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 Homero Hidalgo Silva

Dr. Josué Álvarez Borrego

Dra. Ana Isabel Martínez García
Coordinador del Programa de Posgrado en Ciencias de la Computación

Dra. Rufina Hernández Martínez
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
Director de tesis

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 por sus siglas en inglés Human Computer Interaction. La relevancia de los gestos con las manos, es que con ellos, se puede obtener una interacción natural entre la computadora y el usuario, por lo que se han desarrollado diversos métodos para encontrar un modelo que funcione en tiempo real y en diversas circunstancias. En este trabajo se propone un método que fusiona información proveniente de dos sensores Kinect, para realizar el reconocimiento de gestos estáticos y dinámicos en tiempo real, en circunstancias de baja iluminación y cuando existe obstrucción por parte de los dedos de la mano.

Palabras Clave: **Gestos con las manos, Kinect, baja iluminación, obstrucción.**

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
Thesis director

The recognition of hand gestures has been prominent in different areas of computer science, eg. HCI (Human Computer Interaction), is important because it helps create a natural interaction between the computer and the user, so varios methods have been developed to find the model that works in real time and in different circumstances. In this work we proposed a model that uses the information from the two Kinect devices in order to recognize static and dynamic gestures in real time under poor ligth and partial occlusion.

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

Dedicatoria

A mis padres y a la memoria de mi abuelita Aurora.

Agradecimientos

A mis padres y hermanas por apoyarme siempre que lo necesito, en especial durante la maestría.

Al Dr. Vitaly Kober por permitirme trabajar con él.

A mis compañeros y grandes amigos Darién Miranda y Oscar Peña por las platicas intensas y no tan intensas, por las comidas juntos, por los días de promo, en fin, por estar y ayudarme siempre que lo necesitaba, sin ellos este proceso no habría sido el mismo.

A mi compañera y amiga Julia Diaz por sus valiosos consejos.

A Daniel Miramontes por comentarios y observaciones.

A los miembros de mi comité de tesis, Dr. Hugo Hidalgo y Dr. Josué Álvarez, por sus valiosos comentarios.

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 figuras (continuación)

Figura Página

Lista de tablas

Tabla Página

Lista de tablas (continuación)

Lista de tablas (continuación)

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 cualquier otra 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, en especial el desarrollo de computadoras, su incremento en procesamiento, la reducción de su tamaño y costo ha hecho que estas se incorporen cada vez más y sean parte esencial en nuestra vida diaria. De manera que se han creado y con ello estudiado distintas áreas de las ciencias computacionales, particularmente el área de interacción humano computadora (HCI, por sus siglas en inglés Human Computer Interaction), el 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. La manera en que el humano interactua con la computadora ha sido básicamente la misma desde que se empezó a tener acceso a ellas, fue hasta principios de esta década cuando la interacción ha empezado a cambiar pues ahora existen pantallas táctiles, reconocimiento de voz. Estas formas de interacción han sido aceptadas por los usuarios ya que hacen que la forma en que nos "comunicamos" con la computadora sea fácil y sencilla. 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 enfoques utilizaban como medio de captura sensores como guantes de datos, marcadores de colores y acelerómetros, los cuales se colocaban en la o las manos para poder capturar la posición e identificar la pose realizada. Las técnicas desarrolladas posteriormente obtienen la información necesaria para reconocer el gesto usando distintos tipos de imágenes o videos, que son obtenidos mediante diversos tipos de cámaras, por lo que no es necesario portar algún dispositivo para reconocer un gesto hecho con las manos.

Los métodos que utilizan imágenes o video, son los más utilizados para realizar el reconocimiento 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, debido a que existen diversos aspectos que hay que tener en cuenta para obtener una buena precisión en el reconocimiento del gesto; por ejemplo tener en cuenta la resolución del dispositivo de captura, el ruido que existe en las imágenes, el tiempo de computo para realizar el procesamiento de las imágenes y las situaciones que se pueden presentar en la vida diaria que puedan entorpecer el reconocimiento del gesto.

Aunque existe una gran variedad de métodos y sistemas que hacen el reconocimiento de gestos de las manos, no existe alguno que presente en el reconocimiento un alto grado de precisión en todas las situaciones que se presentan en el mundo real, tales como; condiciones diversas de iluminación ya sea baja o alta, que funcionen en tiempo real, que funcionen a diversas escalas, es decir no importando el tamaño de la mano, que sea invariante a rotación, invariante al color de la piel y cuando exista alguna obstrucción parcial en el área de la mano. 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 obstrucción 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 en ciertas ocasiones es posible obtener 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. Sin embargo, 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, el fondo, que funcione en tiempo real o cuando exista obstrucción en alguna parte de la mano o en los dedos.

1.2. Justificación

Los métodos de reconocimiento de gestos desarrollados logran obtener un buen grado de reconocimiento en ciertas situaciones, generalmente en situaciones controladas. De manera que se necesitan nuevos métodos que funcionen no solamente en condiciones ideales sino en situaciones que se presentan en la vida diaria 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 de reconocimiento de gestos con las manos, gestos estáticos y dinámicos. El sistema debe funcionar bajo ciertas situaciones que se presentan en un ambiente natural. Es decir el sistema debe funcionar en circunstancias de baja iluminación y cuando exista obstrucción causada por los dedos 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 sensor 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.
- Validar el sistema diseñado, en cuanto a su eficiencia con base al reconocimiento de los gestos, en circunstancias de baja iluminación y obstrucción. En el análisis del sistema se usará información real.
- Comparar y validar 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

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

Los primeros enfoques para llevar acabo el reconocimiento de gestos con las manos fue usando modelos de contacto (Rautaray y Agrawal, 2015) y (Nayakwadi, 2014).
El modelo utiliza dispositivos que están en contacto físico con la mano del usuario, ver la Figura 1(a) para reconocer el gesto. Por ejemplo usan guantes de datos, marcadores de colores, acelerómetros y pantallas multi-toque. Este enfoque no es tan aceptado pues entorpecen la naturalidad entre la interacción del humano y la computadora.
Los modelos basados en la visión, ver la Figura 1(b) surgieron como respuesta a esta desventaja. Estos utilizan cámaras para extraer la información necesaria para realizar el reconocimiento. Los dispositivos van desde cámaras web hasta algunas más sofisticadas por ejemplo cámaras de profundidad.

En este trabajo, se toma el enfoque basado en la visión debido a que se busca obtener un sistema que para el usuario la interacción sea natural y la manera de lograrlo es

(a) Dispositivos basados en contacto: a la 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].

(b) Dispositivos basados en visión: en la imagen se observan distintos tipos de cámaras. A la izquierda de la imagen se observa una cámara web [4], en el centro una cámara digital [5] y a la derecha de la imagen una cámara TOF [6].

Figura 1: Dispositivos utilizados para la captura de gestos.

tomando este enfoque.

1.7. Estado del arte

En esta sección se presentan los trabajos relevantes de cada uno de estos enfoques y también se mencionan algunos de los sistemas comerciales importantes.

1.7.1. Modelos de contacto

Los primeros trabajos de reconocimiento de gestos con las manos utilizan este modelo, actualmente se sigue utilizando pero en menor grado.

Un método de reconocimiento de gestos con las manos que utiliza guantes de datos (Yoon *et al.*, 2012) propone un sistema de reconocimiento de gestos estáticos, el cual reconoce veinticuatro gestos tomados del Lenguaje de Señas Americano, ASL (por sus siglas en inglés, American Sign Lenguaje). Este modelo consta de tres etapas. La primera

[1] http://www.technologyreview.com/article/414021/open-source-data-glove/
[2] http://www.digitaltrends.com/computing/the-gloves-that-could-change-the-world/
[3] https://www.nintendo.es/Wii/Wii-94559.html
[4] http://es.ccm.net/download/descargar-2562-driver-de-microsoft-lifecam-vx-3000
[5] http://www.canon.com.mx/ficha.aspx?id=722
[6] http://us.creative.com/p/web-cameras/creative-senz3d

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 cinco personas, trescientas muestras de cada gestos, ocho mil 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 este requiere tiempo para adaptarse a distintos usuarios. Otra desventaja para este sistema es que solo reconoce gestos estáticos.

A finales del año 2014 se lanzó el dispositivo MYO [8], el cual reconoce gestos dinámicos, se muestra en la Figura 2. Este aparato es un brazalete que reconoce cinco gestos dinámicos. Leyendo la actividad de los músculos del antebrazo y mandando estas señales vía bluetooth a la computadora donde estas señales son procesadas. [9]

No se cuenta con la informacion detallada del funcionamiento de MYO, lo único que se conoce es que el reconocimiento consta de tres etapas [10]. La primera es la adquisición de la señales eléctricas que producen los músculos esqueléticos del antebrazo, las cuales son capturadas mediante sensores que 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.

MYO funciona en cualquier ambiente donde haya variaciones en la iluminación y es invariante a rotación. La principal desventaja es la calibración debido a que puede ser tediosa pues es necesario realizar un considerable número de repeticiones de algunos gestos. Otra desventaja es que tiene una cantidad considerable de falsos positivos [12]. Un

[8] https://www.thalmic.com/en/myo/
[9] http://www.digitaltrends.com/pc-accessory-reviews/myo-gesture-control-armband-review/
[10] https://www.quora.com/How-does-MYO-wearable-gesture-control-work
[11] https://www.myo.com/
[12] http://myogroupfive.blogspot.mx/2013/11/benefits-disadvantages-for-business_24.html

Figura 2: Ejemplo del reconocimiento del gesto usando MYO, controlando el volumen de la computadora. El dispositivo es el que aparece en el brazo del sujeto. Imagen recuperada de [11].

punto importante es que para el uso del dispositivo es preferible el uso de manga corta.

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. Existe una gran gama de datos que se pueden extraer con este modelo, usando las técnicas o métodos adecuados para el tipo de datos extraído se puede llegar a tener gran precisión en el reconocimiento. Enseguida se presenta tres trabajos relevantes los cuales utilizaron distintos tipos de cámaras y número de ellas.

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

Antes de realizar 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 de la imagen de la mano el 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 se obtienen 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 se realiza por medio de máquinas 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. Una limitación de sistema es que no es tratado el problema de obstrucción.

Otro trabajo propuesto, (Caputo *et al.*, 2012) 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 web Logitech C910 de alta definición para capturar los gestos. El trabajo está compuesto de cuatro etapas.

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 está basado en la ubicación de la manos y la cabeza. La cámara y los dispositivos Kinect no son sincronizados entre sí.

La parte de la detección y seguimiento se lleva acabo utilizando la librería OPENNI, en específico usando la detección del esqueleto. 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 una mejor aproximación de la localización 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 algo-

ritmo de (Chang *et al.*, 2004), dicho contorno es extraído como polígono y es simplificado con el algoritmo de Douglas Peuker.

El reconocimiento del gesto utiliza el empatamiento de polígonos, el cual se basa en comparar dos polígonos, mediante distancias. Esto usando distancia de momentos HU y ángulo de giro. 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 qué gesto se realice. Para probar la precisión del sistema se crearon dos bases de datos: una con 120 polígonos etiquetados que representan once gestos y otra con 144 gestos de tres personas distintas realizando los once 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 veinticuatro 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.

Después se extraen las características, las cuales son extraídas usando Histogramas Orientados a Gradientes, HOG (Histogram of Oriented Gredient).

Para la clasificación de los gestos, se utiliza el algoritmo de aprendizaje de máquina, máquinas de soporte vectorial. Para el entrenamiento se utilizaron 2400 imágenes, cien por cada letra del alfabeto. Se encontró que existen gestos ambiguos, es decir que no se pueden clasificar correctamente, estos son los gestos que representan la letras A, E, M, N, S, T. Se realizó 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 último 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 [13], MYO [14] y software como Flutter [15], que realizan el reconocimiento de gestos, 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 cómo solucionan algunos de los problemas o cómo mejoran la precisión. Enseguida se describen los sistemas mencionados anteriormente.

El dispositivo Leap Motion (Figura 3) 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 3: Ejemplo del reconocimiento del gesto usando Leap Motion, mostrando una aplicación donde los gestos son representados en 3D. Leap Motion es el dispositivo que se encuentra conectado a la laptop. Imagen recuperada de [13].

El proceso de captura de los datos, la segmentación, la extracción de características, el seguimiento y el reconocimiento del dispositivo no se conoce a detalle, pues no ha sido publicado. Solo se conoce [16] que se utilizan tres cámaras infrarrojas. Con la imágenes

[13] https://www.leapmotion.com/
[14] https://www.myo.com/
[15] https://flutterapp.com/
[16] http://goo.gl/INzrdR

obtenidas con se hace una representación 3D de las manos. Antes de realizar el modelo, se sustrae el fondo de las imágenes para eliminar el ruido generado por la iluminación u otros objetos. Para realizar el seguimiento se extraen la características. Una de ellas son los dedos, el algoritmo de seguimiento interpreta la información 3D e infiere la posición de los objetos ocluidos. Se aplican filtros para suavizar los datos.

