el tamaño de la ventana aumenta pero se pierde información importante de la imagen, de manera que se decidió optar por el tamaño de ventana, antes mencionado. En la imagen también se aprecia el fondo negro, esto es debido a que se discriminó el fondo que estuviera a un distancia de más de 2 m. del sensor.

## 4.2. Detección

En este trabajo se utiliza el algoritmo de detección de objetos desarrollado por Viola y Jones (2001), como se mostró en el capítulo 3 sección 3.2.1, el algoritmo clasifica las imágenes basándose en el valor de características, el clasificador es construido usando el algoritmo de AdaBoost en forma de cascada.

La selección de las características se llevó acabo por medio de una versión modificada del algoritmo AdaBoost; la implementación se realizó utilizando el software OpenCV Haar training classifier<sup>1</sup>. Se entrenó con 1000 imágenes positivas (imágenes de profundidad de la mano), y 2000 negativas, (imágenes de fondo de distintos escenarios). Las imágenes positivas fueron generadas de 300 imágenes de poses; 3 poses distintas, 100 de cada pose, usando el software Create Samples<sup>2</sup>. Todas las imágenes usadas fueron tomadas de nuestra base de datos<sup>3</sup>.

Nuestra base de datos contiene gran cantidad de imágenes de profundidad. Imágenes de fondo y de poses de la mano, estas fueron tomadas a una distancia de entre 60 y 200 cm. Las imágenes de profundidad de la mano fueron tomadas de 6 personas distintas con tres distintas poses: palma con los dedos separados 28(a), palma con dedos juntos 28(c) y finalmente el puño 28(b), como se muestran en la figura 28. Las imágenes de fondo fueron tomadas de distintos escenarios como se muestra en la figura 29. El programa de captura de las imágenes puede ser encontrado en Github 3.

Los parámetros utilizados para la obtención del clasificador final fueron: el porcentaje de precision de detección de [%] y la tasa de falsos positivos aceptados de [%]. El resultado final del entrenamiento fue en clasificador AdoBoost en forma de cascada, que consta de 19 etapas. El clasificador resultante se encuentra en Github 3, en formato XML.

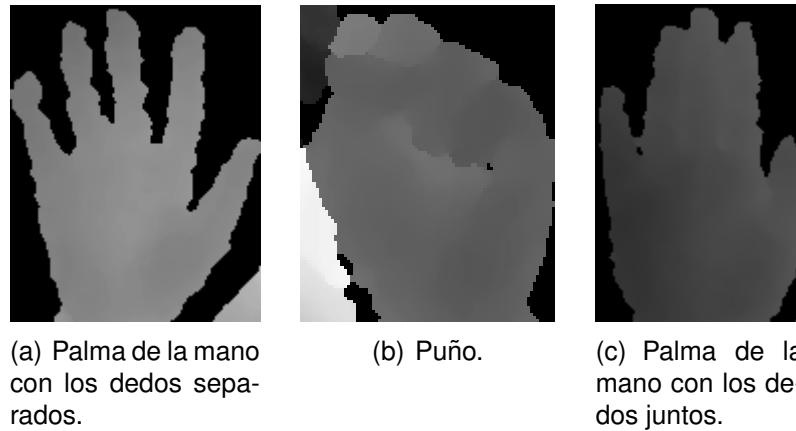
Con el clasificado obtenido, se localiza la mano en cada cuadro proveniente de los dispositivos Kinect, una ventana inicial de tamaño  $ka$  se desliza por la imagen. Para eliminar falsos positivos que pudieran ocurrir en la detección de la mano se utiliza un algoritmo equivalente al de [referencia], este se muestra en el apéndice ??.

---

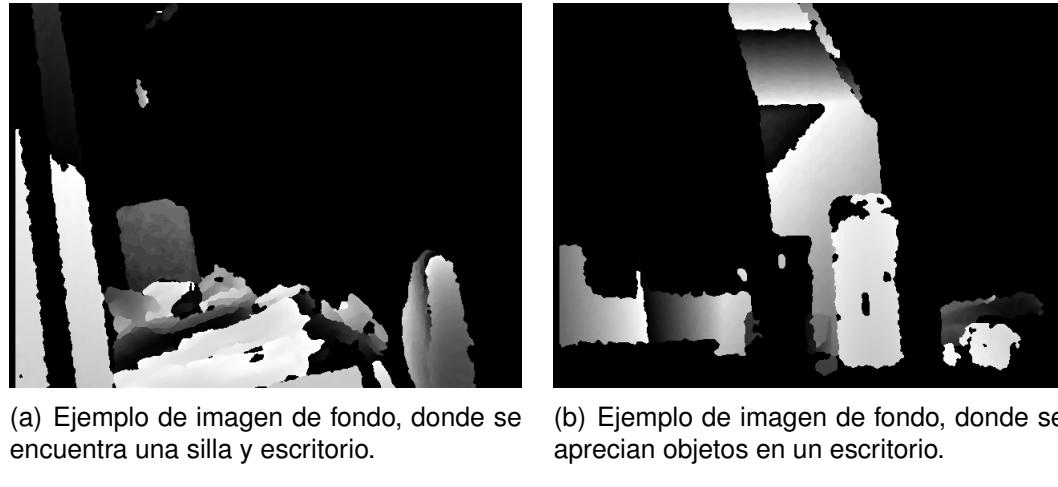
<sup>1</sup><https://github.com/mrnugget/opencv-haar-classifier-training>

