UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

POSGRADO EN CIENCIA E INGENIERÍA DE LA COMPUTACIÓN

SISTEMA HÍBRIDO DE RECONOCIMIENTO FACIAL UTILIZANDO IMÁGENES Y NUBES DE PUNTOS

TESIS
QUE PARA OPTAR POR EL GRADO DE:
MAESTRO EN INGENIERÍA (COMPUTACIÓN)

PRESENTA:
JOSÉ LUIS CRUZ MORA

TUTOR:
DR. BORIS ESCALANTE RAMÍREZ
FACULTAD DE INGENIERÍA

COTUTOR:
DR. JESÚS SAVAGE CARMONA
FACULTAD DE INGENIERÍA

MÉXICO, D. F., ENERO 2016

A mis padres Hilario y Guadalupe, por su apoyo constante y su amor incondicional.
En verdad, gracias por todo.
José Luis.

Reconocimientos

Quisiera reconocer a la Universidad Nacional Autónoma de México por la formación que me ha dado. A mis tutores, Dr. Jesús Savage Carmona y Dr. Boris Escalante Ramírez por sus enseñanzas, consejos y paciencia. Al CONACYT por el apoyo económico que hizo posible el presente trabajo así como a la DGAPA-UNAM por el apoyo proporcionado a través de los proyectos PAPIIT IG100915 y PAPIIT IG100814.

Declaración de autenticidad

Por la presente declaro que, salvo cuando se haga referencia específica al trabajo de otras personas, el contenido de esta tesis es original y no se ha presentado total o parcialmente para su consideración para cualquier otro título o grado en esta o cualquier otra Universidad. Esta tesis es resultado de mi propio trabajo y no incluye nada que sea el resultado de algún trabajo realizado en colaboración, salvo que se indique específicamente en el texto.

JOSÉ LUIS CRUZ MORA. MÉXICO, D.F., ENERO 2016

Resumen

En el presente trabajo se explora el campo de la detección y reconocimiento facial en dos y tres dimensiones. En primer lugar se realiza una revisión de la literatura existente sobre la detección y el reconocimiento facial. Posteriormente se explica la detección en dos dimensiones utilizando cascadas de características tipo Haar y el reconocimiento utilizando eigenfaces. Después se explica la detección en tres dimensiones empleando bosques aleatorios y el reconocimiento utilizando eigenfaces aplicado a imágenes de profundidad. Ambas propuestas son probadas sobre un conjunto de datos a distintas distancias y posiciones y los resultados son comparados.

Con base en los resultados de las pruebas se determina que con el detector en tres dimensiones se obtienen mejores resultados a distancias cortas, mientras que la detección en dos dimensiones resulta mejor a mayores distancias. Además, se observa una mejoría en el proceso de reconocimiento de rostros en tres dimensiones respecto al reconocimiento en dos dimensiones. Finalmente se concluye con la propuesta de un sistema que combina ambas soluciones.

Índice general

Índice de figuras xi

Índice de tablas xiii

1. Introducción 1
- 1.1. Problemática . . . 2
- 1.2. Objetivo . . . 3
- 1.3. Estructura de la tesis . . . 3

2. Marco Teórico 5
- 2.1. Detección de rostros en 2D . . . 6
- 2.2. Reconocimiento de rostros en 2D . . . 8
- 2.3. Detección de rostros en 3D . . . 10
- 2.4. Reconocimiento de rostros en 3D . . . 12

3. Detección y Reconocimiento en 2D 13
- 3.1. Detección del Rostro . . . 14
  - 3.1.1. Extracción de características . . . 15
  - 3.1.2. Clasificación . . . 17
  - 3.1.3. Búsqueda en cascada . . . 18
- 3.2. Preprocesamiento . . . 19
  - 3.2.1. Detección de ojos . . . 21
  - 3.2.2. Alineación . . . 23
  - 3.2.3. Ecualización del histograma . . . 24
- 3.3. Entrenamiento . . . 25
- 3.4. Reconocimiento . . . 25

4. Detección y Reconocimiento en 3D 29
- 4.1. Adquisición . . . 29
- 4.2. Detección . . . 31
  - 4.2.1. Datos de entrenamiento . . . 32
  - 4.2.2. Prueba binaria . . . 33
  - 4.2.3. Clasificación . . . 34

4.2.4. Nodos terminales 35
4.2.5. Proceso de detección 35
4.3. Preprocesamiento 38
4.4. Entrenamiento 40
4.5. Reconocimiento 40

5. Pruebas y Resultados **41**
5.1. Configuración de la prueba 41
5.1.1. Hardware 41
5.1.2. Software 42
5.1.3. Pruebas 43
5.2. Resultados 45
5.2.1. Detección 45
5.2.2. Reconocimiento 48

6. Conclusiones y Trabajo a Futuro **53**
6.1. Conclusiones 53
6.2. Trabajo a futuro 54

A. Configuración para Detección **55**

B. Imágenes para Reconocimiento **57**

C. Cálculo del Error **59**

Bibliografía **61**

Índice de figuras

1.1. Problemas del Reconocimiento Facial 2

2.1. Configuración de un sistema de reconocimiento facial 6

3.1. Proceso de Detección y Reconocimiento en 2D 14
3.2. Características Haar 15
3.3. Imagen Integral 16
3.4. Aplicación de Características Haar 17
3.5. Imágenes de Entrenamiento 18
3.6. Clasificador en Cascada 19
3.7. Detector de Rostros en Cascada 20
3.8. Ecualización de histograma 20
3.9. Zona de búsqueda de ojos 22
3.10. Detección de ojos 22
3.11. Alineación de ojos 23
3.12. Ecualización del Histograma por Partes 24
3.13. Imágenes de entrenamiento 26
3.14. Imagen Promedio 26
3.15. Eigenfaces 27
3.16. Ejemplo de reconstrucción de una imagen 28

4.1. Proceso de Detección y Reconocimiento en 3D 30
4.2. Nube de puntos 30
4.3. Bosque Aleatorio 31
4.4. Muestra de Entrenamiento 3D 34
4.5. Muestras y Votos 36
4.6. Mean Shift 36
4.7. Detección en 3D 37
4.8. Adquisición de imagen de profundidad del rostro detectado. 38
4.9. Comparación de detección ante cambios de iluminación. 39

5.1. Personas de prueba para detección. 43
5.2. Personas de prueba para reconocimiento. 44

5.3. Muestra de detección de rostros en la oscuridad. 48
5.4. Gráficas confiabilidad reconocimiento. 50
5.5. Gráficas de predicción de personas desconocidas. 51

Índice de tablas

3.1. Comparativa de detectores de ojos en OpenCV 21

5.1. Detección de 1 rostro en 2D a tres distancias. 45
5.2. Detección de 2 rostros en 2D a tres distancias. 46
5.3. Detección de 3 rostros en 2D a tres distancias. 46
5.4. Detección de 1 rostro en 3D. 46
5.5. Detección de 2 rostros en 3D. 47
5.6. Detección de 3 rostros en 3D. 47
5.7. Reconocimiento en 2D. 48
5.8. Reconocimiento en 3D. 49

Capítulo 1

Introducción

En los últimos años han surgido varios proyectos de investigación que han demostrado la capacidad de interacción, de una manera natural, entre una computadora y un ser humano. Un claro ejemplo se puede encontrar en el área de la robótica, en donde gracias al rápido crecimiento del poder computacional y el surgimiento de sensores especializados de alta precisión, se ha hecho posible el desarrollo de robots autónomos capaces de interactuar de forma cada vez más natural con los humanos.

Una de las técnicas fundamentales que han permitido tal interacción entre humanos y computadoras es la detección y reconocimiento de rostros. El objetivo de la detección de rostros es determinar la existencia de ninguno, uno o varios rostros en una imagen dada y, de estar presentes, determinar la localización de los mismos. A partir de un rostro previamente detectado y ubicado, otra tarea importante es tratar de identificarlo y reconocer algunas características como el género, la edad, el estado de ánimo, entre otros. Mientras que lo anterior resulta ser una tarea trivial para los seres humanos, para las computadoras es una tarea altamente compleja, que se ha convertido en un tema de investigación muy popular en las últimas décadas [1].

Las fotografías y videos han sido las principales fuentes de datos utilizadas para el reconocimiento facial, y debido a que la adquisición de ésta información no es intrusiva, es decir, no requiere que el usuario realice alguna acción para ser identificado, ésto ha permitido que cobre una gran popularidad entre los sistemas biométricos de identificación, tales como los lectores de huellas dactilares o de escaneo de retina. Al no requerir acciones por parte del usuario, tampoco es necesaria su autorización, ya que basta con ser capturado en una fotografía o video para poder realizar la identificación, convirtiendo al reconocimiento facial en una importante pieza para la video vigilancia no supervisada, tema que ha captado mucha atención y generado mucha polémica en los últimos años.

Sin embargo, aún se tiene varios problemas por resolver debido a que la información disponible para el reconocimiento no siempre es de buena calidad y no siempre es capturada en condiciones ideales, tanto de iluminación y de pose por parte del usuario, lo que dificulta el proceso de reconocimiento quedando aún lejos de replicar la capacidad humana.

1.1. Problemática

La dificultad asociada a la detección y reconocimiento de rostros puede ser atribuida a la gran cantidad de variaciones que se presentan en las imágenes capturadas, tales como los cambios de escala, la orientación del rostro, los gestos faciales, los cambios de iluminación, la oclusión, la edad, entre otros (figura 1.1).

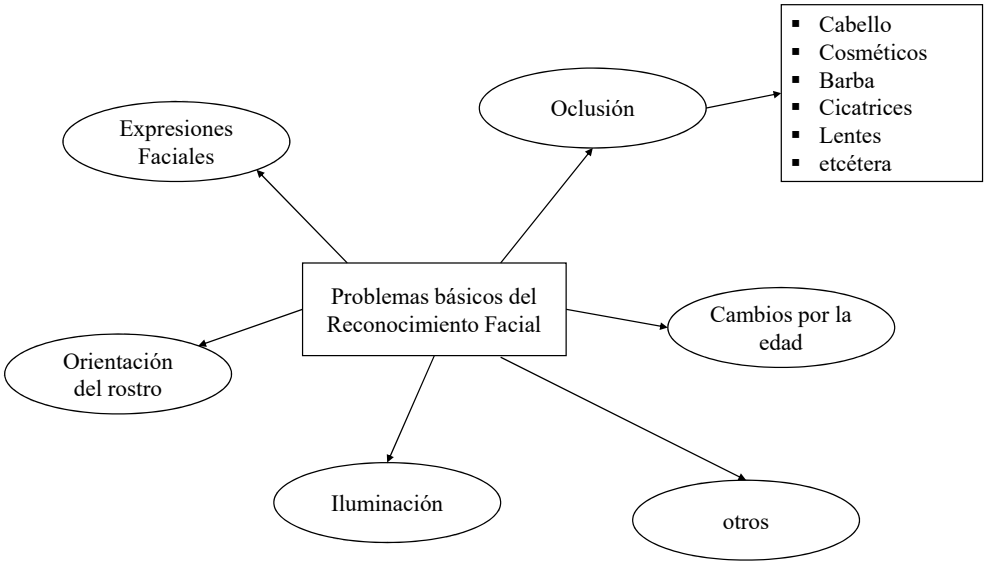


Figura 1.1: Problemas del reconocimiento facial.

A pesar del gran aumento en la calidad y disponibilidad de dispositivos para la adquisición de imágenes, como las cámaras fotográficas, no ha sido posible resolver los problemas ya mencionados. Sin embargo, en la actualidad se han popularizado algunos dispositivos de bajo costo capaces de capturar la distancia a la que se encuentran los objetos respecto al dispositivo, es decir, nos permiten capturar una tercera dimensión correspondiente a la profundidad. Si bien éste tipo de dispositivos han existido desde ya hace varios años, en la actualidad son más accesibles debido a su bajo costo[1] y que la calidad de la información que recaban es aceptable. Con la nueva información es posible enfrentar algunos de los problemas del reconocimiento facial, siendo el problema de la iluminación uno de los más destacados, ya que las mediciones de profundidad no se ven afectadas por los cambios de iluminación.

Si bien se han hecho grandes avances en lo que se refiere a la detección y reconocimiento de rostros utilizando imágenes convencionales, no ha sido posible solucionar el problema con los cambios de iluminación que afectan tanto la detección de rostros como

[1]Por ejemplo el Kinect de Microsoft: https://dev.windows.com/en-us/kinect

la identificación. La detección se ve afectada en imágenes con muy poca o demasiada iluminación, mientras que la identificación se ve afectada por la iluminación, en donde los cambios producidos por las sombras pueden afectar las características requeridas para la identificación. Por otro lado, la información de profundidad puede ayudar a enfrentar tales inconvenientes al no verse afectada por los cambios de iluminación. Lo que resta por hacer es encontrar los algoritmos necesarios que permitan aprovechar dicha información.

1.2. Objetivo

El principal objetivo de este trabajo es proponer un sistema de reconocimiento facial invariante a los cambios de iluminación que haga uso de la información de profundidad que un sensor kinect puede proporcionar. Para lo anterior es necesario encontrar un algoritmo capaz de detectar un rostro en una nube de puntos generada a partir de la información de profundidad obtenida con el kinect. Además, es necesario encontrar un algoritmo que permita el reconocimiento de un rostro a partir de dicha información.

Para cumplir con el objetivo, se partirá de la siguiente hipótesis: Es posible detectar un rostro en una nube de puntos organizada[1], extraerlo y compararlo con otros rostros previamente capturados.

1.3. Estructura de la tesis

El contenido de la presente tesis está organizado para presentar las investigaciones realizadas sobre la detección y reconocimiento facial utilizando información 2D y 3D, lo que permitirá elegir un algoritmo para cada caso, desarrollarlo detalladamente y finalmente realizar una comparación de los resultados obtenidos. El contenido se divide en los siguientes capítulos.

Capítulo 2 - Marco Teórico

En éste capítulo se presenta una revisión de la literatura existente sobre detección y reconocimiento facial, organizada en cuatro secciones: 2.1 detección de rostros en 2D, 2.2 reconocimiento de rostros en 2D, 2.3 detección de rostros en 3D y 2.4 reconocimiento de rostros en 3D.

[1]Una nube de puntos organizada es el nombre que se le da a una nube de puntos estructurada matricialmente en donde los datos se organizan en filas y columnas.

Capítulo 3 - Detección y Reconocimiento en 2D

En éste capítulo se describe el proceso de detección y reconocimiento facial en dos dimensiones utilizando cascadas de características tipo haar y eigenfaces.

Capítulo 4 - Detección y Reconocimiento en 3D

En éste capítulo se describe el proceso de detección y reconocimiento facial en tres dimensiones utilizando bosques aleatorios e imágenes de profundidad.

Capítulo 5 - Pruebas y Resultados

En éste capítulo se presentan la configuración de las pruebas realizadas así como los resultados comparándolos.

Capítulo 6 - Conclusiones y Trabajo a Futuro

En éste último capítulo se presentan las conclusiones obtenidas con base a la comparación de los resultados de las pruebas realizadas y finalmente se propone el trabajo a futuro.

Capítulo 2

Marco Teórico

La identificación de rostros es una capacidad importante, inherente al sistema de percepción humano, y una tarea cotidiana que los seres humanos realizamos todo el tiempo. Sin embargo, automatizar dicha capacidad no es una tarea sencilla siendo aún un tema abierto a la investigación.

El proceso de reconocimiento facial se divide básicamente en tres etapas, tal como se ilustra en la figura 2.1, las cuales se listan a continuación:

- Detección
- Extracción de Características
- Reconocimiento

En la etapa de detección el objetivo principal es encontrar uno o más rostros en una imagen. El origen y la calidad de la imagen puede variar significativamente, y el reto consiste en poder diferenciar entre un rostro y todo el entorno que lo rodea.

En la etapa de extracción de características lo que se busca es obtener un conjunto de datos que representen de manera única al rostro detectado en la etapa anterior. Éste conjunto de características serán almacenadas en una base de datos de rostros conocidos.

En la etapa de reconocimiento el objetivo es identificar un rostro presente en una imagen utilizando sus características. El rostro será identificado siempre que se encuentre almacenado en la base de rostros conocidos.

Se han realizado trabajos enfocados en resolver cada una de las etapas, sobre todo utilizando imágenes 2D como son fotografías y videos, pero también se han realizado estudios utilizando información en 3D, sobre todo en los últimos años debido, en parte, al desarrollo del sensor kinect, un dispositivo de bajo costo que permite obtener, además de la imagen 2D tradicional, una imagen con información de profundidad.

En el presente trabajo se utilizará el término de imagen 2D para referirse a las imágenes obtenidas con cámaras convencionales, que son básicamente un matriz bidimensional que almacena los valores de color o escala de gris que representan una

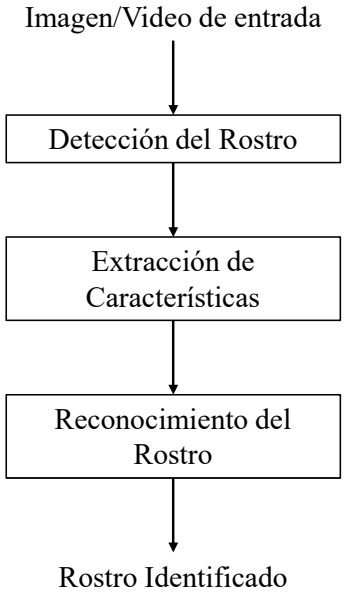


Figura 2.1: Configuración genérica de un sistema de reconocimiento facial.

imagen. También se empleará el término imagen 3D para referirse a las imágenes obtenidas con sensores de profundidad, que son básicamente una matriz bidimensional que almacena la distancia de cada punto de la imagen al sensor.

