

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ

Escuela de Posgrado



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

DISEÑO E IMPLEMENTACIÓN DE UN SISTEMA DE VISIÓN PARA LA ASISTENCIA DE PERSONAS CON DISCAPACIDAD

Tesis para optar al grado de Magister en Ingeniería Mecatrónica

Presentado por:

ANTONIO LINCOLN ANGULO SALAS

Asesorado por:

M.Sc. ERICKA PATRICIA MADRID RUIZ

Marzo 2016

Lima - Perú



© 2016, Antonio Lincoln Angulo Salas

Se autoriza la reproducción total o parcial,
Con fines académicos a través de cualquier
Medio o procedimiento, incluyendo la cita
Bibliográfica del documento.

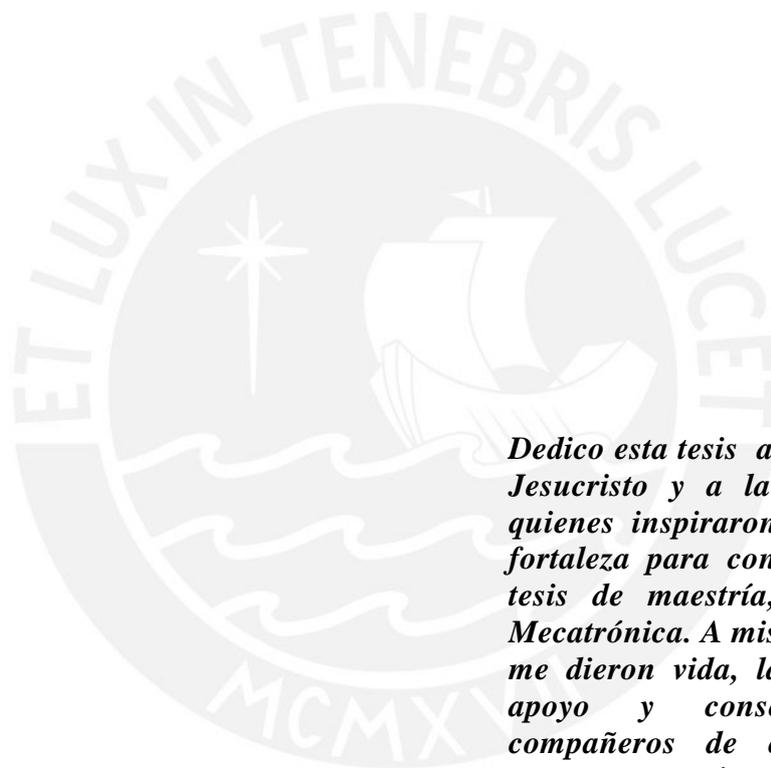
RESUMEN

Alrededor del mundo el 15% de personas viven con alguna condición de discapacidad según el Reporte Mundial en Discapacidad de la Organización Mundial de la Salud. Estas condiciones les impiden desenvolver su vida con normalidad. Por eso en las últimas décadas los gobiernos han empezado a desarrollar normas y programas sobre espacios y herramientas públicas acondicionados para que las personas con discapacidad puedan valerse por sí mismas. La creación de estos espacios ha contribuido al bienestar de las mismas fomentando la participación de las personas con discapacidad como ciudadanos.

Lo que pretende este trabajo de tesis es dar valor agregado a los métodos computacionales de procesamiento de señales y a los instrumentos robóticos móviles aplicándolos a la asistencia de personas con discapacidad. Actualmente existen diversas herramientas tecnológicas que les brindan apoyo y facilitan las labores que desempeñan. Por ejemplo, Sillas de ruedas motorizadas que suben escaleras, robots que ayudan a pasar de una cama a otra a los pacientes de los hospitales evitando así luxaciones de los enfermeros y médicos, etc.

Específicamente se pretende manipular un brazo robótico mediante un sistema de visión que reconozca movimiento articulares de la cabeza y algunos gestos de la cara. Este sistema está especialmente diseñado para personas que han perdido la movilidad en las extremidades y solo puedan mover la cabeza y hacer algunas expresiones faciales. Todo esto mediante una plataforma de desarrollo que tendrá conectada una cámara que captura imágenes del rostro del usuario y a través de algoritmos de visión puedan comandar el brazo **Jaco** de la marca *Kinova*. En el proceso se diseñaron e implementaron la base de datos que sirvió para el reconocimiento de gestos y el sistema electrónico. Se establecieron también límites de seguridad para que el brazo robótico no invadiese zonas del usuario. Se hicieron los cálculos de potencia, consumo y velocidad, y también se analizaron los resultados experimentales en diferentes usuarios de los cuales obtuvimos las curvas de aprendizaje.

Los resultados indicaron que el algoritmo de procesamiento tiene un 89.2% de precisión al comandar el brazo robótico y que el usuario puede controlar el brazo robótico con un retardo de 440 milisegundos. Con estos datos concluimos que es posible usar y controlar dispositivos robóticos mediante movimientos de la cabeza y expresiones faciales sirviendo de apoyo a personas con discapacidad motora de manera rápida y segura.



