

**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 Díaz 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

Resumen en español	ii
Resumen en inglés	iii
Dedicatoria	iv
Agradecimientos	v
Lista de figuras	viii
Lista de tablas	x
1. Introducción	1
1.1. Definición del problema	2
1.2. Justificación	2
1.3. Objetivo general	2
1.4. Objetivos específicos	3
1.5. Limitaciones y suposiciones	3
1.6. Reconocimiento de gestos con la manos	4
1.7. Estado del arte	4
1.7.1. Modelos de contacto	4
1.7.2. Modelos basados en la visión	5
1.7.3. Sistemas comerciales	7
2. Marco teórico	11
2.1. Gestos	11
2.2. Reconocimiento de gestos con la manos	11
2.2.1. Etapas del reconocimiento	12
2.2.1.1. Detección	12
2.2.1.2. Seguimiento	13
2.2.1.3. Reconocimiento	14
2.3. Imagen	15
2.4. Oclusión	15
3. Sistema de reconocimiento de gestos propuesto	17
3.1. Adquisición de los datos	17
3.2. Detección	19
3.2.1. Método detección rápida de objetos usando características simples utilizando el clasificador AdaBoost en forma de cascada	19
3.2.1.1. Características Haar	20
3.2.1.2. Imagen integral	20
3.2.1.3. Algoritmo AdaBoost	22
3.2.1.4. Clasificador AdaBoost en Cascada	23
3.2.2. Binarización	24
3.2.3. Operaciones Morfológicas	25
3.2.3.1. Dilatación	25
3.2.3.2. Erosión	26
3.2.3.3. Apertura	26
3.2.3.4. Cierre	27

Tabla de contenido (continuación)

3.3.	Extracción de características	27
3.4.	Reconocimiento	29
4.	Implementación del sistema de reconocimiento de gestos propuesto	31
4.1.	Adquisición de los datos	31
4.2.	Detección	33
4.3.	Extracción de características	36
4.4.	Reconocimiento	39
	4.4.1. Reconocimiento de gestos estáticos	39
	4.4.2. Reconocimiento de gestos dinámicos	40
5.	Resultados	41
5.1.	Experimentos de gestos estáticos	41
5.2.	Experimentos de gestos dinámicos	41
6.	Conclusiones	42
6.1.	Limitaciones del sistema	42
6.2.	Trabajo futuro	42
	Lista de referencias bibliográficas	43

Lista de figuras

Figura	Página
1. Ejemplo del reconocimiento del gesto usando Leap Motion, que es el dispositivo que se encuentra conectado a la Laptop.	8
2. Ejemplo del reconocimiento del gesto usando MYO, es el dispositivo que esta colocado en el brazo.	9
3. Ejemplo del reconocimiento del gesto usando , que es el dispositivo que se encuentra conectado a la Laptop.	10
4. Procedimiento del algoritmo de detección	12
5. Metodología del sistema propuesto.	17
6. Sensor Kinect para Windows	18
7. Componentes del sensor Kinect	18
8. Proceso de detección de la mano.	19
9. Procedimiento del algoritmo de detección	20
10. Ejemplo de operadores Haar	21
11. Ejemplo del cálculo de la imagen integral	21
12. Regiones de la imagen integral	22
13. Clasificador en forma de cascada	24
14. Ejemplos de elementos estructurales	26
15. Aplicación de operaciones morfológicas (citar)	27
16. Conjunto conexo y convexo (citar)	28
17. 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.	28
18. Clasificación de maquina de soporte usando kernel lineal	30
19. Configuración del sistema de reconocimiento de gestos	31
20. Representación de los datos capturados por los Kinect	32
21. Representación de los datos capturados por los Kinect	32
22. Ejemplo de imágenes de poses de nuestra base de datos.	34
23. Imagen del fondo de nuestra base de datos	34
24. Mano seleccionada	34
25. La imágenes anteriores muestras los distintos casos que aparece el ruido en la imagen aun después de haber aplicado un filtro de medianas	35
26. La imágenes muestran el resultado de aplicar las operaciones morfológicas de apertura y cierre.	35

Lista de figuras (continuación)

Figura	Página
27. Binarización de ROI	36
28. Contorno de la mano	36
29. En esta dibujado la envolvente convexa, los punto en son los defectos de convexidad	37
30. Ejemplo de imágenes de poses de nuestra base de datos.	39

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, generalmente, por no decir siempre, viene acompañada de gestos realizados con la cara, manos o cuerpo. Estos gestos sirven como complemento de la comunicación y 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 interactúa con la computadora. Uno de los objetivos principales de esta área es que la interacción se lleve a cabo 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 sensores como: guantes de datos, marcadores de colores y acelerómetros; los cuales se colocaban en la o las manos para poder capturar la posición, la pose de la o las manos, entre otros datos para poder reconocer el gesto realizado. Las técnicas desarrolladas posteriormente obtienen la información necesaria de la mano 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 ya que el usuario y la computadora interactúan de manera natural, el detalle 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 tomando en cuenta todas las variables.

