



UNIVERSIDAD DE JAÉN
Escuela Politécnica Superior (Jaén)

Trabajo Fin de Máster

SISTEMA DE IDENTIFICACIÓN Y RECONOCIMIENTO DE PERSONAS MEDIANTE VISIÓN POR COMPUTADOR

Alumno/a: Rodríguez Gutiérrez, Alberto Juan

**Tutor/a: Prof. Dña. Elisabet Estevez Estevez
Prof. Dña. Silvia María Satorres Martínez**

Dpto.: Ingeniería Electrónica y Automática

Septiembre, 2017



Universidad de Jaén
Escuela Politécnica Superior de Jaén

Doña Elisabet Estevez Estevez y doña Silvia Maria Satorres Martinez, tutoras del Trabajo Final de Master titulado: Sistema de Identificación y Reconocimiento de Personas mediante Visión por Computador, que presenta Alberto Juan Rodríguez Gutiérrez, autoriza su presentación para defensa y evaluación en la Escuela Politécnica Superior de Jaén.

Jaén, Septiembre de 2017

El alumno:

Alberto Juan Rodríguez Gutierrez

Las tutoras:

Elisabet Estevez Estevez

Silvia Maria Satorres Martinez

Índice de contenido

Índice de figuras.....	III
Índice de tablas.....	IV
1. Introducción.....	1
1.1. Motivación.....	1
1.2. Objetivos del TFM.....	3
1.3. Resumen de la memoria.....	3
2. Sistemas de reconocimiento facial.....	7
2.1. Reconocimiento basado en sistemas 2D.....	8
2.2. Reconocimiento basado en sistemas 3D.....	12
2.3. Reconocimiento basado en sistemas 2D+3D.....	13
2.4. Conclusiones de los sistemas de reconocimiento facial y elección de sistema.....	14
3. Diseño del sistema de adquisición.....	17
3.1. Selección de la cámara.....	19
3.2. Sensores posición de personas.....	27
4. Hardware del sistema.....	29
4.1. Sensor láser para la posición del usuario.....	29
4.2. Display para informar al usuario.....	30
4.3. Sistema de conexión del láser y el display con el ordenador.....	31
4.4. Diseño del prototipo.....	32
5. Software del sistema.....	35
5.1. Arduino en Matlab.....	35
5.2. Detección de la posición de usuario.....	36
5.3. Informar al usuario mediante la pantalla LCD.....	39
5.4. Obtención de la imagen.....	42
5.5. Cálculo de los puntos característicos.....	49
5.6. Base de datos.....	50
5.7. Red neuronal.....	51
6. Experimentos y resultados.....	55
6.1. Experimento 1: Desviación de los puntos en capturas.....	55
6.2. Experimento 2: Variación de las expresiones.....	57
6.3. Experimento 3: Cambios de posición.....	59
7. Caso de estudio: control de entrada y salida de los trabajadores de una planta.....	61

7.1. Control de acceso	61
7.2. Gestión de usuarios	63
7.3. Registro de eventos	64
8. Conclusiones.....	67
9. Bibliografía	69
Anexo I: Requisitos del sistema.....	73
Requisitos de Hardware.....	73
Requisitos de Software	73

Índice de figuras

Figura 1 Tareas fundamentales en el reconocimiento facial	7
Figura 2 Método PCA para el reconocimiento facial	9
Figura 3 Método usando una red neuronal para reconocimiento facial.....	10
Figura 4 Método de filtros Gabor para el reconocimiento facial	11
Figura 5 Método IFS para el reconocimiento facial.....	11
Figura 6 Imagen 2.5D y nube de puntos de una cara	12
Figura 7 Gráfico de precisión de reconocimiento en 2D, 3D y 2D+3D (Chang, et al., 2004) .	13
Figura 8 Esquema del sistema de adquisición.....	17
Figura 9 Diagrama de flujo general del sistema completo	18
Figura 10 Sistema de adquisición 3dMDface.....	19
Figura 11 Sistema de adquisición Minolta Vivid 910.....	20
Figura 12 Sistema de adquisición Artec EVA	21
Figura 13 Sistema de adquisición HDI Advance R1X	22
Figura 14 Sistema de adquisición Microsoft Kinect.....	23
Figura 15 Sistema de adquisición Xtion PRO LIVE	24
Figura 16 Sensor VL53L0X	29
Figura 17 Placa adquirida para la medida de la distancia.....	30
Figura 18 Display LCD 16x2.....	30
Figura 19 Conexión de todos los componentes con la placa arduino	31
Figura 20 Soporte para el sistema de adquisición	33
Figura 21 Soporte finalizado para la cámara	34
Figura 22 Funcionamiento de custom add-on library	36
Figura 24 Función para medir la posición	37
Figura 23 Inicialización del sensor.....	37
Figura 25 Inicialización de la pantalla	39
Figura 26 Función para mostrar mensaje	39
Figura 27 Función para la obtención de los puntos	42
Figura 28 Componentes del sensor Kinect.....	43
Figura 29 Lista de valores del array con los puntos de la cara	47
Figura 30 Red con una capa	51
Figura 31 Proceso de ajuste de pesos en aprendizaje supervisado (Matlab, 2017).....	51
Figura 32 Interfaz de control de acceso.....	62
Figura 33 Interfaz Gestión de usuarios.....	64
Figura 34 Interfaz del Registro de eventos	65

Índice de tablas

Tabla 1 Precisión de los distintos metodos de reconocimiento facial.....	14
Tabla 2 Características discriminantes que se van a usar en el sistema	15
Tabla 3 Características 3dMDface	19
Tabla 4 Características Minolta Vivid 910.....	20
Tabla 5 Características Artec EVA	21
Tabla 6 Características HDI Advance R1X.....	22
Tabla 7 Características Microsoft Kinect	23
Tabla 8 Características Xtion PRO LIVE	24
Tabla 9 Tabla de medidas y pesos para la selección	25
Tabla 10 Tabla de medidas y pesos normalizados para la selección.....	25
Tabla 11 Tabla de selección PROMETHEE	26
Tabla 12 Valores de FaceProperties de la API Kinect	45
Tabla 13 Valores de la API de Kinect usados para el sistema	47
Tabla 14 Pasos fundamentales en el diseño de una red neuronal.....	52
Tabla 15 Resultados del usuario 1 en el experimento 1	55
Tabla 16 Resultados del usuario 2 en el experimento 1	56
Tabla 17 Tabla resumen del experimento 1.....	56
Tabla 18 Resultados del usuario 1 en el experimento 2	57
Tabla 19 Resultados del usuario 2 en el experimento 2	57
Tabla 20 Resultados en el experimento 3	59

1. INTRODUCCIÓN

1.1. Motivación

La biometría es el estudio y reconocimiento automático de las personas mediante sus rasgos significativos, pueden ser físicos (cara, iris, voz, huellas, etc...) o comportamiento (firma, forma de andar, etc...). Esto implica que quedan exentas de la categoría los sistemas no automáticos y los métodos de identificación empleados en la medicina forense.

Poder identificar a una persona por sus rasgos faciales implica haber obtenido previamente las medidas, haberlas almacenado y finalmente compararlas con los datos en una base de datos. Para que una característica sea adecuada, debe cumplir las siguientes propiedades (Moreno, 2004):

- Que cualquier persona la posea
- Que la característica sea diferente en todas las personas
- Que no se pueda falsificar
- Que sea fácil de obtener

La principal razón para realizar un reconocimiento biométrico es la de asegurarse que se identifica a la persona correctamente. Comúnmente se usan contraseñas, llaves, tarjetas y otros sistemas para la identificación, pero no se garantiza que puedan ser perdidos, robados u olvidados. Esto refuerza los esfuerzos en obtener sistemas de reconocimientos más fiables y en la investigación de nuevas características de identificación. Además, los sistemas biométricos presentan la principal ventaja de poder ser identificado en cualquier momento de manera rápida y fiable.

Una de las principales ventajas del reconocimiento facial es que es uno de los métodos de identificación menos intrusivo y más aceptado por la sociedad (Li & Jain, 2011), ya que es la forma natural con la que el ser humano identifica a las personas. Hoy día, el sistema más popular para la identificación biométrica es el uso de la huella dactilar con 52% de las aplicaciones, seguido del uso del reconocimiento facial con un 11,4% y otros como la mano (10%), iris (7,3%), voz (4,1%) y otros (15,2%) (Abate, et al., 2007).

Los primeros trabajos de reconocimiento facial informático aparecieron en los años 70 (Kanade, 1977) (Goldstein, et al., 1971). Estos consistían en marcar manualmente en la imagen los puntos más discriminantes y posteriormente, mediante técnicas estadísticas, se obtenían los resultados en un ordenador. Ya en la década de los 80 comenzaron los trabajos donde los puntos más discriminantes de una imagen eran obtenidos automáticamente.

En la década de los 90 el número de trabajos empezó a aumentar considerablemente, llegando a obtener tasas de acierto mayores del 90% en entornos controlados en el laboratorio (Cottrell & Fleming, 1990) (Lades, et al., 1993). Además, se crearon asociaciones con el fin de seguir estudiando las diferentes opciones.

Actualmente, los sistemas de reconocimiento facial no son solo usados para sistemas de seguridad (control de acceso, identificación policial, agilización en procesos de identificación...), sino que se está apostando por nuevas aplicaciones comerciales en donde el reconocimiento de las personas permite una atención más personalizada (reconocimiento en vehículos, casas, oficinas inteligentes, atención personalizada en espacios comerciales, etc...).

Se realiza este trabajo con el fin de crear un sistema robusto para el reconocimiento de las personas, no siempre se pueden recrear las condiciones de laboratorio. Se pretende también en crear un sistema más rápido que el reconocimiento con la huella dactilar y con el fin de utilizar el sistema en entornos donde no se pueda usar las manos por estar estas ocupadas, por suciedad en el trabajo o por motivos de higiene.

1.2. Objetivos del TFM

Este trabajo tiene los siguientes objetivos:

- Selección de características discriminantes que permitan identificar a una persona.
- Análisis de los diferentes sensores de visión que permitan extraer dichas características. Este objetivo concluye con la selección de la cámara más adecuada.
- Definición de una Base de Datos de individuos a reconocer.
- Aplicación de técnicas de Inteligencia Artificial que dadas unas características permitan decir quién es el individuo identificado.
- Caso de estudio: control de entrada y salida de los trabajadores de una planta.

1.3. Resumen de la memoria

En los siguientes capítulos el lector puede encontrar la siguiente información:

- **Capítulo 2 (Sistemas de reconocimiento facial):**
 - Tecnologías y métodos para el reconocimiento facial: comparativa entre las distintas tecnologías que se emplean.
 - Sistemas 2D: uso de una cámara digital de color.
 - Sistemas 3D: uso de mallas o nubes de puntos para la identificación.
 - Sistemas 2D+3D: uso de ambos sistemas simultáneamente.
 - Conclusión y selección del método elegido para el sistema de reconocimiento.
- **Capítulo 3 (Diseño del sistema de adquisición):**
 - Se presentarán las cámaras 3D comerciales más comunes compatibles con el sistema.
 - Comparativa y selección de la cámara utilizada.
 - Estudio de los distintos sistemas de posición de personas y se seleccionara el más adecuado.

- **Capítulo 4 (Diseño del hardware del sistema):**
 - Presentación del sensor láser utilizado para obtener la posición de la persona.
 - Explicación de la pantalla elegida para el sistema.
 - Estudio y diseño del sistema de sujeción de la cámara, el sensor, la placa Arduino y la pantalla LCD.
- **Capítulo 5 (Diseño del software del sistema)**
 - Explicación de las distintas etapas que se van a utilizar, tanto para agregar usuarios como para reconocer personas.
 - Introducción al funcionamiento de la placa en Matlab.
 - Flujos de trabajo y programación de las partes controladas por arduino:
 - Sensor láser.
 - Pantalla LCD.
 - Explicación, funcionamiento y programación de la cámara Kinect para obtener los puntos más discriminantes.
 - Desarrollo de los métodos para el cálculo de las relaciones entre los puntos característicos.
 - Definición de la base de datos.
 - Elaboración e introducción a la red neuronal diseñada para el sistema.
- **Capítulo 6 (Experimentos y resultados):**
 - Se realizarán experimentos para comprobar la validez de los puntos obtenidos por la cámara:
 - Variación de valores entre capturas.
 - Influencia de los gestos de la cara.
 - Comprobación del sistema frente a cambios de posición.
- **Capítulo 7 (Caso de estudio: control de entrada y salida de los trabajadores de una planta):** Se realizará una aplicación sencilla en donde se ponga en práctica todo lo desarrollado en el trabajo.
- **Capítulo 8 (Conclusiones):** Argumentación y exposición de las conclusiones finales.

1.3. Resumen de la memoria

- **Capítulo 9 (Bibliografía):** Listado de referencias bibliográficas utilizadas para el desarrollo del proyecto.

2. SISTEMAS DE RECONOCIMIENTO FACIAL

Actualmente, se pueden diferenciar entre tres formas de identificación de las personas mediante reconocimiento facial.

- Reconocimiento tradicional: basado en imágenes en dos dimensiones (2D) captadas por una cámara digital.
- Reconocimiento 3D: se basa en el uso de una malla facial o una nube de puntos captados por un sensor de profundidad.
- Una forma mixta de ambas: utiliza imágenes 2D como el método tradicional y también imágenes 3D, combinándolas para obtener las ventajas de ambos.

Los sistemas de reconocimiento facial involucran tres tareas (Zhao, et al., 2003), como muestra la Figura 1.

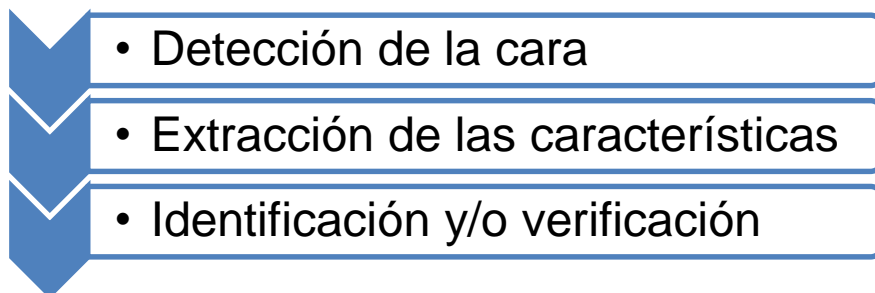


Figura 1 Tareas fundamentales en el reconocimiento facial

En (Li & Jain, 2011) se presentan los principales algoritmos que usan los sistemas de reconocimiento facial:

- Basados en la apariencia (holísticos)
Estos métodos usan toda la región de la cara como método de análisis.
- Basado en las características
Típicamente, en este método, de características locales de los ojos, nariz y boca se extraen sus distancias y relaciones, se miden y se comparan
- Una aproximación híbrida de las anteriores.
El ser humano usa un sistema híbrido de ambos para reconocer una persona, que es justamente lo que propone este método, usar todas las características para identificar a un individuo.

2.1. Reconocimiento basado en sistemas 2D

Históricamente, es el sistema más utilizado. Utiliza imágenes captadas por una cámara digital. Esto implica que sea un sistema muy económico y haya numerosos estudios en este campo, llegando a obtener una tasa de reconocimiento superior al 90% en entornos controlados (Phillips, et al., 2002).

Para el reconocimiento de imágenes en 2D se siguen distintas tendencias (Abate, et al., 2007):

- Métodos de proyección lineal/no lineal
- Redes neuronales
- Filtros Gabor
- Los fractales y los sistemas iterativos de funciones
- Termales e hiperespectrales

Desafortunadamente estos sistemas se encuentran cinco factores clave que pueden afectar significativamente los sistemas de reconocimiento facial (Li & Jain, 2011):

- Variaciones de iluminación: estos problemas vienen debido a la reflectancia de la piel.
- Variaciones de la posición: puede ocurrir que al variarse se introduzcan deformaciones en la proyección o incluso se tape parcialmente alguna de las características empleadas.
- Variación temporal: con el paso del tiempo la cara va variando, como por ejemplo el tono de piel, es un tema difícil de resolver, aunque se puede ir actualizando la base de datos
- Oclusiones: uso de gafas, barba u otros pueden afectar negativamente en la identificación.

Se va a explicar a continuación las distintas líneas de investigación que se siguen actualmente para desarrollar los sistemas de reconocimiento 2D.

2.1. Reconocimiento basado en sistemas 2D

2.1.1. Métodos de proyección lineal/no lineal

Estos métodos se basan en el *análisis de las componentes principales* (PCA), son métodos holísticos que consisten en la extracción de vectores con las características de la cara y posteriormente reducirlas en vectores característicos, para que sea más fácil su procesamiento, tal y como se ve en la Figura 2.