<sup>2</sup><http://note.sonots.com/SciSoftware/haartraining.html>

<sup>3</sup> <https://github.com/americamm>



**Figura 28:** Ejemplo de imágenes de poses de nuestra base de datos.

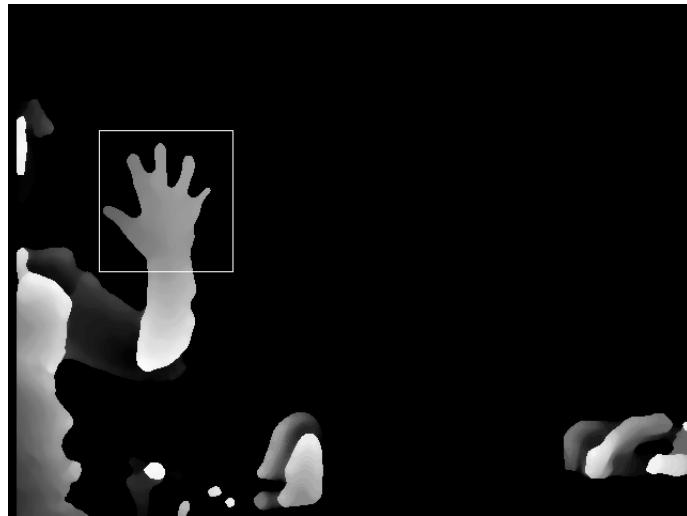


**Figura 29:** Imágenes del fondo de nuestra base de datos.

Una vez que la mano se localiza la región de interés  $ROI(x, y)$  es seleccionada alrededor de la mano, como se observa en la figura 30.

Ya que se tiene localizada el área donde se encuentra la mano, el siguiente paso es segmentar la mano del ROI. La segmentación se realiza binarizando el área del ROI, solo se toma esta área, para que el proceso sea más rápido. La binarización se lleva a cabo usando el algoritmo de Otsu, el resultado se muestra en la figura 31.

Para mejorar la binarización, el ruido existe en la imagen es eliminado aplicando dos operaciones morfológicas, apertura y cierre, en ese orden. Las operaciones anteriores utilizan un elemento estructural rectangular; para la operación de apertura el tamaño del elemento es de  $3 \times 9$  pixeles; para el cierre se aplicó con un tamaño  $3 \times 11$  pixeles.



**Figura 30: Localización y selección de la mano, en la imagen de entrada del Kinect 2.**



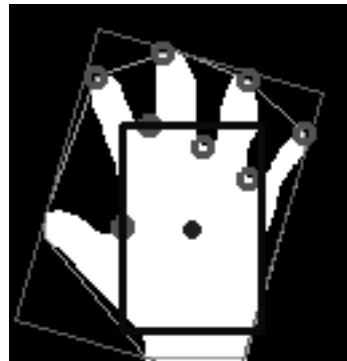
**Figura 31: Binarización de ROI y aplicación de las operaciones morfológicas a la mano localizada en la figura anterior.**

#### 4.3. Extracción de características

Como se vio en el capítulo 3 sección 3.3 las características de la mano son extraídas utilizando los algoritmos de envolvente convexa y defectos de convexidad.

La figura muestra un ejemplo de la aplicación de estos algoritmos al ROI binarizado.

Una vez aplicados estos dos algoritmos se calcula: el número de dedos levantados, la posición de la punta de los dedos, 32 , la posición de la raíz de los dedos, 32 , el centro de la palma de mano, 32 , los ángulos que existe del centro de la mano a la punta de los dedos, los ángulos que existen del centro a la raíz de los dedos, la distancia que existe del centro de los dedos a la raíz de los dedos. y las puntas de estos que son fundamentales para calcular las demás características. En la imagen siguiente se muestran algunas de estas características.