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

Se desconoce cómo funciona el software, 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.

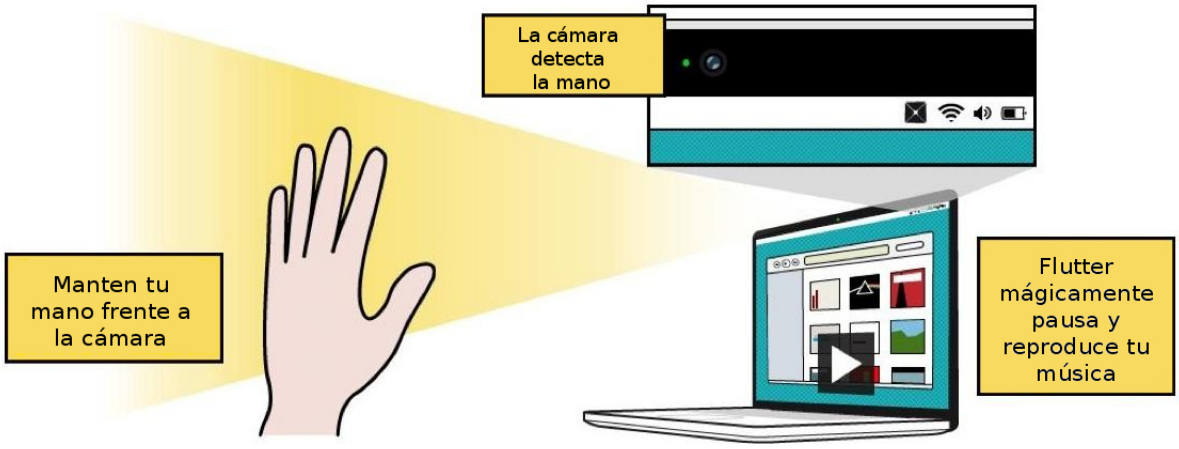


Figura 4: La imagen anterior representa el funcionamiento del software Flutter. Imagen recuperada de [15].

Flutter permite controlar aplicaciones multimedia como: YouTube [17] ,VLC [18], Spotify [19], Netflix [20]. Las limitaciones del software son que solo reconoce gestos estáticos. Una desventaja es que presenta falsos positivos al reconocer acciones del usuario.

Aunque estos dispositivos y software para reconocer gestos solucionan algunos problemas importantes en el área, sigue existiendo el problema de bloqueo e iluminación.

[17] https://www.youtube.com/
[18] http://www.videolan.org/vlc/
[19] https://www.spotify.com/
[20] https://www.netflix.com/

De allí la importancia que existan nuevos modelos que ataquen estos problemas que se presentan frecuentemente en el reconocimiento de los gestos.

1.8. Organización de la tesis

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

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 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 (Mitra *et al.*, 2007). De acuerdo con la literatura 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 (Mitra *et al.*, 2007).

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.

Los métodos basados en la visión (ver Capitulo 1, Sección 1.6) realizan la representación del gesto con diferentes técnicas las cuales se separan en dos categorías (Rautaray y Agrawal, 2015): basados en apariencia y basados en modelo 3D. Los basados en modelo 3D convierten los datos en entrada en una forma espacial y los basados en apariencia utilizan los datos 2D de la imagen de entrada.

De acuerdo con la literatura, el proceso de reconocimiento de gestos basados en la visión se dividen en tres fases que son: detección, extracción de características o seguimiento, dependiendo si los gestos son dinámicos, por último el reconocimiento del gesto (Rautaray y Agrawal, 2015). Otros autores incluyen la etapa de adquisición de datos (Hasan y Mishra, 2012). Las etapas se abordarán en la sección siguiente.

2.2.1. Etapas del reconocimiento

En la sección se abordaron las etapas del reconocimiento y se mencionan los principales algoritmos utilizados en cada una de éstas. El diagrama de la Figura 5 muestra los diferentes pasos en el reconocimiento.

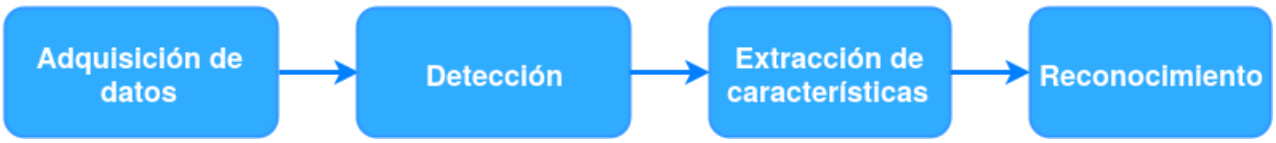


Figura 5: El diagrama ejemplifica el procedimiento del reconocimiento de gestos.

El proceso de reconocimiento varía un poco dependiendo del tipo de gesto, si es estático o dinámico. Por ejemplo en la Figura 5 ejemplifica perfectamente el proceso de reconocimiento de un gesto estático, para los gestos dinámicos se necesita una fase extra, el seguimiento el cual se realiza una vez detectada la mano, 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, el tipo de cámara depende del tipo de información que se quiera obtener de la mano.

Las cámaras mas utilizadas son las RGB, con ellas se puede obtener información de la mano tales como la textura, forma, color, etc. Últimamente se han hecho muy populares las cámaras que proporcionan profundidad, tales como las basadas en tiempo de vuelo TOF, por sus siglas en inglés, o en luz estructurada. Estas cámaras generalmente se utilizan cuando se quiere obtener un modelo 3D de la mano, utilizando otra cámara normalmente RGB.

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 poder detectar la mano como: la de color de la piel, forma, movimiento, entre otras que

generalmente son combinaciones de alguna de estas. Enseguida se describe brevemente cada una de estas.

- Color de la piel: Se basa principalmente en escoger un espacio de 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, éstas pueden ser obstruidas 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 con base en la apariencia y textura. Esto se logra entrenando un clasificador con un conjunto de imágenes.
- Modelo 3D: Depende de cuál modelo se utilice, son las características de la mano requeridas.
- Movimiento: Generalmente ésta se usa con otras formas de detección ya que para utilizarse por sí sola hay que asumir que el único objeto con movimiento es la mano.

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

2.2.1.3. Extracción de características y seguimiento

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

Existen dos tipos de características geométricas y no geométricas. Un ejemplo de características geométricas son las yemas de los dedos, la dirección de los dedos, el contorno de la mano y entre otras características. Y un ejemplo de características no geométricas son el color, siluetas y texturas de la mano. (Murthy y Jadon, 2009).

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

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

El seguimiento 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 algunos 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 más similar en una secuencia de imágenes, la distribución es basada en el color.

2.2.1.4. Reconocimiento

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

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

A continuación se encuentran los principales métodos para llevar acabo el reconocimiento del gestos (Rautaray y Agrawal, 2015).

- K-medias: Es un método de agrupamiento el cual consiste en determinar los k puntos llamados centros para minimizar el error de agrupamiento, que es la suma de las distancias de todo los puntos al centro de cada grupo. El algoritmo empieza localizando aleatoriamente k grupos en el espacio espectral. Cada píxel en la imagen de entrada es entonces asignadas al centro del grupo mas cercano

- Desplazamiento de medias: Es un método iterativo que encuentra el máximo en una función de densidad dada una muestra estadística de los datos.

- Máquinas de soporte vectorial: Consiste en un mapeo no lineal de los datos de entrada a un espacio de dimensión más grande, donde los datos pueden ser separados de forma lineal.

- Modelo oculto de Markov: Es definido como un conjunto de estados, un estado inicial, un conjunto de símbolos de salida y un conjunto de estados de transición. En el reconocimiento de gestos se puede caracterizar a los estados como un conjunto de las posiciones de la mano; las transiciones de los estados como la probabilidad de transición de cierta posición de la mano a otra; el símbolo de salida como una postura 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, véase Figura 6, y los indices de esta indican la posición, (Gonzalez y Woods, 2002).

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

2.4. Obstrucción

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

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

2.5. Métricas de desempeño

Existen diversos métodos para medir el rendimiento de un algoritmo de clasificación, una manera de representarlo es mediante una matriz de confusión.

Si consideramos una clasificación de un conjunto, donde cada elemento del conjunto es mapeado a un elemento del conjunto de etiquetas (positiva o negativa). El clasificador se encarga de mapear los elementos a las clases existentes, indicando la clase a la que

pertenece la elemento. El resultado de clasificación de algún elemento puede resultar en cuatro posibles estados. Si el elemento es positivo y la clasificación es positiva, es un verdadero positivo; si la clasificación es negativa es un falso negativo. Si el elemento es negativo y la clasificación es negativa, es un verdadero negativo; si la clasificación es positiva, es un falso positivo. Dado un clasificador y un conjunto de elementos, una matriz de confusión de 2×2, como la que se muestra en la Figura 7, puede ser construida por el resultado del conjunto de los elementos (Fawcett, 2006). Los valores de la diaginal de

		Clases: Positiva	Clases: Negativa
Predicción	Positiva	Verdadero positivo	Falso positivo
	Negativa	Falso negativo	Verdadero negativo

Figura 7: La siguiente imagen representa una matriz de confusión, de un problema de clasificación de dos clases.

la matriz de confusion representan las predicciones correctas, y los valores fuera de la diagonal representan los errores.

Distintas métricas de desempeño, pueden ser calculadas gracias a esta matriz. Enseguida se presentan algunas de estas métricas.

La tasa de precisión del clasificador puede ser calculado como:

$$Presición = \frac{Verdaderos \quad positivos}{Verdaderos \quad positivos + Falsos \quad positivos} \tag{1}$$

La tasa de exactitud del clasificador puede ser calculado como:

$$Exactitud = \frac{Verdaderos \quad positivos + Verdaderos \quad negativos}{Total \quad de \quad positivos + Total \quad de \quad negativos} \tag{2}$$

La tasa de verdaderos positivos Tp, del clasificador puede ser calculada como:

$$Tp \approx \frac{Positivos \quad clasificados \quad correctamente}{Total \quad de \quad positivos} \tag{3}$$

La tasa de falsos positivos Fp, del clasificador puede ser calculada como:

$$Fp \approx \frac{Negativos \quad clasificados \quad correctamente}{Total \quad de \quad negativos} \tag{4}$$

Capítulo 3. Sistema propuesto de reconocimiento de gestos

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

El sistema propuesto de reconocimiento de gestos consta de cuatro etapas principales, Figura 8. La primera etapa es la adquisición de los datos, en la cual se capturan las imágenes de entrada del sistema. La siguiente 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.

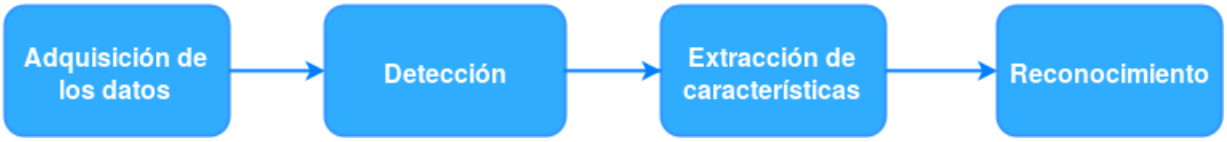


Figura 8: Metodología del sistema propuesto.

3.1. Adquisición de los datos

En la primera etapa del sistema los datos de entrada del sistema son capturados y preprocesados, la Figura 9 muestra las etapas de este proceso. Los datos provienen de los sensores de profundidad de dos dispositivos Kinect. Debido a la naturaleza del sensor las imágenes capturadas contiene ruido por interferencia (Mallick *et al.*, 2014), que puede ser reducido utilizando filtro de medianas (Maimone y Fuchs, 2011).

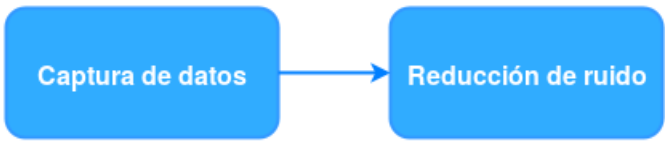


Figura 9: Proceso de la etapa de adquisición de datos.

3.1.1. Kinect

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. El dispositivo

fue desarrollado por la compañia PrimeSense en conjunto con Microsoft [1]. El sensor de profundidad que utiliza el dispositivo fue desarrollado por Zeev Zalesvky, Alexanser Shpunt, Aviad Maizels y Javier Garcia en 2005 [2]

El dispositivo Kinect está equipado con una serie de sensores que permiten obtener imágenes a color y de profundidad (las cuales indican la distancia a la que está ubicada un objeto del sensor). La Figura 10 muestra la parte frontal del dispositivo Kinect para Windows. Los sensores permiten hacer detección y seguimiento de personas. El dispositivo tiene la capacidad de detectar hasta seis personas y hacer el seguimiento de hasta dos personas [3].