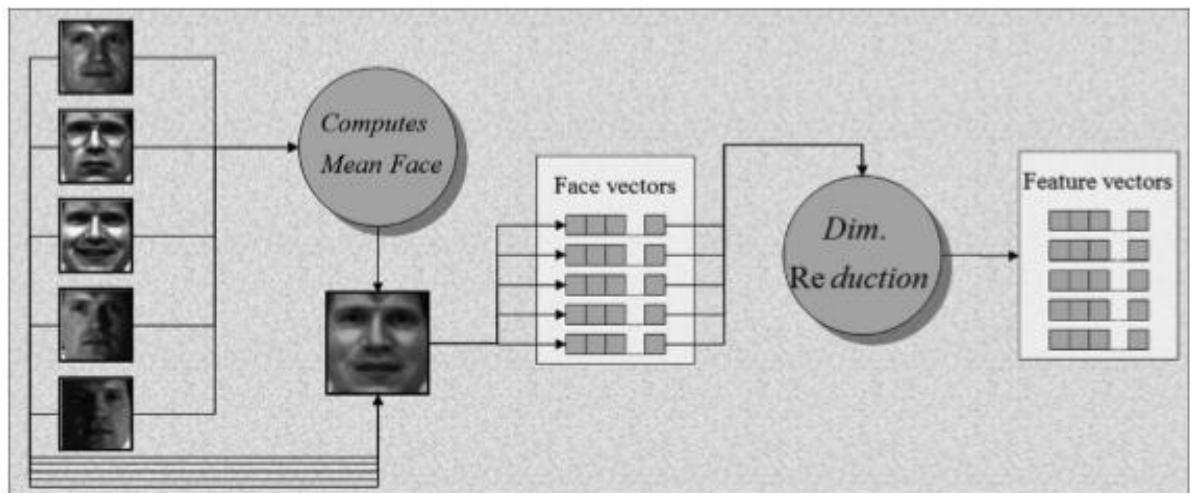


Figura 2 Método PCA para el reconocimiento facial

Uno de los primeros sistemas en utilizar este sistema (PCA) fue Eigenfaces (Kirby & Sirovich, 1990), el cual se toman varias imágenes de tamaño $N \times N$ píxeles y las linealizara en un vector de N^2 . Después de la linealización de todas las imágenes, se obtienen los valores que más se repiten. Cuando se analiza una nueva imagen, se comparará con la base de datos obtenida previamente (por ejemplo, buscando la mínima distancia euclídea).

A parte del PCA, se han estudiado otros métodos como el Linear Discriminant Analysis (LDA) (Lu, et al., 2003) (Martinez & Kak, 2001). El LDA consiste en encontrar la base de vectores que da la mayor discriminación entre clases, buscando que haya una mayor diferencia entre las características a comparar, asegurándose que disminuyen las dependientes entre sí.

2.1.2. Redes neuronales

El uso de las redes neuronales ha sido amplio en muchos campos del reconocimiento de patrones. La idea básica es considerar una red con una neurona por cada pixel en la imagen. Aunque, debido al tamaño de las imágenes se suele usar una técnica de reducción de la imagen.

Una primera solución adoptada (Cottrell & Fleming, 1990) consistió en la utilización de una segunda red neuronal, como indica la Figura 3. Al principio, la imagen representada se linealiza en un vector X , que al pasar por la primera capa se transforma en un vector de menor tamaño que se va a llamar h . El vector h se utilizará en la segunda capa para el reconocimiento.

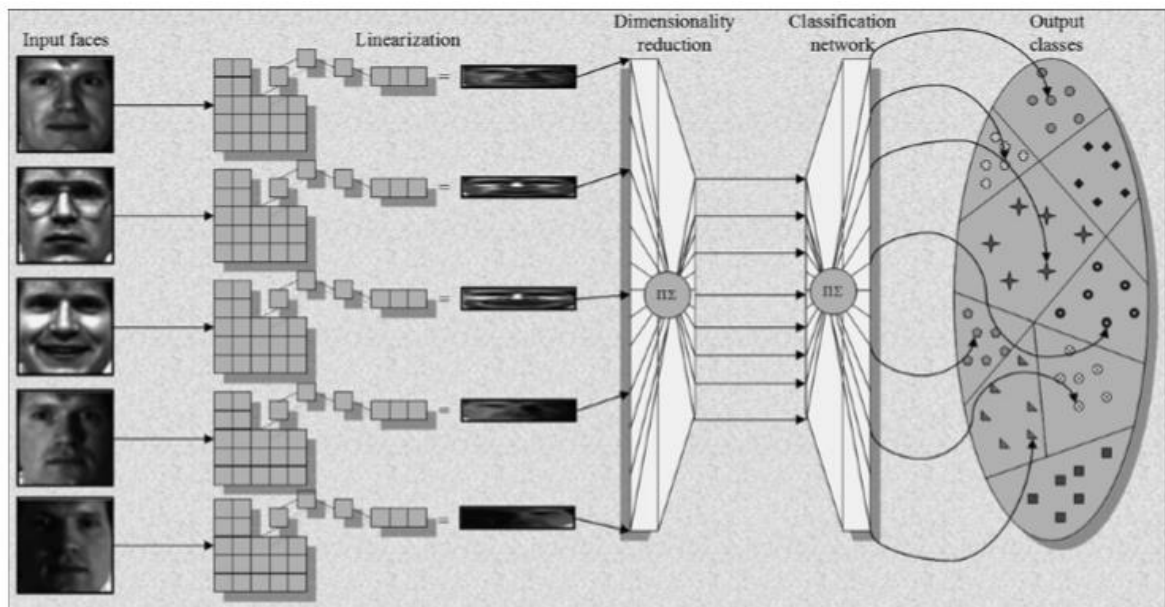


Figura 3 Método usando una red neuronal para reconocimiento facial

Otra aproximación, fue una solución híbrida (Meng, et al., 2002) entre la técnica PCA y la red neuronal. A través de la técnica PCA, se obtuvieron los valores más discriminantes. Estos valores se utilizaron como datos de entrada en la red neuronal.

Finalmente, este método no se utiliza como tal, más bien se usa siempre como un sistema híbrido con los restantes, en donde la introducción de los distintos vectores características se emplea para la identificación de las personas.

2.1. Reconocimiento basado en sistemas 2D

2.1.3. Filtros Gabor

Los filtros Gabor son una herramienta muy poderosa porque tienen la capacidad de capturar características visuales, como la localización espacial, frecuencia y la orientación. En la mayoría de los casos se utilizan para extraer las principales características de las imágenes de las caras (Lades, et al., 1993). Después de aplicar estos filtros se realizara la reducción para obtener el vector de características, véase la Figura 4, el cual se usará para el reconocimiento.

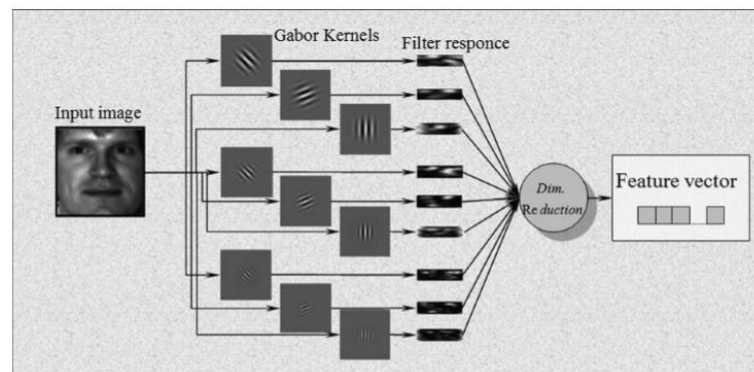


Figura 4 Método de filtros Gabor para el reconocimiento facial

2.1.4. Los fractales y los sistemas iterativos de funciones (IFS)

La teoría IFS consiste en la localización y la extracción trozos de imágenes que son invariantes al escalado y la rotación. Estos trozos de la imagen (fractales) se utilizan en una red neuronal para el reconocimiento de la imagen, los autores del mismo aseguran un 100% con tasa de rechazo de 0 (Riccio & Nappi, 2003).

La principal ventaja de este sistema tiene un menor coste computacional que los anteriores (Abate, et al., 2007).

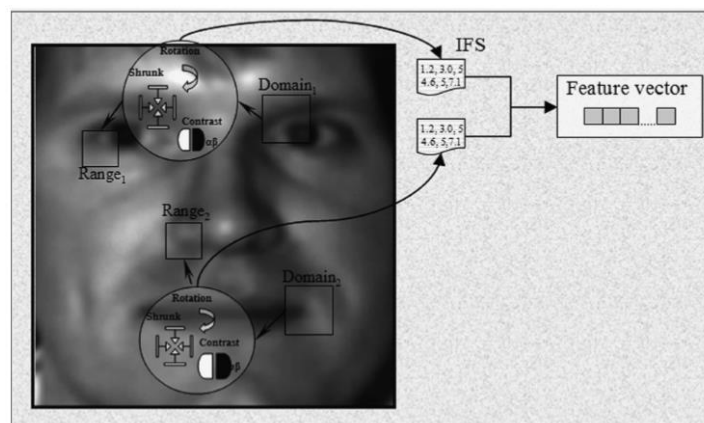


Figura 5 Método IFS para el reconocimiento facial

2.1.5. Termales e hiperspectrales

Estos no utilizan las clásicas imágenes a color, realizan experimentos con otro tipo de cámaras, por ejemplo, una de infrarrojos (Chen, et al., 2003). Estas no se ven afectadas por cambios en la iluminación como las cámaras a color tradicionales. Los algoritmos que se usan en estos experimentos son los mismos que los que se han presentado previamente.

2.2. Reconocimiento basado en sistemas 3D

Las dos principales formas para el modelado de imágenes en el reconocimientos 3D son las 2.5D y las imágenes 3D (Abate, et al., 2007), ejemplo de ello es la Figura 6. Las imágenes 2.5D consisten en una representación una nube de puntos en el plano X-Y, donde los valores son los valores de profundidad del eje Z. Las imágenes 3D pueden presentarse como una malla o una nube de puntos.

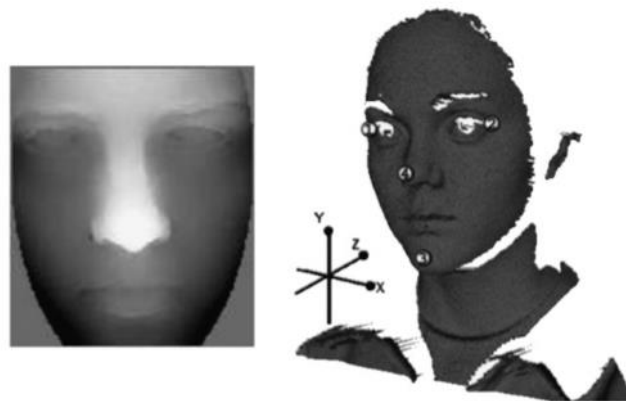


Figura 6 Imagen 2.5D y nube de puntos de una cara

Los inconvenientes es que las imágenes 2.5D (como la imagen de la izquierda de la Figura 6) solo nos muestran un punto de vista, desde un ángulo determinado. Aunque la realización de varias capturas desde distintos ángulos puede dar resultado una imagen 3D.

Uno de los métodos consiste en la comparación de la malla 3D con la de la cara a reconocer mediante el cálculo de las distancias euclídeas entre los puntos (Ansari, et al., 2003). Este estudio uso 26 sujetos y tuvo una tasa de acierto del 96,2% con dos pares de imágenes, una para el entrenamiento y otra para las pruebas.

2.3. Reconocimiento basado en sistemas 2D+3D

Otro método consiste en la extracción de las características de la malla, como por ejemplo son las medidas entre partes de la cara, algunos estudios usando este método llegaron al 100% de tasa de reconocimiento (Chua, et al., 2000).

2.3. Reconocimiento basado en sistemas 2D+3D

Estos sistemas utilizan la combinación de los sistemas 2D y los sistemas 3D, investigaciones (Chang, et al., 2004) proponen una mejora de la biométrica facial 2D integrando la imagen 3D. El método que ellos proponen es usar las técnicas PCA en ambas imágenes y combinar los resultados (ver Figura 7). Los autores obtuvieron 4 conclusiones:

1. Las imágenes 2D y las imágenes 3D tienen similar tasa de reconocimiento cuando son consideradas individualmente
2. Combinando ambos resultados usando un sistema con pesos similares mejora los resultados obtenidos tanto 2D como 3D de manera aislada
3. Combinando 2 o más imágenes 2D se obtienen mejor resultados que cuando solo se usa una sola imagen
4. Combinando 2D+3D se obtienen mejores resultados que cuando se usa 2 o más imágenes 2D

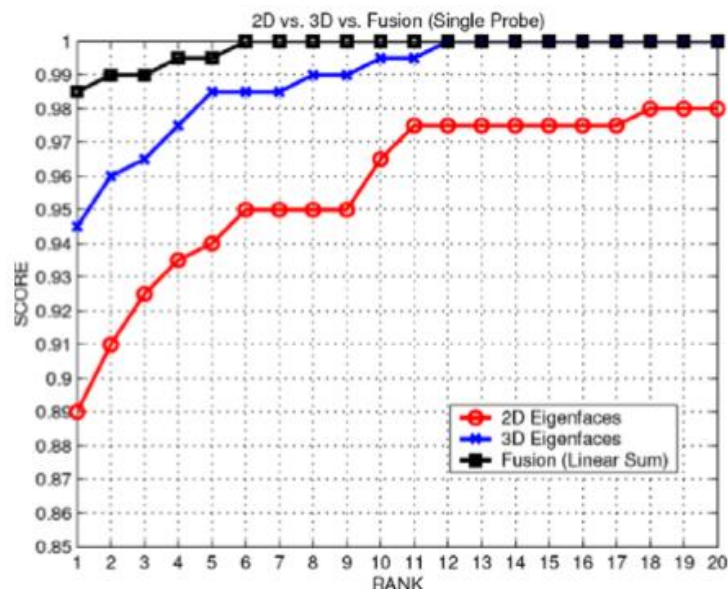


Figura 7 Gráfico de precisión de reconocimiento en 2D, 3D y 2D+3D (Chang, et al., 2004)

2.4. Conclusiones de los sistemas de reconocimiento facial y elección de sistema.

En la Tabla 1 (Abate, et al., 2007) se muestra un resumen y así como la precisión de los distintos métodos.

Tabla 1 Precisión de los distintos metodos de reconocimiento facial

Method		Image/model size	No. subj.	No. images	Recogn. rate (%)	Database characteristics			
Authors	Type					Expr.	Ill.	Pose	Occl.
<i>2D based</i>									
Blanz and Vetter (2003)	2D-based	–	–	4,488 (PIE) + 1,940 (FERET)	95 and 95.9	Yes	Yes	Yes	No
Lu et al. (2004)	2D-based	64 × 64	10	220	85	Yes	Yes	Yes	No
Hu et al. (2004)	2D-based	–	68	41.368	<95	Yes	Yes	Yes	No
<i>3D based</i>									
Ansari and Abdel-Mottaleb (2003)	Set of points	–	26	52	96.2	No	No	No	No
Cook et al. (2004)	GMM (ICP)	4000 points	120	360 (3D_RMA)	97.33	Yes	No	Yes	No
Irfanoglu et al. (2004)	Set of points (ICP)	4000 points	30	90 (3D_RMA)	96.66	Yes	No	Yes	No
Medioni and Waupotitsch (2003)	ICP	–	100	700	98	No	No	Yes	No
Lu et al. (2004)	ICP	320 × 240 (18,000 points)	18	113	95.5	Yes	Yes	Yes	No
Gordon (1991)	Local shape descriptors	–	8	24	100	No	No	No	No
Moreno et al. (2003)	Local shape descriptors	2186 points	60	420 (GavabDB)	78	Yes	Yes	Yes	No
Chua et al. (2000)	Segmentation	–	6	24	100	Yes	No	No	No
Tanaka et al. (1998)	EGI	256 × 256	37	37	100	No	No	No	No
Wang et al. (2004)	Sphere-spin-images	200 × 200	6	31 (SAMPL)	91.68	No	No	Yes	No
Gao and Leung (2002)	PCA-based	10,000 points	37	220	100	Yes	No	No	No
Heseltine et al. (2004a)	PCA-based	–	100	330 (YORK)	<87.3	Yes	No	Yes	No
Heseltine et al. (2004b)	PCA-based	–	280	1770 (YORK)	<88.7	Yes	No	Yes	No
<i>Multimodal</i>									
Chang et al. (2004)	Multimodal	640 × 480	275	–	98.8	Yes	Yes	Yes	No
Bronstein et al. (2003)	Multimodal	2000–2500 points	157	–	–	Yes	No	Yes	No
Tsalakanidou et al. (2003)	Multimodal	571 × 752	50	3000	91.67	Yes	Yes	Yes	Yes
Papatheodorou and Rueckert (2004)	Multimodal	10,000 points	62	~900	66–100	Yes	Yes	Yes	No

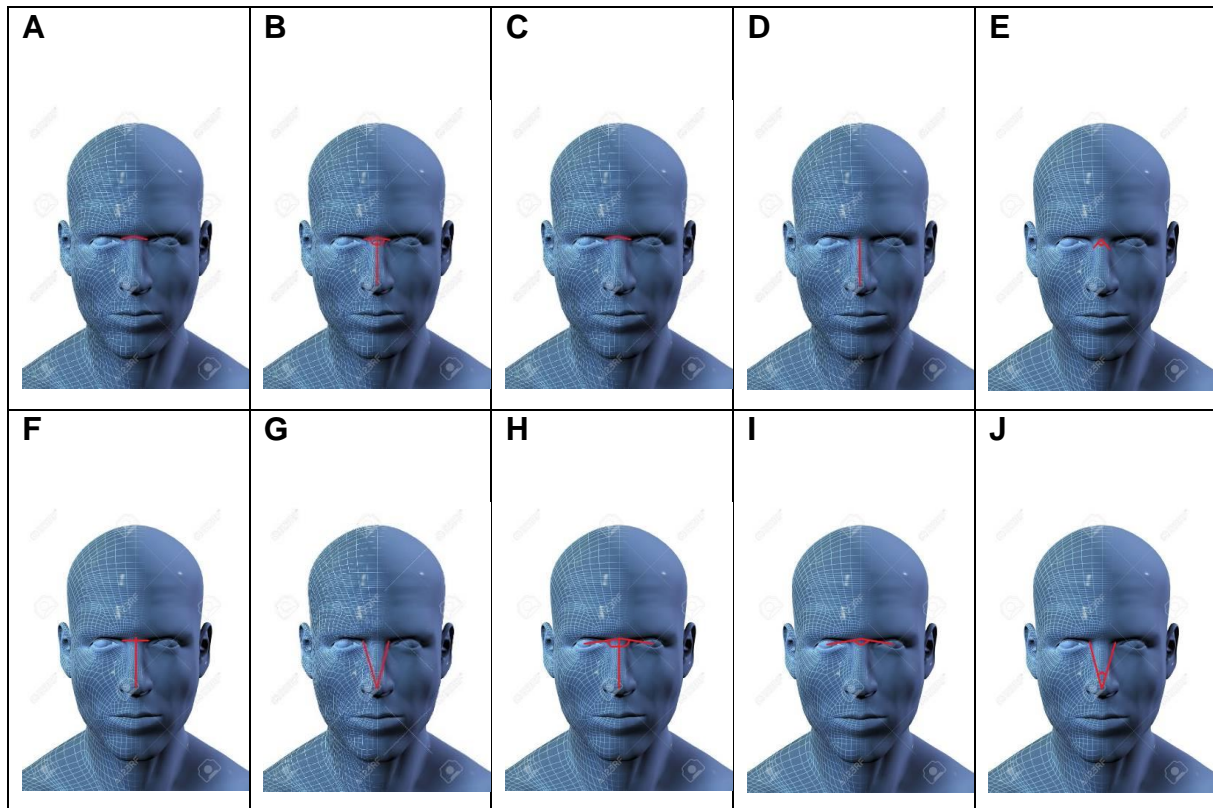
De la tabla anterior, hay que destacar varios estudios con una tasa de reconocimiento del 100% (Chua, et al., 2000), todos ellos realizados con sistemas basados en adquisición de imágenes 3D. Por tanto, para este sistema se va a utilizar un sistema de adquisición 3D al ser bastante fiable y robusto en comparación que los anteriores.