**Figura 32: Resultado del cálculo de las características de la mano.** Se muestra en color azul la punta de los dedos, en color verde la posición de la raíz de los dedos y en rojo el centro de la palma de la mano.

Para la implementación de las características, se tomo parte del código proveniente de <sup>4</sup>.

Las características se guardan en un vector de características de dimensión 32. La valor de la dimension del vector esta dada por la unión del conjunto de características obtenidas por cada imagen de la mano obtenida por cada Kinect. En cada vector las características provenientes del Kinect 1 son almacenadas primero seguidas de las del Kinect 2.

#### 4.4. Reconocimiento

En este trabajo se reconocen gestos estáticos y dinámicos utilizando el algoritmo de clasificación de máquina de soporte vectorial.

Como se vio en el capítulo 3 sección 3.4 SVM es un algoritmo de aprendizaje de máquina supervisado, por lo que es necesario tener imágenes de los gestos a reconocer ya que con estas el clasificador es entrenado y el modelo de clasificación puede ser creado.

La implementación de SVM se lleva acabo usando LibSVMSharp <sup>5</sup> un wrapper de la librería LibSVM (Chang y Lin, 2011).

---

<sup>4</sup>buscar la pagina ;/

<sup>5</sup><https://github.com/ccerhan/LibSVMsharp>

#### 4.4.1. Reconocimiento de gestos estáticos

El sistema reconoce dos gestos estáticos: el puño y la palma de la mano con los dedos separados. El reconocimiento del gesto se lleva analizando un solo cuadro o imagen.

El modelo que reconoce los gestos fue creado mediante SVM. Para el entrenamiento se tomaron 200 imágenes de las dos distintas pose como las que se muestran en la fig 33. Se utilizó un kernel exponencial y validación cruzada con 5 pliegues.



(a) Palma de la mano con los dedos separados.

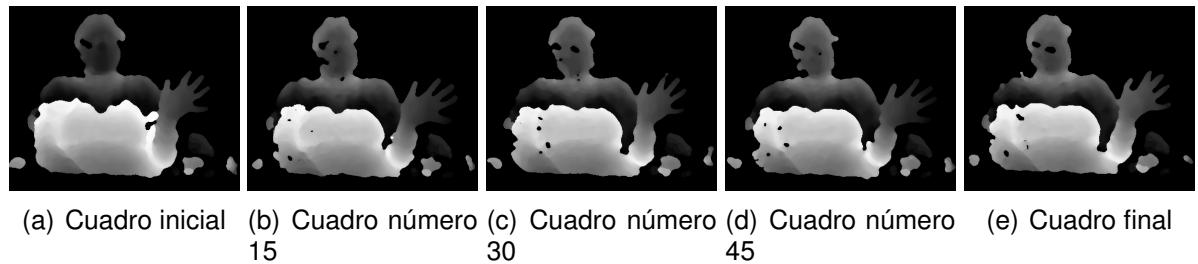
(b) Puño.

**Figura 33: Ejemplo de imágenes de poses de nuestra base de datos.**

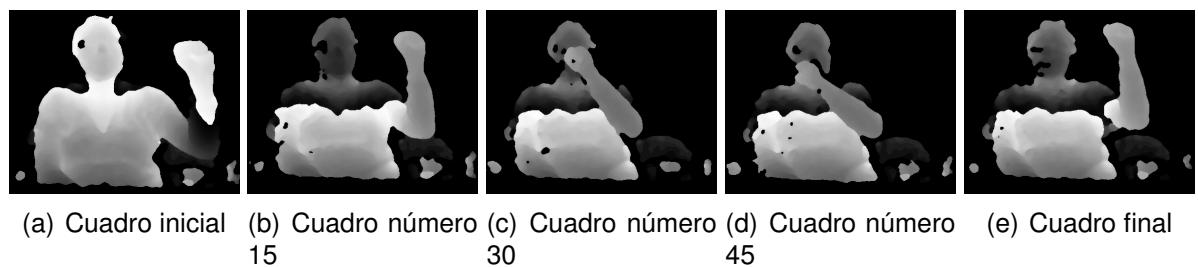
#### 4.4.2. Reconocimiento de gestos dinámicos

En este trabajo se toma un gesto dinámico como una secuencia de 30 cuadros consecutivos en los cuales se realiza misma pose solo que esta en movimiento. De manera que el sistema reconoce el gesto dinámico identificando y siguiendo cada pose en cada cuadro en una secuencia. La trayectoria del gesto es obtenida tomando en cuenta la posición del centro de la mano el cual es calculado en cada cuadro.

El sistema reconoce dos gestos dinámicos, los cuales son los mismos gestos estáticos en movimiento.