Como se mencionó anteriormente, en el presente trabajo se presentarán dos sistemas de reconocimiento facial, uno para imágenes 2D y otro para imágenes 3D, se compararán sus fortalezas y debilidades y al final se propondrá una solución que integre las mejores características de ambos, con el objetivo de hacer un sistema robusto a cambios de iluminación.

A continuación se presentan algunos de los trabajos más relevantes en detección y reconocimiento de rostro, tanto con imágenes en 2D como con imágenes 3D, y se presentarán los algoritmos a utilizar en éste trabajo.

2.1. Detección de rostros en 2D

En el campo de la detección de rostros utilizando imágenes 2D se han hecho grandes progresos en las últimas décadas. Los primeros trabajos datan de 1977 con el trabajo de Takeo Kanade[2] como uno de los pioneros en el área. Desde entonces se han desarrollado distintas técnicas para abordar éste problema.

Algunas de las técnicas se basan en la obtención de características de los rostros [3]. En este enfoque se encuentran los métodos que se basan en la obtención de características geométricas y los métodos basados en la captura de movimiento.

Por un lado, los métodos geométricos se basan en el uso de las características geométricas de los rostros [4], tales como la posición de los ojos, la nariz, la boca, la forma del rostro que puede aproximarse a una elipse [5], etcétera. Con ésta información se pretende localizar un rostro en una imagen. Por otro lado, los métodos basados en movimiento se han utilizado en la segmentación de las partes del cuerpo humano, así como la reconstrucción 3D de escenas a partir de proyecciones 2D de una secuencia de imágenes [3]. En éste tipo de técnicas se pretende usar la información del movimiento de los ojos y el movimiento de la boca para la localización de rostros [6].

Algunos autores [7] han propuesto el uso de histogramas de la orientación de los bordes de una imagen. Los histogramas de orientación son altamente invariantes a cambios globales de iluminación y son capaces de capturar las propiedades geométricas de un rostro; sin embargo, ésta técnica no es invariante a escala.

También se ha propuesto el uso de redes neuronales [8]. En la detección de rostros es utilizado el perceptron multicapa y la convolución de redes neuronales [3]. Según los resultados reportados en dichas investigaciones, la extracción de características toma un tiempo aproximado de 0.5 segundos, pero el entrenamiento de la red toma un tiempo de 4 horas, lo que hace a las redes neuronales una opción sólo cuando la cantidad de datos a entrenar no es demasiado grande.

Por otro lado, en [3] se propone el uso de grafos. Los vértices de los grafos son los puntos característicos obtenidos usando el wavelet Gabor o el operador morfológico multi-escala de dilatación-erosión, mientras que el vector de distancia geométrica de las características conforman las aristas del grafo. Este enfoque no es ideal para aplicaciones de tiempo real, ya que, según los resultados de [3] el empatado de los grafos es lento.

Algunas otras técnicas se basan en transformaciones matemáticas [3], tales como la transformada de Fourier, la transformada Hadamard, la transformada Karhunen-Loeve, la transformada de Foley-Sammon, entre otras, ya que pueden ser utilizadas para la extracción de características de una imagen. Otra aproximación son los métodos de transformación morfológica presentados en [9].

Sin embargo, uno de los métodos de detección que ha sobresalido por su rapidez y buenos resultados es el propuesto por Viola y Jones [10]. Éste método constan de 3 partes: obtención de características tipo Haar [11], utilizando una técnica a la que los autores llaman imagen integral; obtención de un clasificador utilizando el método de aprendizaje AdaBoost [12]; y una optimización al clasificador haciendo el proceso de clasificación en cascada.

Éste algoritmo fue el primer sistema de detección facial [13] en tiempo real y es uno de los más ampliamente conocidos. Es por lo anterior que éste será el algoritmo a utilizar en el presente trabajo. La explicación a detalle del algoritmo se presenta en el capítulo 3.

2.2. Reconocimiento de rostros en 2D

Una de las partes principales de la detección de rostros, es la extracción de características. El uso de eigenfaces es uno de los métodos existentes propuesto por Sirovich y Kirby [14] y utilizado para el reconocimiento facial por Turk y Pentland [15]. El término eigenfaces es el nombre dado a un conjunto de eigenvectores cuando son utilizados en reconocimiento facial.

En [14] Sirovich y Kirby demostraron que el Análisis de Componentes Principales podía ser utilizado en un conjunto de imágenes de rostros para obtener un conjunto de características base. Dichas características, a las que llamaron eigenpictures, podían ser combinadas para reconstruir las imágenes del conjunto original de entrenamiento. El método funciona de la siguiente forma.

1. Dado un conjunto de imágenes de entrenamiento θ con M imágenes, todas del mismo tamaño $n \times m$, cada una de las imágenes del conjunto es representada como un vector columna de tamaño $n \times m$ y almacenado en una matriz T en donde cada columna corresponderá a cada una de las imágenes de entrenamiento.

2. Obtener la imagen promedio Υ y restarla a cada una de las imágenes en T.

$$\Upsilon = \frac{1}{M} \sum_{n=1}^{M} T_n$$

$$T = T - \Upsilon$$

3. Obtener la matriz de covarianza S de cada imagen en T

$$S_i = T_i T_i^T$$

   Donde i es cada una de las imágenes (columnas) de T. Si se utilizan imágenes 256×256 pixeles, entonces su vector tendrá un tamaño de 65536 elementos y su matriz de covarianza tendrá un dimensión de 65536×65536.

4. Obtener los eigenvectores y eigenvalores de cada S

$$Sv_i = \lambda_i v_i = TT^T v_i$$

   Donde λ_i es el eigenvalor, y v_i es el eigenvector. Cada uno de los eigenvectores tendrá un tamaño $n \times m$ igual al tamaño de las imágenes originales, es decir, si se tienen imágenes de entrenamiento de 256×256 pixeles, cada eigenvector tendrá un tamaño de 65536. A estos eigenvectores se les conoce con el nombre de eigenfaces.

5. Elegir los componentes principales. Para esto se ordenan de manera descendente los eigenvalores junto con sus respectivos eigenvectores. El número k de componentes principales a elegir se determina de manera arbitraria estableciendo un umbral ϵ tal que

$$\frac{(\lambda_1 + \lambda_2 + \ldots + \lambda_k)}{(\lambda_1 + \lambda_2 + \ldots + \lambda_n)} > \epsilon$$

   Donde n es el número total de eigenvalores que es igual al número total M de imágenes de entrenamiento.

Como se puede observar, la obtención de los eigenvectores y eigenvalores tienen un costo computacional alto debido al tamaño de la matriz de covarianza. Para solucionar dicho problema, Turk y Pentland [15] propusieron una optimización al cálculo de los eigenvectores. Lo que ellos proponen es usar T^T en lugar de T para el cálculo de la matriz de covarianzas, es decir, utilizar los vectores fila de T^T, de un tamaño M, en lugar de los vectores columna de T, de un tamaño $n \times m$. Como la cantidad de imágenes de entrenamiento es menor al número de pixeles de cada imagen, las matrices de covarianza serán de un menor tamaño. Por ejemplo, suponiendo un conjunto de 300 imágenes, cada una de 100 pixeles, la matriz de covarianza $T_i^T T_i$ tendría un tamaño de 300×300, mientras que la matriz de covarianza $T_i T_i^T$ tendría un tamaño de 10000×10000.

La operación anterior es posible debido a lo siguiente.

Sea T la matriz de imágenes de entrenamiento, con cada imagen como vector columna y con la imagen promedio α substraída. Su matriz de covarianza está definida como

$$S = TT^T$$

El eigenvector y eigenvalor de S están dados por

$$Sv_i = TT^T v_i = \lambda_i v_i$$

Pero como se ha visto, la matriz TT^T puede ser muy grande. Si se considera el eigenvector u_i de S tal que

$$T^T T u_i = \lambda_i u_i$$

Pre-multiplicando ambos lados por T se obtiene

$$TT^T T u_i = \lambda_i T u_i$$

Haciendo $v_i = Tu_i$ entonces se tiene que $v_i = Tu_i$ es un eigenvector de S. Siguiendo éste análisis fue como en [15] se llegó a está reducción.

Éste método fue utilizado para la fase de reconocimiento de rostros en 2D. Los detalles del algoritmo se presentan en el capítulo 3.

2.3. Detección de rostros en 3D

Los primeros trabajos hechos en esta área datan de principios de los años 90's, época en la cual se realizaron muy pocos trabajos, siendo que en años recientes la cantidad de trabajo en esta área de investigación se ha incrementado considerablemente.

Cabe destacar que los primeros trabajos reportaban resultados de hasta el 100 % de exactitud en el reconocimiento de rostros, debido a que, en un principio, los conjuntos de datos de prueba eran relativamente pequeños (no más de 100 rostros de personas distintas) y no se empleaban imágenes que tuvieran variaciones en la posición y en las expresiones faciales. No fue hasta 2004 cuando surgió el ya extinto Face Recognition Grand Challenge[1], que se empezaron a utilizar bases de datos de pruebas más grandes y con más variaciones en expresiones faciales.

Uno de los primero trabajos fue el de Cartoux et al. [16], en el cual proponen encontrar un plano de simetría bilateral a través del rostro plasmado en una imagen de profundidad. Dicho plano era utilizado para hacer la normalización de la posición del rostro. Para el reconocimiento, propusieron un método de empatado del perfil del plano de simetría y la superficie de la cara, obteniendo resultados del 100 % de exactitud en el reconocimiento.

Un año después, Lee y Milios [17] propusieron el uso de imágenes gaussianas extendidas (EGI, por sus siglas en inglés), que eran creadas a partir de la segmentación de regiones convexas de una imagen de profundidad. Las EGI describen la forma de un objeto a través de la distribución de las normales sobre la superficie del mismo. Las regiones convexas cambian de forma en menor proporción que otras regiones en respuesta a los cambios en las expresiones faciales, lo que da la habilidad al algoritmo de ser robusto a cambios de expresión en los rostros. Sin embargo, las EGI no son sensibles a los cambios de escala, por lo que dos rostros con una forma similar pero de distintos tamaños serían indistinguibles.

Para 1992, Nagamine et al. [18], propusieron el uso de puntos característicos para estandarizar la posición de los rostros y utilizar un empatado de curvas a través de la información del rostro ya estandarizado. Los costos computacionales resultaban ser muy altos para la época.

Para 2004, Xu y su equipo [19], propusieron un método en donde la nube de puntos obtenida mediante un sensor de profundidad, era convertida a una malla regular. De esta malla se obtiene la posición de la punta de la nariz, y a partir de dicha posición, se determina la posición de la boca y de ambos ojos para generar un vector de características de estas regiones. Para reducir las dimensiones del vector de características, se emplea PCA, y el empatado de las características se basa en la distancia mínima de los vectores. Las pruebas fueron realizadas sobre dos conjuntos de datos, uno de 120 personas y otro de 30. Los resultados obtenidos fueron de 72 % y 96 % de exactitud para cada base de datos, lo que demuestra las variaciones en la eficiencia de los algoritmos según el tamaño del conjunto de datos que se emplee, medida que no se toma en cuenta

[1] http://www.nist.gov/itl/iad/ig/frgc.cfm

en la mayoría de los trabajos de investigación.

En 2005, Gökberk y su equipo [20], hicieron una comparación de cinco métodos distintos: imágenes Gaussianas, empatado ICP, perfil de alcance, PCA y análisis discriminante lineal (LDA). En sus resultados encontraron que los métodos ICP y LDA tienen los mejores resultados. Además, exploraron métodos para combinar distintos enfoques logrando el 99 % de exactitud en sus pruebas. Lo anterior supuso una innovación muy importante, ya que fueron los primeros en comparar el rendimiento de diferentes algoritmos de reconocimiento en 3D y que además, lograron incrementar el rendimiento general combinando los resultados de varios de estos algoritmos.

Fanelli y su equipo propusieron en 2013 el uso de bosques aleatorios [21] para la detección de rostros en imágenes 3D. Éste método resulta interesante, pues de cierta forma es similar al propuesto en [10] por Viola y Jones.

Los bosques aleatorios son un modelo de aprendizaje de ensamble utilizado para clasificación. Un modelo de aprendizaje de ensamble (conocido en inglés como Ensemble Learning) es un paradigma del aprendizaje automatizado donde múltiples algoritmos de aprendizaje son entrenados para resolver un mismo problema [22]. En contraste con las propuestas ordinarias del aprendizaje automatizado, en los cuales se intenta aprender una hipótesis del conjunto de entrenamiento, los modelos de ensamble intentan construir un conjunto de hipótesis simples y combinarlas para utilizarlas en la clasificación. Cabe mencionar que el algoritmo AdaBoost utilizado en [10] para la detección de rostros en 2D es una variante del algoritmo Boosting, un algoritmo perteneciente al modelo de aprendizaje de ensamble.

Ahora bien, un bosque aleatorio es una colección de árboles de decisión, cada uno de los cuales es entrenado con una muestra tomada aleatoriamente del total de datos disponibles [23]. Eso reduce el sobre-entrenamiento que puede presentarse si se utilizara un sólo árbol de decisión entrenado con el total de datos de entrenamiento disponible.

Los árboles de decisión son una colección de nodos y aristas organizados en una estructura jerárquica sin bucles y que son utilizados para tomar decisiones [23].

Los árboles de decisión utilizados son árboles binarios, con exactamente dos nodos hijos por cada nodo padre. Los árboles de decisión funcionan partiendo del nodo raíz, evaluando un criterio dado y decidiendo a que nodo continuar, izquierdo o derecho, hasta llegar a un nodo terminal. Este tipo de árboles pueden ser interpretados como una técnica para dividir un problema complejo en una jerarquía de problemas más simples [23].

El sobre-ajuste o sobre-entrenamiento (también conocido como overfitting en inglés) se presentan cuando un algoritmo de aprendizaje es entrenado en exceso [22]. Un algoritmo de aprendizaje normalmente es entrenado con un conjunto de casos con el fin de poder predecir, a partir de lo aprendido, el resultado en un caso distinto a los entrenados, generalizando la solución para resolver problemas en situaciones diferentes a las previamente conocidas. Por el contrario, si el algoritmo se entrena con una cantidad muy grande de datos, éste puede quedar ajustado a características muy específicas a los casos de entrenamiento por lo que podría perder generalidad produciendo una mayor cantidad de errores al momento de clasificar.

La extracción de características propuesta en [21] también es muy similar a la vista en [10]. Se emplean características muy similares a las tipo Haar y se calculan haciendo uso de imágenes integrales.

Éste método resulta bastante interesante, pues podría verse como una extensión del algoritmo [10] para imágenes 3D. Es por lo anterior que se eligió este algoritmo para esta fase. Los detalles del algoritmo se presentan en el capítulo 4.

2.4. Reconocimiento de rostros en 3D

Acherman et al. [24], extendió el uso de Eigenfaces para ser utilizado con imágenes de profundidad. En sus resultados, reportaron un 100 % de exactitud en el reconocimiento empleando la adaptación que hicieron de los métodos 2D mencionados.

En un trabajo de Beumier y Acheroy [25] que data de 2001, se propuso la combinación de dos métodos, uno en 3D y otro en 2D. Básicamente se propuso el uso de una suma ponderada de las medidas de similitud obtenidas en 3D y en 2D de un perfil central y un perfil lateral de un rostro. Como resultado de la investigación de Beumier y Acheroy, se encontró que la combinación de métodos aplicados a imágenes 2D y 3D proporciona mejores resultados que los mismos métodos aplicados individualmente.

Otro trabajo con resultados similares fue presentado en 2003 por Tsalakanidou y su equipo [26]. En su trabajo combinaron imágenes de profundidad (3D) con imágenes de color (2D). En sus pruebas, utilizaron un algoritmo de reconocimiento basado en PCA, y probaron la eficiencia del algoritmo utilizando individualmente las imágenes de profundidad y las imágenes a color, así como combinándolas. El equipo de Tsalakanidou logró un 99 % de eficiencia de su algoritmo combinando los resultados en ambas imágenes, y se encontró que la eficiencia del algoritmo era mayor con los resultados combinados que con las pruebas individuales.

Tan sólo un año después, Papatheodorou y Rueckert [27], propusieron utilizar una generalización de ICP basada en la distancia entre puntos en un espacio de 4 dimensiones (x, y, z, intensidad). Con dicha propuesta se logró integrar tanto información de la forma de un rostro como la textura del mismo en una sola etapa, cosa que no ocurría en los demás métodos propuestos con anterioridad, pues siempre se necesitaba de una segunda etapa de decisión sobre el tipo de imagen a utilizar, para la posterior combinación de resultados. La desventaja de su propuesta radica en el hecho de que no es robusto a rotaciones y a cambios en la expresión facial.

Finalmente, el método que se eligió utilizar es uno basado en eigenfaces. En lugar de utilizar imágenes 2D, se propone el uso de imágenes de profundidad, tal como se hizo en [24]. Los detalles se presentan en el capítulo 4.

Capítulo 3

Detección y Reconocimiento en 2D

A continuación se presentará el proceso de detección y reconocimiento de rostros utilizando imágenes RGB. Éste proceso consiste básicamente de 4 pasos que se ilustran en la figura 3.1 y se listan a continuación:

- Detección: Dada una imagen, localizar la posición de uno o varios rostros dentro de la misma.
- Preprocesamiento: Dada la imagen de un rostro, ajustar los niveles de brillo, posición y escala para obtener una imagen de un rostro con iluminación uniforme y a una escala fija.
- Entrenamiento: Dadas varias imágenes preprocesadas de rostros de distintas personas, etiquetarlas y almacenarlas en una base de rostros conocidos.
- Reconocimiento: Dada una imagen de un rostro, compararla con las imágenes/-características almacenadas en la base de rostros conocidos para determinar con cual tiene un mayor parecido.