Principalmente, se usan los métodos basados en las características de la cara. Para ello, se va a partir del trabajo de “Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional” (Moreno, 2004) en el cual realiza un estudio de los invariantes más significativos para el reconocimiento facial.

2.4. Conclusiones de los sistemas de reconocimiento facial y elección de sistema.

Según el trabajo previamente comentado, las características más discriminantes, ordenadas según el poder discriminatorio, se presentan en la Tabla 2:

Tabla 2 Características discriminantes que se van a usar en el sistema



- A. Ángulo entre los centros de masas de la zona situada encima de la nariz, zona cóncava que está situada a la derecha del ojo izquierdo y el lado simétrico
- B. Media de los ángulos simétricos que forma la punta de la nariz, la zona superior de la nariz y la zona cóncava de los ojos que está pegada a la nariz
- C. Media de las distancias de la parte superior de la nariz con la zona cóncava de los ojos que está pegado a la nariz
- D. Distancias entre la punta de la nariz y la parte superior de la nariz
- E. Angulo entre extremo superior de la línea del contorno de la nariz, con la parte superior de la nariz y extremo superior del otro lado del contorno de la nariz

- F. Relación entre las distancias punta de la nariz y parte superior con las distancias entre las zonas cóncavas de la cuenca de los ojos pegadas a la nariz
- G. Medias de las distancias entre la punta de la nariz y el punto de la zona cóncava de la cuenca de los ojos pegada a la nariz.
- H. Media de los ángulos simétricos que se forma entre los ojos, la punta de la nariz y la parte superior de la nariz
- I. Ángulos entre los ojos y la parte superior de la nariz
- J. Angulo entre en la zona cóncava de la cuenca del ojo izquierdo pegado a la nariz con la punta de la nariz y con la zona de la otra cuenca del otro ojo

Una vez obtenidas las características más discriminantes, se va a usar una **red neuronal**, utilizado en todos los sistemas para el reconocimiento.

3. DISEÑO DEL SISTEMA DE ADQUISICIÓN

En este apartado se va a estudiar las distintas posibilidades a la hora de realizar el sistema de adquisición.

El funcionamiento debe de ser el siguiente: cuando una persona se sitúe delante de la cámara, ésta debe ser detectada y se procederá a la captura de la imagen. Además, se va a prever de un sistema para informar al usuario que no se mueva durante la captura de la imagen y comunicarle si está en la posición idónea. El sistema será similar a la Figura 8.

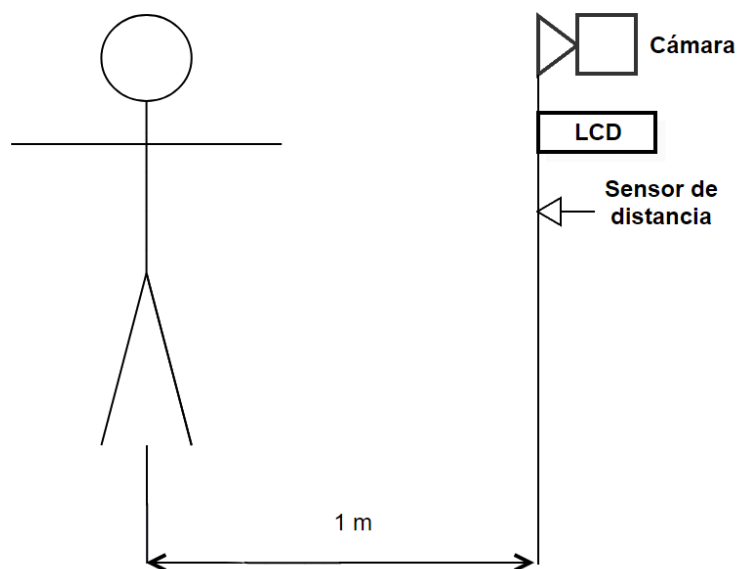


Figura 8 Esquema del sistema de adquisición

Primero se va a presentar el diagrama de flujo básico (Figura 9), más adelante se ampliará para el caso de estudio. En este diagrama de flujo se va a poner indistintamente la fase de reconocimiento y la de entrenamiento del sistema, realmente serían dos aplicaciones totalmente distintas e incompatibles entre ellas.

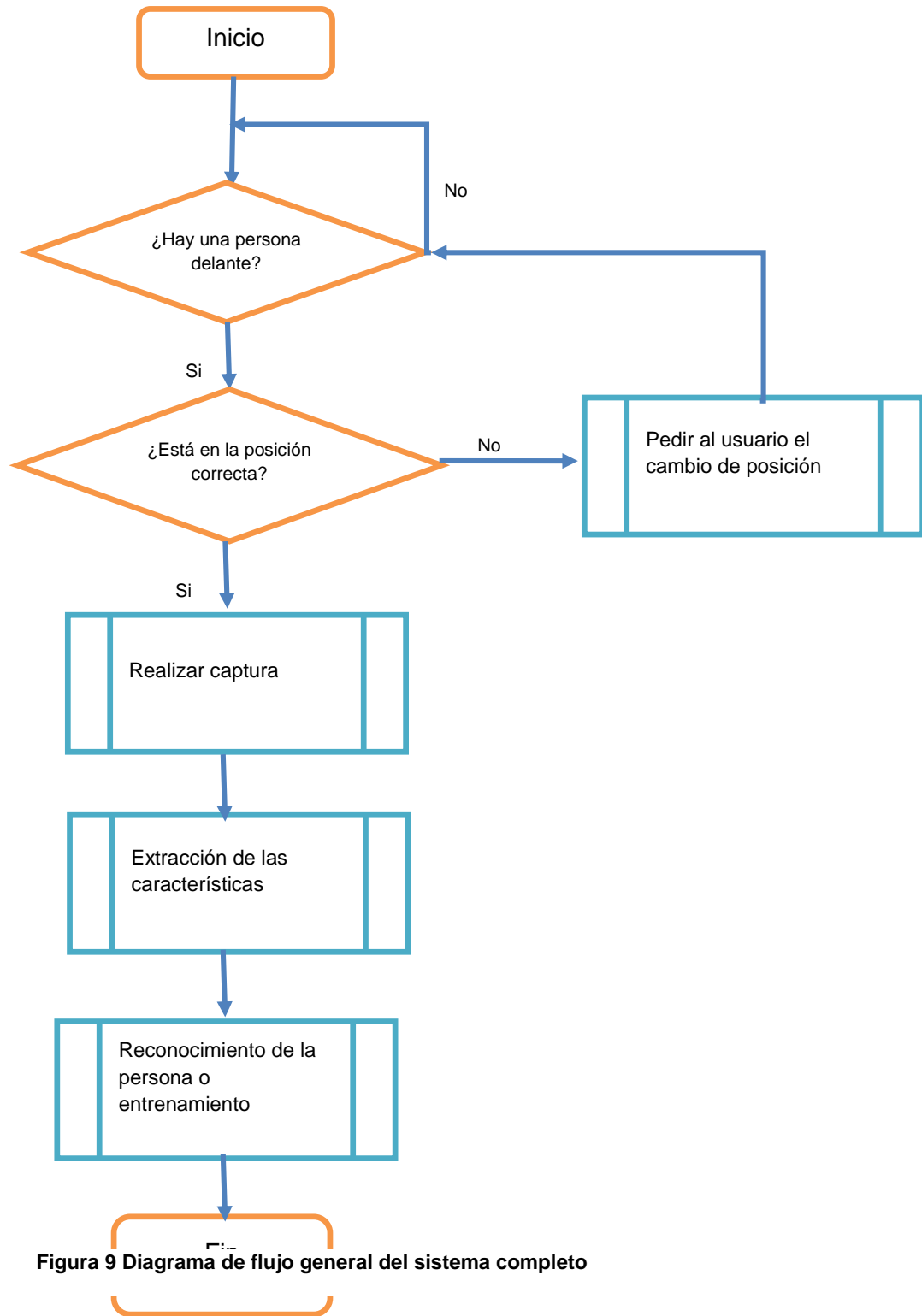


Figura 9 Diagrama de flujo general del sistema completo

3.1. Selección de la cámara

3.1. Selección de la cámara

Lo primero que se va a elegir es el sistema de adquisición de la imagen tridimensional, para ello, se va a buscar entre distintas opciones comerciales. Al final del capítulo se procederá a seleccionar la que se va a usar.

3.1.1.3dMDface System. 3dMD

El primer sistema que se va a analizar es el de 3dMD (3DMD, s.f.). 3dMD es una empresa especializada en escáneres 3D con fines médicos. El modelo de Figura 10 corresponde con el modelo específico para la cara.



Figura 10 Sistema de adquisición 3dMDface

Se caracteriza por ser un sistema que realiza una representación de 180°, cubriendo de oreja a oreja. Está formado por dos módulos de seis cámaras y un flash sincronizado. Sus principales características se expresan en la Tabla 3:

Tabla 3 Características 3dMDface

Precisión	<0,2mm o mejor dependiendo del modo
Tiempo de renderizado	7 segundos
Tiempo de escaneo	0,0015 s
Precio	10.000-20.000€

Otras características:

- Fácil transporte del equipo
- Generado automático de la malla de la cara mediante el controlador incluido en la propia cámara.

3.1.2. Minolta Vivid 910. Minolta

El escáner 3D de Minolta de la Figura 11 es un equipo compacto (Minolta, 2009), con la capacidad de ser cambiado de ubicación sin ningún impedimento, su principal utilidad es como equipo para realizar mediciones de ingeniería inversa.



Figura 11 Sistema de adquisición Minolta Vivid 910

En la Tabla 4 se presentan las características que se van a usar para el estudio:

Tabla 4 Características Minolta Vivid 910

Precisión	0,1mm
Tiempo de renderizado	0 segundos
Tiempo de escaneo	0,3-2,5s
Precio	25.000-50.000€

Además, tiene otras especificaciones que son importantes de destacar:

- Resolución: 307.000 pixeles en FINE mode: 76.800 pixeles en FAST mode
- Tarjeta de memoria
- Accesorios para ajustar la lente según el espacio
- Distancia óptima de medida: 0,6 a 1,2 m

3.1. Selección de la cámara

3.1.3. Artec EVA. Artec

El Artec EVA (Artec, s.f.) de la Figura 12 es un escáner de mano con bastante empleo en la industria debido a su precio y a su portabilidad. Artec es una empresa de escáneres 3D con gran prestigio que lleva trabajando desde 2.008.



Figura 12 Sistema de adquisición Artec EVA

Es un equipo que no necesita de calibración previa ni tiempo para preparación. En la Tabla 5 se ven las principales características:

Tabla 5 Características Artec EVA

Precisión	0,5mm
Tiempo de renderizado	0 segundos
Tiempo de escaneo	0,0002s
Precio	13.000€

Otras características:

- Distancia de trabajo 0,4-1m
- Área de escaneo en el rango más lejano 536x371mm.
- Captura de imagen a color
- Distancia de trabajo: 0,4-1m
- Requerimientos del pc:
 - Procesador: Intel i5 o i7
 - Memoria: 12GB
 - USB 2.0

3.1.4. HDI Advance R1X. LMI Technologies

El modelo de la Figura 13 es el más básico de LMI (Polyga, 2017), formado por dos cámaras de 1,3 megapíxeles. LMI es una empresa que lleva aportando productos desde 1970, que adquirió en 2014 3d3 Solutions, empresa encargada en la realización de los escáneres 3D. Actualmente su desarrollo y mantenimiento ha sido transferido a la empresa polyga.



Figura 13 Sistema de adquisición HDI Advance R1X

En la Tabla 6 se presentan las principales características:

Tabla 6 Características HDI Advance R1X

Precisión	0,1mm
Tiempo de renderizado	0 segundos
Tiempo de escaneo	1,3s
Precio	17.000€

Otras características:

- Campos de visión (diagonal en mm): 165, 310, 455
- Resultados de la nube de puntos: 1,1 millones/escaneo
- Número de polígonos de la malla: 2,2 millones/escaneo
- Cámara a color
- Tamaño compacto para el transporte

3.1. Selección de la cámara

3.1.5. Kinect 2. Microsoft

Inicialmente el dispositivo Kinect (Microsoft, s.f.) como el de la Figura 14 está pensado para jugar con la consola XBOX, pero dada sus posibilidades se decidió dar soporte para ordenador. Esto propició el desarrollo de aplicaciones para escritorio. Actualmente, esta es la segunda versión del dispositivo.



Figura 14 Sistema de adquisición Microsoft Kinect

La hoja de características se presenta en la Tabla 7:

Tabla 7 Características Microsoft Kinect

Precisión	1 mm a la distancia de 1m
Tiempo de renderizado	0 segundos
Tiempo de escaneo	0,03s
Precio	100€ cámara + 50€ adaptador

Otras características son:

- Campo de visión 70° en horizontal y 60° en vertical
- Rango de profundidad: 0,5 a 4,5 m
- Dispone de SDK propio con capacidad para seguimiento de personas
- Incorpora micrófono
- USB 3.0

3.1.6. Xtion PRO LIVE. Asus

Dispositivo creado por Asus (Anon., s.f.), como el de la Figura 15, para competir contra Kinect, no está muy extendido debido a la popularidad del anterior. Asus ha sido capaz de meter la tecnología en diferentes portátiles. Es un equipo de tamaño parecido a Kinect, que salió en 2011. Los sensores fueron desarrollados por PrimeSense, los mismos que diseñaron los sensores de la primera versión de la Microsoft Kinect.



Figura 15 Sistema de adquisición Xtion PRO LIVE

Sus características se ven en la Tabla 8:

Tabla 8 Características Xtion PRO LIVE

Precisión	2 mm
Tiempo de renderizado	0 segundos
Tiempo de escaneo	0,03s
Precio	355€

Otras características:

- Rango de profundidad: 0,8 a 3,5 m
- SDK disponible con capacidad de reconocimiento facial
- USB 2.0
- Micrófonos
- Cámara a color

3.1. Selección de la cámara

3.1.7. Selección de la cámara

Para la selección de la cámara se utilizara el método de selección multicriterio PROMETHEE (Brans, 1982). El método PROMETHEE se basa en el uso de criterios normales, de distintos tipos, para establecer las relaciones de superación entre varias alternativas.

En la Tabla 9 se va a establecer son las medidas a comparar y el grado de importancia que se le va a dar, evaluándolo de 1 a 10.

Tabla 9 Tabla de medidas y pesos para la selección

Escáner	Tiempo escaneo (s) Peso 6	Tiempo recarga (s) Peso 2	Coste (€) Peso 10	Precisión profundidad (mm) Peso 8
3dMD	0,0015	7	15000	0,2
Minolta	2,5	0	30000	0,1
Artec EVA	0,063	0	13700	0,5
HDI Advance	1,3	0	17000	0,3
Kinect 2	0,03	0	150	1
Xtion PRO LIVE	1	0	355	2

Siguiendo el método, lo próximo que hay que realizar es normalizar la matriz de decisión. Para ello, primero se normalizan los pesos, en donde vasta dividir cada uno de ellos por la suma de todos los pesos. Para obtener el resto y teniendo en cuenta que el valor de mayor utilidad es el mínimo, lo que hay que hacer en cada columna es buscar el mínimo y dividirlo por el correspondiente. Viéndose el resultado final de este proceso en la Tabla 10:

Tabla 10 Tabla de medidas y pesos normalizados para la selección

Escáner	Tiempo escaneo (s) Peso 0,2307	Tiempo recarga (s) Peso 0,0769	Coste (€) Peso 0,3846	Precisión profundidad (mm) Peso 0,3076
3dMD	1	0	0,01	0,5
Minolta	0,0006	1	0,005	1
Artec EVA	0,023809524	1	0,01094891	0,2
HDI Advance	0,001153846	1	0,00882353	0,333333333
Kinect 2	0,05	1	1	0,1
Xtion PRO LIVE	0,0015	1	0,42253521	0,05

3. Diseño del sistema de adquisición

Una vez realizado esto, se calculan los índices de preferencia, tal y como explica el método, en donde se van a comparar los distintos valores.