Figura 10: Parte frontal del dispositivo Kinect en su versión para Windows, imagen recuperada de [4].

El sensor está 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 (Jana, 2013). Enseguida se describen brevemente cada uno de los componentes del sensor Kinect, estos se muestran en la Figura 11.

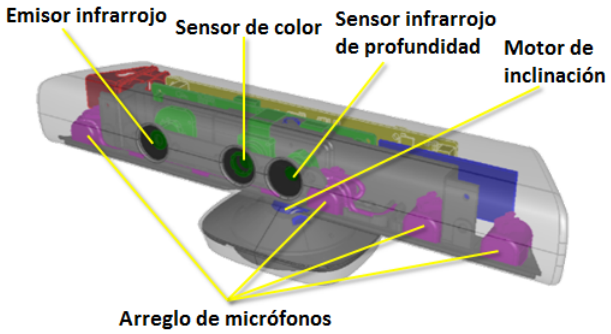


Figura 11: Componentes del sensor Kinect, imagen recuperada de [5].

[1] engadget.com/2013/06/21/life-after-kinect-primesense-post-microsoft/
[2] https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2007043036&recNum=1&maxRec=&office=&prevFilter=&sortOption=&queryString=&tab=PCT+Biblio
[3] https://msdn.microsoft.com/en-us/library/hh973074.aspx
[4] http://goo.gl/2GlwI
[5] https://msdn.microsoft.com/en-us/library/jj131033.aspx

- 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 treinta 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, estos puntos son proyectados frente al sensor. Estos puntos junto con el sensor de profundidad es posible medir la distancia que existe del sensor a algún objeto que esté frente a él.

- El sensor infrarrojo lee los puntos infrarrojos proyectados por el emisor infrarrojo, con la lectura de los datos se calcula la distancia que existe entre el objeto y el sensor. El sensor transmite los datos de profundidad con una velocidad de hasta treinta 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 un ángulo de $-27°$ a $+27°$, con respecto al eje vertical del sensor.

- La entrada de audio compuesta por un arreglo de cuatro micrófonos permite capturar el sonido y calcular la posición de la fuente.

- LED indica el estado del sensor. El LED en color verde indica que el sensor está disponible para utilizarse. El LED en color rojo indica que no existe conexión entre el Kinect y la computadora. La ausencia de color en el LED indica que el Kinect no se encuentra conectado a la fuente de energía eléctrica.

3.1.2. Filtro de mediana

Existen distintos métodos para eliminar el ruido por interferencia causado por dos o más Kinect, algunos de estos métodos son invasivos pues el hardware del Kinect es modificado (Mallick *et al.*, 2014). Una opción es utilizar filtro de mediana, debido a que rellena los valores faltantes en la imagen proveniente del Kinect (Maimone y Fuchs, 2011).

Sea W_{mn} el vecindario de tamaño $m \times n$, centrado en el pixel, s, que se encuentra en la posición (x, y). El filtro de mediana de la imagen $S(x, y)$, se calcula como:

$$s(x, y) = \underset{(u,v) \in W_{mn}}{\text{mediana}}\{S(u, v)\} \tag{5}$$

El filtro de mediana reemplaza el valor del pixel usando la mediana de las intensidades del vecindario del pixel, el valor del pixel en la posición (x,y) es incluido en el calculo (Gonzalez y Woods, 2002).

3.2. Detección

En esta etapa del sistema el objetivo es localizar la mano en la imagen y segmentar la mano del fondo de la imagen. Para extraer las características necesarias para el reconocimiento. La Figura 12 muestra el proceso para realizar la etapa de detección. El primer paso es localizar la mano, en este trabajo se utiliza el método de detección rápida de objetos; el siguiente paso es segmentar la mano del fondo, binarizando la imagen de la mano usando el algoritmo propuesto por (Otsu, 1979); finalmente se aplican las operaciones morfológicas apertura y cierre, para mejorar la segmentación, es decir eliminar ruido existente.

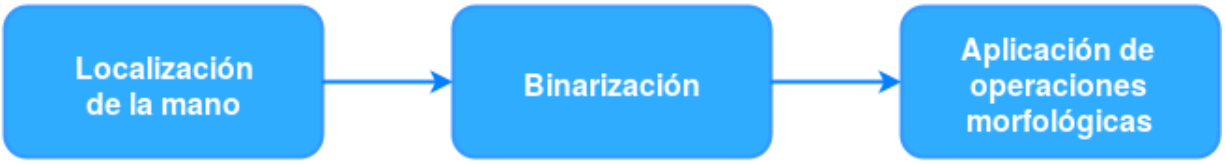


Figura 12: Proceso de detección de la mano.

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 propuesto por (Viola y Jones, 2001), el cual fue creado originalmente para atacar el problema de detección de rostros. El método puede ser usando para detectar cualquier objeto debido a que la detección se realiza clasificando imágenes basándose en el valor de características simples.

La técnica detecta 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 éstas 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 formarán el clasificador. Estas se escogen mediante el algoritmo de AdaBoost y su valor es calculado usando la imagen integral. El paso final involucra construir el clasificador mediante el uso de Adaboost, en forma de cascada.

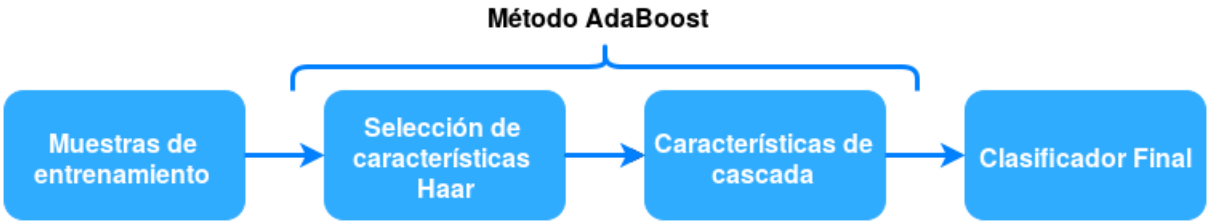


Figura 13: Procedimiento del algoritmo de detección rápida de objetos.

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 explicarán los operadores Haar básicos:

- Las características con dos rectángulos Figura 14(a), Figura 14(b), contienen dos regiones rectangulares adyacentes, el valor de la característica se calcula tomando la diferencia de la suma de ambas regiones.
- Las características con tres rectángulos Figura 14(c), contienen tres regiones rectangulares adyacentes, el valor de la característica se calcula sumando las regiones exteriores y restando la suma de la región interior.

- Las características con cuatro rectángulos Figura 14(d), contienen cuatro regiones rectangulares adyacentes, el valor de la característica se obtiene con la diferencia entre la suma de las regiones pares diagonales.

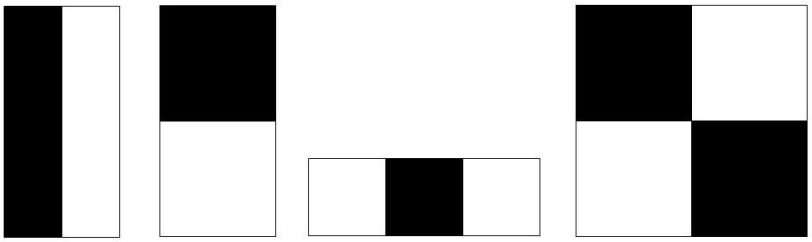


Figura 14: Ejemplo de tipos de operadores Haar.

3.2.1.2. Imagen integral

Uno de los aportes del método desarrollado por Viola y Jones es el concepto de imagen integral con la cual se calcula el valor de las características de manera rápida, es decir en tiempo constante, $O(1)$.

La imagen integral, SI, de una imagen, $S(x, y)$, es calculada como la suma del valor de los pixeles que se encuentran arriba y a la izquierda de cierta posición de la imagen a la cual se le quiere hacer el cálculo. Lo anterior se puede escribir como: [6]

$$SI(x, y) = S(x, y) + S(x - 1, y) + SI(x, y - 1) - SI(x - 1, y - 1). \quad (6)$$

La Figura 15 muestra un ejemplo donde se calcula la imagen integral, Figura 15(b), de la imagen original Figura 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.

[6] https://goo.gl/0oIJBz

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: [7]

$$REG(\alpha) = SI(A) + SI(D) - SI(B) - SI(C), \tag{7}$$

donde $REG(\alpha)$ es la región a la cual se quiere calcular el valor de la suma de sus pixeles; A, B, C, D son las esquinas de dicha región. Como se muestra en la Figura 16, la región α se encuentra resaltada en color azul.

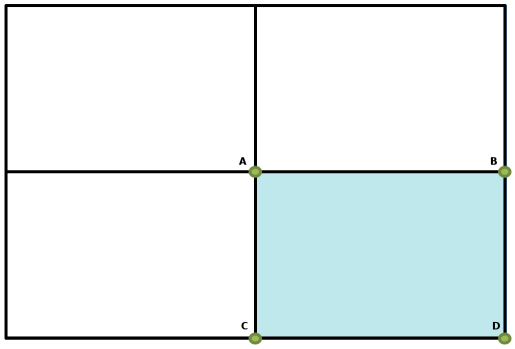


Figura 16: Regiones de la imagen integral.

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. También es utilizado para construir el clasificador final pero en forma de cascada, el cual es explicado en la sección 3.2.1.4.

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

$$h_i(x) = \begin{cases} 1, & si \quad p_i f_i(x) < p_i \theta_i \\ 0, & de \quad otra \quad forma. \end{cases}, \tag{8}$$

[7] https://goo.gl/0oIJBz

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

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

$$h(x) = \alpha_1 h_1(x) + \alpha_2 h_2(x) + \cdots + \alpha_n h_n(x), \tag{9}$$

donde n es el número de características, α_i es el valor asociado a cada característica, el cual va entre 0 y 1. En el Apéndice A se encuentra el algoritmo para calcular el clasificador fuerte.

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.

El clasificador en cascada está compuesto por etapas Figura 17, cada una de estas es un clasificador fuerte. Este clasificador es entrenado por medio de AdaBoost. El cual se encarga de encontrar el orden de evaluación de las características relevantes. La selección se realiza como se muestra en el Algoritmo 2, cumpliendo cierta precisión en la detección D, ver Ecuación 10, cierta tasa de falsos positivos F, ver Ecuación 11.

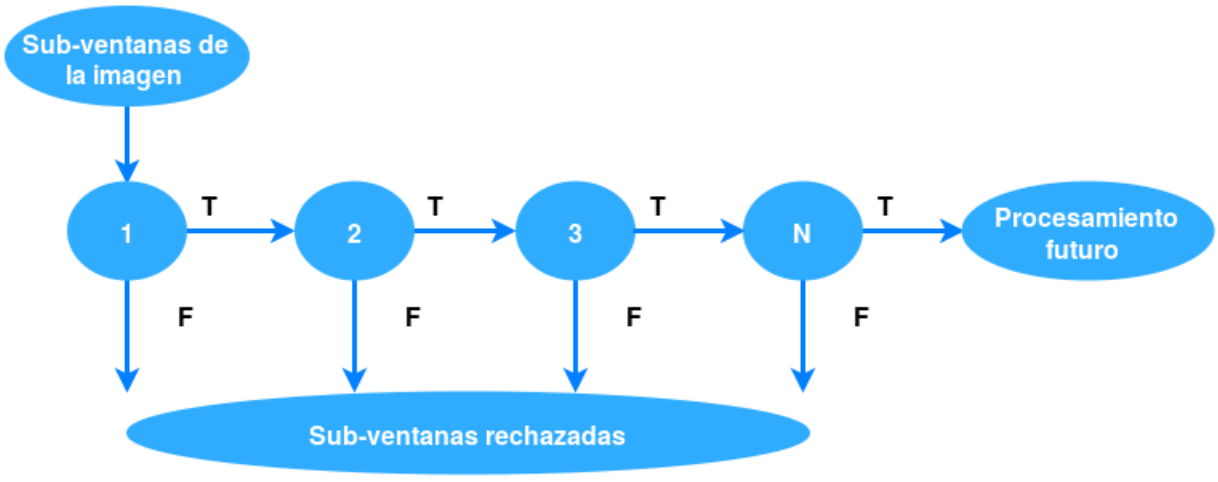


Figura 17: Proceso del clasificador en forma de cascada, donde F representa la tasa de falsos positivos del clasificador de cascada y T representa el número de características.

El proceso de detección funciona de la siguiente manera: la sub-ventana es evaluada

en el primer clasificador; un resultado positivo desencadena la evaluación de un segundo clasificador, el cual también ha sido ajustado para alcanzar cierta precisión; un resultado positivo en el segundo clasificador desencadena una tercera evaluación en el siguiente clasificador y así sucesivamente. Un resultado negativo en cualquier punto del proceso de evaluación conduce al rechazo inmediato de la sub-ventana.