Aunque existe gran variedad de métodos y sistemas que hacen el reconocimiento de gestos de las manos no existe alguno que su 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 el de controlar la computadora mediante gestos, con la ayuda del sensor Kinect como herramienta para la captura de la información de entrada.

1.1. Definición del problema

Existen diversas técnicas que logran obtener buena precisión en el reconocimiento de gestos realizados con las manos, pero hay técnicas que puedan tener buena precisión y que se adecue a todo tipo de situaciones de 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 en el caso de los gestos dinámicos 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 y tiempo de respuesta, en circunstancias de baja iluminación y oclusión. En el análisis del sistema se usará información real.
- Comparar los modelos propuestos con los existentes, en base al tiempo de respuesta y la eficiencia en cuanto al reconocimiento del gesto.

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.

Se supone el área de trabajo como un cuarto estándar con buena iluminación.

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

Los primeros acercamientos para llevar acabo el reconocimiento de gestos 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, esto 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 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.

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

1.7. Estado del arte

La sección anterior explica los modelos para llevar acabo el reconocimiento de gestos, enseguida se presentan los trabajos relevantes de cada uno de estos enfoques.

1.7.1. Modelos de contacto

En la sección anterior se presentó la definición de este modelo, a pesar que este modelo no es tan utilizado existen diversos trabajos que se basan en este enfoque, 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 modelo de mezclas adaptativo, usando un guante de datos, la principal limitante para este sistema es que solo reconoce gestos estáticos.

Aunque estos sistemas nos evitan algunos problemas que son consecuencia de los modelos basados en la visión, nos son perfectos, lo cual veremos enseguida.

Por otra parte uno de los dispositivos recientes es MYO ¹, aunque de este se hablará en la última parte de esta sección.

Como se describió en la sección anterior en los modelos de contacto la principal limitante es el uso de dispositivos en el cuerpo para el reconocimiento de los gestos, por esta razón la mayoría de los sistemas para el reconocimiento están enfocados en modelos basados en visión. Por lo que resulta natural que la investigación propuesta tome un enfoque basado en la visión.

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 nos pueden proporcionar gran cantidad de información que hace que el reconocimiento tenga mayor precisión. Enseguida se presentan tres trabajos relevantes los cuales utilizaron distintos tipos de cámaras y número de ellas.

En el trabajo propuesto por Premaratne (2013) realizan un modelo de reconocimiento de gestos estático y dinámico basados en el algoritmo de Lucas-Kanade. Las principales ventajas de este método son que es invariante a rotación, escala y al fondo. Aunque el modelo es afectado por los cambios en la iluminación.

Por otro lado (Huang *et al.*, 2011), propone un método para calcular gestos estáticos y dinámicos usando los filtros Gabor y haciendo una estimación del ángulo en el que se encuentra la mano. 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.

El método propuesto reconoce 11 gestos estáticos y dinámicos, la aportación del trabajo es en la parte de la segmentación de la mano que se lleva a cabo usando filtros Gabor. El sistema propuesto utiliza una cámara ccd para obtener la información de entrada. El sistema es robusto a la iluminación.

¹<https://www.thalmic.com/en/myo/>

Antes de hacer la segmentación de la (el brazo creo) se le aplica a la imagen un pre-procesamiento. El primer paso es aplicar un filtro de Gabor luego se escoge uno de los tres modelos del color; YCbCr, Gaussiano, Soriano, escogiendo el nivel de gris referente white[], reference White modificado[], mundo gris[], después se procede a calcular el ángulo de la mano mediante . Ya que se realizó el preprocesamiento el paso siguiente es segmentar la mano del antebrazo para esto se hace un barrido de la imagen por filas. Y se segmenta la imagen respecto a la distancia de arriba de la imagen a un valor que denota el número máximo de pixeles de un solo valor (el valor mayor el histograma).

Una vez realizada la segmentación el siguiente paso es obtener las características para el reconocimiento, las características son obtenidas mediante análisis de componentes principales y para encontrar la matriz de dispersión encuentra el eigenvector de la matriz de reducción.

La clasificación la hacen usando SVM aquí van comparando las imágenes de dos en dos, ya que SVM es un clasificador binario, por lo que se requieren $m(m-1)/2$ clasificaciones.

La precisión del reconocimiento varía dependiendo de si son imágenes reales o imágenes que contienen filtros de Gabor, también cambia si el usuario usa manga corto o larga. Hacen el reconocimiento de gestos dinámicos pero no me queda muy claro si cuadro a cuadro van haciendo el mismo proceso que con los gestos estáticos, que es el mencionado anteriormente.