Tabla 11 Tabla de selección PROMETHEE

	3dMD	Minolta	Artec EVA	HDI Advance	Kinect 2	Xtion PRO LIVE	Flujos +
3dMD	0,0000	0,6154	0,5385	0,9231	0,5385	0,5385	3,1538
Minolta	0,3846	0,0000	0,3077	0,9231	0,3077	0,3077	2,2308
Artec EVA	0,4615	0,6154	0,0000	0,3077	0,3077	0,5385	2,2308
HDI Advance	0,0769	0,6154	0,3077	0,0000	0,3077	0,3077	1,6154
Kinect 2	0,4615	0,6154	0,6154	0,6154	0,0000	0,9231	3,2308
Xtion PRO LIVE	0,4615	0,6154	0,3846	0,6154	0,0000	0,0000	2,0769
Flujos-	1,8462	3,0769	2,1538	3,3846	1,4615	2,6154	

Se selecciona el de menor valor de la fila Flujos- (1,4615) de la Tabla 11 y el de mayor valor de la columna Flujos+ (3,2308) de la misma tabla. El resultado obtenido es Kinect 2 de Microsoft.

A partir de este momento se va a dimensionar el resto de herramientas con el fin de que se adapte de la mejor manera a la cámara seleccionada. Los principales aspectos que se van a tener en cuenta:

- Medida óptima a 1 m de distancia
- Campo de visión
- Ordenador con USB 3.0 y características mínimas para su completo funcionamiento
- SDK para C++ o .NET (aunque como se verá más adelante se ha podido adaptar completamente para ser usado en Matlab).

3.2. Sensores posición de personas

Como se ha expuesto al principio del apartado 3, se necesita un sistema para detectar la presencia de una persona delante del mismo. Se van a estudiar distintas opciones y finalmente se elegirá la más adecuada. Los tipos de sensores que permiten medir distancias (Pérez García, et al., 2003):

- **Sensor inductivo**

Se basan en el efecto que las diversas variables físicas ocasionan o pueden ocasionar sobre los parámetros que afectan a los valores de auto-inductancia o de acoplamiento magnético.

El sensor magnético de proximidad de tipo inductivo es uno de los más empleados en la detección de proximidad. Su funcionamiento se basa en las modificaciones sobre la distribución del campo magnético ocasionadas por la presencia de objetos metálicos y en el correspondiente cambio en el valor de la inductancia de la bobina.

Estos sensores tienen un alcance de 0,1 a 20 cm y solo sirve para objetos metálicos. Tienen un alcance selectivo

- **Sensor capacitivo**

Basan su funcionamiento en la variación de la capacidad entre dos o más conductores entre los que se encuentra un dieléctrico, en respuesta a la variación de alguna magnitud física.

Detectan la presencia de objetos próximos, tanto metálicos como no metálicos sin requerir contacto directo entre ellos. Cuando no se tiene presente un objeto, la capacidad del sensor es baja y, por tanto, la amplitud de la oscilación es pequeña. Cuando el objeto se aproxima a la cabeza sensora se incrementa la capacidad y, con ello, la amplitud de la oscilación, provocando la conmutación del circuito de salida del detector. Normalmente tienen un alcance de 0,1 a 3 cm. Tiene un alcance selectivo.

- **Sensor ultrasónico**

El funcionamiento de estos sensores consiste en emitir un pulso de ultrasonido y se escuchan los ecos; el tiempo transcurrido entre el pulso emitido y el eco recibido multiplicado por la velocidad de propagación en ese medio nos informa del doble de la distancia que separa el emisor del objeto que produjo el eco.

El alcance de estos sensores y la amplitud dependen de la forma del cono, aumentando el precio al aumentar la distancia y hacerlo más selectivo.

- Sensor láser

Es un tipo de detector de proximidad óptico, está compuesto de una fuente de luz y un fotodetector. El láser y el fotodetector están montados sobre la misma cabeza de forma que en presencia de un objeto parte de la luz procedente del láser se refleja en el mismo y alcanza el fotodiodo. El alcance máximo de estos detectores suele ser de unos 2 metros. Son muy selectivos.

Vistas las distintas opciones, se debe elegir entre el sensor láser y el sensor ultrasónico dado que la distancia óptima es de 1 metro. El sensor ultrasónico puede permitir distancias mayores y una buena selectividad con un cono adecuado. El inconveniente es que el modelo concreto para este caso tiene un precio 13 veces superior que un sensor láser. El coste del ultrasónico es de 130€ frente a los 10€ del sensor láser. Debido a la diferencia de precio se va a elegir el **sensor láser**.

4. HARDWARE DEL SISTEMA

En este apartado se van a ver la selección de los distintos apartados del sistema, se realizará la selección del sensor láser, el sistema de comunicación de este con el ordenador, el display para informar al usuario y la estructura que va soportar todo el equipo.

4.1. Sensor láser para la posición del usuario

El sensor elegido es VL53L0X fabricado por ST (STMicroelectronics, s.f.). Es un sensor definido como el más pequeño Time-of-Flight (ToF) ranging sensor. La luz emisora del láser es invisible para el ojo humano y seguro según el estándar IEC 60825-1:2014. En la Figura 16 se ve el micro controlador embebido junto al sensor.



Figura 16 Sensor VL53L0X

Sus características:

- Medidas en un rango de hasta 2 m
- Medida independiente de la reflectancia del objeto
- Capaz de operar en niveles de luz ambiental con altos niveles de infrarrojos
- No precisa de óptica adicional
- Medidas: 4,4 x 2,4 x 1 mm
- Protocolo de comunicación I²C
- SDK preparado para la obtención de las medidas sencillamente

El modelo comprado (Adafruit, s.f.) viene preparado para ser conectado (Figura 17):



Figura 17 Placa adquirida para la medida de la distancia

Los pines que se ven en la son:

- VCC: Puerto de alimentación, el chip está diseñado para usarlo a 2,8V, pero la placa comprada esta prepara para alimentarse a 3 o 5 V
- GND: Puesta a tierra de la placa
- SCL: Pin de señal del reloj I²C
- SDA: Pin de datos de I²C
- GPIO2: Es un pin de control que usa el sensor para indicar que los datos están listos
- XSHUT: Es el pin de apagado del sensor

4.2. Display para informar al usuario

Para informar al usuario de si está en la posición correcta o si se tiene que estar quieto para que le tome la imagen se va a usar una pantalla LCD. Debido a que se va a utilizar un Arduino, se va a elegir las pantallas usualmente utilizadas, debido a su bajo coste y al gran desarrollo que hay disponible.

La pantalla elegida es Pantalla LCD 16x2 1602 (Figura 18), se ha elegido esta pantalla porque tiene dos filas de caracteres, esto puede ser interesante para futuras ampliaciones o mostrar información adicional.



Figura 18 Display LCD 16x2

4.3. Sistema de conexión del láser y el display con el ordenador

Una vez seleccionada la pantalla y el sensor láser se debe de buscar la manera más idónea y económica para comunicarse con el ordenador.

El dispositivo debe de contar con 7 patillas de entradas/salidas digitales para la comunicación con la pantalla, también debe de contar con capacidad para establecer la comunicación I2C con el sensor láser.

Para esta tarea se debe de usar un microcontrolador tipo PIC o similar, se va a elegir una placa de desarrollo tipo Arduino al ser más versátil y facilitar el conexionado.

La placa elegida es una Arduino UNO (Arduino, 2017), la principal razón se debe a que es el modelo más económico. El arduino UNO utiliza el microcontrolador ATmega328P. Va a ser alimentado a través del puerto USB porque se va a utilizar la comunicación serie del mismo para comunicarse con el programa.

El conexionado de los componentes con el arduino se muestra en la Figura 19. Se tiene que instalar un potenciómetro de 10kΩ para ajustar el contraste de la pantalla LCD.

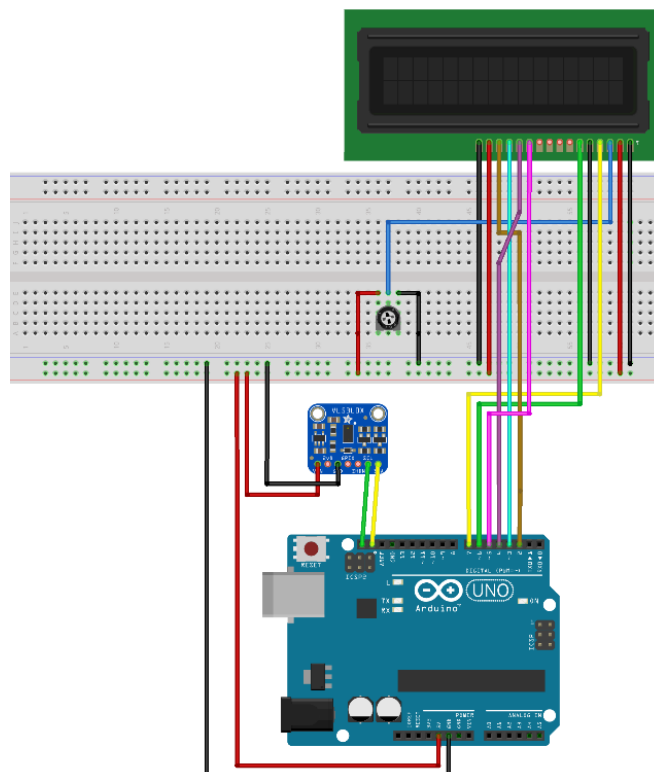


Figura 19 Conexionado de todos los componentes con la placa arduino

4.4. Diseño del prototipo

Una vez seleccionado el equipamiento se va a proceder al diseño óptimo de la solución. Primero se tendrá en cuenta las especificaciones de la cámara, debido a que su colocación va a ser fundamental para el sistema de adquisición.

Como se ha expuesto anteriormente, el Angulo de captura de la cámara es de 60 grados y la distancia ideal es de un metro. Según el informe de (OECD, 2009) la altura media de un hombre español es de 174 cm y el de una mujer española de 163 cm, aunque para el reconocimiento, la altura idónea es a nivel de la nariz, por lo que la cámara se va a instalar a una altura de 165 cm.

La pantalla se va a colocar justo debajo de la cámara con el fin de que la cara del usuario este orientada en la dirección de la cámara. Con esta disposición se van a evitar posibles medidas incorrecta y ralentización del flujo de personas.

El sensor láser se va a colocar justo debajo de la pantalla, con el fin de localizar a cualquier persona y minimizar los posibles errores en la medida. Además, este diseño permitirá ahorrar en tamaño y en plástico.

Como soporte se va a utilizar un trípode, de medidas 35 mm en el diámetro del tubo superior. Se diseñará la pieza de tal manera que se pueda introducir en él. También se hará teniendo en cuenta de poder ser usada de manera aislada durante el periodo de pruebas y estudios, por lo que tendrá que tener una capacidad autoportante.

La pieza se va a fabricar en plástico con una impresora 3D, se realizará en dos piezas. La primera de ellas estará diseñada de tal manera que soporte la placa de arduino, la pantalla y el sensor láser. La otra pieza tendrá como función soportar la cámara, la pieza anterior y una base para poder soportar el peso de todo.

Como se comentó en el apartado anterior, la pieza que va a estar en la parte superior es la pantalla LCD.

4.4. Diseño del prototipo

Realizando las medidas con el pie de rey se obtiene que el diámetro de los agujeros es de 3 mm, horizontalmente hay una distancia de 75 mm entre centro de agujeros y verticalmente hay una distancia 31 mm entre centro de agujeros.

La pieza que se va a poner debajo es el sensor láser, que volviendo a medir tiene un diámetro de 3mm y una distancia entre centros de 20 mm

Finalmente, se situará en la parte inferior el arduino, el cual posee su propia base para poder diseñar el sistema de anclaje.

Una vez tomadas las medidas, se va a diseñar la pieza con la forma de la Figura 20, dado que el criterio es que tenga un soporte autoportante, se va a optar por realizar una base superior donde se va a apoyar la cámara, y el resto de componentes irán colocados en los pilares que sujetan dicha base.

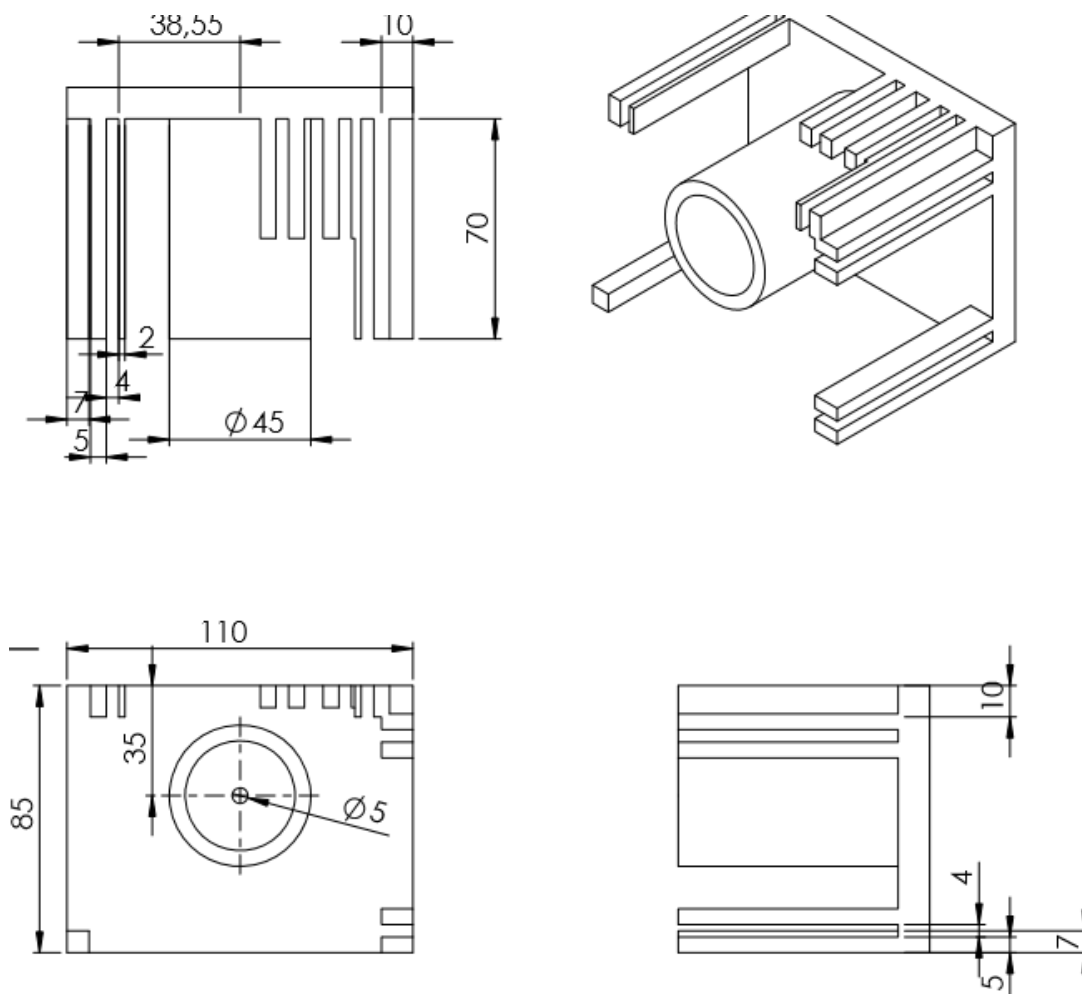


Figura 20 Soporte para el sistema de adquisición

El resultado final montado de todo el conjunto sin la cámara montada puede verse en la Figura 21 Soporte finalizado para la cámara:

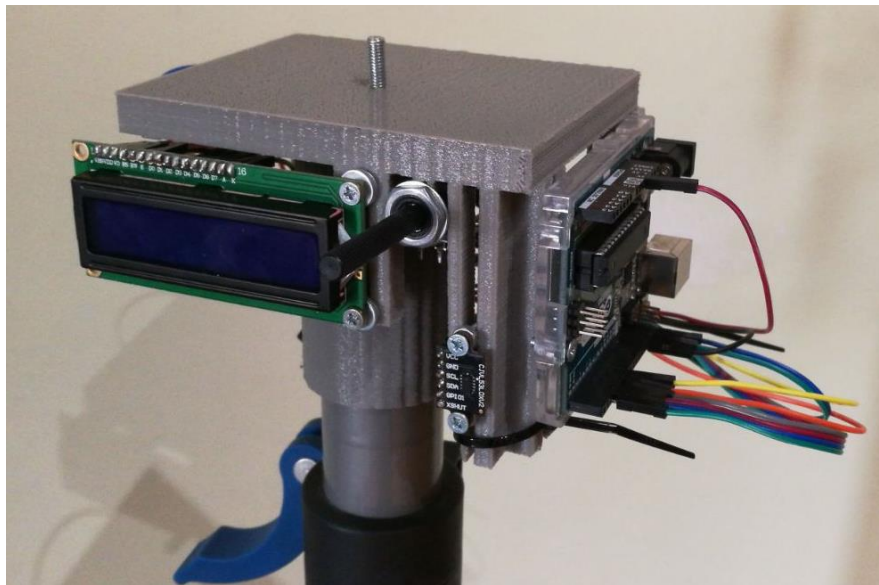


Figura 21 Soporte finalizado para la cámara

5. SOFTWARE DEL SISTEMA

Como se ha visto en los apartados anteriores, las principales funciones del software del proyecto son las siguientes:

- Detección de la posición del usuario
- Informar al usuario mediante la pantalla LCD
- Obtención de la imagen
- Cálculo de los puntos característicos
- Entrenamiento de la red neuronal
- Identificación a través de la red neuronal

En este apartado se van a proponer las soluciones para cada una de las distintas funciones que se deben de realizar.