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

$$D = \prod_{i=1}^{K} d_i, \tag{10}$$

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

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

$$F = \prod_{i=1}^{K} f_i, \tag{11}$$

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

El algoritmo para calcular el clasificador en forma de cascada se encuentra en el Apéndice B.

3.2.2. Binarización

La binarización es una técnica de procesamiento de imágenes la cual se encarga de transformar una imagen en escala de grises $S(x, y)$ en una imagen binaria $B(x, y)$. Es decir los pixeles de la imagen toman un valor de 0 ó 1. Para formar la imagen binaria un valor o umbral de la imagen en escala de grises es seleccionado.
Una vez seleccionado el umbral, T, los pixeles de la imagen son discriminados. Si el valor de los pixeles de la imagen es mayor o igual al umbral entonces el valor de los pixeles de la imagen binaria es 1, si no toma el valor de 0. Es decir:

$$B(x, y) = \begin{cases} 1, & Si \quad S(x, y) \geq T \\ 0, & de \quad otra \quad forma \end{cases} . \tag{12}$$

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

El técnica desarrolla por Otsu es un método de binarización global. El algoritmo supone que existen dos clases de pixeles: los del fondo y los que representan el primer plano de la imagen.

El método calcula el umbral optimo T que separa a estas dos clases para el cual la varianza dentro las clases es la mínima. Para calcular T que minimice las varianzas dentro de las clases, se define como una suma ponderada de las varianzas de las dos clases. La varianza ponderada dentro de las clases es:

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t), \tag{13}$$

donde las probabilidades de cada clase son calculadas mediante:

$$q_1(t) = \sum_{i=0}^{t} P(i) \qquad \text{y} \qquad q_2(t) = \sum_{i=t+1}^{255} P(i), \tag{14}$$

y la media de cada clase es calculada como:

$$\mu_1(t) = \sum_{i=0}^{t} \frac{i \cdot P(i)}{q_1(t)} \qquad \text{y} \qquad \mu_2(t) = \sum_{i=t+1}^{255} \frac{i \cdot P(i)}{q_2(t)}. \tag{15}$$

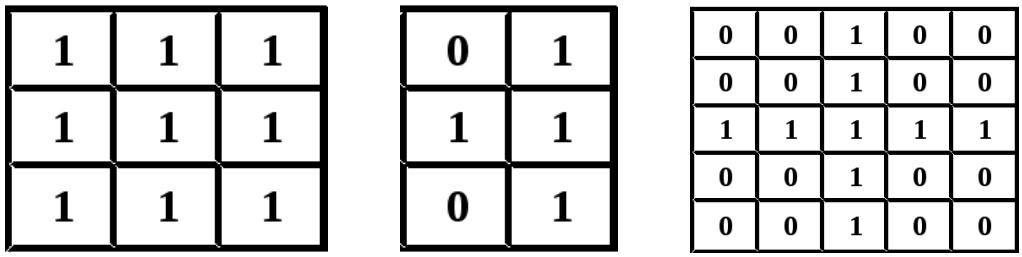
La varianza total σ^2, es igual a la suma de la varianza dentro de clase y la varianza entre clases $\sigma_b^2(t)$, es decir:

$$\sigma^2 = \sigma_w^2(t) + \sigma_b^2(t), \tag{16}$$

donde $\sigma_b^2(t) = q_1(t)\left[1 - q_1(t)\right]\left[\mu_1(t) - \mu_2(t)\right]^2$. Como la varianza total no depende de t, es una constante, entonces para encontrar el umbral que minimice la varianza dentro de clases equivale a encontrar el máximo de $\sigma_b^2(t)$.

3.2.3. Operaciones Morfológicas

Otra técnica muy utilizada en procesamiento de imágenes son las operaciones morfológicas. Que son un conjunto de operaciones no lineales. La idea es que al aplicar alguna de estas operaciones el ruido sea removido tomando en cuenta la forma y estructura de la imagen. Las operaciones morfológicas (Premaratne, 2013) utilizan un elemento estructural el cual se aplica por toda la imagen. Los elementos estructurales pueden ser de distintas formas como los que se muestran en la Figura 18.



(a) Rectángulo de $3x3$. (b) Figura de $3x2$. (c) Cruz de $5x5$.

Figura 18: Ejemplos de elementos estructurales.

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

3.2.3.1. Dilatación

La dilatación es una operación que añade pixeles a la orilla de los objetos que se encuentran en la imagen. En la Figura 19(b) se aplica esta operación a la Figura 19(a). La dilatación se define como:

$$S \oplus EX = \{S | EX_S \subseteq S\}, \tag{17}$$

donde EX_S es el elemento estructural trasladado con la imagen.

3.2.3.2. Erosión

La erosión remueve pixeles a la orilla de los objetos que se encuentran en la imagen. En la Figura 19(c) se muestra el resultado de aplicar la operación a la Figura 19(a). La erosión se define como:

$$S \ominus EX = \{S | EX_S \subseteq S\}, \tag{18}$$

donde EX_S es el elemento estructural trasladado con la imagen.

3.2.3.3. Apertura

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

$$S \circ EX = (S \ominus EX) \oplus EX. \tag{19}$$

La Figura 19(d) muestra el resultado de aplicar la operación apertura a la Figura 19(a).

3.2.3.4. Cierre

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

$$S \bullet EX = (S \oplus EX) \ominus EX. \tag{20}$$

La Figura 19(e) muestra el resultado de aplicar la operación a la Figura 19(a).

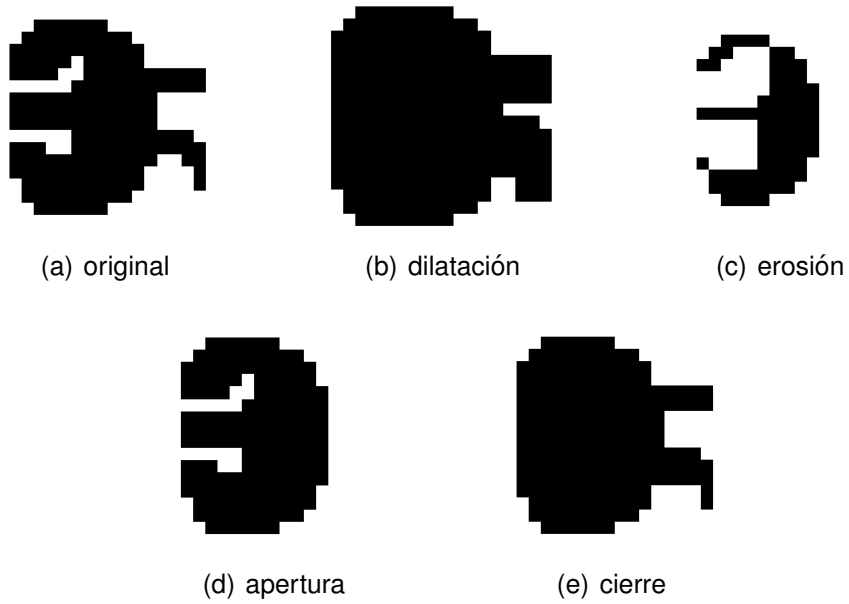


Figura 19: Aplicación de las principales operaciones morfológicas a la imagen que se encuentra en el inciso a), (Smith, 1999).

3.3. Extracción de características

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

En este trabajo se extraen características geométricas, las cuales son extraídas de la siguiente forma, ver Figura 20: el primer paso es encontrar la envolvente convexa de la mano para posteriormente calcular los defectos de convexidad, una vez aplicados estos algoritmos se calcula el número de dedos de la mano entre otras características; finalmente las características calculadas se guardan en un vector.

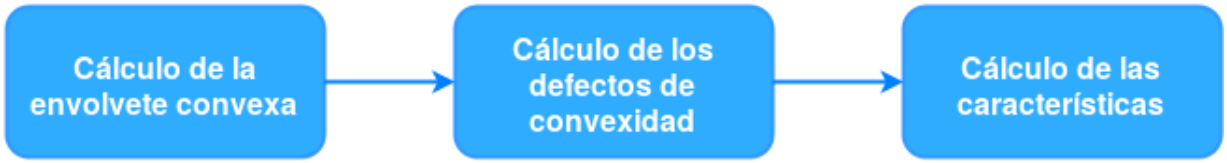
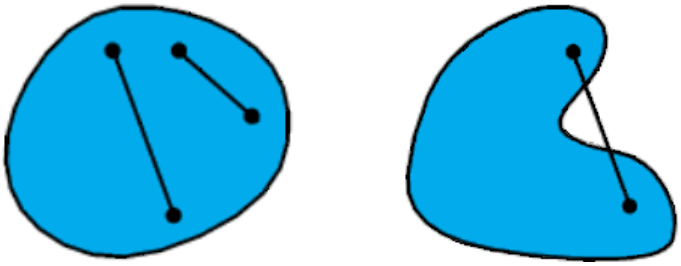


Figura 20: Proceso de la extracción de características.

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

Sea A un conjunto en el espacio euclidiano $\Re^d$, donde d es la dimensión del espacio euclidiano. A es un conjunto convexo [8] si contiene todos los segmentos de línea que unen a cualquier par de puntos pertenecientes al conjunto.



(a) Convexo (b) conexo

Figura 21: Ejemplo de un conjunto conexo y un convexo. Image recuperada de 8.

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 Figura 22 se

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

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 a la envolvente convexa. 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 está compuesto por tres elementos: el punto de inicio del defecto $s_i(x, y)$, el punto con mayor distancia de la envolvente al objeto, $d_i(x, y)$ y el punto final del defecto, $e_i(x, y)$. En la Figura 22 los puntos amarillos representan los puntos de profundidad de los defectos de convexidad.

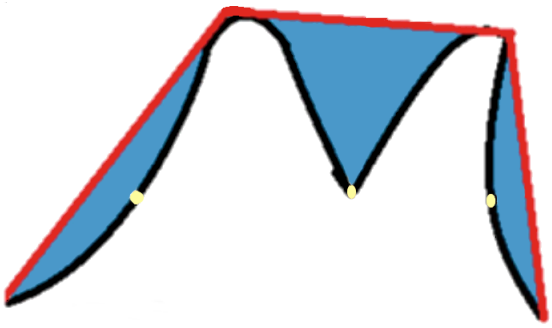


Figura 22: En la imagen se aprecia de color rojo la envolvente convexa, de negro el contorno de la figura y los puntos amarillos son el punto de profundidad de los defectos de convexidad.

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

El número de dedos que se encuentran levantados es calculado con el Algoritmo 3 desarrollado por (Kathuria y Yoshitaka, 2011), ver Apéndice C, el cual utiliza los defectos de convexidad en especifico los conjuntos de puntos de inicio, $\mathcal{S} = \{s_1(x, y), s_2(x, y), \cdots, s_n(x, y)\}$, los puntos de mayor distancia, $\mathcal{D} = \{d_1(x, y), d_2(x, y), \cdots, d_n(x, y)\}$, las distancias del punto de inicio al de mayor distancia, $\delta = \{\delta_1(x, y), \delta_2(x, y), \cdots, \delta_n(x, y)\}$, donde

n es el número total de defectos de la envolvente convexa. Sea $C_r(x, y)$, el punto que representa el centroide del rectángulo más pequeño que encierra a la mano, L_r, la altura del rectángulo y k una constante.

La posición de la raíz de los dedos, $Fr(s, y)$ puede ser calculada usando los defectos de convexidad (Hummel *et al.*, 2014), en especifico los puntos de profundidad, $d(x, y)$. Se calcula tomando el punto medio de los puntos de profundidad consecutivos encontrados en medio de los dedos, como se muestra en la Figura 23. Es decir:

$$Fr_i(x, y) = \left(\frac{x_{d_i} + x_{d_{i-1}}}{2}, \frac{y_{d_i} + y_{d_{i-1}}}{2} \right), \tag{21}$$

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

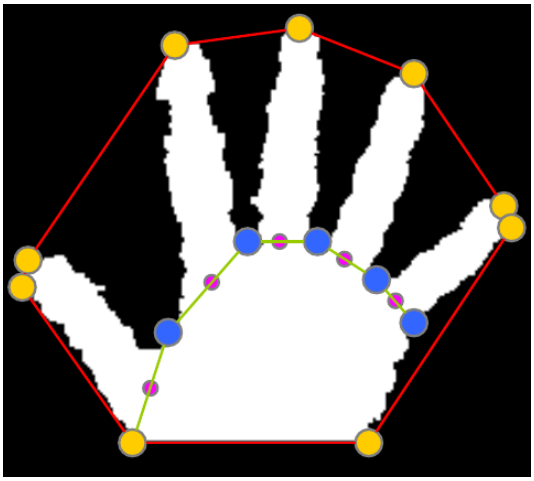


Figura 23: La figura muestra parte de la mano y en ella se aprecia los siguientes elementos: en color rojo la envolvente convexa, en amarillo los puntos de inicio y final de los defectos de convexidad, en color azul los puntos de profundidad de los defectos, en verde la linea que une a los puntos de profundidad consecutivos y finalmente en morado los puntos medios, (Hummel *et al.*, 2014).

