Universidad de Los Andes

Tesis de Maestría

# Reconocimiento del Lenguaje de Señas Manuales con el Kinect

*Autor:*
Gustavo Alejandro Realpe Fresneda

*Supervisor:*
Ph.D Fernando Lozano

*Tesis presentada en cumplimiento de los requisitos para el grado de Tesis de Maestría*

*en el*

Centro de Microelectrónica (Machine Learning)
Departamento de Ingeniería Eléctrica y Electrónica

Julio 2013

*"El habla es suficiente para el lenguaje, pero no es necesaria..."*

William Stokoe

UNIVERSIDAD DE LOS ANDES

# *Resumen*

Departamento de Ingeniería Eléctrica y Electrónica

Tesis de Maestría

**Reconocimiento del Lenguaje de Señas Manuales con el Kinect**

por Gustavo Alejandro Realpe Fresneda

En este documento se presenta un análisis de dos características fundamentales del lenguaje de señas, el análisis de gestos estáticos y con movimiento, con un énfasis especial en el reconocimiento de la forma de la mano, para esto se utiliza la información del esqueleto que brinda el Kinect, realizando el procesamiento de las imágenes de color y profundidad. El reconocimiento de lenguaje de señas desarrollado en este trabajo reconoce el abecedario de sordos Colombiano, que se compone de letras con y sin movimiento, y por último el reconocimiento de palabras ejecutadas con una sola mano. La técnica utilizada para la clasificación de las letras sin movimiento es un algoritmo de aprendizaje supervisado (SVM), y para la clasificación de letras o palabras con movimiento se utiliza Modelos Ocultos de Markov (HMM).

En el reconocimiento de letras estáticas se logra una precisión del 94.7 %, en las letras con movimiento un 93.8 %, y en las palabras un 99.1 % de aciertos.

# *Agradecimientos*

Quiero agradecer a todas las personas que hicieron posible que este proyecto resultara exitoso, iniciando con mi hermana Paola y mi prima Kathe, ya que su colaboración y aportes fueron muy importantes para el desarrollo de esta tesis, también agradecer a las personas que colaboraron como voluntarios para la adquisición de cada una de las imágenes, así mi más sentidos agradecimientos a Omar, Gloria, James, Felipe, Esperanza, y de forma muy especial a mi padres por todo su apoyo durante todo este proceso.

A mi supervisor Fernando Lozano quiero agradecerle por su apoyo, comprensión, su guía y lo más importante por haber creído en mí, y que también se expanda mi gratitud a mis compañeros de Grupo de Investigación.

Y para terminar, gracias a todos mis compañeros de LSI por todo su apoyo.

# Índice general

**Abstract** II

**Agradecimientos** III

**Lista de Figuras** VII

**Lista de Tablas** VIII

**Abreviaciones** IX

**1. Introducción** 1
- 1.1. Introducción . . . 1
- 1.2. Propuesta de Tesis . . . 2
  - 1.2.1. Organización del Documento . . . 4

**2. Revisión de Literatura** 5
- 2.1. Lenguaje de Señas . . . 5
  - 2.1.1. Historia . . . 5
- 2.2. Historia en el reconocimiento . . . 7
  - 2.2.1. Reconocimiento de gestos basado en guantes . . . 8
  - 2.2.2. Reconocimiento de gestos basado en visión 2D y 3D . . . 9
    - 2.2.2.1. Reconocimiento de gestos con Kinect . . . 11
- 2.3. Kinect . . . 13
  - 2.3.1. Características . . . 14
  - 2.3.2. SDK para Windows . . . 15
- 2.4. Descriptores de la forma de la mano . . . 15
  - 2.4.1. Descriptores de Fourier . . . 16
  - 2.4.2. Descriptores de Fourier Invariantes 2 . . . 17
  - 2.4.3. Ellipse Fourier . . . 17
  - 2.4.4. Momentos Invariantes . . . 20
  - 2.4.5. Análisis de Componentes Principales . . . 21
- 2.5. Hidden Markovs Models . . . 22
- 2.6. Espacio de color Yuv . . . 23

**3. Reconocimiento Letras Estáticas** 24
- 3.1. Puntos del Esqueleto . . . 24
- 3.2. Modelo de Color elíptico UV . . . 25
- 3.3. Detección y seguimiento de la mano . . . 25

3.3.1. Modelo de color personalizado . . . 26
3.3.2. Contorno de la mano . . . 27
3.3.2.1. Preparación del contorno . . . 28
3.3.3. Información de la mano de Profundidad . . . 29
3.4. Aplicación de los descriptores la mano . . . 30
3.4.1. Descriptores de Fourier . . . 30
3.4.2. Descriptores de Fourier Invariantes 2 . . . 31
3.4.3. Descriptores Elípticos de Fourier . . . 31
3.4.4. Función de distancia y tangente . . . 32
3.4.5. Momentos Invariantes . . . 33
3.4.6. Análisis de Componentes Principales . . . 33
3.5. Experimentos, entrenamiento y resultados . . . 34
3.5.1. Pre-procesamiento de datos . . . 34
3.5.2. Adquisición de datos . . . 35
3.5.3. Clasificación SVM . . . 36
3.5.4. Resultados . . . 37
3.5.5. Conclusiones . . . 39

**4. Reconocimiento de Letras con Movimiento** **40**
4.1. Detección y seguimiento de la mano . . . 40
4.2. Descriptores . . . 40
4.2.1. Forma de la mano . . . 41
4.2.2. Dirección de Movimiento . . . 42
4.2.3. Descriptores de Dirección . . . 43
4.3. Experimentos, entrenamiento y resultados . . . 44
4.3.1. pre-procesamiento de datos . . . 45
4.3.2. Adquisición de datos . . . 45
4.3.3. Clasificación con HMM . . . 46
4.3.4. Resultados . . . 46
4.3.5. Conclusiones . . . 47

**5. Reconocimiento de Palabras** **48**
5.1. Detección y seguimiento de la mano . . . 48
5.2. Descriptores . . . 48
5.2.1. Forma de la mano . . . 49
5.2.2. Descriptores de movimiento y dirección . . . 49
5.2.3. Distancia Normalizada Cabeza-Mano . . . 50
5.2.4. Cosenos Directores Cabeza-Mano . . . 50
5.3. Experimentos, entrenamiento y resultados . . . 50
5.3.1. pre-procesamiento de datos . . . 51
5.3.2. Adquisición de datos . . . 51
5.3.3. HMM . . . 52
5.3.4. Resultados . . . 52
5.3.5. Conclusiones . . . 53

**6. Conclusiones y trabajo futuro** **54**

**Bibliografía** **56**

# Índice de figuras

2.1. Esqueleto Kinect . . . . . 15
2.2. Codificación de una cadena de Freeman. . . . . 18
2.3. Ejemplo contorno binario. . . . . 18
2.4. Topología HMM. . . . . 22

3.1. Puntos del esqueleto Kinect . . . . . 25
3.2. Modelo de color elíptico. . . . . 26
3.3. Modelo de Color personalizado . . . . . 27
3.4. Contorno Mano . . . . . 28
3.5. Contorno Mano Normalizado . . . . . 29
3.6. Información profundidad Mano . . . . . 30
3.7. Función de Distancia y la tangente . . . . . 33
3.8. Ejemplos de letra C. . . . . 36

# Índice de tablas

2.1. Fomatos de Color, Kinect. . . . . 14
2.2. Fomatos de 3D, Kinect. . . . . 14

3.1. Cantidad de muestras de Letras. . . . . 36
3.2. Resultado entrenamiento Letras Prueba 1. . . . . 37
3.3. Resultado entrenamiento Letras con DFI2_PCA_FDT(32). . . . . 37
3.4. Resultado entrenamiento Letras Prueba 2. . . . . 38
3.5. Resultado entrenamiento Letras con DFI_PCA_FDT(32) . . . . . 38
3.6. Resultado entrenamiento Letras Básicas. . . . . 38

4.1. Cantidad de muestras de letras con movimiento. . . . . 46
4.2. Resultado entrenamiento Letras con movimiento. . . . . 47

5.1. Cantidad de muestras de palabras. . . . . 52
5.2. Resultado entrenamiento Palabras con movimiento. . . . . 52
5.3. Resultado entrenamiento Palabras con movimiento. . . . . 53

# Abreviaciones

| | |
|---|---|
| **ASL** | **A**merican **S**ign **L**anguage |
| **FPS** | **F**rames **P**or **S**egundo |
| **HMM** | **H**idden **M**arkov **M**odel |
| **LS** | **L**enguaje de **S**eñas |
| **PCA** | **P**rincipal **C**omponent **A**nalysis |
| **RLS** | **R**econocimiento del **L**enguaje de **S**eñas |
| **SDK** | **S**oftware **D**evelopment **K**it |
| **SVM** | **S**upport **V**ector **M**achine |

# Capítulo 1

# Introducción

## 1.1. Introducción

El reconocimiento del lenguaje de señas es un área de investigación en plena evolución, si se compara con el reconocimiento de voz, se podría decir que este se encuentra en una etapa más avanzada que el de señas. Para intentar solucionar este problema se han propuesto muchas soluciones, iniciando con la utilización de guantes con sensores que pueden identificar con mucha precisión la forma de la mano, se continua con soluciones basadas en visión, que es una forma más natural de reconocimiento, aunque la tarea de reconocer la forma de la mano se vuelve mucho más complicada; en este tipo de reconocimiento, se han propuesto soluciones utilizando guantes de colores, análisis de imágenes en 2D y 3D y más recientemente la utilización del Kinect. Una de las grandes ventajas del reconocimiento por visión es que se pueden observar todas las característica de la persona que realiza la seña, esto es importante debido a que (cómo se verá más adelante) hay algunos gestos que además de la posición y forma de la mano, también es importante la posición del cuerpo, los gestos faciales, a estos se les llama gestos no manuales. Una de las tareas más complejas en el momento de reconocer una seña es que está no se genera de forma secuencial, como sucede en el lenguaje hablado en donde una letra o palabra va detrás de la otra, cuando se realiza un gesto hay que tener en cuenta la forma de la mano, la dirección y velocidad de movimiento, la posición del cuerpo, la forma de la cara y todo esto se hace al mismo tiempo, donde la variación en una de ellas puede cambiar el significado de lo que se quiere decir.

En este trabajo se aplica un modelo basado en visión, en donde se utiliza como medio de comunicación, entre el exterior y la máquina, un dispositivo desarrollado por Microsoft, el Kinect, el cual posee unas características que lo hace muy útil en el reconocimiento del lenguaje de señas. También se enfatiza y es uno de los objetivos, el reconocimiento de la forma de la mano cuando se está realizando un gesto o seña, pero no se tiene en cuanta gestos no manuales, solo importa la dirección y forma de la mano para caracterizar una seña. Para llevar a cabo el reconocimiento de la forma de la mano o un gesto se proponen una serie de descriptores simples, que se considera podrían caracterizarlos y después se pasa a hacer el análisis de estos.

Para el reconocimiento de la forma de la mano se utiliza un algoritmos de aprendizaje supervisado, llamado Máquinas de vectores de Soporte (o por sus siglas en ingles *SVM*) y para el reconocimiento de letras o palabras que incluyen movimiento se utiliza Redes Ocultas de Markov (o por sus siglas en ingles *HMM*).

Este trabajo se basa en el Lenguaje de Señas Colombiano, en una primera aproximación se hace el reconocimiento de 17 letras del abecedario Colombiano que no incluyen movimiento, después se continua con 4 letras (G,J,S,Z) del abecedario que incluyen movimiento y por último se hace el reconocimiento de 4 palabras que comparten la característica que solo incluyen una mano para su ejecución.

Todo el desarrollo se hizo sobre el lenguaje de programación C#, con ayuda de herramientas como EmguCV ([1]) que se utiliza para todo el procesamiento de imágenes y SVM, Accord .NET ([2]), que se utiliza para la implementación de HMM.

## 1.2. Propuesta de Tesis

Si hablamos de las señas, estos son gestos que existe desde siempre y que no es exclusivamente para las personas que tienen deficiencias auditivas, todos los seres humanos utilizamos la comunicación a través de las señas para afirmar (moviendo la cabeza hacia arriba y abajo), para negar (moviendo la cabeza hacia los lados), para expresar emociones y mucho más. Esto se ve reflejado en que si nos muestran un vídeo sin audio de personas comunicándose, somos capaces de saber si las personas están discutiendo, si están hablando de forma normal, si están alegres y demás; con esto, se quiere mostrar que las señas, para las personas sin ninguna discapacidad en el habla, son totalmente

naturales, normales y de uso diario. Es debido a esto, que en los últimos años se ha habido un gran interés y avance en el área de reconocimiento de gestos [3], incluyéndolo en vídeo juegos[4], simulaciones de realidad virtual, robótica, tele-medicina, aprendizaje a distancia, etc [5].

Pero el reconocimiento de gestos es diferente a el lenguaje de Señas, debido a que este último contiene una estructura ya definida, reglas, semántica, gramática, etc, y en algunos caros no es tan obvia como se tendería a pensar. Hay que tener claro que el lenguaje de señas para las personas con dificultades en el habla es su primera lengua, si ellos aprenden a escribir el español, para ellos ya es como una segunda lengua, así como lo sería para las personas hablantes el aprender este lenguaje. Con esto quiero enfatizar en la necesidad de implementar sistemas que permitan la interacción con ellos de forma natural, para que así se pueda lograr una inclusión en la sociedad y el acceso a todas las herramientas que, los que no tenemos con esa discapacidad, tenemos disponibles[6].

De forma muy general se puede decir que el lenguaje de señas de caracteriza por gestos manuales y gestos no manuales, en este trabajo se hace un enfoque solo en los gestos manuales, esto quiere decir que se ignora toda la información que pueda provenir de gestos faciales, de posición de la mano y de la velocidad de ejecución de una seña.

También se propone la utilización del kinect como dispositivo de interacción entre el usuario y el algoritmo de reconocimiento, se elige el Kinect debido a las características especiales que tiene (ver Sección 2.3) que se pueden resumir en bajo costo, información de color, información de profundidad, tracking del cuerpo de una persona, y más las herramientas adicionales ya desarrolladas sobre le kinect.

Los objetivos de esta tesis son,

- Proponer descriptores para reconocer la forma de la mano
- Clasificación y reconocimiento de la forma de la mano
- Reconocimiento de letras de la abecedario Colombiano sin movimiento
- Reconocimiento de letras de la abecedario Colombiano con movimiento
- Reconocimiento de palabras con movimiento

Todo el reconocimiento de las letras o las palabras se basa en los movimientos realizados con una sola mano, y específicamente para el reconocimiento de las palabras con movimiento es obligatorio el uso de un guante.

### 1.2.1. Organización del Documento

La tesis esta organizada y dividida en cinco capítulos, en el primer capítulo se encuentra la introducción y el propósito de esta; en el segundo capítulo se realizo una recopilación de investigaciones y desarrollos que se han realizado en los últimos años con relación al reconocimiento de señas; en el tercer capítulo se presentara la metodología que se utilizo para la identificación y reconocimiento de las letras estáticas del abecedario del lenguaje de señas Colombiano, en el cuarto capítulo se muestra el procedimiento que se siguió para el reconocimiento de las letras con movimiento del abecedario de señas Colombiano. En el quinto y último capítulo encontrara el desarrollo que se llevo acabo para lograr el reconocimiento de palabras del lenguaje de señas.