Se va a utilizar el entorno de Matlab para realizar el software porque es sencillo y tiene todas las herramientas para la elaboración del sistema:

- Toolbox Neural Network: utilidad para la creación de redes neuronales
- Guide: herramienta para crear interfaces gráficas con el fin de controlar aplicaciones de manera sencilla por cualquier persona
- Paquete de soporte Arduino: permite la conexión entre Matlab y Arduino
- Paquete para la adquisición de imágenes con Kinect: con él se puede hacer el control de todas las funcionalidades de Kinect en Matlab

5.1. Arduino en Matlab

Matlab incluye un paquete para poder controlar Arduino desde la versión 2012a, este paquete permite a Matlab comunicarse con la placa Arduino a través del cable USB. Está basado en un programa servidor ejecutándose sobre la placa, que escucha los comandos que llegan a través del puerto serie, ejecuta los comandos y, si es necesario, devuelve un resultado (Matlab, 2017).

Además, Matlab tiene de una herramienta llamada **custom add-on library** que permite la comunicación entre código C++ ejecutado en la placa Arduino y código Matlab ejecutado en el ordenador.

La Figura 22 muestra la construcción de una librería add-on:

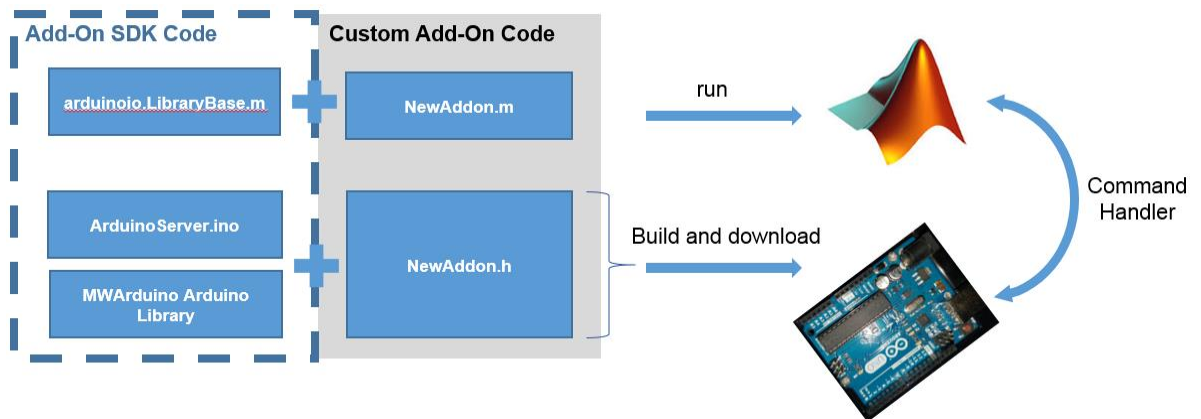


Figura 22 Funcionamiento de custom add-on library

El concepto Command Handler que se incluye en el esquema es la comunicación entre el dispositivo Arduino y el código Matlab. La función *sendCommand* manda un comando desde Matlab, que es el dispositivo que actúan como cliente. El dispositivo Arduino recibe el comando en el archivo de cabecera de C++. La función de intercambio determina que segmento de código se ejecuta en el dispositivo

5.2. Detección de la posición de usuario

Esta función la desarrolla el sensor láser, se desarrollará conjuntamente con la pantalla LCD, a la cual se le mandará los datos para que pueda informar al usuario la posición correcta. Además, una vez este en la posición debe informar a la cámara para que realice la captura.

El sensor cuenta con su propia API desarrollado por STElectronics para C++ (STMicroelectronics, s.f.), aunque ha sido adaptado para usarlo en Matlab mediante una *custom add-on library* (Courane, 2017).

5.2. Detección de la posición de usuario

5.2.1. Diagramas de Flujo

Estos flujogramas (Figura 23 y Figura 24) son simplificaciones, la intención es dar una idea del funcionamiento del mismo. Las partes de información en la pantalla se verán en 4.2.

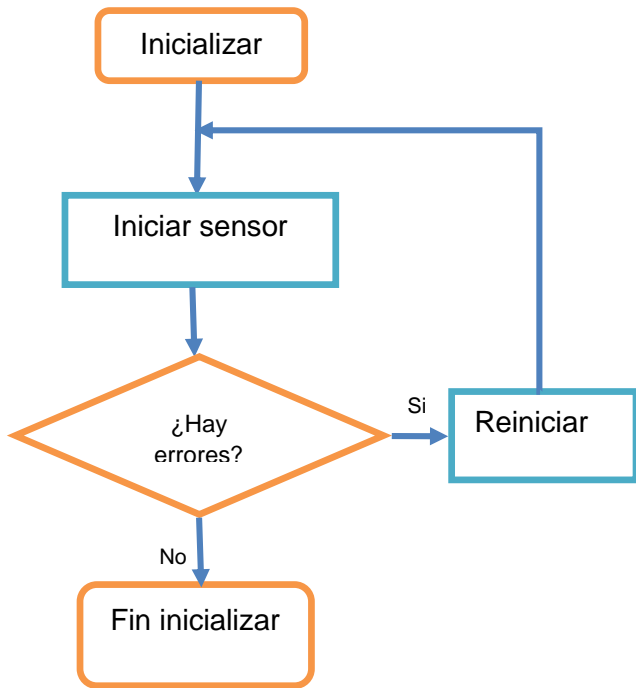


Figura 24 Inicialización del sensor

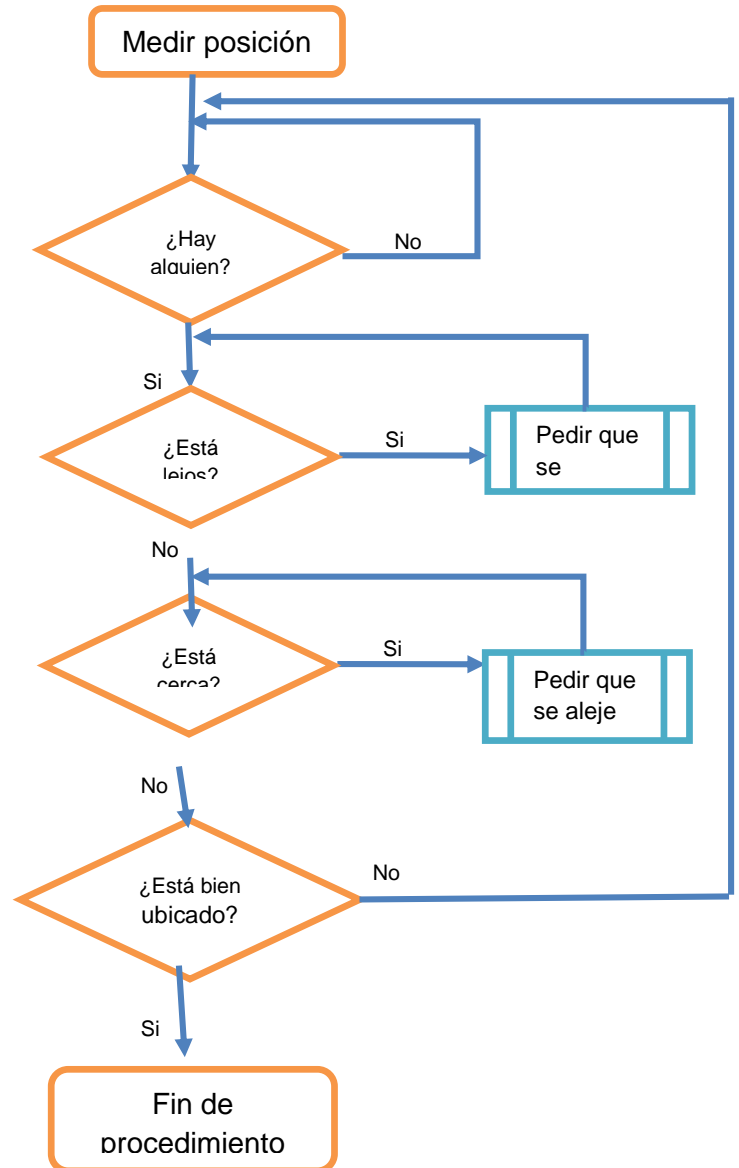


Figura 23 Función para medir la posición

Programación

El estándar I2C (Inter-Integrated Circuit) fue desarrollado por Philips en 1982. Se utiliza principalmente internamente para la comunicación entre diferentes partes de un circuito, por ejemplo, entre un controlador y circuitos periféricos integrados (Anon., 2017).

La interfaz I2C utiliza dos señales: serial data line (SDA) y serial clock line (SCL). Cada dispositivo es conectado al bus usando una única dirección y una simple relación maestro/esclavo.

Para inicializar el arduino, las librerías y el sensor se deben introducir los siguientes comandos:

```
a =
arduino('COM3', 'Uno', 'Libraries', 'Adafruit/VL53L0X', 'ExampleLCD/LCDAddon')

v = addon(a, 'Adafruit/VL53L0X')

begin(v);
```

Para medir los datos con el sensor:

```
rangeMilliMeter(v)
```

Lo primero que se realiza es llamar al espacio de trabajo la placa arduino, para ello se llama a la función arduino. Hay que definir el puerto en el que está ubicado, en el caso de este sistema es el puerto COM. También hay que especificar el tipo de placa (Uno) y las librerías que se deben de cargar, para la pantalla es 'ExampleLCD/LCDAddon' y para el sensor es 'Adafruit/VL53L0X'.

Una vez cargado el arduino hay que asignarle un valor al Add-on (capítulo 5.1), esto se realiza con la función addon.

El caso del sensor hay que realizar una inicialización previa en la que se definirá el tipo de medición, en el caso del proyecto, consiste en devolver solo un valor cuando se solicita la medida con la función rangeMilliMeter. Hasta que recibe la señal, el sistema se mantiene en espera.

5.3. Informar al usuario mediante la pantalla LCD

Se va a usar la pantalla para informar al usuario de si su posición es correcta, que se quite las gafas y de cuando se realiza la captura

Lista de mensajes a mostrar

Mensaje	Función que manda el mensaje
Por favor acérquese a la cámara	Medir posición
Por favor, aléjese de la cámara	Medir posición
Por favor, no se mueva	Obtención de puntos
Captura realizada, muchas gracias	Obtención de puntos

5.3.1. Diagramas de Flujo

Estos flujogramas son simplificaciones, la intención es dar una idea del funcionamiento del mismo.

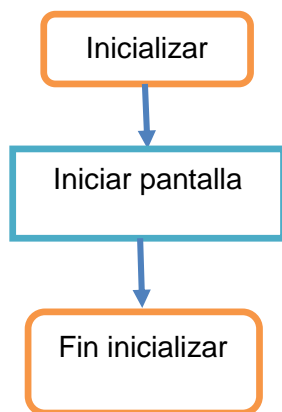


Figura 25 Inicialización de la pantalla

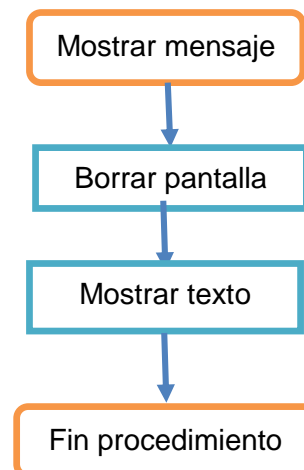


Figura 26 Función para mostrar mensaje

5.3.2. Programación

Funcionamiento de la pantalla

Las pantallas de cristal líquido LCD están pensadas para representar cualquier carácter. Está formado por una matriz de 16 caracteres por línea y dos líneas.

La pantalla dispone de una memoria interna donde se almacena una tabla con todos los caracteres que pueden ser visualizados. Cada uno de estos tiene su representación binaria de 8 bits. Para visualizar cada uno de ellos tiene que recibir por el bus de datos el código correspondiente. Por ejemplo:

"A" → b'01000001'

Table 4 Correspondence between Character Codes and Character Patterns (ROM Code: A00)

Lower 4 bits	Upper 4 bits	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001	1010	1011	1100	1101	1110	1111	
xxxx0000	CG RAM (1)			0	1	P	\	P					-	夕	Ξ	α	p	
xxxx0001	(2)		!	1	A	Q	a	q					。	ア	チ	△	ä	q
xxxx0010	(3)		"	2	B	R	b	r					「	イ	ツ	×	β	θ
xxxx0011	(4)		#	3	C	S	c	s					」	ウ	テ	ε	∞	
xxxx0100	(5)		\$	4	D	T	d	t					、	エ	ト	†	μ	Ω
xxxx0101	(6)		%	5	E	U	e	u					・	オ	ナ	1	σ	Ü
xxxx0110	(7)		&	6	F	V	f	v					ヲ	カ	ニ	ヨ	ρ	Σ
xxxx0111	(8)		'	7	G	W	g	w					フ	キ	ヌ	ラ	g	π
xxxx1000	(1)		<	8	H	X	h	x					イ	ク	ネ	リ	フ	×
xxxx1001	(2)		>	9	I	Y	i	y					ウ	ケ	ル	リ	ウ	
xxxx1010	(3)		*	:	J	Z	j	z					エ	コ	ハ	レ	j	〒
xxxx1011	(4)		+	;	K	C	k	c					オ	サ	ヒ	ロ	*	斤
xxxx1100	(5)		,	<	L	¥	l	l					ハ	シ	フ	フ	Φ	円
xxxx1101	(6)		-	=	M	J	m	}					ユ	ス	ハ	ン	も	÷
xxxx1110	(7)		.	>	N	^	n	→					ヨ	セ	ホ	ッ	ñ	
xxxx1111	(8)		/	?	O	_	o	←					ッ	ソ	マ	°	ö	■

Note: The user can specify any pattern for character-generator RAM.

Además, dispondrá de comandos de control para la configuración del mismo, con funciones como la de apagado del display, borrado de pantalla...

5.3. Informar al usuario mediante la pantalla LCD

Aunque en nuestro caso se utilizará una librería que va a traducir los mensajes directamente para que los represente. La custom add-on library ha sido suministrada directamente por Matlab.

Para inicializar el arduino, las librerías y el sensor se deben introducir los siguientes comandos:

```
a =  
arduino('COM3', 'Uno', 'Libraries', 'Adafruit/VL53L0X', 'ExampleLCD/LCDAddon')  
  
lcd = addon(a, 'ExampleLCD/LCDAddon', {'D7', 'D6', 'D5', 'D4', 'D3', 'D2'});  
  
initializeLCD(lcd);
```

Para mandar los mensajes:

```
printLCD(lcd, 'Introducir mensaje');
```

Para borrar la pantalla:

```
clearLCD(lcd);
```

Como el caso anterior hay que cargar el arduino y el add-on. Para cargar el addon hay que definir que patillas de la placa arduino se utilizan, en el caso del sistema se van a usar las correspondientes a la Figura 19.

Una vez cargado al espacio de trabajo la pantalla se procede a inicializar la pantalla como en el diagrama de flujos con la función initializeLCD.

Para mandar el mensaje se usa la función printLCD, la cual solo puede tener 16 caracteres. Primero se escribe en la línea superior, la siguiente vez que se mande un mensaje irá a la segunda línea.

Para limpiar la pantalla se usa la función clearLCD.

5.4. Obtención de la imagen

Esta función utiliza principalmente el dispositivo Kinect y secundariamente la pantalla para informar al usuario.

5.4.1. Diagramas de Flujo

La Figura 27 muestra el diagrama de flujo de la obtención de los puntos de la cara.

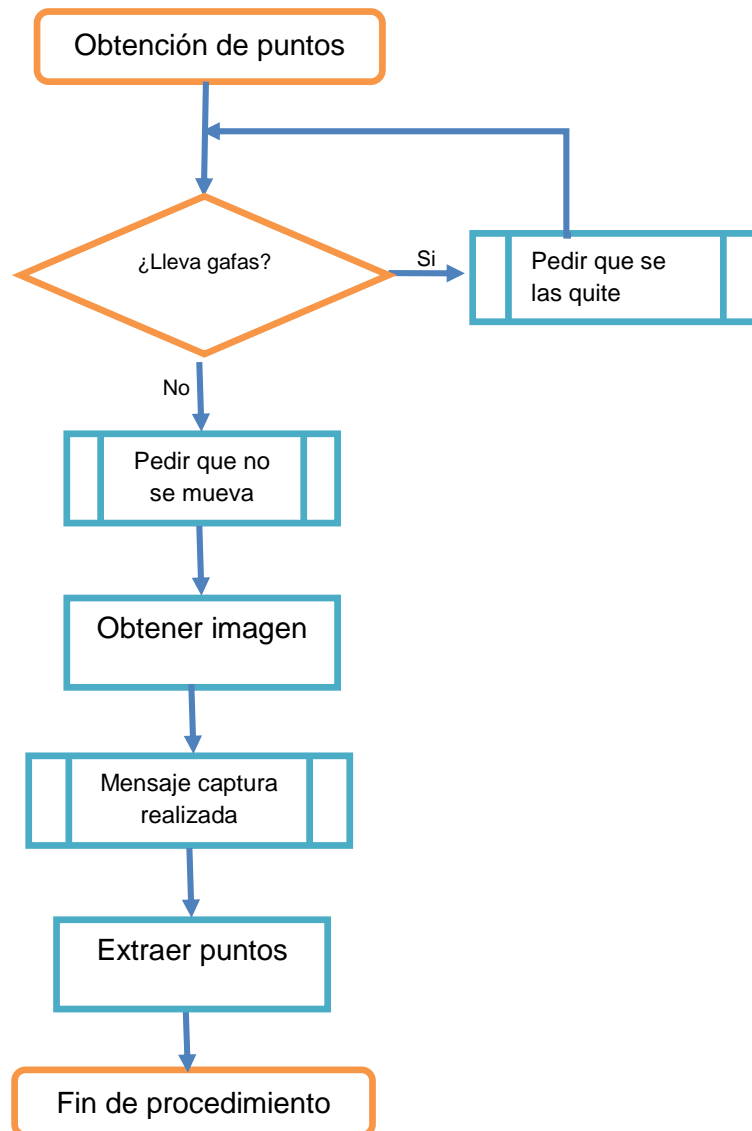


Figura 27 Función para la obtención de los puntos

5.4. Obtención de la imagen

5.4.2. Programación

Funcionamiento de la cámara

El dispositivo Kinect está formado por una cámara de profundidad, una cámara de color, una cámara infrarroja y un micrófono.