**Figura 34:** Secuencia del gesto dinámico de la palma de la mano abierta, la vista es desde el Kinect frontal.



**Figura 35:** Secuencia del gesto dinámico del puño, la vista es desde el Kinect frontal.

# Capítulo 5. Resultados

---

En este capítulo se presentan los resultados de las pruebas realizadas al sistema. El desempeño del sistema es evaluado con respecto al error de clasificación.

Los experimentos y el sistema propuesto fue implementado en una computadora de escritorio Dell con un procesador Intel(R) Xeon(R) CPU E5-1603, 16GB de memoria RAM, Windows 7 de 64 bits. La implementación del sistema se realizó en C# utilizando Emgu 2.4.10<sup>1</sup> un wrapper de OpenCV<sup>2</sup>.

Para analizar la tasa de precisión de reconocimiento del sistema propuesto se realizaron varios experimentos en distintas circunstancias. Por ejemplo se analizó el rendimiento del sistema a tres diferentes distancias 70, 80, 90 cm. del Kinect frontal. También en diferentes circunstancias de iluminación, con iluminación estándar, media y sin iluminación.

Se utilizaron imágenes reales capturadas por los sensores de profundidad de  $640 \times 480$  pixeles. Las imágenes son de 5 personas distintas, realizando los gestos de puño y el de palma de la mano con los dedos separados, para los gestos estáticos. Para los gestos dinámicos se tomaron los gestos de 3 personas cada una de ellas realizaron los gestos estáticos anteriores en movimiento.

En las secciones siguientes se explica cada experimento y resultados de estos.

## 5.1. Experimentos de gestos estáticos

La evaluación del sistema en cuanto al reconocimiento de los gestos estáticos, se determinó conforme al resultado de los experimentos realizados en circunstancias de iluminación, (estándar, media, baja). En cada conjunto de experimentos se tomó en cuenta la distancia, ya que en cada grupo se analizaron tres distancias, 70, 80, 90 cm.

Se analizaron dos gestos estáticos la palma de la mano con los dedos separados, Gesto 1 y el puño, Gesto 2, de 5 usuarios distintos. Para cada experimento se escogió al azar 200 imágenes de cada gesto del conjunto de las imágenes capturadas.

Enseguida se presentan los resultados de cada experimento realizado.

---

<sup>1</sup>[http://www.emgu.com/wiki/index.php/Main\\_Page](http://www.emgu.com/wiki/index.php/Main_Page)

<sup>2</sup><http://opencv.org/>

### 5.1.1. Experimentos con iluminación

Para este experimento las imágenes se capturaron en un laboratorio con iluminación estándar, como la que se muestra en la figura 36.



**Figura 36: Laboratorio en condiciones estándar de iluminación.**

En el primer experimento el usuario esta a una distancia de 70 cm. del Kinect frontal Se muestra un ejemplo de las imágenes obtenidas de un usuario, para las imágenes de los demás usuarios ver el apéndice . En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	170	30
Gesto 2	21	179

**Tabla 1: Matriz de confusión del experimento uno, utilizando ambos Kinect.**

	Gesto 1	Gesto 2
Gesto 1	127	73
Gesto 2	6	194

**Tabla 2: Matriz de confusión del experimento uno, utilizando solo el Kinect frontal.**

En el segundo experimento el usuario esta a una distancia de 80 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

En el tercer experimento el usuario esta a una distancia de 90 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran

	Gesto 1	Gesto 2
Gesto 1	96	104
Gesto 2	16	185

**Tabla 3: Matriz de confusión del experimento dos, con dos Kinect**

	Gesto 1	Gesto 2
Gesto 1	88	112
Gesto 2	6	183

**Tabla 4: Matriz de confusión del experimento dos, utilizando solo el Kinect frontal.**

los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	93	107
Gesto 2	10	190

**Tabla 5: Matriz de confusión, los datos se capturaron usando dos Kinect.**

	Gesto 1	Gesto 2
Gesto 1	101	99
Gesto 2	17	183

**Tabla 6: Matriz de confusión de los gestos 1 y 2. Utilizando un Kinect.**

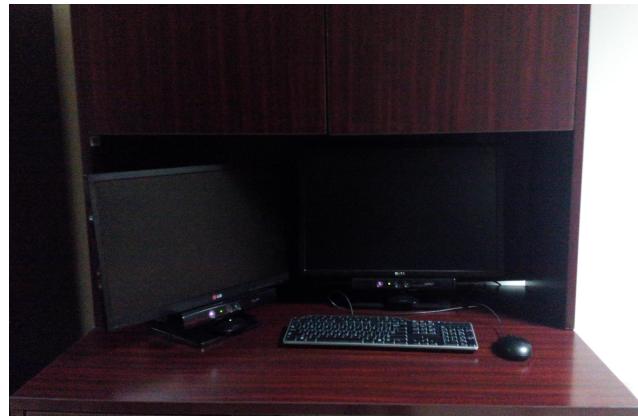
### 5.1.2. Experimentos con iluminación media

Para el conjunto de estos experimentos, las imágenes se capturaron en un laboratorio con iluminación media, como la que se muestra en la figura. A tres distancias 70, 80 y 90 cm.

