**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

# Tabla de contenido (continuación)

# Lista de figuras

Figura Página

# Lista de figuras (continuación)

# Lista de tablas

Tabla Página

# 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 empezarón 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 vídeos, que son obtenidos mediante diversos tipos de cámaras.

Los métodos que utilizan imágenes o vídeo 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 acabo 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, 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 2 , 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.

**Figura 1: Dispositivos basados en contacto: en la parte izquierda de la imagen se observan los guantes de datos [1], en el centro los guantes de colores [2] y a la derecha se encuentra el dispositivo wii [3]**


**Figura 2: Dispositivos basados en visión: en la imagen se observan distintos tipos de cámaras en la parte izquierda de la imagen se observa una web [4], en el centro una digital [5] y a la derecha de la imagen una TOF [6].**

**Figura 3: 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

[6]http://www.technologyreview.com/article/414021/open-source-data-glove/
[6]http://www.digitaltrends.com/computing/the-gloves-that-could-change-the-world/
[6]https://www.nintendo.es/Wii/Wii-94559.html
[6]http://es.ccm.net/download/descargar-2562-driver-de-microsoft-lifecam-vx-3000
[6]http://www.canon.com.mx/ficha.aspx?id=722
[6]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 manosutilizaba 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 [7], el cual reconoce gestos dinámi-

[7] 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 empatamiento 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 [8], MYO [9], y software como: Flutter [10], 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 anteriomente.

[8] https://www.leapmotion.com/
[9] https://www.myo.com/
[10] https://flutterapp.com/

El dispositivo Leap Motion, fig 4 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 $60 \times 60 \times 60\ cm.$ y con la información capturada se construye un modelo 3D de las manos (Weichert *et al.*, 2013).

**Figura 4: 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.**

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 [11] que se utilizan tres cámaras infrarrojas, con la imágenes obtenidas con 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. [12]

No se cuenta con la informacion detallada del funcionamiento de MYO, lo único que se conoce es que el reconocimiento consta de tres etapas [13]. La primera es la adquisición de

[11] http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/
[12] http://www.digitaltrends.com/pc-accessory-reviews/myo-gesture-control-armband-review/
[13] 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 5: Ejemplo del reconocimiento del gesto usando MYO, controlando el volumen de la computadora. El dispositivo es el aparece en el brazo del sujeto.**

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 use del dispositivo requiere uso de manga corta; otra desventaja es que tiene una cantidad considerable de falsos positivos. [14]

Enseguida se explica el software de reconocimiento de gestos estáticos Flutter, fig:6 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.

[14] http://myogroupfive.blogspot.mx/2013/11/benefits-disadvantages-for-business$_2 4.html$

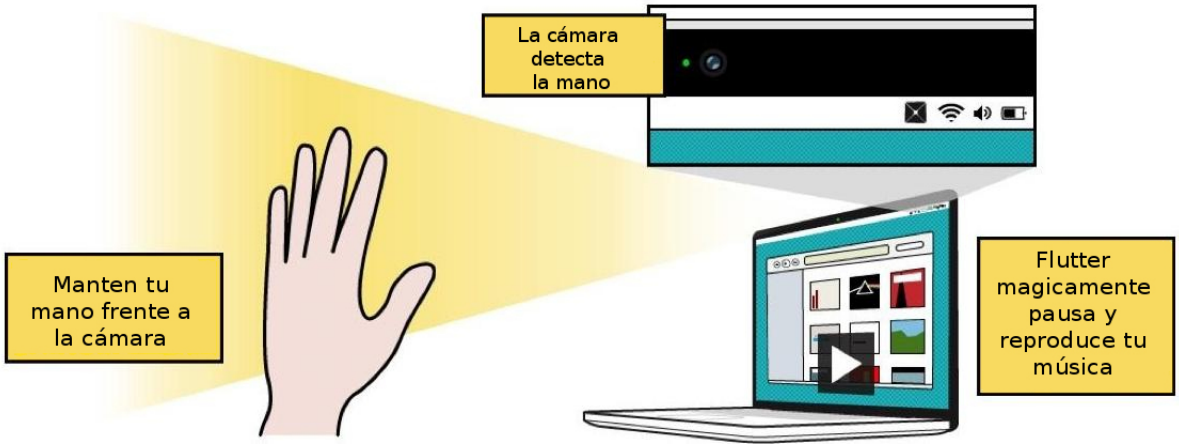



**Figura 6: La imagen anterior representa el funcionamiento del software Flutter**

Flutter permite controlar aplicaciones multimedia como: YouTube [15] ,VLC [16], Spotify [17], Netflix [18]. 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 oclusiones 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 utiliza 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.

[15] https://www.youtube.com/
[16] http://www.videolan.org/vlc/
[17] https://www.spotify.com/
[18] 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 modelos 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 7 muestra las etapas del reconocimiento.