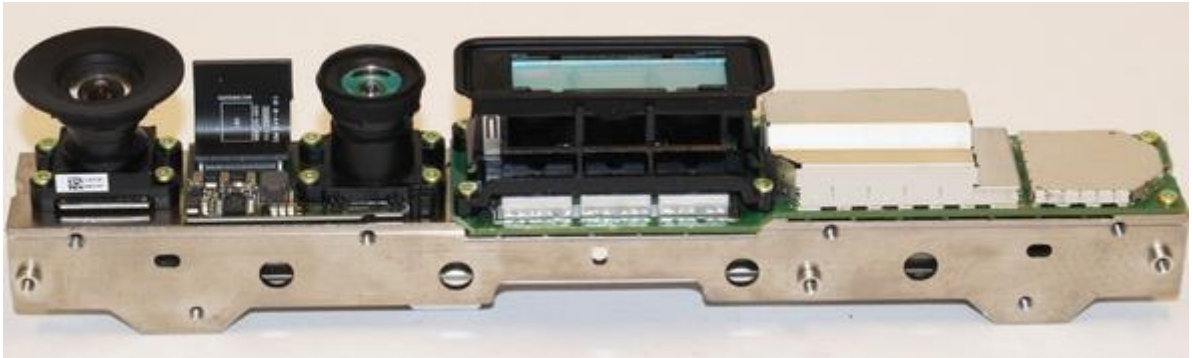


Figura 28 Componentes del sensor Kinect

Es lo que se conoce como un sensor inteligente, capaz de enviar directamente la información al ordenador. Está programada para el reconocimiento de personas y detección de las partes del cuerpo. El ordenador recibe información de ubicación y estado de cada una de ellas, con capacidad de identificar hasta 6 personas, centrándose principalmente en dos. Todo esto lo realiza a través de los sensores mencionados previamente.

El sensor de profundidad es del tipo time-of-flight, usa el mismo principio que el sensor de posicionamiento. La diferencia con el sensor de posicionamiento es que, en vez de ser un solo sensor láser, es un haz de láseres.

El sensor devuelve una imagen 2.5D, que como se ha visto anteriormente es una matriz en donde cada uno de los valores representa la distancia que hay desde el emisor hasta el punto donde refleja.

Inicialmente se planea utilizar esta imagen para localizar los puntos más relevantes de la cara. Para localizar la cara se utilizaría el algoritmo de Viola-Jones. Posteriormente, al tener localizada la cara, el punto de menor profundidad correspondería a la punta de la nariz.

Se estuvieron realizando ensayos con este sistema. Se tuvo que descartar debido a la imposibilidad de hacerlos coincidir en cada una de las distintas pruebas (fallo en el experimento 1 del capítulo 5).

Habiendo descartado este método se procede a buscar una alternativa para localizar los puntos. Partiendo de la base de poseer una cámara Kinect, se procede a documentarse de la capacidad del sensor. Se descubre que el propio SDK está capacitado para localizar los puntos más significantes, así que se utilizara como base de estudio del sistema.

SDK del dispositivo Kinect

Microsoft ha desarrollado un completo SDK para aprovechar las posibilidades que han sido desarrolladas para el dispositivo. La versión 2.0 del SDK se compone de los siguientes espacios de trabajo:

- Estado general del sensor: en este espacio se detalla el funcionamiento de los distintos aspectos del sensor y se pueden solicitar los datos brutos del mismo, como son la imagen captadas por el sensor de profundidad.
- Entradas que recibe el sensor: aquí se manda al sensor que tipos de datos se requieren. Se puede definir si solo interesa que mande la información de la cara o tipo de gestos concretos con la mano. Se realiza con el fin de que el controlador se centre en estos.
- Seguimiento del cuerpo: se devuelven los datos de cantidad de personas hay delante de la cámara y la posición de sus articulaciones. Devuelve los valores en un array con cada una de las posiciones en eje XYZ.
- Cara: se proporciona información referente a la cara, como posición y estado.
- Fusión: este espacio permite usar varios dispositivos Kinects con la finalidad de ser más preciso al disponer de varios ángulos.
- Constructor de gestos: va a crear una base de datos con combinaciones de gestos con el fin de que el sistema los reconozca de manera nativa y devuelva al sistema si se han realizados los mismos.

5.4. Obtención de la imagen

El sistema se va centrar principalmente en las APIs para la cara, que es el objeto de estudio. Lo primero que se va a especificar es que se va a utilizar la versión específica para Matlab, llamado Kin2 (Terven & Cordova, 2017).

Va a tener 2 modos: el procesado simple y el procesado HD de la cara. El procesado simple utiliza esta función (getFaces) se obtienen los siguientes datos:

- **FaceBox:** rectángulo que contiene la cara del usuario, devuelve un vector con la posición en el espacio de la cámara de color [Izquierda, arriba, derecha, abajo].
- **FacePoints:** matriz de 2x5 que representa 5 marcas de la cara (ojo izquierdo, ojo derecho, nariz, esquinas izquierda y derecha de la boca).
- **FaceRotation:** vector de 3 elementos que representa la orientación de la cara expresado como ángulos de Euler.
- **FaceProperties:** es un vector de los 8 elementos de la Tabla 12 en donde los valores son 0, 1, 2 y 3, que representan respectivamente, *desconocido, no, tal vez o sí*.

Tabla 12 Valores de FaceProperties de la API Kinect

Nombre	Descripción
Happy	El usuario muestra una sonrisa
Engaged	Combina el resultado de si el usuario tiene el ojo cerrado y está mirando a la cámara
WearinGlasses	El usuario lleva gafas
LeftEyeClosed	El ojo izquierdo está cerrado
RightEyeClosed	El ojo derecho está cerrado
MouthOpen	El usuario tiene la boca abierta
MouthMoved	El usuario ha movido la boca
LookingAway	Determina si el usuario está mirando a otro lado distinto de la cámara

El procesado HD utiliza la función getHDFaces devuelve una estructura con los siguientes campos:

- FaceBox: rectángulo que contiene la cara del usuario, devuelve un vector con la posición en el espacio de la cámara de color [Izquierda, arriba, derecha, abajo].
- FaceRotation: vector de 3 elementos que representa la orientación de la cara expresado como ángulos de Euler.
- HeadPivot: vector de 3 elementos en el cual se dan los valores de la cara rotada en torno al centro de la cabeza.
- AnimationUnit: 17 expresiones que capta la cámara.
- ShapeUnits: Forma una malla y este valor muestra la deformación de dicha en valores comprendidos entre -2 y 2.
- FaceModel: array de 3x1347 vértices que informan de la posición de los distintos puntos de la cara. Los puntos vienen expresados en el XYZ. En la Figura 29 se muestran los vértices más comunes.

5.4. Obtención de la imagen

```

HighDetailFacePoints_EyeLeft = 0,
HighDetailFacePoints_LefteyeInnercorner = 210,
HighDetailFacePoints_LefteyeOutercorner = 469,
HighDetailFacePoints_LefteyeMidtop = 241,
HighDetailFacePoints_LefteyeMidbottom = 1104,
HighDetailFacePoints_RighteyeInnercorner = 843,
HighDetailFacePoints_RighteyeOutercorner = 1117,
HighDetailFacePoints_RighteyeMidtop = 731,
HighDetailFacePoints_RighteyeMidbottom = 1090,
HighDetailFacePoints_LefteyebrowInner = 346,
HighDetailFacePoints_LefteyebrowOuter = 140,
HighDetailFacePoints_LefteyebrowCenter = 222,
HighDetailFacePoints_RighteyebrowInner = 803,
HighDetailFacePoints_RighteyebrowOuter = 758,
HighDetailFacePoints_RighteyebrowCenter = 849,
HighDetailFacePoints_MouthLeftcorner = 91,
HighDetailFacePoints_MouthRightcorner = 687,
HighDetailFacePoints_MouthUpperlipMidtop = 19,
HighDetailFacePoints_MouthUpperlipMidbottom = 1072,
HighDetailFacePoints_MouthLowerlipMidtop = 10,
HighDetailFacePoints_MouthLowerlipMidbottom = 8,
HighDetailFacePoints_NoseTip = 18,
HighDetailFacePoints_NoseBottom = 14,
HighDetailFacePoints_NoseBottomleft = 156,
HighDetailFacePoints_NoseBottomright = 783,
HighDetailFacePoints_NoseTop = 24,
HighDetailFacePoints_NoseTopleft = 151,
HighDetailFacePoints_NoseTopright = 772,
HighDetailFacePoints_ForeheadCenter = 28,
HighDetailFacePoints_LeftcheekCenter = 412,
HighDetailFacePoints_RightcheekCenter = 933,
HighDetailFacePoints_Leftcheekbone = 458,
HighDetailFacePoints_Rightcheekbone = 674,
HighDetailFacePoints_ChinCenter = 4,
HighDetailFacePoints_LowerjawLeftend = 1307,
HighDetailFacePoints_LowerjawRightend = 1327

```

Figura 29 Lista de valores del array con los puntos de la cara

De acuerdo con lo visto en el capítulo 2.4 los puntos que se deben de obtener para el sistema de reconocimiento están definidos en la Tabla 13

Tabla 13 Valores de la API de Kinect usados para el sistema

Descripción	Número
Punta de la nariz	14
Entrecejo	12
Punto interior ojo izquierdo	1088
Punto interior ojo derecho	1078
Ojo izquierdo	992
Ojo derecho	175

Se van a utilizar las dos funciones, la simple y la HD. Con la simple se identificará si el usuario tiene gafas, una vez comprobado que no, se procederá a obtener los puntos.

```

datosvalidos=0
while datosvalidos==0
    % Se obtienen las capturas procedentes del Kinect
    validData = k2.updateData;

    % Antes de procesar los datos hay que asegurarse, que lo obtenido es
    % valido
    if validData
        caras2=k2.getFaces;
        if size(caras2,2) > 0 & sum(caras2.FaceProperties)>7;
datosvalidos=1;end
        end
        pause(0.02)
    end

    obtenido = 0
    while obtenido == 0
        % Se obtienen las capturas procedentes del kinect
        validData = k2.updateData;

        % Antes de procesar los datos hay que asegurarse, que lo obtenido es
        % valido
        if validData
            caras1=k2.getHDFaces('WithVertices','true');
            if size(caras1,2) > 0 & sum(caras1.FaceBox)>0;obtenido=1;end
        end
        pause(0.02)
    end
end

```

Lo primero que se debe realizar es comprobar que la captura obtenida con el sensor es buena, en caso de que fuera mala, al obtener la información de la captura devolvería 0.

Una vez que se tienen los datos validos se procede a comprobar si el usuario usa gafas, para ello se usa la función getFaces y se toma el valor 3 de la estructura FaceProperties. Dando por solicitar que se quite las gafas si devuelve el valor 2(tal vez) o 3(si).

Finalmente, cuando el usuario no tenga las gafas puestas se llamará a la función getHDFaces donde se obtendrán los puntos de la cara que se van buscando.

5.5. Cálculo de los puntos característicos

Los puntos que dados como resultado en el apartado anterior vienen dados en coordenadas cartesianas, por lo que habrá que obtener las medidas y los ángulos que se han definido en el apartado 2.4.

Lo primero que se realizará, será crear un vector entre los puntos (P y Q en el ejemplo) a objeto de estudio, siendo:

$$P(x_1, y_1, z_1) \quad \text{y} \quad Q(x_2, y_2, z_2)$$

$$\overrightarrow{PQ} = (x_2 - x_1, y_2 - y_1, z_2 - z_1)$$

Para calcular la distancia, matemáticamente, será el módulo del vector dado:

$$|\overrightarrow{PQ}| = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2 + (z_2 - z_1)^2}$$

Para calcular el ángulo formado por dos vectores (\vec{A} y \vec{B}) (Anon., 2012):

$$\alpha = \cos^{-1} \left(\frac{\vec{A} \cdot \vec{B}}{|\vec{A}| \cdot |\vec{B}|} \right)$$

Funciones creadas:

```
function b=medida(a)
    b=sqrt(sum(a.^2));
end
```

```
function c=angulo(d,e)
    c=acos((d(1)*e(1)+d(2)*e(2)+d(3)*e(3))/(medida(d)+medida(e)));
end
```


5.6. Base de datos

Para la creación de la base de datos se va a usar una *estructura* de Matlab. Las estructuras, también llamadas registros en algunos lenguajes de programación modernos. Una estructura permite almacenar una combinación de datos de distintos tipos y , por tanto representar cualquier entidad real o imaginaria (Francisco, 2016).

Los campos de la estructura van a ser:

- Nombre: Cadena de caracteres donde se va a introducir el nombre del usuario
- Apellido: Cadena de caracteres con el apellido del usuario
- Acceso: Valores binarios, no se ha explicado antes, pero va a ser utilizado para el caso de estudio, donde el 1 significa puede pasar y el 0 no
- Características: vector con la media de los 10 valores obtenidos en la captura
- Desviación: vector con las desviaciones típicas de los valores obtenidas durante la captura

Por cada usuario habrá un campo en el array de estructuras. El fichero va a ser guardado en un archivo .mat (propietario de Matlab), aunque por las características se podría guardar como archivo .txt.

La base de datos se va a diseñar para una capacidad de 20 usuarios, se va a fijar con el fin de establecer el tamaño de la red neuronal.

5.7. Red neuronal

Como se ha comentado en el capítulo 2.4, se va a usar una red neuronal para la identificación de la persona una vez obtenidos los puntos característicos.

Las neuronas se van a organizar en capas, de modo que la red va a estar formado por varias de ellas. Cada neurona va recibir el valor de la entrada por un determinado peso (Figura 30), que son sumados y operados por una función de transferencia antes de transmitirse a la siguiente capa o como salida de la red.

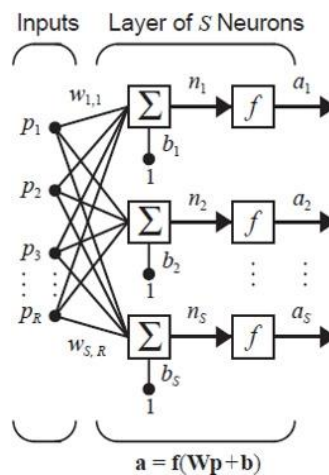


Figura 30 Red con una capa

Básicamente, la red está formada por una capa de entrada, una de salida y el resto son capas ocultas.

En la Figura 30 se muestra un esquema del aprendizaje supervisado, del mismo tipo que va a ser usado para nuestro sistema, donde se conoce los valores esperados según el valor de entrada, lo que va a permitir ajustar iterativamente los pesos de la red.

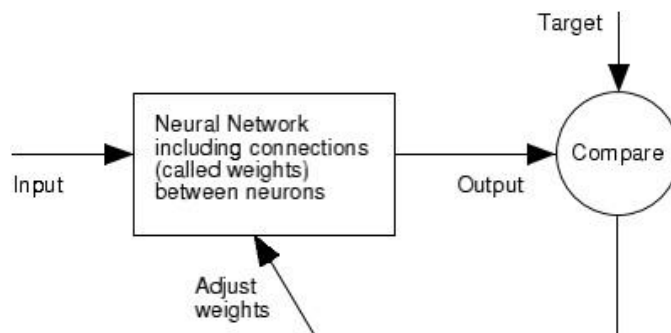


Figura 31 Proceso de ajuste de pesos en aprendizaje supervisado (Matlab, 2017)

La Tabla 14 define los pasos principales en el diseño de una red Neuronal

Tabla 14 Pasos fundamentales en el diseño de una red neuronal

Número	Paso
1	Coleccionar los datos
2	Crear la red
3	Configurar la red
4	Inicializar los pesos de las conexiones
5	Entrenar la red
6	Validar la red
7	Usar la red neuronal

El paso 1 es la recolección previa de todos los vectores características de los distintos usuarios que pertenecen a la base de datos. El resto de pasos se van a exponer en los capítulos 5.7.1, 5.7.2 y 5.7.3.

5.7.1. Creación de la red neuronal

En esta etapa del proceso se van a realizar los pasos 2, 3 y 4 de la Tabla 14. La herramienta de redes neuronales tiene una función específica para el reconocimiento de patrones.

Inicialmente se va a configurar la red definiendo 10 capas ocultas, en caso de que durante las pruebas el resultado no fuera el esperado, se debería de cambiar el número de capas ocultas.

Se va a determinar la división de los datos para el entrenamiento de la red:

- 70% dedicado a entrenar la red.
- 15% se usa para validar la red.
- 15% restante para probar la red.

Los códigos que se usan:

```
% Crear la Red neuronal para el reconocimiento
hiddenLayerSize = 10;
net = patternnet(hiddenLayerSize);

% Configurar los datos para entrenamiento, validación y prueba
% Van a ser un 70%, un 15% y un 15% respectivamente
```

5.7. Red neuronal

```
net.divideParam.trainRatio = 70/100;  
net.divideParam.valRatio = 15/100;  
net.divideParam.testRatio = 15/100;
```

La función `patternnet` se encarga de la creación de la red neuronal.

5.7.2. Entrenamiento de la red neuronal

Una vez creada, y antes de ser usada, se debe de entrenar la red neuronal. En esta etapa se van a realizar los pasos 5 y 6 de la Tabla 14.

La propia función de entrenamiento se encarga de la propia validación del resultado con el 15% de los datos introducidos. Cada vez que se añada un nuevo usuario se deberán de volver a reajustar los pesos de las conexiones entre las neuronas que componen la red.