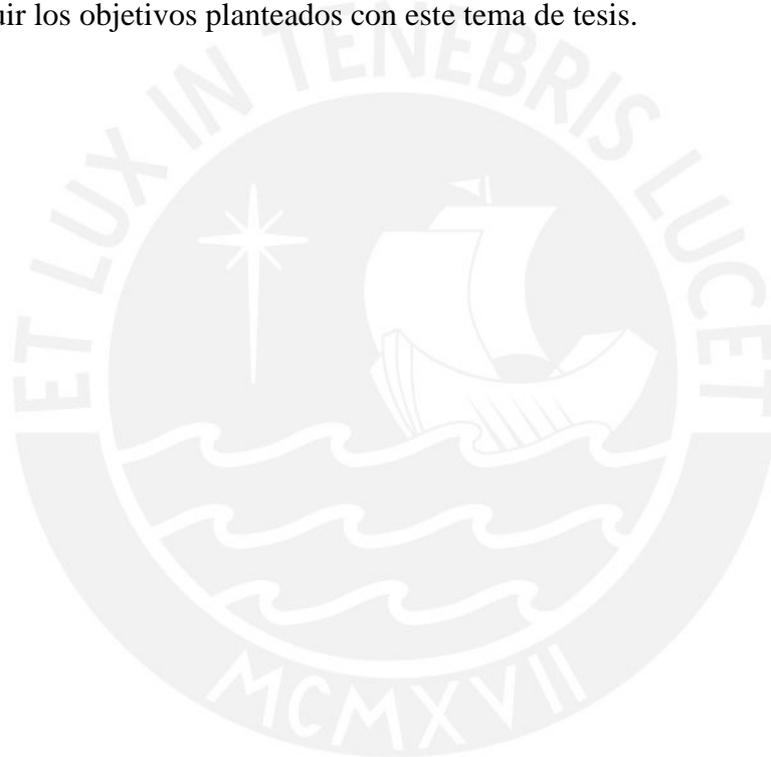
Dedico esta tesis a DIOS, a su hijo Jesucristo y a la Virgen María, quienes inspiraron mi espíritu de fortaleza para continuar con esta tesis de maestría, en Ingeniería Mecatrónica. A mis padres quienes me dieron vida, la fe, educación, apoyo y consejos. A mis compañeros de estudio, a mis maestros y amigos, quienes sin su ayuda nunca hubiera podido hacer esta tesis. A todos ellos se los agradezco desde el fondo de mi alma y mi corazón. Para todos ellos hago esta dedicatoria.

AGRADECIMIENTOS

Al FONDECYT y al PAIP-PUCP que financiaron la implementación y culminación de mi tema de tesis: “Diseño e implementación de un sistema de visión para la asistencia de personas con discapacidad”.

A todas las personas que colaboraron con su tiempo para realizar las pruebas del funcionamiento del sistema de visión.

A mi asesora a la profesora Ericka Madrid que me guio y sirvió su tiempo para conseguir los objetivos planteados con este tema de tesis.



Índice de contenido

	Pág.
RESUMEN.....	iii
DEDICATORIA.....	iv
AGRADECIMIENTOS.....	v
Índice de contenidos.....	vi
Índice de tablas.....	viii
Índice de figuras.....	ix
<i>INTRODUCCIÓN</i>	1
CAPÍTULO 1: Contextualización de las personas con discapacidad en la sociedad .	3
1.1 Personas con discapacidad en la sociedad	3
1.2 Trastornos en las personas con discapacidad	4
1.3 Problemática.....	4
CAPÍTULO 2: Antecedentes	6
2.1 Sistemas robóticos de asistencia	6
2.2 Sistemas de Visión	9
2.3 Detección y reconocimiento gestos.....	11
2.4 Interacción humano-robot	15
CAPÍTULO 3: Desarrollo del sistema de visión para la asistencia de personas con discapacidad	17
3.1 Diseño conceptual del funcionamiento del sistema	17
3.2 Base de datos.....	18
3.2.1 Consideraciones técnicas	
3.2.2 Herramientas y consideraciones metodológicas	
3.2.3 Implementación	
3.3 Algoritmo del sistema de visión.....	23
3.3.1 Selección del algoritmo	
3.3.2 Descripción del algoritmo	
3.4 Mecanismo de asistencia.....	30
3.4.1 Unidad robótica	

3.4.2 Programación del sistema robótico	
3.5 Sistema electrónico	34
3.5.1 Requerimientos de la plataforma de desarrollo	
3.5.2 Selección de cámara	
3.5.3 Alimentación del sistema	
3.6 Interface de usuario y modos de operación.....	38
3.7 Integración del sistema.....	40
CAPÍTULO 4: Pruebas y resultados	41
4.1 Pruebas	41
4.1.1 Comparación de la base de datos	
4.1.2 Análisis de la curva de aprendizaje	
4.1.3 Iluminación del sistema	
4.2 Resultados	46
4.2.1 Curva de aprendizaje	
4.2.2 Base de datos	
CONCLUSIONES	48
ANEXOS	50
Anexo 1: Programa en C++ para el control del brazo robotico con movimientos predefinidos.....	50
Anexo 2: Programa en C++ para del sistema de visión total.....	59
Anexo 3: Programa en C++ para la adquisición de imagenes de la base de datos.....	61