El trabajo propuesto por (Manuel Caputo referenciar) realiza el reconocimiento de gestos dinámicos y estáticos, estos últimos son utilizados para determinar el inicio y el término de los gestos dinámicos. Se utilizan dos sensores Kinect y una cámara Logitech C910 de alta definición para capturar los gestos. El trabajo esta compuesto (numero) de etapas, las cuales se explican enseguida.

La cámara y los dispositivos Kinect no son sincronizados entre si. 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 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 un ROI es seleccionado, 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 emparejamiento 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 que utilizaron los 11 gestos. La precisión obtenida usando la distancia de angulo de giro es de 85 %, usando la distancia de momentos HU la precisión es de 58 %.

A pesar que la mayoría de los modelos vistos en la parte de arriba solucionan muchos de los problemas de los modelos basados en la visión. Ninguno de ellos puede resolver el problema de iluminación y oclusión, formada por lo dedos. Allí la importancia de la investigación propuesta, pues dará solución a estos inconvenientes al momento de reconocer los gestos.

1.7.3. Sistemas comerciales

Existen dispositivos como: Leap Motion ², MYO, y software como: Flutter ³, que realizan el reconocimiento de gestos, y el reconocimiento lo aplican 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 detalle es que

²<https://www.leapmotion.com/>

³<https://flutterapp.com/>

los desarrolladores de los dispositivos o software no dan a conocer los detalles de como es que lo solucionan algunos de los problemas o como mejoran la precision.

El dispositivo Leap Motion, fig 1 fue creado para el seguimiento de manos y dedos, este también hace el reconocimiento de ciertos gestos estáticos y dinámicos. El dispositivo consta (referenciar articulo internet) 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.



Figura 1: Ejemplo del reconocimiento del gesto usando Leap Motion, 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, lo único que se conoce ⁴ es que con la imágenes obtenidas con el dispositivo, se hace una representación 3D de las manos, antes de realizar el modelo las imágenes son segmentadas del fondo para quitar la iluminación u otros objetos que causen ruido en la imágenes. Para realizar el seguimiento se extraen la características como los dedos y otras, 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 son procesadas estas señales. ⁵

No se cuenta con la informacion detallada del funcionamiento de MYO, lo único que se

⁴<http://blog.leapmotion.com/hardware-to-software-how-does-the-leap-motion-controller-work/>

⁵<http://www.digitaltrends.com/pc-accessory-reviews/myo-gesture-control-armband-review/>

conoce es que el reconocimiento consta de tres etapas ⁶. La primera es la adquisición de la señales eléctricas que producen los músculos del antebrazo, las cuales son capturadas mediante sensores EMG (), giroscopio, acelerómetro y magnetómetro; en la segunda esta se amplifica la señal y se aplican filtro pasa banda y por último el procesamiento de la señal donde se reconoce el gesto usando un algoritmo de aprendizaje de máquina desarrollado por la compañía.



Figura 2: Ejemplo del reconocimiento del gesto usando MYO, es el dispositivo que esta colocado en el brazo.

MYO funciona en cualquier ambiente donde haya variaciones en la iluminación y es invariante a rotación. La desventaja que tiene es que la calibración puede ser tediosa ya que requiere algunos intentos, pues requiere varias repeticiones de los gestos; el uso del dispositivo requiere uso de manga corta; otra desventaja es que tiene una cantidad considerable de falsos positivos. ⁷

Flutter es un software que reconoce cuatro gestos estáticos detectando la palma de la mano, usando la cámara web como dispositivo de entrada. Flutter permite controlar aplicaciones multimedia de la computadora. Las limitaciones del software son que solo reconoce gestos estáticos, realiza acciones no deseadas al hacer gestos involuntarios y no siempre reconoce los gestos.

⁶<https://www.quora.com/How-does-MYO-wearable-gesture-control-work>

⁷)



Figura 3: Ejemplo del reconocimiento del gesto usando , que es el dispositivo que se encuentra conectado a la Laptop.

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

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

2.2. Reconocimiento de gestos con la manos

El reconocimiento de gestos se divide en tres fases (Rautaray y Agrawal, 2012), detección o segmentación; extracción de características seguimiento; dependiendo si los gestos son dinámicos, por último la etapa final el reconocimiento del gesto. Este se clasifican en dos modelos, basados en la visión y en contacto, esta clasificación depende de la manera en que son capturados los datos, es decir la forma en que se obtiene el gesto, para posteriormente poderlo reconocer.

Los primeros acercamientos para llevar acabo el reconocimiento de gestos 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, esto 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 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.

En este trabajo, se toma el enfoque basado en la visión ya que se quiere obtener un sistema que para el usuario sea fácil de interactuar, y esta interacción sea natural y una manera de lograr esto es tomando este enfoque. estos tienen mayor complejidad (acomodar este párrafo :P)

Los métodos basados en la visión se pueden representar por dos modelos (Rautaray y Agrawal, 2012), los basados en 3D, da una descripción espacial en 3D de la mano, y los basados en apariencia, como su nombre lo dice se basan en la apariencia de la mano. Los modelos basados en apariencia se dividen en dos categorías, los estáticos (modelo de silueta, de contorno deformables) y de movimiento (de color y movimiento).