# Capítulo 2

# Revisión de Literatura

## 2.1. Lenguaje de Señas

### 2.1.1. Historia

La comunicación ha sido fundamental en la historia de la humanidad por ser indispensable para que exista vida social, para poder asociarse, integrarse, por esto desde el origen de los grupos sociales y en el desarrollo de la sociedad, surgieron por diferentes factores y hasta por casualidad se seleccionaron diferentes tipos de comunicación; algunas especies decidieron interpretar señales químicas, físicas y el adoptado por los seres humanos, el lenguaje [7].

El grupo social de sordos fue producto de marginamiento, pero con la evolución de la sociedad y el perfeccionamiento de los mecanismos para comunicarse , el de las señas se consideró un lenguaje cuando en 1960, él lingüista estadounidense William Stokoe que lo planteo (propuso) en su primera obra: “Gramática de un lenguaje por señas”. La publicación la realizó basándose en que los sordos estaban socializados y requerían comunicarse, y no aprendieron esto solamente por la capacidad de leer los labios o de hablar por gestos guturales, lo acogieron porque así como ellos, sus semejantes también utilizan señas, pues utilizan un canal diferente del lenguaje. Es decir, para que haya interacción entre los miembros de una comunidad, utilizan el “Sistema interactivo central” de su cultura, su lenguaje propio. Diez años después de esta publicación, ya fue un hecho, las señas de los sordos son un lenguaje que necesita un estudio y desarrollo propio. Pues

como dice Stokoe en su libro "El lenguaje de las manos", "El habla es suficiente para el lenguaje, pero no es necesaria" [7].

Al igual que en el hablado, el lenguaje de las señas que cuenta con su propia gramática y por ende vocabulario, tiene muchos y diferentes contrastes, que se consideran importantes y variantes según el dialecto . Un ejemplo de estos contrastes en el habla es la diferencia de la s y la z, pues dependiendo de estas cambian el significado de las palabras. Esto mismo ocurre con las señas pues la parte superior del cuerpo puede hacer tantos contrastes como el tracto vocal humano, y la visión del hombre puede detectar una gran cantidad de expresiones en los movimientos que se hacen, pues la velocidad con la que se realiza el movimiento, donde una cadencia lenta de los brazos significa algo que sigue sucediendo, y uno más rápido significa algo que ya concluyó [7].

Los rasgos gramaticales principales del lenguaje de señas, los artículos no se utilizan pues no son necesarios para preceder los nombres, los nombres son siempre concretos por ejemplo el gesto que implica comer sirve para signar comida o comedor, el contexto es el encargado de dar la semántica, los adjetivos siempre se sitúan después del nombre sin nada que los preceda, se poseen ciertos gestos para designar los pronombres, los verbos se expresan en infinitivo siempre, y en el lenguaje se utilizan muchos los adverbios, estos no se dan al agregar signos para modificar el adjetivo, se dan por la manera en que se realiza el movimiento, con diferente velocidad o fuerza en el movimiento un ejemplo es comer rápido pues hace el mismo signo de comer pero agilizándolo [8].

En el lenguaje por señas, existen dos tipos de signos, los manuales(MS), los cuales son realizados por los movimientos de las manos y los brazos, y los no manuales (NMS), los que llevan expresiones faciales, movimientos de cabeza, tronco, boca, ojos y posturas del cuerpo [9].

El movimiento de las manos es importante para reconocer un signo, al igual que la rapidez, la intensidad con la cual se elaboran los gestos que lo acompañan y la expresividad del rostro. Por lo mismo para entender el lenguaje, no es suficiente conocer el léxico de los signos y gestos, se necesita mirar la sincronización de estos juntos al hacer el movimiento con una comprensión clara de estos [9] [8].

Como casi siempre los sordos están rodeados de personas que oyen/hablan, han debido adaptarse a la cultura de la mayoría, en consecuencia, su idioma está influido por el medio

predominante, llevándolos a agregar nuevas expresiones o tomar términos del lenguaje hablado de los que los circundan. Esto obviamente demuestra que los lenguajes de señas difieren debido a los cambios culturales. La lengua por señas no es universal, existen tantas, dependiendo de los grupos o comunidades de personas sordas se conforman. Por lo tanto, varían en los diferentes países [7].

En nuestra nación, este lenguaje se conoce como el lenguaje de señas colombiana, y fue reconocido oficialmente en 1996 como la lengua utilizada por la comunidad sorda de nuestro país [10].

## 2.2. Historia en el reconocimiento

Aunque en los últimos años se han logrado grandes avances en el reconocimiento del lenguaje, aún hace falta mucho camino para que un ser humano se comunique de forma natural con una máquina [6]. Cuando se habla de lenguaje siempre se intenta asociar con el lenguaje hablado o escrito, y aunque el lenguaje hablado y el lenguaje de señas tiene cosas en común, y algunas técnicas utilizadas para el reconocimiento del habla en cierto grado, son aplicables al de señas la estructura del lenguaje cambia[11], mientras que el lenguaje hablado los símbolos son secuenciales, palabra pronunciada le continua otra palabra, en el lenguaje de señas los símbolos se forman de forma paralela, debido a que una seña o palabra puede ser expresada por el movimiento de la(s) mano(s), la forma(s) de la(s) mano(s), gestos faciales, movimiento de la cabeza [12], y todos estos se ejecutan durante la realización de la seña y cada uno de ellos juegan un papel en cuanto al significado, por tanto para poder reconocer realmente una seña hay que tener todos estos canales en cuenta.

El reconocimiento del lenguaje de señas ha sido de mucho interés, hace ya varios años y actualmente hay muchos investigadores trabajando en los avances en está área. En las últimas dos décadas han habido muchos adelantos y propuestas de diferentes métodos para poder atacar este problema, en la literatura hay dos tipos de enfoques, uno es basado en la utilización de guantes con dispositivos electromecánicos para sensar los diferentes movimientos y formas de la mano, y el otro se basa en dispositivos de visión. El enfoque basado en guantes, tiene buenos resultados detectando la forma exacta de los dedos, pero tiene el inconveniente que toca utilizar un dispositivo adicional al cuerpo,

por lo cual, se vuelve incomodo para ciertas personas, se hará una revisión más a fondo de esta técnica en la Sección 2.2.1. En el enfoque de las técnicas por visión, ha tenido una gran aceptación, porque la persona no necesita ponerse ningún hardware adicional y han habido una gran cantidad de métodos propuestos, iniciando con la utilización de guantes de colores, análisis y procesamiento de imágenes en 2D, 3D, con el uso de varias cámaras y unas de las mas recientes es la incorporación del Kinect como dispositivo de desarrollo en esta área, se hará una revisión más a fondo de esta técnica en la Sección 2.2.2.

Las señas están compuestas, no solo por el movimiento de las manos y los brazos (gestos manuales), sino también incluyen gestos faciales, movimiento de la cabeza, posición del cuerpo (gestos no manuales). [12]

### 2.2.1. Reconocimiento de gestos basado en guantes

Este método de reconocimiento fue uno de los primeros que utilizaron, hoy en día hay algunas aplicaciones que aún lo utilizan pero ya en menos proporción, debido al inconveniente de necesitar que la persona utilice un hardware adicional en su cuerpo, lo que puede ser incomodo y más importante que como se dijo en la Sección 2.1.1, una seña esta compuesta, no solo los movimientos de la mano, sino también posiciones del cuerpo y gestos faciales, que con está técnica se pierden por completo, y solo es posible obtener información de la posición y forma exacta de la mano. A pesar de las desventajas presentadas, esta forma de reconocimiento logra muy buenos resultados en palabras donde la forma de la mano es muy significativa, un ejemplo de esto es el abecedario de los sordos, en el cual la forma de la mano es muy importante, y además hay algunas letras que tienen forma de la mano parecida; en estas características son muy fuertes y logran mejores resultados, más precisión y menos complejidad que los métodos basados en 2D-visión.

En el trabajo de Rebollar et al.[13] se presenta una solución a este problema, en la cual se utiliza un guante que ellos llaman Accele Glove, este se basa en cinco MEMS(por sus siglas en ingles de Sistema Micro-Electromecánico) los cuales transmiten sus mediciones a un micro controlador, que va conectado a un PC, que ejecuta un algoritmo de reconocimiento, donde se implementa un clasificador de un tipo de árbol de decisión, el cual es capaz de reconocer 28 patrones de mano, que son, las 26 letras del ASL y 2 simbolos más que representan un espacio y un cambio de línea, con este, se obtiene el resultado

de 100 % del reconocimiento con 21 de las 26 letras y en el peor caso obtuvieron un 78 % con la letra U. Liang y Ouhyoung[14], presentan un trabajo en donde reconocen 250 palabras del lenguaje de señas Tailandes en tiempo real y de modo continuo , a través de modelos ocultos de Markov (HMM) con unos resultados de 80.4 % de reconocimiento. En [15] Swee et al. presenta una solución al problema de reconocimiento del lenguaje de señas utilizado en Malasia a través de un guante inalámbrico y acelerómetro en la muñeca, codo y hombro, para el reconocimiento utilizan HMM y es capaz de identificar 25 palabras y traducirlas a voz.

### 2.2.2. Reconocimiento de gestos basado en visión 2D y 3D

En las últimas décadas, este enfoque ha sido de mucho estudio y ha llamado mucho la atención por que presenta una solución más natural para el hablante del lenguaje de señas ya que no tiene que llevar un hardware adicional. Las diferentes soluciones propuestas inician con el análisis de imágenes en 2D con guantes o marcadores de colores para identificar la manos, después se presentan trabajos para que los guantes no fueran necesarios, se han presentado soluciones con 1, 2 y hasta 3 cámaras, y uno de los desarrollos más recientes es la utilización de cámaras en 3D, que aparte de la información de color también da información de la distancia del objeto y la cámara. Este enfoque es el que más nos interesa, por que en él esta basado nuestro trabajo.

Kelly et al.[16] presentan un programa que permite a las personas aprender y corregir ciertos gestos manuales utilizados en el lenguaje de señas, para su desarrollo utilizan una webcam y guantes de colores diferentes para las dos manos, y se extraen las características básicas de la mano, que son, forma, posición y movimiento de la mano. Para obtener la forma de la mano utilizan funciones de tamaño, a estas las combinan con PCA para reducir la dimensionalidad y finalmente utilizan momentos invariantes, con esta propuesta alcanza una precisión del 93.8 %. Otro trabajo presentado por Kelly et al. [17] presentan una solución que involucra también guantes de colores para cada una de las manos, la diferencia del trabajo anterior es que ahora se hace un reconocimiento de palabras continuo en las que intervienen las dos manos, se detectan las manos, la cara y los ojos del que ejecuta la seña, se hace también, un análisis de cuales son los mejores descriptores del movimiento para las señas en estudio, y se aplica HMM para poder identificar cuando inicia y finaliza un gesto, y se usa otro HMM para reconocer el

gesto que se está realizando, con esta propuesta y utilizando 8 diferentes gestos se tiene una precisión del 95.6 %. Continuando con Kelly et al. [18] a finales de 2009 presentan un muy buen trabajo, en el que además de detectar el movimiento de las manos, el inicio y fin de un gesto, se acercan un poco a un reconocimiento real de una seña, adicionando la detección de gestos no manuales (movimiento de la cabeza) en secuencias continuas de gestos, para realizar esto, proponen entrenar de forma independiente HMM multi-dimensionales y HMM con umbral, obteniendo una precisión del 95 %. En un trabajo reciente Zhu and Wong[19] presentan una solución al problema del reconocimiento del abecedario ASL, en imágenes, sin incluir las letras con movimiento, utilizando un descriptor Kernel, el cual fusiona la información de las imágenes de color e información de profundidad, con esta aproximación y con datos de entrenamientos reducidos se tiene una precisión de 88.94 %. Tanibata et al. [20] proponen una solución al reconocimiento de palabras del lenguaje de señas japones, sin guantes, dando solución a las oclusiones que se presentan entre las manos y la cara, para el color de la piel, Tanibata et al. utilizan una plantilla inicial, de la cual obtienen el modelo de color de las manos y de la cara, en este trabajo no se detecta el inicio y el final de un gesto y las características que utilizan del movimiento son ángulo de la mano, ángulo de movimiento de la mano, distancia de la mano a la cabeza y distancia de la muñeca a los puntos de contorno de la mano, y adicionalmente se detecta la posición del codo, luego con esos descriptores, se entrenan 2 HMM, uno para cada mano, y un modelo por gesto, al final se combinan las dos probabilidades para elegir el gesto más probable. Cooper et al. [21] publican un trabajo muy interesante, en este trabajo se utilizan guantes de colores para cada una de las manos, se utilizan características de 2D y 3D; la propuesta que hacen Cooper et al. es que los descriptores de forma de la mano y de movimiento se discreticen, para lograr este objetivo, se inicia con los descriptores de la ubicación de la mano, para esto se necesita saber en donde se esta haciendo el movimiento con respecto a la persona, en una primera aproximación, se crea una grilla basada en el ancho de la cabeza de la persona, que se centra en la cabeza de acuerdo a donde se detecte la mano, codificando la posición, y en una segunda propuesta, se utiliza la información de 3D, y puntos específicos del cuerpo para describir la ubicación, en cuanto al movimiento, se obtiene por frame que movimiento esta haciendo la mano, por ejemplo, abajo,arriba, derecha e izquierda, y adicionalmente se codifica la información de la relación entre las dos manos, si se aleja, si se mueven juntas o se alejan, y en cuanto a la forma de la mano, se dice que

hay 12 formas básicas de la mano, que todos los gestos incluyen y entrenan un clasificador Random Forest, para que realice esta tarea, con estos descriptores se prueba con un HMM y con un Sequential Pattern Boosting, obteniendo los resultados de precisión de 54 % y 76 % respectivamente. Isaacs J. y Foo S. [22] presentan una solución con imágenes 2D a la que les aplican filtros wavelet para generar los descriptores de la forma de la mano, después, los descriptores obtenidos son reconocidos a través de una red neuronal que es capaz de reconocer 24 diferentes letras del ASL, con una precisión del 99.9 %.

#### 2.2.2.1. Reconocimiento de gestos con Kinect

El kinect es un dispositivo que se creó con una idea muy clara y enfocada hacia los vídeo juegos, pero durante los últimos años se ha demostrado que reúne unas características muy interesantes para diversas aplicaciones como, interfaz para controlar diferentes programas a distancia, reconstrucción de imágenes en 3D, reconocimiento de objetos, aplicaciones de mercadeo y mucho más. Y para el caso específico del RLS los hace muy interesantes por la información que proporciona y los desarrollos que se han realizado, que se muestran a continuación. Información adicional sobre el Kinect se amplia en la Sección 2.3.