Índice de tablas

	Pág.
Tabla 3.1: Comparación entre algunas plataformas de desarrollo	35
Tabla 3.2: Tabla de potencias del sistema de visión implementado en PcDuino V3	37



Índice de figuras

	Pág.
Figura 2.1: Silla de ruedas motorizada de la marca Jiangnan Machinery Co., Ltd.....	7
Figura 2.2: Brazo robótico de la marca <i>Kinova</i> montado en una silla de ruedas.....	8
Figura 2.3: Robot humanoide de asistencia <i>Robear caregiver</i>	8
Figura 2.4: Diagrama de bloques de un sistema de visión	9
Figura 2.5: Diagrama de flujo para el reconocimiento de un objetivo	11
Figura 2.6: Acciones unitarias y algunas uniones básicas de los FACS.....	12
Figura 2.7: Resultados del tracking usando el detector de bordes Canny.....	14
Figura 2.8: Piezas de volúmenes de Biezer en el rostro durante el tracking	15
Figura 3.1: Ejes de rotación del cuerpo humano.....	18
Figura 3.2: Espacio de trabajo y herramientas para la recopilación de la base de datos	22
Figura 3.3: Algunas imágenes de la base de datos.....	22
Figura 3.4: Taza de error de Eigenfaces (Rojo), Eigenfaces sin sus tres primeras principales (Azul) y Fisherfaces (Verde)	24
Figura 3.5: Diagrama de bloques del algoritmo de reconocimiento de patrones.....	24
Figura 3.6: Parches Haar-like features	25
Figura 3.7: Parche aplicado en la imagen del rostro de una persona	25
Figura 3.8: Ejemplo de la obtención de la integral de una imagen	26
Figura 3.9: Calculo de la suma de pixeles en un área en una imagen integral	27
Figura 3.10: Esquemático de la detección cascada	28
Figura 3.11: Brazo Jaco de <i>Kinova</i> y sus partes	30
Figura 3.12: Diagrama de flujo del funcionamiento del brazo robótico.....	31

Figura 3.13: Brazo Jaco de Kinova en su zona de seguridad.....	34
Figura 3.14: Comparación de performance entre los lenguajes de programación....	36
Figura 3.15: Camara seleccionada (Microsoft Lifecam HD-3000)	37
Figura 3.16: Batería seleccionada (Batería LiPo 8000mA/h - 3 celdas).....	38
Figura 3.17: Seguimiento del rostro y predicción de un gesto por parte del sistema de visión.....	39
Figura 3.18: Configuración de las conexiones del hardware del sistema de visión para la asistencia de personas con discapacidad.....	40
Figura 3.19: Implementación física del sistema de visión en el PcDuino V3 integrado con el brazo.....	40
Figura 4.1: Curva de aprendizaje por parte de los usuarios donde T (tiempo de duración de la prueba) y P (# de prueba).....	43
Figura 4.2: Control del brazo robótico mediante movimientos articulares de la cabeza. (a) arriba, (b) neutro, (c) izquierda, (d) derecha, (e) abajo	44
Figura 4.3: Pruebas de iluminación para el funcionamiento del sistema.....	45
Figura 4.4: Tendencia de la curva de aprendizaje.....	46

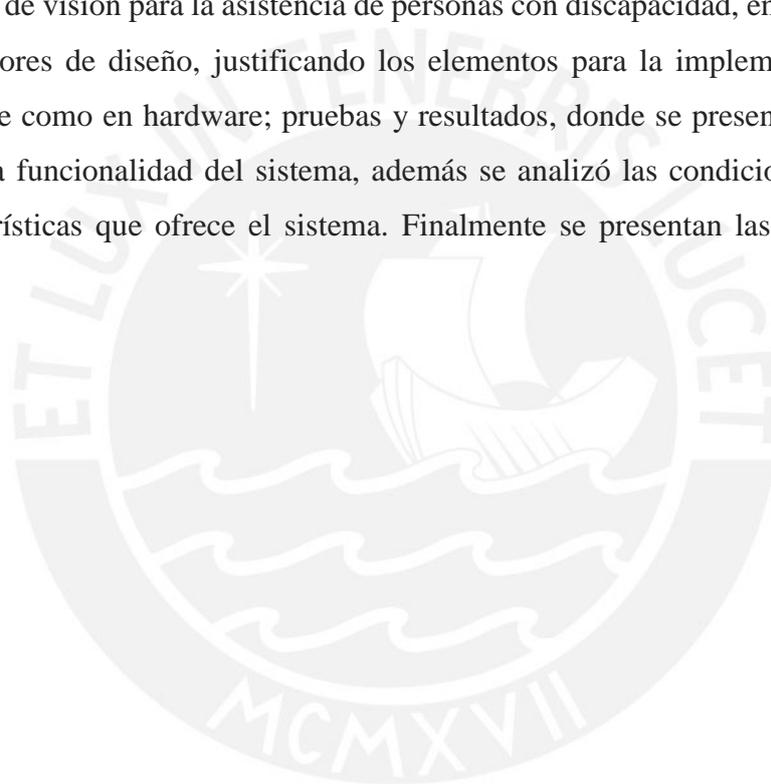
INTRODUCCIÓN

Los sistemas de visión son una herramienta dentro de la robótica que permite obtener información del medio. La cuantificación y evaluación de dicha información puede ayudar a definir una interacción humano-robot adecuada. Para algunas personas con problemas con discapacidad motora este tipo de sistemas pueden menguar los trastornos generados por la falta de autonomía y así desenvolverse naturalmente tanto en la esfera privada como en la pública. Según un trabajo de Serrano Martín y Santos del Riego del 2001 la parálisis cerebral, la polio, la esclerosis, entre otras enfermedades, generan condiciones de discapacidad motora que producen alteraciones musculares, de postura y movimiento, y además presentan trastornos asociados como: visuales, auditivos, cognitivos, emocionales y del lenguaje [1]