El código que se usa para el entrenamiento es:

```
% Entrenar la red  
[net,tr] = train(net,inputs,targets);  
  
% Crear la funcion para ser usada posteriormente  
genFunction(net,'RED');
```

Donde `inputs` son los valores de entrenamiento, `targets` son los valores deseados de salida y `net` es la propia red. Una vez creada la red se va a crear una función para usar la red neuronal.

5.7.3. Uso de la red

En la función creada después del entrenamiento se introducen los 10 descriptores obtenidos de la captura.

Los resultados dados por esta función representan en valores entre 1 y 0 del usuario que más se acerca a ese descriptor. Para poder determinar el usuario más valido se van a sumar cada uno de los valores. El usuario seleccionado será el que mayor valor tenga.

6. EXPERIMENTOS Y RESULTADOS

Una vez definidos los puntos característicos y la forma de extraerlos se deben de corroborar su idoneidad para el sistema de reconocimiento. En este capítulo se realizarán diferentes experimentos para comprobarlo.

6.1. Experimento 1: Desviación de los puntos en capturas

El primer experimento que se va a realizar es comprobar el funcionamiento del SDK para localizar los puntos. Para realizar esta tarea, se van a realizar 10 tomas de los puntos característicos. Una vez obtenidas, se procederá calcular la media de los valores y su desviación típica.

Para esta prueba se solicitar a los usuarios que mantengan la misma expresión durante las 10 tomas. No se va a proceder a la variación del ángulo de la cámara ni variaciones en la luz del cuarto.

6.1.1. Resultados obtenidos en la medición

Usuario 1 (Tabla 15)

Tabla 15 Resultados del usuario 1 en el experimento 1

Puntos	1	2	3	4	5	6	7	8	9	10
Toma										
1	1,57206	1,57609	0,02298	0,02748	1,57206	0,80252	0,04263	1,57525	1,58841	1,55636
2	1,57173	1,57642	0,02217	0,02725	1,57173	0,83459	0,04230	1,57544	1,58899	1,55594
3	1,57177	1,57647	0,02226	0,02732	1,57177	0,83209	0,04248	1,57544	1,58897	1,55589
4	1,57155	1,57644	0,02187	0,02714	1,57155	0,84890	0,04205	1,57546	1,58911	1,55585
5	1,57145	1,57584	0,02210	0,02701	1,57145	0,83999	0,04140	1,57522	1,58863	1,55634
6	1,57162	1,57616	0,02213	0,02716	1,57162	0,83735	0,04191	1,57535	1,58885	1,55612
7	1,57166	1,57614	0,02221	0,02721	1,57166	0,83450	0,04198	1,57534	1,58879	1,55614
8	1,57154	1,57601	0,02213	0,02714	1,57154	0,83969	0,04171	1,57528	1,58872	1,55620
9	1,57157	1,57598	0,02219	0,02717	1,57157	0,83707	0,04174	1,57528	1,58868	1,55624
10	1,57149	1,57631	0,02183	0,02707	1,57149	0,85025	0,04181	1,57541	1,58904	1,55595
Media	1,57164	1,57619	0,02219	0,02720	1,57164	0,83569	0,04200	1,57535	1,58882	1,55610
Desv.	0,00017	0,00021	0,00030	0,00013	0,00017	0,01242	0,00036	0,00008	0,00021	0,00018

Usuario 2 (Tabla 16)**Tabla 16 Resultados del usuario 2 en el experimento 1**

Puntos	1	2	3	4	5	6	7	8	9	10
Toma										
1	1,57216	1,57710	0,02281	0,02738	1,57216	0,80206	0,04362	1,57568	1,58929	1,55566
2	1,57216	1,57685	0,02274	0,02741	1,57216	0,80516	0,04330	1,57561	1,58916	1,55583
3	1,57261	1,57696	0,02358	0,02788	1,57261	0,77828	0,04437	1,57559	1,58886	1,55584
4	1,57217	1,57677	0,02278	0,02736	1,57217	0,80197	0,04320	1,57556	1,58908	1,55592
5	1,57230	1,57720	0,02304	0,02753	1,57230	0,79429	0,04401	1,57573	1,58929	1,55561
6	1,57228	1,57708	0,02300	0,02748	1,57228	0,79494	0,04380	1,57566	1,58922	1,55571
7	1,57240	1,57732	0,02329	0,02760	1,57240	0,78574	0,04437	1,57573	1,58929	1,55557
8	1,57242	1,57726	0,02334	0,02766	1,57242	0,78495	0,04437	1,57568	1,58920	1,55560
9	1,57195	1,57649	0,02246	0,02721	1,57195	0,81566	0,04255	1,57547	1,58901	1,55606
10	1,57206	1,57703	0,02263	0,02734	1,57206	0,81025	0,04339	1,57564	1,58931	1,55566
Media	1,57225	1,57701	0,02297	0,02748	1,57225	0,79733	0,04370	1,57563	1,58917	1,55575
Desv	0,00018	0,00024	0,00033	0,00018	0,00018	0,01128	0,00057	0,00008	0,00014	0,00015

6.1.2. Conclusiones:

En la Tabla 17 se observa que el valor de las desviaciones típicas es bastante bajo por lo que las mediciones entre las distintas imágenes es muy buena y la medición va a ser robusta.

También se observa que los discriminantes 4, 5 y 8 pueden ser los más favorables para la identificación porque el valor de la desviación típica es de los más bajos y se pueden usar en caso de un poco probable empate.

Tabla 17 Tabla resumen del experimento 1

	1	2	3	4	5	6	7	8	9	10
Media 1	1,57164	1,57619	0,02219	0,02720	1,57164	0,83569	0,04200	1,57535	1,58882	1,55610
Desv. 1	0,00017	0,00021	0,00030	0,00013	0,00017	0,01242	0,00036	0,00008	0,00021	0,00018
Media 2	1,57225	1,57701	0,02297	0,02748	1,57225	0,79733	0,04370	1,57563	1,58917	1,55575
Desv 2	0,00018	0,00024	0,00033	0,00018	0,00018	0,01128	0,00057	0,00008	0,00014	0,00015

6.2. Experimento 2: Variación de las expresiones

El segundo experimento consistirá en ver la variación de los puntos con diferentes expresiones con el fin de comprobar que los puntos no varían con las expresiones. Las expresiones a objeto de estudio serán:

- Neutra
- Feliz
- Triste
- Abrir la boca

6.2.1. Resultados obtenidos en la medición

Usuario 1 (Tabla 18)

Tabla 18 Resultados del usuario 1 en el experimento 2

Puntos	1	2	3	4	5	6	7	8	9	10
Toma										
Neutra	1,57170	1,57631	0,02221	0,02714	1,57170	0,83095	0,04211	1,57537	1,58888	1,55607
Feliz	1,57188	1,57651	0,02245	0,02719	1,57188	0,81791	0,04255	1,57541	1,58892	1,55601
Triste	1,57133	1,57597	0,02183	0,02724	1,57133	0,86132	0,04154	1,57531	1,58876	1,55604
Abrir la boca	1,57136	1,57546	0,02244	0,02720	1,57136	0,83646	0,04130	1,57505	1,58815	1,55654
Media	1,57157	1,57606	0,02223	0,02719	1,57157	0,83666	0,04188	1,57528	1,58868	1,55617
Desv	0,00026	0,00046	0,00029	0,00004	0,00026	0,01819	0,00056	0,00016	0,00036	0,00025

Usuario 2 (Tabla 19)

Tabla 19 Resultados del usuario 2 en el experimento 2

Puntos	1	2	3	4	5	6	7	8	9	10
Neutra	1,57223	1,57718	0,02293	0,02756	1,57223	0,80126	0,04393	1,57571	1,58931	1,55556
Feliz	1,57206	1,57650	0,02270	0,02731	1,57206	0,80703	0,04279	1,57543	1,58887	1,55609
Triste	1,57229	1,57718	0,02305	0,02779	1,57229	0,80218	0,04419	1,57571	1,58922	1,55549
Abrir la boca	1,57227	1,57651	0,02303	0,02761	1,57227	0,79831	0,04327	1,57545	1,58876	1,55607
Media	1,57221	1,57684	0,02293	0,02757	1,57221	0,80220	0,04354	1,57557	1,58904	1,55580
Desv	0,00010	0,00039	0,00016	0,00020	0,00010	0,00362	0,00063	0,00016	0,00026	0,00032

6.2.2. Conclusiones:

Se observa que cuando se le pide al usuario que abra la boca de manera exagerada se produce una variación mayor respecto a los otros. Se debe de tener en cuenta a la hora de explicárselo a los usuarios o en su defecto programar al sistema para que detecte si tiene la boca abierta para realizar otra medida.

6.3. Experimento 3: Cambios de posición

El último experimento consistirá en ver la posible discrepancia entre las distintas posiciones. Las posiciones a objeto de estudio serán:

- Posición neutra
- Inclined hacia arriba
- Inclined hacia abajo
- Girada a la derecha
- Girada a la izquierda

Para este experimento se realizará solo un usuario ya que como se ha visto en el experimento 1 las medidas que realiza el sistema son bastante buenas.

6.3.1. Resultados obtenidos en la medición

Usuario 1 (Tabla 18)

Tabla 20 Resultados en el experimento 3

Puntos	1	2	3	4	5	6	7	8	9	10
Toma										
Centrada	1,57171	1,57611	0,02237	0,02729	1,57171	0,82929	0,04211	1,57529	1,58867	1,55617
Arriba	1,57180	1,57664	0,02218	0,02739	1,57180	0,83628	0,04268	1,57557	1,58913	1,55574
Abajo	1,57154	1,57642	0,02188	0,02721	1,57154	0,85071	0,04209	1,57545	1,58905	1,55583
Derecha	1,57174	1,57678	0,02214	0,02733	1,57174	0,83729	0,04277	1,57556	1,58928	1,55564
Izquierda	1,57140	1,57609	0,02177	0,02705	1,57140	0,85487	0,04150	1,57534	1,58897	1,55607
Media	1,57164	1,57641	0,02207	0,02725	1,57164	0,84169	0,04223	1,57544	1,58902	1,55589
Desv	0,00015	0,00027	0,00021	0,00012	0,00015	0,00956	0,00046	0,00011	0,00020	0,00020

6.3.2. Conclusiones:

Después de realizar este experimento queda claro que no es determinante el ángulo de la cara, salvo en el descriptor número 6, donde se ve influenciado si la cara esta girada a la derecha o a la izquierda

7. CASO DE ESTUDIO: CONTROL DE ENTRADA Y SALIDA DE LOS TRABAJADORES DE UNA PLANTA.

Se propuso como caso de estudio el control de entrada en una planta, el sistema debería identificar a la persona y comprobar si posee los permisos necesarios para acceder al mismo.

Para este caso se va a crear una interfaz sencilla que va a ser capaz de introducir un nuevo usuario, administrar la base de datos de usuarios, la función de registro de accesos y la activación del mismo sistema.

Se va a utilizar la herramienta GUIDE de Matlab con la intención de que sea fácilmente utilizada si necesidad de tener grandes conocimientos del sistema.

Se dividirá la interfaz en tres partes:

- Control de acceso: activación del sistema, donde se permitirá el acceso a la persona reconocida siempre y cuando esté autorizado.
- Gestión de usuarios: se accede a la base de datos de manera sencilla, donde se puede dar acceso a cualquier usuario e introducir un nuevo usuario.
- Registro de eventos: control de personas que han entrado e introducción de nuevos usuarios, marcando la fecha y la hora a la que ha sucedido.

7.1. Control de acceso

Este módulo se encargará del seguimiento, la activación y la desactivación del sistema de identificación para el acceso de los usuarios. Se va a requerir que tenga un botón para activar y desactivar el sistema. También se va a introducir una pantalla en la cual se informa al administrador del mismo en qué etapa se encuentra del proceso de identificación.

Se diseña la interfaz con dos apartados: zona de texto para informar y conjunto de botones para activar y desactivar el sistema (Figura 32).

7. Caso de estudio: control de entrada y salida de los trabajadores de una planta.

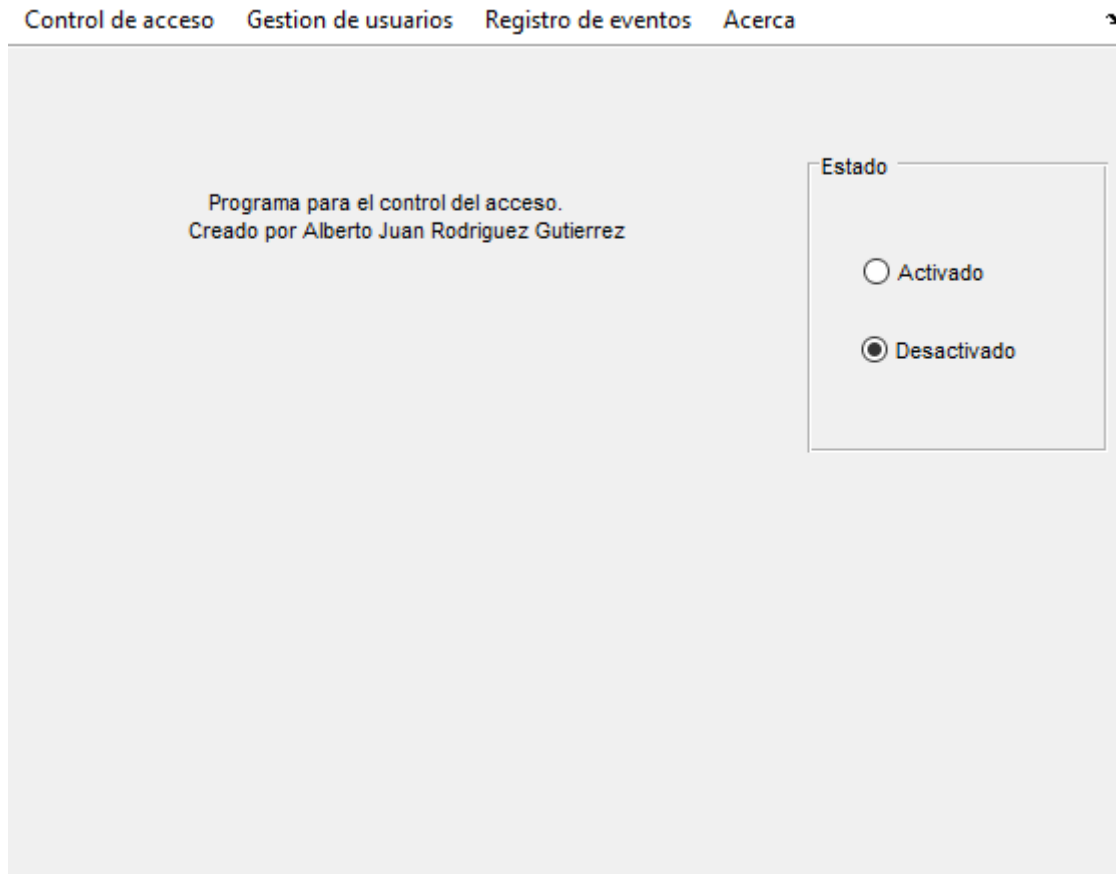


Figura 32 Interfaz de control de acceso

Al pulsarse el botón **Activado** se ejecuta el siguiente código y se desactivara al pulsar cualquier otro botón.

```
set(handles.texto1, 'String', 'Activado');
while get(hObject, 'Value')
    mm = rangeMilliMeter(handles.sensor)
    while mm < 950 | mm > 1200
        if get(handles.desactivado, 'Value')==1
            break;
        end
        clearLCD(handles.lcd)
        if mm==0
            clearLCD(handles.lcd)
            set(handles.texto1, 'String', sprintf('Activado\nEsperando que
venga alguien'));
        else
            if mm < 950
                printLCD(handles.lcd, 'Alejate please');
                set(handles.texto1, 'String', sprintf('Activado\nEsperando a
que el usuario se aleje'));
            else
                printLCD(handles.lcd, 'Acercate please');
                set(handles.texto1, 'String', sprintf('Activado\nEsperando a
que el usuario se acerque'));
            end
        end
        mm = rangeMilliMeter(handles.sensor);
    end
end
```

7.2. Gestión de usuarios

```
        pause(2)
    end
    clearLCD(handles.lcd);
    %% Se realizan las capturas
    if get(handles.desactivado, 'Value')==1
        break;
    else
        clearLCD(handles.lcd);
        set(handles.texto1, 'String', sprintf('Activado\nTomando la
captura'));
        while true
            [puntos gafas]=captura;
            clearLCD(handles.lcd);
            printLCD(handles.lcd, 'Quitate las');
            printLCD(handles.lcd, 'gafas');
            pause(3)
        end
        clearLCD(handles.lcd);
        % Se obtienen los puntos caracteristicos
        set(handles.texto1, 'String', sprintf('Activado\nLocalizando los
puntos'));
        [ descriptor ] = caracterizacion( puntos );

        %% Reconocimiento en red neuronal
        set(handles.texto1, 'String', sprintf('Activado\nIdentificando al
usuario'));
        RED(descriptor)
    end
end
```

Todas las funciones usadas son las mismas que se han ido documentadas en el capítulo 5.

7.2. Gestión de usuarios

Esta parte se va a encargar de la gestión de los usuarios, va a tener dos funciones muy diferentes, por un lado, la posibilidad de agregar un usuario nuevo y por otra parte la de ver los usuarios registrados y administrar el acceso sobre estos.