2.2.1. Etapas del reconocimiento

del reconocimiento de gestos Enseguida se describen las etapas del reconocimiento de gestos 4, con los métodos para llevar cada una de estas.

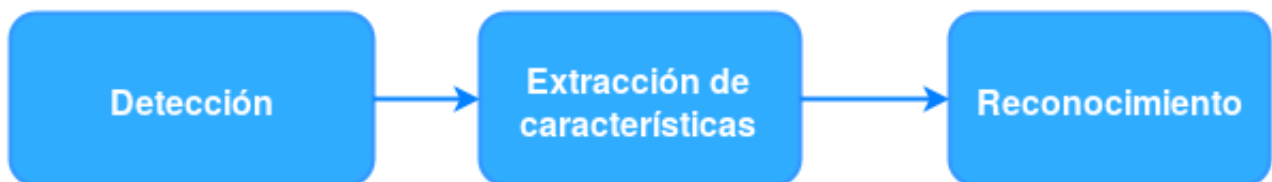


Figura 4: Procedimiento del algoritmo de detección

2.2.1.1. Detección

En esta etapa se detecta y segmenta la información relevante de la imagen (la mano), con la del fondo, 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 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.

- **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.

2.2.1.2. Seguimiento

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 a cabo 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.

2.2.1.3. 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 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 línea 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, y los índices 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 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. 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 5: 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 compañía 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 6.

El dispositivo Kinect está equipado con una serie de sensores que permiten obtener imágenes a color y de profundidad (imágenes que indican a la distancia que está 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 está equipado con los siguientes componentes: una 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 ⁷.

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



Figura 6: Sensor Kinect para Windows

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

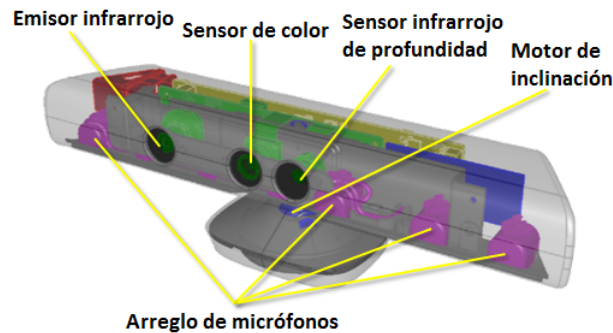


Figura 7: Componentes del sensor Kinect

- 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 1280×960 píxeles. La velocidad de los cuadros por segundo varía 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 640×480 píxeles.
- 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 $+27^\circ$.

- 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 lleva a cabo de la siguiente manera, figura 8 el primer paso es localizar la mano, se lleva a cabo 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.

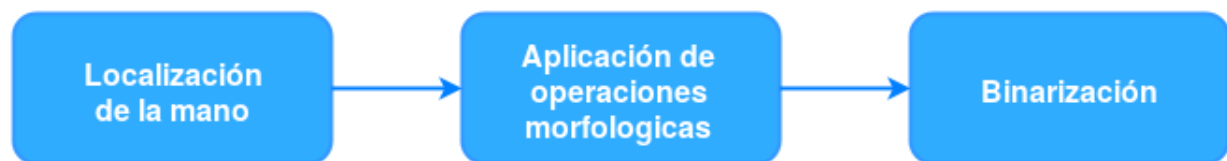


Figura 8: Proceso de detección de la mano.

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 9 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.

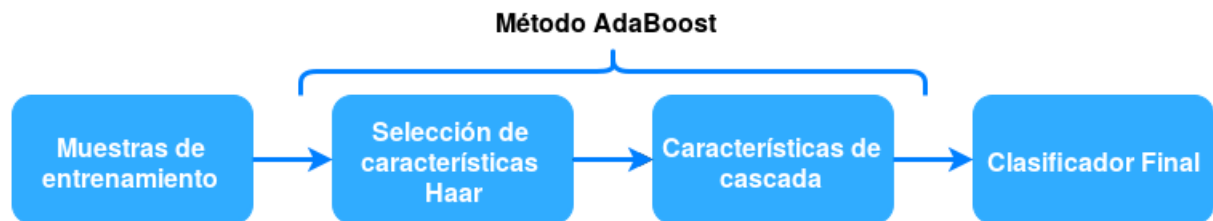


Figura 9: Procedimiento del algoritmo de detección

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 10. A continuación se explicaran los operadores Haar básicos:

- Las características con dos rectángulos 10(a), 10(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 10(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 10(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

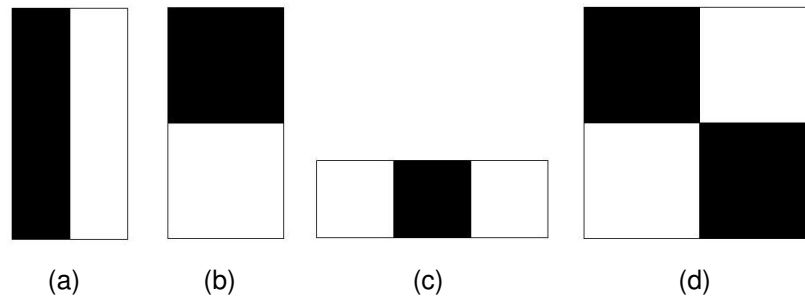


Figura 10: Ejemplo de operadores Haar

tiempo constante.