El presente trabajo de investigación presenta el desarrollo de un sistema de visión diseñado para la interacción con dispositivos robóticos mediante el reconocimiento de expresiones faciales y movimientos articulares de la cabeza para que sirva de asistencia a personas con discapacidad. Para esto se diseñó e implementó una base de datos de expresiones faciales y movimientos articulares de la cabeza. Se buscó en el estado del arte y se seleccionó el algoritmo de detección y reconocimiento que emplee la mencionada base datos. Estos sirvieron para comandar un dispositivo robótico, el brazo Jaco de la marca Kinova, con movimientos incrementales en un plano y con movimientos complejos predefinidos. Finalmente, se evaluó: La respuesta entre el gesto y la acción del brazo robótico para determinar la precisión gesto-movimiento; el

tiempo de respuesta entre la acción realizada y el movimiento del brazo; y además la capacidad de los usuarios, a través del análisis de la curva de aprendizaje, para controlar el dispositivo.

La tesis comprende cuatro capítulos: Contextualización de las personas con discapacidad en la sociedad, en el cual se presenta la problemática entorno a la asistencia de personas con discapacidad; antecedentes, en donde se resumen algunos trabajos previos en los cuales se presentan tecnologías de asistencia a personas con discapacidad y técnicas de detección y reconocimiento de patrones; desarrollo del sistema de visión para la asistencia de personas con discapacidad, en el cual se analizó los factores de diseño, justificando los elementos para la implementación tanto en software como en hardware; pruebas y resultados, donde se presentan los resultados sobre la funcionalidad del sistema, además se analizó las condiciones de uso, y las características que ofrece el sistema. Finalmente se presentan las conclusiones del trabajo.



CAPÍTULO 1

Contextualización de las personas con discapacidad en la sociedad

Este primer capítulo contextualiza la situación actual de las personas con discapacidad, describiendo la problemática en la que se enmarca el presente trabajo de tesis.

1.1 *Personas con discapacidad en la sociedad*

En la actualidad, más de 200 millones de personas sufren de algún tipo de discapacidad [2] y en el contexto actual existen pocos espacios y herramientas acondicionados para personas con discapacidad. Esto dificulta el desenvolvimiento normal de este tipo de personas.

Hace algunos años atrás se empezó a luchar para que los gobiernos brinden espacios públicos en los cuales las personas con algún tipo de discapacidad puedan desarrollar su vida con total normalidad.

Ed Roberts que fue un activista estadounidense fue uno de los pioneros en luchar por los derechos de las personas con discapacidad logrando establecer normas que ayuden a la convivencia y a la independización, en cualquier ambiente, de las personas con algún tipo de discapacidad.

Actualmente vemos que existen espacios públicos y herramientas diseñados para personas con discapacidades, por ejemplo casetas telefónicas con alturas cortas, elevadores, rampas para subir a las veredas, etc. Este tipo de infraestructuras brindan una mayor comodidad y una mejor calidad de vida a las personas con discapacidad.

1.2 *Trastornos en las personas con discapacidad*

La parálisis cerebral, la polio, la esclerosis, etc., son condiciones de discapacidad motriz que “además de las alteraciones en el tono muscular, la postura y el movimiento, presentan trastornos asociados: visuales, auditivos, cognitivos, emocionales y del lenguaje” [1]. Estos son factores que alteran la comunicación. Para un individuo podría resultar frustrante no tener la atención y la capacidad de expresarse ante los demás como cualquier otro [3]. Por tal motivo la importancia de desarrollar herramientas que faciliten la comunicación de las personas que poseen algún tipo de trastorno motriz.

1.3 *Problemática*

La Organización Mundial de la Salud en un comunicado del 2011 afirma que el 15% de la población mundial posee algún tipo de discapacidad, lo equivalente a más de 1000 millones de personas [4]. Estas personas poseen una desventaja física que les impide realizar sus actividades independientemente, ya que necesitan de una persona que los ayude.

La vigilancia y la asistencia permanente en personas con discapacidad pueden contribuir a mejorar la calidad de vida de las mismas, menguando así los trastornos generados a consecuencia de su incapacidad física. Por lo que, el desarrollo de una herramienta tecnológica ayudaría a la independización del individuo.