El centro de la palma de la mano también se calcula, con los puntos de profundidad de los defectos de convexidad. Se toma como centro de la palma de la mano, el centro del rectángulo más chico que une rodea a los puntos de convexidad.

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

(Sgouropoulos *et al.*, 2014).

$$\angle RC = 90° - \tan^{-1}\left(\frac{y_{Fr} - y_{Ch}}{x_{Fr} - x_{Ch}}\right). \tag{22}$$

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

$$\angle TC = 90° - \tan^{-1}\left(\frac{y_{Ft} - y_{Ch}}{x_{Ft} - x_{Ch}}\right). \tag{23}$$

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

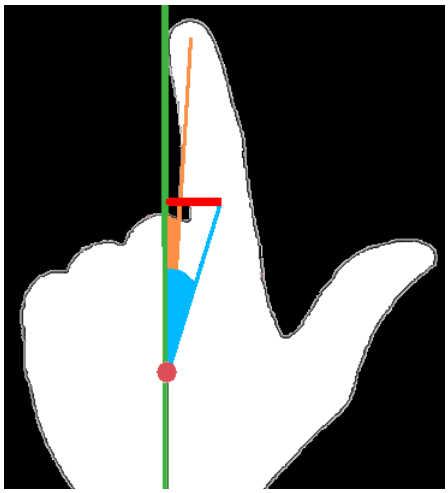


Figura 24: En la imagen se representan los siguientes elementos, el eje vertical con respecto a la mano se encuentra como una línea de color verde; la línea roja represente la distancia del eje vertical a la raíz de los dedos; el punto rosa representa el centro de la palma de la mano, el área azul representa el ángulo de que existe de la línea que une al centro con la raíz de los dedos y finalmente el área anaranjada representa el ángulo que forma la línea del centro a la punta de los dedos, (Sgouropoulos *et al.*, 2014).

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

3.4. Reconocimiento

Es la etapa final del reconocimiento, es donde el gesto realizado por el usuario finalmente puede ser interpretado por la computadora. En este trabajo el reconocimiento se realiza utilizando el algoritmo de máquinas de soporte vectorial SVM, por sus siglas en inglés (Cortes y Vapnik, 1995), un método de aprendizaje de máquina supervisado el cual es utilizado para resolver problemas de clasificación y regresión. SVM tiene como objetivo crear un modelo basado en datos conocidos, datos de entrenamiento, donde este modelo es capaz de predecir a que clase pertenecen datos nuevos.

SVM realiza la clasificación separando las clases calculando el hiperplano que tengan el margen de separación más grande. En la Figura 25 se muestra en conjunto de clases separables por medio de un hiperplano.

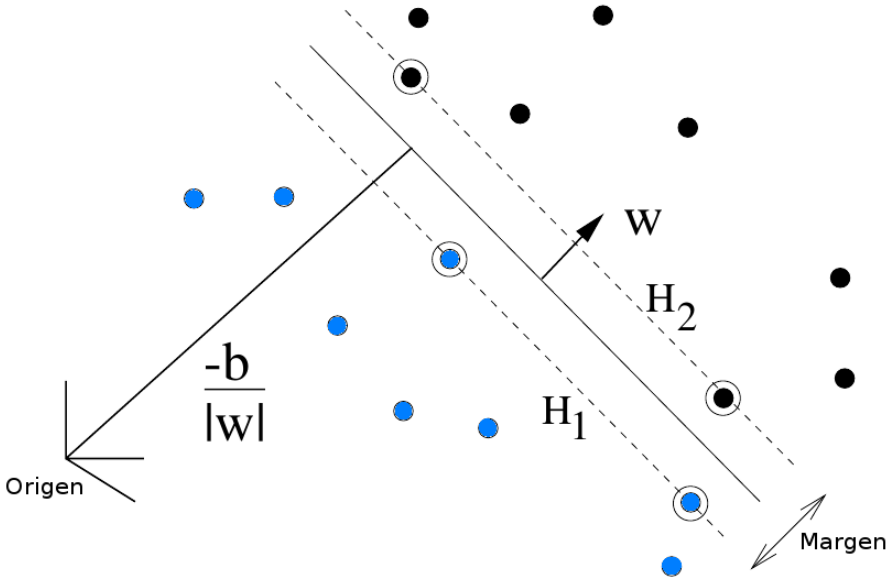


Figura 25: La imagen muestra la separación de dos clases, (los círculos en color azul y negro), mediante un hiperplano óptimo; donde w representa la normal al hiperplano, $\frac{-b}{w}$ la distancia el hiperplano al origen (Burges, 1998).

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

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

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

Sea

$$\mathsf{w} \cdot x + \mathsf{b} = 0, \tag{24}$$

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

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

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

$$\mathsf{w} \cdot x_i + \mathsf{b} \geqslant +1 \quad \text{para} \quad y_i = +1 \tag{25}$$

$$\mathsf{w} \cdot x_i + \mathsf{b} \leqslant -1 \quad \text{para} \quad y_i = -1 \tag{26}$$

Combinando las desigualdades anteriores, se obtiene:

$$y_i(x_i \cdot \mathsf{w} + \mathsf{b}) - 1 \geqslant 0 \qquad \forall i \tag{27}$$

Tomando en cuenta los puntos en donde se cumple la igualdad de la Ecuación 25. Estos puntos se encuentran sobre el hiperplano H_1, el cual se escribe como:

$$\mathsf{w} \cdot x_i + \mathsf{b} = +1, \tag{28}$$

con normal w y una distancia perpendicular desde el origen de $\frac{|1-\mathsf{b}|}{\|\mathsf{w}\|}$. Similarmente para la Ecuación 26, entonces el hiperplano H_2 se describe como:

$$\mathsf{w} \cdot x_i + \mathsf{b} = -1, \tag{29}$$

con normal w y una distancia perpendicular desde el origen de $\frac{|-1-\mathsf{b}|}{\|\mathsf{w}\|}$. Como $d_+ = d_- = \frac{1}{\|\mathsf{w}\|}$, el margen es $\frac{2}{\|\mathsf{w}\|}$. Los hiperplanos son paralelos pues tienen la misma normal, tam-

bién ninguna muestra de entrenamiento caen entre ellos. Entonces se puede encontrar un par de hiperplanos que tengan un margen máximo minimizando $\|\mathsf{w}\|^2$, es decir:

$$\text{min} \quad \frac{1}{2}\|\mathsf{w}\| \quad \text{tal que} \quad y_i(\mathsf{w} \cdot x_i + \mathsf{b}) - 1 \geq 0. \tag{30}$$

Capítulo 4. Implementación del sistema propuesto de reconocimiento de gestos

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

4.1. Adquisición de los datos

Como se vió en el Capítulo 3, Sección 3.1, los datos provienen de los sensores de profundidad de dos dispositivos Kinect. Estos se encuentran ubicados uno frente al usuario (Kinect frontal) y otro al lado izquierdo (Kinect lateral), con una distancia de 74 y 79 cm respectivamente; y entre ellos de 46 cm como se muestra en la Figura 26.

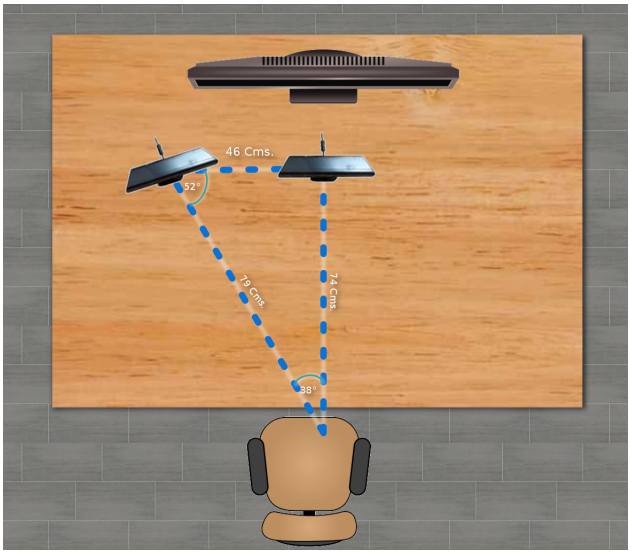


Figura 26: Configuración del sistema de reconocimiento de gestos.

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 pueden 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, Figura 27.

Por la naturaleza del Kinect, las imágenes obtenidas de ambos sensores contienen ruido, como el que se muestra en la Figura 28; el ruido es reducido usando un filtro de

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

mediana, este es aplicado en toda la imagen usando una ventada de tamaño trece. La imagen resultante $S(x, y)$ es como la que se muestra en la Figura 28.

Figura 28: Imagen capturada por el Kinect, a la cual se le aplicó un filtro de mediana.

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

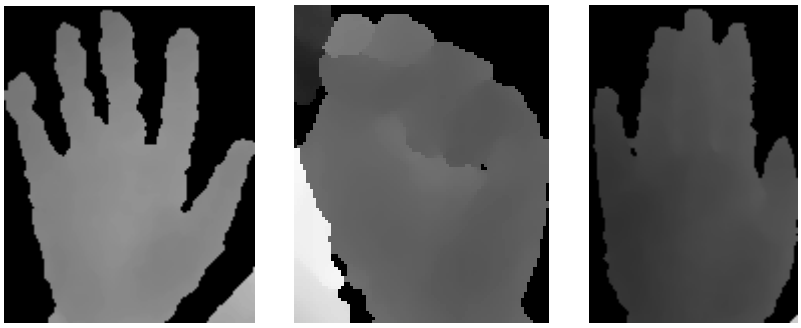
4.2. Detección

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 mil imágenes positivas (imágenes de profundidad de la mano) y dos mil negativas, (imágenes de fondo de distintos escenarios). Las imágenes positivas fueron generadas de trescientos imágenes de poses; tres poses distintas, cien de cada pose, 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 poses de la mano, estas fueron tomadas a una distancia de entre 60 y 200 cm. Las imágenes de profundidad de la mano fueron tomadas de seis personas distintas con tres distintas poses: palma con los dedos separados, como en la Figura 29(a), palma con dedos juntos, ver Figura 29(c) y finalmente el puño, ver Figura 29(b), como se muestran en la Figura 29. Las imágenes de fondo fueron tomadas de distintos escenarios como se muestra en la Figura 30. El programa de captura de las imágenes puede ser encontrado en Github [3].



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

(b) Puño.

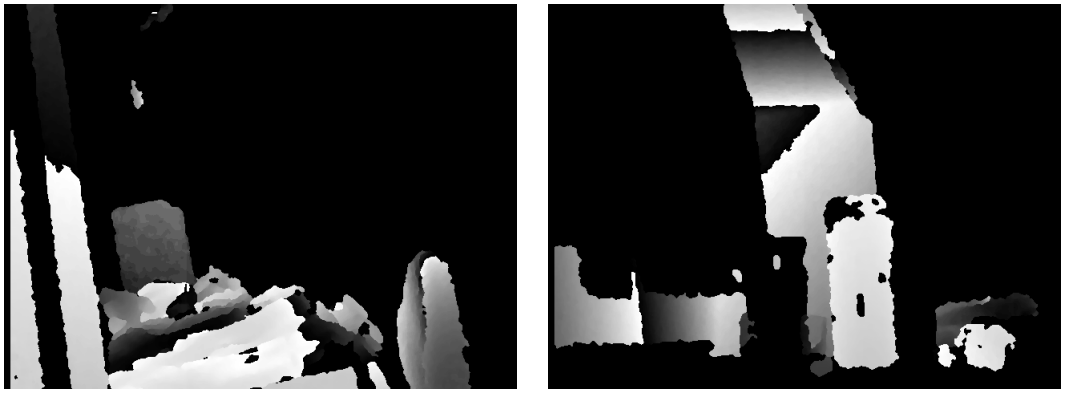
(c) Palma de la mano con los dedos juntos.

Figura 29: Ejemplo de imágenes de poses de nuestra base de datos.

Los parámetros utilizados para la obtención del clasificador final fueron: el porcentaje de precisión de detección de $95\,\%$ y la tasa de falsos positivos aceptados de $5\,\%$. El resultado final del entrenamiento fue en clasificador AdoBoost en forma de cascada, que consta de diecinueve etapas. El clasificador resultante se encuentra en Github [3], en formato XML.

Con el clasificador obtenido, se localiza la mano en cada cuadro proveniente de los

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



(a) Ejemplo de imagen de fondo, donde se encuentra una silla y escritorio.

(b) Ejemplo de imagen de fondo, donde se aprecian objetos en un escritorio.

Figura 30: Imágenes del fondo de nuestra base de datos.

dispositivos Kinect, una ventana inicial de tamaño 60×60 pixeles se desliza por la imagen. Para eliminar falsos positivos que pudieran ocurrir en la detección de la mano se utiliza un algoritmo equivalente al de (Mei *et al.*, 2015), este se muestra en el apéndice D.