En el primer experimento el usuario esta a una distancia de 70 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	178	22
Gesto 2	84	105

**Tabla 7: Matriz de confusión**



**Figura 37: Laboratorio en condiciones con iluminación media.**

	Gesto 1	Gesto 2
Gesto 1	136	64
Gesto 2	7	193

**Tabla 8: Matriz de confusión. Se utilizo un Kinect para el análisis del reconocimiento**

En el segundo experimento el usuario esta a una distancia de 80 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1		
Gesto 2		

**Tabla 9: Matriz de confusión**



	Gesto 1	Gesto 2
Gesto 1		
Gesto 2		

**Tabla 10: Matriz de confusión, utilizando un Kinect**

En el tercer experimento el usuario esta a una distancia de 90 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

Para el análisis de uno y dos Kinect, como en todos los casos se analizaron las mismas imágenes. Debido a la posición de la mano solo se detectaron 149 gestos de la clase uno y 0 de la clase dos, cuando se utiliza el Kinect frontal.

	Gesto 1	Gesto 2
Gesto 1	153	47
Gesto 2	26	174

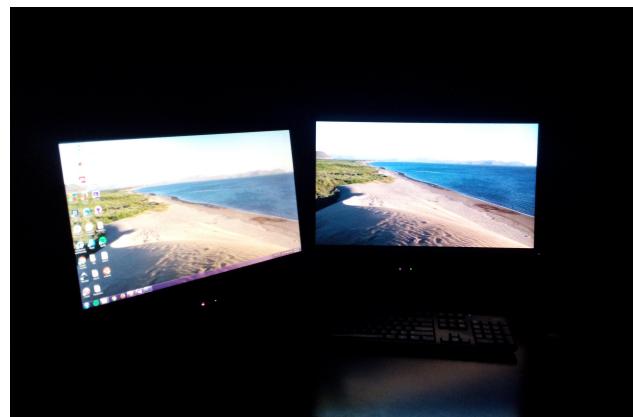
**Tabla 11: Matriz de confusión**

	Gesto 1	Gesto 2
Gesto 1	54	95
Gesto 2	0	0

**Tabla 12: Matriz de confusión, utilizando un Kinect**

### 5.1.3. Experimentos sin iluminación

Para este experimento las imágenes se capturaron en un laboratorio sin iluminación, como la que se muestra en la figura.

**Figura 38: Laboratorio en condiciones con baja iluminación.**

En el primer experimento el usuario esta a una distancia de 70 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	190	110
Gesto 2	196	4

**Tabla 13: Matriz de confusión**

En el segundo experimento el usuario esta a una distancia de 80 cm. del Kinect frontal.

	Gesto 1	Gesto 2
Gesto 1	154	46
Gesto 2	167	33

**Tabla 14: Matriz de confusión del reconocimiento de los dos gestos estáticos.**

Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	187	13
Gesto 2	90	110

**Tabla 15: Matriz de confusión**

	Gesto 1	Gesto 2
Gesto 1	162	37
Gesto 2	4	166

**Tabla 16: Matriz de confusión del reconocimiento de los dos gestos estáticos.**

	Gesto 1	Gesto 2
Gesto 1	187	13
Gesto 2	90	110

En el tercer experimento el usuario esta a una distancia de 90 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

	Gesto 1	Gesto 2
Gesto 1	150	50
Gesto 2	6	194

**Tabla 17: Matriz de confusión**

No se detecto ningún elemento de la clase dos y solo se detectaron 98 elementos de la primera clase. Debido que por la vista del kinect frontal no se pudo detectar la pose.

Por las matrices anteriores se observa que hay un error mayor cuando se quieras identificar el gesto uno, esto debido a las imágenes provenientes del sensor, ya que no

	Gesto 1	Gesto 2
Gesto 1	138	61
Gesto 2	0	0

**Tabla 18: Matriz de confusión**

siempre es posible obtener una imagen completa de la mano, ni utilizando los dos dispositivos.