En la actualidad se usan sistemas de visión y otras herramientas de sensado y control en diversas aplicaciones; entre ellas la asistencia de personas con discapacidad. Por ejemplo, se puede mencionar el caso de Stephen Hawking renombrado científico británico que se comunica con fluidez a través de un sistema de visión que detecta los

movimientos oculares. También es importante mencionar a Ed Roberts un activista Norteamericano que en su estado de discapacidad luchó por los derechos de las personas como él, en discapacidad, teniendo siempre el apoyo de su silla de ruedas motorizada que le permitía desplazarse con facilidad. Por tal motivo, es de gran importancia estas herramientas robóticas y tecnológicas las cuales día a día permiten dar una mejor calidad de vida a las personas con discapacidad.



CAPÍTULO 2

Antecedentes

En el presente capítulo se describe el estado del arte entorno a este trabajo de tesis. Se presentarán los diversos sistemas y herramientas para la asistencia de personas con discapacidad. Dando así a conocer la realidad y los alcances a los que se aspira en el desarrollo de este tema de tesis.

2.1 *Sistemas robóticos de asistencia*

Actualmente la ingeniería ha desarrollado una gran variedad de sistemas de asistencia. Estos sistemas son variados y se adaptan a la discapacidad de los usuarios. Por lo general, las personas con discapacidad necesitan que estas herramientas les permitan rehabilitarse, comunicarse y/o realizar sus actividades eficazmente. A continuación se mostraran algunos sistemas de asistencia que precisan las personas con discapacidad.

2.1.1 Sillas de ruedas

Las sillas de ruedas son los dispositivos más usados por las personas con discapacidad ya que les permiten desplazarse. Según la OMS el 15% de la población mundial posee algún tipo de discapacidad, esto equivale aproximadamente a más de 1000 millones de personas, de las cuales unos 67 millones usan sillas de ruedas[4].

En la actualidad, con la tecnología las sillas de ruedas han sido modificadas para disminuir el esfuerzo que se necesita para desplazarlas. Al pasar los años se han agregado a las sillas de ruedas sistemas eléctricos y electrónicos [5]–[7], como se ve

en la figura 2.1, con los cuales se ha podido dar un mayor control de las sillas de ruedas a las personas. Además, estas mejoras le han dado a la silla de ruedas mayor velocidad, estabilidad, y capacidad de esquivar y/o sobrepasar obstáculos.



Figura 2.1: Silla de ruedas motorizada de la marca Jiangnan Machinery Co., Ltd

2.1.2 Brazos robóticos

Mayormente los brazos robóticos han sido usados en la industria para facilitar algunos procesos donde se requiere precisión, fuerza, velocidad, y también donde se pueda salvaguardar una vida. Últimamente este sistema se está utilizando también para la asistencia de personas con discapacidad, facilitando la realización de actividades por parte de los usuarios. Algunos de estos son controlados mediante Joysticks, otros tienen sistemas más complejos con sensores que se adaptan a la realidad física de la persona para controlar al brazo robótico. Estos brazos también pueden ser adaptados a sillas de ruedas para complementar y aumentar la capacidad de los usuarios [8]. Por ejemplo, se puede apreciar en la figura 2.2 el brazo robótico **Mico** de la marca **Kinova** el cual está montado en una silla y puede ser controlado mediante un Joystick.



Figura 2.2: Brazo robótico Mico de la marca *Kinova* montado en una silla de ruedas

2.1.3 Robot humanoide de asistencia

Otro tipo de dispositivo de asistencia de personas son los robots humanoides que poseen forma humana y que sirven de asistencia en hospitales. Estos realizan labores donde se requiere una gran cantidad de fuerza como por ejemplo, para desplazar a los pacientes de una camilla a otra. Actualmente son robots muy caros por lo que su uso no es muy común. En Japón se han desarrollado algunos de estos robots de asistencia como el Robear Caregiver que se ve en la figura 2.3.



Figura 2.3: Robot humanoide de asistencia *Robear caregiver*

2.2 *Sistemas de Visión*

Los sistemas de visión son una herramienta de sensado y control de amplio uso dentro de la robótica. Existe un sin fin de aplicaciones que solucionan en forma robusta la detección de patrones para la automatización de sistemas. También su uso se ha extendido al diagnóstico y la asistencia de personas con alguna discapacidad física.

Los sistemas de visión por lo general nos dan información que puede ser utilizada para cambiar el medio. En el trabajo hecho por Blankenship y Alouani en la Universidad Tecnológica de Tennessee se estructuró un diagrama de bloques que podría ser usado para describir de manera general un sistema de visión [9] (Figura 2.4). En dicha figura se puede observar el proceso de reconocimiento del objetivo, el control e interacción de los componentes del sistema de visión. Se menciona también en un artículo del Instituto de Electrónica y Telecomunicaciones de Rennes dado por Kpalma y Ronsin sobre un punto de vista general del procesamiento de imágenes que generalmente se sigue orden [10]. Se aprecia que se hace un primer procesamiento (pre-procesamiento), luego de capturar la imagen, después se extraen los descriptores para así clasificar las características deseadas con ayuda de una base de datos. Finalmente, el sistema decide y ejecuta una acción que pueda cambiar la situación actual del medio.

