

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

ESCOLA POLITECNICA SUPERIOR DE GANDIA

Grado en Ing. Sist. de Telecom., Sonido e Imagen

---



UNIVERSITAT  
POLITÈCNICA  
DE VALÈNCIA



ESCOLA POLITÈCNICA  
SUPERIOR DE GANDIA

“**DESARROLLO DE UNA HERRAMIENTA PARA LA DETECCIÓN Y EL  
RECONOCIMIENTO DE GESTOS Y ACTITUDES CORPORALES  
SEMICONSCIENTES E INCONSCIENTES EMPLEANDO EL SENSOR KINECT**”

***TRABAJO FINAL DE GRADO***

Autor/a:

**Josep Àngel Capilla Blanch**

Tutor/a:

**Jose Ignacio Herranz Herruzo**

Cotutor:

**José Marín-Roig Ramón**

***GANDIA, 2017***

## **Resumen**

En este proyecto se aborda el desarrollo de una herramienta para la detección y reconocimiento de los gestos de la "escultura" de las personas (posición del tronco, cabeza, brazos y piernas) empleando el sensor Kinect para su posterior interpretación conforme a la sinergología™. Esta disciplina estudia e interpreta los gestos y actitudes corporales que no son realizados de forma plenamente consciente.

## **Resum**

En aquest projecte s'aborda el desenvolupament d'una ferramenta per a la detecció i reconeixement dels gestos de la "escultura" de les persones (posició del tronc, cap, braços i cames) emprant el sensor Kinect per a la seua posterior interpretació d'acord amb la sinergologia™. Esta disciplina estudia i interpreta els gestos i actituds corporals que no són realitzats de forma plenament conscient.

## **Abstract**

This project addresses the development of a tool for the detection and recognition of the "sculpture" of people (trunk position, head, arms and legs) using Kinect sensor for later interpretation according to synergology™. This discipline studies and interprets corporal gestures and attitudes which are not carried out in a fully conscious way.

## **Palabras clave**

Programación, Sinergología, Kinect, Reconocimiento, Gestos, Aplicación.

## **Keywords**

Programming, Synergology, Kinect, Recognition, Gestures, Application.

# Índice

Capítulo 1.	Introducción y objetivos .....	3
1.1	Introducción.....	3
1.2	Objetivos .....	3
1.2.1	Objetivo principal .....	3
1.2.2	Objetivos secundarios .....	3
Capítulo 2.	Sinergología.....	4
2.1	Introducción.....	4
2.2	Apertura y cierre.....	5
2.3	Tipologías.....	6
2.3.1	La agresividad .....	6
2.3.2	La sumisión .....	7
2.3.3	La confianza en uno mismo .....	8
2.3.4	La huida.....	9
Capítulo 3.	Requerimientos hardware y software necesarios.....	10
3.1	Kinect .....	10
3.1.1	Especificaciones técnicas Kinect v2.0 .....	11
3.1.2	Funcionamiento Kinect v2.0 .....	13
3.2	Matlab.....	15
3.2.1	Librería Image Acquisition toolbox .....	15
	Librería Image processing toolbox .....	18
3.2.2.....		18
Capítulo 4.	Resolución del algoritmo .....	19
4.1	Algoritmo principal .....	19
4.1.1	Inicialización.....	19
4.2	Captación de frames .....	20
4.2.1	Visualización del esqueleto.....	21
4.3	Funciones secundarias .....	22
4.3.1	calculadist.....	22
4.3.2	Cuerpodetectado.....	23
4.4	Funciones para la detección de gestos.....	23
4.4.1	Función codocogido.....	23
4.4.2	Función distanciacodos .....	24
4.4.3	Función Distanciainterlocutor.....	25
4.4.4	Función Distanciamanos .....	26

4.4.5	Funcion Distanciapiernas .....	26
Capítulo 5.	Interfaz gráfica de usuario .....	27
5.1	Introducción.....	27
5.2	Detalle de la interfaz.....	27
Capítulo 6.	Análisis de resultados .....	29
6.1	Introducción.....	29
6.2	Pruebas de funcionamiento.....	29
6.2.1	Posiciones de inseguridad y cierre .....	29
6.2.2	Posiciones de seguridad o agresividad.....	31
6.3	Análisis de las pruebas .....	32
6.3.1	Resultados de las pruebas para posiciones de inseguridad y cierre .....	33
6.3.2	Resultados de las pruebas para posiciones de seguridad o agresividad .....	33
Capítulo 7.	Conclusiones y líneas futuras .....	34
Capítulo 8.	Bibliografía .....	35

# Capítulo 1. Introducción y objetivos

## 1.1 Introducción

Se estima que en una conversación entre humanos un 35% de la comunicación corresponde a las palabras mientras que alrededor de un 65% corresponde a componentes no verbales[2] (gestos, movimientos, señales, etc... ).

Teniendo en cuenta este dato es posible suponer el importante papel que juega la comunicación no verbal cuando nos expresamos y cuanta información puede pasar desapercibida para un receptor que no sepa de la existencia de este código tan desconocido.

La sinergología estudia el lenguaje corporal semiconsciente y no consciente que emite una persona en situación de comunicación y los sinérgicos son expertos en este tipo de comunicación, es decir, son capaces de extraer más información sobre lo que expresa el interlocutor que una persona que se limite únicamente a escuchar lo que este dice.

En el ámbito de los negocios es de vital importancia tanto saber transmitir bien como dar la sensación de seguridad y convencimiento de lo que se está exponiendo al receptor. Esto explica el creciente interés por perfeccionar la comunicación no verbal, las expresiones y en definitiva lo que transmite el interlocutor por las escuelas de negocios.

Este proyecto cobra sentido cuando se habla de la necesidad de las escuelas de negocio comentada anteriormente ya que está enfocado a complementar la labor de un sinérgico, es decir, el programa detectará y procesará tipologías concretas definidas por la sinergología tales como Aceptación, Negación, Timidez, Seguridad etc.. y dará al usuario una aproximación del estado del interlocutor.

Es importante destacar que se contará además con el apoyo de una autoridad en el campo de la Sinergología como es Maria José Arlandis autora del libro[1].

## 1.2 Objetivos

### 1.2.1 *Objetivo principal*

El objetivo principal de este proyecto es enfrentarse al problema real de diseñar e implementar desde cero un programa en Matlab capaz de detectar y procesar los gestos que realiza una persona en tiempo real mientras realiza una exposición, haciendo frente a todos los contratiempos que puedan surgir utilizando las competencias adquiridas en la universidad. Para ello se hará uso de la Kinect V2, hardware propietario de Microsoft, un sensor capaz de detectar 25 puntos del cuerpo humano de hasta 6 personas simultáneamente.

### 1.2.2 *Objetivos secundarios*

Para abordar este proyecto correctamente se hará uso del conocimiento adquirido en asignaturas como “Matlab para Ingeniería de telecomunicaciones”, “Programación” y especialmente “Tratamiento digital de imagen y video” asentando así las bases creadas durante el periodo lectivo y ampliándolas si fuera necesario.

El aprendizaje y la expansión del conocimiento a otros ámbitos diferentes al que se ha estudiado también es importante ya que muchas veces los proyectos no tratan de una materia en concreto sino de una mezcla como es el caso de éste, requiriendo también conocimientos en sinergología. Por tanto otro de los objetivos será el aprendizaje y asentamiento de unas bases en esta disciplina.

## Capítulo 2. Sinergología

### 2.1 Introducción

La Sinergología es una disciplina creada por Philippe Turchet[2] en los años 80 que estudia el lenguaje corporal inconsciente y semiconsciente cuando una persona se está comunicando. Mediante esta técnica se consigue descubrir el estado interno del emisor, es decir, se puede llegar a intuir lo que siente al expresarse.

Se les llama ítems a los signos corporales correctamente etiquetados o codificados de los que se vale la Sinergología para realizar el estudio emocional de la persona a la que se está analizando.

Todos los seres humanos no tenemos la misma gestualidad para todo, es por ello que el porcentaje de acierto de esta disciplina oscila entre el 80 y el 90%[3], aunque sí tenemos muchos patrones de conducta similares, ya que nuestro cerebro primitivo funciona de la misma manera y es este aspecto el que aprovecha la Sinergología.

La sinergología trabaja el análisis gestual con lo que se denominan las “**tres miradas**”[1].

1 – La observación de la **estatua**: Se observa la posición de la persona en un contexto general es decir, se fija la atención en la postura del cuerpo desde la cabeza hasta los pies sin entrar en detalles.

2 – La observación de la **actitud interior**: Este tipo de observación amplía la percepción y es más específica, se trata de fijarse en las articulaciones del cuerpo (muñecas, tobillos, hombros...) prestando atención en si existe relajación o por el contrario rigidez.

3 – La observación de los **Micromovimientos**: Se trata de los movimientos más sutiles y pequeños que se hacen con el cuerpo a través de las manos (pequeños rascados, sutiles caricias etc...).