## 5.2. Experimentos de gestos dinámicos

La evaluación del sistema en cuanto al reconocimiento de los gestos dinámicos se realizo de la siguiente forma; al igual que en los gestos estáticos se hicieron dos conjuntos de experimentos cada uno con diferentes tipos de iluminación, media y sin iluminación. En cada conjunto de experimentos se tomo en cuenta la distancia, ya que en cada grupo se analizaron tres distancias, 70, 80, 90 cm.

Los gestos analizados fueron dos. El primero corresponde a la palma de la mano con los dedos separados, Gesto 3, en movimiento y el segundo el puño de la mano, Gesto 4, también en movimiento. Estos gestos fueron realizados por tres usuarios distintos. Para cada experimento se analizaron 150 cuadros por lo que se analizaron 5 repeticiones del gesto y cada uno tenía una duración de 30 cuadros por segundo. Se tomo como gesto valido si el porcentaje de reconocimiento de cada gesto estático presente en cada segundo es mayor o igual a 80 %. 

Enseguida se presentan los resultados de cada experimento realizado, por usuario. También se presenta el rendimiento del sistema usando solo un solo Kinect.

### 5.2.1. Experimentos con iluminación media

Para el conjunto de estos experimentos, las imágenes se capturaron en un laboratorio con iluminación media, como la que se muestra en la figura.

En el primer experimento el usuario esta a una distancia de 70 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario, para las imágenes de los demás usuarios ver el apéndice. En la tabla se encuentran los resultados del reconocimiento de los dos gestos y por usuario.

Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect						
		R1	R2	R3	R4	R5
P1	G3	77	80	83	73	70
	G4	93	100	100	93	97
P2	G3	87	80	87	70	97
	G4	97	93	97	93	93
P3	G3	83	100	80	53	90
	G4	100	93	93	87	100

Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect						
		R1	R2	R3	R4	R5
P1	G3	73	63	77	73	63
	G4	100	100	100	100	100
P2	G3	100	93	87	93	87
	G4	93	77	77	67	60
P3	G3	100	100	93	97	93
	G4	73	90	83	63	83

En el segundo experimento el usuario esta a una distancia de 80 cm. del Kinect frontal.

Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect						
		R1	R2	R3	R4	R5
P1	G3	33	27	30	50	57
	G4	100	100	70	87	77
P2	G3	37	27	33	38	29
	G4	100	93	90	80	75
P3	G3	47	87	60	60	63
	G4	100	90	97	92	95

Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect						
		R1	R2	R3	R4	R5
P1	G3	83	86	67	87	70
	G4	35	53	70	77	57
P2	G3	33	57	57	73	60
	G4	93	93	90	90	80
P3	G3	93	100	67	100	80
	G4	100	93	87	83	90

En el tercer experimento el usuario esta a una distancia de 90 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect						
		R1	R2	R3	R4	R5
P1	G3	87	87	53	87	67
	G4	100	100	100	90	92
P2	G3	87	87	90	93	80
	G4	83	73	97	80	75
P3	G3	90	97	97	100	90
	G4	77	90	80	75	80

Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect						
		R1	R2	R3	R4	R5
P1	G3	30	37	13	13	30
	G4	100	98	95	100	98
P2	G3	50	60	67	70	75
	G4	77	78	75	80	70
P3	G3	77	73	80	97	90
	G4	77	90	91	80	75

### 5.2.2. Experimentos sin iluminación

Para este experimento las imágenes se capturaron en un laboratorio sin iluminación, como la que se muestra en la figura.

En el primer experimento el usuario esta a una distancia de 70 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario, para las imágenes de los demás usuarios ver el apéndice . En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

		Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect				
		R1	R2	R3	R4	R5
P1	G3	77	80	57	70	60
	G4	100	97	93	100	97
P2	G3	97	57	80	53	87
	G4	100	97	83	87	87
P3	G3	73	43	43	47	60
	G4	100	93	97	93	93

		Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect				
		R1	R2	R3	R4	R5
P1	G3	93	83	100	77	87
	G4	100	100	97	97	100
P2	G3	97	97	87	93	97
	G4	80	93	90	97	97
P3	G3	83	87	97	97	100
	G4	100	97	93	87	90

En el segundo experimento el usuario esta a una distancia de 80 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect						
		R1	R2	R3	R4	R5
P1	G3	57	63	60	63	63
	G4	100	100	100	100	100
P2	G3	40	43	60	70	75
	G4	100	93	97	93	98
P3	G3	80	87	67	77	63
	G4	97	87	100	90	87

Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect						
		R1	R2	R3	R4	R5
P1	G3	63	70	90	80	77
	G4	87	100	93	90	93
P2	G3	60	73	47	63	73
	G4	100	97	87	90	90
P3	G3	87	100	73	83	80
	G4	73	73	93	83	93

En el tercer experimento el usuario esta a una distancia de 90 cm. del Kinect frontal. Se muestra un ejemplo de las imágenes obtenidas de un usuario. En la tabla se encuentran los resultados del reconocimiento de los dos gestos.