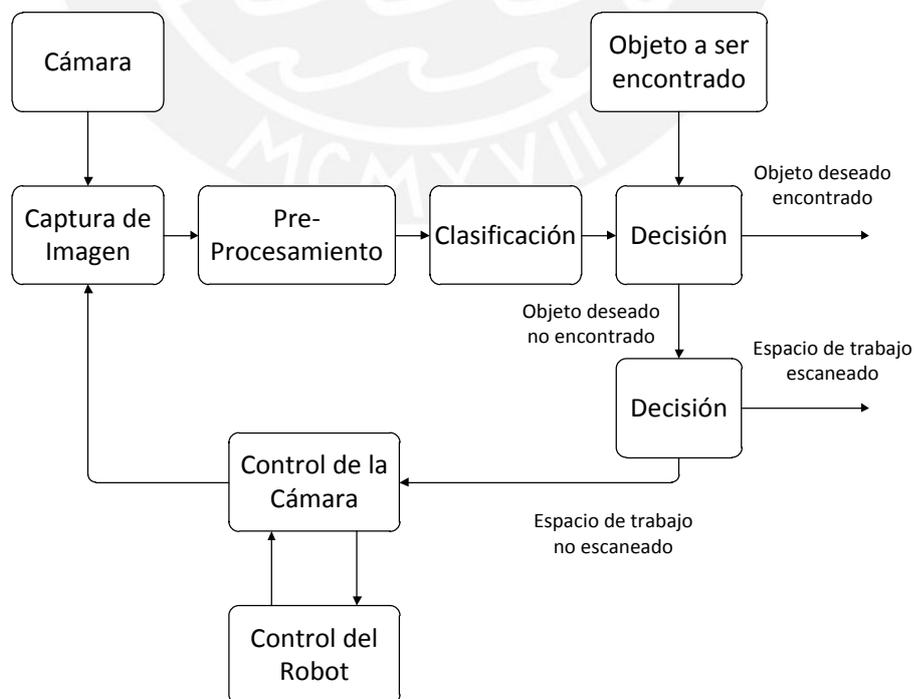


Figura 2.4: Diagrama de bloques de un sistema de visión [9]

Una función esencial de los sistemas de visión es reconocer objetivos, los cuales permitirán encaminar todo el proceso a su fin. En el artículo presentado por Li, Yin y Huang de la Universidad Naval de Ingeniería de China se explican los pasos que un sistema de visión debe realizar para detectar un objetivo [11]. El artículo muestra el diagrama de flujo del proceso de reconocimiento de un objetivo (Figura 2.5). Es fundamental, en este proceso, el ajuste de los parámetros de forma experimental para que el sistema pueda reconocer correctamente el o los objetivos.

Otros trabajos como los del Instituto de Electrónica y Telecomunicaciones de Rennes y de la Universidad de Miami también mencionan la importancia de los sistemas de visión en la automatización de procesos sobre todo por la precisión y velocidad de respuesta en sus resultados [10], [12]. Estos usan algoritmos complejos los cuales no son una tarea fácil de desarrollar. Día a día estos requieren un hardware con mayores prestaciones que le permitan ejecutarse en tiempo real. Las aplicaciones de los sistemas de visión son muchas y variadas. Se puede mencionar entre estos los sistemas de navegación de robots móviles, basados en visión, que pueden ser usados en ambientes peligrosos donde no puede llegar el ser humano. Se tiene un ejemplo en un trabajo presentado por la Corporación Toshiba donde se presenta robots que trabajan en plantas nucleares [13].

En la Universidad de Pavía (Italia) se realizó un trabajo en donde se muestra la importancia de la retroalimentación visual en los sistemas de navegación de robots para un control preciso y robusto sobre robots móviles [14]. Otras aplicaciones de los sistemas de visión están orientadas a la detección de objetos, estabilización de imágenes, calibración de ojo-mano, seguimiento de objetos en tiempo real, vehículos autónomos, estimación, predicción para compensación de errores, etc. Todas estas aplicaciones han ayudado a mejorar los procesos y la calidad de vida de muchas personas de ahí su importancia en la sociedad moderna.

De acuerdo con Lasher y Narayanan de la Universidad de Miami los sistemas de visión son indispensables para la automatización de los sistemas robóticos, siendo estos complementos entre ellos [12]. Siendo esta unión favorable para aplicaciones relacionadas a medidas, inspecciones y orientación. En la actualidad las plataformas de desarrollo permiten un mayor flujo de información con una mayor velocidad

facilitando así la implementación de algoritmos de reconocimientos de patrones muy complejos que optimizan los procesos de manera fiable.