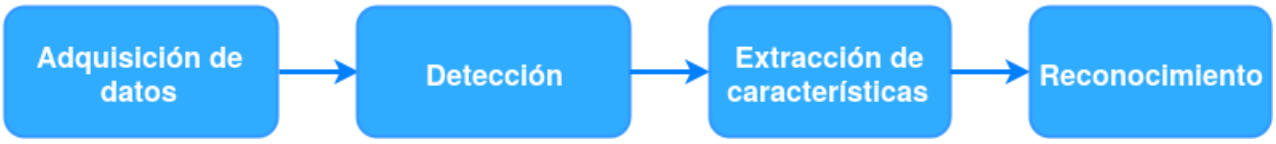



**Figura 7: El diagrama ejemplifica el procedimiento del reconocimiento de gestos.**

El proceso de reconocimiento varia un poco dependiendo del tipo de gesto, si es estático o dinámico. Por ejemplo el diagrama 7 ejemplifica perfectamente el proceso de reconocimiento de un gestos estático, para los gestos dinámicos se necesita una fase extra, el seguimiento el cual se realizada 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 representado 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 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 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.

#### 2.2.1.3. Extracción de características y seguimiento

La extracción de características

Consiste en localizar la mano en cada cuadro (imagen). Se lleva acabo 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 la clasificación del gesto, la etapa final del reconocimiento, la clasificación se puede hacer dependiendo del gesto. Para gestos estáticos basta con usar algún clasificador o empatar el gesto con una plantilla. En los dinámicos se requiere otro tipo de algoritmos de aprendizaje de máquina. A continuación se encuentran los principales métodos para llevar acabo el reconocimiento del gestos.

- K-medias: 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

- K-vecinos cercanos (KNN, por sus siglas en inglés): Este es un método para clasificar objetos basado en las muestras de entrenamiento en el espacio de características.
- 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 (SVM, por sus siglas en inglés). 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 (HMM, por sus siglas en inglés) es definido como un conjunto de estados donde un estado es el 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 especifica 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 8, y los indices de esta indican la posición, (Gonzalez y Woods, 2002).

## 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


**Figura 8: Representación de un imagen digital. Recuperada de (Shin, 2013)**

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 de reconocimiento, junto con los métodos o herramientas que son utilizados en cada una de ellas.

El sistema de reconocimiento de gestos propuesto consta de cuatro etapas principales 9. La primera etapa es la adquisición de los datos, en la cual se capturan las imágenes de entrada del sistema; la segunda etapa 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.

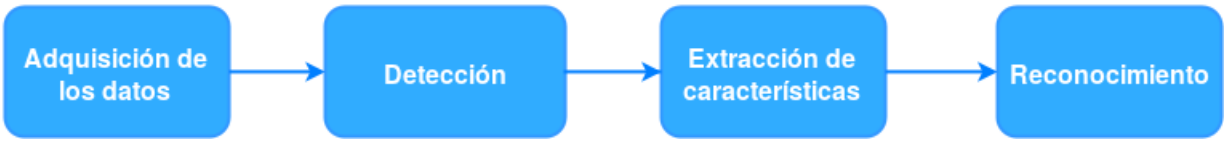



**Figura 9: Metodología del sistema propuesto.**

## 3.1. Adquisición de los datos

Es la primera etapa del sistema, donde se capturan los datos que son la entrada del sistema. Los datos provienen de los sensores de profundidad de dos dispositivos Kinect. A continuación se describe las características de este dispositivo.

En noviembre del 2010 la compañia Microsoft lanzó el sensor Kinect para consolas de vídeo juego Xbox 360 y en febrero del 2011 lanzó la versión para Windows, que se muestra en la figura 10.

El dispositivo Kinect esta equipado con una serie de sensores que permiten obtener imágenes a color y de profundidad (imágenes que indican a la distancia que esta un objeto del sensor), los cuales permiten hacer detección y seguimiento de personas. Detecta $6$ personas y hace el seguimiento de $2$ personas [1].

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 11.

[1] https://msdn.microsoft.com/en-us/library/hh973074.aspx


**Figura 10: Sensor Kinect para Windows**

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

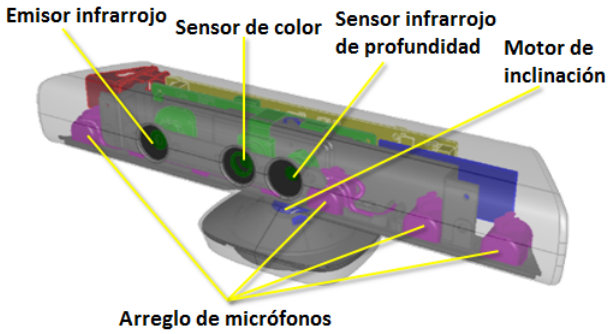



**Figura 11: Componentes del sensor Kinect**

- 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 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 $1280\,x\,960$ píxeles. La velocidad de los cuadros por segundo varia según la resolución de la imagen.

- El emisor infrarrojo proyecta puntos de luz infrarroja frente al sensor, con estos puntos y el sensor de profundidad se puede medir la profundidad que existe del sensor.

- El sensor infrarrojo lee los puntos infrarrojos proyectados 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 $640\,x\,480$ 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 $-27°$ a $+27°$.

- Arreglo de micrófonos, consta de $4$ micrófonos, captura el sonido y localiza la dirección en la que proviene.
- LED indica el estado del sensor.