Los algoritmos de detección de rostros sobre imágenes RGB actuales tienen un nivel de confiabilidad bastante alto [28], utilizando imágenes obtenidas en ambientes no controlados. Sin embargo, los algoritmos de reconocimiento facial tienen un menor grado de confiabilidad tratándose de este tipo de imágenes. Por ejemplo, en la literatura se pueden encontrar varios ejemplos [28] de algoritmos cuyos resultados reportados se encuentran por arriba del 90 % de precisión, sin embargo, el entrenamiento y las pruebas se realizan en ambientes controlados con condiciones muy similares. En estos casos, los algoritmos de reconocimiento tienen un alto grado de precisión. El problema de los algoritmos existentes es la alta sensibilidad a los cambios en el rostro, como son los cambios de iluminación, las expresiones faciales, cambios de posición, etcétera, que afectan considerablemente los resultados. Este tipo de cambios es poco común en ambientes controlados, como son las pruebas de laboratorio hechas en la literatura existente, pero en ambientes no controlados son muy comunes y un problema complejo de resolver. Por lo tanto es muy importante tener un buen conjunto de datos como

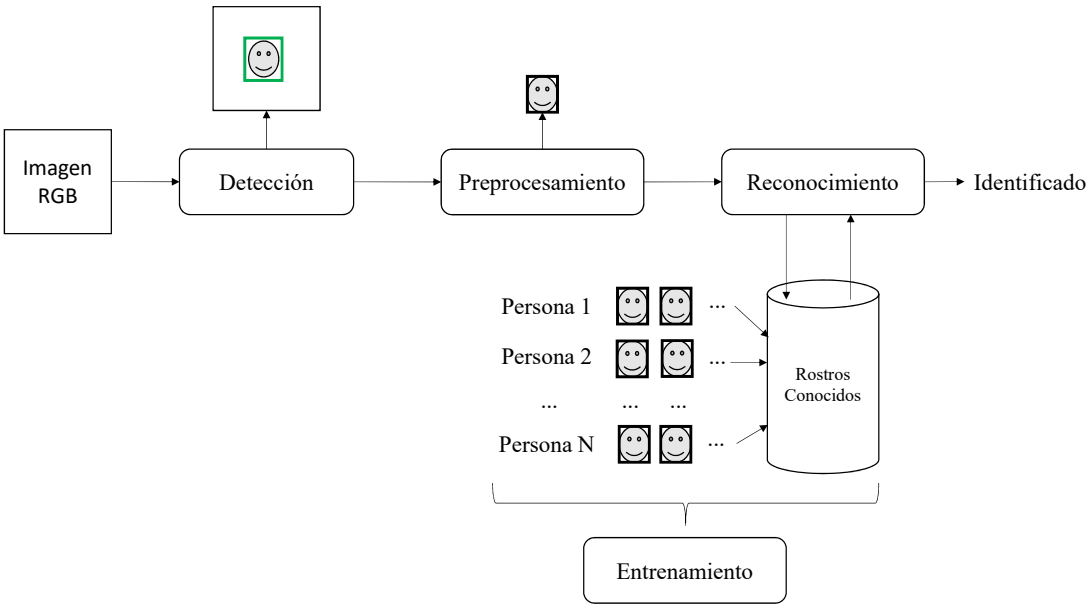


Figura 3.1: Proceso de detección y reconocimiento de rostros en 2D.

entrenamiento, en el cual se tengan imágenes entrenadas que no presenten los problemas comunes ya mencionados.

A continuación se explica con mayor detalle el funcionamiento de cada una de los pasos.

3.1. Detección del Rostro

Existe una gran variedad de algoritmos utilizados en la detección de rostros, pero el algoritmo desarrollado por Paul Viola y Michael Jones en 2001 [10] marcó una diferencia importante. El desarrollo de este algoritmo permitió la creación de un detector de rostros (y de otros objetos en general) rápido (en tiempo real para sistemas de cómputo convencionales) y de alta precisión.

Dicho algoritmo fue utilizado para la detección de rostros presentado en esta sección. A continuación se explica brevemente el funcionamiento.

El algoritmo propuesto consta básicamente de tres partes:

- Un extractor de características basado en características tipo Haar, que utiliza imágenes integrales para agilizar el cálculo de las características.
- Un clasificador basado en el algoritmo de aprendizaje AdaBoost.
- Una búsqueda en cascada para optimizar la localización de rostros en una imagen y descartar rápidamente el fondo sobrante.

3.1.1. Extracción de características

Para la obtención de características, se hace uso de características tipo Haar como las mostradas en la figura 3.2, que fueron utilizadas por primera vez por Papageorgiou [29]. Como se observa en la figura 3.2, se utilizan 3 tipos de características; de dos rectángulos, tres rectángulos y cuatro rectángulos.

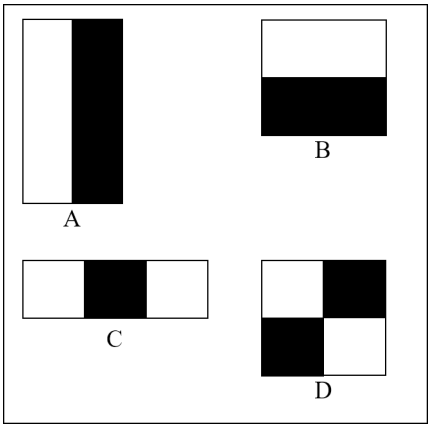


Figura 3.2: Características tipo Haar. A y B) Características de dos rectángulos. C) Característica de tres rectángulos. D) Características de cuatro rectángulos.

Se pueden observar dos zonas, una de color blanco y la otra de color negro. Estos rectángulos se sobreponen a distintas escalas sobre distintas regiones de la imagen a examinar, para obtener una característica que represente cada región. Para esto, se suman los valores de los pixeles que caen dentro de la zona blanca y se restan a la suma de los valores de los pixeles que caen dentro de la zona negra. El valor obtenido es la característica que representa la zona.

Como la aplicación de las características se debe realizar a distintas escalas y con las cuatro formas, en [10] se propone el uso de una imagen intermediaria conocida como imagen integral. Esta imagen contiene en el punto (x, y) la suma de los valores de los pixeles hacia arriba y a la izquierda del punto incluyéndolo.

$$ii(x, y) = \sum_{x' \leq x, y' \leq y} i(x', y') \tag{3.1}$$

Donde $ii(x, y)$ es la imagen integral e $i(x, y)$ la imagen original.

Usando el siguiente par de recurrencias,

$$s(x, y) = s(x, y - 1) + i(x, y) \tag{3.2}$$

$$ii(x, y) = ii(x - 1, y) + s(x, y) \tag{3.3}$$

Donde $s(x, y)$ es la suma acumulada de las filas. Con $s(x, -1) = 0$ y $ii(-1, y) = 0$, se puede obtener toda la imagen integral en una sola pasada.

De la imagen integral es posible obtener la suma de una porción rectangular de la misma utilizando el valor de 4 puntos tal como se muestra en la figura 3.3.

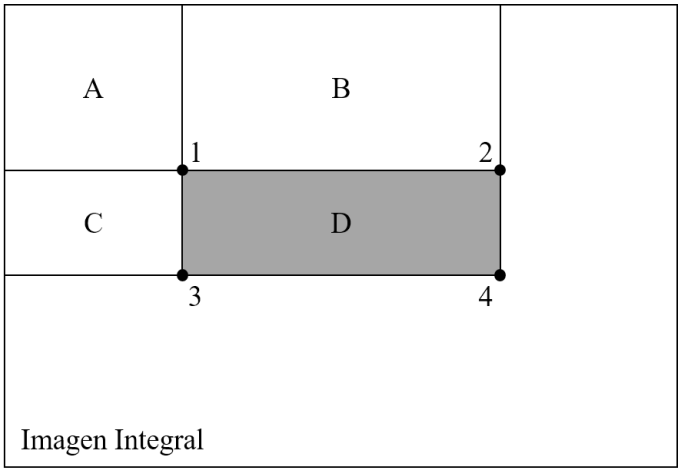


Figura 3.3: Uso de una imagen integral para obtener la suma de los valores dentro de una zona específica. La suma de los valores dentro de la región sombreada D es igual al valor de los puntos 4+1-(2+3).

Por ejemplo, si se quiere obtener la suma de los pixeles dentro de la región A de la figura 3.3, basta con tomar el valor del punto 1. Para obtener el valor de las regiones $A + B$ se toma el valor del punto 2. Para obtener la suma de los pixeles dentro de la región D basta con sumar el valor del punto 4 más el punto 1 y restar al resultado la suma del punto 2 más el punto 3.

La zona D de la figura 3.3 representa uno de los rectángulos que forman cada una de las características tipo Haar. Entonces, para las características de dos rectángulos se necesitan 6 puntos, para las de tres rectángulos se necesitan 8 puntos y para la de cuatro rectángulos se necesitan 9 puntos. Como se puede apreciar, el tiempo de cálculo de las características es constante.

Como la posición y el tamaño de un rostro en una imagen no es un parámetro fijo y conocido, la localización de un rostro en una imagen se debe realizar mediante una ventana de búsqueda que divida la imagen en sub-imágenes de menor tamaño, llamadas ventanas, en las cuales se aplicarán las características tipo Haar a distintos tamaños y posiciones, tal como se muestra en la figura 3.4. Además, las ventanas deben ser de distintos tamaños y cubrir todas las regiones de la imagen.

En [10] se propone el uso de alrededor de 180000 características, en cada una de las ventanas que se utilicen, con un tamaño base de la ventana de 24×24 pixeles. Aunque el cálculo de cada característica se puede realizar de manera muy eficiente debido al uso de imágenes integrales, el cálculo de todas las características resulta prohibitivo. Para resolver este problema, en [10] se propone el uso de sólo una parte de todas las características, tal que combinadas puedan generar un buen clasificador. Para lograr lo

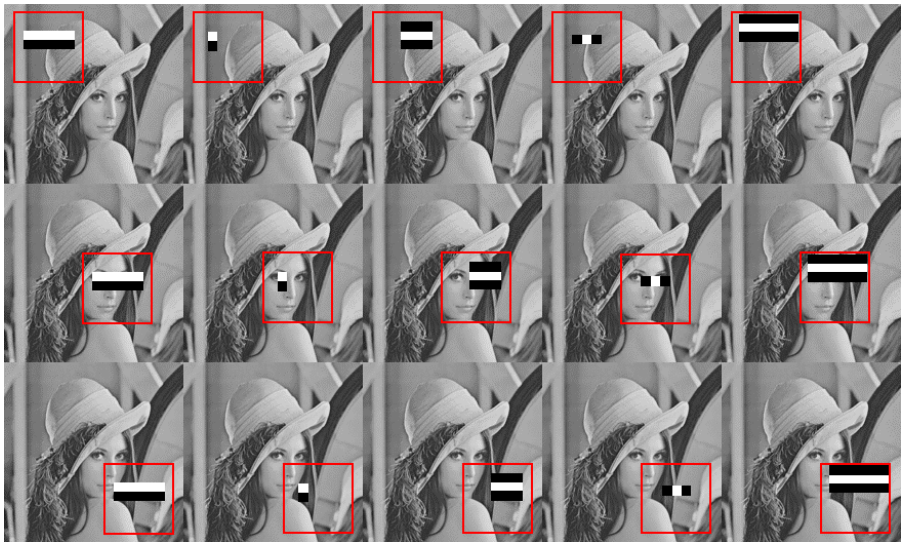


Figura 3.4: Algunas características tipo Haar aplicadas a una imagen de Lena en tres ventanas diferentes.

anterior, se deben elegir las características que por su posición y tamaño sirvan para construir un buen clasificador.

3.1.2. Clasificación

Para la selección del conjunto de características a utilizar en [10] se propone el uso del algoritmo AdaBoost (Adaptative Boosting) presentado por primera vez en [12]. AdaBoost básicamente es un algoritmo de aprendizaje simple (también llamado débil) cuya capacidad de clasificación es estimulada a través de distintas etapas, obteniendo un mejor clasificador en cada una de ellas. Al final se obtiene un clasificador óptimo.

Para elegir las características a utilizar, se recolecta un conjunto de muestras de imágenes con rostros en ellas, además de otro conjunto de imágenes sin rostros, a las que llamaremos muestras positivas y negativas respectivamente. Estas imágenes deben ser etiquetadas con la clase a la que corresponden, estando en la clase 0 las muestras positivas, y en la clase 1 las muestras negativas. En la figura 3.5 se muestra un ejemplo del conjunto de entrenamiento utilizado en [13].

En la primer etapa, se le asigna un mismo peso a cada característica y se define un umbral de clasificación arbitrario. Se evalúa el valor de cada característica comparándolo contra el umbral y si el valor de la característica resulta dentro del rango del umbral, se verifica a que clase corresponde la muestra evaluada. Se eligen las características que se encuentren dentro del umbral y que pertenezcan a muestras positivas, y se les asigna un mayor peso a éstas. El proceso se repite n cantidad de veces y al final se toman las características de mayor peso, las cuales serán utilizadas para la clasificación. En los

Figura 3.5: Muestras de imágenes de entrenamiento. (Izquierda) Muestras positivas. (Derecha) Muestras negativas. Imagen tomada de [13].

primeros experimentos reportados en [10] se demostró que un clasificador construido con 200 características lograba un 95 % de precisión en la detección de rostros teniendo un falso positivo por cada 14084 muestras. Este primer clasificador requería 0.7 segundos para procesar una imagen de 384×288 pixeles, pero para la mayoría de los casos en ambientes no controlados no resultó suficientemente preciso. Para mejorar la precisión del clasificador es necesario agregar más características, pero esto implica hacer más operaciones y por lo tanto el costo computacional aumenta.

3.1.3. Búsqueda en cascada

La solución fue utilizar un modelo en cascada como el mostrado en la figura 3.6, el cual consiste en dividir el proceso de detección de rostros en etapas. En cada etapa se utiliza un clasificador distinto, con un grado de precisión mayor (con un mayor número de características) ajustando el umbral de clasificación en cada etapa para disminuir la cantidad de falsos negativos.

En las primeras etapas del clasificador en cascada se aplican clasificadores muy simples a cada ventana, los cuales la rechazan o aceptan rápidamente, debido a que los primeros clasificadores tiene muy pocas características (el primer clasificador tiene una característica, el segundo tiene 10, el tercero y cuarto 25 y el quinto 50 [10]). Si la ventana es aceptada como posible rostro, pasa a una siguiente etapa donde se aplica un clasificador más riguroso (con una cantidad mayor de características). Si la ventana analizada pasa la última etapa entonces se considera como un rostro.

Como se observa en la figura 3.7, a un mayor número de etapas, un menor número de falsos positivos. Para el detector propuesto en [10], se plantea el uso de un clasificador en cascada de 38 etapas con 6061 características utilizadas en total. Entonces, el clasificador es aplicado a cada una de las ventanas utilizadas.

El número de ventanas depende del factor de escala s que se utilice y de la cantidad

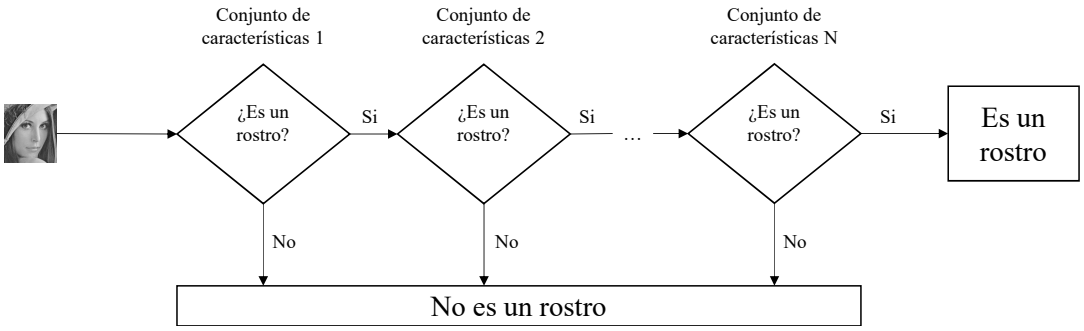


Figura 3.6: Clasificación en Cascada.

de pixeles Δ que cada ventana avance en cada paso. Con base a los resultados de [10], un factor de escala de $s = 1.25$ es suficiente para una buena detección. La cantidad de pixeles Δ que la ventana avanza es directamente proporcional a la escala $[s\Delta]$, siendo buenos candidatos valores $\Delta = 1.0$ y $\Delta = 1.5$.

Como el detector es insensible a los pequeños cambios en escala y traslación, puede ocurrir el caso en el cual varias detecciones se realizan sobre un mismo rostro con ligeras variaciones en la posición de la ventana (figura 3.7). Para eliminar las detecciones excedentes se utiliza el siguiente criterio [13]: para evitar los falsos positivos, se considera una cara detectada cuando se enciman una cierta cantidad de ventanas adyacentes del mismo tamaño conservando la ventana de mayor confiabilidad, es decir, la ventana con mayor número de características válidas. Para eliminar las ventanas encimadas de distintos tamaños, figura 3.7 (d), si el centro de una ventana de menor tamaño se encuentra dentro de una ventana de mayor tamaño, se conserva la de mayor confiabilidad, en otro caso se conservan ambas.

3.2. Preprocesamiento

Debido a que las condiciones de iluminación en las que se capturan las muestras puede variar y a que es una de las variables que más afectan a éste tipo de imágenes, es conveniente tener una fase de preprocesamiento, con la cual se intente corregir lo mejor posible los defectos que puedan tener las imágenes de los rostros obtenidas en la fase de detección.

La forma más sencilla de mejorar la imagen es aplicando una ecualización a su histograma, con lo cual se puede obtener una distribución uniforme de los niveles de grises. Como resultado de la ecualización, se maximiza el contraste de la imagen, lo que podría ayudar en el proceso de reconocimiento. Debido a que las fuentes de iluminación no son constantes, es posible que aplicar una ecualización a toda la imagen sólo beneficie una parte de la misma, tal como se ejemplifica en la figura 3.8.