La imagen integral, SI , de una imagen, $S(x, y)$, es calculada como la suma del valor de los píxeles 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 11 muestra un ejemplo donde se calcula la imagen integral, fig. 11(b), de la imagen original 11(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 11: Ejemplo del cálculo de la imagen integral

La imagen integral permite calcular la suma de los píxeles 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 píxeles;

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 12, la region α se encuentra resaltada en color azul.

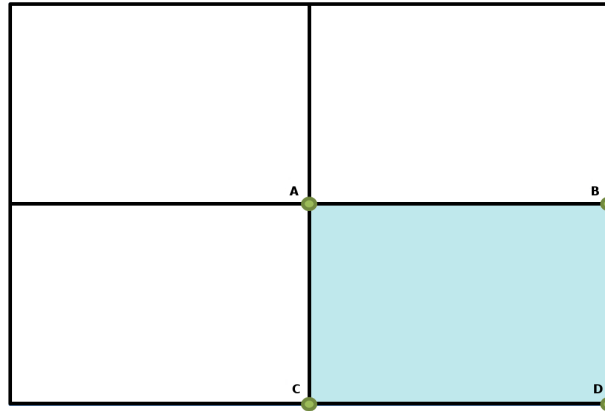


Figura 12: Regiones de la imagen integral

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 precisión mayor que los otros 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, θ 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, α_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 $\{(x_1, y_1), \dots, (x_n, y_n)\}$ 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 ϵ 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:

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

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

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 precisión de la detección y del número de falsos positivos. El clasificador en cascada 13 esta compuesto

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

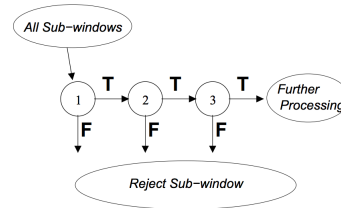


Figure 4: Schematic depiction of a the detection cascade.

Figura 13: Clasificador en forma de cascada

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.

- 1: $F_0 = 1, D_0 = 1$
 - 2: $i = 0$
 - 3: **mientras** $F_i > F_{Tarjet}$ **hacer**
 - 4: $i = i + 1$
 - 5: $n_i = 0, F_i = F_{i-1}$
 - 6: **mientras** $F_i > F \times fp_{i-1}$ **hacer**
 - 7: $n_i = n_i + 1$
 - 8: Entrenar un clasificador usando AdaBoost con P, N y n_i características.
 - 9: 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 $d \times D_i - 1$.
 - 11: **fin mientras**
 - 12: $N = 0$
 - 13: **si** $F_i > F_{Tarjet}$ **entonces**
 - 14: Evaluar el actual clasificador en cascada en el conjunto de imágenes negativas y poner cualquier detección falsa en el conjunto N .
 - 15: **fin si**
 - 16: **fin mientras**
-

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:

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

Existen diversas técnicas para binarizar una imagen, estas se pueden clasificar en dos grupos dependiendo de la manera en que se 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 método 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 $b \times b$. 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 remueve tomando en cuenta la forma y estructura de la imagen. Las operaciones morfológicas utilizan un elemento estructural el cual se aplica por toda la imagen, los elementos estructurales pueden ser de distintas formas como 14

Existen distintas operaciones morfológicas, las principales o básicas 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. La dilatación se define como:

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

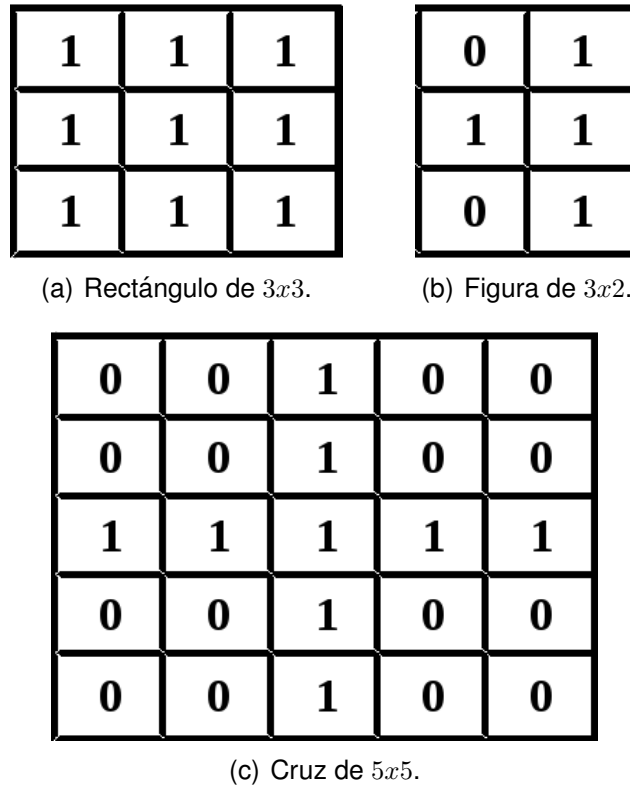


Figura 14: Ejemplos de elementos estructurales

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.