## 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 lleve acabo de la siguiente manera, figura 12 el primer paso es localizar la mano, se lleva acabo usando el método de detección rápida de objetos; el siguiente paso es mejorar la imagen de la mano aplicando operaciones morfológicas y finalmente se la imagen que contiene la mano es binarizada.

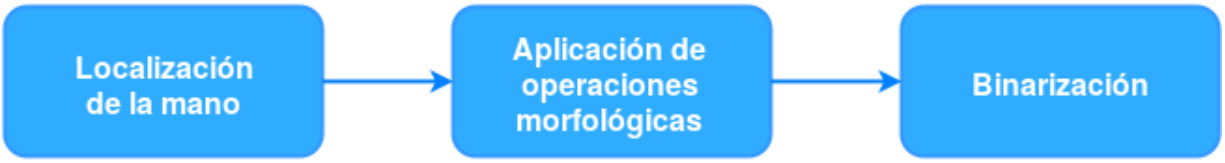



**Figura 12: Proceso de detección de la mano.**

### 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 usando 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 13 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 es utilizando Adaboost en forma de cascada.

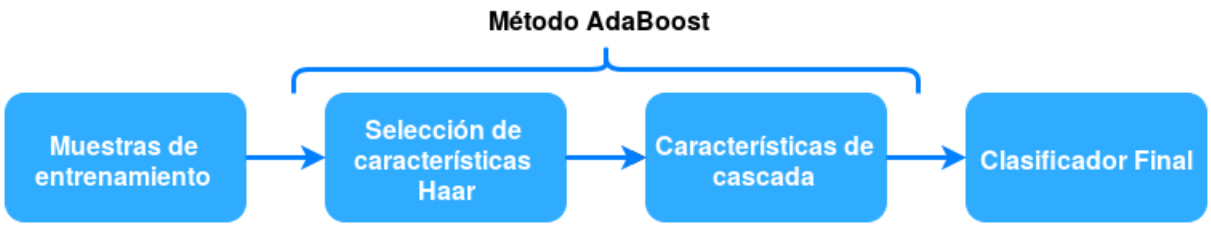



**Figura 13: Procedimiento del algoritmo de detección**

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

#### 3.2.1.1. Características Haar

Las características Haar, son operadores rectangulares como los que se muestran en la figura 14. A continuación se explicaran los operadores Haar básicos:

- Las características con dos rectángulos 14(a), 14(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 14(c), contienen tres regiones rectangulares adyacentes, y el valor de la característica se calcula la suma de las regiones exteriores y se resta la suma de la región interior.
- Las características con cuatro rectángulos 14(d), contienen cuatro regiones rectangulares adyacentes, y el valor de la característica se obtiene con la diferencia entre las regiones pares diagonales.

#### 3.2.1.2. Imagen integral

Uno de los aportes del método desarrolla 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 el

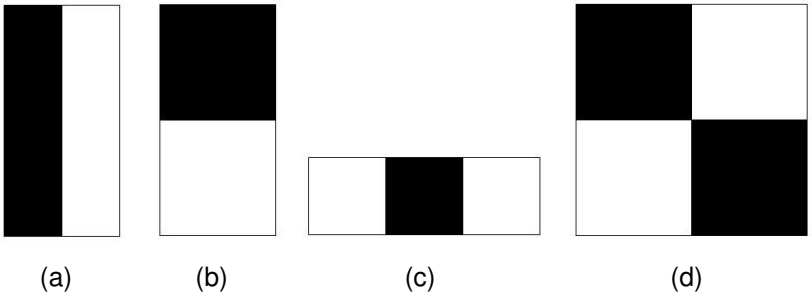



**Figura 14: Ejemplo de operadores Haar**

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 quiere hacer el cálculo. Lo anterior se puede escribir como: [2]

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

La figura 15 muestra un ejemplo donde se calcula la imagen intregral, fig. 15(b), de la imagen original 15(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 15: 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: [3]

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

donde $REG(\alpha)$ es la región a la cual se quiere calcular el valor de la suma de sus pixeles;

[2] https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/
[3] https://computersciencesource.wordpress.com/2010/09/03/computer-vision-the-integral-image/

$A, B, C, D$ son las esquinas de dicha región, como se muestra en la figura 16, la region $\alpha$ se encuentra resaltada en color azul.

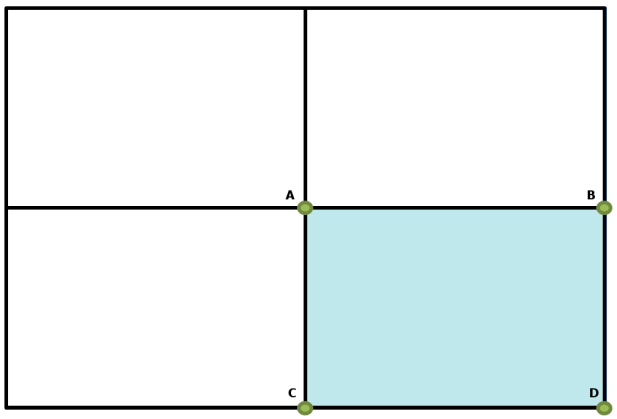



**Figura 16: Regiones de la imagen integral**

#### 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.

El algoritmo AdaBoost realiza su clasificación construyendo un clasificador fuerte, $h(x)$, llamado así debido a que tiene una precision mayor que los otros clasificadores con los que es construido, clasificadores débiles, $h_i(x)$. Los clasificares débiles son calculados de la siguiente manera:

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

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:

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

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 de AdaBoost:

---

**Algoritmo 1**

---

**Entrada:** El conjunto $\{(x_1, y_1), \cdots, (x_n, y_n)\}$ 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 $\beta_t = \frac{\epsilon_T}{1-\epsilon_t}$, $e_i = 0$ si $x_i$ es clasificado correctamente de otra forma $e_i = 1$.