Un método más eficiente es realizar la ecualización en dos partes independientes,

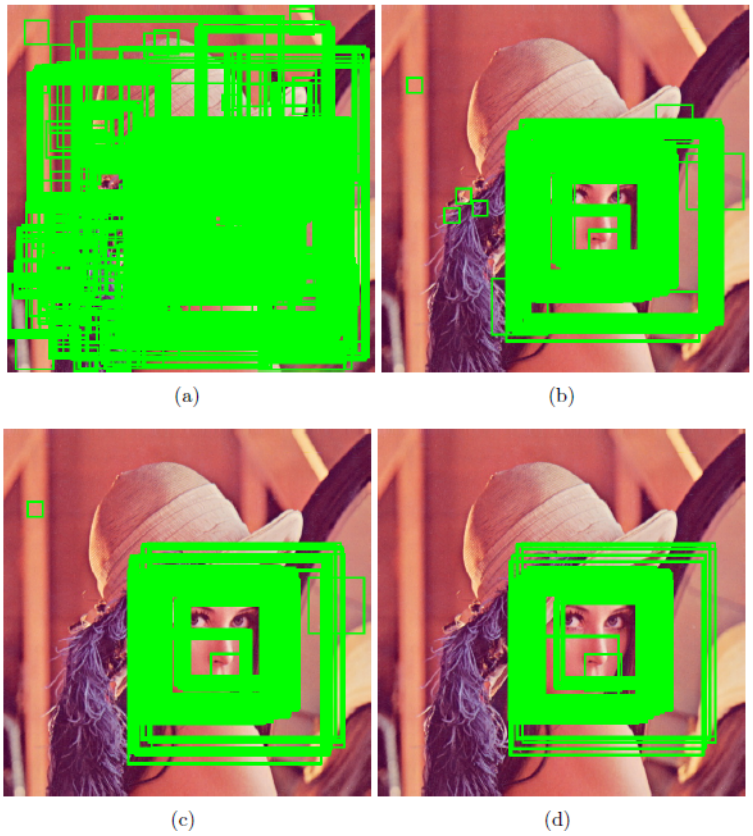


Figura 3.7: Resultados de aplicar un detector de (a) 16 etapas, (b) 21 etapas, (c) 26 etapas y (d) 31 etapas. Imagen tomada de [13]

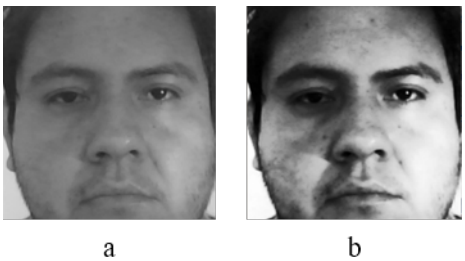


Figura 3.8: Ejemplo de ecualización del histograma. a) Imagen original. b) Imagen ecualizada. Como se observa, la ecualización no beneficia el lado izquierdo del rostro.

es decir, aplicar una ecualización a cada lado de la cara por separado. Para ésto, se proponen los siguientes pasos a seguir.

- Detectar la posición de los ojos.
- Alinear el rostro.
- Ecualizar el histograma por separado.

En las siguientes secciones se explica cada una de estas etapas.

3.2.1. Detección de ojos

El detector propuesto en [10], como se mencionó anteriormente, es un detector que puede ser entrenado para otros objetos, como son los ojos de una persona. Aprovechando que de la etapa anterior se tienen imágenes que contienen exclusivamente rostros, es posible definir áreas de búsqueda de ojos, aplicando los entrenamientos ya disponibles para OpenCV. En el cuadro 3.1 se muestran la confiabilidad, la velocidad y la capacidad de detectar ojos cerrados y con lentes de cada uno de los entrenamientos ya disponibles [30].

Clasificar	Conf.(%)	Vel.(ms)	Cerrados	Lentes
haarcascade_mcs_lefteye.xml	80	18	Si	No
haarcascade_lefteye_2splits.xml	60	7	Si	No
haarcascade_eye.xml	40	5	No	No
haarcascade_eye_tree_eyeglasses.xml	15	10	No	Si

Tabla 3.1: Tabla comparativa de los detectores de ojos disponibles en OpenCV.

Los valores de confiabilidad y velocidad que se muestran en la tabla 3.1 fueron obtenidos a partir de un conjunto de 1000 imágenes a una escala de 320×240 pixeles según lo reportado en [30]. La confiabilidad se refiere al porcentaje de imágenes en las que se detectaron ambos ojos por cada uno de los entrenamientos.

De la tabla 3.1 se puede observar que el primer entrenamiento produce buenos resultados en poco tiempo. Dicho entrenamiento tiene dos versiones, uno para el ojo izquierdo y otro para el derecho. Para reducir el tiempo de búsqueda, se puede reducir el área de búsqueda, definiendo dos zonas dentro de la imagen del rostro, una zona para el ojo izquierdo y otra para el derecho. El tamaño de las zonas es relativo al tamaño de la imagen, ya que deben estar ubicadas en la misma posición sin importar la escala del rostro analizado. Experimentalmente se definió de la siguiente forma. Partiendo de la esquina superior izquierda de la imagen, se define la esquina superior izquierda de la

zona de búsqueda al 10 % del ancho y 20 % del alto de la imagen. La zona de búsqueda tendrá un ancho del 40 % de la imagen y un alto del 35 %. Lo mismo aplica para la zona del ojo derecho tomando como referencia la esquina superior derecha de la imagen. Un ejemplo se puede ver en la figura 3.9.

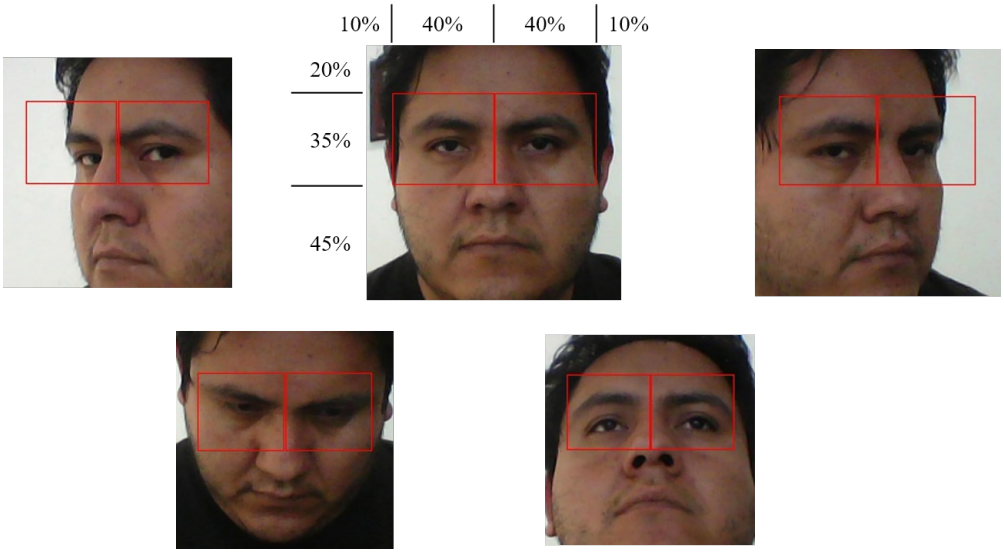


Figura 3.9: Ejemplos de la zona de búsqueda de ojos. Las zonas están indicadas en color rojo.

A cada una de las zonas de búsqueda, como las mostradas en la figura 3.9, se le aplica el detector de ojos mencionado anteriormente. Éste detector lo que hará es encontrar la ubicación de los ojos en la imagen. En la figura 3.10 se muestra un ejemplo del resultado de la aplicación del detector de ojos. Los ojos detectados se indican con un recuadro color azul, mientras que el centro de los ojos se indica con un circulo color rojo.

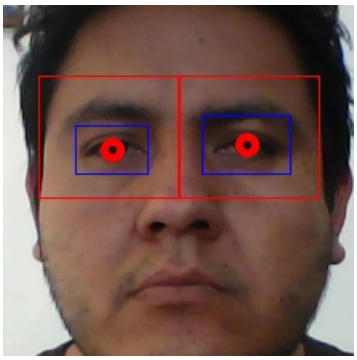


Figura 3.10: Ejemplo de la detección de ojos.

Con la ubicación de los ojos en la imagen, el siguiente paso es alinearlos y tratar de ubicarlos en una misma posición, pues uno de los requisitos del algoritmo de reconocimiento es que al menos los ojos se encuentren ubicados en una misma posición para todas las imágenes de entrenamiento [15], además de que las imágenes deben ser del mismo tamaño.

3.2.2. Alineación

Para asegurar que el algoritmo de reconocimiento facial compare las partes de los rostros correctamente, es necesario alinear el rostro obtenido en la fase anterior, de tal manera que los ojos se encuentren en un área conocida. Además, es requisito que las imágenes de entrenamiento sean del mismo tamaño. Aprovechando que en la fase anterior se obtuvo la posición de ambos ojos en la imagen, se procede a hacer lo siguiente.

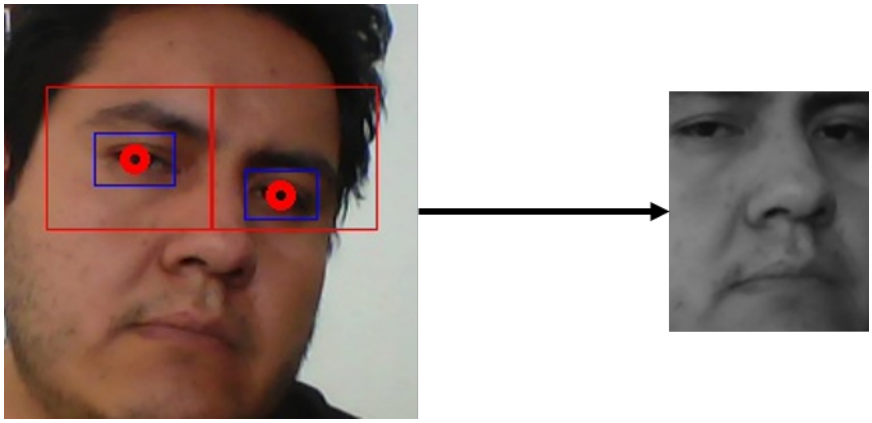


Figura 3.11: Ejemplo del resultado del proceso de alineación de los ojos.

1. Calcular el ángulo entre los ojos.
2. Definir el tamaño que tendrán las imágenes. Para nuestro caso se definió un tamaño arbitrario de 100×120 pixeles.
3. Definir el factor de escala para ajustar la imagen al tamaño deseado.
4. Obtener la matriz de transformación para rotar la imagen de tal forma que los ojos queden alineados horizontalmente y la imagen quede al tamaño deseado.

Realizando los cálculos anteriores, se obtendrá la imagen corregida para, por último, recortarla de tal forma que sólo sea visible el área del rostro. Un ejemplo de este proceso de muestra en la figura 3.11.

3.2.3. Ecualización del histograma

El rostro es un objeto tridimensional complejo, que puede tener muchos cambios de iluminación y variaciones de sombras debido a las distintas fuentes de iluminación. Una de las principales diferencias es la variación de iluminación presente en ambos lados de la cara tal como se ejemplificó en la sección de preprocesamiento de éste capítulo.

Aunque tratar de resolver éste problema para todos los casos posible es muy difícil, se intentará mejorar la iluminación de ambos lados de una cara. Para logra esto se propone lo siguiente.

1. Generar una nueva imagen con la ecualización del histograma de toda la imagen.
2. Generar una nueva imagen con la ecualización del histograma de la mitad izquierda de la imagen.
3. Generar una nueva imagen con la ecualización del histograma de la mitad derecha de la imagen.
4. Crear una nueva imagen mezclando las 3 imágenes anteriores de modo que la parte central corresponda a la parte central de 1 y las partes izquierda y derecha correspondan a 2 y 3 respectivamente.

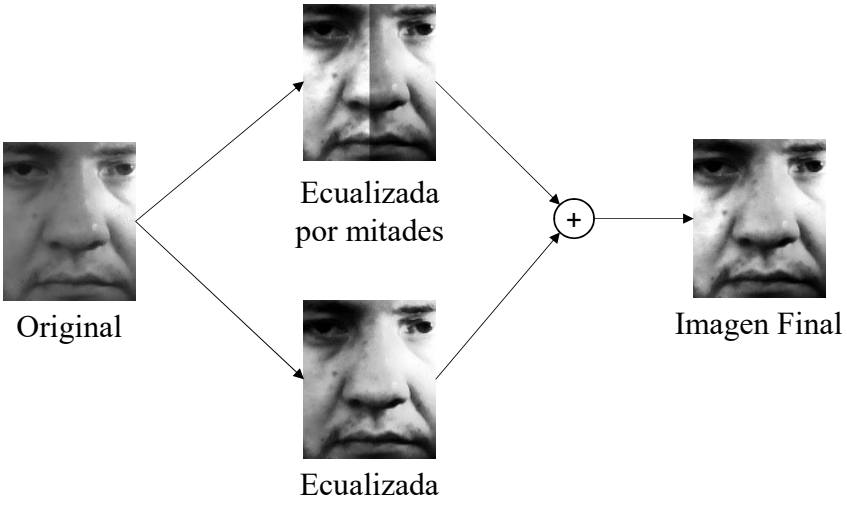


Figura 3.12: Ecualización del histograma por partes. Como se puede apreciar, la imagen final tiene una mejor iluminación que la imagen ecualizada.

Si bien la mejora podría no ser muy significativa en algunos casos, en otros podría resultar en una mejora que beneficiaría al proceso de reconocimiento. En la figura 3.12 se muestra un ejemplo de proceso descrito.

3.3. Entrenamiento

El entrenamiento es la fase del proceso en donde se recolectan varias muestras de distintos rostros de las personas que se desean reconocer. Ha estas imágenes se les conoce como entrenamientos y por cada persona que se desee reconocer, se deben obtener varias imágenes de entrenamiento. Es importante que dichas imágenes sean distintas, pues no tiene sentido entrenar con la misma imagen varias veces. También es importante que se entrenen imágenes con las posiciones del rostro y los gestos que se podrían presentar en la fase de reconocimiento. Por ejemplo, si se espera que los rostros siempre se encuentren mirando hacia enfrente con una expresión seria, entonces es conveniente entrenar rostros mirando hacia enfrente con expresión seria. Si por el contrario, se espera reconocer personas mirando hacia los lados y haciendo gestos, es recomendable entrenar la mayor variedad posible de posiciones y gestos por cada persona.

Para realizar la captura de los rostros se hace uso del detector expuesto al inicio de este capítulo. Cada imagen capturada se etiqueta con números, comenzando por cero, donde los números corresponden a la persona entrenada. Por ejemplo, para la persona 0, se tendrán n imágenes, y de igual manera para la persona 1, 2, etcétera. Adicionalmente se almacena un identificador por cada persona, ya sea su nombre o un seudónimo.

Hay que resaltar que las imágenes deben ser en la mayor cantidad de poses y gestos posibles para cada una de las personas. De igual forma, las imágenes capturadas deben ser del mismo tamaño y la parte de los ojos alineada de tal forma que se encuentre más o menos en la misma posición para todos los rostros.

3.4. Reconocimiento

Para la fase de reconocimiento es necesario obtener descriptores que permitan diferenciar un rostro de otro. Para esto, se hace uso de los eigenfaces[15]. En primer lugar se debe obtener un conjunto de descriptores de las imágenes de entrenamiento. Tal como se mencionó en el capítulo 2, el procedimiento es el siguiente.

1. Dado un conjunto de imágenes de entrenamiento θ con M imágenes, todas del mismo tamaño $n \times m$, cada una de las imágenes del conjunto es representada como un vector columna de tamaño $n \times m$ y almacenado en una matriz T en donde cada columna corresponderá a cada una de las imágenes de entrenamiento. En la figura 3.13 se muestra un ejemplo de un conjunto de imágenes a entrenar.

2. Obtener la imagen promedio Υ y restarla a cada una de las imágenes en T.

$$\Upsilon = \frac{1}{M} \sum_{n=1}^{M} T_n$$

$$T = T - \Upsilon$$

Figura 3.13: Ejemplo de un conjunto de imágenes de entrenamiento.

En la figura 3.14 se muestra la imagen promedio resultante del conjunto entrenamiento de ejemplo de la figura 3.13.

Figura 3.14: Ejemplo de la imagen promedio del conjunto de entrenamiento de la figura 3.13.

3. Obtener la matriz de covarianza S de T

$$S = T^T T$$

Hay que notar que las matrices de covarianza se obtienen en base a la cantidad de imágenes de entrenamiento, y no al tamaño de cada imagen como se explico en el capitulo 2.

4. Obtener los eigenvectores y eigenvalores de cada S

$$Sv_i = \lambda_i v_i$$

Donde λ_i son los eigenvalores, y v_i los eigenvectores. En éste paso se tendrán M eigenvectores con su respectivo eigenvalor. Los eigenvectores pueden ser vistos como imágenes. En la figura 3.15 se muestran los 8 eigenvectores correspondientes al conjunto de entrenamiento de la figura 3.13.

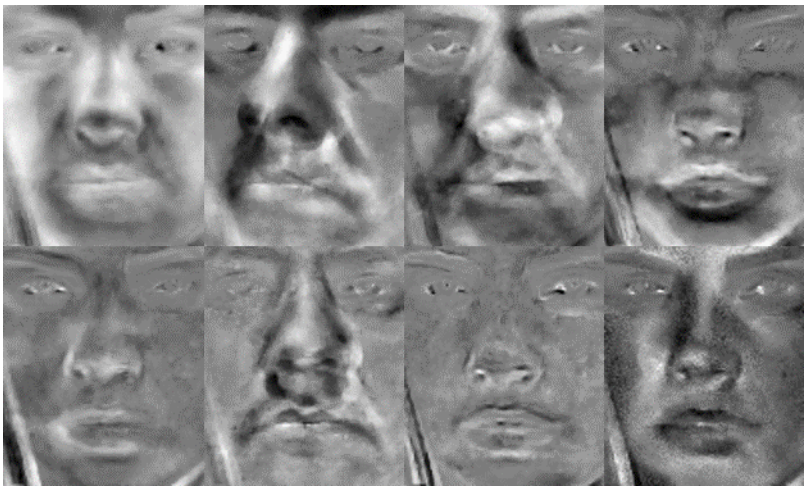


Figura 3.15: Eigenfaces correspondientes al conjunto de entrenamiento de la figura 3.13.