El proyecto CopyCat esta enfocado a desarrollar juegos que le permitan a los niños sordos mejorar sus habilidades del lenguaje, y en uno de esos intentos Zafrulla et al.[23] presentan una aproximación realizada con el Kinect, el objetivo del juego es que los niños puedan formar un número de oraciones ya definidas, con ciertas palabras, la estructura de la oración debe ser sujeto-objeto-preposición-adjetivo. Zafrulla et al. utilizan la información del esqueleto, para generar los vectores de características y utiliza HMM para poder reconocer las diferentes palabras propuestas por diferentes personas.Otro buen trabajo presentado por Pugeault et al.[24] en el 2011, en que desarrolla una aplicación que permite a través del abecedario del ASL que se escriban palabras y oraciones completas. Pugeault et al. lo primero que hacen es extraer la región en donde se encuentra la mano, después a esta imágenes se le aplican unos tipos de los filtro de gabor para obtener el vector de características y a través de un clasificador multi-clase random forest clasifican las palabras que han sido aprendido con anterioridad. En la aplicación propuesta aparecen una serie de letras que son las más probables según el gesto de la persona, y ella, puede elegir entre esas posibilidades. Recientemente Lang et al.[25] presentan un trabajo

en donde desarrollan un marco de referencia en el reconocimiento de Señas utilizando HMM, el cual llaman DragonFly (por las sigla en ingles **Dra**w **G**estures **on** the **Fly**). A través de este marco de referencia se pueden reconocer gestos del LS sin tener en cuenta la forma de la mano, ni los gestos faciales, y también tiene la posibilidad de enseñarles unos nuevos, para esto utilizan la información de profundidad y del esqueleto que provee el Kinect; probando el sistema con 25 señas del Lenguaje de señas Alemán alcanzan una precisión del 97 %. Lang S. en su tesis de maestría ([26],[25]) desarrolla un marco de referencia para el reconocimiento de gestos, sin incluir la forma de la mano, ni los gestos faciales, utilizando la información de profundidad; el método de clasificación utilizado es HMM, para este también se genera un algoritmo para el entrenamiento y la selección de cantidad de estados por seña, alcanzando una precisión del 97 %. En su tesis de maestría Li Y. [27] se enfoca en el problema de reconocimiento de gestos con la mano, para darle solución, Li Y. utiliza solo la información de profundidad, detectando el contorno de la mano y a través de defectos convexos en el contorno detecta los dedos, alcanzando una precisión del 84 % con una mano. Los SDK disponibles para el Kinect, no presentan la funcionalidad de poder detectar los dedos de las mano, debido a la baja resolución de las imágenes que provee, pero Ryan D. J.[28] presenta en su tesis de maestría una solución a este problema, utilizando la información de profundidad para obtener el contorno de la mano, utilizando DTW (Dynamic Time Warping) para hacer el reconocimiento de los gestos de la mano, Ryan también presenta en su trabajo un API implementado en C#. Siguiendo con el reconocimiento de la mano con el Kinect Raheja et al.[29] en el 2011 presenta un trabajo en donde son capaces de detectar, aproximadamente, la punta de los dedos y el centro de la mano, para esto utilizan un detector de mano que provee NITE para el seguimiento de las manos, después de detectar la mano se hace una umbralización sobre la información de profundidad, se aplica un filtro circular, para quitar la palma de la mano, que queden solo los dedos y se aplica una transformada de distancia para obtener el centro de la palma de la mano. Ren et al. [30] también presenta el reconocimiento de formas de la mano con el Kinect, utilizando la información de profundidad y de color, y para clasificar las diferentes formas de la mano utiliza FEMD (Finger-Earth Mover's Distance), probando el método propuesto en el juego de piedra, papel y tijera, y en una calculadora con operaciones básicas. En un trabajo presentado en el 2011 Oikonomidis et al. [31] hacen una muy buena aproximación al reconocimiento y seguimiento de la mano en 3D, generando un modelo virtual de la mano articulado con 27 grados de libertad; Oikonomidis et al. utilizan la información de color y profundidad que provee el

Kinect, se genera un modelo de color, y adicionalmente, según la observación genera una hipotesis del modelo 3D de la mano, el cual minimice la diferencia entre la observación y la hipótesis, este problema de optimización es solucionado a través de PSO (Particle Swarm Optimization).

## 2.3. Kinect

Desarrollado por Microsoft y PrimeSense, su nombre inicial fue Proyecto Natal, nombre que se le dio a este dispositivo antes de su lanzamiento oficial en el 2010 cuando solo eran rumores, después de su lanzamiento fue presentado como Kinect. Este dispositivo fue desarrollado única y exclusivamente para la consola de vídeo juegos XBOX 360, y presentaba una propuesta en la que el usuario no necesitaba ningun control para interactuar con la consola de juegos. Y se cree que ni Microsoft se imaginaba el éxito que iba a tener este dispositivo, unos días después de haberse hecho el lanzamiento oficial una empresa llamada AdaFruit ofreció una gran cantidad de dinero a la persona que fuera capaz de "hackear" el Kinect, no paso mucho tiempo, en sí pasaron horas antes de que esa persona reclamara su premio. Después de esto Kinect se convirtió en una de las herramientas mas usadas para diferentes tipos de aplicaciones de científicos y aficionados. Y en el 2011 Microsoft publicó un SDK para Windows 7 y 8, y lanzó una versión de Kinect con características mejoradas para el PC y con esto se abrió más la puerta para el desarrollo de muchas aplicaciones.

Con esta herramienta se han creado aplicaciones para el mercadeo, como la creada por Ar Door llamada AR Kinect Fitting Room[32], que permite a los usuarios de una tienda de ropa ver como le queda la prenda que quiere comprar sin necesidad de ponérsela físicamente. En medicina, un profesor de la universidad de Munich muestra una aplicación que llama "espejo mágico", el cual crea la ilusión de que una persona pueda ver el interior de su cuerpo, esta aplicación es utilizada para enseñar anatomía[33], y muchas más aplicaciones en muchas diferentes áreas, incluyendo las aplicaciones nombradas en la Sección 2.2.2.1([34]).

Una pregunta que se puede realizar es, que lo hace tan atractivo al Kinect para los investigadores y desarrolladores, en primer lugar se puede pensar en su precio, debido a

que es un dispositivo económico comparado con las aplicaciones potenciales que se pueden generar a partir de él y en segundo lugar por que tiene la capacidad de identificar simultáneamente hasta 6 personas, esta disponible la información de color, de profundidad y la información del puntos claves del esqueleto de la persona, esto se describe con más detalle en la Sección 2.3.1.

### 2.3.1. Características

El Kinect está conformado por una cámara de color, un par de sensores infrarrojos que son los encargados de dar la información de profundidad o 3D, un arreglo de micrófonos que son capaces de identificar de donde viene el sonido por triangulación. Y a través de un algoritmo interno, puede detectar hasta seis personas simultáneamente que se encuentren dentro del rango de visión, pero solo permite dos usuarios activos, y adicionalmente, provee la información del esqueleto de una persona en dos tipos, el primero son 20 puntos claves del cuerpo humano, como se muestra en la Figura 2.1, que incluye 3 puntos en las 4 extremidades, 4 puntos en la conexión con las extremidades, el torso, el cuello, la cabeza y la cintura, y el segundo, es el modo *"seated"* que provee solo los 3 puntos de los brazos, los hombros, el cuello y la cabeza, esta es una versión de puntos, para las aplicaciones que no necesitan toda la información del cuerpo. En cada uno de estos puntos se obtiene una coordenada $(x, y, z)$ y adicionalmente por cada punto clave también se tiene la información de la rotación de ese punto, con respecto a la cámara, o con respecto a un punto de referencia del esqueleto.

Tiene un rango de que va de 40 cm hasta 3 metros, se pueden obtener las imágenes a color en tres formatos, RGB, Yuv y Raw, pero dependiendo del formato que se elija se tienen diferentes frames por segundo que se pueden ver en la Tabla 2.1, en cuanto a la información de profundidad, se puede configurar de diferentes resoluciones, como se muestra en la Tabla 2.2, y se obtiene la distancia aproximada en milímetros desde la cámara a cada uno de los puntos de la imagen.

| Formato Color | Resolución |
|---|---|
| RGB o RawBayer | 640x480 @30fps<br>1280x960 @12fps |
| Yuv o RawYuv | 640x480 @15fps |

Tabla 2.1: Fomatos de Color, Kinect.

| Resolución |
|---|
| 80x60 @30fps |
| 320x240 @30fps |
| 640x480 @30fps |

Tabla 2.2: Fomatos de 3D, Kinect.

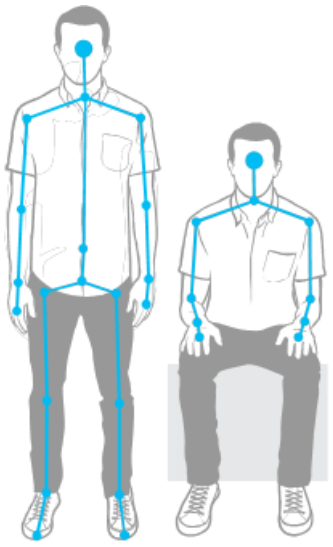


FIGURA 2.1: Esqueleto Kinect. [1]

### 2.3.2. SDK para Windows

Antes que Windows lanzara su propio SDK, había un SDK que permitía los desarrollos sobre el Kinect, este es OpenNI y el middleware NITE, que dejaban acceder a todas las características ya mensionadas del Kinect, adicionandoles nuevas funcionalidades, y con un gran atractivo para los desarrolladores debido a que permitía desarrollos en múltiples plataformas. En el 2011 Microsoft lanzó su propio SDK, el cual permite a los desarrolladores trabajar con el Kinect, sobre la plataforma Windows 7 y Windows 8, a través de los lenguajes de programación C#, visual basic, C++; y en su versión más reciente (1.7), permite la integración con Matlab y con Open CV.

Para el desarrollo de esta tesis se utilizó el SDK para windows de Microsoft versión 1.7.

## 2.4. Descriptores de la forma de la mano

Para el desarrollo de esta tesis es muy importante y uno de sus objetivos es, poder detectar y reconocer la forma de la mano, para lograr este objetivo, se utiliza la información de color y la información de 3D o profundidad, después de obtener la región de la mano a través de un modelo de color presentado en las siguientes secciones, nace la necesidad y la pregunta, ¿cómo se puede caracterizar una forma de la mano?, para responder esta pregunta se hace un estudio de los métodos más utilizados para reconocer la forma de la

[1] Esta imagen fue tomada de http://msdn.microsoft.com/en-us/library/hh973077.aspx

mano a partir del contorno de la imagen de color, y también los métodos más utilizados para reconocer la forma de la mano a partir de la imagen de profundidad. Y estos son los métodos que se van a presentar a continuación. La información de esta sección se basan en los libros de Gonzalez C. R. y Woods E. R *Digital Image Processing*[35] y de Sonka M. et al. *Image Processing, Analysis, and Machina Vision*[36].

### 2.4.1. Descriptores de Fourier

Teniendo un contorno cerrado de $K$ puntos en un plano $x$-$y$. Iniciando en un punto cualquiera se obtiene un conjunto de puntos $C = (x_0, y_0), ..., (x_{k-1}, y_{k-1})$, y cada uno de estos puede ser expresado como un número complejo de la forma,

$$s(k) = x_k + jy_k \quad k \in [0, K]. \tag{2.1}$$

Una de las ventajas de expresarlo de esta forma es que se pasa de un problema de 2 dimensiones a un problema unidimensional. Según la expresión de una transformada discreta de fourier (DFT) que se muestra en la Ecuación 2.2, se puede transformar $s(d)$ en el dominio de la frecuencia a través de la siguiente ecuación,

$$a(u) = \sum_{k=0}^{K-1} s(k) e^{-j2\pi uk/K} \quad u \in [0, K], \tag{2.2}$$

donde los coeficientes $a(u)$ son llamados descriptores de fourier. Se ha demostrado que para tener una buena aproximación aplicando la transformada inversa, no es necesario utilizar todos los K descriptores de fourier, sino que se puede tomar un conjunto de descriptores, con los cuales se pueda llegar a una buena aproximación del contorno inicial, con esta última característica obtenemos una reducción de dimensionalidad de los datos a procesar.

Hasta este momento los descriptores hallados son dependientes del punto donde inicia el contorno, del tamaño del contorno y a la rotación del contorno, y esto no es lo que se espera, por tanto, se hacen unas modificaciones a los descriptores obtenidos, para que puedan ser independientes a estas características[37].

Como se ve en la Ecuación 2.2, cuando $u = 0$ se tiene el componente de frecuencia real, que es el responsable del desplazamiento del contorno, haciendo el primer descriptor a

cero, obtenemos **Invarianza en traslación**,

$$a(0) = 0 \tag{2.3}$$

Para conseguir la **Invarianza en escala** lo que se propone es dividir cada uno de los descriptores por la magnitud de $a(1)$,

$$a(u) = \frac{a(u)}{|a(1)|} \quad u \in [0, K] \tag{2.4}$$

y por último para la **Invarianza en rotación y punto de inicio del contorno**, lo que se hace es tomar la magnitud del descriptor complejo $a(u)$, debido a que la parte imaginaria es la que lleva la mayor información de fase, aunque haciendo esta variación, hay perdida de información en el contorno,

$$a(u) = |a(u)| \quad u \in [0, K] \tag{2.5}$$

### 2.4.2. Descriptores de Fourier Invariantes 2

En el trabajo de Bourennane S. y C. Fossati [38], se hace referencia a unos descriptores de fourier modificados que también los hacen invariantes, y comentan que los descriptores planteados en la Sección 2.4.1 son invarientes más no estables, y definen un nuevo conjunto de descriptores que se definen a continuación, tomando a $k_0 = 0$ y $k_1 = 1$ y partiendo de los descriptores encontrados en la Ecuación 2.2,

$$\begin{aligned} I(0) &= |a_0| \\ I(1) &= |a_1| \\ I(k) &= \frac{a(k)a(0)^{1-k}a(1)^k}{I(0)^{0,5-k}I(1)^{k-0,5}} \end{aligned} \tag{2.6}$$

### 2.4.3. Ellipse Fourier

Los descriptores elípticos de fourier fueron por Kuhl F. y Giardina C. en 1982 [39], presentan una forma de utilizar los coeficientes de fourier en contornos cerrados descritos por cadenas de Freeman y los coeficientes resultantes son invariantes a la rotación, escala y traslación.

Una cadena de Freeman de tamaño $k$ esta definida por una cantidad de números concatenados como se ve en la Ecuación 2.7 donde un $c_i$ corresponde a un número definido entre 0 y 7 orientado en la dirección $(\pi/4) * c_i$ y tiene una longitud de 1 o $\sqrt{2}$ dependiendo si $c_i$ es par o impar, esta codificación de una cadena de Freeman se puede ver gráficamente en la Figura 2.2.

$$C = c_0 c_1 c_2 \dots c_k \tag{2.7}$$

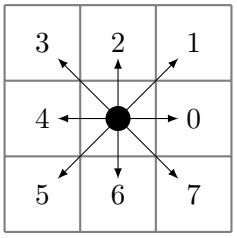



FIGURA 2.2: Codificación de una cadena de Freeman.

Tomando el contorno de una imagen binaria mostrado en la Figura 2.3, y sabiendo que el punto de inicio del contorno está representado por el recuadro rojo y se recorre en sentido de las manecillas del reloj, se puede definir una cadena de freeman de tamaño 13 ($k = 13$) como,

$$C = 0077636334421$$

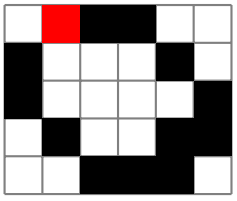


FIGURA 2.3: Ejemplo contorno binario.