8: **fin para**

9: El clasificador final o clasificador fuerte es:

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

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

---

#### 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. Lo cual se realiza seleccionando las características relevantes que se evalúan primero. Esta selección se realiza como lo muestra el algoritmo 2 cumpliendo cierto valor en la precision de la detección y del número de falsos positivos. El clasificador en cascada 17 esta compuesto

por etapas cada una de estas es un clasificador fuerte que es entrenado por medio de AdaBoost.

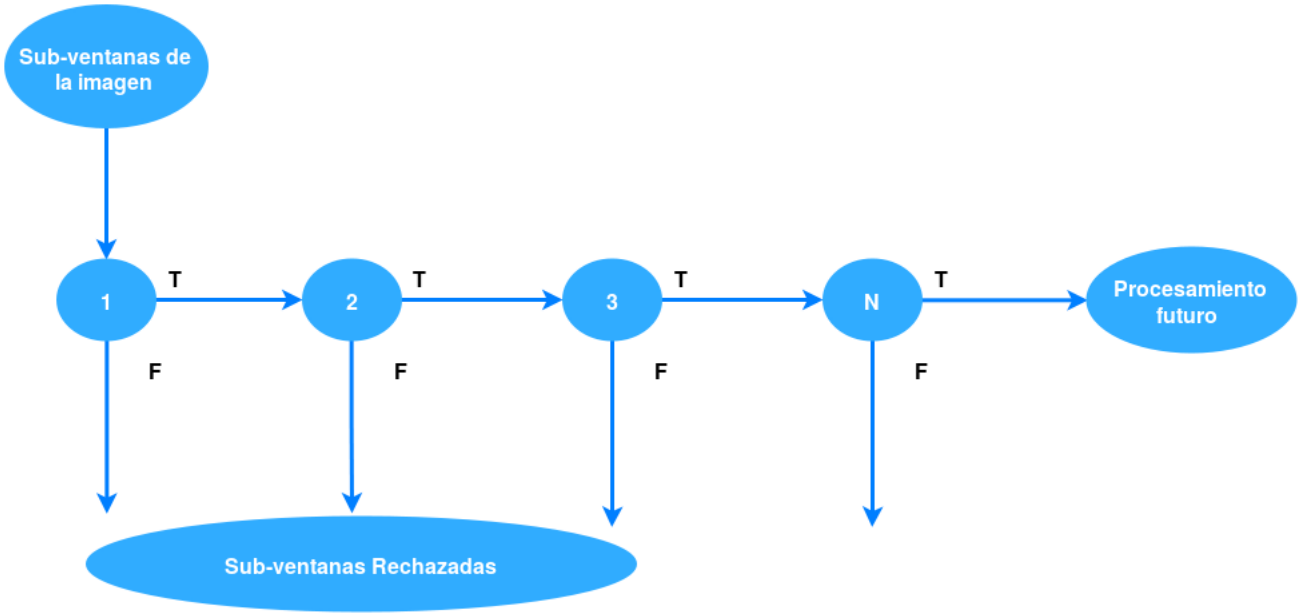



**Figura 17: Clasificador en forma de cascada**

### 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, umbral, de la imagen en escala de grises es seleccionado. Ya que se tiene el umbral, $T$, los pixeles de la imagen son discriminados dependiendo si su valor es mayor o igual al umbral entonces el valor de los pixeles en la imagen binaria es $1$ el resto toma valor de $0$. Es decir:

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

Existen diversas técnicas para binarizar una imagen, estas se pueden clasificar en dos grupos dependiendo de la manera es que es calcula el umbral, global o local. Los métodos globales calculan un umbral que es usado en toda la imagen y los métodos locales calculan varios umbrales para ciertas regiones de la imagen (Chaki *et al.*, 2014).

Un método de binarización muy utilizado es el de NiBlack,(Niblack, 1985) es un méto-

**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 de la detección por etapa, .
**Salida:** El clasificador en forma de cascada.