Con la imagen promedio Υ y los eigenfaces v_i obtenidos en en el proceso anterior, ahora se procede a hacer el reconocimiento de la siguiente forma.

1. Con una imagen Γ de entrada, se obtiene la diferencia con la imagen promedio Υ y ésta diferencia se multiplica por la traspuesta del vector v_i de eigenfaces. Ha este proceso se le conoce como proyección al espacio rostro. El resultado será un vector Ω que contendrá la contribución de cada eigenface en la representación de la imagen Γ.

$$\omega_k = v_k^T (\Gamma - \Upsilon) \qquad\qquad \Omega^T = [\omega_1, \omega_2, ..., \omega_M]$$

   Para $k = 1, 2, ..., M$

2. A partir de los valores de la proyección Ω es posible reconstruir una imagen de entrenamiento multiplicando Ω^T por v_x^T y sumando al resultado la imagen promedio Υ.

Si la imagen de entrada fuera una del conjunto de entrenamiento, la reconstrucción sería casi idéntica (figura 3.16) y lo mismo aplica para imágenes muy similares.

Sabiendo lo anterior, el proceso de reconocimiento se hará obteniendo el error entre cada imagen de entrenamiento y la reconstrucción de la imagen de prueba. Para la obtención del error se calculará la norma L2 de la diferencia de las imágenes, tal como

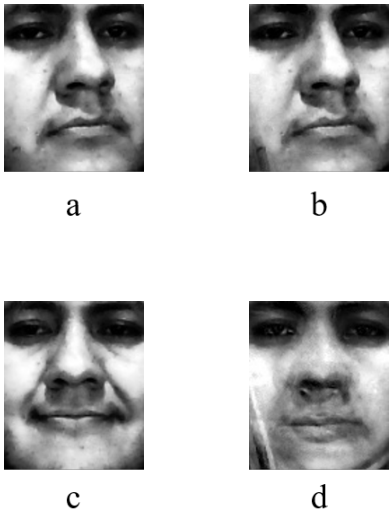


Figura 3.16: Ejemplo de la reconstrucción de una imagen. a) es una imagen del entrenamiento mostrado en la figura 3.13 que es proyectada al espacio rostro. b) es la imagen reconstruida a partir de la proyección de a). c) es una imagen que no fue entrenada. d) es la reconstrucción a partir de la proyección de c).

se explica en el apéndice C. Se tomará la comparación con menor error como el posible rostro identificado y se establecerá un umbral de error máximo de tal forma que si la comparación con el menor error está por arriba del umbral se concluirá como un desconocido.

Capítulo 4

Detección y Reconocimiento en 3D

En éste capítulo se explicará el procedimiento y el algoritmo empleado para la detección y el reconocimiento de rostros, utilizando como datos de entrada una nube de puntos.

El proceso de reconocimiento facial en 3D es muy similar al presentado en el capítulo anterior, correspondiente al proceso de reconocimiento facial en 2D. En la figura 4.1 se ilustran las fases del proceso, las cuales se describen brevemente a continuación.

- Adquisición. Se adquieren los datos con los que se trabajará, en este caso, se refiere a la adquisición de una nube de puntos utilizando un sensor de profundidad.
- Detección. Con la nube de puntos de la etapa anterior, se procede a la detección de posibles rostros humanos dentro de la nube de puntos.
- Preprocesamiento. Con los rostros localizados, se procede a aplicar las transformaciones necesarias para separar y homogeneizar los datos, preparándolos para la siguiente etapa.
- Entrenamiento. Almacenar los datos obtenidos para crear una base de conocimientos de rostros conocidos.
- Reconocimiento. Con la información de los rostros obtenida en las etapas anteriores, se compara con la base de conocimientos en busca de coincidencias.

A continuación se describe cada etapa con mayor detalle.

4.1. Adquisición

A diferencia del proceso para detección de rostros con imágenes 2D, en este caso se utilizará una nube de puntos. Las nubes de puntos son un conjunto de vértices en un sistema de coordenadas tridimensional. Estos puntos se pueden obtener a través de un

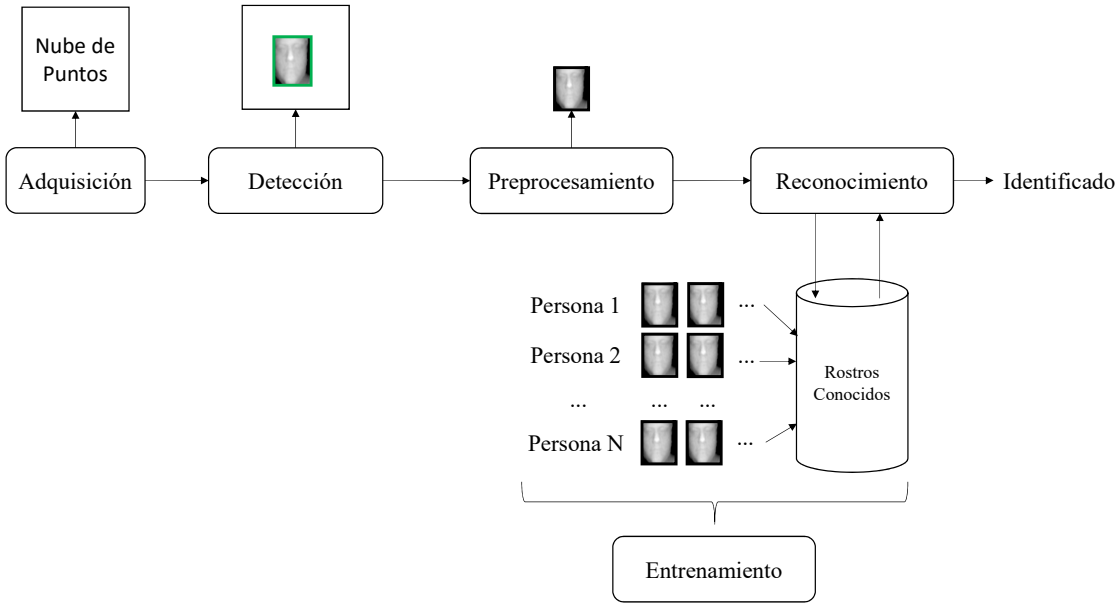


Figura 4.1: Proceso de detección y reconocimiento de rostros en 3D.

Figura 4.2: Ejemplo de una nube de punto obtenida con un kinect.

escáner tridimensional o un sensor de profundidad como el kinect. En la figura 4.2 se muestra un ejemplo de una nube de puntos obtenida con un kinect versión 2.

Para la adquisición de las nubes de puntos utilizadas en éste trabajo se utilizó un sensor Kinect versión 2. Para la adquisición de la información del sensor se utilizó un controlador desarrollado por Migue Ángel Vico [31].

4.2. Detección

Para la detección de rostros en la nube de puntos se utilizó un método llamado bosques aleatorios, como fue propuestos en [21]. En la figura 4.3 se muestra un ejemplo de un bosque aleatorio. A continuación se describe brevemente el método.

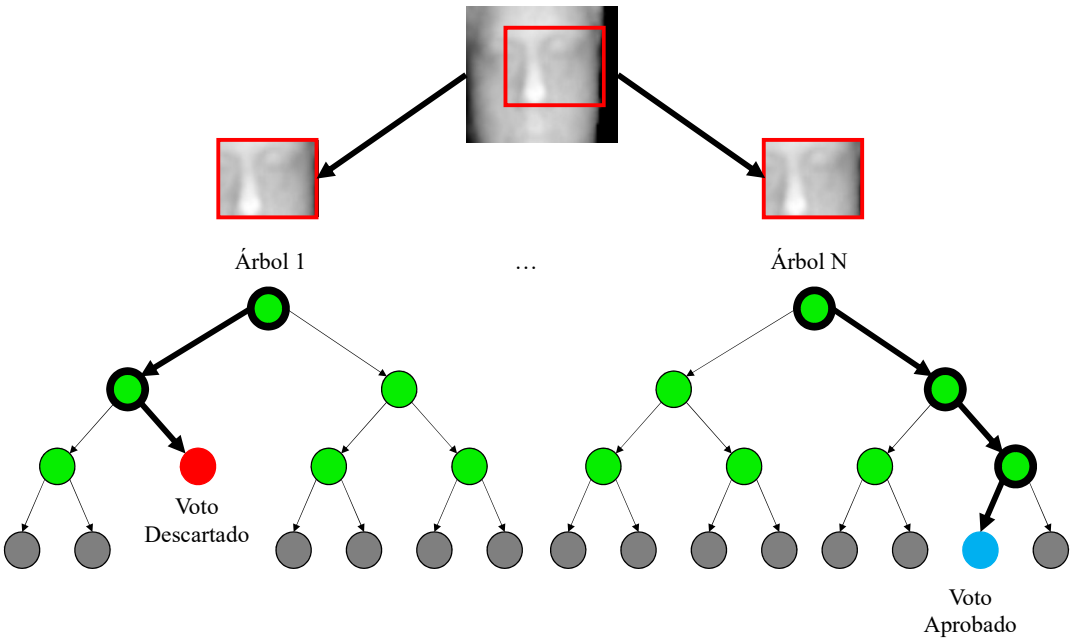


Figura 4.3: Ejemplo de un bosque aleatorio compuesto por un conjunto de N árboles binarios de decisión. Cada árbol de decisión está entrenado con un subconjunto de muestras tomadas aleatoriamente de un conjunto de imágenes de entrenamiento. Dichos árboles son utilizados para determinar si una muestra a evaluar pertenece a un rostro, emitiendo un voto de muestra aprobada o descartada. Una misma muestra es evaluada por los N árboles de decisión que componen al bosque, y al final se obtendrá una cantidad n de votos que ayudarán a determinar si la muestra evaluada es un rostro.

En términos generales, los bosques aleatorios funcionan de la siguiente manera: un árbol de decisión T en un bosque τ es construido a partir de un conjunto de muestras etiquetadas, extraídas aleatoriamente de un conjunto de imágenes de entrenamiento P. Empezando por el nodo raíz, cada árbol es construido recursivamente asignándole una prueba binaria $\phi(I) \rightarrow \{0, 1\}$ a cada uno de los nodos no terminales. En cada prueba se envían cada una de las muestras al nodo izquierdo o derecho siguiente, de ésta forma, las muestras de entrenamiento que llegan a cada nodo son divididas en dos grupos $P_L(\phi^*)$ y $P_R(\phi^*)$.

La mejor prueba ϕ^* es tomada de un conjunto de pruebas generadas aleatoriamente. Todas las muestras que llegan a un nodo son evaluados por cada uno de las pruebas del conjunto y una ganancia de información predefinida $IG(\phi)$ es maximizada como sigue:

$$\phi^* = arg\ max\ IG(\phi) \tag{4.1}$$

$$IG(\phi) = H(P) - \sum_{i \in \{L,R\}} w_i H(P_i(\phi)) \tag{4.2}$$

Donde $w_i = \dfrac{|P_i(\phi)|}{|P|}$ es la razón de muestras enviadas a cada uno de los nodos y $H(P)$ es una medida de clasificación del conjunto de muestras P. Ésta medida de clasificación puede tener diferentes formas dependiendo del uso que se le quiera dar al bosque, como puede ser para regresión, para clasificación o una combinación de ambos. Si el bosque se utiliza como una combinación de regresión y clasificación, es posible obtener, además de un clasificador de rostros, una estimación de la orientación de los rostros detectados.

El proceso continua con el siguiente nodo hacia la izquierda o derecha utilizando sus correspondientes conjuntos de entrenamiento $P_L(\phi^*)$ o $P_R(\phi^*)$ hasta llegar a un nodo terminal o hasta que se alcance una profundidad mínima especifica.

De lo anterior queda claro que se necesita lo siguiente:

- Los datos de entrenamiento P etiquetados
- Definir una prueba binaria ϕ
- Definir una medida de clasificación $H(P)$
- Definir un modelo de distribución que será almacenado en los nodos terminales

A continuación se explica brevemente cada uno de estos puntos.

4.2.1. Datos de entrenamiento

Los datos de entrenamiento a utilizar son imágenes o mapas de profundidad, obtenidas mediante un sensor kinect, para éste caso, etiquetadas con la localización 3D de

la nariz. De las imágenes de profundidad se toman de manera aleatoria muestras de tamaño fijo y se etiquetan con la información de dos vectores; un vector $\theta^1 = \{\theta_x, \theta_y, \theta_z\}$ correspondiente al desplazamiento entre el centro de la región extraída y la punta de la nariz en la imagen de profundidad, y un segundo vector $\theta^2 = \{\theta_{ya}, \theta_{pi}, \theta_{ro}\}$ correspondiente a la orientación de la cabeza en ángulos de Euler. El tamaño de las muestras propuesto en [21] es de 120×80 pixeles.

Para el caso de las partes de la cabeza que no son de interés, como son el cabello y los hombros, las muestras son tomadas aleatoriamente de todas las partes de la cabeza visibles, incluyendo el cabello, el cuello y los hombros. Para diferenciar las muestras, éstas se clasifican en dos clases c_i, donde $c_i = 1$ corresponde a una muestra de la cara y $c_i = 0$ a una muestra de una parte distinta a una cara. Entonces, cada muestra P_i es etiquetada con la clase a la que corresponde.

De lo anterior, el conjunto de imágenes de entrenamiento esta dado por $P = \{P_i = (I_i, c_i, \theta_i)\}$, donde $\theta = (\theta^1, \theta^2)$ e I_i representa las características de la imagen correspondientes a la muestra P_i. Tales características corresponden a los valores de profundidad dentro de la muestra.

4.2.2. Prueba binaria

La prueba binaria empleada en cada nodo de los árboles está definida como:

$$\frac{\sum_{q \in F_1} I(q)}{|F_1|} - \frac{\sum_{q \in F_2} I(q)}{|F_2|} > \tau \tag{4.3}$$

Donde F_1 y F_2 son dos rectángulos asimétricos definidos aleatoriamente dentro de la muestra como los de la figura 4.4, y τ es un umbral de decisión. El tamaño de los rectángulos propuesto en [21] es de 40 pixeles.

La ecuación 4.3 representa una generalización de las características tipo Haar presentadas en el capítulo anterior. Como se puede apreciar, la prueba binaria consiste en calcular el valor promedio de las características dentro de cada rectángulo F_1 y F_2 y obtener la diferencia de los dos valores. Para este caso, las características utilizadas son los valores de profundidad que caen dentro de cada rectángulo. El cálculo de los promedios se realiza de manera eficiente utilizando imágenes integrales.

Durante el proceso de entrenamiento, para cada nodo no terminal y comenzando desde el nodo raíz, se genera un conjunto de pruebas binarias eligiendo aleatoriamente F_1, F_2 y τ. Se elige la prueba que maximiza una función de optimización específica y los datos de entrenamiento son divididos en dos grupos utilizando la prueba binaria elegida, creando dos nuevos nodos en el árbol, uno a la izquierda y otro a la derecha. El proceso continua con los nuevos nodos y termina hasta que se alcanza una profundidad específica o quedan menos de una cierta cantidad de muestras de entrenamiento. En [32] se propone una profundidad máxima de 15 y una cantidad mínima de muestras de 20.

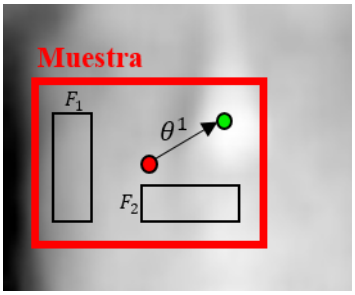


Figura 4.4: Ejemplo de una muestra de entrenamiento. El rectángulo rojo corresponde al área de la muestra. Los rectángulos color negro F_1 y F_2 corresponden a las áreas utilizadas para la prueba binaria, es decir, la diferencia entre el promedio de los valores de profundidad de ambos rectángulos. La flecha corresponde al vector θ^1 que almacena la distancia entre el punto central de la muestra (punto en rojo) y la punta de la nariz (punto en verde). Los rectángulos F_1 y F_2 son generados aleatoriamente durante el entrenamiento.

4.2.3. Clasificación

Los bosques aleatorios propuestos se utilizan para determinar si las muestras probadas pertenecen o no a una cabeza. Para lo anterior se utilizan dos medidas [32], una de clasificación $H(P)_C\left(\{P|t^k\}\right)$ y una de regresión $H(P)_R\left(\{P|t^k\}\right)$ para medir el nivel de incertidumbre de la clasificación.

La medida de clasificación propuesta en [32] es como sigue.

$$H(P)_C(\{P|t^k\}) = \frac{|P_L| \cdot \sum_c p(c|P_L) ln(p(c|P_L)) + |P_R| \cdot \sum_c p(c|P_R) ln(p(c|P_R))}{|P_L| + |P_R|} \quad (4.4)$$

Donde $p(c|P)$ es la proporción de muestras pertenecientes a la clase $c \in \{0, 1\}$ en el conjunto de muestras P.

Respecto a la medida de regresión en [32] se propone la siguiente.

$$H(P)_R(\{P|t^k\}) = \log(|\Sigma^1| + |\Sigma^2|) - \sum_{i=\{L,R\}} w_i \log(|\Sigma_i^1| + |\Sigma_i^2|) \quad (4.5)$$

Donde Σ^1 y Σ^2 son las matrices de covarianza de θ^1 y θ^2, correspondientes al desplazamiento del centro de las muestras a la nariz y la orientación de la cabeza respectivamente.