Figura 1 Estatua

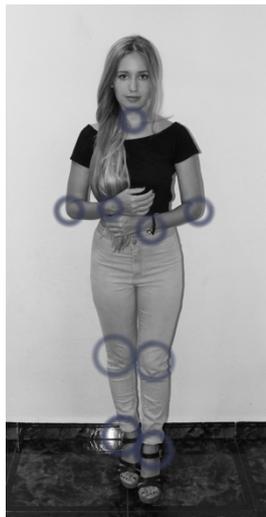


Figura 2 Actitud interior



Figura 3  
Micromovimientos

Por tanto este proyecto se va a centrar en análisis de una situación concreta, una persona sola hablando y exponiendo en público, y como objetivo principal el programa deberá realizar una lectura fiable de la actitud de la persona y concluir si sus gestos son propios de una persona segura y por tanto está segura de lo que está exponiendo o por el contrario la persona tiene una actitud más bien insegura. Para ello el programa hará uso de unos gestos concretos llamados ítems, proporcionados por la sinergología, y se centrará en el análisis utilizando dos de las tres miradas comentadas anteriormente: la observación de la estatua y la actitud interior

## 2.2 Apertura y cierre

Cómo es lógico, con la observación de un solo ítem no es suficiente para realizar un diagnóstico fiable del estado del interlocutor. Por tanto, para que la lectura sea fiable se deberán detectar en el sujeto varios ítems los cuales apunten todos en un mismo sentido, para así obtener lo que se llama el “horizonte de sentido”.

Estos ítems se pueden distinguir generalmente entre:

Ítems de **apertura**: El cuerpo del interlocutor se percibe relajado, abierto y expandido, el sujeto está cómodo porque no percibe ninguna amenaza. Si el interlocutor tiene una actitud abierta se mostrará cercano a la persona con la que está hablando, sus articulaciones estarán relajadas y sus movimientos no serán muy rápidos y más bien armónicos. Si la persona está segura de sí misma es posible que una de las señales de apertura que exponga sea la ligera expansión del pecho. El exponer el cuerpo es una forma de demostrar que se está abierto y por tanto seguro de lo que se está hablando.

Otros signos de apertura corresponden a la actitud verbal, es decir, el interlocutor se mostrará cercano, atento y participativo en la exposición oral. Existirá además cierta inclinación a establecer un contacto visual así como una voluntad de comunicar de forma agradable.

Ítems de **cierre**: El interlocutor se encuentra incómodo por alguna razón, el cuerpo reacciona como si percibiese alguna amenaza, se bloquea, se protege y cubre los órganos sensoriales. Si el interlocutor muestra una actitud cerrada se pueden observar tensiones en las articulaciones y se puede percibir en el habla una situación de estrés y nerviosismo. Los movimientos de los brazos al explicarse serán más bien rígidos y no se percibirá un armonía en ellos mientras que la estatua de la persona será de repliegue sobre sí mismo. Se tiende a esconder las manos, cruzar las piernas y brazos, es decir la persona pondrá barreras inconscientemente entre ella y el público. Un interlocutor con actitud cerrada no buscará el contacto visual con el oyente y tendrá una actitud menos cercana.

Así pues la tarea principal del programa una vez detectado el cuerpo del individuo que está exponiendo será detectar si ésta emite señales de apertura o de cierre. Esto corresponderá a una primera identificación del estado del interlocutor y permitirá al programa realizar un estudio genérico de la estatua de la persona para más tarde profundizar en ello analizando los gestos más detalladamente y detectando posiciones más específicas. La finalidad es informar al usuario del programa cuáles son los aspectos que se podrían mejorar para realizar una exposición oral o cuáles son los aspectos que se están realizando bien.

Debido a que el hardware Kinect tiene sus limitaciones, el programa no será capaz de detectar los microgestos. Aunque se haya hablado de estos anteriormente con el objetivo de poner en situación al lector, en este proyecto no se trabajará esta parte. No obstante,

no se descarta la posibilidad de poder incluir un módulo de microgestos en versiones posteriores del programa. Sobre este tema se hablará también más adelante. Así pues, como se ha expuesto anteriormente el programa se centrará en el análisis de la estatua y la actitud interior.

Para detectar a que tiende la estatua, el programa analizará la distancia entre brazos y tronco así como la distancia entre las piernas lo cual aportará información sobre el estado de expansión o repliegue del cuerpo.

En cuanto a la actitud interior, con la cual se analizan gestos mas complejos y en los que actúan mas articulaciones, se hará uso de los 25 puntos del cuerpo humano que es capaz de detectar la Kinect. El objetivo es profundizar en la estatua y detectar gestos más específicos etiquetados en la sinergología, tales como cruce de piernas y de brazos así como posturas típicas según la tipología, de las cuales se hablará en el siguiente punto.

## **2.3 Tipologías**

En el ámbito de la Sinergología existen varias tipologías, las cuales definen cuales son los gestos y las reacciones comunes que tenemos los seres humanos cuando nos enfrentamos a según qué situaciones o cuando tenemos un sentimiento en concreto. Estas reacciones están alojadas en la parte del cerebro mas primitiva “el cerebro reptiliano” el cual se encarga de las funciones primarias y regula los elementos mas básicos de la supervivencia. Debido a que todos los seres humanos disponemos de este “cerebro reptiliano” se pueden definir unos patrones típicos para las siguientes tipologías[1]:

- 1- La agresividad.
- 2- La sumisión.
- 3- La confianza en uno mismo.
- 4- El rechazo.
- 5- La huida.
- 6- La empatía.
- 7- La seducción.

Estos son un ejemplo de algunas de las tipologías con las que trabaja la Sinergología y las cuales tienen ciertos patrones concretos que se comentarán a continuación. Debido a que el ámbito de estudio donde el programa va a funcionar se va a desarrollar en un entorno profesional no se comentarán todos los ejemplos descritos anteriormente, sino una selección de los que pueden ser útiles para el correcto funcionamiento del programa.

### **2.3.1 La agresividad**

El instinto de supervivencia está regido por el cerebro reptiliano, inscrito en él desde hace millones de años, ya que es capaz de anticipar las posibles reacciones de agresividad de un interlocutor que puedan resultar determinantes para la integridad física. Es por ello que detectar estos indicativos es de suma importancia.

Seguidamente se analizarán los ítems mas importantes de la agresividad.

La estatua de una persona agresiva es en general territorial es decir, el sujeto reclamará espacio desplegando su cuerpo en una posición muy característica y a menudo poniendo los brazos en jarras y manteniendo una gran separación entre las piernas. Esta posición indica claramente que el sujeto controla la situación. En cuanto a la actitud interior, mantendrá una postura erguida y rígida, ya que está preparado para la “acción”. La mirada juega un papel muy importante a la hora de detectar la agresividad en un sujeto. Una mirada demasiado fija con tensión en los músculos que rodean los ojos y con

escasos parpadeos es un indicativo claro, así como la tensión en los puños cuando se cierran.



**Figura 4 brazos en jarras**

Es de suma importancia observar si el sujeto invade el espacio personal de otras personas, es decir, la posición del sujeto en cuanto al espacio debería ser una posición en la que no se viole el espacio vital de otro ser humano.

Así pues para detectar un posible caso de agresividad en el sujeto se tendrán en cuenta los siguientes aspectos:

- 1- Una posición expandida, que reclame territorio.
- 2- Posición característica de brazos en jarra y piernas separadas.
- 3- Puños cerrados.
- 4- Posición del interlocutor respecto a los oyentes.

### **2.3.2 La sumisión**

La sumisión supone el ejemplo contrario al caso descrito en el punto anterior. El comportamiento de un sujeto sumiso es más bien pasivo, acepta las opiniones de los que le rodean sin tener en cuenta la suya propia. Son sujetos que no toman la iniciativa. El comportamiento pasivo produce sentimiento de malestar y a la larga puede producir depresión y ansiedad.

En las personas sumisas la estatua suele ser replegada, el sujeto tiende a esconderse, a ocupar el menor espacio posible para minimizar su cuerpo. Al contrario que en el caso anterior, el sujeto no reclama territorio sino que lo cede en su actitud de sumisión. La persona no presenta tensión en las articulaciones, y se encontrarán posturas más bien relajadas, ya que su actitud es de aceptar lo que viene. La posición de la cabeza será ladeada y con el mentón hacia abajo exponiendo un lado del cuello.

En cuanto a la actitud interior se puede destacar que normalmente, los brazos de sujeto se pegarán al cuerpo y lo mismo sucederá con las piernas, las cuales se mantendrán pegadas, en el caso de un sujeto sentado las rodillas tenderán a juntarse y las manos a buscar un escondite.

Al igual que la agresividad tiene una posición característica, la sumisión o inseguridad la tiene también, debido a que el sujeto piensa que la situación le supera y es inferior a la audiencia (en el caso de una exposición). Éste tiende a realizar el “auto abrazo” que no es otra cosa que una reacción para proporcionarse a sí mismo alivio, consuelo y tranquilidad. Por tanto la posición típica de auto abrazo sumiso será cuando la persona sostiene el brazo izquierdo con el brazo derecho como aparece en la imagen.



Figura 5 ej. autoabrazo 1



Figura 6 ej. autoabrazo 2

Una vez descritos los típicos ítems que caracterizan a una persona sumisa se enumeran los que se podrían utilizar para detectar la sumisión:

- 1- Brazos pegados al cuerpo.
- 2- Piernas pegadas al cuerpo.
- 3- Manos escondidas.
- 4- Cabeza ladeada.
- 5- Posición del interlocutor respecto al oyente.
- 6- Auto abrazo.
- 7- Brazos o piernas cruzadas.