$F_0 = 1$, $D_0 = 1$
$i = 0$
**mientras** $F_i > F_{Tarjet}$ **hacer**
  $i = i + 1$
  $n_i = 0$, $F_i = F_{i-1}$
  **mientras** $F_I > F \times fp_{i-1}$ **hacer**
    $n_i = n_i + 1$
    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.
    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**
  $N = 0$
  **si** $F_i > F_{Tarjet}$ **entonces**
    Evaluar el actual clasificador en cascada en el conjunto de imágenes negativas y poner cualquier detección falsa en el conjunto $N$.
  **fin si**
**fin mientras**

do local y adaptativo ya que adapta el umbral basándose en la media $m(i,j) =$ y la desviación estándar $\sigma(i,j)$ de una ventana deslizante de tamaño $bxb$. El umbral $T$ se calcula como:

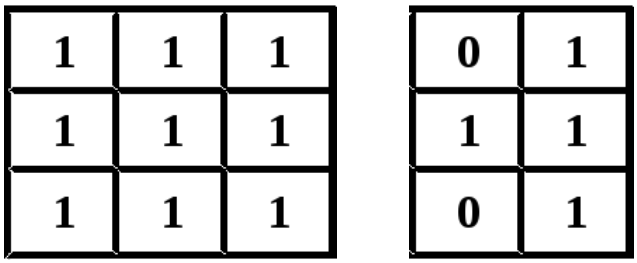
$$T(i,j) = m(i,j) + k \cdot \sigma(i,j)$$

donde $k \in [0,1]$ el valor de la constante determina que tanta parte del contorno es preservado (Chaki *et al.*, 2014).

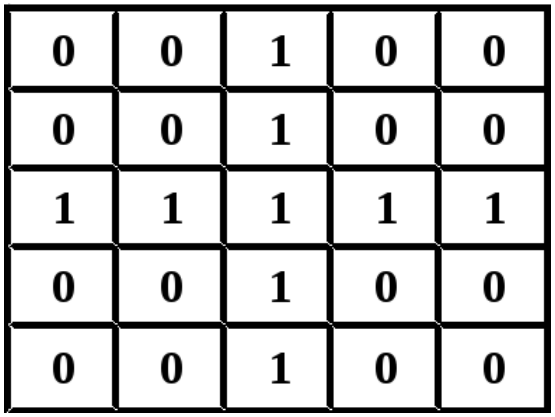
### 3.2.3. Operaciones Morfológicas

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 se removido tomando en cuanta la forma y estructura de la imagen. La operaciones morfológicas utilizan un elemento estructural el cual se aplica por toda la imagen, los elementos estructurales pueden ser de distintas formas como 18

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




(a) Rectángulo de $3x3$. (b) Figura de $3x2$.




(c) Cruz de $5x5$.

**Figura 18: Ejemplos de elementos estructurales**

#### 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. 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. La erosión se define como:

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

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

(a) original (b) dilatación (c) erosión

(d) apertura (e) cierre

**Figura 19: Aplicación de operaciones morfológicas (citar)**

#### 3.2.3.3. Apertura

La operación apertura abre huecos entre objetos conectados por un enlace delgado de pixeles.

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

#### 3.2.3.4. Cierre

La operación cierre elimina huecos pequeños y rellena huecos en las

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

### 3.3. Extracción de características

La idea de esta etapa es extraer 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: el primer paso es encontrar la envolvente convexa de la mano para pos-

teriormente calcular los defectos de convexidad una vez realizado lo anterior se pueden calcular el numero de dedos de la mano entre otras características; finalmente las características calculadas se guardan en un vector.

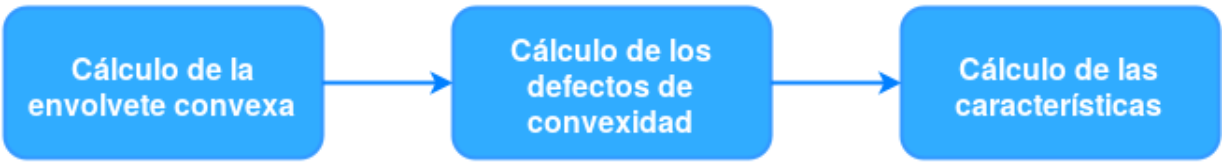
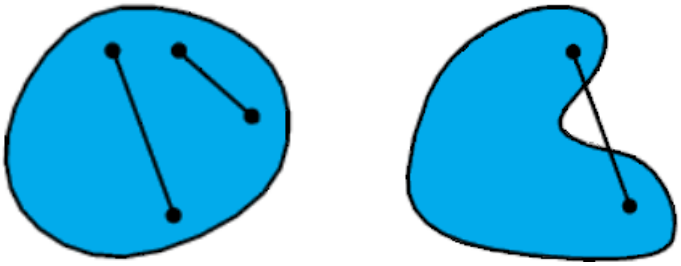



**Figura 20: Proceso de la extracción de características.**

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

Sea $A$ un conjunto en el espacio euclidiano $\mathbf{R}^d$, $A$ es un conjunto convexo [4] si contiene todos los segmentos de línea que unen a cualquier par de puntos pertenecientes al conjunto. Donde $d$ es la dimensión del espacio euclidiano.



(a) Conexo (b) convexo

**Figura 21: Conjunto conexo y convexo (citar)**

Sea $B$ un conjunto de puntos en el plano Euclidiano, la envolvente convexa de $B$ es el conjunto convexo más pequeño que contiene a todos los puntos en $B$. En la imagen 22 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 $CD = \{cd_1, cd_2, \cdots, cd_n\}$ el conjunto de defectos de convexidad de una envolvente convexa. Cada defecto esta compuesto por tres elementos: el punto de inicio del defecto

[4] Weisstein, Eric W. Çonvex.”From MathWorld–A Wolfram Web Resource. http://mathworld.wolfram.com/Convex.html

$s_i(x, y)$; 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 22 los puntos amarillos representan los puntos de profundidad de los defectos de convexidad.