Las mediciones 4.4 y 4.5 se combinan para obtener una sola medición de la siguiente forma.

$$H(P)_C + \left(1 - e^{-\frac{d}{\lambda}}\right) H(P)_R \quad (4.6)$$

Donde d es la profundidad del nodo y λ especifica la pendiente de cambio. De esta manera, el peso de la medida de regresión se incrementa exponencialmente conforme más se acerque a los nodos terminales, mientras que en los nodos más cercanos al nodo raíz la medida de clasificación tiene un mayor peso. Según los experimentos de [32] se propone el valor $\lambda = 5$.

4.2.4. Nodos terminales

En los nodos terminales se almacenan las probabilidades de cada clase $p(c = k|P)$, que para éste caso se tiene solamente dos clases $c = \{0, 1\}$. Además se almacenan los parámetros $p(\theta^1) = (\theta^1, \overline{\theta^1}, \Sigma^1)$ y $p(\theta^2) = (\theta^2, \overline{\theta^2}, \Sigma^2)$, donde $\overline{\theta^1}$ y $\overline{\theta^2}$ son la media de los vectores de desplazamientos y orientaciones de las muestras que llegaron al mismo nodo terminal, mientras que Σ^1 y Σ^2 son las matrices de covarianza de los mismos. Estos valores son obtenidos en la fase de entrenamiento.

4.2.5. Proceso de detección

Teniendo los árboles entrenados como se menciona anteriormente, se extraen regiones de una imagen de prueba y se envían a todos los árboles del bosque. Cada muestra es evaluada en cada nodo intermedio por la prueba binaria almacenada en éste y es dirigida hacia el fondo de cada árbol. Cuando la muestra llega a un nodo terminal, en dicho nodo se encuentra almacenada la probabilidad de que la muestra sea parte del cuerpo o de la cabeza. Entonces, conociendo a que parte del cuerpo pertenece cada muestra, se toman las muestras cuya probabilidad $p(c = k|P) = 1$. Además se filtran las muestras con una alta varianza descartando las muestras que llegan a nodos terminales con una traza $tr(\Sigma^1) > max_v$, es decir, cuya suma de los elementos de la diagonal principal de la matriz de covarianza del desplazamiento sea mayor a un umbral preestablecido. El valor de max_v es obtenido de manera empírica en [21] con un valor de 400.

Teniendo una muestra que llega a un nodo terminal con una probabilidad $p(c = k|P) = 1$ y con un traza $tr(\Sigma^1) < max_v$, se procede a estimar la posición de la nariz sumando al centro de la muestra el desplazamiento promedio $\overline{\theta^1}$ del nodo terminal. Esta posición estimada es considerada como un voto favorable, por lo que cada muestra que cumpla con las especificaciones mencionadas emitirá un voto con la posible ubicación de la nariz en la cabeza detectada, tal como en el ejemplo de la figura 4.5.

Para eliminar los falsos positivos se agrupan los votos de la siguiente forma; se toman todos los votos con una cierta distancia entre ellos cercana al diámetro promedio de una cabeza. Posteriormente, al conjunto de votos agrupados se les aplica 10 iteraciones del algoritmo mean shift utilizando un kernel esférico de radio igual a un sexto del radio de la cabeza promedio con el objetivo de encontrar el centroide del conjunto de votos.

El algoritmo Mean Shift es muy sencillo y se explica a continuación. Como se puede observar en la figura 4.6 se tiene un conjunto de puntos y se desea encontrar el lugar donde se acumulan la mayor cantidad de éstos. Dada un área de trabajo A con centro A_c, también conocida como kernel o ventana, que en la figura se muestra como un

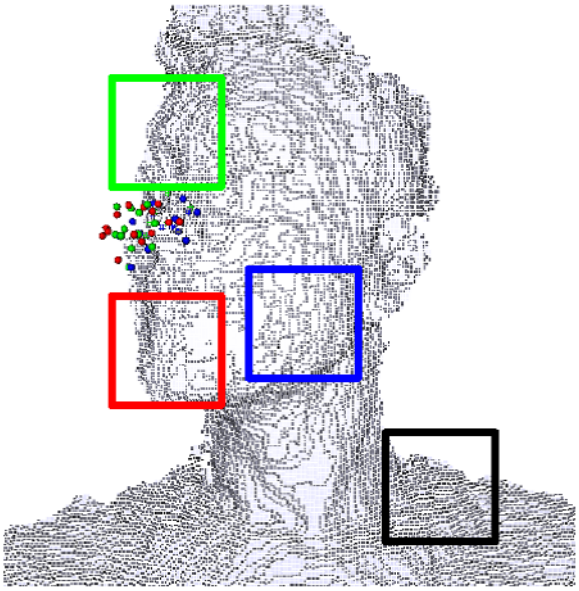


Figura 4.5: Ejemplo de toma de muestras y emisión de votos. Los recuadros de color verde, rojo y azul corresponden a muestras positivas y sus respectivos votos indicados con esferas del mismo color. El recuadro color negro corresponde a una muestra negativa que no corresponde a la cabeza. Imagen tomada de [21].

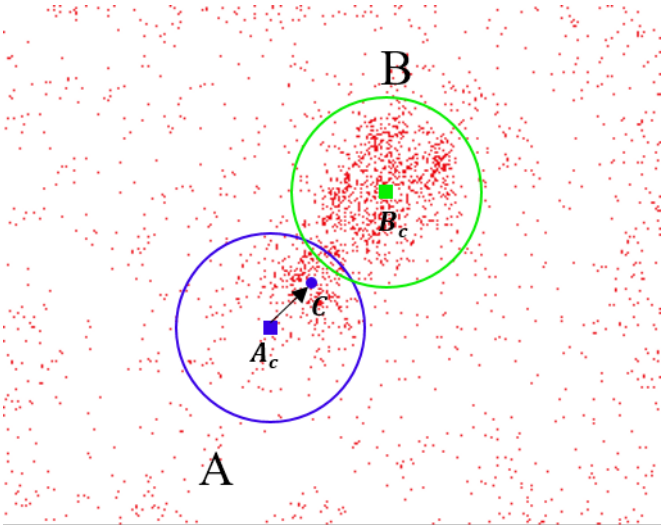


Figura 4.6: Algoritmo Mean Shift.

circulo color azul, se busca el centroide C de los puntos dentro del área A, el cual se indica con un punto color azul en la figura. Si el centroide C encontrado no coincide con el centro A_c de la ventana se procede a mover ésta última de tal forma que A_c coincida con el C. Éste proceso se repite hasta que el centro de la ventana y el centroide de los puntos coincidan o la distancia sea muy pequeña, como es el caso del circulo B color verde cuyo centro B_c está indicado con un recuadro del mismo color y que a su ves es el centroide de los puntos. Al final del proceso se tendrá la máxima concentración de puntos dentro de la ventana.

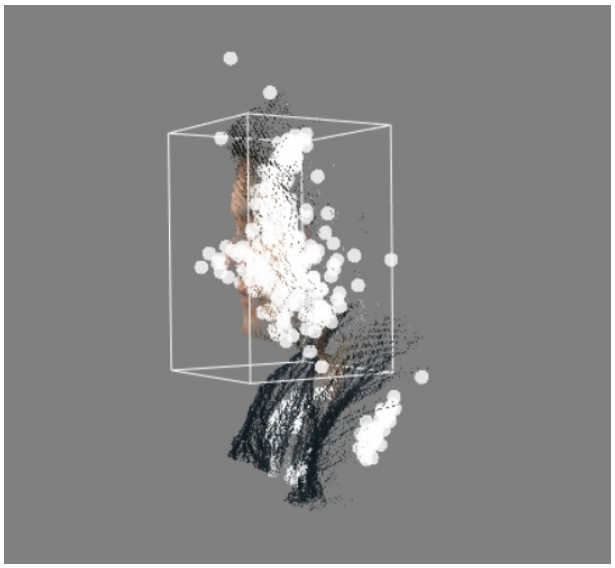


Figura 4.7: Ejemplo de la detección de rostros sobre una nube de puntos. Las esferas blancas indican los votos obtenidos y el rostro detectado se indica con un prisma rectangular color blanco.

Una vez obtenidos los votos finales, se declara el conjunto como una cabeza encontrada si el número de votos es mayor o igual a un umbral definido. El umbral está relacionado a la cantidad de votos que se pueden obtener y la cantidad de muestras. La cantidad de votos es directamente proporcional a la cantidad de árboles que contenga el bosque, ya que cada árbol puede emitir un voto por cada muestra, mientras que la cantidad de muestras es inversamente proporcional al cuadrado de la distancia entre cada muestra, conocido como *stride*. En [21] se propone la siguiente formula para la determinación del umbral.

$$\beta \frac{\textit{número de árboles}}{(\textit{distancia entre muestras})^2} \tag{4.7}$$

Donde $\beta = 300$ según lo propuesto en [21].

Para la obtención de la estimación final de la posición y orientación de la cara, los vectores θ^1 y θ^2 de cada voto final son promediados para obtener la posición final

estimada de la nariz así como la orientación final estimada del rostro. En la figura 4.7 se muestra un ejemplo del sistema de votaciones y la detección del rostro de la figura 4.2.

4.3. Preprocesamiento

Como se explicó en la sección anterior, haciendo uso de un bosque aleatorio es posible obtener la posición de un rostro dentro de una nube de puntos. Una vez que el rostro es ubicado, el siguiente paso es extraerlo de la nube de puntos para eliminar el fondo y las partes del cuerpo, como el cuello y el cabello, que no proporcionan información útil para el reconocimiento facial.

Mediante el proceso de detección de rostros es posible obtener un aproximado de la posición de la nariz y la orientación del rostro. En base a la posición de los rostros, se pueden clasificar en dos grupos: rostros que se encuentran viendo hacia enfrente, y rostros que se encuentran viendo hacia los costados, arriba o abajo. Para el primer caso se cuenta con la información necesaria para realizar el recorte del rostro y posteriormente el reconocimiento, pero para el segundo caso es evidente que hay perdida de información de las partes no visibles del rostro detectado. En éste último caso se tendría que encontrar un técnica que permita la reconstrucción de las partes no visibles del rostro, ya sea a partir de un modelo preestablecido o a partir de la información visible del rostro. Para efectos prácticos se consideraran solamente los rostros del primer caso ideal, dejando el segundo caso como objeto de estudio para un trabajo a futuro.

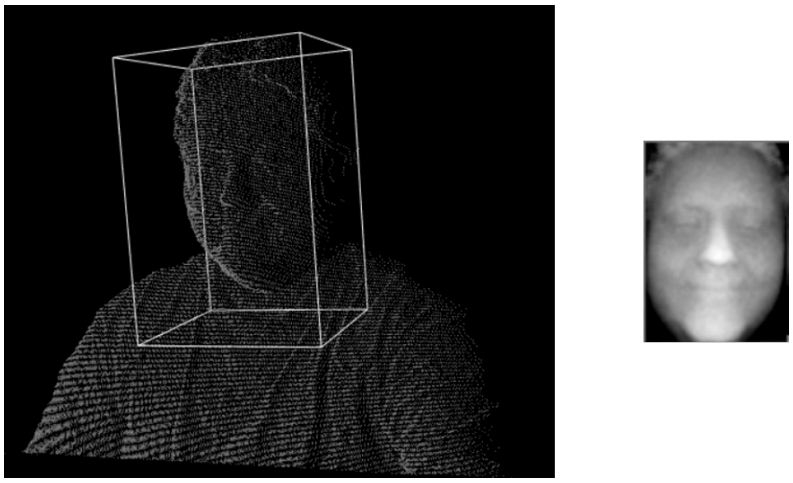


Figura 4.8: Ejemplo de la obtención de la imagen de profundidad a partir de la detección del rostro en la nube de puntos.

Partiendo de la posición de la nariz, se definieron experimentalmente las siguientes medidas: de la nariz hacia arriba 10 cm, de la nariz hacia abajo 10 cm y de la nariz

hacia la izquierda y derecha 7 cm. De la nube de puntos se toman únicamente aquellos que se encuentren dentro del área definida por las medidas anteriores y se genera una matriz $n \times m$ donde se almacena la distancia de cada punto al sensor. El tamaño de la matriz dependerá de la distancia del rostro al sensor. Un ejemplo se muestra en la figura 4.8.

Una de las ventajas de utilizar la información de profundidad es que las imágenes no son afectadas por los cambios de iluminación, como es el caso de la imágenes de intensidad utilizadas en la detección en 2D. En la figura 4.9 se muestra una comparación de ambas técnicas, detección en 3D y en 2D. Como se puede apreciar en la figura, los cambios de iluminación afectan considerablemente la calidad de las imágenes utilizadas en el reconocimiento en 2D, mientras que utilizando las información de profundidad no se tiene tales problemas, lo que lo hace una mejor opción en ambientes con cambios de iluminación constantes.

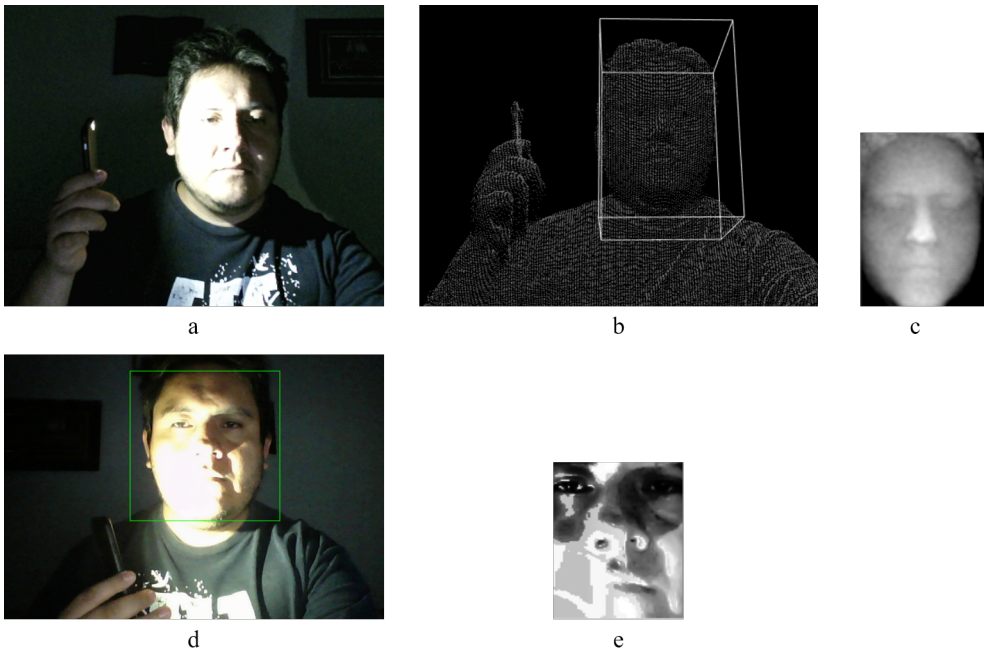


Figura 4.9: Comparación de detección ante cambios de iluminación. a) Imagen de prueba para detección en 3D. b) La nube de puntos correspondiente a la imagen de prueba. c) Imagen obtenida de b) que será utilizada para el reconocimiento. d) Imagen de prueba para detección en 2D. e) Imagen obtenida de d) que será utilizada para el reconocimiento.

Debido a que los rostros pueden estar a distintas distancias del sensor, es necesario normalizar dichas distancias. Para esto se debe trasladar el rostro de tal forma que la nariz, el punto más cercano al sensor en un rostro viendo de frente, se ubique en la distancia cero. Una vez lo anterior, las distancias se normalizan en un rango de 0 a 255

para representarlas en un escala de grises con 256 niveles, y se genera un imagen de profundidad. La imagen se escala a un tamaño de 100×120 pixeles para ser utilizada en la fase de reconocimiento.

4.4. Entrenamiento

Para realizar el entrenamiento se sigue un proceso similar al propuesto en el capítulo anterior. Por cada persona a entrenar es conveniente tomar varias muestras en distintas posiciones y haciendo distintos gestos para abarcar la mayor cantidad de variaciones posibles.

A diferencia del proceso de entrenamiento para el reconocimiento en 2D, en éste caso no es necesario realizar la ecualización del histograma debido a que las variaciones de iluminación no afectan el proceso.

Las personas a entrenar se numeran comenzando por el número cero para la primer persona. Las muestras tomadas por cada persona se numeran y etiquetan con el número de la persona entrenada. Adicionalmente se almacena un identificador, como el nombre de la persona entrenada o un seudónimo.

4.5. Reconocimiento

Para la etapa de reconocimiento se hará uso de la misma técnica explicada en el capítulo 3. Se obtendrán los eigenvectores y eigenvalores de las imágenes de profundidad y se utilizará el mismo proceso de clasificación con el objetivo de comparar los resultados de ambos métodos. Las pruebas y resultados se presentarán en el siguiente capítulo.

Capítulo 5

Pruebas y Resultados

En éste capítulo se presentan las pruebas y resultados de detección y reconocimiento facial utilizando las técnicas descritas en los capítulos anteriores para 2D y 3D.

5.1. Configuración de la prueba

En esta sección se explicará la configuración de software y hardware utilizada para las pruebas, así como la organización de de las mismas.

5.1.1. Hardware

Para la captura de las imágenes RGB y de profundidad se utilizó un sensor kinect versión 2 utilizando la misma escena para el caso 2D y 3D. Se decidió utilizar un kinect 2 debido a la mejor fidelidad del mapa de profundidad que el sensor provee[1], lo que podría mejorar la precisión en la etapa de reconocimiento, además de que la imagen RGB es de mayor resolución respecto a la versión anterior [2].

Las pruebas fueron realizadas en un equipo de computo con las siguientes características:

- Procesador Intel core i7-4500U a 2.99 GHz como velocidad máxima.
- 8 GB de memoria RAM DDR3 a 1600 MHz.