### 2.3.3 *La confianza en uno mismo*

El primer paso hacia la confianza es la autoestima y el conocerse a uno mismo, conocer perfectamente tanto las debilidades como las fortalezas que se tienen y actuar en consecuencia. Una persona segura de sí misma y con confianza cree en lo que dice, cree en la validez de sus palabras y se lanza sin dejarse llevar por los nervios. De ahí la espontaneidad y la naturalidad tanto del habla como de la expresiones no verbales de los individuos que tienen confianza en si mismos.

La estatua del individuo con confianza en sí mismo es firme y flexible al mismo tiempo, expandiendo el cuerpo pero no exageradamente como en el caso de la agresividad. En este caso el individuo no necesita más espacio del necesario. Por tanto se percibirá una estatua no forzada, muy natural.

Al estar relajados no presentarán la cabeza demasiado baja ni demasiado alta, ni existirán extremos ladeos de cabeza. En cuanto a la posición de las manos, tenderán a

estar visibles siempre (no hay nada que esconder) y los movimientos al expresarse serán fluidos y seguros.

Como se puede deducir por lo comentado anteriormente, el sujeto seguro de sí mismo adopta una postura muy natural y por tanto el programa funcionará por descarte, es decir, tratará de descartar que el sujeto es sumiso o agresivo para más tarde analizar los posibles gestos de confianza. Básicamente se analizará:

- 1- Posición neutral en cuanto a brazos y piernas, ni demasiado expandidos ni demasiado replegados.
- 2- Posición correcta del cuello, barbilla en posición correcta y sin ladeos de cabeza excesivos.
- 3- Posición de las manos, siempre visibles.
- 4- Posición del individuo respecto a los oyentes.

#### **2.3.4 La huida**

Cuando el ser humano no está cómodo en una situación en la que no desea estar, el cerebro pone en marcha el mecanismo de huida del sistema límbico y prepara al cuerpo para ello aunque sea de forma inconsciente. En este apartado se nombrarán aquellos ítems que muestran ese deseo de “huida” en el interlocutor.

La estatua de la persona que quiere huir de una situación o de una persona es de distanciamiento, es decir, el sujeto se distancia del público al igual que le sucede a una persona sumisa con la diferencia de que la persona que quiere huir mantiene signos de “escape” hacia algún lado.

En cuanto a la actitud interior, será importante fijarse en el eje de la cabeza, la cual es posible que esté ligeramente alejada y ladeada hacia alguno de los dos lados, la mirada buscará la salida en vez de prestar atención al oyente u oyentes y alguno de los pies puede que apunte hacia la salida mas próxima. Un ítem de huida muy característico son los picos tanto en la parte posterior de pierna derecha como en la espalda.

Se puede deducir pues que si una persona de encuentra de brazos cruzados con la cabeza ladeada y distanciada y con las piernas cruzadas hacia la salida, el sujeto no está cómodo con la situación en la que se encuentra y tiende a huir.

Con el fin de detectar la huida en el interlocutor con el programa realizado para este proyecto se podría tener en cuenta los siguientes ítems:

- 1- La posición del sujeto respecto del oyente u oyentes.
- 2- La posición de la cabeza (si está ladeada o no) junto con la distancia de la misma respecto del publico.
- 3- Cruce de manos y cruce de piernas.

## Capítulo 3. Requerimientos hardware y software necesarios.

### 3.1 Kinect

La cámara de la que hace uso el programa es la Kinect v2.0, un hardware propietario de Microsoft el cual fue creado para la videoconsola Xbox One. Este hardware es el predecesor directo de la Kinect creada para la Xbox 360 y ofrece varias mejoras respecto a esta.

La Kinect fue un éxito de ventas debido a todas las novedades que ofrecía respecto a sus competidoras. En Junio de 2011 trascendió del ámbito de los videojuegos al de la programación debido a que Microsoft diseñó el driver (Kinect for Windows) para que ésta pudiese ser interpretada y utilizada por un PC. Esto abrió el abanico de posibilidades de la máquina y pasó de ser una webcam de videoconsola a una herramienta de trabajo para muchos programadores y aficionados de la programación. Desde entonces se han creado infinidad de programas que hacen uso de este aparato ya que puede ser increíblemente útil debido a toda la información respecto a la persona que capta que puede obtener el programador.

La Kinect v2.0 es utilizada principalmente para el reconocimiento corporal y gestual de los seres humanos en el ámbito de la programación, ya que es capaz de reconocer 25 puntos del cuerpo humano de hasta 6 personas simultáneamente. El aparato cuenta además con una cámara RGB, con un procesador interno el cual ejecuta el software patentado por Microsoft, proporcionando la captura del movimiento de todo el cuerpo en 3D, con un sensor de profundidad el cual será utilizado por el programa y con un micrófono interno del que no se hará uso en este proyecto. No obstante es importante señalarlo de cara a nuevas versiones ya que podría ser una útil herramienta para realizar comandos por voz.

La razón principal por la cual utilizar la Kinect v2.0 en el programa no es otra que la de partir de una base firme respecto al reconocimiento de los puntos estratégicos del cuerpo humano. En un principio se propuso la idea de diseñar un programa el cual reconociese estos puntos para más tarde programar el reconocimiento gestual con los fundamentos que dicta la Sinergología. Muy pronto se desechó la idea debido al gran volumen de trabajo y de tiempo que esto supondría. La opción de utilizar la Kinect v2.0 supuso un gran avance para el programa ya que se partía de toda la información que puede obtener el aparato, eliminando así la tarea de diseñar un programa que reconociese a personas.

En los siguientes puntos se hablará de las características técnicas de la Kinect v2.0 y se profundizará en el funcionamiento de la misma tanto hardware como software (programación de la Kinect v2.0 en Matlab).



Figura 7 Vista lateral Kinect



Figura 8 Vista frontal Kinect



Figura 9 Vista lateral Kinect

### 3.1.1 Especificaciones técnicas Kinect v2.0

No es casualidad que programadores de todo el mundo elijan la Kinect v2.0 para realizar su trabajo ya que este aparato cuenta con unas especificaciones propias de un hardware profesional.

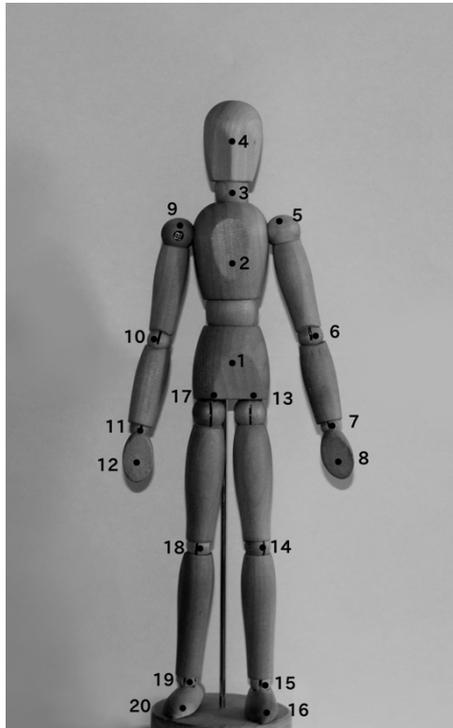
Dispone de 3 tipos de cámara[10]:

- Full HD, la cual tiene una resolución de 1920 x 1080 píxeles, permitiendo detectar con mayor facilidad todo el entorno y pudiendo incluso diferenciar la orientación del cuerpo.
- Detector de profundidad, con esta función la cámara baja la resolución a 512 x 424 píxeles y ofrece información en una escala de grises sobre la profundidad de las 25 articulaciones del cuerpo humano que detecta.
- Cámara infrarroja, cómo su propio nombre indica se vale de un sensor infrarrojos (IR) de los que dispone la Kinect v2.0 en su parte frontal para poder detectar en la oscuridad.

Detecta 25 articulaciones de cada persona

1	Spine base	Columna vertebral
2	Spine Mid	Punto medio de la columna
3	Neck	Cuello
4	Head	Cabeza
5	Shoulder Left	Hombro izquierdo
6	Elbow Left	Codo izquierdo
7	Wrist Left	Muñeca izquierda
8	Hand Left	Mano izquierda
9	Shoulder Right	Hombro derecho
10	Elbow Right	Codo derecho
11	Wrist Right	Muñeca derecha
12	Hand Right	Mano derecha
13	Hip Left	Lado izquierdo cadera
14	Knee Left	Rodilla izquierda
15	Ankle Left	Tobillo izquierdo
16	Foot Left	Pie izquierdo
17	Hip Right	Lado derecho cadera
18	Knee Right	Rodilla derecha
19	Ankle Right	Tobillo derecho
20	Foot Right	Pie derecho
21	Spine Shoulder	Espina dorsal
22	Hand Tip Left	Mitad mano izquierda
23	Thump Left	Pulgar izquierdo
24	Hand Tip Right	Mitad mano derecha
25	Thump Right	Pulgar derecho

Tabla 1 Numeración de las articulaciones detectadas por la Kinect[10]



**Figura 10** Puntos del cuerpo detectados por la cámara Kinect



**Figura 11** Puntos unidos a modo de esqueleto

El campo de visión de la Kinect es de 70° en horizontal y 60° en vertical. Esto le permite detectar a más personas que su predecesora ya que amplía su rango de visión (hasta un total de 6). Se puede destacar, no obstante, que esta nueva versión no cuenta con el motor de inclinación con el que contaba la anterior.