3.2.3.3. Apertura

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

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

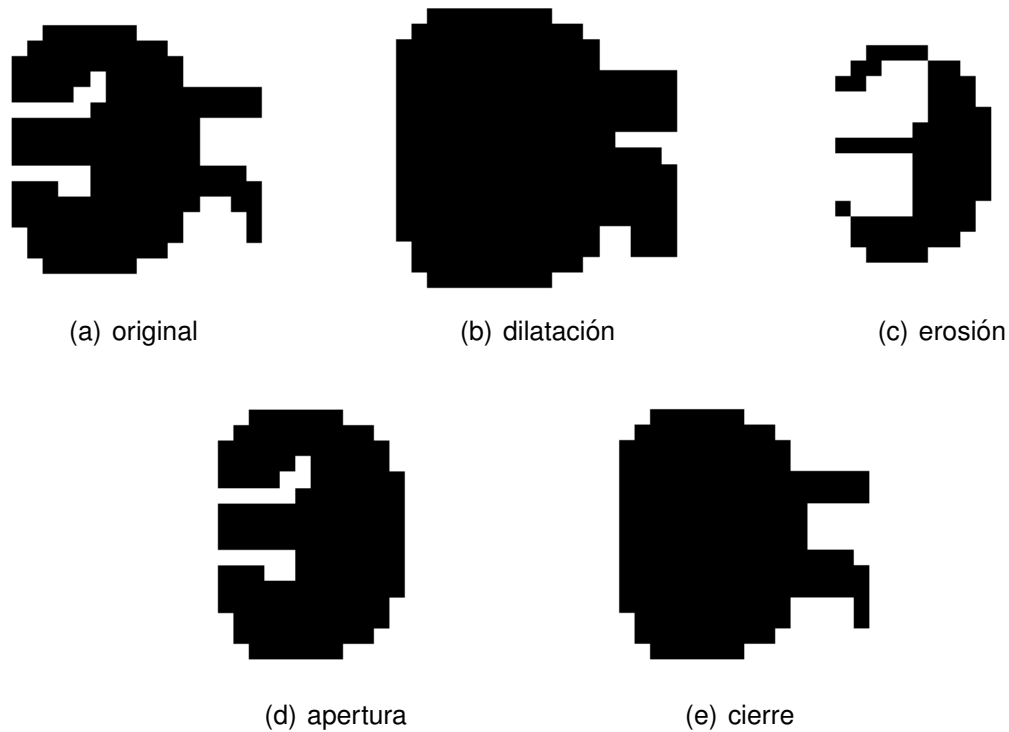


Figura 15: Aplicación de operaciones morfológicas (citar)

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 encontrar las características que sean capaces de describir la mano, de manera que con estas, se pueda reconocer los gestos realizados por la o las manos. Las características se guardan en un vector, llamado vector de características, donde la dimensión del vector es el número de características que describen la mano.

En este trabajo se utilizan características geométricas, para extraerlas se hace uso de la envolvente convexa y de los defectos de convexidad. A continuación se definen los conceptos anteriores y el de conjunto convexo.

Sea A un conjunto en el espacio euclidiano \mathbb{R}^d , A es un conjunto convexo ⁴ si contie-

⁴Weisstein, Eric W. "Convex." From MathWorld—A Wolfram Web Resource.

ne todos los segmentos de línea que unen a cualquier par de puntos pertenecientes al conjunto. Donde d es la dimensión del espacio euclidiano.

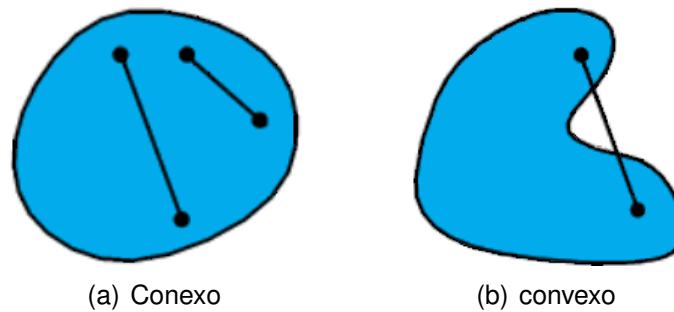


Figura 16: Conjunto conexo y convexo (citar)

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

Los defectos de convexidad de un con casco convexo, es el conjunto de puntos que no pertenecen al casco convexo. El defecto es el espacio entre la línea y el objeto. Sea $CD = \{cd_1, cd_2, \dots, cd_n\}$ los defectos de convexidad de una envolvente convexa. Cada defecto esta compuesto de tres elementos $\{s_i(x, y), d_i(x, y), e_i(x, y)\} \in cd_i, \quad i = \{1, 2, \dots, n\}$. En la imagen 17 se encuentran de color amarillo los puntos de profundidad de los defectos de convexidad.