Una vez que la mano se localiza, la región de interés $ROI(x, y)$ es seleccionada alrededor de la mano, como se observa en la Figura 31.

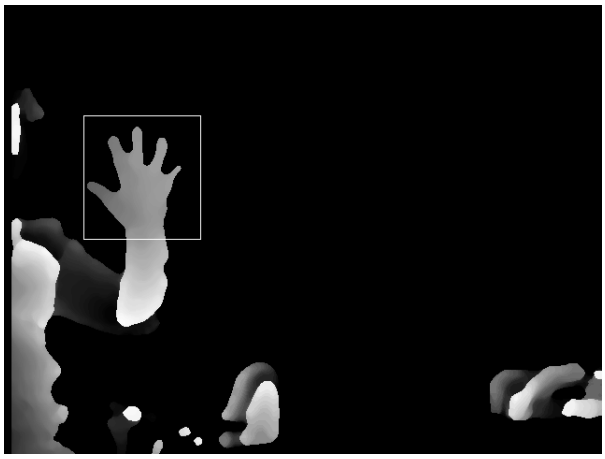


Figura 31: Localización y selección de la mano, en la imagen de entrada del Kinect 2.

Después de localizar el área donde se encuentra la mano, el siguiente paso es segmentar la mano del ROI. La segmentación se realiza binarizando el área del ROI, solo se toma esta área, para que el proceso sea más rápido. La binarización se lleva acabo usando el algoritmo de Otsu, el resultado se muestra en la Figura 32.

Para mejorar la binarización, el ruido existente en la imagen es eliminado aplicando

Figura 32: Binarización de ROI y aplicación de las operaciones morfológicas de la mano localizada en la Figura anterior.

dos operaciones morfológicas, apertura y cierre, en respectivo ese orden. Las operaciones anteriores utilizan un elemento estructural rectangular. Para la operación de apertura el tamaño del elemento es de $3\,x\,9$ pixeles. Para el cierre se aplicó con un tamaño $3\,x\,11$ pixeles.

4.3. Extracción de características

Como se vió en el Capítulo 3, Sección 3.3, las características de la mano son extraídas utilizando los algoritmos de envolvente convexa y defectos de convexidad.

Una vez aplicados estos dos algoritmos se calcula: el número de dedos levantados, la posición de la punta de los dedos, la posición de la raíz de los dedos, el centro de la palma de mano, los ángulos que existe del centro de la mano a la punta de los dedos, los ángulos que existen del centro a la raíz de los dedos, la distancia que existe del centro de los dedos a la raíz de los dedos. En la Figura 33 se muestran algunas de estas características. Para la implementación de la extracción de las características, se tomó parte del código proveniente de la página de internet [4].

Las características calculadas se guardan en un vector de características de dimensión 32. El valor de la dimensión del vector está dada por la unión del conjunto de características obtenidas por cada imagen de la mano. En cada vector las características provenientes del Kinect frontal son almacenadas primero, seguidas de las del Kinect lateral.

[4] `http://goo.gl/2R7Cg`

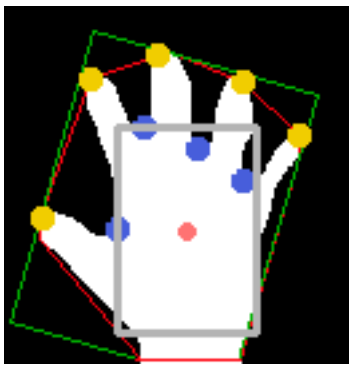


Figura 33: La imagen muestra algunas características de la mano. El contorno rojo representa la envolvente convexa, el rectángulo verde es el rectángulo que rodea a la mano, el rectángulo gris representa el área de la palma de la mano, los círculos en color amarillo la punta de los dedos, en color azul se encuentran los puntos de profundidad encontrados en medio de los dedos, en rosa el centro de la palma de la mano.

4.4. Reconocimiento

En este trabajo se reconocen gestos estáticos y dinámicos utilizando el método de maquinas de soporte vectorial.

Como se redactó 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. Con el conjunto de imágenes el clasificador es entrenado y el modelo de clasificación puede ser calculado.

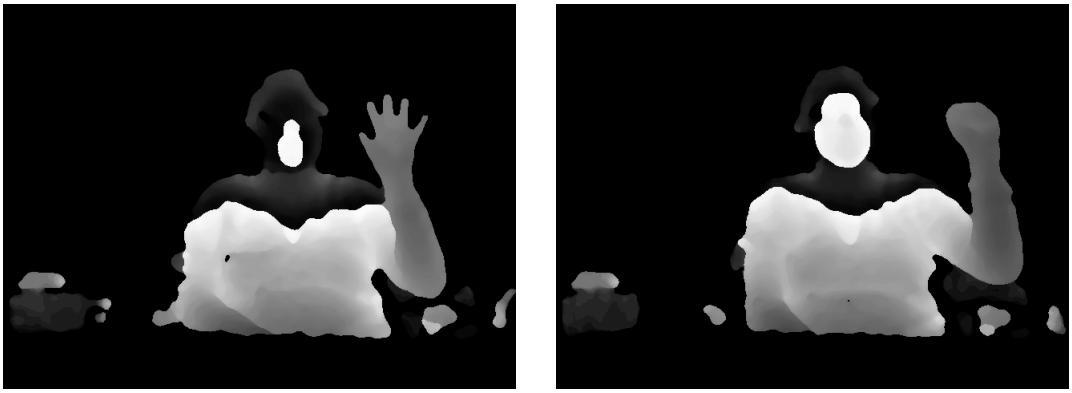
La implementación de SVM se lleva acabo usando LibSVMSharp [5] una adaptación de la librería LibSVM (Chang y Lin, 2011).

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. El reconocimiento del gesto se lleva analizando un solo cuadro o imagen.

El modelo que reconoce los gestos fue creado mediante SVM. Para el entrenamiento se tomaron doscientas imágenes de las dos distintas poses como las que se muestran en la Figura 34. Se utilizó un kernel exponencial y validación cruzada con cinco pliegues.

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



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

(b) Puño.

Figura 34: Ejemplo de imágenes de poses de nuestra base de datos.

4.4.2. Reconocimiento de gestos dinámicos

En este trabajo, se toma un gesto dinámico como una secuencia de treinta cuadros consecutivos en los cuales la misma pose es realizada. La diferencia entre los gestos estáticos es que la pose cambia de posición, es decir se encuentra en movimiento. De manera que el sistema reconoce el gesto dinámico identificando y siguiendo cada pose en cada cuadro. La trayectoria del gesto es obtenida tomando en cuenta la posición del centro de la mano el cual es calculado en cada cuadro.

El sistema reconoce dos gestos dinámicos, los cuales son los gestos estáticos en movimiento. La Figura 35 muestra dos secuencias del gesto dinámico de la palma de la mano. La Figura 36 muestra dos secuencias del gesto dinámico del mano en forma de puño.

(a) Cuadro inicial (b) Cuadro número 15 (c) Cuadro número 30 (d) Cuadro número 45 (e) Cuadro final

Figura 35: Secuencia del gesto dinámico de la palma de la mano con los dedos separados, la vista es desde el Kinect frontal.

(a) Cuadro inicial (b) Cuadro número 15 (c) Cuadro número 30 (d) Cuadro número 45 (e) Cuadro final

Figura 36: Secuencia del gesto dinámico del puño, la vista es desde el Kinect frontal.

Capítulo 5. Resultados

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

Los experimentos y el sistema propuesto fue implementado en una computadora de escritorio Dell con un procesador Intel(R) Xeon(R) CPU E5-1603, 16GB de memoria RAM, Windows 7 de 64 bits. La implementación del sistema se realizó en Microsoft C# utilizando Emgu 2.410 [1] una adaptacion de OpenCV[2] para C#.

Para analizar la tasa de precisión de reconocimiento del sistema propuesto se realizaron varios experimentos en distintas circunstancias. Se analizó el rendimiento del sistema a tres diferentes distancias 70, 80, $90\ cm$ del Kinect frontal. También en diferentes circunstancias de iluminación, con iluminación estándar, media y sin iluminación. Se escogió este tipo de escenarios para evaluar el sistema, para obtener el rendimiento en situaciones que representen o estén en condiciones naturales o reales.

Se utilizaron imágenes reales capturadas por ambos sensores de profundidad del Kinect de 640×480 pixeles. Las imágenes son cinco personas distintas, realizando los gestos de puño y el de palma de la mano con los dedos separados, para los gestos estáticos. Para los gestos dinámicos se tomaron los gestos de tres personas cada una de ellas realizando los gestos estáticos en movimiento.

Los gestos se realizaron usando una sola mano, en este caso todos los usuarios optaron por usar la mano derecha. Pero el sistema está hecho para que funcione sin importar la mano que se utilice. La mano del usuario tiene que tener una ligera rotación en el eje vertical para que el sistema pueda detectar el gesto.

En cada experimento se analizó el rendimiento del sistema usando un solo dispositivo, el Kinect frontal y usando los dos dispositivos.

5.1. Experimentos de gestos estáticos

La evaluación del sistema en cuanto al reconocimiento de los gestos estáticos, se determinó conforme al resultado de los experimentos realizados en circunstancias de ilu-

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

minación, (estándar, media, baja). En cada conjunto de experimentos se tomó en cuenta la distancia, ya que en cada grupo se analizaron tres diferentes distancias: 70, 80, 90 cm.

Se analizaron dos gestos estáticos, la palma de la mano con los dedos separados, Gesto 1 y el puño, Gesto 2, de cinco usuarios distintos. Para cada experimento se escogió al azar doscientas imágenes de cada gesto del conjunto de las imágenes capturadas. Cada imagen de un gesto tenia su correspondiente imagen con el Kinect lateral, de manera que eran cuatrocientas imágenes por cada gesto.

En el análisis del sistema con un solo Kinect, solo se tomaron las imágenes provenientes del Kinect frontal. Para el análisis con dos Kinect se tomaron estas mismas imágenes del Kinect frontal con su correspondiente imagen del Kinect lateral.

Enseguida se presentan los resultados de cada experimento realizado.

5.1.1. Experimentos con iluminación

Para este experimento se usaron imágenes capturadas en un laboratorio con iluminación estándar, como la que se muestra en la Figura 37. Se tomaron en cuenta tres distancias distintas.

Figura 37: Laboratorio en condiciones estándar de iluminación.

- En el primer experimento el usuario se encuentra a una distancia de 70 cm del Kinect frontal. En la Tabla 1 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 2 los resultados usando un dispositivo.

Tabla 1: Matriz de confusión del experimento con iluminación estándar, a una distancia de 70 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 170 gestos de la clase 1 y 179 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $87.25\,\%$

Tabla 2: Matriz de confusión del experimento con iluminación estándar, a una distancia de 70 cm utilizando el Kinect frontal.

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

La matriz de confusión muestra que 127 gestos de la clase 1 y 194 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $80.25\,\%$.

Como se observa en las matrices de confusión, se obtiene una mayor exactitud en el reconocimiento del gesto utilizando dos dispositivos Kinect.

- En el segundo experimento el usuario está una distancia de $80\ cm$ del Kinect frontal. En la Tabla 3 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 4 los resultados usando un dispositivo.

Tabla 3: Matriz de confusión del experimento con iluminación estándar, a una distancia de 80 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 96 gestos de la clase 1 y 185 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $70.25\,\%$

Tabla 4: Matriz de confusión del experimento con iluminación estándar, a una distancia de 80 cm utilizando el Kinect frontal.

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

La matriz de confusión muestra que 88 gestos de la clase 1 y 183 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $70.5\,\%$

En este experimento la exactitud del reconocimiento es baja para el uso de ambos o un dispositivo Kinect. Se observa, que clasifica los gestos de la clase uno como de la clase dos, esto es debido a la calidad de la imágenes debido a que el sensor no siempre proporciona información detallada o completa de la mano.

- En el tercer experimento el usuario esta a una distancia de $90\ cm$ del Kinect frontal. En la Tabla 5 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 6 los resultados usando un dispositivo.

Tabla 5: Matriz de confusión del experimento con iluminación estándar, a una distancia de 90 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 93 gestos de la clase 1 y 190 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $70.75\,\%$

Tabla 6: Matriz de confusión del experimento con iluminación estándar, a una distancia de 90 cm utilizando el Kinect frontal.

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

La matriz de confusión muestra que 101 gestos de la clase 1 y 183 de la clase 2

fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $71\,\%$

En este experimento se observa un resultado similar al anterior, pues otra vez se obtiene baja exactitud en el reconocimiento porque clasifica mal el Gesto 1. Esto por la misma razón del experimento anterior.

5.1.2. Experimentos con iluminación media

Para el conjunto de estos experimentos, las imágenes se capturaron en un laboratorio con iluminación media, como la que se muestra en la Figura 38. A dos distancias 70 y 90 cm.