Incorpora además un sensor de profundidad interno el cual tiene un rango de actuación de 0.5m a 4.5m lo que quiere decir que es posible detectar la profundidad de cada uno de los puntos detectados en ese rango descrito. Esta función es muy útil en el programa que se desarrolla en este proyecto para detectar la posición del individuo respecto de los oyentes.

El micrófono del que dispone tiene tecnología “multiarray”. Esto permite no solo poder reconocer la voz del interlocutor o interlocutores si no la procedencia de las mismas. Todo esto gracias a tres micrófonos, dos situados a los extremos y uno frontal, los cuales calculan la procedencia del sonido dependiendo de la potencia de entrada. Aunque como se ha mencionado anteriormente esta característica no va a ser utilizada por el programa no está de más mencionarla ya que puede ser interesante tenerla en cuenta de cara a posibles futuras actualizaciones.

La conexión al ordenador se realiza mediante USB 3.0 (siendo requisito indispensable a la hora de programar con ella). Esto permite una óptima velocidad de transmisión de datos con el ordenador ya que como se sabe el USB 3.0 puede trabajar a velocidades de hasta 5Gbps, disminuyendo la latencia del sensor a 60ms. Es importante mencionar que para poder conectar la Kinect al ordenador es indispensable realizar la compra de un módulo a parte llamado “Kinect Adapter” el cual dispone de un transformador especial para poder realizar la conexión de la Kinect v2.0 al ordenador mediante USB 3.0

Por último cabe mencionar la posibilidad que tiene de captar movimientos a oscuras y esto es gracias al emisor de infrarrojos (IR) del que dispone, el cual se hablará de su funcionamiento en el siguiente punto.

Seguidamente se muestra una tabla resumen sobre las características técnicas mas destacables de la Kinect 2.0

Cámara a color	1920 x 1080 – 30fps
Cámara de profundidad	512 x 424
Cámara infrarroja	512 x 424 – 30 Hz
Máxima distancia de profundidad	4.5m
Mínima distancia de profundidad	0.5m
Ángulo de visión horizontal	70°
Ángulo de visión vertical	60°
Articulaciones del esqueleto detectadas	25
Máximo de personas que detecta	6
Conexión al ordenador	USB 3.0
Sistemas operativos	Windows 8/8.1 y 10

**Tabla 2 Resumen especificaciones técnicas de la Kinect[11]**

### **3.1.2 Funcionamiento Kinect v2.0**

En este apartado se explicará de una forma técnica el funcionamiento de dos de las propiedades mas importantes que tiene este hardware: el sensor de profundidad del que hace uso el software de detección de personas y el array de micrófonos del que dispone.

#### **3.1.2.1 Sensor de profundidad**

El sensor de profundidad utiliza la tecnología Time-of-Flight (ToF)[12] y es una herramienta de la cual dispone la Kinect que es muy útil para reconocer la profundidad a la que se encuentran diferentes partes del cuerpo. Funciona siguiendo 4 pasos fundamentalmente:

- 1- Un foco de luz infrarroja compuesto por un rayo por pixel ilumina al sujeto y al mismo tiempo este refleja la luz al sensor.
- 2- El sensor contiene un chip el cual mide la distancia que ha recorrido la luz para cada uno de los píxeles de la imagen.
- 3- El software interno del cual está equipado la Kinect utiliza la información del mapa de profundidad (información que recibe de los cálculos hechos en el punto 2) para percibir e identificar objetos en tiempo real.
- 4- Esta identificación es interpretada por el usuario final según el fin que tenga, una videoconsola, una video cámara de seguridad, un software de detección de humanos etc...

Para explicar el funcionamiento del sensor de profundidad con mas detalle se utilizará el ejemplo de la versión más simple del ToF. La cámara utiliza un solo pulso de luz infrarroja por un periodo muy corto de tiempo, este pulso de luz infrarroja ilumina la escena y el objeto en el que haya incidido la refleja, este reflejo es recogido por un sensor.

Dependiendo de la distancia a la que se encuentra el objeto la luz reflejada experimenta un retardo mayor o menor.

Se sabe que la velocidad de la luz es  $c = 300 \cdot 10^6$  m/s, por tanto para obtener el retardo que produce según que distancia se utilizará la siguiente fórmula:

$$t_D = 2 \cdot \frac{D}{c}$$

La Kinect utilizará entonces la siguiente fórmula para obtener la distancia dependiendo del retardo de cada rayo de luz infrarroja:

$$D = c \cdot \frac{t_D}{2}$$

Siendo D la distancia en metros, c la velocidad de la luz en m/s y  $t_D$  el tiempo de retardo en segundos.

El sensor está compuesto por un fotodiodo por cada píxel, el cual convierte la luz entrante en una corriente eléctrica. Cada píxel está conectado a un fotodetector que dispone de un temporizador encargado de realizar las operaciones descritas anteriormente y calcular la distancia a la que está el objeto.

Cada cámara que utilice la tecnología ToF dispone de los siguientes elementos:

- Unidad de iluminación: Esta unidad utiliza la iluminación infrarroja, ya que no puede ser percibida por el ojo humano y por tanto no es molesta.
- Lente óptica: Se encarga de reunir la luz reflejada e imprime imágenes del entorno en el sensor. Se utiliza un filtro paso banda óptico para filtrar únicamente la luz con la misma longitud de onda que la unidad de iluminación. Con esto se consigue reducir ruido y que no todas las fuentes de iluminación puedan afectar al funcionamiento del sensor.
- Sensor de imagen: Según se ha explicado anteriormente, cada píxel mide el tiempo que ha tardado la luz en viajar desde que es proyectada hasta el objeto que refleja.
- CPU interna: Para poder realizar estas operaciones en un tiempo del orden de picosegundos se necesita un procesador potente. Es por ello que la Kinect dispone de uno interno el cual realiza todos estos cálculos.
- Interfaz: Debido a que la Kinect realiza los cálculos internamente, no es necesario el uso de un ordenador. No obstante ésta transfiere los datos calculados al mismo haciéndose valer de su interfaz USB.

### **3.1.2.2 Array de micrófonos**

La Kinect dispone de un Array de micrófonos con el cual es posible calcular el origen de la fuente de sonido debido a que a cada micrófono le llega el sonido en un instante diferente. Teniendo en cuenta el desfase captado por los diferentes micrófonos se aplican una serie de algoritmos para que el aparato sea capaz de ubicar la fuente de sonido en el espacio.

Una vez es calculada la posición de donde proviene el sonido, se procesa el audio y se fusionan la señales de todos los micrófonos para producir una señal de sonido de mayor calidad. Es importante resaltar que el rango de identificación de sonido que tiene la Kinect oscila desde 50 a -50 radianes.

Otra de las tareas que realiza el array de micrófonos de la Kinect es el reconocimiento de voz, así como la eliminación de ruido ambiente. Para ello aplica un filtro en el rango de frecuencias al que opera la voz humana. Una vez filtrada la voz, se utilizan varios procesadores de señal digital (DSPs), los cuales disponen de todos los algoritmos propios de Microsoft preinstalados para producir un mejor reconocimiento de voz.

## 3.2 Matlab

La implementación del programa se realizará en Matlab, ya que se ha trabajado con él durante todo el periodo lectivo y la familiarización es alta. Además de ser muy potente, Matlab ofrece infinidad de librerías para trabajar en diversas áreas así como una amplia base de datos informativa para dar soporte a todo aquel que decida programar con él.

Se utilizará la versión R2016b de Matlab instalada en un PC portátil Lenovo ThinkPad T440s con especificaciones:

- Sistema operativo: Windows 8
- Procesador: Intel core i5
- Velocidad de CPU: 2.5 GHz
- Arquitectura procesador: x64
- Memoria RAM: 4 GB
- Memoria interna: 500 GB

Es importante destacar que este equipo dispone de la cantidad mínima de RAM con la cual el programa puede funcionar. Esto puede ocasionar fallos en algunos momentos de sobrecarga, así como una fluidez menor que en equipos con mayor capacidad de procesamiento y de memoria RAM. Para evitar este problema sería recomendable funcionar con un equipo que disponga de un procesador Intel i7 y una memoria RAM de 16 GB.

Para el buen desarrollo de la aplicación se usarán las librerías para Matlab “Image Acquisition toolbox” and “Image processing toolbox”, herramientas con las cuales se realizará la adquisición de toda la información que nos proporciona la Kinect y el posterior procesamiento de las imágenes junto con los datos con el fin de recrear los gestos realizados de una manera visual para el usuario final.

### 3.2.1 Librería *Image Acquisition toolbox*

Image Acquisition Toolbox[9] es la librería que se utilizará para la adquisición de los datos técnicos sobre el cuerpo detectado que proporciona la Kinect, como la posición en el espacio de los 25 puntos del esqueleto, la distancia a la que la persona se encuentra e incluso su voz.