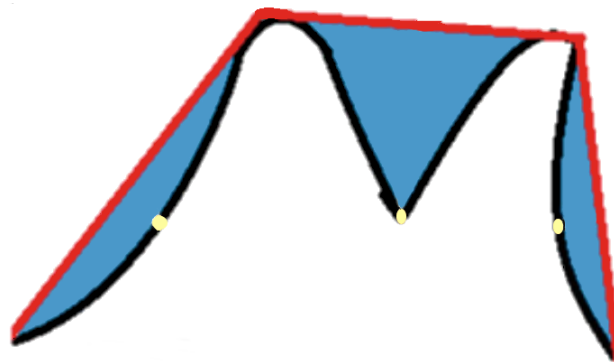


Figura 17: En la imagen se aprecia de color rojo la envolvente convexa, de negro el contorno de la figura, y los puntos amarillos son el punto de profundidad de los defectos de convexidad.

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 conocidos (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:

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

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.

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

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

Maximizar el margen, encontrar el mínimo de w .

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

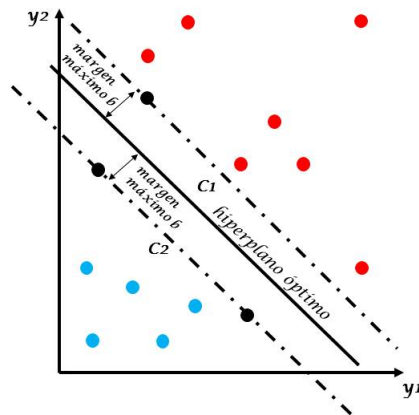


Figura 18: Clasificación de maquina de soporte usando kernel lineal

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 19.

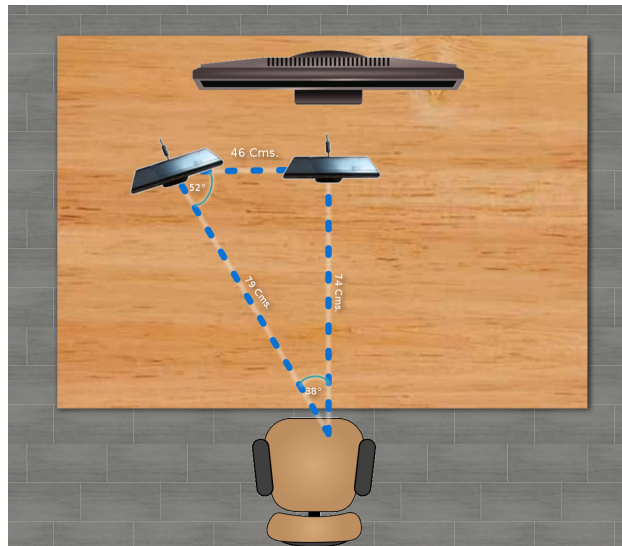


Figura 19: Configuración del sistema de reconocimiento de gestos

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. 20

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



Figura 20: Representación de los datos capturados por los Kinect

provenientes de los sensores son como la figura 21; el ruido es reducido usando filtros de mediana este es aplicado en toda la imagen usando una ventana de tamaño 13. La imagen resultante $S(x, y)$ es como la que se muestra en la figura 21.

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



Figura 21: 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 a cabo 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 22(a), palma con dedos juntos 22(c) y finalmente el puño 22(b), como se muestran en la figura 22. Las imágenes de fondo fueron tomadas de distintos escenarios como se muestra en la figura 23. 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 24.

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/americanm>

⁴<https://github.com/americanm>

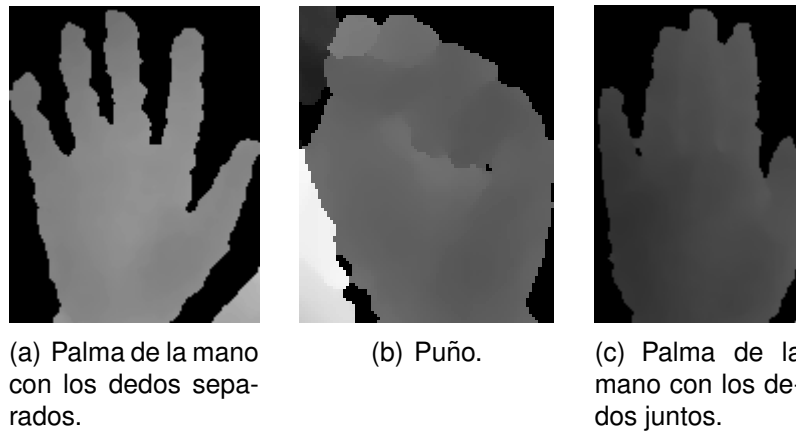


Figura 22: Ejemplo de imágenes de poses de nuestra base de datos.