Siguiendo la descripción en [40], se define la longitud de un elemento en la cadena de Freeman como,

$$|c_i| = dt_i = \begin{cases} \sqrt{2} & si \quad c_i \text{ es impar} \\ 1 & si \quad c_i \text{ es par} \end{cases} \tag{2.8}$$

Por tanto, la longitud de una cadena en en punto $n$ siendo $n < K$ esta dada por,

$$t_n = \sum_{i=1}^{n} dt_i \tag{2.9}$$

ahora se define $dx_i$ y $dy_i$ como las proyecciones de $c_i$ sobre los ejes $x$ y $y$ respectivamente, y se define como,

$$dx_i = signo(6 - c_i) * signo(2 - c_i) \tag{2.10}$$

$$dy_i = signo(4 - c_i) * signo(c_i) \tag{2.11}$$

donde,

$$signo(\Omega) = \begin{cases} 1 & si \quad \Omega > 0 \\ 0 & si \quad \Omega = 0 \\ -1 & si \quad \Omega < 0 \end{cases}$$

El principio fundamental de los descriptores elípticos de fourier es representar los ángulos de un contorno como una suma de armónicos elípticos([40]), por tanto se definen los coeficientes de fourier como,

$$\begin{aligned} a_n &= \frac{T}{2n^2\pi^2} \sum_{i=1}^{k} \frac{dx_i}{dti} \left[ cos\frac{2n\pi t_i}{T} - cos\frac{2n\pi t_{i-1}}{T} \right] \\ b_n &= \frac{T}{2n^2\pi^2} \sum_{i=1}^{k} \frac{dx_i}{dti} \left[ sin\frac{2n\pi t_i}{T} - sin\frac{2n\pi t_{i-1}}{T} \right] \\ c_n &= \frac{T}{2n^2\pi^2} \sum_{i=1}^{k} \frac{dy_i}{dti} \left[ cos\frac{2n\pi t_i}{T} - cos\frac{2n\pi t_{i-1}}{T} \right] \\ d_n &= \frac{T}{2n^2\pi^2} \sum_{i=1}^{k} \frac{dy_i}{dti} \left[ sin\frac{2n\pi t_i}{T} - sin\frac{2n\pi t_{i-1}}{T} \right] \end{aligned} \tag{2.12}$$

Pero estos coeficientes no tienen propiedades invariantes, por tanto, se proponen unas transformaciones para que estos descriptores lo sean, lo primero es que se hace es que sea **Invariante al punto de inicio**, por tanto se calcula el cambio de fase,

$$\theta_1 = \frac{1}{2} arctg \left( \frac{2\left(a_1 b_1 + c_1 d_1\right)}{a_1^2 + c_1^2 - b_1^2 - d_1^2} \right) \tag{2.13}$$

y se rotan los coeficientes, para dejar la fase en 0,

$$\begin{bmatrix} a_n^{'} & b_n^{'} \\ c_n^{'} & d_n^{'} \end{bmatrix} = \begin{bmatrix} a_n & b_n \\ c_n & d_n \end{bmatrix} \begin{bmatrix} cos(n\theta_1) & -sin(n\theta_1) \\ sin(n\theta_1) & cos(n\theta_1) \end{bmatrix} \tag{2.14}$$

Después se define para que los coeficientes sean **Invariantes a la rotación**, para esto el eje mayor del primer armónico se alinea con el eje X, se calcula una matriz de rotación,

y se rotan los coeficientes que en este punto son invariantes a el punto de inicio,

$$\begin{bmatrix} cos\phi_1 & sin\phi_1 \\ -sin\phi_1 & cos\phi_1 \end{bmatrix} \quad \text{donde,} \quad \phi_1 = arctg\frac{c_1^{'}}{a_1^{'}} \tag{2.15}$$

$$\begin{bmatrix} a_n^{''} & b_n^{''} \\ c_n^{''} & d_n^{''} \end{bmatrix} = \begin{bmatrix} a_n^{'} & b_n^{'} \\ c_n^{'} & d_n^{'} \end{bmatrix} \begin{bmatrix} cos\phi_1 & sin\phi_1 \\ -sin\phi_1 & cos\phi_1 \end{bmatrix} \tag{2.16}$$

De última se define la **Invarianza en la escala** y esta se logra dividiendo por la longitud del eje semimayor del primer armónico definida como,

$$L = \sqrt{a_1^{'2} + c_1^{'2}} \tag{2.17}$$

y se dividen los coeficientes por este valor, y se obtienen los coeficientes que tienen las propiedades de invarianza de punto de inicio, de rotación y de escala,

$$\begin{bmatrix} a_n^{'''} & b_n^{'''} \\ c_n^{'''} & d_n^{'''} \end{bmatrix} = \begin{bmatrix} a_n^{''} & b_n^{''} \\ c_n^{''} & d_n^{''} \end{bmatrix} \frac{1}{L} \tag{2.18}$$

### 2.4.4. Momentos Invariantes

Se puede definir el momento de una imagen de 2D con tamano de $MxN$ con orden $(p+q)$ como,

$$m_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x^p y^q f\,(x,y) \tag{2.19}$$

donde $p, q \in \mathbb{Z}^+$. El *Momento central* se define como,

$$\mu_{pq} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (x-X)^p\,(y-Y)^q\,f\,(x,y) \tag{2.20}$$

donde,

$$\bar{x} = \frac{m_{10}}{m_{00}} \quad \text{y} \quad \bar{y} = \frac{m_{01}}{m_{00}} \tag{2.21}$$

y se definen los *Momentos centrales normalizados*

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\gamma}} \quad \text{donde,} \quad \gamma = \frac{p+q}{2} + 1 \tag{2.22}$$

Y de aquí se proponen 7 momentos invariantes que están definidos como,

$$\begin{aligned}
\phi_1 &= \eta_{20} + \eta_{02} \\
\phi_2 &= (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \\
\phi_3 &= (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2 \\
\phi_4 &= (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \\
\phi_5 &= (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2\right] + \\
&\quad (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{012}) - (\eta_{21} + \eta_{03})\right] \\
\phi_6 &= (\eta_{20} - \eta_{02})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2\right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} - \eta_{03}) \\
\phi_7 &= (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} - \eta_{03})^2\right] + \\
&\quad (3\eta_{12} - \eta_{30})(\eta_{21} - \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} - \eta_{03})^2\right]
\end{aligned} \tag{2.23}$$

### 2.4.5. Análisis de Componentes Principales

Normalmente se utiliza el análisis de componentes principales sobre diferentes muestras sobre varios experimentos o imágenes, pero lo que se quiere en este momento es caracterizar una sola imagen de 2D. Si se tiene una imagen NxM, cada característica es representada por cada fila de la imagen, por tanto, se tiene un conjunto de N vectores de entrada cada uno de tamaño M, se forma la siguiente matriz $\mathbf{x} = [\mathbf{x_1}, \mathbf{x_2}, \ldots, \mathbf{x_N}]^\mathbf{T}$, y se plantea una transformación donde $\mathbf{A}$ es la matriz de transformación que hace un mapeo de los $\mathbf{x}$s a los $\mathbf{y}$s,

$$\mathbf{y} = \mathbf{A}(\mathbf{x} - \mathbf{m_x})$$

donde $\mathbf{m_x}$ es un vector de valores medio de todas la entradas,

$$\mathbf{m_x} = \frac{1}{M}\sum_{k=1}^{M}\mathbf{x}_k$$

La matriz $\mathbf{A}$ se obtiene de $\mathbf{C_x}$ debido a que las files de $\mathbf{A}$ son los eigenvectores de $\mathbf{C_x}$ ordenado de forma descendente, y se puede definir a $\mathbf{C_x}$ como,

$$\mathbf{C_x} = E\{(\mathbf{x} - \mathbf{m_x})(\mathbf{x} - \mathbf{m_x})^\mathbf{T}\} = \frac{1}{M}\sum_{k=1}^{M}(\mathbf{x_k}\mathbf{x_k^T}) - (\mathbf{m_x}\mathbf{m_x^T})$$

$\mathbf{C_x}$ es la matriz de covarianza, y es de tamaño de NxN, los valores sobre la diagonal representan las varianzas de $\mathbf{x}$ y los otros valores de la matriz $\mathbf{C(i,j)_x}$ la covarianza entre

las variables de entrada $\mathbf{x_i}$ y $\mathbf{x_j}$.

## 2.5. Hidden Markovs Models

Los HMM son un modelo estadístico utilizado, capaz de representar información espacio-temporal de un forma muy natural, esté tipo de modelos han sido muy exitosos en el reconocimiento del habla, y también han habido varios estudios en la aplicación de estos modelos en el reconocimiento de gestos y en el reconocimiento del lenguaje de señas.

Los HMM tiene algoritmos muy eficientes para el aprendizaje y el reconocimiento como son el algoritmo de Baum-Welch y el algoritmo de Viterbi.

En los modelos ocultos de Markov, los estados no son observables, lo único que se conoce son las observaciones y a través de estas se intenta generar una función de salida para cada uno de los estados. La cantidad de estados normalmente se elige de manera heurística.

Un HMM es básicamente unos estados unidos por unas transiciones, un ejemplo se ve en la Figura 2.4, cada uno de estos estados tiene asociados unas probabilidades, la primera es la probabilidad de hacer una transición a algún otro estado de la red, inclusive y la otra es la de salida que representa la probabilidad de salida de cada símbolo según el estado.

Un HMM se representa como $\lambda = (A, B, \pi)$, donde $A$ representa la matriz de transición, esto quiere decir que el elemento $a_{ij}$ de la matriz $A$ representa la probabilidad que se haga una transición del estado $s_i$ al estado $s_j$, un ejemplo de la matriz de transición de el modelo de 2.4 se puede ver en 2.24. $B$ es una matriz que representa la probabilidad de cada uno de los símbolos, y por último $\pi$ es un vector que contiene las probabilidades iniciales de cada estado. Para esta tesis, la salida de los estados esta representada por una función Gaussiana(Ecuación 2.25) debido a que en las observaciones se manejan valores reales.

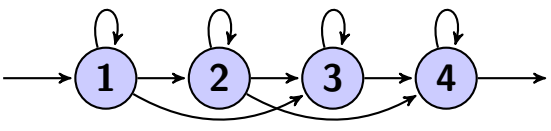



FIGURA 2.4: Topología HMM.

$$A = \begin{pmatrix} a_{11} & a_{12} & a_{13} & 0 \\ 0 & a_{22} & a_{23} & a_{24} \\ 0 & 0 & a_{33} & a_{34} \\ 0 & 0 & 0 & a_{44} \end{pmatrix} \tag{2.24}$$

$$b_j\left(O_t\right) = \frac{1}{\sqrt{(2\pi)^n \left|\Sigma_j\right|}} exp\left(\frac{1}{2}\left(O_t - \mu\right)' \Sigma_j^{-1}\left(O_t - \mu\right)\right) \tag{2.25}$$

## 2.6. Espacio de color Yuv

El modelo de Color yuv, esta compuesto por una componente de luminancia(y) y dos componentes de crominancia (uv), este modelo de color se asemeja un poco más que el RGB, hacia la forma como el ojo humano percibe los colores. Es muy utilizado para la transmisión y compresión de vídeo.

Los valores de Yuv se pueden obtener de los RGB con la siguiente transformación:

$$\begin{bmatrix} y \\ u \\ v \end{bmatrix} = \begin{bmatrix} 0{,}299 & 0{,}587 & 0{,}114 \\ -0{,}147 & 0{,}289 & 0{,}436 \\ 0{,}615 & -0{,}515 & 0{,}100 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{2.26}$$

Y la transformación inversa se puede lograr aplicando la siguiente transformación,

$$\begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1{,}139 \\ 1 & -0{,}394 & -0{,}5806 \\ 1 & 2{,}032 & 0 \end{bmatrix} \begin{bmatrix} y \\ u \\ v \end{bmatrix} \tag{2.27}$$

# Capítulo 3

# Reconocimiento Letras Estáticas

En este capítulo se desarrolla el trabajo realizado para la identificación y el reconocimiento de 17 de las Letras estáticas del abecedario de los sordos colombiano. Para esto se propone el modelo de color para obtener las características de la mano a partir del contorno, se utiliza el algoritmo de aprendizaje SVM para clasificar las diferentes palabras. La forma para hacer una palabra se puede encontrar en [41].

## 3.1. Puntos del Esqueleto

Según se vio en la Sección 2.3.1, hay dos formas en la que el Kinect devuelve la información del esqueleto, y la que se va a utilizar en esta tesis es el tipo *seated* de la cual obtenemos la información de posición y de rotación de cada uno de los 10 puntos (mano derecha/izquierda, muñeca derecha/izquierda, codo derecho/izquierdo, hombro derecho/izquierdo, cuello y cabeza) como se puede ver en la Figura 3.1. Se elige este modo debido a que cuando se realiza una seña los puntos del cuerpo que más se involucran en esta son los mencionados anteriormente, los demás puntos se podría decir que se mantienen estáticos.

Para el desarrollo de las demás secciones, los puntos que se tienen en cuenta con las manos, las muñecas, los hombros y la cabeza.

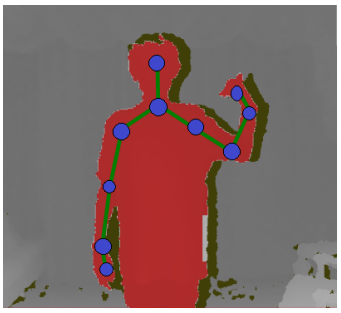


FIGURA 3.1: Puntos del esqueleto Kinect.

## 3.2. Modelo de Color elíptico UV

Se emplea el modelo de color utilizado por Soontranon et al. [42], en donde se parte del modelo de color Yuv descrito en la Sección 2.6, para robustez en la solución, se toma solo la información de crominancia y se ignora la información de luminancia para que el modelo no sea tan sensible a lo cambios de iluminación.

Según los estudios realizados, y las pruebas hechas, se puede ver que en este nuevo espacio 2D con ejes UV, el color de piel describe una forma de elipse como se puede ver en la Figura 3.2, por tanto, los valores de $U$ y $V$ que estén incluidos en la elipse, son los que se consideran como color piel. Para detectar el color piel en una imagen que está en formato Yuv, se recorre toda la imagen pixel por pixel y según el valor de uv, se etiqueta ese pixel como piel o no piel.

Este es el método utilizado para detectar el color de piel en las imágenes de color en esta tesis, es un método sencillo que da buenos resultados bajo las condiciones adecuadas. El detalle de como se aplica esta técnica se describe en la Sección 3.3.

## 3.3. Detección y seguimiento de la mano

Para esta tesis se configura el formato de la imagen a *RGB*, la resolución de *640x480* y la cantidad de tramas por segundo de *30*, de las posibles combinaciones mostradas en la Tabla 2.1, se elige esta configuración, por que es con el formato que se obtiene la mayor cantidad de frames por segundo y permite que el reconocimiento sea más continuo, aunque esta no sea la mejor resolución. El reconocimiento de la mano en imágenes 2D de baja resolución y aún en imágenes 3D es una tarea no trivial. Las tareas que se van

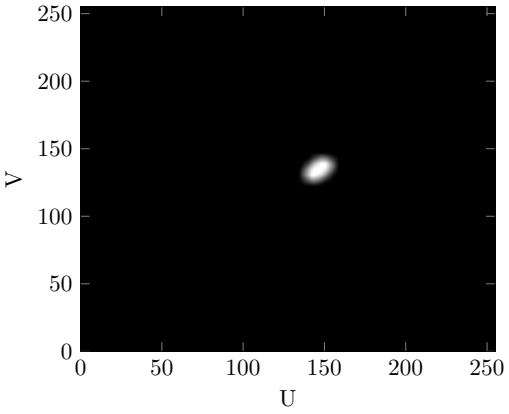



FIGURA 3.2: Ejemplo de modelo de color elíptico de piel. La parte negra es todo lo que se considera que no es piel, y lo que encierra la elipse de color blanco, se consideraría color piel.

a desarrollar en esta sección es la localización, seguimiento de la mano y la generación del contorno de la mano en las imágenes de color.