Esta librería dispone de bloques que permiten conectar cámaras industriales y científicas e incluye una aplicación que permite a Matlab detectar y configurar interactivamente las propiedades del hardware. Además soporta los principales

estándares y proveedores de hardware, puede detectar cámaras de profundidad 3D (Microsoft Kinect), cámaras de visión artificial y captadores de frames así como dispositivos industriales de alta gama.

Dentro de esta librería existen bloques pre-programados que dan acceso a las diferentes opciones y configuraciones que tiene la cámara Kinect, entre las cuales se encuentran las siguientes:

<b>imaqtool</b>	Lanza la herramienta de adquisición de imagen.
<b>videoinput</b>	Crea un objeto de entrada de video.
<b>imaq.VideoDevice getdata</b>	Adquiere un frame del dispositivo de video.
<b>peekdata</b>	Muestra los datos mas recientes adquiridos
<b>getsnapshot</b>	Devuelve inmediatamente un frame en forma de imagen de lo que capta la cámara en ese momento.
<b>start</b>	Indica el comienzo del funcionamiento de captación de imagen mediante el dispositivo.
<b>islogging</b>	Determina si el objeto creado como entrada de video está registrado
<b>isrunning</b>	Determina si el objeto creado como entrada de video está en funcionamiento.
<b>isvalid</b>	Determina si el objeto creado como entrada de video está asociado con el dispositivo de captación de imagen.
<b>wait</b>	Espera hasta que el dispositivo de captación de imagen deje de ejecutarse o se registre.
<b>stop</b>	Detiene el objeto creado como entrada de video.
<b>clear</b>	Limpia el registro de adquisición de imagen de Matlab.
<b>delete</b>	Elimina el objeto creado como entrada de video de la memoria de trabajo de Matlab
<b>flushdata</b>	Elimina los datos del buffer de almacenamiento de frames de imágenes adquiridas.
<b>obj2mfile</b>	Convierte los objetos de entrada de video en código Matlab.
<b>imaqmontage</b>	Crea una secuencia de frames a modo de video.
<b>load</b>	Carga el objeto de adquisición de imagen en el espacio de trabajo de Matlab.
<b>save</b>	Guarda los objetos de adquisición de imagen como archivo MAT.
<b>trigger</b>	Inicia el registro de datos.
<b>triggerconfig</b>	Configura las propiedades de activación de los objetos de entrada de video.

<b>triggerinfo</b>	Proporciona información sobre los objetos de entrada de video.
--------------------	--

**Tabla 3 Resumen bloques pre programados Image Acquisition Toolbox[9]**

Una vez se crea el objeto de video, automáticamente se le asignan unos “metadata”. Los metadata almacenan los datos importantes del entorno que la cámara va captando periódicamente gracias a sus sensores. Para poder acceder a estos datos, que normalmente están almacenados en forma de matriz, Matlab dispone de unas palabras clave para cada una de las variables:

<b>IsBodyTracked</b>	Se trata de una matriz booleana 1x6, en el caso que se detecte cuerpo aparecerá un 1 en el caso de que no aparecerá un 0.
<b>BodyTrackingID</b>	Matriz booleana 1x6 y representa la identificación de detección de cada cuerpo, se pueden detectar hasta 6 cuerpos, por tanto habrá hasta 6 identificaciones diferentes.
<b>ColorJointIndices</b>	Matriz doble 25x2x6 con coordenadas espaciales X e Y las cuales representan la situación de las 25 articulaciones detectadas en el espacio (en pixels) de la imagen a color.
<b>DepthJointIndices</b>	Matriz doble 25x2x6 con coordenadas espaciales X e Y las cuales representan la situación de las 25 articulaciones detectadas en el espacio (en pixels) de la imagen del sensor de profundidad.
<b>BodyIndexFrame</b>	Matriz 424x512 la cual indica qué píxeles pertenecen a un cuerpo captado por la cámara y cuales no.
<b>HandLeftState</b>	Matriz 1x6 la cual identifica el posible estado de la mano izquierda: 0 Indefinido 1 Mano no detectada 2 Abierta 3 Cerrada 4 Entrelazada
<b>HandRightState</b>	Matriz 1x6 la cual indica lo mismo que la anterior pero referente a la mano derecha.
<b>HandLeftConfidence</b>	Matriz 1x6 la cual define la calidad de seguimiento de la mano izquierda. 0 Baja 1 Alta
<b>HandRightConfidence</b>	Matriz 1x6 la cual indica lo mismo que la anterior pero referente a la mano derecha.
<b>JointTrackingStates</b>	Matriz 25x6 que define la detección de cada una de las articulaciones con valores: 0 No detectado 1 Con interferencias 2 Detectado
<b>JointPositions</b>	Matriz 25x3x6 con coordenadas X, Y, Z que indican la posición en un espacio 3D de las 25 articulaciones detectadas por la

Tabla 4 Resumen metadata Image Acquisition Toolbox

### 3.2.2 *Librería Image processing toolbox*

Image processing toolbox[4] se utilizará para el posterior procesamiento de las imágenes, ya que lo que se busca no solo es interpretar internamente los datos sino plasmar de una forma visual e intuitiva todo aquello que se está captando, facilitando así al usuario final la tarea del reconocimiento gestual.

Esta librería proporciona un conjunto de algoritmos y apps para el procesamiento, el análisis y la visualización de imágenes. Se puede llevar a cabo segmentación de imágenes, mejora de las mismas reduciendo el ruido, transformaciones geométricas así como procesamiento de imágenes en 3D.

En este proyecto se ha hecho uso de esta librería para realizar tareas como la representación de los puntos de las articulaciones, las uniones entre los puntos a modo de esqueleto así como la mejora de diversos aspectos gráficos del interface.

Las imágenes que proporciona la cámara Kinect son de una altísima calidad y por tanto no se ha necesitado de ningún procesamiento para las mismas.

## Capítulo 4. Resolución del algoritmo

En este apartado se hablará sobre la resolución tanto del algoritmo principal como de las funciones secundarias creadas a raíz de este. El programa se puede dividir en cuatro partes bien diferenciadas, tres de las cuales pertenecen al algoritmo principal o base del programa y una que han sido creada a partir de funciones secundarias.

Las diferentes partes del algoritmo son las siguientes:

1. Inicialización
2. Captación de frames
3. Visualización del esqueleto
4. Detección de gestos e interpretación

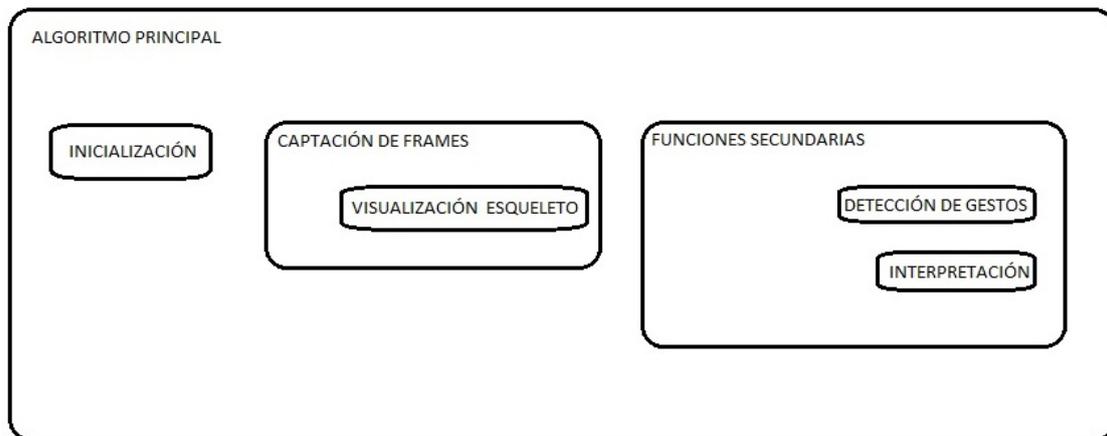


Figura 12: Diagrama de bloques del algoritmo

### 4.1 Algoritmo principal

Se ha llamado algoritmo principal a la base del programa, función de la cual nacen todas las demás. A continuación se explicará la implementación del algoritmo comentando algunas funciones y algunos trozos de código que se consideran de gran importancia para el buen funcionamiento del programa o que han sido de mayor dificultad en el proceso de desarrollo.

#### 4.1.1 Inicialización

Como se ha comentado anteriormente la inicialización es el primer paso que toma el programa. Consiste en inicializar y configurar las funciones de la Kinect que se quieren tener activas guardándolas en variables para utilizarlas posteriormente.