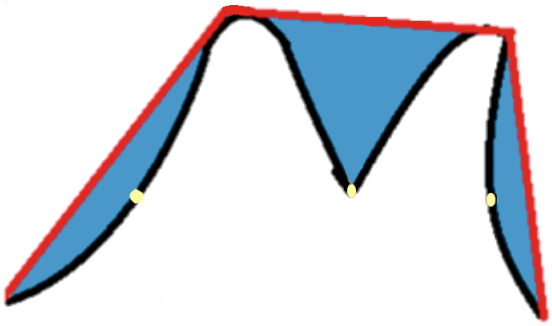


**Figura 22: En la imagen se aprecia de color rojo la envolvente convexa, de negro el contorno de la figura, y los puntos amarillos son el punto de profundidad de los defectos de convexidad.**

Usando las técnicas anteriores podemos extraer características importantes como el numero de dedos, la posición del centro de la mano, el ángulo que existe entre cada dedo.

El número de dedos que se encuentran levantados es calculado con el algoritmo 3, **referenciar**, que utiliza los defectos de convexidad en especifico los conjuntos de puntos de inicio, $\mathcal{S} = \{s_1(x, y), s_2(x, y), \cdots, s_n(x, y)\}$ y el de mayor distancia, $\mathcal{D} = \{d_1(x, y), d_2(x, y), \cdots, d_n(x, y)\}$ donde $n$ es el número total de defectos de la envolvente convexa.

El ángulo $\alpha$ entre los dedos $Nf_j$ y $Nf_{j+1}$ es calculado como:

$$\alpha_{f_j} = \tan^{-1} \left| \frac{m_{j+1} - m_j}{1 + m_{j+1} m_j} \right|$$

donde $m_{j+1}, m_j$ son las pendientes de las rectas que pasa por los puntos $d_j(x, y)$ y $s_{j+1}(x, y)$; $d_j(x, y)$ y $s_j(x, y)$; $j = \{1, 2, 3, 4, 5\}$.

El ángulo del centro de la mano a la punta de los dedos puede ser obtenido como:

$$\theta_{Nf_j} = \tan^{-1} |m_j - 90°|$$

donde $m_j$ es la pendiente de la recta que pasa por el centro de la mano y la punta de los dedos $s_i(x, y)$.

Una vez que todas las características son calculas están se guardan en un vector,

**Algoritmo 3** Calcula el número de dedos de la mano.

**Entrada:** Los conjuntos $\mathcal{S}$, $\mathcal{D}$.
**Salida:** Número de dedos, $Nf$.

**para** $i = 1$ hasta $n$ **hacer**
  $minDist = 20$, $maxAng = 60$, $antecesor = 0$, $sucesor = 0$.
  **si** $d_i(x, y) < minDepth$ **entonces**
    **continuar**
  **fin si**
  **si** i=0 **entonces**
    $antecesor = n - 1$
  **si no**
    $antecesor = i - 1$
  **fin si**
  **si** i=n-1 **entonces**
    $sucesor = 0$
  **si no**
    $sucesor = i + 1$
  **fin si**
  Calcular el ángulo entre $s_{antecesor}(x, y)$, $d_i$ $s_{sucesor}(x, y)$
  **si** ángulo $\geqslant maxAng$ **entonces**
    **continuar**
  **fin si**
  $Nf = Nf + 1$.
**fin para**

llamado vector de características. La dimension del vector es el numero 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 con el método de máquinas de soporte vectorial (SVM, por sus siglas en ingles, support vector machine), un método de aprendizaje de máquina supervisado que es utilizado para resolver problemas de clasificación y regresión. El cual tiene como objetivo crear un modelo basado en datos conocimos (datos de entrenamiento), que predice a que clase pertenecen datos nuevos.

SVM realiza la clasificación separando las clases buscando el hiperplano que tengan el margen de separación más grande. Enseguida se explica a detalle el funcionamiento del método.

Dado $N$ puntos de entrenamiento $x_i$, de dimensión $D$, dos clases distintas $y_i = -1$ o $+1$ es decir:

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

Hiperplano óptimo

$$w \cdot x + b = 0$$

donde $w$ es la normal al hiperplano, $\frac{b}{|w|}$ es la distancia perpendicular desde el hiperplano al origen.

$$w \cdot x + b = +1 \text{ para } y_i = +1$$

$$w \cdot x + b = -1 \text{ para } y_i = -1$$

Maximizar el margen, encontrar el mínimo de $w$.

$$Min\|w\| \quad \text{tal que} \quad y_i(w \cdot x_i + b) - 1 \geq 0$$

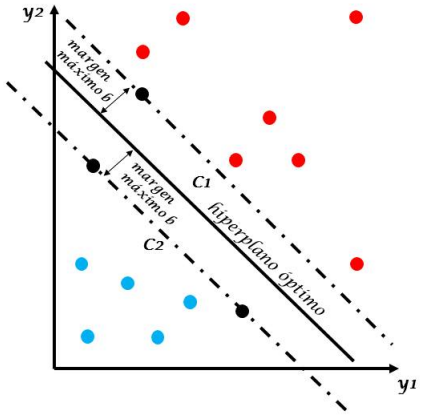



**Figura 23: Clasificación de maquina de soporte usando kernel lineal**

# 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 y otro al lado izquierdo, con una distancia de $74$ y $79$ $cm.$ respectivamente; y entre ellos de $46$ $cm.$ como se muestra en la figura 24.