La comunicación entre el kinect y el equipo de computo es a través de un puerto USB 3.0 que es requerido por el sensor.

[1] Según la información proveída por Microsoft: https://dev.windows.com/en-us/kinect/hardware

[2] 1920×1080 y 960×540 pixeles a 30 cuadros por segundo [31] para el kinect 2 en comparación con los 640×480 pixeles a 30 cuadros por segundo del kinect 1

5.1.2. Software

El software utilizado y las versiones se listan a continuación:

- Windows 10 64 bits. El controlador oficial del sensor solamente está disponible para Windows 8.1 y superiores.
- Kinect for Windows SDK 2.0. Incluye el controlador oficial del sensor para Windows.
- OpenNI2 versión 2.3.0. Utilizado para la gestión del sensor.
- OpenNI2 Kinect v2 Driver [31]. Controlador para OpenNI2 que da soporte para utilizar el sensor kinect versión 2. Requiere tener instalado el controlador oficial.
- OpenCV versión 2.4.9. Biblioteca de código abierto para visión computacional y aprendizaje automatizado[1].
- PCL version 1.7.2. Point Cloud Library[2] es una biblioteca para procesamiento de imágenes 2D/3D y nubes de puntos.
- Visual C++ 2010. Utilizado para el desarrollo de las aplicaciones.

Para la detección de rostros se emplearon los siguientes clasificadores:

- *haarcascade_frontalface_alt.xml*. Clasificador disponible para OpenCV para la detección de rostros de frente en imágenes 2D. Creado por Rainer Lienhart [33] con rostros de 20×20 pixeles y 21 etapas.
- *haarcascade_mcs_lefteye.xml*. Clasificador disponible para OpenCV para la detección del ojo izquierdo en un rostro de frente en imágenes 2D. Creador por Modesto Castrillon-Santana [34] con 7000 muestras positiva de 18×12 pixeles y 17 etapas.
- *haarcascade_mcs_righteye.xml*. Clasificador disponible para OpenCV para la detección del ojo derecho en un rostro de frente en imágenes 2D. Creador por Modesto Castrillon-Santana [34] con 7000 muestras positiva de 18×12 pixeles y 13 etapas.
- *forest.txt*. Clasificador disponible para PCL[3] entrenado con la base de datos Biwi Kinect Head Pose Database[4] que contiene 15 mil muestras de 20 personas tal como es propuesto por Gabriele Fanelli en [21] para la detección de rostros en 3D.

[1] Para más detalles véase el sitio http://opencv.org/about.html

[2] Para más información véase el sitio http://pointclouds.org/

[3] https://github.com/PointCloudLibrary/data/tree/master/biwi_face_database

[4] http://www.vision.ee.ethz.ch/~gfanelli/head_pose/head_forest.html

5.1.3. Pruebas

La resolución de la imagen RGB y del mapa de profundidad utilizados en las pruebas son:

- Imagen RGB 960×540 pixeles
- Mapa de profundidad 512×424 pixeles

Las pruebas se dividieron en dos etapas, la primera para detección de personas y la segunda para reconocimiento. Para ambas etapas se realizó su correspondiente en 2D y 3D por separado utilizando las mismas escenas para ambos casos. Para el caso 2D se utilizaron las imágenes RGB obtenidas con el sensor kinect, mientras que para el caso 3D se utilizaron los mapas de profundidad correspondientes a las imágenes del caso 2D, también obtenidas con kinect.

La etapa de detección fue organizada como sigue:

- Para la prueba se utilizaron 3 persona identificadas como *persona A*, *persona B* y *persona C*.

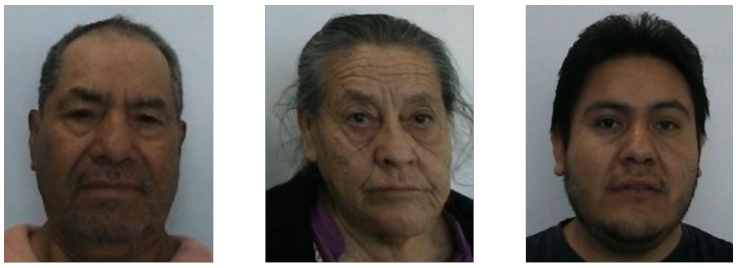


Figura 5.1: Personas que participaron en la prueba de detección. A) Persona A, B) Persona B y C) Persona C.

- Se realizaron 4 tipos de pruebas: Detección de 1 persona en el día, detección de 2 personas en el día, detección de 3 personas en el día y detección de 1 persona en la oscuridad.
- Las pruebas se realizaron a tres distancias del sensor, 1 metro, 2 metros y 3 metros.

Un ejemplo de la configuración de éstas pruebas se puede encontrar en el apéndice A.

La etapa de reconocimiento se organizó de la siguiente forma:

- Se utilizaron 6 personas identificadas como *persona A, persona B, persona C, persona D, persona E y persona F*, donde B y E son mujeres y los demás hombres. Se tomaron de forma aleatoria 30 muestras del rostro de cada una a 1 metro de distancia del sensor.

Figura 5.2: Personas que participaron en la prueba de reconocimiento.

- 20 de las 30 muestras de cada persona se tomaron para entrenamiento y las otras 10 restantes para reconocimiento.
- Se crearon 7 bases de entrenamiento:

  1. Utilizando las 20 muestras de cada persona. Todas las personas conocidas.
  2. Utilizando las muestras de las personas B, C, D, E, F. Persona A desconocida.
  3. Utilizando las muestras de las personas A, C, D, E, F. Persona B desconocida.
  4. Utilizando las muestras de las personas A, B, D, E, F. Persona C desconocida.
  5. Utilizando las muestras de las personas A, B, C, E, F. Persona D desconocida.
  6. Utilizando las muestras de las personas A, B, C, D, F. Persona E desconocida.

7. Utilizando las muestras de las personas A, B, C, D, E. Persona F desconocida.

- Para cada entrenamiento del punto anterior se probaron a reconocer las 10 muestras de cada persona. Cada muestra a probar no pertenece al conjunto de entrenamiento. Los resultados del reconocimiento se muestran en la siguiente sección.

Para cada una de las pruebas se mide el tiempo de ejecución y se repite 100 veces para obtener un promedio del tiempo. Una muestra de los datos de entrenamiento se presenta en el apéndice B.

5.2. Resultados

A continuación se muestran los resultados de las pruebas realizadas.

5.2.1. Detección

Se realizó la prueba de detección del rostro de 1 persona en 2D a tres distancias; 1 metro, 2 metros y 3 metros. Para cada distancia se realizaron 10 pruebas con imágenes distintas de una misma escena. Los resultados obtenidos se resumen en la tabla 5.1.

1 persona - detección 2D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1+	1+	1+	1+	1+	1+	1+	1+	1+	1+
2 m	1	1	1	1	1	1	1	1	1	1
3 m	1	0	1	0	1	1	0	1	0	1

Tabla 5.1: Tabla de resultados de detección de 1 rostro en 2D. 1+ corresponde a la detección del rostro y los ojos, 1 corresponde únicamente a la detección del rostro y 0 corresponde a rostro no detectado.

Como se puede apreciar en la tabla 5.1, la mejor detección se tiene cuando las personas se encuentran cerca del sensor, mientras que el rendimiento disminuye cuando las personas se alejan.

La siguiente prueba fue detectar 2 personas a las misma 3 distancias del sensor. Los resultados se muestran en la tabla 5.2, y como se puede apreciar el resultado es muy similar al caso de 1 persona.

La misma prueba se repitió con 3 personas a las mismas 3 distancias que las pruebas anteriores. En la tabla 5.3 se pueden observar los resultados obtenidos.

2 personas - detección 2D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1+	1+	1+	1+	1+	1+	1+	1+	1+	1+
2 m	1	1	1	1	1	1	1	1	1	1
3 m	0	0	0	0	0	0	0	0	0	0

Tabla 5.2: Tabla de resultados de detección de 2 rostros en 2D. 1+ corresponde a la detección de cara y ojos, 1 corresponde únicamente a la detección del rostro y 0 corresponde a rostro no detectado.

3 personas - detección 2D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1+	1+	1+	1+	1+	1+	1+	1+	1+	1+
2 m	1	1	1	1	1	1	1	1	1	1
3 m	0	0	0	0	0	0	0	0	0	0

Tabla 5.3: Tabla de resultados de detección de 3 rostros en 2D. 1+ corresponde a la detección de cara y ojos, 1 corresponde únicamente a la detección del rostro y 0 corresponde a rostro no detectado.

1 persona - detección 3D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1	1	1	1	1	1	1	1	1	1
2 m	0	0	0	0	0	0	0	0	0	0
3 m	0	0	0	0	0	0	0	0	0	0

Tabla 5.4: Tabla de resultados de detección de 1 rostro en 3D a 1 metro de distancia. 1 indica rostro detectado.

2 personas - detección 3D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1	1	1	1	1	1	1	1	1	1
2 m	0	0	0	0	0	0	0	0	0	0
3 m	0	0	0	0	0	0	0	0	0	0

Tabla 5.5: Tabla de resultados de detección de 2 rostros en 3D. 1 indica rostro detectado.

3 personas - detección 3D										
	P1	P2	P3	P4	P5	P6	P7	P8	P9	P10
1 m	1	1	1	1	1	1	1	1	1	1
2 m	0	0	0	0	0	0	0	0	0	0
3 m	0	0	0	0	0	0	0	0	0	0

Tabla 5.6: Tabla de resultados de detección de 3 rostros en 3D. 1 indica rostro detectado.

Aplicando el algoritmo de detección de rostros en 3D sobre las mismas escenas se obtuvieron los resultados mostrados en las tabas 5.4, 5.5 y 5.6. En base a estos resultados se puede decir que para detectar rostros a una mayor distancia, la detección utilizando información en 2D resulta mejor que la 3D, sin embargo ha ambos métodos les afecta considerablemente la distancia de los rostros al sensor. Por otro lado, la prueba de detección de un rostro a 1 metro de distancia se repitió en la oscuridad. El detector en 2D no pudo detectar el rostro debido a la falta de luz, mientras que el detector en 3D detectó el rostro sin ningún problema. Una muestra de la prueba se presenta en la figura 5.3.

Respecto al tiempo de detección se realizaron 100 veces cada prueba obteniendo el promedio de tiempo de ejecución de ambos métodos de detección. Al final se promediaron los tiempo y se obtuvo un tiempo promedio de 95.92 mili segundos para la detección en 2D. Por otro lado, se repitió el mismo procedimiento para 3D y se obtuvo un tiempo promedio de ejecución de 118.84 mili segundos. Cabe destacar que para la prueba de tiempo se aislaron los algoritmos de detección del proceso de adquisición. Lo anterior debido a que la adquisición de las imágenes del kinect consume una gran cantidad de recursos lo que provoca que los tiempos de ejecución aumenten.

A B C

Figura 5.3: Muestra de prueba de detección de rostros en la oscuridad a 1 metro de distancia del sensor. A) Imagen RGB con un brillo aumentado 40 %. B) Nube de puntos correspondiente a la persona en la escena. C) Imagen de profundidad del rostro encontrado.

5.2.2. Reconocimiento

En la tabla 5.7 se muestra la tabla de los resultados obtenidos para el reconocimiento 2D. Para ésta prueba se entrenaron las 6 personas con 20 muestras cada una y se identificaron con 10 muestras distintas a las entrenadas.

Reconocimiento 2D						
	PE-A	PE-B	PE-C	PE-D	PE-E	PE-F
PP-A	10	-	-	-	-	-
PP-B	-	10	-	-	-	-
PP-C	-	-	10	-	-	-
PP-D	-	-	-	9	1	-
PP-E	-	-	-	-	10	-
PP-F	-	-	-	-	-	10

Tabla 5.7: Tabla de confusión para reconocimiento en 2D. PE - Persona Entrenada. PP - Persona Prueba. Por cada persona se hicieron 10 pruebas. Todas las personas fueron entrenadas con 20 muestras cada una.

En la tabla 5.8 se muestran los resultados obtenidos para el reconocimiento 3D. La configuración es similar a la utilizada para 2D pero utilizando imágenes de profundidad.

Reconocimiento 3D						
	PE-A	PE-B	PE-C	PE-D	PE-E	PE-F
PP-A	10	-	-	-	-	-
PP-B	-	10	-	-	-	-
PP-C	-	-	10	-	-	-
PP-D	-	-	-	10	-	-
PP-E	-	-	-	-	10	-
PP-F	-	-	-	-	-	10

Tabla 5.8: Tabla de confusión para reconocimiento en 3D. PE - Persona Entrenada. PP - Persona Prueba. Por cada persona se hicieron 10 pruebas. Todas las personas fueron entrenadas con 20 muestras cada una.

Como se puede observar en las tablas 5.7 y 5.8, ambos reconocimientos presentan buenos resultados en condiciones similares. Lo que se pudo notar en las pruebas fue el nivel de confiabilidad de los resultados. La confiabilidad se determinó de la siguiente manera:

- Se calcula el error (Apéndice C) entre la imagen reconstruida de cada prueba y cada una de las imágenes de entrenamiento.
- Por cada imagen de prueba, se elige la imagen de entrenamiento con el menor error, es decir, la imagen de entrenamiento más parecida a la imagen de prueba.
- Se normaliza el error del paso anterior entre 0 y 1.
- Finalmente, se establece la confiabilidad como la diferencia entre 1 y el error.

En la figura 5.4 se muestran los resultados de la confiabilidad de ambas pruebas. Como se aprecia en las gráficas el reconocimiento en 3D muestra una mejor estabilidad en la confiabilidad de sus resultados mientras que el reconocimiento 2D muestra una mayor variabilidad.

La siguiente prueba realizada fue quitar el entrenamiento de cada uno de los sujetos de prueba y repetir el reconocimiento. La intensión es observar el comportamiento de ambos sistemas ante personas desconocidas. En la figura 5.5 se muestran las gráficas comparativas respecto a la prueba mencionada. En la gráfica se muestra por cada persona de prueba, el resultado de la identificación, es decir, la persona a la que más se parece el sujeto de prueba. Como se puede apreciar en las gráficas, el reconocimiento en 3D muestra una predicción más estable mientras que las predicciones para el reconocimiento en 2D se muestran más variables. Lo anterior puede ser un indicio de

que al utilizar la información de profundidad se pueden obtener mejores resultados comparando dos rostros de una misma persona.

Respecto al tiempo de ejecución del reconocimiento, cada prueba fue repetida 100 veces por cada sujeto de prueba obteniendo un valor promedio del tiempo de cada ejecución de la prueba. Después se promediaron todos los tiempos y se obtuvo un tiempo de ejecución de 3.6 milisegundos para la detección en 2D y 3D ya que en ambos se utilizó el mismo método.

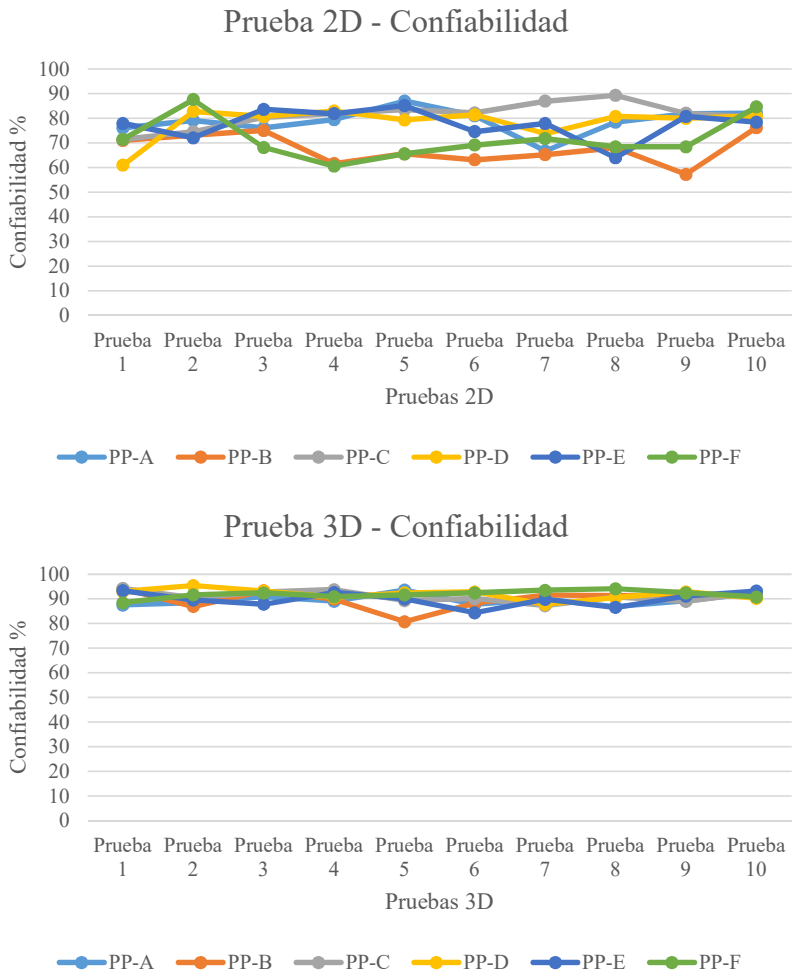


Figura 5.4: Gráficas comparativas de la confiabilidad del reconocimiento 2D y 3D. PP significa Persona de Prueba.