Porcentajes de reconocimiento en cada secuencia de 30 seg. 1 Kinect						
		R1	R2	R3	R4	R5
P1	G3	57	70	70	67	77
	G4					
P2	G3	70	80	80	77	80
	G4	73	80	83	97	80
P3	G3	70	83	80	100	97
	G4	87	97	90	70	90

Porcentajes de reconocimiento en cada secuencia de 30 seg. 2 Kinect						
		R1	R2	R3	R4	R5
P1	G3	47	47	53	37	40
	G4					
P2	G3	37	63	63	75	80
	G4	100	98	95	90	92
P3	G3	80	97	87	83	87
	G4	87	100	97	78	95

En las tablas anteriores el gesto cuatro no pudo ser ubicado para el participante uno.

Como se vio en cada tabla se analizo el porcentaje de gestos localizados en cada secuencia de 30, analizando cuando se tenia uno y dos dispositivos. En general si se observa cada elemento o gesto, se puede observar que el sistema tienen un mayor grado de reconocimiento cuando se utilizan los dos dispositivos.

También se observa que en algunas ocasiones el reconocimiento es bajo, usualmente para el gesto número tres, esto debido a la resolución de sensor, ya que en ocasiones las regiones de los dedos no son captadas y esto hace que reconozca erróneamente el gesto.

La secuencia de imágenes de cada experimento fue procesa como provenía del sensor, no se escogieron las mejores imágenes. Los resultados prometen que con una mejor resolución del sensor el grado de reconocimiento seria mayor



# Capítulo 6. Conclusiones

---

El objetivo del trabajo era reconocer gestos con las manos bajo distintas circunstancias.

Se decidió como dispositivo de captura de las imágenes dos sensores Kinect, esto para sobrellevar la oclusión de los dedos. Para reconocer los gestos se analizó la información proveniente de cada uno de estos sensores por separado y cuadro por cuadro, sin necesidad de calibrar los dispositivos. Por lo que la detección y extracción de características se realizó por separado, la cual se funcionó en un vector de características para así poder ser entrenado mediante SVM.

Los resultados sugieren que el sistema propuesto cumple con su objetivo pues funciona bajo distintas circunstancias y se justifica el uso de los dos dispositivos gracias a la mejora de la precisión en comparación cuando se utiliza solo un dispositivo.

## 6.1. Limitaciones del sistema

~~Esta investigación utiliza dos dispositivos Kinect como medio de entrada del sistema. De manera que las limitaciones del sistema propuesto están dadas por las características de dicho dispositivo, tales como la distancia a la que se encuentra el dispositivo con el usuario, 0.4m. a 2m., la resolución de las imágenes a color 640 x 480 pixeles y la resolución del sensor infrarrojo 640 x 480 pixeles y el ruido proveniente de ambos dispositivos.~~

~~Otra limitante del sistema corresponde a las manos y por ende a los gestos que reconoce el sistema. Por ejemplo solo está programado para detectar una mano y esta tiene que tener una ligera rotación en el eje vertical para que el sistema la detecte. El número de gestos que reconoce el sistema es limitados, pues solo reconoce dos gestos estáticos y dos dinámicos.~~

## 6.2. Aportaciones

Debido a la realización de este trabajo se lograron las siguientes aportaciones, aparte del sistema creado:

- Creación de una base de datos de imágenes de profundidad, de la mano y de distintos fondos.
- Creación de dos detectores usando el método desarrollado por Viola y Jones (2001). Uno detecta la palma de la mano con los dedos separados entre ellos. El segundo también detecta la pose anterior y dos poses más, la palma de la mano con los dedos juntos y el puño.
- Publicación de artículo y presentación de póster en el congreso SPIE Optics and Photonics 2015.

### **6.3. Trabajo futuro**

En la sección anterior se menciono que una limitante es la resolución del sensor, una opción sería probar con la nueva versión del sensor Kinect, ya que el dispositivo tiene mayor resolución y las imágenes provenientes del sensor contienen menos ruido en comparación con la versión anterior.

El sistema podría mejorarse y alcanzar un mayor grado de precisión, si se mejora la detección, la propuesta es entrenar nuevamente el clasificador; incrementando el número de imágenes de entrenamiento, que contengan distintas poses, para así tener un número mayor de gestos a reconocer.

Otro punto que se puede explorar es abordar de manera distinta el reconocimiento de los gestos dinámicos, una buena propuesta sería utilizar un modelo estadístico como el Modelo Oculto de Markov, el cual permitiría implementar gestos dinámicos más complejos.