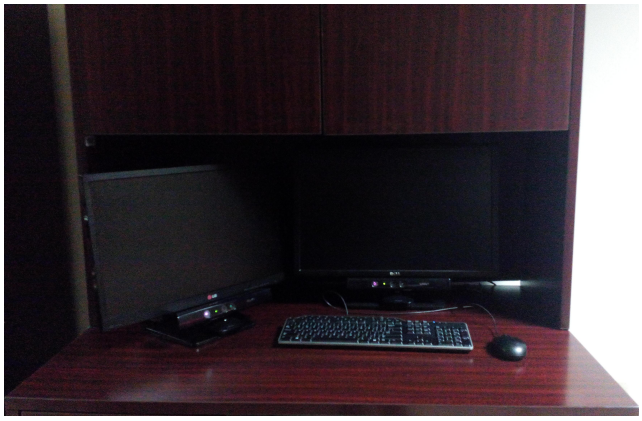


Figura 38: Laboratorio en condiciones con iluminación media.

- En el primer experimento el usuario esta a una distancia de 70 cm del Kinect frontal. En la Tabla 7 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 8 los resultados usando un dispositivo.

Tabla 7: Matriz de confusión del experimento con iluminación media, a una distancia de 70 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 178 gestos de la clase 1 y 105 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $72.75\,\%$

Tabla 8: Matriz de confusión del experimento con iluminación media, a una distancia de 70 cm utilizando el Kinect frontal.

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

La matriz de confusión muestra que 136 gestos de la clase 1 y 193 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $82.25\,\%$

En este caso se observa una mejor exactitud cuando se utiliza solo un Kinect.

- En el segundo experimento el usuario está a una distancia de $90\ cm$ del Kinect frontal. En la Tabla 9 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 10 los resultados usando un dispositivo.

Tabla 9: Matriz de confusión del experimento con iluminación media, a una distancia de 90 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 153 gestos de la clase 1 y 174 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $81.75\,\%$

Tabla 10: Matriz de confusión del experimento con iluminación media, a una distancia de 90 cm utilizando el Kinect frontal.

	Gesto 1	**Gesto 2**
Gesto 1	54	95

En este caso cuando solo se utilizó un Kinect, éste no pudo identificar el Gesto 2, ni tampoco todos los gestos pertenecientes al Gesto 1. Debido a la posición que se encontraba la mano, solo se detectaron 149 gestos de la clase uno de los cuales 54 fueron clasificados correctamente.

El experimento muestra que el uso de dos Kinect brinda en algunos casos mayor exactitud en el reconocimiento. Debido a que se tiene otra perspectiva de la mano.

5.1.3. Experimentos sin iluminación

Para estos experimentos las imágenes se capturaron en un laboratorio sin iluminación, como la que se muestra en la Figura 39.

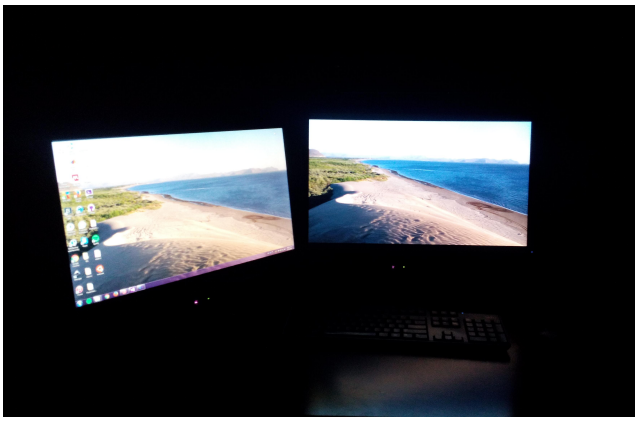


Figura 39: Laboratorio en condiciones con baja iluminación.

- En el primer experimento el usuario esta a una distancia de $70\ cm$ del Kinect frontal. En la Tabla 11 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 12 los resultados usando un dispositivo.

Tabla 11: Matriz de confusión del experimento sin iluminación, a una distancia de 70 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 190 gestos de la clase 1 y 4 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $48.5\,\%$

La matriz de confusión muestra que 154 gestos de la clase 1 y 133 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $46.75\,\%$

Tabla 12: Matriz de confusión del experimento sin iluminación, a una distancia de 70 cm utilizando el Kinect frontal.

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

- En el segundo experimento el usuario está a una distancia de $80\ cm$ del Kinect frontal. En la Tabla 13 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos, y en la Tabla 14 los resultados usando un dispositivo.

Tabla 13: Matriz de confusión del experimento sin iluminación, a una distancia de 80 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 187 gestos de la clase 1 y 110 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $74.25\,\%$

Tabla 14: Matriz de confusión del experimento sin iluminación, a una distancia de 80 cm utilizando el Kinect frontal.

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

La matriz de confusión muestra que 162 gestos de la clase 1 y 166 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $88.8\,\%$

En este caso se obtiene una mayor exactitud utilizando un Kinect, sin embargo se observa que el uso de dos dispositivos ayuda a clasificar mejor el Gesto 1, pues se tiene mayor información con la cual se puede saber si es una palma con los dedos extendidos o solo es ruido proveniente del sensor.

- En el tercer experimento el usuario está a una distancia de $90\ cm$ del Kinect frontal. En la Tabla 15 se encuentran los resultados del reconocimiento de los dos gestos utilizando los dos dispositivos y en la Tabla 16 los resultados usando un dispositivo.

Tabla 15: Matriz de confusión del experimento sin iluminación, a una distancia de 90 cm utilizando ambos Kinect.

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

La matriz de confusión muestra que 150 gestos de la clase 1 y 194 de la clase 2 fueron clasificados correctamente. De manera que se obtuvo una tasa de exactitud de $86\,\%$

Tabla 16: Matriz de confusión del experimento sin iluminación, a una distancia de 90 cm utilizando el Kinect frontal.

	Gesto 1	**Gesto 2**
Gesto 1	138	61

En el caso cuando se toma como entrada solo un Kinect, al igual que en el experimento de iluminación media y con distancia también de $90\ cm$. Solo se detectaron los gestos de la clase 1, de los cuales se clasificaron correctamente 138 de 199 detectados.

En este experimento se observa que hay mayor exactitud en el reconocimiento utilizando dos Kinect y al igual que en la mayoría de los casos, hay un error mayor en la clasificación de la clase 1.

Los experimentos explicados anteriormente muestran que el Gesto 1 produce un error mayor, debido a la resolución de sensor, pues no en todas las imágenes la mano está completa, en los casos que no está completamente vista por el sensor frontal y logra ser detectada por el sensor lateral con información favorable, es cuando se puede apreciar una mayor exactitud de reconocimiento usando ambos Kinect. Sería conveniente hacer un experimento donde las imágenes tengan una buena resolución y ninguna obstrucción para validar el rendimiento del sistema. Debido a que la imágenes utilizadas no

eran ideales pues contenían ruido o obstrucciones. Se decidió hacer este tipo de pruebas para lograr resultados más apegados a la realidad es decir en ambientes naturales.

5.2. Experimentos de gestos dinámicos

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

Los gestos analizados fueron dos. El primero corresponde a la palma de la mano con los dedos separados, Gesto 3, en movimiento y el segundo el puño de la mano, Gesto 4, también en movimiento. Los gestos fueron realizados por tres usuarios distintos. Para cada experimento se realizaron cinco repeticiones de cada gesto y cada uno tenia una duración de treinta cuadros por segundo. Se tomó como gesto válido si el porcentaje de reconocimiento de cada gestos estático presente en cada segundo es mayor o igual a $80\,\%$, esto con base al trabajo realizado por (Sultana y Rajapuspha, 2012). También se probó el sistema usando un solo Kinect, el frontal, en las mismas circunstancias mencionadas anteriormente. Enseguida se presentan los resultados de cada experimento realizado, por usuario.

5.2.1. Experimentos con iluminación media

Para el conjunto de estos experimentos, las imágenes se capturaron en un laboratorio con iluminación media, véase la Figura 38.

- En el primer experimento el usuario está a una distancia de $70\ cm$ del Kinect frontal. En la Tabla 17 se encuentran los resultados del reconocimiento de los dos gestos utilizando solo el Kinect frontal. En la Tabla 18 se presentan los resultados utilizando los dos dispositivos Kinect.

  En la tabla 17 se observa que arriba del $90\,\%$ de los gestos realizados por cada participante son reconocidos correctamente. A excepción del participante uno al realizar el Gesto 3, pues solo reconoció dos gestos dinámicos de los cinco realizados. Esto

Tabla 17: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 70 cm utilizando el Kinect frontal. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	77	80	83	73	70
	G4	93	100	100	93	97
P2	G3	87	80	87	70	97
	G4	97	93	97	93	93
P3	G3	83	100	80	53	90
	G4	100	93	93	87	100

se debe al sensor, pues las manos del participante uno no son tan grandes y el sensor no logra captar en todas la imágenes los dedos del participante.

Tabla 18: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 70 cm utilizando ambos Kinect. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	73	63	77	73	63
P2	G3	100	93	87	93	87
	G4	93	77	77	67	60
P3	G3	100	100	93	97	93
	G4	73	90	83	63	83

Al utilizar ambos Kinect se observa que hay menor exactitud de reconocimiento de los gestos. Cómo se usan las mismas imágenes para uno y dos Kinect en cada prueba, se tiene la misma justificación en la exactitud del participante uno.

- En el segundo experimento el usuario está a una distancia de $80\ cm$ del Kinect frontal. En la Tabla 19 se encuentran los resultados del reconocimiento de los dos gestos utilizando solo el Kinect frontal. En la Tabla 20 se presentan los resultados utilizando los dos dispositivos Kinect.

  Para ambas pruebas con uno y dos Kinect se obtuvieron resultados similares, desafortunadamente hubo mayores casos de reconocimiento fallido, en especial cuando

Tabla 19: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 80 cm utilizando el Kinect frontal. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

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

Tabla 20: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 80 cm utilizando ambos Kinect. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	83	86	67	87	70
	G4	35	53	70	77	57
P2	G3	33	57	57	73	60
	G4	93	93	90	90	80
P3	G3	93	100	67	100	80
	G4	100	93	87	83	90

se realiza el Gesto 3. Aquí se puede observar un mejor reconocimiento con el uso de ambos Kinect.

- En el tercer experimento el usuario está a una distancia de $90\ cm$ del Kinect frontal. En la Tabla 21 se encuentran los resultados del reconocimiento de los dos gestos utilizando solo el Kinect frontal. En la Tabla 22 se presentan los resultados utilizando los dos dispositivos Kinect.

  En este caso el experimento mostró un mejor reconocimiento al usar un solo dispositivo, si existe gran diferencia en el uso de dos dispositivos generalmente en la realización del Gesto 3.

Tabla 21: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 90 cm utilizando el Kinect frontal. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	87	87	53	87	67
	G4	100	100	100	90	92
P2	G3	87	87	90	93	80
	G4	83	73	97	80	75
P3	G3	90	97	97	100	90
	G4	77	90	80	75	80

Tabla 22: Precisión de gestos realizados en un ambiente de iluminación media a una distancia de 90 cm utilizando ambos Kinect. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	30	37	13	13	30
	G4	100	98	95	100	98
P2	G3	50	60	67	70	75
	G4	77	78	75	80	70
P3	G3	77	73	80	97	90
	G4	77	90	91	80	75

5.2.2. Experimentos sin iluminación

Para este experimento las imágenes se capturaron en un laboratorio sin iluminación, véase la Figura 39.

- En el primer experimento el usuario está a una distancia de $70\ cm$ del Kinect frontal. En la Tabla 23 se encuentran los resultados del reconocimiento de los dos gestos utilizando solo el Kinect frontal. En la Tabla 24 se presentan los resultados utilizando los dos dispositivos Kinect.

  Analizando cada repetición de cada gesto se observa que una mayor reconocimiento cuando se utilizan los dos dispositivos, en especial cuando se reconoce el Gesto 3 de los participantes uno y tres.

Tabla 23: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 70 cm utilizando el Kinect frontal. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	77	80	57	70	60
	G4	100	97	93	100	97
P2	G3	97	57	80	53	87
	G4	100	97	83	87	87
P3	G3	73	43	43	47	60
	G4	100	93	97	93	93

Tabla 24: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 70 cm utilizando ambos Kinect. P1, P2, P3 representan a los participantes, R1, R2, R3, R4, R5 representan el número de repeticiones

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	93	83	100	77	87
	G4	100	100	97	97	100
P2	G3	97	97	87	93	97
	G4	80	93	90	97	97
P3	G3	83	87	97	97	100
	G4	100	97	93	87	90

- En el segundo experimento el usuario está a una distancia de $80\ cm$ del Kinect frontal. En la Tabla 25 se encuentran los resultados del reconocimiento de los dos gestos utilizando solo el Kinect frontal. En la Tabla 26 se presentan los resultados utilizando los dos dispositivos Kinect.