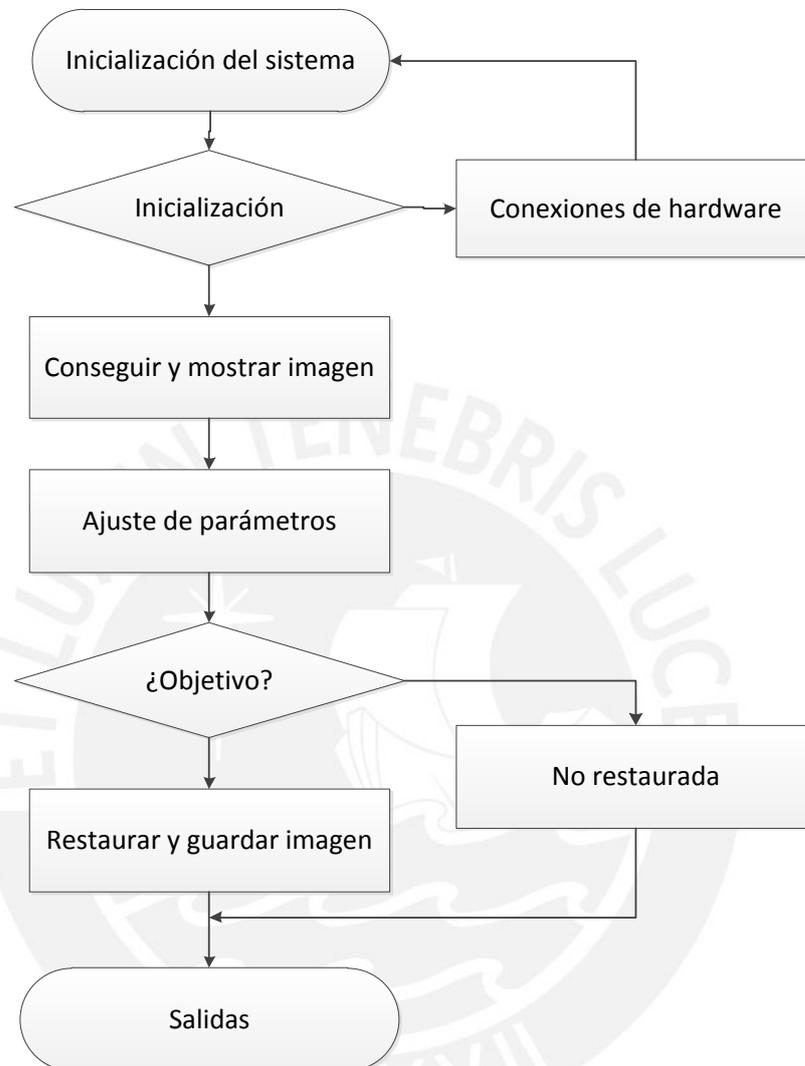


Figura 2.5: Diagrama de flujo para el reconocimiento de un objetivo [11]

2.3 *Detección y reconocimiento gestos*

2.3.1 Representación de gestos

El análisis anatómico de las expresiones faciales y movimientos articulares de la cabeza permite la descripción de patrones separada de la subjetividad, por lo que muchas áreas de estudio lo utilizan en investigaciones para evitar sesgos subjetivos.

Actualmente se ha extendido la aplicación de estas herramientas en diferentes áreas, tales como la psicología que originalmente encontró útil estas descripciones para reconocer y diferenciar patrones de comportamiento en las personas.

Los métodos de comunicación no verbal, ciencias forenses, sistemas de reconocimiento de patrones, y muchas otras aplicaciones han encontrado también una respuesta para validar sus investigaciones.

En el trabajo de Ekman & Friesen de 1977 se presenta un estudio extenso de un sistema de codificación de los actos faciales [15]. Este sistema, conocido con su nombre en inglés “Facial Action Coding System” (FACS), ha clasificado toda posible expresión facial anatómica. Las expresiones faciales se representan mediante acciones unitarias (AUs) las cuales dividen la cara en parte superior e inferior. En la figura 2.6 se muestran algunos códigos básicos y sus uniones para conseguir ciertos gestos. Las AUs tiene también un nivel de intensidad que puede ser especificado y que sirven para distinguir gestos.