### 3.3.1. Modelo de color personalizado

Uno de los mayores inconvenientes en los modelos de color es que dependen en gran parte de la iluminación y de los diferentes colores de piel existentes, en esta tesis, se quiere proponer una adquisición del modelo del color cada vez que la persona vaya a realizar un gesto, con esto se crea un modelo de color de la persona que va a realizar el gesto, y adicionalmente se ajusta a las condiciones de iluminación y del ambiente, que de igual manera tienen que ser aceptables para que un buen modelo de color pueda ser obtenido.

Para iniciar el proceso, la aplicación verifica que la mano esté por encima de la posición de codo y al usuario se le pide que ubique la palma hacia al frente, con los dedos de la mano unidos como de muestra en la Figura 3.3(a), y la mano debe permanecer en esa posición y sin mayor movimiento por 5 segundos. La ubicación aproximada de la mano y del codo se obtiene de la información de punto de la mano que da el Kinect(Sección 3.1), con esta se calcula una región con dimensiones de 20 pixeles de alto y 15 pixeles de ancho, como se observa en la Figura 3.3(a) con un recuadro rojo, de la que se toma unas muestras del color (de 5 a 10 muestras) de la piel como se puede ver en la Figura 3.3(b), estas se convierte al formato de color Yuv, y se promedia sus valores para obtener un

modelo de color como se muestra en la Figura 3.3(c), este modelo se almacena en una matriz bidimensional de tamaño $256x256$.

(a) Cuadro Color (b) color Piel (c) Modelo Color

FIGURA 3.3: Modelo de Color personalizado. (a) muestra la posición que debe mantener la persona y el recuadro rojo representa la zona de la que se toma la muestra del color de la piel. (b) Representa una muestra de lo que se obtiene del recuadro rojo de (a). (c)Modelo de color obtenido a partir de las muestras de color de piel de una persona.

### 3.3.2. Contorno de la mano

Ya con el modelo de color de la persona, la siguiente tarea es obtener el recuadro de la mano, y la forma de la misma a través del contorno, para esto, se basa de nuevo en la posición de la mano que nos devuelve el Kinect y se toma un recuadro de la mano de un tamaño de 128 x 128 pixeles centrado en la posición de la mano, a través de varias pruebas se pudo deducir que este es un tamaño adecuado y que es suficiente para siempre albergar la mano a diferentes distancias.

Después de que se tiene el recuadro (Figura 3.4(a)), se convierte al formato Yuv y se le aplica el modelo de color hallado anteriormente, con esto se obtiene las regiones que coinciden con el color de piel descrito, en esta parte pueden aparecer regiones que coinciden con el modelo pero que no son piel como se ve en la Figura 3.4(b), para solucionar esto primero se aplica un filtro Mediana obteniendo el resultado que se ve en la Figura 3.4(c), adicionalmente a esto, a las regiones restantes se agrupan y se revisa el tamaño de cada una de ellas, y se deja solo la región más grande que se supone es la mano y por último se obtiene el contorno de la región hallada que corresponde a la forma de la mano, como se puede ver en verde en la Figura 3.4(d).

El contorno que se obtiene se conforma de un conjunto de tuplas de valores $(x, y)$ organizadas, que definen puntos claves y que uniéndolos describen visualmente la forma exterior de un objeto, en este caso la mano.

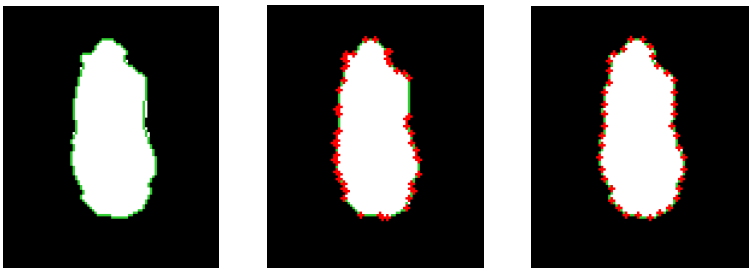
(a) (b) (c) (d)

FIGURA 3.4: Proceso para obtener contorno de la mano realizando la letra B. (a) Recuadro que se sabe contiene la mano. (b) Pixeles que probablemente son piel según el modelo de color. (c) Región de piel filtrada y suavizada. (d) Se extrae el contorno de la región hallada de la mano según el color de piel.

Para ayuda de la detección de la zona de la mano, las personas deben usar ropa de manga larga, y deben usar una banda negra que se pone alrededor de la muñeca, eso es para hacer más fácil la detección de la mano en sí.

#### 3.3.2.1. Preparación del contorno

Antes de aplicarle al contorno los descriptores, para hacerlo más inmune al ruido, y para dejar los puntos que siempre estén a una distancia igual entre ellos y así simular puntos periódicos, y de esta forma poderle aplicar los descriptores de Fourier con mejores resultados. Los puntos claves hallados por el algoritmo que calcula el contorno en la imagen binaria, se muestra en la Figura 3.5(b), y se puede ver que los puntos claves hallados no presentan una forma periódica o equidistantes, por tanto es necesario redistribuir los puntos, para esto se aplica un algoritmo (1) que lo que hace es, hacer una interpolación lineal entre los valores existentes del controno y la distancia deseada que se obtiene dividiendo el perímetro total del contorno entre el número de puntos deseados en el contorno, el resultado del algoritmo se puede ver en la Figura 3.5(c) con un total de 32 puntos deseados, el contorno se muestra en verde, y los puntos rojos representan los puntos normalizados del contorno.



(a) (b) (c)

FIGURA 3.5: Contorno de la mano normalizado, donde el verde representa el contorno de la mano, y los puntos rojos los puntos claves del contorno. (a) Contorno de la mano hallado. (b) Puntos de contorno hallados. (c) Puntos de contorno Normalizados.

**Algoritmo 1** Redistribución equidistante de los puntos de contorno

**Entrada:** $P$= Perímetro, $C$ =Puntos de contorno, $V$ =Cantidad de puntos deseados.

1: $d \leftarrow P/V$ {se calcula la distancia que debe haber entre cada uno de los puntos}
2: **para todo** todos los puntos del contorno C(i) **hacer**
3: $Di \leftarrow d * i$ {se calcula la distancia de donde se desea que esté el nuevo punto}
4: $(p0, p1) \leftarrow$ se obtienen los dos puntos de contorno original que contienen el nuevo punto a $Di$ distancia.
5: $C'(i) \leftarrow$ interpolación lineal entre $p0$ y $p1$ {las coordenadas del nuevo punto se obtiene.}
6: **fin para**

### 3.3.3. Información de la mano de Profundidad

En las secciones anteriores se analizo la forma de la mano partiendo de la información de color, en esta sección se ve a obtener la información de profundidad de la región de la mano.

Al igual como se hizo con la parte de color en la Sección 3.3.2, con la información de la posición de la mano se extrae de la imagen de profundidad de 640x480, un recuadro de 128x128 con centro en la posición de la mano como se muestra en la Figura 3.6(a), después de tener la región se le aplica una umbralización por profundidad. Para esto en el momento de la toma del modelo de color, se obtiene el largo total de la mano, y con este valor, del recuadro 3.6(a), se calcula el punto que está más cerca a la cámara y a este punto se le resta el largo total de la mano y esta es la región que se toma. Si se define $d$ como el valor del punto más cercano a la cámara más el tamaño de la mano, y si el recuadro de profundidad esta descrito por una función $D(x, y)$ que representa el valor en distancia en milímetros del objeto a la cámara en los puntos $x$ y $y$ dados, con valores posibles de $x = 1, 2, ..., 128$ y $y = 1, 2, ..., 128$, entonces la nueva imagen $D(x, y)$

está descrita por,

$$D'(x,y) = \begin{cases} D(x,y) & D(x,y) < d \\ 0 & en\ otro\ caso \end{cases}$$

Los valores de profundidad que se tiene al ser en milimetros, son valores grandes, por tanto, se le aplica a la nueva imagen hallada $D'(x,y)$, una ecualización de histograma, normalizando los valores entre 0 y 255. Y el resultado es como el que se muestra en la Figura 3.6(b).

(a) (b)

FIGURA 3.6: Información de la profundidad de la mano. (a) Muestra la información de profundidad sin procesar. (b) Información de profundidad umbralizada y normalizada.

## 3.4. Aplicación de los descriptores la mano

Hasta el momento se tiene una serie de puntos que describen visualmente la forma de la mano (Sección 3.3.2), y la información en una imagen de profundidad (Sección 3.3.3), ahora hay que encontrar la manera de poder caracterizar la forma de la mano, y que tenga ciertas propiedades para que puedan ser reconocidas a diferentes escalas, en distintas posiciones y con varias rotaciones, esto para que sea más fácil el reconocimiento entre una personas y otra, con los gestos a diferentes distancias y en distintos lugares. Y esta es la labor de los descriptores, intentar caracterizar la forma de la mano de la mejor manera y con una cantidad de datos aceptable. En esta sección se describen la utilización de los diferentes descriptores propuestos en la Sección 2.4.

### 3.4.1. Descriptores de Fourier

Con los datos del contorno ecualizados (3.3.2.1), se aplica ahora la transformada de Fourier descrita en la Sección 2.4.1.

Este descriptor se aplica al contorno, por tanto, la lista de puntos $(x, y)$ ordenada que se obtiene del contorno se convierte, a través de la Ecuación 2.1, en una lista ordenada de números complejos, los cuales son reemplazados en la Ecuación 2.2 y se utilizaron 2 valores para $K$, $K = 32$ y $K = 64$, después de obtener los diferentes $a(u)$ se aplican varias transformaciones de las ecuaciones 2.3, 2.4 y 2.5, para hacer los descriptores invariantes a la escala, a el punto de inicio y a el desplazamiento.

Los vectores de características obtenidos son de valores reales de tamaño 32 y 64 (a partir de los $K$).

### 3.4.2. Descriptores de Fourier Invariantes 2

Con los datos del contorno ecualizados (3.3.2.1), se aplica ahora la modificación de las características invariantes descrita en la Sección 2.4.2.

Con los mismo valores de $a(u)$ hallados en Sección 3.4.1, antes de aplicar las transformadas propuestas en esa sección, y con los mismos valores de $K = 32$ y $K = 64$, se utilizan en la Ecuación 2.6 obteniendo dos nuevos vectores de características con características invariantes.

Los vectores de características obtenidos son de valores reales de tamaño 32 y 64 (a partir de los $K$).

### 3.4.3. Descriptores Elípticos de Fourier

Con los datos del contorno ecualizados (3.3.2.1), lo primero que hay que hacer es representar el contorno como una cadena de Freeman y después se aplican las formulas para hallar los descriptores del contorno como de describe en la Sección 2.4.3.

Como se mencionó antes, lo primero que hay que hacer es hallar la cadena de Freeman del contorno de la mano, para esto hay que hacer la codificación a partir de los datos de posición que se tienen, recordemos que el contorno esta dado por una lista de coordenadas $(x, y)$, por tanto, la codificación se hace restando las respectivas coordenadas, se analizan los signos resultantes de la operación y a través de una tabla de conversiones de obtiene la codificación de los número del 0 al 7.

Ya teniendo la cadena de Freeman, el siguiente paso es hallar los valores de $dt_i$, $t_n$, $dx_i$ y $dy_i$ de las ecuaciones 2.8, 2.9, 2.10 y 2.11 respectivamente. Y por último hallamos los coeficientes que van a describir la forma de la mano, se hicieron pruebas caracterizando diferentes formas con $K = \{16, 32, 64\}$ y siempre los mejores resultados se obtuvieron con $K = 16$, hay que tener en cuenta que por cada descriptor se tienen cuatro valores $(a, b, c, d)$, por tanto después de calcular los coeficientes invariantes partiendo de la Ecuación 2.12 se obtiene un vector real de 64 datos.

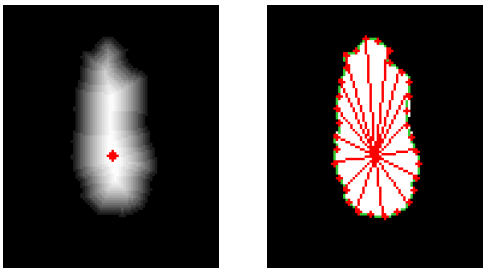
### 3.4.4. Función de distancia y tangente

Este descriptor se aplica al contorno de la imagen hallado en la Sección 3.3.2.1, lo primero que se hace es hallar el centro de la mano, esto se hace aplicando una transformada de distancia a la imagen de la zona de la piel (Figura 3.4(c)) y después se elije el punto de mayor intensidad, el centro de la mano, un ejemplo se puede ver en la Figura 3.7(a), en donde el punto rojo representa el centro de la mano, y a partir de esta referencia se lanzan rayos a todos los puntos del contorno hallados anteriormente(Figura 3.7(b)), a cada uno de estos rayos se toman 2 informaciones, la primera es la distancia euclidiana del centro de la mano al punto de análisis, y para normalizar la distancia y que sea independiente del tamaño de la mano, se divide este valor por la longitud del perímetro($L$), y por ultimo se halla el punto tangente de la siguiente forma,

$$r = \frac{\sqrt{(x_n - x_c)^2 + (y_n - y_c)^2}}{L} \quad \text{y} \quad \theta = tan^{-1}\left(\frac{y_n - y_c}{x_n - x_c}\right)$$

donde $x_c$, $y_c$ son las coordenadas del punto centro de la mano, y $x_n$, $y_n$ con las coordenadas del punto $n$ del contorno que se esta analizando.

Por tanto para este descriptor, por punto del contorno se tienen dos valores, lo que quiere decir que la longitud del vector característico va a ser dos veces los puntos del contorno.



(a) (b)

FIGURA 3.7: Función de Distancia y la tangente. (a) Muestra la transformada de distancia, y el punto rojo que representa el centro de la mano. (b) Rayos desde el centro a cada uno de los puntos del contorno.

### 3.4.5. Momentos Invariantes

#### Contorno

Con los datos del contorno encontrado en la Sección 3.3.2.1, se calculan los 7 momentos invariantes según la Ecuación 2.23, por tanto en esta sección, siempre se va a obtener un vector de características fijo de tamaño 7 y de coeficientes reales.

#### A la imagen de profundidad

En esta parte, los descriptores no se van a aplicar al contorno, sino que esta vez se van a utilizar en la imagen de profundidad umbralizada y normalizada encontrada en la Sección 3.3.3, se hallan los *Momentos centrales*(2.21), los *Momentos centrales normalizados*(2.22) y por último se calcula los 7 momentos invariantes propuestos en la Ecuación 2.23, e igual que con el contorno, se obtiene un vector de características con 7 valores reales.