## Lista de referencias bibliográficas

- Burges, C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, **2**(2): 121–167.
- Caputo, M., Denker, K., Dums, B., y Umlauf, G. (2012). 3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware. *Mensch & Computer 2012: interaktiv informiert – allgegenwärtig und allumfassend!?*, pp. 293–302.
- Chaki, N., Shaikh, S. H., y Saeed, K. (2014). *Exploring Image Binarization Techniques*. Springer, primera edición. p. 90.
- Chang, C.-C. y Lin, C.-J. (2011). Libsvm. *ACM Transactions on Intelligent Systems and Technology*, **2**(3): 1–27.
- Chih-Wei Hsu, Chih-Chung Chang y Lin, C.-J. (2008). A Practical Guide to Support Vector Classification. *BJU international*, **101**(1): 1396–400.
- Cortes, C. y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**(3): 273–297.
- Freund, Y. y Schapire, R. (1995). A desicion-theoretic generalization of on-line learning and an application to boosting. *Computational learning theory*, **55**(1): 119–139.
- Gonzalez, R. y Woods, R. (2002). *Digital image processing*. p. 190.
- Hasan, M. M. y Mishra, P. K. (2012). Hand Gesture Modeling and Recognition using Geometric Features : A Review. *3*(1).
- Huang, D.-Y., Hu, W.-C., y Chang, S.-H. (2011). Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Systems with Applications*, **38**(5): 6031–6042.
- Hummel, S., Häfner, V., Häfner, P., y Ovtcharova, J. (2014). New Techniques for Hand Pose Estimation Based on Kinect Depth Data.
- Ibraheem, N. A. (2013). Comparative Study of Skin Color based Segmentation Techniques. *5*(10): 24–38.
- Jana, A. (2013). *Kinect for Windows SDK - Programming Guide - Face Tracking*. Packt, primera edición. p. 392.
- Kang, C., Bernhard, P., Kim, S., Srinivasa, P., y Satti, R. (2013). A Framework for Hand Gesture Recognition with Machine Learning Techniques.
- Kathuria, P. (2011). Hand Gesture Recognition. *2012*(25): 63–69.
- Mitra, S., Member, S., y Acharya, T. (2007). Gesture Recognition : A Survey. *37*(3): 311–324.
- Mohd Asaari, M. S., Rosdi, B. A., y Suandi, S. A. (2014). Adaptive Kalman Filter Incorporated Eigenhand (AKFIE) for real-time hand tracking system. *Multimedia Tools and Applications*.

- Murthy, G. R. S. y Jadon, R. S. (2009). A REVIEW OF VISION BASED HAND GESTURES RECOGNITION. **2**(2): 405–410.
- Nayakwadi, V. (2014). Natural Hand Gestures Recognition System for Intelligent HCI : A Survey. **3**(1): 10–19.
- Ong, K. C., Teh, H. C., y Tan, T. S. (1998). Resolving occlusion in image sequence made easy. *The Visual Computer*, **14**(4): 153–165.
- Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**(1): 62–66.
- Premaratne, P. (2013). *Human Computer Interaction Using Hand Gestures*. Springer, primera edición. p. 182.
- Rautaray, S. S. y Agrawal, A. (2012). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*.
- Sgouropoulos, K., Stergiopoulou, E., y Papamarkos, N. (2014). A Dynamic Gesture and Posture Recognition System. *Journal of Intelligent & Robotic Systems*, **76**(2): 283–296.
- Shin, S. (2013). *Emgu CV Essentials*. Packt Publishing, primera edición. p. 118.
- Silva, C. y Santos-Victor, J. (2001). Motion from occlusions. *Robotics and Autonomous Systems*, **35**(3-4): 153–162.
- Smith, S. W. (1999). *Digital signal processing*. p. 688.
- Viola, P. y Jones, M. (2001). Rapid object detection using a boosted cascade of simple features. *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR 2001*, **1**.
- Weichert, F., Bachmann, D., Rudak, B., y Fisseler, D. (2013). Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors*, **13**(5): 6380–6393.
- Ye, M., Zhang, Q., Wang, L., y Zhu, J. (????). A Survey on Human Motion Analysis. pp. 149–187.
- Yilmaz, A., Omar, J., y Mubarak, S. (2006). Object tracking: a survey. *ACM Computing Surveys (CSUR)*, **38**(4): 45.
- Yoon, J. W., Yang, S. I., y Cho, S. B. (2012). Adaptive mixture-of-experts models for data glove interface with multiple users. *Expert Systems with Applications*, **39**(5): 4898–4907.