Cómo se ha comentado en el apartado 3.1.1 de este proyecto la cámara Kinect tiene tres tipos de video, Full HD, detector de profundidad y cámara infrarroja, los cuales hay que inicializar y guardar en variables al comienzo del programa, ya que estos contienen datos importantes sobre lo que la cámara capta en cada momento.

```
depthVid = videoinput('kinect', 2);  
colorvid = videoinput('kinect', 1);
```

La anterior instrucción[9] guarda en la variable `depthVid` toda la información proporcionada por la cámara numero 2 que corresponde al sensor de profundidad y en la

variable `colorvid` la información referente a la cámara número 1, la cual corresponde a la cámara fullHD

```
triggerconfig(depthVid, 'manual');  
depthVid.FramesPerTrigger = 1;  
depthVid.TriggerRepeat = inf;
```

Es importante configurar cómo se quiere que Matlab represente la información en cuanto a los frames. En este caso se ha especificado que el disparador de frames de la cámara de profundidad se configurará manualmente y posteriormente se especifica que cada disparo tendrá un frame y que se repetirá infinitamente. Esto permitirá al programa capturar los datos que proporciona el sensor de profundidad cada vez que cambie el frame y además permitirá a Matlab la generación de un video en vivo utilizando estos frames en forma de película.

Dentro de las opciones previamente programadas de las que dispone el sensor de profundidad está el `BodyTracking` o seguimiento del cuerpo, la cual interesa activar ya que es la que proporciona la información de la posición en el espacio de los 25 puntos corporales que se necesitarán para captar los gestos. Por tanto, se indicará que la función está activa de la siguiente forma:

```
srcDepth = getselectedsource(depthVid);  
srcDepth.EnableBodyTracking = 'on';
```

Como se aprecia, el primer paso es guardar en una variable la fuente de datos, en este caso en la variable `srcDepth` y el segundo paso es activar la función `BodyTracking`.

Para finalizar la inicialización se inicia el proceso del sensor de profundidad con un `start`:

```
start(depthVid);
```

Con esta función el proceso empieza a funcionar manteniendo la configuración que se le ha especificado anteriormente.

## 4.2 Captación de frames

Se pretende crear una interfaz gráfica en la cual aparezcan dos videos en tiempo real. El primero pertenece a la cámara del sensor de profundidad, superponiendo en el dibujo los puntos que se detectan, unidos dando la forma del esqueleto humano. El segundo pertenece a la cámara full HD, para proporcionar una visión de más detalle. Para ello se necesitará captar los frames de cada una de las cámaras y mostrarlos por pantalla en tiempo real creando así dos videos.

Dentro de un `while` infinito es decir, que se repita periódicamente, se programa lo siguiente[10]:

```
[depthMap, depthMetaData] = getsnapshot(depthVid);  
[image, imageMetaData] = getsnapshot(colorvid);
```

Estas dos líneas guardan los “metadata” de cada una de las dos cámaras en diferentes variables para utilizarlos más adelante. El siguiente paso será mostrar por pantalla las dos imágenes de las cámaras, las cuales hemos guardado en dos variables diferentes descritas en las anteriores dos líneas, llamadas `depthMap` y `image`. Se muestran de la siguiente forma[4]:

```
axes(handles.axes2)
imshow(image, [0 4096]);
axes(handles.axes1)
imshow(depthMap, [0 4096]);
```

De esta manera se captan las imágenes y se muestra cada una en una figura de la interfaz, captando así periódicamente un frame diferente para crear dos videos en tiempo real.

#### 4.2.1 Visualización del esqueleto

De momento ya se pueden observar dos videos en vivo. Una vez conseguido esto, el siguiente paso es visualizar el esqueleto impreso directamente sobre el video de una forma visual e intuitiva. Para ello el primer paso es saber si la cámara ha detectado un cuerpo y en caso afirmativo actuar:

```
haycuerpo = depthMetaData.IsBodyTracked;
[i,detectado] = cuerpodetectado(haycuerpo);
```

Se crea la variable haycuerpo y se utiliza el metadata explicado anteriormente en la tabla numero 4 “IsBodyTracked”, de forma que si se detecta un cuerpo la variable detectado cambiará a 1 y si no se detecta permanecerá a 0. Esto nos permite crear una condición basada en si se ha detectado o no el cuerpo empleando la variable detectado .

En caso de que se detecte un cuerpo, se procede a captar y guardar en una variable los datos referentes a todas las articulaciones detectadas. Esto se consigue accediendo a los metadata “DepthJointIndices” y “JointPositions” explicados anteriormente en la tabla numero 4:

```
cuerpo = depthMetaData.DepthJointIndices(:, :, i);
```

Se guardan las coordenadas de cada articulación en la variable cuerpo. Los pasos siguientes son representar las coordenadas guardadas en la variable cuerpo sobre los frames que forman el video y seguidamente unirlos en forma de esqueleto para que quede más visual. Para ello se han creado dos funciones “pintacuerpo” y “esqueleto”, las cuales se explican a continuación:

##### 4.2.1.1 Función pintacuerpo

La función pintacuerpo utiliza las coordenadas X e Y de las posiciones de las articulaciones. Para poder representarlas se realiza un bucle for recorriendo los 25 puntos y se utiliza un plot para cada una de las coordenadas de la forma que se ve a continuación:

```
function pintacuerpo(datoscuerpo)

hold on %El hold on se realiza para superponer los puntos que se crean a las imágenes del video.

for (i=1:25)

    plot (datoscuerpo(i,1),datoscuerpo(i,2),'*g')
```

```
end
```

#### 4.2.1.2 Función esqueleto

La función esqueleto crea líneas entre las diferentes coordenadas dándole forma de esqueleto humano. Como se observa a continuación se utiliza la función “line” y se divide en 5 partes: unión de tronco, de brazo izquierdo, de brazo derecho, de pierna derecha y de pierna izquierda.

```
function esqueleto( cuerpo )  
  
%Union tronco  
line(cuerpo([3,4],1), cuerpo([3,4],2));  
line(cuerpo([3,2],1), cuerpo([3,2],2));  
line(cuerpo([2,1],1), cuerpo([2,1],2));  
%Union brazo Izquierdo  
line(cuerpo([3,5],1), cuerpo([3,5],2));  
line(cuerpo([5,6],1), cuerpo([5,6],2));  
line(cuerpo([6,7],1), cuerpo([6,7],2));  
line(cuerpo([7,8],1), cuerpo([7,8],2));  
%Union brazo derecho  
line(cuerpo([3,9],1), cuerpo([3,9],2));  
line(cuerpo([9,10],1), cuerpo([9,10],2));  
line(cuerpo([10,11],1), cuerpo([10,11],2));  
line(cuerpo([11,12],1), cuerpo([11,12],2));  
%Union pierna derecha  
line(cuerpo([1,17],1), cuerpo([1,17],2));  
line(cuerpo([17,18],1), cuerpo([17,18],2));  
line(cuerpo([18,19],1), cuerpo([18,19],2));  
line(cuerpo([19,20],1), cuerpo([19,20],2));  
%Union pierna izquierda  
line(cuerpo([1,13],1), cuerpo([1,13],2));  
line(cuerpo([13,14],1), cuerpo([13,14],2));  
line(cuerpo([14,15],1), cuerpo([14,15],2));  
line(cuerpo([15,16],1), cuerpo([15,16],2));  
  
end
```

### 4.3 Funciones secundarias

#### 4.3.1 calculadist

La función calculadist está creada para calcular la distancia euclidiana de las coordenadas en tres dimensiones que proporciona la Kinect.

La distancia euclidiana entre los puntos  $P = (p_1, p_2, \dots, p_n)$  y  $Q = (q_1, q_2, \dots, q_n)$  del espacio euclídeo n-dimensional se define como:

$$d_E(P, Q) = \sqrt{(p_1 - q_1)^2 + (p_2 - q_2)^2 + \dots + (p_n - q_n)^2}$$

En el caso de este proyecto las coordenadas obtenidas vienen dadas en una matriz de tres dimensiones con variables (X, Y, Z), por tanto la ecuación será la siguiente:

$$d_E(P, Q) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + (z_1 - z_2)^2}$$

A continuación se define el algoritmo de la función calculadist en Matlab:

```
function [ distancia ] = calculadist( xuno, xdos, yuno, ydos, zuno,
zdos )
distancia = sqrt((xuno-xdos)^2+(yuno-ydos)^2+(zuno-zdos)^2);
end
```

Como se aprecia calculadist no es una función muy compleja, se le entregan las tres coordenadas de los dos puntos que se quiere calcular la distancia y devuelve la variable “distancia”, la cual es el resultado de la ecuación anteriormente descrita.

### 4.3.2 *Cuerpodetectado*

Esta función tiene como objetivo saber si la cámara Kinect ha reconocido un cuerpo o no, con el fin de comenzar el algoritmo de detección solo si la cámara ha detectado el cuerpo.

Para ello se hace uso del metadata “IsBodyTracked” explicado en la tabla numero 4 en el apartado 3.2.1 de este documento, mediante el cual se obtiene una matriz binaria 1x6, indicando 1 cuando uno de los seis cuerpos es detectado y 0 si no es detectado. La función está construida como sigue:

```
function [ i, detection ] = cuerpodetectado( haycuerpo )

for (i=1:6)
    if haycuerpo(1,i) ~= 0
        detection = 1;
        break
    else
        detection = 0;
    end
end
end
```

Se crea un bucle for el cual recorre toda la matriz comprobando si existe alguna unidad. en el caso de respuesta afirmativa se guarda un 1 en la variable detection y se termina el algoritmo. En caso negativo va recorriendo toda la matriz hasta finalizar y guarda un 0 en la variable descrita anteriormente.