### 3.4.6. Análisis de Componentes Principales

Este método se aplica sobre la imagen de profundidad umbralizada y normalizada encontrada en la Sección 3.3.3, como se muestra en la Sección 2.4.5, se ejecuta el análisis de componentes principales, se calculan los $\mathbf{m_x}$ y los $\mathbf{C_x}$, se obtiene los eigenvalores y 32 de estos son los que se van a tomar como el vector característico de la imagen que representan.

## 3.5. Experimentos, entrenamiento y resultados

Resumiendo lo que se tiene hasta ahora, se han generado vectores característicos del contorno de una imagen de color y de la imagen de profundidad, y se listan a continuación,

- **Descriptores de Contorno**
  - Descriptor de Fourier Invariante. *(DFI)*
  - Descriptor de Fourier Invariante 2. *(DFI2)*
  - Descriptor Elíptico de Fourier. *(DEF)*
  - Momentos Invariantes. *(MIC)*
  - Función de distancia y la tangente. *(FDT)*
- **Descriptores de imagen de profundidad**
  - Momentos Invariantes. *(MII)*
  - PCA.
- **Combinaciones predeterminadas**
  - PCA y Descriptor de Fourier Invariante. *(PCA_DFI)*
  - Momentos Invariantes Imagen y Descriptor de Fourier Invariante. *(MII_DFI)*
  - PCA y Función de distancia y la tangente. *(PCA_FDT)*

Y se prueban variaciones en cuanto a la cantidad de puntos tomados para representar el contorno, se prueba con 32 y 64 puntos por contorno, esto afecta a todos los descriptores de las diferentes formas de fourier, y adicionalmente se varia la cantidad de coeficientes que se toman.

### 3.5.1. Pre-procesamiento de datos

Antes de pasarle los datos al clasificador, hay que hacer un pre-procesamiento de los datos, para este caso se implementa el promedio y desviación estándar de los datos.

Si se define una vector que incluye algún tipo de descriptor de tamaño $n$, $\mathbf{x} = \{x_1, \ldots, x_n\}$, por tanto el vector de $m$ características estaría dado por,

$$\mathbf{X} = \begin{bmatrix} \mathbf{x_1} \\ \vdots \\ \mathbf{x_m} \end{bmatrix}.$$

Se calcula el vector de valores promedio ($\mu$) de los datos por columna,

$$\mu(i) = \frac{1}{m}\sum_{k=1}^{m}\mathbf{X}(k,i) \tag{3.1}$$

donde $i = 1,\ldots,n$, y ahora se calcula el vector de desviación estándar($\sigma$),

$$\sigma(i) = \sqrt{\frac{1}{m-1}\sum_{k=1}^{m}\left(\mathbf{X}(k,i) - \mu(i)\right)^2} \tag{3.2}$$

y por último se aplica la normalización de los datos,

$$\mathbf{X}'(j,i) = \frac{\mathbf{X}(j,i) - \mu(i)}{\sigma(i)} \tag{3.3}$$

donde $i = 1,\ldots,n$ y $j = 1,\ldots,m$. Con esto se obtiene el nuevo vector de características normalizado $\mathbf{X}'(j,i)$ el cual ahora tiene media 0 y desviación estándar 1.

### 3.5.2. Adquisición de datos

Para la adquisición de los datos de entrenamiento y pruebas se utiliza el Kinect y un programa que se instala con el SDK desarrollado por Microsoft llamado kinect studio([43]) y permite almacenar en archivos .xed la información de color y de profundidad a través de una aplicación que esté conectada al Kinect, y por esta misma aplicación se pueden reproducir los archivos almacenados, un inconveniente de estos archivos es, que el formato en que se almacena es muy grande.

La información de las letras se adquieren de 6 personas, que no son hablantes del lenguaje de señas, se adquiere un promedio de 150 imágenes de cada letra por persona, el total de datos por letras que se tiene se muestra en la Tabla 3.1, y en la Figura 3.8 se puede ver unas imágenes de la letra C realizada por las 6 personas, pertenecientes al conjunto de datos.

Para esta captura, se les pidió el favor a las personas que utilizaran un brazalete negro, y se necesita un fondo oscuro y que la persona no vista prendas de color similar al de la piel o muy claros.

| Letra | # Muestras |
|---|---|
| A | 1215 |
| B | 1135 |
| C | 1106 |
| D | 1103 |
| E | 1136 |
| H | 1143 |

| Letra | # Muestras |
|---|---|
| I | 1054 |
| L | 1154 |
| M | 987 |
| N | 1049 |
| O | 1181 |
| Q | 920 |

| Letra | # Muestras |
|---|---|
| R | 938 |
| U | 1003 |
| V | 1070 |
| W | 756 |
| Y | 907 |
| **Total** | **17857** |

TABLA 3.1: Cantidad de muestras de Letras.

(a) (b) (c) (d) (e) (f)

FIGURA 3.8: las imágenes de la (a) a la (f) se muestra un ejemplo de la letra C del conjunto de datos

### 3.5.3. Clasificación SVM

Ya se tiene los datos listos que caracterizan a cada una de las letras, ahora lo que se implementa es un algoritmo de aprendizaje que para este caso se utiliza un algoritmo de aprendizaje supervisado llamado Máquinas de Vectores de Soporte (o por sus siglas en ingles SVM).

Se utiliza un C-SVM con un kernel de base radial(RBF) que esta definida en la Ecuación 3.4, por tanto los parámetros a encontrar son $C$ y $\gamma$, siendo $C$ el parámetro de regularización. La búsqueda de estos parámetros se hace de forma heurística a través de una grilla de valores que varían tanto a $C$ como a $\gamma$ e intentan encontrar la mejor combinación que dé los mejores resultados. La búsqueda de estos parámetros y entrenamiento se aplica validación cruzada para evitar el sobre-ajuste sobre los datos de entrenamiento y se realiza en 3-iteraciones. Como las etiquetas de los datos de entrenamiento son mas de dos hay que utilizar un SVM multi-clase, el utilizado en este proyecto es uno-contra-uno.

En la Tabla 3.1 están todos los datos disponibles para el aprendizaje, pero todos los datos nos son utilizados para el entrenamiento, se parten los datos de forma que queden un 70 % de los datos para el entrenamiento y un 30 % para las pruebas, esto quiere decir que el clasificador SVM en la fase de entrenamiento solo va a ver el 70 % del total de los datos.

$$K(x, x') = exp\left(\gamma \|x - x'\|^2\right) \quad \text{donde} \quad \gamma = \frac{-1}{2\sigma^2} \tag{3.4}$$

### 3.5.4. Resultados

Como se planteó al inicio de la sección, se tiene diferentes tipos de descriptores, tanto para el contorno como para la imagen, y no se sabe de antemano cual es el mejor, por tanto, se hace una combinación de todos ellos y como resultado se tiene un total de 135 posibles conjuntos.

Se hacen dos pruebas, la primera es tomando todas las 6 personas y dividir los datos de entrenamiento entre el total de ellas en 70 % y 30 % de pruebas, se entrenan con el clasificador descrito en 3.5.3, en la Tabla 3.2 se ven los resultados de las diez mejores combinaciones, y en la Tabla 3.3 se pueden ver el porcentaje de aciertos por cada una de las letras, cuando se entreno con la mejor combinación que para esta caso fue DFI2_PCA_FDT(32),

| **Descriptor** | **aciertos( %)** |
|---|---|
| DFI2_PCA_FDT(32) | 99.67 |
| MIC_PCA_FDT(32,64) | 99.66 |
| DFI2_PCA_FDT(64) | 99.61 |
| DFI_PCA_FDT(32) | 99.53 |
| DEF_PCA_DFI(32) | 99.53 |
| DFI_PCA_FDT(64) | 99.43 |
| DEF_PCA_DFI(64) | 99.42 |
| DEF_PCA_FDT(32) | 99.39 |
| DEF_PCA_FDT(64) | 99.39 |
| MIC_PCA_DFI(32) | 99.38 |

Tabla 3.2: Resultado entrenamiento Letras Prueba 1.

| **Letras** | **aciertos( %)** |
|---|---|
| A | 100 |
| B | 100 |
| C | 99.70 |
| D | 100 |
| E | 100 |
| H | 100 |
| I | 100 |
| L | 100 |
| M | 98.64 |
| N | 98.08 |
| O | 99.71 |
| Q | 99.27 |
| R | 100 |
| U | 100 |
| V | 99.37 |
| W | 100 |
| Y | 99.63 |

Tabla 3.3: Resultado entrenamiento Letras con DFI2_PCA_FDT(32).

Y la segunda prueba fue tomar los datos de 5 personas y dividirlos 70 % de entrenamiento y 30 % de pruebas, con la diferencia de que se dejan todos los datos de una persona solo

para pruebas. Con el clasificador descrito en 3.5.3, en la Tabla 3.4 se ven los resultados de las diez mejores combinaciones, y en la Tabla 3.5 se pueden ver el porcentaje de aciertos por cada una de las letras, cuando se entreno con la mejor combinación que para esta caso fue DFI_PCA_FDT(32),

| **Descriptor** | **aciertos( %)** |
|---|---|
| DFI_PCA_FDT(32) | 94.70 |
| DFI2_PCA_FDT(32) | 94.55 |
| MII_DFI _PCA_FDT(32) | 94.23 |
| DFI_PCA_FDT(32) | 93.71 |
| DEF_PCA_DFI(32) | 93.71 |
| DFI_DEF(32) | 93.63 |
| DFI2_DEF(32) | 93.61 |
| MIC_PCA_FDT(32) | 93.48 |
| MII_DFI_MIC(32) | 92.50 |
| DFI_DEF(32) | 92.21 |

TABLA 3.4: Resultado entrenamiento Letras Prueba 2.

| **Letras** | **aciertos( %)** |
|---|---|
| A | 95.37 |
| B | 100 |
| C | 100 |
| D | 83.22 |
| E | 98.85 |
| H | 100 |
| I | 99.61 |
| L | 97.58 |
| M | 98.72 |
| N | 84.61 |
| O | 98.85 |
| Q | 96.47 |
| R | 96.32 |
| U | 99.14 |
| V | 92.19 |
| W | 96.96 |
| Y | 68.88 |

TABLA 3.5: Resultado entrenamiento Letras con DFI_PCA_FDT(32) .

Y por último se muestra en la Tabla 3.6, el desempeño de cada uno de los clasificadores sin combinación alguna,

| **Descriptor** | **aciertos( %)** |
|---|---|
| DFI(32) | 86.48 |
| DEF(32,64) | 85.58 |
| DFI2(32) | 80.30 |
| DFI(64) | 78.61 |
| MIC(32,64) | 77.34 |
| DFI2(64) | 76.24 |
| FDT(64) | 74.86 |
| FDT(32) | 74.37 |
| DFI(32) | 69.25 |
| DFI(64) | 65.62 |
| PCA(32,64) | 58.76 |
| MII(32,64) | 57.56 |

TABLA 3.6: Resultado entrenamiento Letras Básicas.

### 3.5.5. Conclusiones

Los resultados obtenidos en la primera prueba, demuestra que se tienen buenos resultados con los descriptores propuestos, y los porcentajes tan altos de aciertos podrían indicar que hay datos similares en la base de datos.

En la segunda prueba, los resultados obtenidos son muy satisfactorios, se tiene un muy buen desempeño con datos de una persona que es totalmente ajeno al entrenamiento, las dos letras que presenta más errores al clasificar es la D y la Y.

En todos los mejores resultados se ve que la utilización del PCA es una muy buena opción cuando se quieren buenos resultados, y además representa que la información almacenada en la imagen de profundidad es importante.

Los Descriptores por si solos tienen (algunos) un buen rendimiento, pero se ve notoriamente mejorado cuando se combinan con más de uno de ellos.

# Capítulo 4

# Reconocimiento de Letras con Movimiento

En este capítulo se muestra el reconocimiento de las letras del alfabeto que incluyen el movimiento como es el caso de la G, J, S y Z. Como una solución se presentan varios descriptores, utilizando todas las herramientas planteadas en el Capítulo 3 y adicionando el uso de HMMs. La forma como de hace una palabra se puede encontrar en [41].

## 4.1. Detección y seguimiento de la mano

La detección y seguimiento de la mano se realiza de la misma forma como se hizo en el Capítulo 3, se basa en la información que brinda el Kinect del esqueleto, y se utilizan las misma técnica de modelo de color y de la generación del contorno de la mano.

## 4.2. Descriptores

A diferencia del Capítulo 3, en este tipo de reconocimiento se necesita caracterizar un movimiento, y para describirlo, se propone, saber que como está la forma de la mano,

cuanto y hacia donde se movió, por eso, para las letras en movimiento se proponen tres tipos de descriptores,

- Forma de la mano
- Descriptores de Movimiento
- Descriptores de Dirección

cada uno de ellos se explicara en el desarrollo de la sección. Para describir la letra solo se tiene en cuenta la posición de la mano sin ninguna referencia adicional.

### 4.2.1. Forma de la mano

Para poder describir la forma de la mano, se van a utilizar todas las herramientas descritas en el Capítulo 3, por tanto, se hace una base de datos con las formas de la mano de las diferentes letras sobre todo el recorrido que hace la mano.

Para obtener los descriptores de la mano, se utilizan los descriptores que mejor se comportaron y se adiciona unos que considero se comportaron bien, que incluyen solo la información del color, debido a que en algunas letras como la Z, la mano se acerca mucho al cuerpo y genera sombras o datos adicionales indeseables en la información de profundidad.

También se puede ver que en ciertas partes del movimiento de las manos de dos letras diferentes, hay formas que tienen el contorno similar.
Viendo esto, se propone para esta tarea, incluir una técnica de Machine Learning llamada **K-Means**([44]), que es una técnica de agrupamiento que divide los datos en K partes, en donde una observación se asigna al grupo en donde menor distancia haya de la media de este a la observación, se hace varias pasadas hasta que se logre un cierto equilibrio, como por ejemplo cuando las asignaciones no cambien, el objetivo es minimizar la suma de los cuadrados en cada uno de los grupos. Se puede utilizar diferentes funciones de distancias, la utilizada para este proyecto es la distancia euclidiana.

Dicho lo anterior, antes de que los datos se pasen al algoritmo de aprendizaje *SVM*, se ponen todos los datos en una misma "bolsa" sin etiqueta alguna y se les aplica *K-Means*

con $K = 8, 10, 12, 15$, y las etiquetas de estos nuevo datos, no son las de la letra que representan, sino las que da como resultado el algoritmo de agrupamiento.

En resumen, se utilizan los siguientes descriptores de la forma de la mano,

- Descriptor de Fourier Invariante. *(DFI)*
- Descriptor de Fourier Invariante 2. *(DFI2)*
- Descriptor Elíptico de Fourier. *(DEF)*
- Momentos Invariantes. *(MIC)*
- Función de distancia y la tangente. *(FDT)*

Se utilizan solo los descriptores que incluyen el color, debido a que en las pruebas iniciales se comportaron mejor que los que incluían la información de profundidad. Y adicionalmente se prueba cada uno de ellos con sin agrupamiento y con los agrupamiento propuesto, generando 25 posibles combinaciones de los descriptores.

### 4.2.2. Dirección de Movimiento

En esta sección se presentan tres tipos de descriptores de movimiento, representa la cantidad de desplazamiento realizado por una persona cuando describe el movimiento de una letra. Todos se basan en la posición de la mano.