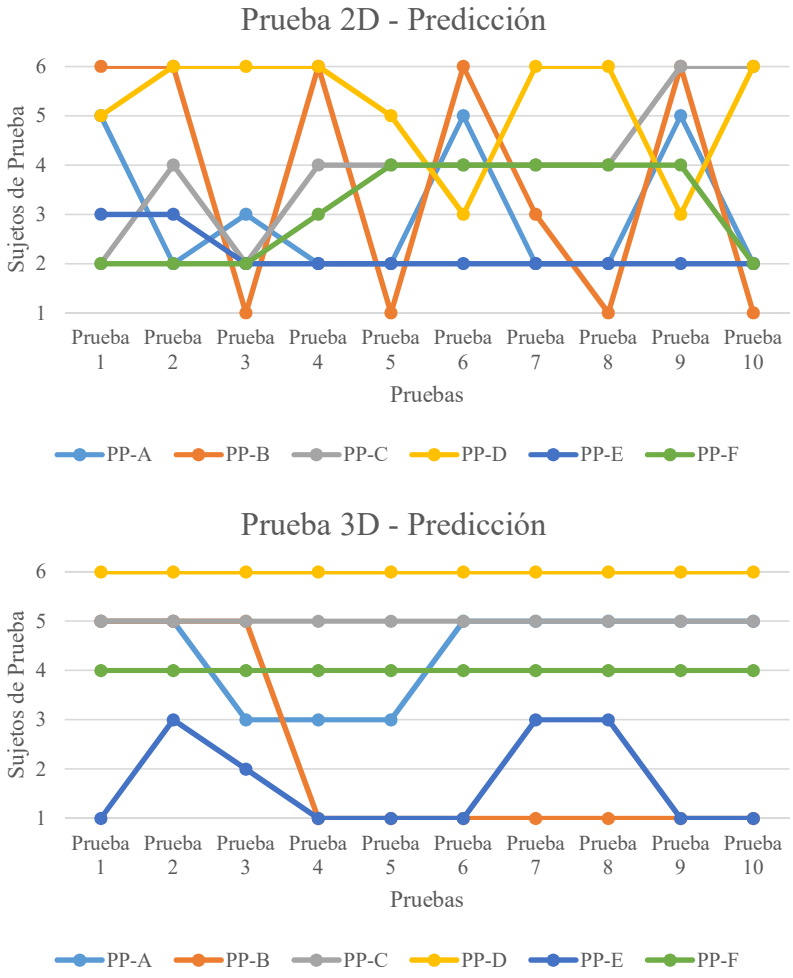


Figura 5.5: Gráficas comparativas que muestran las predicciones en 2D y 3D para personas desconocidas. Por cada persona de prueba se muestra la persona a la que más se parece (predicción) del conjunto de personas conocidas por el sistema. Cabe aclarar que cada persona de prueba es desconocida para el sistema. PP significa Persona de Prueba.

Capítulo 6

Conclusiones y Trabajo a Futuro

6.1. Conclusiones

Con base a los resultados obtenidos en el capítulo anterior se puede concluir lo siguiente:

- Los algoritmos de detección expuestos en este trabajo presentan problemas en la detección de rostros a distancia debido a la poca información que se obtiene en estos casos. A pesar de que en los dos algoritmos se puede variar la ventana de búsqueda para intentar detectar a distintas escalas, siempre se esta limitado al tamaño de los rostros y las ventanas utilizadas en el entrenamiento del clasificador, por lo que sería conveniente tener un entrenamiento para cada situación en que se necesite, pero esto último resulta complicado debido a la gran cantidad de situaciones posibles.

- Como se demostró en las pruebas, utilizando la información de profundidad proporcionada por el kinect es posible detectar rostros en distintas condiciones de iluminación sin que se vea afectado. Ésta es una de las ventajas de éste tipo de sensores, lo que podría convertirlos en una opción para ciertas tareas como son la video vigilancia.

- Respecto al proceso de reconocimiento facial, se mostró evidencia que indica que utilizar la información de profundidad podría proporcionar mejores resultados en comparación con las imágenes tradicionales, incluso utilizando métodos ya conocidos. Ésto no solo aplica para la identificación de rostros ya que se puede extender a la identificación de objetos.

- Es posible mezclar las ventajas de ambos métodos, 2D y 3D, dependiendo de la situación pudiendo utilizar la detección en 2D para determinar la posición del rostro en una escena y obtener la imagen de profundidad correspondiente, la cual sería utilizada para el reconocimiento. Como se mostró en la sección de resultados la detección facial en 2D tiene mejores resultados a mayores distancias

que la detección 3D. Por el contrario, el reconocimiento en 3D muestra mejores resultados que en 2D.

6.2. Trabajo a futuro

- Todas las pruebas realizadas fueron con personas viendo hacia el sensor. Ésta situación no siempre se cumple en un entorno real por lo que podría ser necesario encontrar otro método capaz de detectar y reconocer rostros en distintas poses.
- Es posible obtener la información geométrica de un rostro mediante un sensor de profundidad, lo que podría ser utilizado para el reconocimiento facial. La técnica que se presentó no aprovecha del todo ésta información. Sería conveniente encontrar un algoritmo que permita codificar las características geométricas de un rostro y compararlas con otros.
- La detección del rostro en 3D nos puede proporcionar información acerca de la posición y orientación de la cabeza de una persona. El problema es que para ésto, el algoritmo presentado funciona únicamente para distancias cortas. Sería conveniente examinar otras opciones para encontrar una mejor solución al problema de la escala.
- Para el reconocimiento podría utilizarse la información de la curvatura del rostro aprovechando la información de profundidad. En algunos trabajos [35] se hace uso de la curvatura de la superficie para la detección de rostros. Sería conveniente evaluar dicha alternativa.

Apéndice A

Configuración para Detección

A continuación se presenta un ejemplo de la configuración de la prueba de detección facial con 1, 2 y 3 personas a 1, 2 y 3 metros de distancia al sensor.

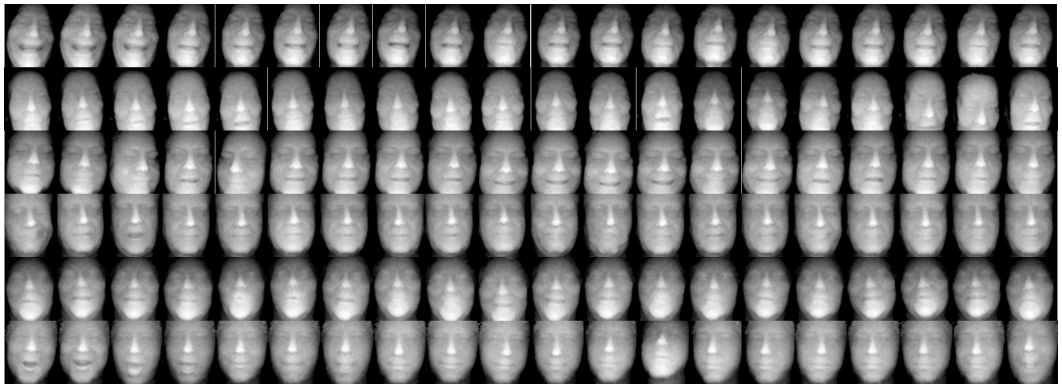
Cada una de las configuraciones se probó con distintos gestos y posiciones del rostro, siendo las pruebas a 3 metros las que arrojaron los peores resultados.

Apéndice B

Imágenes para Reconocimiento

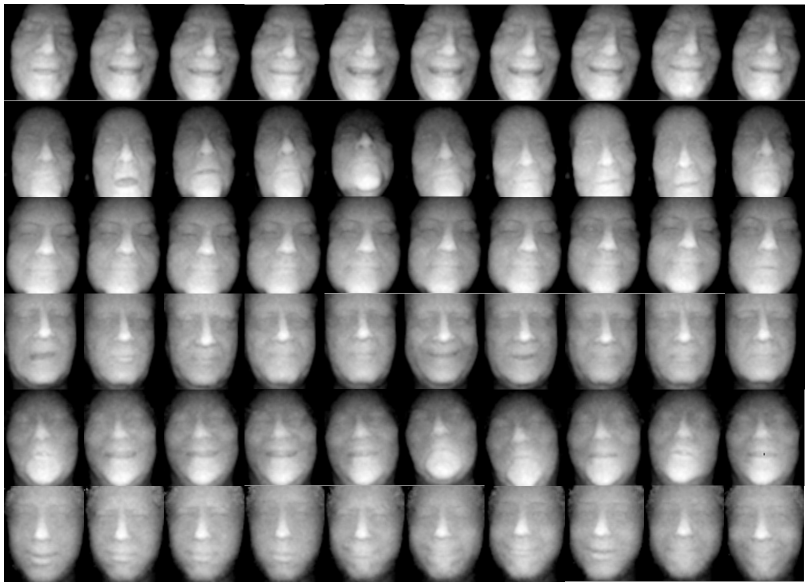
Imágenes de entrenamiento utilizadas para el reconocimiento 2D. Cada fila corresponde a una persona de prueba.

Imágenes de entrenamiento utilizadas para el reconocimiento 3D. Cada fila corresponde a una persona de prueba.



Imágenes de prueba utilizadas para el reconocimiento 2D.

Imágenes de prueba utilizadas para el reconocimiento 3D.



Apéndice C

Cálculo del Error

Si consideramos una imagen como un vector de pixeles, para determinar la diferencia entre dos imágenes, a lo que llamaremos error, se determina la distancia euclidiana entre los valores de los pixeles de las dos imágenes.

$$|Img1 - Img2| = \sqrt{\sum_{I}(Img1[I] - Img2[I])^2}$$

Lo anterior es lo mismo que obtener la norma L2 de la diferencia de los vectores. En OpenCV se puede obtener dicho valor utilizando la función:

```
double errorL2 = norm(Mat A, Mat B, CV_L2);
```

Cada pixel es normalizado en el intervalo $[0, 1]$ por lo que el valor de la norma L2 puede ser a lo más el número de pixeles en la imagen, así que lo normalizamos en el intervalo $[0, 1]$ para obtener el error de las imágenes.

```
double getError(const Mat A, const Mat B) {
        double errorL2 = norm(A, B, CV_L2);
        double errornorm = errorL2 / (double)(A.rows * A.cols);
        return errornorm;
}
```

Finalmente el valor de $1 - error$ nos dará la semejanza de las imágenes, a lo que llamaremos confiabilidad o certidumbre.

Bibliografía

[1] *Face Recognition*, Octubre 2015. `http://www.face-rec.org/general-info/`. 1

[2] Kanade, Takeo: *Computer Recognition of Human Faces*. Interdisciplinary Systems Research, 47, 1977. 6

[3] Lu, Yongzhong, Jingli Zhou y Shengsheng Yu: *A survey of face detection, extraction and recognition*. Computing and Informatics, 22(2), 2003. 6, 7

[4] Jeng, Shi Hong, Hong Yuan Mark Liao, Chin Chuan Han, Ming Yang Chern y Yao Tsorng Liu: *Facial feature detection using geometrical face model: An efficient approach*. Pattern Recognition, 31(3):273 – 282, 1998. 7

[5] Wang, Jianguo y Tieniu Tan: *A new face detection method based on shape information*. Pattern Recognition Letters, 21(6–7):463 – 471, 2000. 7

[6] Aggarwal, Jake K. y Quin Cai: *Human Motion Analysis: A Review*. Computer Vision and Image Understanding, 73:428–440, 1999. 7

[7] Levi, Kobi y Yair Weiss: *Learning object detection from a small number of examples: the importance of good features*. En *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volumen 2, páginas II–53–II–60 Vol.2, Junio 2004. 7

[8] Lin, Shang Hung, Sun Yuan Kung y Long Ji Lin: *Face recognition/detection by probabilistic decision-based neural network*. Neural Networks, IEEE Transactions on, 8(1):114–132, Enero 1997, ISSN 1045-9227. 7

[9] Kotropoulos, Constantine, Anastasios Tefas y Ioannis Pitas: *Frontal face authentication using morphological elastic graph matching*. Image Processing, IEEE Transactions on, 9(4):555–560, Abril 2000, ISSN 1057-7149. 7

[10] Viola, Paul y Michael J. Jones: *Robust Real-Time Face Detection*. International Journal of Computer Vision, 57(2):137–154, 2004. 7, 11, 12, 14, 15, 16, 17, 18, 19, 21

[11] Viola, Paul y Michael Jones: *Fast and Robust Classification using Asymmetric AdaBoost and a Detector Cascade.* En *Advances in Neural Information Processing System 14*, páginas 1311–1318. MIT Press, 2001. 7

[12] Freund, Yoav y Robert E. Schapire: *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting.* Journal of Computer and System Sciences, 55(1):119–139, 1997. 7, 17

[13] Wang, Yi Qing: *An Analysis of the Viola-Jones Face Detection Algorithm.* Image Processing On Line, 4(1):128–148, 2014. 7, 17, 18, 19, 20

[14] Sirovich, Lawrence y Michael Kirby: *Low-dimensional procedure for the characterization of human faces.* J. Opt. Soc. Am. A, 4(3):519–524, Marzo 1987. 8

[15] Turk, Matthew y Alex Pentland: *Eigenfaces for Recognition.* Journal of Cognitive Neuroscience, 3(1):71–86, 1991. 8, 9, 23, 25

[16] Cartoux, Jean Yves, Jean Thierry Lapreste y Marc Richetin: *Face authentification or recognition by profile extraction from range images.* En *Interpretation of 3D Scenes, 1989. Proceedings., Workshop on*, páginas 194–199, Noviembre 1989. 10

[17] Lee, John C. y Evangelos Milios: *Matching range images of human faces.* En *Computer Vision, 1990. Proceedings, Third International Conference on*, páginas 722–726, Diciembre 1990. 10

[18] Nagamine, Takashi, Tetsuya Uemura y Isao Masuda: *3D facial image analysis for human identification.* En *Pattern Recognition, 1992. Vol.I. Conference A: Computer Vision and Applications, Proceedings., 11th IAPR International Conference on*, páginas 324–327, Agosto 1992. 10

[19] Xu, Chenghua, Yunhong Wang, Tieniu Tan y Long Quan: *Automatic 3D face recognition combining global geometric features with local shape variation information.* En *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, páginas 308–313, Mayo 2004. 10

[20] Gökberk, Berk, AlbertAli Salah y Lale Akarun: *Rank-Based Decision Fusion for 3D Shape-Based Face Recognition.* En Kanade, Takeo, Anil Jain y NaliniK. Ratha (editores): *Audio- and Video-Based Biometric Person Authentication*, volumen 3546 de *Lecture Notes in Computer Science*, páginas 1019–1028. Springer Berlin Heidelberg, 2005, ISBN 978-3-540-27887-0. 11

[21] Fanelli, Gabriele, Matthias Dantone, Juergen Gall, Andrea Fossati y Luc Van Gool: *Random Forests for Real Time 3D Face Analysis.* International Journal of Computer Vision, 101(3):437–458, 2013. 11, 12, 31, 33, 35, 36, 37, 42

[22] Zhou, Zhi Hua: *Ensemble Learning.* En Li, Stan Z. y Anil K. Jain (editores): *Encyclopedia of Biometrics*, páginas 270–273. Springer US, 2009, ISBN 978-0-387-73003-5. 11

[23] Criminisi, Antonio, Jamie Shotton y Ender Konukoglu: *Decision Forests for Classification, Regression, Density Estimation, Manifold Learning and Semi-Supervised Learning.* Informe técnico, Microsoft Research, 2011. 11

[24] Achermann, Bernard, Xiaoyi Jiang y Horst Bunke: *Face recognition using range images.* En *Virtual Systems and MultiMedia, 1997. VSMM '97. Proceedings., International Conference on*, páginas 129–136, Septiembre 1997. 12

[25] Beumier, Charles y Marc Acheroy: *Automatic Face Verification from 3D and Grey Level Clues.* En *11th Portuguese Conference on Pattern Recognition*, páginas 95–101, 2000. 12

[26] Tsalakanidou, Filareti, Dimitrios Tzovaras y Michael G. Strintzis: *Use of depth and colour eigenfaces for face recognition.* Pattern Recognition Letters, 24(9–10):1427 – 1435, 2003. 12

[27] Papatheodorou, Theodoros y Daniel Rueckert: *Evaluation of automatic 4D face recognition using surface and texture registration.* En *Automatic Face and Gesture Recognition, 2004. Proceedings. Sixth IEEE International Conference on*, páginas 321–326, Mayo 2004. 12

[28] Zhao, Wen Yi, Rama Chellappa, P. Jonathon Phillips y A. Rosenfeld: *Face Recognition: A Literature Survey.* ACM Comput. Surv., 35(4):399–458, Diciembre 2003. 13

[29] Papageorgiou, Constantine, Michael Oren y Tomaso Poggio: *A general framework for object detection.* International Conference on Computer Vision, páginas 555–562, 1998. 15

[30] Baggio, Daniel Lélis, Shervin Emami, David Millan Escriva, Khvedchenia Ievgen, Naureen Mahmood, Jasonl Saragih y Roy Shilkrot: *Mastering OpenCV with Practical Computer Vision Projects.* Packt Publishing, Limited, 2012, ISBN 9781849517829. 21

[31] *Kinect2 Driver*, Octubre 2015. `https://github.com/occipital/OpenNI2/tree/kinect2/Source/Drivers/Kinect2`. 31, 41, 42

[32] Fanelli, Gabriele, Thibaut Weise, Juergen Gall, Andrea Fossati y Luc Van Gool: *Real Time Head Pose Estimation from Consumer Depth Cameras.* Proceedings of the 33rd International Conference on Pattern Recognition, 6835:101–110, 2011. 33, 34, 35

[33] Lienhart, Rainer y Jochen Maydt: *An extended set of Haar-like features for rapid object detection.* En *Image Processing. 2002. Proceedings. 2002 International Conference on*, volumen 1, páginas I–900–I–903 vol.1, 2002. 42

[34] Castrillón Santana, Marco, Oscar Déniz Suárez, Mario Hernández Tejera y Cayetano Guerra Artal: *ENCARA2: Real-time Detection of Multiple Faces at Different Resolutions in Video Streams.* Journal of Visual Communication and Image Representation, páginas 130–140, April 2007. 42

[35] Gordon, Gaile G.: *Face recognition based on depth maps and surface curvature.* En *SPIE Geometric methods in Computer Vision*, páginas 234–247, Septiembre 1991. 54