## 4.4 **Funciones para la detección de gestos**

### 4.4.1 *Función codocogido*

Esta función detecta una de los gestos mas comunes de inseguridad, el autoabrazo (Imagen 5). Para ello se crea un algoritmo que mide la distancia entre la mano derecha y el codo izquierdo, llegando en este caso al compromiso de si la distancia entre ellos es menor que 0.07m se considera que están juntos y por tanto es un autoabrazo. El algoritmo se divide en dos partes.

1. Se recogen las medidas de la mano derecha y del codo izquierdo y se calcula la distancia (Se realiza igual para todas las funciones de detección de gestos).
2. Se establece la condición  $\text{distancia} < 0.07\text{m}$  autoabrazo,  $\text{distancia} > 0.07\text{m}$  no existe autoabrazo.

#### Parte 1

```
function [ interpretacion ] = codocogido( profundidad )
interpretacion = 0;

Xcodoiz = profundidad(6,1);
Xmanoder = profundidad(24,1);

Ycodoiz = profundidad(6,2);
Ymanoder = profundidad(24,2);

Zcodoiz = profundidad (6,3);
Zmanoder = profundidad (12,3);

interpretacion
=calculadist(Xcodoiz,Xmanoder,Ycodoiz,Ymanoder,Zcodoiz,Zmanoder);
```

#### Parte 2

```
codo =codocogido(profundidad);

if codo < 0.07

set(handles.edit3,'String','DETECTADO','BackgroundColor','red');

else

set(handles.edit3,'String','NODETECTADO','BackgroundColor','white');

end
```

#### 4.4.2 *Función distanciacodos*

Como se ha descrito anteriormente la distancia entre los brazos juega un papel muy importante y puede revelar mucho en cuanto al estado del interlocutor. En esta función se mide la distancia entre los brazos y para ello se han elegido los puntos correspondientes a los codos. La forma del algoritmo es la misma que la del apartado anterior, primero se realizará la medida entre los dos codos y más tarde se llega al siguiente compromiso:

- Actitud insegura: distancia entre codos menor que 0.37m
- Actitud normal: distancia entre codos mayor que 0.37m y menor que 0.50m
- Actitud segura o agresiva: distancia entre codos mayor que 0.50m

```
codos = distanciacodos(profundidad);

if codos < 0.37
```

```

set(handles.edit2,'String','INSEGURIDAD','BackgroundColor','red');

    else if codos > 0.37 && codos < 0.50
        set(handles.edit2,'String','NORMAL','BackgroundColor','blue');
    else if codos > 0.50

set(handles.edit2,'String','SEGURIDAD','BackgroundColor','green');

    else

set(handles.edit2,'String','NODETECTADO','BackgroundColor','black');

        drawnow;

    end

```

#### 4.4.3 *Función Distanciainterlocutor*

El objetivo de esta función es calcular la distancia que hay desde el interlocutor hasta la audiencia, puesto que según la distancia a la que se encuentre puede ser un indicativo de que el sujeto esta seguro o por el contrario inseguro. Para realizarlo se ha tenido en cuenta solamente el eje Z el cual mide la profundidad y se ha elegido el punto nº2 del esqueleto descrito en la tabla 1, que corresponde al punto medio de la espina dorsal. Por tanto el algoritmo es el siguiente:

```

dinterlocutor = distanciainterlocutor(profundidad);

if dinterlocutor > 2
    set(handles.edit8,'String','INSEGURIDAD','BackgroundColor','red');
else if dinterlocutor > 1.5 && dinterlocutor < 2
    set(handles.edit8,'String','NORMAL','BackgroundColor','blue');
    else if dinterlocutor < 1.5

set(handles.edit8,'String','SEGURIDAD','BackgroundColor','green');

    else

set(handles.edit8,String','String','NODETECTADO','BackgroundColor','black');

    end

```

Esta vez se ha establecido

- Distancia de inseguridad: mayor de 2m
- Distancia normal: de 2m a 1.5m

- Distancia persona agresiva: menor de 1.5m

Cabe destacar que estas distancias dependen de la amplitud de la sala y se pueden configurar según convenga.

#### 4.4.4 *Función Distanciamanos*

Otro aspecto a tener en cuenta es cuando el individuo tiene las manos juntas, esto es un indicativo de inseguridad. Un tiempo prolongado con las manos juntas mientras se está exponiendo algo al público junto con otros factores puede transmitir una situación de inseguridad. En el algoritmo distanciamanos se mide la distancia entre la mano izquierda y la derecha. La estructura de la función es la misma que anterior, se calcula la distancia entre los puntos 12 y 8 correspondientes a la mano derecha y la mano izquierda respectivamente. Se establece que si la distancia entre las dos manos es menor de 0.03m las manos están juntas y por tanto se activa el indicador de “inseguridad”.

```
dmanos = distanciamanos(profundidad);

if dmanos < 0.03

    set(handles.edit9,'String','MANOSJUNTAS','BackgroundColor','red');
else
set(handles.edit9,'String','MANOSSEPARADAS','BackgroundColor','green');
end
```

#### 4.4.5 *Funcion Distanciapiernas*

Siguiendo la misma dinámica que en las funciones anteriores, esta vez se va a medir la distancia entre las piernas, la cual se considera indicativo de inseguridad si se repliegan, o de seguridad si se expanden demasiado. En esta función se han elegido los puntos pertenecientes a tobillo derecho (punto 19) y tobillo izquierdo (punto 15). Se mide la distancia entre ellos de forma análoga a las demás funciones y se establece la siguiente condición:

```
piernas = distanciapiernas(profundidad);

if piernas < 0.1

    set(handles.edit7,'String','INSEGURIDAD','BackgroundColor','red');
else if piernas > 0.1 && piernas < 0.4

    set(handles.edit7,'String','NORMAL','BackgroundColor','blue');
else if piernas > 0.4

set(handles.edit7,'String','SEGURIDAD','BackgroundColor','green');
end
```

## Capítulo 5. Interfaz gráfica de usuario

### 5.1 Introducción.

En este programa se ha creado una interfaz gráfica intuitiva y fácil para el usuario final. Así pues, todos los cálculos de distancias que se realizan son procesados internamente para mas tarde mostrar los resultados de una forma gráfica.

Para la interpretación de los resultados se ha elegido un código de color:

- Rojo : Comportamiento inseguro o temeroso.
- Azul: Comportamiento normal, no suscita sospechas claras ni de inseguridad ni de seguridad.
- Verde: Comportamiento seguro o agresivo.

De forma que cuando el programa detecta ítems que correspondan a cualquiera de los comportamientos anteriormente mencionados emite su color correspondiente. Con este sistema se puede percibir de una forma visual todo lo que el programa va captando.

### 5.2 Detalle de la interfaz.

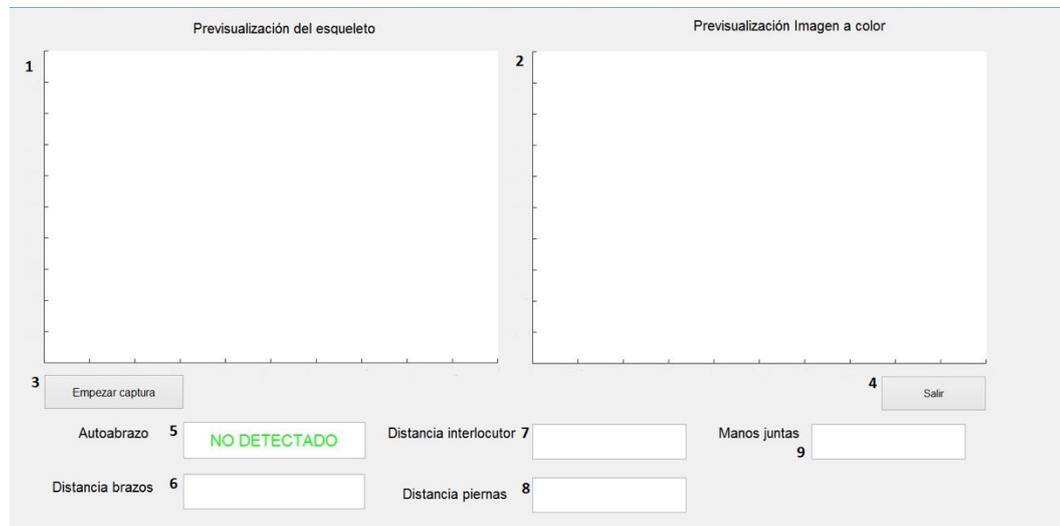


Figura 13 Interfaz gráfica del programa

En la pantalla principal se observan dos botones, “Empezar captura”(3) y “Salir”(4). Cuando se presiona el botón iniciar, como su nombre indica se inicia todo el proceso de captación de gestos, el programa arranca dos videos en tiempo real, el video del sensor de profundidad(1) y el correspondiente al sensor de video Full HD(2). Con esto se pretende que el usuario pueda ver en todo momento que es lo que está sucediendo en la sala con detalle y que información está captando la Kinect. Además de esto, cuando el programa está iniciado ejecuta constantemente todos los algoritmos definidos

anteriormente en el capítulo 4, realizando todos los cálculos mientras la persona está exponiendo.