#### Deltas de Movimiento

Los deltas de movimientos simplemente mira la cantidad de desplazamiento realizado entre dos frames consecutivos, si se tienen dos puntos $p_0$ y $p_1$, el delta de movimiento se daría como,

$$\Delta M = p_1 - p_0$$

en cada frame se tiene un conjunto de tres valores que representan este descriptor.

#### Deltas de Movimiento Inicial

Este descriptor es muy parecido al anterior, lo único que cambia es que la diferencia ya no se hace entre dos frames consecutivos, sino que ahora $p_0$ es el punto en donde inicia el

movimiento, y lo que se quiere representar con este descriptor es, en cada punto cuanto cambio se realizó con respecto a el punto en donde inició el movimiento.

#### Cantidad de Desplazamiento

Esta muy relacionado con los dos anteriores, pero la diferencia es que aquí se obtiene no la resta de las coordenadas, sino se obtiene una distancia, y esta es medida a través de la formula euclidiana. Por tanto si se tienen los puntos $p_0$ y $p_1$, la distancia entre ellos estaría dado por,

$$d = |p_1 - p_0|$$

### 4.2.3. Descriptores de Dirección

En esta sección se describen las características de hacia donde se mueve la mano cuando una persona cuando describe el movimiento de una letra, para esto se proponen 4 características, que son obtenidas por cada frame y que se describen a continuación.

#### Discretizar direcciones 3D

Cuando se discretiza un movimiento en 2D se puede hacer con 9 posibles posiciones, las primeras 4 de abajo, arriba, derecha e izquierda, las 4 diagonales separadas 45 grados, y por ultimo la de no movimiento. Ahora, se quiere discretizar pero ahora en 3-dimensiones, esto genera 27 posibilidades de desplazamiento, incluido el nulo. Por tanto, para aplicarlo a los datos que se tienen, que son posiciones $(x, y, z)$, se hace una resta entre dos puntos consecutivos, se analizan los signos resultantes y a través de una tabla se obtiene la codificación de 0 a 26. Como se puede el descriptor resultantes es un único número.

#### Cosenos Directores

Para calcular los cosenos directores de un vector $\vec{v} = (x, y, z)$, se hace a través de la fórmula,

$$cos(\alpha) = \frac{x}{|\vec{v}|}, \quad cos(\beta) = \frac{y}{|\vec{v}|}, \quad cos(\gamma) = \frac{z}{|\vec{v}|} \tag{4.1}$$

En este caso los cosenos directores se obtienen partiendo del vector que se genera de las coordenadas de la posición de la mano dos frames consecutivos, y los valores de $\alpha$, $\beta$ y $\gamma$ son los valores que se utilizan para describir la dirección de movimiento en un frame dado.

#### Ángulo dos Vectores

Se calcula el ángulo de los dos vectores que se forman en tres frames consecutivos. Se obtienen 3 coordenadas de posición de la mano, $P_0$, $P_1$ y $P_2$, se hallan dos vectores $\vec{u} = P_1 - P_0$ y $\vec{v} = P_1 - P_2$ y el ángulo formado por estos dos vectores está descrito por,

$$cos(\alpha) = \frac{\vec{u} \cdot \vec{v}}{|\vec{u}||\vec{v}|} \tag{4.2}$$

Y se toma el ángulo $\alpha$ resultante como descriptor de la dirección de movimiento un frame.

#### Ángulo de rotación

Como se menciono en la Sección 2.3.1, el kinect por cada punto clave del esqueleto devuelve una matriz que representa la rotación de ese punto con respecto al eje de coordenadas de la cámara, normalmente es utilizada para las animaciones, pero en esta caso se utiliza estos datos, como una característica de dirección, representando la rotación de la mano, la cual es expresada a través de 3 ángulos que corresponden a la rotación que hay que hacer en cada uno de estos.

## 4.3. Experimentos, entrenamiento y resultados

Resumiendo lo que se ha dicho hasta ahora, se han generado descriptores que caracterizan un movimiento teniendo en cuanta, la forma de la mano durante la realización de la letra, el movimiento que realiza, en cuanto a desplazamiento y dirección, y se puede resumir en,

- **Forma de la mano (con K-means)**
  - Descriptor de Fourier Invariante. *(DFI)*

    - Descriptor de Fourier Invariante 2. *(DFI2)*
    - Descriptor Elíptico de Fourier. *(DEF)*
    - Momentos Invariantes. *(MIC)*
- **Dirección de Movimiento**
    - Deltas de Movimiento
    - Deltas de Movimiento Inicial
    - Cantidad de Desplazamiento
- **Descriptores de Dirección**
    - Discretizar direcciones 3D
    - Cosenos Directores
    - Ángulo dos Vectores
    - Ángulo de rotación

Cada uno de los ítem representa un vector característico, los cuales son combinados para elegir cual es la mejor combinación.

A diferencia de las imágenes estáticas, en este caso se tiene por letra un grupo de características que han sido obtenidas a través de todos los frames que componen el movimiento de la letra, y cada uno de estos tiene un significado temporal.

### 4.3.1. pre-procesamiento de datos

El pre-procesamiento de los datos es el mismo que está descrito en 3.5.1

### 4.3.2. Adquisición de datos

La adquisición de datos se hace de forma similar como se describió en la Sección 3.5.2, se utiliza el Kinect y la aplicación kinect studio, para grabar las imágenes.

La información de las letras son adquiridas de 6 personas, que no son hablantes del lenguaje de señas, se captura un promedio de 15 repeticiones de una letra por persona, el total de letras capturadas de las 6 personas se muestra en la Tabla 4.1,

Las letras capturadas se le pide que tengan un movimiento continuo con pausas largas durante su ejecución, pero si es necesario que antes de iniciar el movimiento y el finalizar haya un momento donde la mano esté estática por al menos 5 segundos.

| **Letra** | **# Muestras** |
|---|---|
| G | 100 |
| J | 120 |
| S | 92 |
| Z | 94 |

TABLA 4.1: Cantidad de muestras de letras con movimiento.

### 4.3.3. Clasificación con HMM

Como se describe en la Sección 2.5, HMM han sido utilizado y exitoso en el moldeamiento de características temporales. Para este proyecto, se propone un modelo de Markov con una topología como se muestra en la Figura 2.4, que aunque no es exactamente como sucede en el lenguaje de señas la aproximación es buena. El parámetro heurístico en los HMM es la cantidad de estados ocultos, que en un problema como este no se sabe exactamente cual la mejor cantidad de estado que representa a una letra, una aproximación basada en una propuesta presenta por Kelly et al. ([18]), donde si se tiene un conjunto de observaciones $\{O_0, O_1, \ldots, O_{k-1}\}$ de una misma letra, a cada una de ellas se les aplica *PCA* y se obtienen los componentes principales de la observación en análisis $O_j$ y a los componentes hallados se hace una busqueda a través del algoritmo de agrupamiento, *K-means*, para elegir cual es mejor valor de K que divide los datos; para saber cual es mejor valor de K, se analizan las distancias de las medias de los grupos hallados por el algoritmo, y el que mayor distancia tenga es el que se elige como $K^*$. Este procedimiento se repite para todos los $O$, y al final se hace un promedio entre todos los $K_j^*$ hallados y ese es la cantidad de estados en el que se comienza la búsqueda de la cantidad de estado que minimiza el error en los datos. El algoritmo de entrenamiento utilizado es el de Baum-Welch. Se entrena un HMM por cada letra.

### 4.3.4. Resultados

Los vectores de características siempre están compuestos de un descriptor de la forma de la mano, de un descriptor de movimiento y uno de dirección, por tanto, se hacen todas las posibles combinaciones entre ellos y se entrenan para obtener el que menor error tenga sobre los datos.

Igual que se hizo en el capítulo anterior, el total de los datos que se muestran en la

Tabla 4.1. se dividen en un 70 % para los datos de pruebas y un 30 % para los datos de entrenamiento, y los resultados la mejor combinación se presentan a continuación,

- **Forma de la mano (con K-means)**
  - Descriptor Elíptico de Fourier($K = 12$)
- **Dirección de Movimiento**
  - Deltas de Movimiento
- **Descriptores de Dirección**
  - Discretizar direcciones 3D
- **Cantidad de estados HMM**
  - G:16, J:40, S:8, Z:32

En la Tabla 4.2, se muestra la cantidad del aciertos del clasificador HMM descrito, sobre los datos de pruebas, en promedio, se tuvo un porcentaje de 93.82 % de aciertos,

| **Descriptor** | **aciertos( %)** |
|---|---|
| G | 90.47 |
| J | 86.67 |
| S | 100 |
| Z | 100 |

TABLA 4.2: Resultado entrenamiento Letras con movimiento.

### 4.3.5. Conclusiones

Las letras G y J, son las que menos porcentaje de aciertos tiene, esto se puede deber a la similitud en la forma de la mano y el recorrido al ejecutar la letras. Se puede ver también que realizar agrupamiento sobre las formas de la mano presentan un mejor comportamiento que si se entrenaran sin este. no hay problemas en el reconocimiento de las letras S y Z, debido a que la forma de la mano, y el movimiento realizado por la mano son muy diferentes a los demás, para mejorar los resultados sobre G y J, es posible implementar otros descriptores o hacer un filtro de los datos entes de el entrenamiento.

# Capítulo 5

# Reconocimiento de Palabras

En esta sección se describe el reconocimiento de algunas palabras del alfabeto de sordos colombiano, que incluyen una sola mano, y se utilizan las técnicas y herramientas utilizadas en el Capítulo 4, de los cual se puede ver que la diferencia básica es la descripción de los descriptores, pero la parte de entrenamiento y concepto e similar.

## 5.1. Detección y seguimiento de la mano

La detección y seguimiento de la mano se realiza de la misma forma como se hizo en el Capítulo 3, se basa en la información que brinda el Kinect del esqueleto, y aunque se utilizan las misma técnica de modelo de color y de la generación del contorno de la mano, para realizar el análisis de la forma de la mano, se les pide a los colaboradores que utilicen un guante de color azul para ejecutar la palabra, se elige un color que pueda ser expresado a través del mismo modelo de color planteado en el Capítulo 3.

## 5.2. Descriptores

Como hemos visto a través de el Capítulo 3 y el Capítulo 4 la elección de los descriptores adecuados es muy importante para el buen desarrollo del clasificador.

Para este caso donde se quiere reconocer una palabra, se tiene que tener en cuenta aspectos que no se habían tenido en cuenta con las letras, como es la relación entre la mano y el cuerpo de la persona que lo ejecuta, debido a que es importante saber, con respecto al cuerpo, donde inicia y finaliza la palabra. por tanto los descriptores necesitan que diferencien entre las formas de la mano,el movimiento respecto a un punto de referencia de la persona, dicho esto se proponen los siguientes descriptores.

### 5.2.1. Forma de la mano

Para la forma de la mano, de igual que en la Sección 4.2.1, se utiliza la información de color, y algunos de los mejores descriptores, pero no se utiliza K-means, en cambio de ello lo que se implementa es tomar cada uno de los frames que representa a una palabra, y dividir el total de los frames en tres partes, y solo se van a tomar las imágenes del primer tercio y del último tercio del recorrido, esto se hace con el fin de evitar formas de la mano que no ayudan en absoluto a su clasificación, y debido a la resolución de las imágenes durante el desplazamiento se presenta un comportamiento llamado blur, que es una distorsión en la imagen.

Los descriptores de la mano que se utilizan son,

- Descriptor de Fourier Invariante. *(DFI)*
- Descriptor de Fourier Invariante 2. *(DFI2)*
- Descriptor Elíptico de Fourier. *(DEF)*
- Momentos Invariantes. *(MIC)*
- Función de distancia y la tangente. *(FDT)*

### 5.2.2. Descriptores de movimiento y dirección

Para los descriptores de movimiento se utiliza la **discretizacion de la dirección 3D** (ver 4.2.3), el **ángulo dos Vectores** (ver 4.2.3) y se integra un nuevo descriptor que consiste en **discretizar la posición en 2D**, esto consiste en expresar la direccion de movimiento como 9 posibles posiciones, las primeras 4 de abajo, arriba, derecha e izquierda, las 4 diagonales separadas 45 grados, y por ultimo la de no movimiento.

### 5.2.3. Distancia Normalizada Cabeza-Mano

Se incluye este nuevo descriptor, para referenciar el movimiento de la mano con respecto a una parte del cuerpo, en este caso la cabeza, en este caso representa la cantidad de movimiento. Para obtener este descriptor se toma la información de posición de la mano y de la cabeza que brinda el kinect, y se calcula la distancia euclidiana de estos dos puntos vistos en un solo frame; este descriptor depende solo del frame actual.

### 5.2.4. Cosenos Directores Cabeza-Mano

Este descriptor también se incluye para describir la referencia Cabeza-Mano, pero en este caso, representa la fase de estos dos puntos. Los cosenos directores se calcula de igual forma que en la Sección 4.2.3 con la diferencia que ahora los puntos involucrados son los de posicion de cabeza y mano, y se toman se un solo frame.

## 5.3. Experimentos, entrenamiento y resultados

Las palabras elegidas para representar son,

- Azul
- Decir
- Fumar
- Gracias
- Oyente
- Valle

todas ellas comparten la característica de que se realizan con una sola mano. se eligieron estas palabras, por que se pueden ver varias formas de mano y varios tipos de movimiento, y hay un caso especial con *Decir* y *gracias*, debido a que para estas dos palabras el movimiento es muy similar, lo único que lo diferencia es la forma de la mano. Su ejecución se puede ver en [45].

Para resumir, los descriptores utilizados son,

- Forma de la mano

    - Descriptor de Fourier Invariante. *(DFI)*
    - Descriptor de Fourier Invariante 2. *(DFI2)*
    - Descriptor Elíptico de Fourier. *(DEF)*
    - Momentos Invariantes. *(MIC)*
    - Función de distancia y la tangente. *(FDT)*
- Descriptores de movimiento y dirección
    - Discretizar direcciones 3D.
    - Discretizar direcciones 2D.
    - Ángulo dos Vectores
- Distancia Normalizada Cabeza-Mano
- Cosenos Directores Cabeza-Mano

de todos los descriptores propuestos se hace una combinación de cada uno de ellos, para elegir el que mejor represente los datos.

### 5.3.1. pre-procesamiento de datos

En el momento que la mano se acerca a la cara, en algunos casos el kinect no da una posición exacta, sino que da una posición aproximada, lo que causa que en algunos datos haya saltos bruscos en la posición, para evitar eso, en los descriptores con valores reales se hace un promedio con una ventana de 5 frames, como es el caso de los ángulos y las distancias, y para los descriptores que incluyen datos discretizados, igual se toma una ventana de 5 frames en donde el valor para todos los datos de esa ventana se elige tomando el valor que más se repita, esto aplica para las formas de la mano, y para los valores discretos de direcciones.

### 5.3.2. Adquisición de datos

La adquisición de datos se realiza de la misma forma que en la Sección 4.3.2, la única diferencia es que en esta sección se utiliza un guante para detectar la mano, esto es debido a que para estas palabras hay mucha interacción entre la mano y la cara, que con el modelo de color utilizado el los otros capítulos seria imposible separar la mano de la cara. La información de las letras son adquiridas de 6 personas, que no son hablantes del

lenguaje de señas, se captura un promedio de 15 repeticiones de una letra por persona, el total de letras capturadas de las 6 personas se muestra en la Tabla 5.1,

| **Palabra** | **# Muestras** |
|---|---|
| Azul | 135 |
| Decir | 132 |
| Fumar | 144 |
| Gracias | 153 |
| Oyente | 167 |
| Valle | 154 |

Tabla 5.1: Cantidad de muestras de palabras.