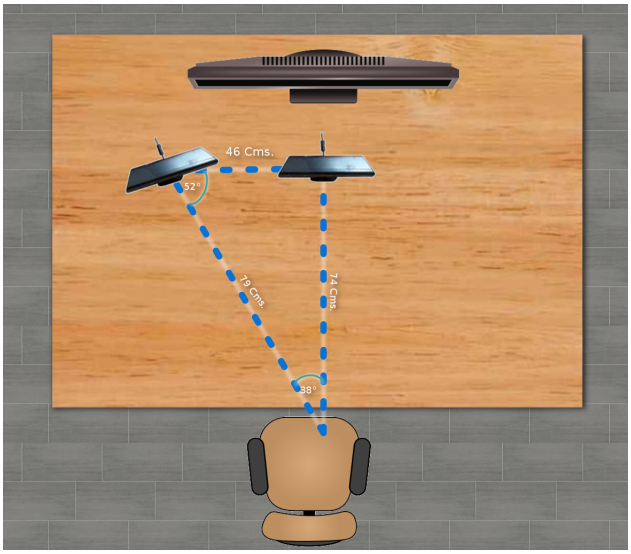



**Figura 24: Configuración del sistema de reconocimiento de gestos**

Una vez que le 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. 25

Debido a la naturaleza del funcionamiento del Kinect, las imágenes obtenidas de ambos sensores contiene ruido del tipo (poner tipo) (referencia), de manera que la imágenes

**Figura 25: Representación de los datos capturados por los Kinect**

provenientes de los sensores son como la figura 26; el ruido es reducido usando filtros de mediana 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 26.

Como se aprecia en la imagen siguiente gran parte del ruido es reducido obteniendo

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

una mejor imagen, pero desafortunadamente todavía sigue habiendo ruido en la imagen, este puede ser eliminado casi en su mayoría si el tamaño de la ventana aumenta pero se pierde información de la imagen de ,manera que se decidió optar por ese tamaño de ventana; la perdida de información será subsanada posteriormente. En la imagen también se aprecia el fondo negro, esto es debido a que se discrimino 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 [1]. 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 $100$ imágenes de la mano usando el software Create Samples [2]. Todas las imágenes usadas fueron tomadas de nuestra base de datos [3].

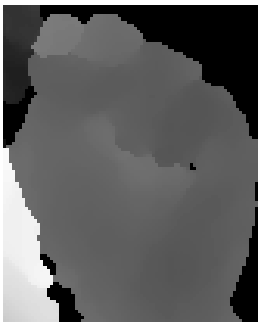
Nuestra base de datos contiene gran cantidad de imágenes de profundidad. Imágenes de fondo y de mano, estas fueron tomadas a una distancia de entre $60$ y $200$ $cm.$. Las imágenes de profundidad de la mano fueron tomadas de $6$ personas distintas con tres distintas poses: palma con los dedos separados 27(a), palma con dedos juntos 27(c) y finalmente el puño 27(b), como se muestran en la figura 27. Las imágenes de fondo fueron tomadas de distintos escenarios como se muestra en la figura 28. El programa para la captura de las imágenes puede ser encontrado en github [4].

Para localizar la mano en cada cuadro proveniente de los dispositivos Kinect, una ventana de tamaño $ka$ se desliza por la imagen, una vez que la mano se localiza la región de interés $ROI(x, y)$ es seleccionada alrededor de la mano, como se puede ver en la figura 29.

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 encontrando los contornos existentes en la ROI y se toma el contorno más grande como el contorno de la mano, antes de calcular el contorno se realizan una serie de procesamientos al ROI para eliminar ruido y que el cálculo del contorno sea más preciso.

[1] https://github.com/mrnugget/opencv-haar-classifier-training
[2] http://note.sonots.com/SciSoftware/haartraining.html
[3] https://github.com/americamm
[4] https://github.com/americamm

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



(b) Puño.

(c) Palma de la mano con los dedos juntos.