En la Figura 33 se va a dividir en dos zonas:

- Tabla: se va a colocar una tabla donde estén todos los usuarios y estarán los campos del nombre, apellido y acceso (siendo este marcable o desmarcable)
- Zona de agregar usuario: va a estar compuesta por dos cuadros de texto para introducir el nombre y el apellido, un popup para otorgar el acceso y finalmente un botón de Agregar usuario

7. Caso de estudio: control de entrada y salida de los trabajadores de una planta.

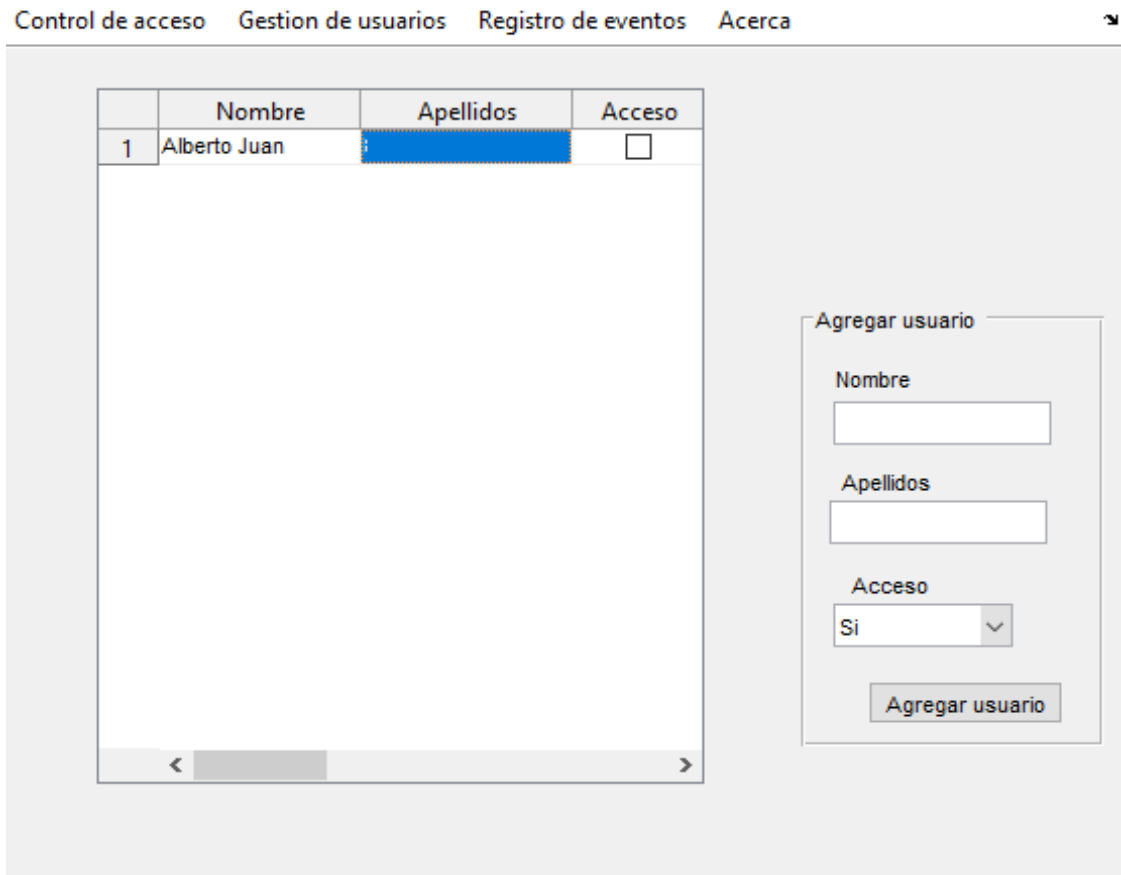


Figura 33 Interfaz Gestión de usuarios

Una vez que se pulse el botón de agregar usuario se va a ejecutar el código de realizar 10 capturas y se procederá a entrenar la red con el nuevo usuario.

7.3. Registro de eventos

Se ha decido agregar este módulo con el fin de saber quién ha accedido al sistema y poder monitorizar las entradas. Tiene básicamente dos funciones: registrar la entrada de un usuario nuevo a la base de datos y registrar la entrada de alguien a la zona controlada por el sistema.

En la Figura 34 se observa que para este control se va a usar una tabla que va a seguir un orden cronológico. Donde va a tener 3 campos: fecha, hora y el evento que se ha producido.

7.3. Registro de eventos

Control de acceso Gestion de usuarios Registro de eventos Acerca

	Fecha	Hora	Evento
1	28/08/2017	20:45:05	Creacion de la base de datos

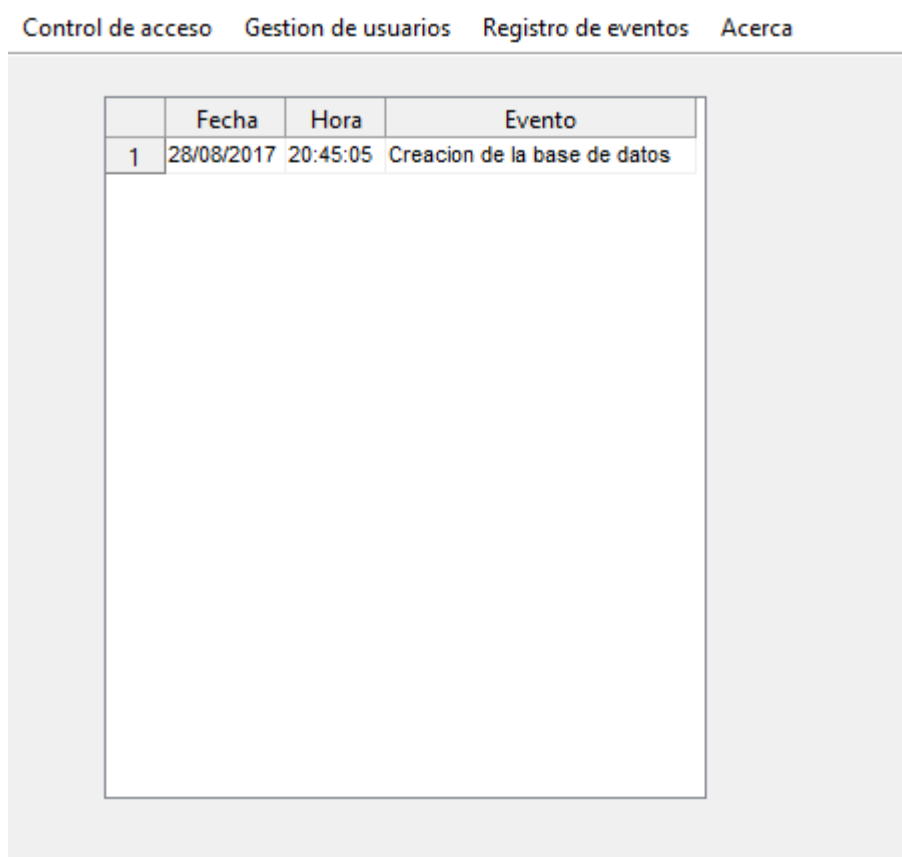


Figura 34 Interfaz del Registro de eventos

8. CONCLUSIONES

El trabajo desarrollado ha tratado de presentar, analizar y proponer una solución al reconocimiento facial en distintos tipos de entornos, tanto controlados en el laboratorio como ubicados en cualquier lugar, utilizando para ello un sistema de reconocimiento tridimensional y una técnica basada en los puntos característicos de la cara.

Debido a que el tema del reconocimiento facial está estudiado en el campo de 2D y su problemática, se ha decidido probar un método de reconocimiento facial más robusto. Se han estudiado todos los posibles métodos, decantándose finalmente por la extracción de características.

Se han usado seis puntos característicos de los cuales se han establecido diez relaciones entre ellos usando geometría básica de distancias y ángulos.

Después se ha seleccionado el equipamiento necesario para dicha tarea, con el principal objetivo de buscar un equipo asequible y fiable. Se ha seleccionado una cámara Kinect, un sensor láser, una pantalla LCD y una placa Arduino. Por lo que el coste de todo el sistema es inferior a 200€.

Posteriormente, se ha desarrollado el método más sencillo para la localización de los puntos característicos. Se ha hecho uso de las propias herramientas del dispositivo Kinect. Este hecho ha propiciado que la obtención sea rápida y más precisa.

Conviene resaltar que el sistema es bastante fiable en la mayoría de los casos, pero como se ha visto con los experimentos, al abrir la boca se puede producir una distorsión de los resultados, por lo que se debe tener en cuenta.

Quedando como trabajo futuro, la búsqueda de otros puntos característicos de manera automática y ampliando el número de relaciones con el fin de aumentar el grado de certeza del sistema. También, aumentando el tamaño de la red para saber el grado de precisión con una muestra más grande de usuarios en la base de datos.

9. BIBLIOGRAFÍA

3DMD, s.f. *3dMD LLC*. [En línea]

Available at: <http://www.3dmd.com/3dmd-systems/#face>

Abate, A. F., Nappi, M., Riccio, D. & Sabatino, G., 2007. 2D and 3D face recognition: A survey. *Pattern Recognition Letters*, pp. 1885-1906.

Adafruit, s.f. *Adafruit*. [En línea]

Available at: <https://www.adafruit.com/product/3317>

Alfonso, M. R., 2009. *Desarrollo de una interfaz gráfica de redes neuronales usando Matlab*. Madrid: Universidad Carlos III de Madrid.

Anon., 2012. *Departamento de Física Aplicada*. [En línea]

Available at:

http://laplace.us.es/wiki/index.php/Calcular_el_%C3%A1ngulo_entre_dos_vectores

Anon., 2017. *Wikipedia*. [En línea]

Available at: <https://es.wikipedia.org/wiki/Índice>

Anon., s.f. *ASUSTeK COMPUTER INC.* [En línea]

Available at: https://www.asus.com/es/3D-Sensor/Xtion_PRO_LIVE/

Ansari, Y., Moses, Y. & Ullman, S., 2003. Face recognition: The problem of compensating for changes in illumination direction. *IEEE Trans. Pattern Anal. Machine Intell.* 19, pp. 721-732.

Arduino, 2017. *Arduino*. [En línea]

Available at: <https://www.arduino.cc/>

Arduino, 2017. *Arduino*. [En línea]

Available at: <https://www.arduino.cc/>

Artec, s.f. *Artec Europe*. [En línea]

Available at: <https://www.artec3d.com/es/3d-scanner/artec-eva>

Brans, J., 1982. L'Ingénierie de la décision. Elaboration d'instruments d'Aide. *Université LAVAL. Colloque d'Aide*, pp. 183-213.

Chang, K. I., Bowyer, K. W. & Flynn, P. J., 2004. An Evaluation of Multimodal 2D+3D Face Biometrics. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, pp. 619-624.

Chen, X., Flynn, P. & Bowyer, K., 2003. PCA-based face recognition in infrared imagery: Baseline and comparative studies. *Proc. IEEE Internat. Workshop on Analysis and Modeling of Faces and Gestures*, pp. 127-134.

Chua, C., Han, F. & Ho, Y., 2000. 3D human face recognition using point signature. *Proc, 4th IEEE Internat. Conf. on Automatic Face and Gesture Recognition*, pp. 233-238.

Cottrell, G. & Fleming, M., 1990. Face recognition using unsupervised feature extraction. *Proc. Intell. Neural Network Conf.*, pp. 322-325.

- Courane, S., 2017. *Github*. [En línea]
Available at: <https://github.com/sandorcourane/matlab-vl53l0x.git>
- Francisco, M. d. R., 2016. *Programación en Matlab*. Jaén: Universidad de Jaén.
- Goldstein, A., Harmon, L. & Lesk, A., 1971. Identification of Human Faces. *Proc. IEEE, Vol.59 n°5*.
- Gordon, G., 1991. Face recognition based on depth maps and surface curvature. *Proc. Geometric Methods in Computer Vision*, pp. 1-12.
- Hu, Y. X., Jiang, D. L., Yan, S. C. & Zhang, H. J., 2004. *Automatic 3D Reconstruction for face Recognition*. Seoul, s.n., pp. 843-850.
- Kanade, T., 1977. Computer Recognition of Human Faces. *Interdisciplinary Systems Research, Vol. 47*.
- Kirby, M. & Sirovich, L., 1990. Application of the Karhunen-Loeve procedure for the characterization of human faces. *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 103-108.
- Lades, M. y otros, 1993. Distortion invariant object recognition in the dynamic link architecture. *IEEE Trans. Comput.* 42, pp. 300-311.
- Li, S. Z. & Jain, A. K., 2011. Introduction. En: *Handbook of Face Recognition*. Londres: Springer, pp. 1-18.
- Lu, J., Plataniotis Kostantions, N. & Venetsanopoulos Anastasios, N., 2003. Face recognition using LDA-based algorithms. *IEEE Trans. Neural Networks* 14, pp. 572-581.
- Lu, X., Colbry, D. & Jain, A. K., 2004. *Three-dimensional model based face recognition*. Cambridge, s.n., pp. 362-366.
- Martinez, A. & Kak, A., 2001. PCA versus LDA. *IEEE Trans. Pattern Anal. Machine Intell.* 23, pp. 228-233.
- Matlab, 2017. *Matlab*. [En línea]
Available at: <https://es.mathworks.com/help/>
- Meng, J., Shiquian, W., Juwei, L. & Hock, L., 2002. Face recognition with radial basis function (RBF) neural networks. *IEEE Trans. Neural Networks* 13, pp. 697-710.
- Microsoft, s.f. *Microsoft inc.* [En línea]
Available at: <https://developer.microsoft.com/es-es/windows/kinect/hardware>
- Minolta, 2009. *Konica Minolta Business Solutions Spain S.A.*. [En línea]
Available at:
https://www.konicaminolta.com/instruments/download/instruction_manual/3d/pdf/vivid-910-vi-910-instruction-eng.pdf
- Moreno, A. B. D., 2004. *Reconocimiento Facial Automático mediante Técnicas de Visión Tridimensional*. Madrid: Universidad Politécnica de Madrid.

OECD, 2009. Height. En: *Society at a Glance 2009: OECD Social Indicators*. Paris: OECD Publishing.

Papatheodorou, T. & Rueckert, D., 2004. Evaluation 4D face recognition using surface and texture registration. *Proc. Sixth IEEE International Conf. on Automatic Face and Gesture Recognition*, pp. 321-326.

Pérez García, M. A., Álvarez Antón, J. C., Campo Rodríguez, J. C. & Grillo Ortega, G. J., 2003. *Instrumentación Electrónica*. s.l.:S.A. Ediciones Paraninfo.

Phillips, P. y otros, 2002. Face Recognition Vendor Test: Evaluation Report.

Polyga, 2017. *Polyga*. [En línea]

Available at: <https://www.polyga.com/hdi-3d-scanner/hdi-advance/>

Riccio, D. & Nappi, M., 2003. *Defering range/domain comparisions in fractal image compression*. s.l., s.n., pp. 412-417.

STMicroelectronics, s.f. *STMicroelectronics*. [En línea]

Available at: <http://www.st.com/resource/en/datasheet/vl53l0x.pdf>

Terven, J. R. & Cordova, D. M., 2017. *Kin2. User Guide*. s.l.:s.n.

Tsalknidou, F., Tzovaras, D. & Strintzis, M., 2003. Use of depth and colour Eigenfaces for face recognition. *Pattern Recognition Letter 24*, pp. 1427-1435.

Vetter, T. & Blanz, V., 2003. Face rrecognition based on fitting a 3D model. *IEEE Trans. Pattern Anal. Machine Intell.*, pp. 1063-1074.

Zhao, W., Chellappa, R., Phillips, P. & Rosenfeld, A., 2003. Face Recognition: A Literature Survey. *ACM Computing Surveys, Vol.35, No 4, December 2003*, pp. 399-458.

ANEXO I: REQUISITOS DEL SISTEMA

Requisitos de Hardware

Para poder usar el sistema es necesario que el equipo, se van a desglosar según los distintos elementos que lo componen.

Requisitos para Matlab 2017a.

- Procesador: cualquier procesador AMD o Intel es válido.
- Disco duro: 6GB de espacio en el disco duro.
- Memoria RAM: 2GB de memoria.
- Tarjeta gráfica: Se recomienda 1GB de GPU, no hace falta ningún modelo en particular.

Requisitos para sensor kinect:

- Procesador: doble núcleo físico de 3,1 GHz
- Controlador USB 3.0
- Memoria RAM: 4GB de memoria
- Tarjeta gráfica: debe de admitir DirectX 11

Requisitos de Software

Para poder usar los distintos componentes del sistema hay que configurar cada uno de los distintos componentes que forman el sistema.

Funcionamiento del sensor Kinect

Para el correcto funcionamiento de la cámara se deben de seguir los siguientes pasos:

- 1- Instalar SDK de Kinect para Windows, obtenido de: <https://www.microsoft.com/en-us/download/details.aspx?id=44561>
- 2- Instalar Visual Studio 2012 o superior.
- 3- Instalar en Matlab el siguiente Add-on desde su administrador: Image Acquisiton Toolbox Support Package for Windows Sensor

- 4- Descargar la adaptación del sdk para Matlab (Kin2) de:
<https://github.com/jrterven/Kin2>
- 5- Añadir al Path de Windows los ficheros Kinect20Fusion.dll y Kinect20.Face.dll.
- 6- Ejecutar el siguiente programa en Matlab, incluido en la carpeta de Kin2:

compile_cpp_files.m

Funcionamiento del arduino

Para que funcione la placa arduino hay que instalar el Add-on para arduino: MATLAB Support Package for Arduino Hardware

También es necesario instalar el software Arduino IDE, descargado de:
<https://www.arduino.cc/en/Main/Software>

Funcionamiento de la pantalla LCD

Solo hay que añadir el addon ExampleLCD a la carpeta de trabajo. Este se encuentra en la documentación de Matlab.

Funcionamiento del sensor

Hay que descargar la librería Adafruit_VL53L0X desde el gestor de librerías de la aplicación de arduino.

Además, se debe agregar la carpeta del add-on VL53L0X a la carpeta de trabajo, descargado de:

<https://github.com/sandorcourane/matlab-vl53l0x>