  En este experimento se puede observar que las repeticiones del Gesto 4 con cada participante obtienen mayor precisión utilizando un Kinect. En este caso se observa que el Gesto 3 no es clasificado correctamente en la mayoría de las repeticiones de los tres participantes a excepción del participante tres, que muestra mayor desempeño cuando se utilizan los dos dispositivos.

- En el tercer experimento el usuario está a una distancia de $90\ cm$ del Kinect frontal. En la Tabla 27 se encuentran los resultados del reconocimiento de los dos gestos

Tabla 25: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 80 cm utilizando el Kinect frontal. P1, P2, P3 representan a los participantes, R1, R2, R3, R4, R5 representan el número de repeticiones

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	57	63	60	63	63
	G4	100	100	100	100	100
P2	G3	40	43	60	70	75
	G4	100	93	97	93	98
P3	G3	80	87	67	77	63
	G4	97	87	100	90	87

Tabla 26: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 80 cm utilizando ambos Kinect. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	63	70	90	80	77
	G4	87	100	93	90	93
P2	G3	60	73	47	63	73
	G4	100	97	87	90	90
P3	G3	87	100	73	83	80
	G4	73	73	93	83	93

utilizando solo el Kinect frontal. En la Tabla 28 se presentan los resultados utilizando los dos dispositivos Kinect.

Tabla 27: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 90 cm utilizando el Kinect frontal. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	57	70	70	67	77
P2	G3	70	80	80	77	80
	G4	73	80	83	97	80
P3	G3	70	83	80	100	97
	G4	87	97	90	70	90

Tabla 28: Precisión de gestos realizados en un ambiente sin iluminación a una distancia de 90 cm utilizando ambos Kinect. P1, P2 y P3 representan a los participantes, G3 y G4 representan el Gesto 3 y Gesto 4 respectivamente, R1, R2, R3, R4 y R5 representa el número de repeticiones.

Porcentajes de precisión del reconocimiento de cada repetición.						
		R1	**R2**	**R3**	**R4**	**R5**
P1	G3	47	47	53	37	40
P2	G3	37	63	63	75	80
	G4	100	98	95	90	92
P3	G3	80	97	87	83	87
	G4	87	100	97	78	95

En las Tablas 27, 28 no se muestra ningún valor para el Gesto 4 del participante uno porque el Gesto 4 no pudo ser ubicado, debido a las imágenes obtenidas. Observando las repeticiones de cada participante se observa un mejor desempeño para el participante tres usando ambos dispositivos.

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

También se observa que en algunas ocasiones el reconocimiento es bajo, usualmente para el Gesto 3, debido a la resolución de sensor, ya que en ocasiones las regiones de los dedos no son captadas y esto hace que reconozca erróneamente el gesto.

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

5.3. Comparación con estado del arte.

En el estado del arte los sistemas se enfocan al reconocimiento en ambientes ideales, o por lo menos las pruebas se realizan bajo esos estándares. De manera que resulta complicado comparar los resultados obtenidos con los existentes.

Los experimentos realizados se formularon para validar el sistema en un ambiente na-

tural, de manera que las imágenes utilizadas no siempre contenían información adecuada o completa para la clasificación del gesto. Tomando en cuenta los mejores resultados el sistema obtuvo un $88\,\%$ en el reconocimiento de gestos estáticos y para los dinámicos se presentó un $100\,\%$ en ciertos participantes para las repeticiones presentadas.

El trabajo realizado por Caputo (véase el Capítulo 1, Sección 1.7) utiliza también dos dispositivos Kinect como medio de captura. Ellos logran obtener hasta un $85\,\%$ en el reconocimiento de los gestos, en los gestos dinámicos no mencionan el porcentaje de exactitud. Las condiciones en que es probado su sistema no son mencionadas.

Nuestro sistema logra superar el porcentaje de reconocimiento que se encuentra en el estado del arte, pese a las condiciones en que nuestro sistema es probado.

Capítulo 6. Conclusiones

El reconocimiento de gestos basando en el modelo de la visión es un problema muy complejo ya que hay varios aspectos que se tienen que tomar en cuenta, el dispositivo de captura, la resolución de este, la variación en la intensidad de la luz del ambiente, el fondo, el tamaño de la mano, su color y la rotación que éstas pueda presentar. Tomando en cuenta que el reconocimiento debe funcionar en tiempo real para que sea escalable a aplicaciones que pueden ser utilizadas en la vida diaria.

En este trabajo se abordó el reconocimiento de gestos proponiendo una sistema que utiliza como media de captura dos dispositivos Kinect, en especifico los sensores de profundidad de estos dispositivos. La idea de utilizar estos dispositivos es que el reconocimiento de gestos funcione en condiciones bajas de iluminación y cuando exista obstrucción de la mano.

Los resultados sugieren que, como era de esperarse, en algunas situaciones el reconocimiento mejora con la utilización de dos dispositivos. Estas situaciones son donde la mano no es vista por el Kinect principal o la información proporcionada por este dispositivo no es suficiente para discriminar el gesto.

Los experimentos fueron realizados en ambientes naturales, de manera que no siempre se obtuvo un alto grado de exactitud en el reconocimiento pero se llegó a obtener hasta $88\,\%$ para los gestos estáticos y para los dinámicos un $100\,\%$ de reconocimiento en secciones de cinco segundos. Los experimentos sugieren que en situaciones controladas la tasa de reconocimiento aumentaría considerablemente.

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

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

6.1. Trabajo futuro

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

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

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

6.2. Trabajo derivado de esta tesis

Publicación de articulo y presentación de póster en el congreso SPIE Optics and Photonics 2015.

Lista de referencias bibliográficas

Asaari, M. S. M., Rosdi, B. A., y Suandi, S. A. (2014). Adaptive Kalman Filter Incorporated Eigenhand (AKFIE) for real-time hand tracking system. *Multimedia Tools and Applications*, pp. 1–27.

Burges, C. (1998). A Tutorial on Support Vector Machines for Pattern Recognition. *Data Mining and Knowledge Discovery*, **2**(2): 121–167.

Caputo, M., Denker, K., Dums, B., y Umlauf, G. (2012). 3D Hand Gesture Recognition Based on Sensor Fusion of Commodity Hardware. *Mensch & Computer 2012: interaktiv informiert – allgegenwärtig und allumfassend!?*, pp. 293–302.

Chaki, N., Shaikh, S. H., y Saeed, K. (2014). *Exploring Image Binarization Techniques*. Springer India. p. 90.

Chang, C.-C. y Lin, C.-J. (2011). Libsvm. *ACM Transactions on Intelligent Systems and Technology*, **2**(3): 1–27.

Chang, F., Chen, C. J., y Lu, C. J. (2004). A linear-time component-labeling algorithm using contour tracing technique. *Computer Vision and Image Understanding*, **93**(2): 206–220.

Chih-Wei Hsu, Chih-Chung Chang y Lin, C.-J. (2008). A Practical Guide to Support Vector Classification. *BJU international*, **101**(1): 1396–400.

Cortes, C. y Vapnik, V. (1995). Support-vector networks. *Machine Learning*, **20**(3): 273–297.

Cruz, L., Lucio, D., y Velho, L. (2012). Kinect and RGBD Images: Challenges and Applications. En: *2012 25th SIBGRAPI Conference on Graphics, Patterns and Images Tutorials*. IEEE, pp. 36–49.

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters*, **27**(8): 861–874.

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*. Pearson Education, Inc. p. 190.

Hasan, M. M. y Mishra, P. K. (2012). Hand Gesture Modeling and Recognition using Geometric Features : A Review. *Canadian Journal on Image Processing and Computer Vision*, **3**(1): 12–26.

Huang, D.-Y., Hu, W.-C., y Chang, S.-H. (2011). Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination. *Expert Systems with Applications*, **38**(5): 6031–6042.

Hummel, S., Häfner, V., Häfner, P., y Ovtcharova, J. (2014). New Techniques for Hand Pose Estimation Based on Kinect Depth Data. En: *EuroVR 2014 - Conference and Exhibition of the European Association of Virtual and Augmented Reality*. The Eurographics Association.

Ibraheem, N. A. (2013). Comparative Study of Skin Color based Segmentation Techniques. *International Journal of Applied Information Systems*, **5**(10): 24–38.

Jana, A. (2013). *Kinect for Windows SDK - Programming Guide - Face Tracking*. Packt. p. 392.

Kang, C., Bernhard, P., Kim, S., Srinivasa, P., y Satti, R. (2013). A Framework for Hand Gesture Recognition with Machine Learning Techniques.

Kathuria, P. y Yoshitaka, A. (2011). Hand Gesture Recognition by using Logical Heuristics. *Research report Human-Computer Interaction (HCI)*, **2012**(25): 63–69.

Maimone, A. y Fuchs, H. (2011). Encumbrance-free telepresence system with real-time 3D capture and display using commodity depth cameras. *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 137–146.

Mallick, T., Das, P. P., y Majumdar, A. K. (2014). Characterizations of Noise in Kinect Depth Images: A Review. *IEEE Sensors Journal*, **14**(6): 1731–1740.

Mei, K., Xu, L., Li, B., Lin, B., y Wang, F. (2015). A real-time hand detection system based on multi-feature. *Neurocomputing*, **158**(C): 184–193.

Mitra, S., Member, S., y Acharya, T. (2007). Gesture Recognition : A Survey. *IEEE Transactions on Systems, Man, and Cybernetics, Part C: Applications and Reviews*, **37**(3): 311–324.

Murthy, G. R. S. y Jadon, R. S. (2009). A Review of Vision Based Hand Gestures Recognition. *International Journal. Of Information Technology and Knowledge*, **2**(2): 405–410.

Nayakwadi, V. (2014). Natural Hand Gestures Recognition System for Intelligent HCI : A Survey. *International Journal of Computer Applications Technology and Research*, **3**(1): 10–19.

Ong, K. C., Teh, H. C., y Tan, T. S. (1998). Resolving occlusion in image sequence made easy. *The Visual Computer*, **14**(4): 153–165.

Otsu, N. (1979). A Threshold Selection Method from Gray-Level Histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, **9**(1): 62–66.

Premaratne, P. (2013). *Human Computer Interaction Using Hand Gestures*. Springer. p. 182.

Rautaray, S. S. y Agrawal, A. (2015). Vision based hand gesture recognition for human computer interaction: a survey. *Artificial Intelligence Review*, **43**(1): 1–54.

Sgouropoulos, K., Stergiopoulou, E., y Papamarkos, N. (2014). A Dynamic Gesture and Posture Recognition System. *Journal of Intelligent & Robotic Systems*, **76**(2): 283–296.

Shin, S. (2013). *Emgu CV Essentials*. Packt Publishing. p. 118.

Silva, C. y Santos-Victor, J. (2001). Motion from occlusions. *Robotics and Autonomous Systems*, **35**(3-4): 153–162.

Smith, S. W. (1999). *Digital signal processing*. California Technical Publishing, segunda edición. p. 688.

Sultana, a. y Rajapuspha, T. (2012). Vision Based Gesture Recognition for Alphabetical Hand Gestures Using the SVM Classifier. *Ijcset.Com*, **3**(7): 218–223.

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. (2013). A Survey on Human Motion Analysis. En: M. Grzegorzek, C. Theobalt, R. Koch, y A. Kolb (eds.), *Time-of-Flight and Depth Imaging. Sensors, Algorithms, and Applications*, Vol. 8200. Springer Berlin Heidelberg, 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.

Apéndice A. Algoritmo 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 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 $\beta_t = \frac{\epsilon_T}{1-\epsilon_t}$, el valor de $e_i = 0$ si x_i es clasificado correctamente de otra forma $e_i = 1$.

8: **fin para**

9: El clasificador final o clasificador fuerte es:

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

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

Apéndice B. Algoritmo Adaboost 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 en 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

Apéndice C. Algoritmo que calcula el número de dedos.

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

Entrada: Los conjuntos $\mathcal{S}$, $\mathcal{D}$, el punto C_r y el valor L_r.
Salida: Número de dedos levantados, Nf.

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

Apéndice D. Algoritmo de reducción de falsos positivos en la localización de la mano.

Algoritmo 4 Unión de ROIs (regiones de interés) sobrepuestas.

Entrada: $RoiA$ arreglo de rectángulos que representan los ROIs encontrados por el clasificador. N número de elementos de $RoiA$.

Salida: $Rect$ ROI resultante.

1: $área = 0$.
2: $t = 0.6$.
3: $Rect = RoiA[1]$
4: **para** $i = 2$ hasta N **hacer**
5: Calcular el $área$ sobrepuesta de los rectángulos $Rect$ y $RoiA[i]$
6: **si** $área \geq t$ **entonces**
7: $Rect$ es unión de los rectángulos.
8: **fin si**
9: **fin para**