**(5)** indica si se ha detectado o no el gesto de inseguridad denominado Autoabrazo.

**(6),(7)** y **(8)** Calcula las distancias entre brazos, locutor y piernas con un algoritmo y expresa en terminología de colores en que estado puede que se encuentre la persona detectada.

**(9)** Un signo de inseguridad que es importante realizar lo menos posible es juntar las manos durante un tiempo prolongado mientras se habla, en este apartado del programa se mide la distancia entre las manos y se informa sobre si las manos están juntas o no.

## Capítulo 6. Análisis de resultados

### 6.1 Introducción

En este apartado se va a comprobar el funcionamiento del programa. Para ello se han elegido 3 sujetos de diferente edad y diferente estatura los cuales se expondrán al análisis del programa para posteriormente analizar los resultados obtenidos. Seguidamente se comprobará si los resultados obtenidos son satisfactorios o no y se expondrá cómo corregir estos fallos.

Las características de los tres sujetos sometidos a las pruebas son las siguientes:

- Persona 1: de 28 años y 1.70m de altura.
- Persona 2: de 54 años y 1.55m de altura.
- Persona 3: de 55 años y 1.73m de altura.

### 6.2 Pruebas de funcionamiento

#### 6.2.1 Posiciones de inseguridad y cierre



Figura 14 Posición auto abrazo persona 1



Figura 15 Posición auto abrazo persona 2



Figura 16 Posición indecisión persona 3



Figura 17 Posición cierre persona 2



Figura 18 Posición auto abrazo persona 3



Figura 19 Posición tímida persona 2

### 6.2.2 Posiciones de seguridad o agresividad



Figura 20 Posición seguridad persona 1

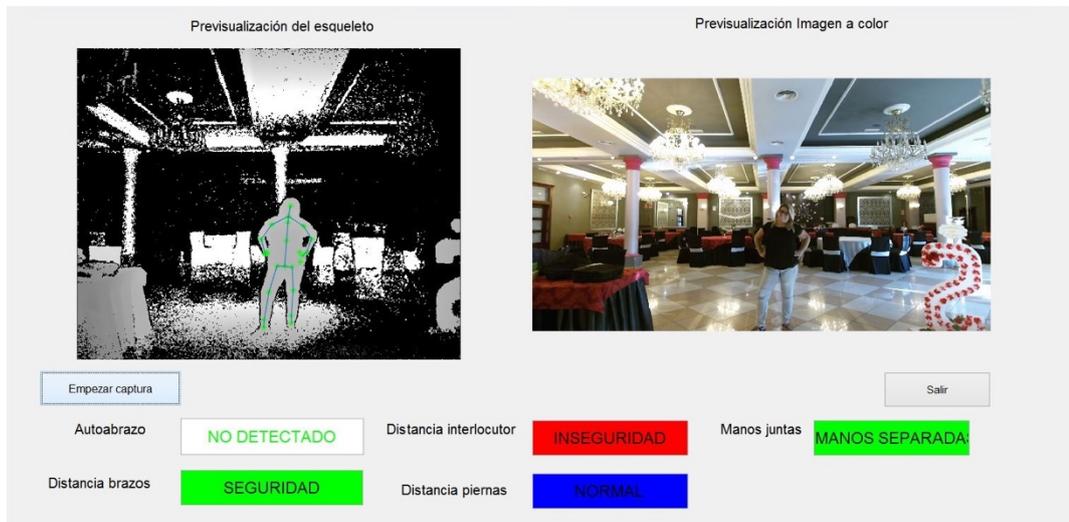


Figura 21 Posición seguridad persona 2



**Figura 22 Posición seguridad persona 3**



**Figura 23 Posición apertura y confianza persona 3**

### 6.3 Análisis de las pruebas

Los ítems que se ha decidido detectar:

- Autoabrazo.
- Distancia entre los brazos.
- Distancia del interlocutor.
- Distancia de piernas.
- Posición de las manos (Juntas o separadas).

Cabe destacar que todas las distancias establecidas como límite para la elección de un ítem u otro se han realizado a criterio personal y sin asesoramiento de ninguna persona con conocimientos de sinergología. Por lo tanto es posible que los resultados no sean del todo rigurosos con la disciplina. Aún así se ha realizado teniendo en cuenta las indicaciones del autor en el libro de María José Arlandis[1].

### **6.3.1 *Resultados de las pruebas para posiciones de inseguridad y cierre***

Respecto a las pruebas del autoabrazo (fotografías 14,15 y 18) el programa detecta varios signos de inseguridad en todos los casos. Se detectan los pies demasiado juntos, se detectan los brazos pegados al cuerpo y la distancia del interlocutor demasiado alejada del oyente. Además en dos de los tres casos se detecta autoabrazo (fotografías 14 y 15) mientras que para el sujeto de la fotografía 18 no. Esto indica que la precisión a la hora de detectar este ítem no es lo suficientemente correcta para todos cuerpos y posiblemente se debería de escalar los límites atendiendo a la altura de cada sujeto.

Respecto a la fotografía número 16 en la que se intenta representar una postura de indecisión e inseguridad, se puede observar que el programa mide bien todos los ítems menos la distancia entre las piernas. La solución a este problema podría ser como anteriormente el escalado según la altura.

Para los demás casos (fotografías 17 y 19) se observa que el programa funciona bien indicando todos los signos de inseguridad.

### **6.3.2 *Resultados de las pruebas para posiciones de seguridad o agresividad***

Al analizar la posición de seguridad en uno mismo “brazos en jarras” (fotografías 20,21 y 22) se observa que en la fotografía 20 se detectan todos los ítems correctamente mientras que en las fotografías 21 y 22 los sujetos no abren tanto las piernas y por tanto el programa detecta una apertura normal. Para solucionar este aspecto sería interesante contar con el asesoramiento de una persona con buenos conocimientos en sinergología y cambiar los límites de distancia.

En la fotografía numero 23 se ha emulado una posición de confianza en uno mismo, por tanto es un ítem de apertura. En este caso el programa funciona perfectamente iluminando todos los indicadores de seguridad.

## **Capítulo 7. Conclusiones y líneas futuras**

Una vez realizadas todas las pruebas se llega a la conclusión de que el programa consigue detectar algunos gestos de la escultura del cuerpo y consigue clasificarlas bien. Por tanto cumple con su cometido. No obstante, al tratarse de una versión beta tiene algunas carencias y algunos aspectos que podrían ser mejorados en el futuro.

Sería necesaria la colaboración activa de una persona con conocimientos de sinergología con el fin de poder afinar y perfeccionar la detección de gestos, tener en cuenta aspectos como la longitud exacta entre las piernas, el tiempo en el que la persona tiene las manos juntas o las veces que realiza un autoabrazo podría mejorar la calidad y la experiencia del programa.

Por otro lado existe la interesante posibilidad de agregar diferentes módulos al programa como el reconocimiento facial para tener en cuenta los microgestos, el reconocimiento de voz o incluso un detector de estrés que mida las pulsaciones por minuto del sujeto. Esto le daría una vuelta de tuerca mas y proporcionaría información muy valiosa y variada al usuario final.

El programa está abierto a infinidad de posibilidades para enriquecerlo y pulirlo. Son muchas las opciones que da la Kinect y partiendo de la base que esta primera versión del programa ofrece se pueden realizar muchos proyectos interesantes.

## Capítulo 8. Bibliografía

- [1] Arlandis, M.J.(2012). *Los gestos hablan*. Barcelona: Plataforma
- [2] Turchet, P.(2010). *El lenguaje del cuerpo* Bilbao: Mensajero.
- [3] Pease, A.(2010). *El lenguaje del cuerpo: Como interpretar a los demás a través de sus gestos*. España: Amat editorial
- [4] Pratt W.K.(1978). *Digital Image processing*. Toronto: Wiley & Sons
- [5] Domingo, A.(1993). *Tratamiento Digital de Imágenes* Madrid: Anaya
- [6] Russ J.C.(1999). *The Image processing Handbook* Heidelberg: Springer-Verlag
- [7] McAndrew A.(2004). *An Introduction to Digital Image Processing with Matlab*. Disponible en: <http://www.math.hkbu.edu.hk/~zeng/Teaching/math3615/ls.pdf> [Consulta: 2017, 5 de Julio].
- [8] Thyagarajan K.S.(2010). *Still Image and Video Compression with MATLAB* John Wiley & Sons.
- [9] The Math Works Inc.(2016). *Image acquisition toolbox User's Guide*. Disponible en: [http://www.uow.edu.au/~phung/resources/image/imaq\\_ug.pdf](http://www.uow.edu.au/~phung/resources/image/imaq_ug.pdf) [Consulta: 2017, 30 de Junio]
- [10] Catuhe D.(2012). *Programming with the Kinect for Windows*. Redmond, Washington: Microsoft press.
- [11] Wasenmüller O. & Stricker D.(2016). *Comparison of Kinect v1 and v2 depth images*. Disponible en: <file:///Users/Josep/Downloads/wasenmuller2016comparison.pdf> [Consulta: 2017, 3 de Agosto]
- [12] Cong R. & Winters R.(2017). *How does Xbox Kinect work*. Disponible en: <http://www.jameco.com/jameco/workshop/howitworks/xboxkinect.html> [Consulta: 2017 6 de junio].