Figura 23: Imagen del fondo de nuestra base de datos



Figura 24: 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 25(a) o unir pequeños hoyos que existen en la imagen, como los que se encuentran en la figura 25(b).

Las operaciones anteriores utilizan un elemento estructural rectangular; para la operación de apertura el tamaño del elemento es de 3×7 píxeles; para el cierre se aplicó con un tamaño 7×7 píxeles.



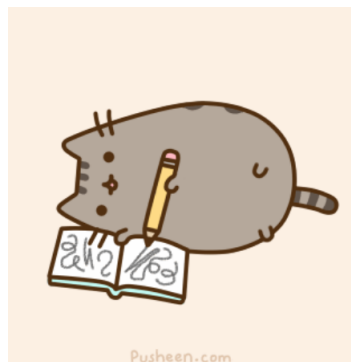
(a) ROI donde existe una unión entre los dedos.



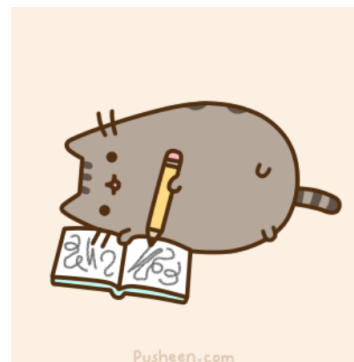
(b) ROI donde se aprecian hoyos en la mano.

Figura 25: Las imágenes anteriores muestran 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 26: Las 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 3×3 píxeles. La figura siguiente es la imagen binarizada.



Figura 27: 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. 28 muestra en color blabla el contorno de la mano.

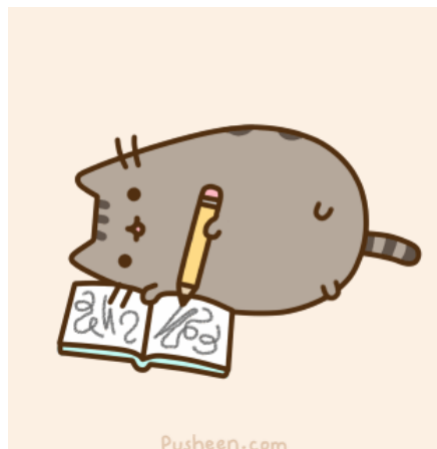


Figura 28: Contorno de la mano

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 29: 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.

Sea $CD = \{cd_1, cd_2, \dots, cd_n\}$ los defectos de convexidad de una envolvente convexa. Cada defecto esta compuesto de tres elementos $\{s_i(x, y), d_i(x, y), e_i(x, y)\} \in cd_i$, $i = \{1, 2, \dots, n\}$. Sea D el conjunto de las distancias de $d_i(x, y)$ a la orilla de la envolvente convexa.

El ángulo α entre los dedos f_j y f_{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:

$$\theta_{f_j} = \tan^{-1} |m_j - 90^\circ|$$

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

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

Algoritmo 3 Calcula el número de dedos de la mano.

Entrada: $D = \{\delta_1, \delta_2, \dots, \delta_n\}$.

Salida: Número de dedos, Nf .

```

1: para  $i = 1$  hasta  $n$  hacer
2:    $minDist = 20, maxAng = 60, antecesor = 0, sucesor = 0$ .
3:   si  $\delta_i < minDepth$  entonces
4:     continuar
5:   fin si
6:   si  $i=0$  entonces
7:      $antecesor = n - 1$ 
8:   si no
9:      $antecesor = i - 1$ 
10:  fin si
11:  si  $i=n-1$  entonces
12:     $sucesor = 0$ 
13:  si no
14:     $sucesor = i + 1$ 
15:  fin si
16:  Calcular el angulo entre  $s_{antecesor}(x, y)$  y  $s_{sucesor}(x, y)$ 
17:  si  $angulo \geq maxAng$  entonces
18:    regresar falso
19:  fin si
20:   $Nf = Nf + 1$ .
21: fin para

```

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 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 30, divididas en partes iguales. De tamaño 640 por 480 pixeles.



Figura 30: 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.

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

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 2.410¹ un wrapper de OpenCV².

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

5.1. Experimentos de gestos estáticos

5.2. Experimentos de gestos dinámicos

¹http://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 640×480 pixeles y la resolución del sensor infrarrojo 640×480 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

- 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 decision-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.
- 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.
- 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*.
- 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*.
- 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**.
- 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.