**Figura 27: Ejemplo de imágenes de poses de nuestra base de datos.**

**Figura 28: Imagen del fondo de nuestra base de datos**

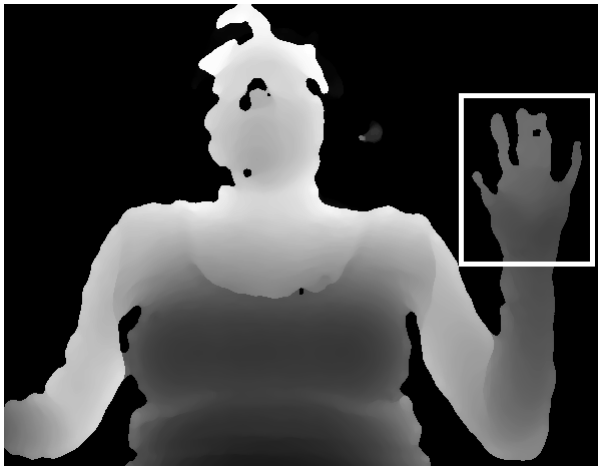


**Figura 29: Mano seleccionada**

El primer procesamiento que se aplica al ROI, son las operaciones morfológicas: apertura y cierre, en ese orden. Se utilizan para eliminar uniones pequeñas que existen en la imagen como el de la figura 30(a) o unir pequeños hoyos que existen en la imagen, como los que se encuentran en la figura 30(b).
Las operaciones anteriores utilizan un elemento estructural rectangular; para la operación de apertura el tamaño del elemento es de $3\,x\,7$ pixeles; para el cierre se aplicó con un tamaño $7\,x\,7$ pixeles.

(a) ROI donde existe una unión entre los dedos.

(b) ROI donde se aprecian hoyos en la mano.

**Figura 30: La imágenes anteriores muestras los distintos casos que aparece el ruido en la imagen aun después de haber aplicado un filtro de medianas**

Las imágenes siguientes muestran el resultado de aplicar las operaciones apertura y cierre al ROI.


(a)

(b)

**Figura 31: La imágenes muestran el resultado de aplicar las operaciones morfológicas de apertura y cierre.**

Una vez aplicadas las operaciones morfológicas el siguiente paso es binarizar la región de interés, se lleva acabo aplicando el algoritmo desarrollado por (Niblack, 1985), se decidió usar este método debido a la naturaleza de la imagen. Los parámetros que fueron usados fueron $k = 0.5$ y una ventana de $3x3$ píxeles. La figura siguiente es la imagen binarizada.


**Figura 32: Binarización de ROI**

Cuando la ROI es binarizada el siguiente paso es encontrar los contornos existentes dentro del ROI. Los contornos se calcularon utilizando el algoritmo de $blabla$ con tales parámetros. La fig. 33 muestra en color blabla el contorno de la mano.


**Figura 33: Contorno de la mano**

## 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.


**Figura 34: En esta dibujado la envolvente convexa, los punto en son los defectos de convexidad**

Una vez aplicados estos dos algoritmos se pueden calcular el número de dedos y las puntas de estos que son fundamentales para calcular las demás características. Enseguida se describe el algoritmo 3 para calcular el número de dedos.

Las características se guardan en un vector de características de dimensión $26$.

## 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 capitulo 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 [5] un wrapper de la librería LibSVM (Chang y Lin, 2011).

### 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. Para el entrenamiento se tomaron $num$ imágenes de los dos distintos gestos como los de la fig 35, divididas en partes iguales. De tamaño $640$ por $480$ pixeles.

[5] https://github.com/ccerhan/LibSVMsharp


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

(b) Puño.

**Figura 35: Ejemplo de imágenes de poses de nuestra base de datos.**

Para el entrenamiento de la máquina de soporte se utilizo un kernel exponencial, (poner los demás parámetros) y se utilizó validación cruzada con 5 pliegues.

### 4.4.2. Reconocimiento de gestos dinámicos

El sistema reconoce numero de gestos dinámicos.

# Capítulo 5. Resultados

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 2.410 [1] un wrapper de OpenCV[2].

Para probar la precisión del sistema se realizaron diversas pruebas con los distintos tipos de gestos y en diferentes circunstancias. En las secciones siguientes se explica cada experimento y resultados de estos.

## 5.1. Experimentos de gestos estáticos

## 5.2. Experimentos de gestos dinámicos

[1] http://www.emgu.com/wiki/index.php/Main_Page
[2] http://opencv.org/

# Capítulo 6. Conclusiones

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

## 6.1. Limitaciones del sistema

Esta investigación utiliza el dispositivo 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 encuentra el dispositivo con el usuario, $0.4m$ a $3m$ , la resolución de las imágenes a color $640\,x\,480$ pixeles y la resolución del sensor infrarrojo $640\,x\,480$ pixeles.
También el sistema depende de dos sensores Kinect, que se utilizarán en el caso que exista oclusión.
Otra limitante es el número de gestos que reconoce el sistema, pues solo reconoce dos gestos estáticos y (numero) dinámicos.

## 6.2. Trabajo futuro

Como se expreso en la sección anterior una limitante es la resolución del sensor, una opción seria probar con la nueva versión del sensor Kinect, pues debido a como obtiene los datos el sensor cuanta con mayor resolución y el ruido de las .

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 atacar de manera distintas el reconocimiento de los gestos dinámicos, utilizando por ejemplo un modelo estadístico como el Modelo Oculto de Markov, el cual permitiría implementar gestos más complejos.

# Lista de referencias bibliográficas


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.

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.

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.

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.

Niblack, W. (1985). *An introduction to digital image processing*. Strandberg Publishing Company Birkeroed, Denmark,. p. 215.

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.

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*.

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.

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.