### 5.3.3. HMM

La parte de entrenamiento y análisis de el modelo HMM no cambia en nada a el presentado en la Sección 4.3.3.

### 5.3.4. Resultados

Los vectores de características siempre están compuestos de un descriptor de la forma de la mano, de un descriptor de movimiento y uno de dirección, Distancia Normalizada Cabeza-Mano y Cosenos Directores Cabeza-Mano, por tanto, se hacen todas las posibles combinaciones entre ellos y se entrenan para obtener el que menor error tenga sobre los datos.

Igual que se hizo en el capítulo anterior, el total de los datos que se muestran en la Tabla 5.1. se dividen en un 70 % para los datos de entrenamiento y un 30 % para los datos de pruebas, en la Tabla 5.2, se muestra los 3 mejores resultados obtenidos, y en la

| **Características** | **Estados** | **Entrenamiento ( %)** | **Pruebas ( %)** |
|---|---|---|---|
| DFI_FDT_Angulo Tres puntos | **4-3-4-3-3-4** | 100 | 99.1 |
| DFI_FDT_Discretizada 2D | **4-4-4-3-4-4** | 100 | 99.1 |
| DFI_FDT_Discretizada 3D | **4-3-4-3-4-4** | 100 | 99.1 |

Tabla 5.2: Resultado entrenamiento Palabras con movimiento.

Tabla 5.3, se presenta una comparación de aciertos por palabra entre las combinación de los tres mejores resultados que se muestran en la Tabla 5.2,

| Palabras | Angulo tres puntos | | Discretiza 2D | | Discretiza 3D | |
|---|---|---|---|---|---|---|
| • | Train( %) | Test( %) | Train( %) | Test( %) | Train( %) | Test( %) |
| Azul | 100 | 100 | 100 | 100 | 100 | 100 |
| Decir | 100 | 100 | 100 | 97.1 | 100 | 100 |
| Fumar | 100 | 97.1 | 100 | 100 | 100 | 97.1 |
| Gracias | 100 | 97.1 | 100 | 97.1 | 100 | 97.1 |
| Oyente | 100 | 100 | 100 | 100 | 100 | 100 |
| Valle | 100 | 100 | 100 | 100 | 100 | 100 |

Tabla 5.3: Resultado entrenamiento Palabras con movimiento.

### 5.3.5. Conclusiones

Los resultados obtenidos en las palabras elegidas son muy satisfactorios, se alcanza un muy buen nivel en el reconocimiento. Y hay un caso especial para resaltar y es el de las palabras decir y gracias, que como se había dicho antes tiene el mismo movimiento, y lo único que varía es la forma de la mano, y en este caso también se obtiene muy buenos resultados. Se podría concluir que los descriptores propuestos y resultantes caracterizan muy bien la palabra.

# Capítulo 6

# Conclusiones y trabajo futuro

Este documento describe como se puede lograr el reconocimiento ciertas letras del lenguaje de señas colombiano, con y sin movimiento. Se muestra como se puede extraer el modelo de color personalizado por persona y como extraer características importantes, de imágenes de color y profundidad, sobre la forma de la mano.

Se pudo ver que para el reconocimiento de gestos con movimientos, se puede utilizar el mismo modelo de clasificación, y lo que habría que ajustar son los descriptores correctos que describan las características que se deben tener en cuanta, con esto se deje un marco de referencia que pude ser utilizado y ampliado diferentes tipos de reconocimiento de diferentes tipos de gestos.

En este trabajo se plantea la adquisición de un modelo de color que es personalizado según las condiciones ambientales y el color de piel de la persona, lo que lleva a que se tenga una buena aproximación de la forma de la mano a través de la imagen de color.

Para el reconocimiento de las palabra es obligatorio el uso de un guante, para que se pueda diferenciar cuando la palabra se ejecuta cerca de la cara. El color del guante debe ser un color pastel, que pueda incluido en el modelo de color propuesto.

Los resultados obtenidos son satisfactorios, en cada uno de las tres secciones se logró el objetivo, y cabe resaltar los resultados del reconocimiento de letras con movimiento que logro aciertos del más del 99 %.

Una de las limitaciones presentadas por este trabajo es que solo se tiene en cuenta el movimiento de una sola mano, pero no sería difícil expandirlo a la utilización de dos manos, utilizando los descriptores correctos.

Este documento marca un inició, para el reconocimiento del gestos o señas en donde la información de la mano es importante, como trabajo futuro, se ve la inclusión de gestos que incluyan las dos manos y que en trabajos posteriores puedan también reconocer gestos manuales y gestos no manuales, así como un posible reconocimiento continuo.

# Bibliografía


[1] Emgu CV. Emgu cv: Opencv in .net (c#, vb, c++ and more). URL http://www.emgu.com.

[2] César Souza. The accord.net framework, 2013. URL https://code.google.com/p/accord/.

[3] D. M. Gavrila. The visual analysis of human movement: A survey. *Computer Vision and Image Understanding*, 73:82–98, 1999.

[4] Helene Brashear, Valerie Henderson, Kwang-Hyun Park, Harley Hamilton, Seungyon Lee, and Thad Starner. American sign language recognition in game development for deaf children. In *Proceedings of the 8th international ACM SIGACCESS conference on Computers and accessibility*, Assets '06, pages 79–86, New York, NY, USA, 2006. ACM. ISBN 1-59593-290-9. doi: 10.1145/1168987.1169002.

[5] P. K. Vijay, N. N. Suhas, C. S. Chandrashekhar, and D. K. Dhananjay. Recent developments in sign language recognition : A review. *International Journal on Advanced Computer Engineering and Communication Technology (IJACECT)*, 1: 21–26, 2012. URL http://www.irdindia.in/Journal_IJACECT/PDF/Vol1_Iss2/5.pdf.

[6] Christian Vogler and Dimitris Metaxas. A framework for recognizing the simultaneous aspects of american sign language. *Computer Vision and Image Understanding*, 81(3):358 – 384, 2001. ISSN 1077-3142. doi: http://dx.doi.org/10.1006/cviu.2000.0895. URL http://www.sciencedirect.com/science/article/pii/S1077314200908956.

[7] Stokoe William. *El lenguaje de las manos*. Fondo de cultura económica, México, primera edition, 2004.

[8] Tejedor C. Antonio. *Mil palabras con las manos.* Ciencias de la educación preescolar y especial, 2001.

[9] Sylvie C.W. Ong and Surendra Ranganath. Automatic sign language analysis: A survey and the future beyond lexical meaning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(6):873–891, 2005. ISSN 0162-8828. doi: http://doi.ieeecomputersociety.org/10.1109/TPAMI.2005.112.

[10] Instituto Caro y Cuervo. La lengua de señas colombiana. URL http://www.lenguasdecolombia.gov.co/content/lengua-de-se%C3%B1as-colombiana.

[11] David Rybach, Thomas Deselaers, Morteza Zahedi, and Hermann Ney. Speech recognition techniques for a sign language recognition system. In *In Interspeech 2007 - Eu rospeech*, 2007.

[12] S.C.W. Ong and S. Ranganath. Automatic sign language analysis: a survey and the future beyond lexical meaning. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 27(6):873–891, 2005. ISSN 0162-8828. doi: 10.1109/TPAMI.2005.112.

[13] J.L. Hernandez-Rebollar, R.W. Lindeman, and N. Kyriakopoulos. A multi-class pattern recognition system for practical finger spelling translation. In *Multimodal Interfaces, 2002. Proceedings. Fourth IEEE International Conference on*, pages 185–190, 2002. doi: 10.1109/ICMI.2002.1166990.

[14] Rung-Huei Liang and Ming Ouhyoung. A real-time continuous gesture recognition system for sign language. In *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*, pages 558–567, 1998. doi: 10.1109/AFGR.1998.671007.

[15] T.T. Swee, A. K. Ariff, S.-H. Salleh, Siew Kean Seng, and Leong Seng Huat. Wireless data gloves malay sign language recognition system. In *Information, Communications Signal Processing, 2007 6th International Conference on*, pages 1–4, 2007. doi: 10.1109/ICICS.2007.4449599.

[16] Daniel Kelly, J. McDonald, T. Lysaght, and C. Markham. Analysis of sign language gestures using size functions and principal component analysis. In *Machine Vision and Image Processing Conference, 2008. IMVIP '08. International*, pages 31–36, 2008. doi: 10.1109/IMVIP.2008.17.

[17] Daniel Kelly, J. McDonald, and C. Markham. Continuous recognition of motion based gestures in sign language. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1073–1080, 2009. doi: 10.1109/ICCVW.2009.5457585.

[18] Daniel Kelly, Jane Reilly Delannoy, John Mc Donald, and Charles Markham. A framework for continuous multimodal sign language recognition. In *Proceedings of the 2009 international conference on Multimodal interfaces*, ICMI-MLMI '09, pages 351–358, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-772-1. doi: 10.1145/1647314.1647387. URL http://doi.acm.org/10.1145/1647314.1647387.

[19] Xiaolong Zhu and K.K. Wong. Single-frame hand gesture recognition using color and depth kernel descriptors. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 2989–2992, 2012.

[20] Nobuhiko Tanibata, Nobutaka Shimada, and Yoshiaki Shirai. Extraction of hand features for recognition of sign language words. In *In International Conference on Vision Interface*, pages 391–398, 2002.

[21] Helen Cooper, Eng-Jon Ong, Nicolas Pugeault, and Richard Bowden. Sign language recognition using sub-units. *Journal of Machine Learning Research*, 13:2205–2231, Jul 2012. URL http://jmlr.csail.mit.edu/papers/volume13/cooper12a/cooper12a.pdf.

[22] J. Isaacs and S. Foo. Hand pose estimation for american sign language recognition. In *System Theory, 2004. Proceedings of the Thirty-Sixth Southeastern Symposium on*, pages 132–136, 2004. doi: 10.1109/SSST.2004.1295634.

[23] Zahoor Zafrulla, Helene Brashear, Thad Starner, Harley Hamilton, and Peter Presti. American sign language recognition with the kinect. In *Proceedings of the 13th international conference on multimodal interfaces*, ICMI '11, pages 279–286, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0641-6. doi: 10.1145/2070481.2070532. URL http://doi.acm.org/10.1145/2070481.2070532.

[24] N. Pugeault and R. Bowden. Spelling it out: Real-time asl fingerspelling recognition. In *Computer Vision Workshops (ICCV Workshops), 2011 IEEE International Conference on*, pages 1114–1119, 2011. doi: 10.1109/ICCVW.2011.6130290.

[25] Simon Lang, Marco Block, and Raúl Rojas. Sign language recognition using kinect. In Leszek Rutkowski, Marcin Korytkowski, Rafał Scherer, Ryszard Tadeusiewicz, LotfiA. Zadeh, and JacekM. Zurada, editors, *Artificial Intelligence and Soft Computing*, volume 7267 of *Lecture Notes in Computer Science*, pages 394–402. Springer Berlin Heidelberg, 2012. ISBN 978-3-642-29346-7. doi: 10.1007/978-3-642-29347-4_46. URL http://dx.doi.org/10.1007/978-3-642-29347-4_46.

[26] Simon Lang. Sign language recognition with kinect. Master's thesis, Freie Universität Berlin, 2011.

[27] Yi Li. Hand gesture recognition using kinect. Master's thesis, University of Louisville, Department of Computer Engineering and Computer Sciences, Mayo 2012.

[28] Daniel James Ryan. Finger and gesture recognition with microsoft kinect. Master's thesis, University of Stavanger, Norway, Department of Electrical and Computer Engineering, 2012. URL http://brage.bibsys.no/uis/bitstream/URN:NBN:no-bibsys_brage_33088/1/Ryan%2c%20Daniel%20James.pdf.

[29] J.L. Raheja, A. Chaudhary, and K. Singal. Tracking of fingertips and centers of palm using kinect. In *Computational Intelligence, Modelling and Simulation (CIMSiM), 2011 Third International Conference on*, pages 248–252, 2011. doi: 10.1109/CIMSim.2011.51.

[30] Zhou Ren, Jingjing Meng, Junsong Yuan, and Zhengyou Zhang. Robust hand gesture recognition with kinect sensor. In *Proceedings of the 19th ACM international conference on Multimedia*, MM '11, pages 759–760, New York, NY, USA, 2011. ACM. ISBN 978-1-4503-0616-4. doi: 10.1145/2072298.2072443. URL http://doi.acm.org/10.1145/2072298.2072443.

[31] A Argyros I Oikonomidis, N Kyriazis. Efficient model-based 3d tracking of hand articulations using kinect. In *British Machine Vision Conference (BMVC)*, pages 132–136, 2011.

[32] Ar Door. Virtual fitting room for topshop, 2011. URL http://ar-door.com/2011/05/virtualnaya-primerochnaya-dlya-topshop/?lang=en.

[33] mirracle. mirracle, 2011. URL http://campar.cs.tum.edu/files/mirracle/.

[34] César Villaseñor. Sdk de kinect: historia de éxito accidental de microsoft, 2011. URL http://www.pcworld.com.mx/Articulos/13303.htm.

[35] Rafael C. Gonzalez and Richard E. Woods. *Digital Image Processing (3rd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006. ISBN 013168728X.

[36] Milan Sonka, Vaclav Hlavac, and Roger Boyle. *Image Processing, Analysis, and Machine Vision*. Thomson-Engineering, 2007. ISBN 049508252X.

[37] Fourier Descriptors. Computer vision and remote sensing. Berlin University of Technology, Noviembre 2009. URL https://visual-attention-processing.googlecode.com/files/Fourier%20Descriptor.pdf.

[38] S. Bourennane and C. Fossati. Comparison of shape descriptors for hand posture recognition in video. *Signal, Image and Video Processing*, 6(1):147–157, 2012. ISSN 1863-1703. doi: 10.1007/s11760-010-0176-6. URL http://dx.doi.org/10.1007/s11760-010-0176-6.

[39] F. P. Kuhl and C. R. Giardina. Elliptic Fourier features of a closed contour. *Graphical Models /graphical Models and Image Processing /computer Vision, Graphics, and Image Processing*, 18:236–258, 1982. doi: 10.1016/0146-664X(82)90034-X.

[40] B. Ballar, P. G. Reas, and D. Tegolo. Elliptical Fourier Descriptors for shape retrieval in biological images.

[41] Sordchannel. Alfabeto lengua de señas colombiano. blog, Noviembre 2011. URL http://nsenate.blogspot.com/2011/12/alfabeto-lengua-de-senas-colombiano.html.

[42] N. Soontranon, S. Aramvith, and T.H. Chalidabhongse. Face and hands localization and tracking for sign language recognition. In *Communications and Information Technology, 2004. ISCIT 2004. IEEE International Symposium on*, volume 2, pages 1246–1251 vol.2, 2004. doi: 10.1109/ISCIT.2004.1413919.

[43] Microsoft. Kinect studio. URL http://msdn.microsoft.com/en-us/library/hh855389.aspx.

[44] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley*

*Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press, 1967.

[45] Colombia Aprende. Lenguaje de señas... un idioma para conocer - vocabulario por orden alfabético. URL http://mail.colombiaaprende.edu.co:8080/recursos/lengua_senas/.