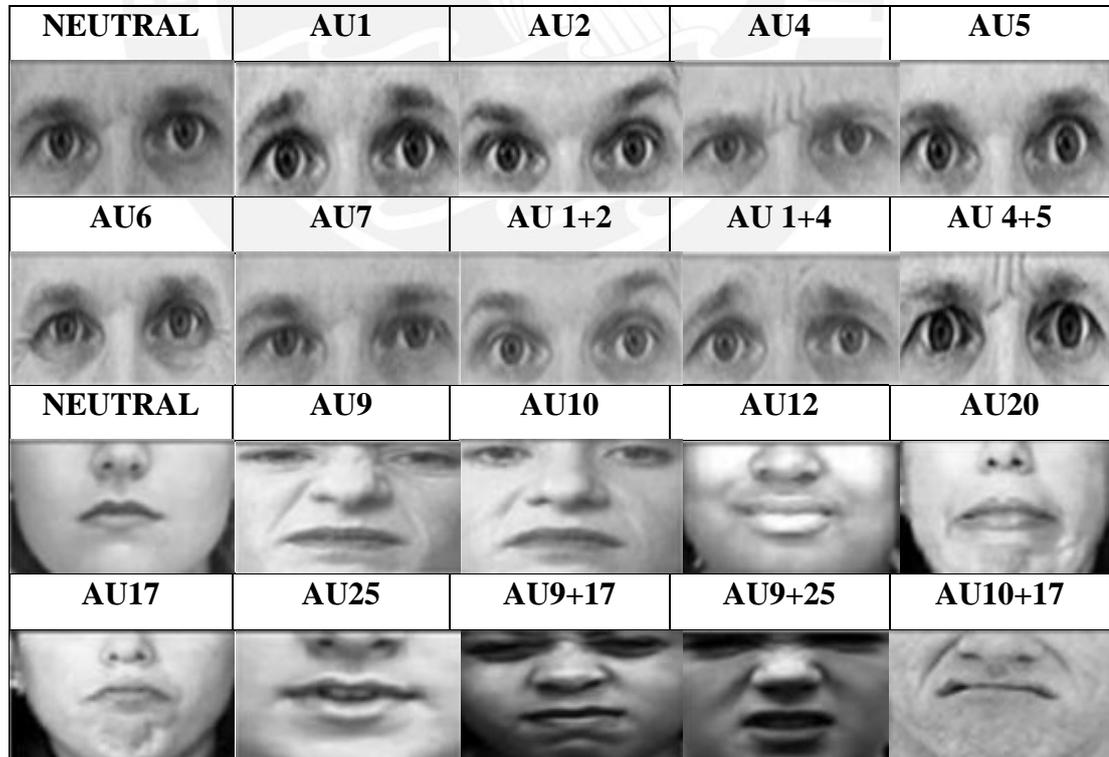


Figura 2.6: Acciones unitarias y algunas uniones básicas de los FACS [16]

Usando las letras A hasta la E se puede representar el grado de intensidad, en donde:

- A Rastro
- B Ligero
- C Marcado o Pronunciado
- D Severo o Extrema
- E Máxima

También se pueden representar emociones específicas como por ejemplo, el disgusto es representado por la unión las AUs 9 + 17 y el asco está conformada por AUs 9 + 25. Por ser una herramienta objetiva el estudio de Ekman & Friesen es útil para estudios en donde se necesita obtener información cuantificable de las expresiones faciales.

En el trabajo de Youssef, Shimodaira & Braude del 2013 en la Universidad de Edimburgo se muestra también un estudio sobre los movimientos articulares de la cabeza en el cual se comprueba que dichos movimientos están relacionados al comportamiento y actividades que realizan las personas [17]. Este estudio realizó pruebas para conocer qué estímulos provocaban los movimientos articulares. Los resultados fueron variando entre los participantes debido a la propia personalidad de los sujetos. Se usaron para la medición de los movimientos de la cabeza un dispositivo que medía electromagnéticamente los movimientos de la cabeza (EMA).

2.3.2 Algoritmos

Los sistemas de visión se han convertido en una herramienta importante para conseguir un control preciso y robusto de sistemas automáticos por lo que el desarrollo de algoritmos ha ido en aumento. Actualmente se han desarrollado una gran variedad de algoritmos para detectar y reconocer expresiones faciales y movimientos de la cabeza [16], [18], [19], [20], [21], [22], [23], [24], [25].

Existen diversos trabajos relacionados al reconocimiento de gestos, entre los más resaltantes se puede mencionar el trabajo presentado por Tian, Canade y Cohn en el “Transactions On Pattern Analysis And Machine Intelligence” del 2001 en donde se presenta un algoritmo implementado en un sistema de tiempo real con un promedio

93.3% de precisión en el reconocimiento de gestos [16]. Este algoritmo usa filtros de Gabor y el detector de bordes de Canny para obtener las características del rostro las que luego son clasificadas con “Artificial Neural Network” (ANN). Este trabajo divide el rostro en dos zonas, parte superior (ojos) e inferior (boca) como se ve en la figura 2.7.



Figura 2.7: Resultados del tracking usando el detector de bordes Canny [16]

En un artículo de Tao y Huang hecho en la Universidad de Illinois se analizó el rostro de manera volumétrica y se usó un algoritmo de deformación de volúmenes conocido por su nombre en inglés “Piecewise Bezier Volume Deformation” (PBVD), en donde el rostro se divide en 16 zonas y en las cuales el grado de deformación en cada una ayuda a distinguir los gestos [25]. En la figura 2.8 se observa cómo se divide esta máscara en el rostro y cómo se deforma cuando se ejecutan algunos gestos. Este algoritmo trabaja con puntos de control que son características del rostro humano.

En un trabajo presentado en el “Transaction On Systems, Man, And Cybernetics”, en el 2006, por Anderson y McOwan consideraron usar el algoritmo “Spatial Ratio Template” con el clasificador “Support Vector Machine” (SVM) obteniendo una precisión de reconocimiento del rostro en movimiento de 80.52% [19]. Este algoritmo se muestra como una plantilla que se aplica en la región de interés en la cara y la divide en zonas para describirla.

En algunas aplicaciones es necesario procesar la data en tiempo real por eso se han desarrollado técnicas computacionales que reducen el tamaño de la información con lo que se logra una mayor velocidad para la ejecución del programa. Por ejemplo, en un trabajo de Aleksic y Katsaggelos presentado en el “Transaction On Information Forensics and Security”, en el 2006, se utiliza el análisis de la componentes principales de la imagen para disminuir la data y “Multistream Hidden Markov Model System” (MS-HMMs) para clasificarla logrando procesar en tiempo real el algoritmo. Este trabajo presenta que la máxima precisión obtenida fue de 93.66% al reconocer el contorno de los labios [26].

En el trabajo de Bettadapura en el Instituto de Tecnología de Georgia se muestra una recopilación de varios algoritmos y métodos para lograr el objetivo de identificar los gestos para la implementación de sistemas de visión [21]. En este último también existen una serie de comparaciones con respecto al procesamiento, la base de datos y los resultados.

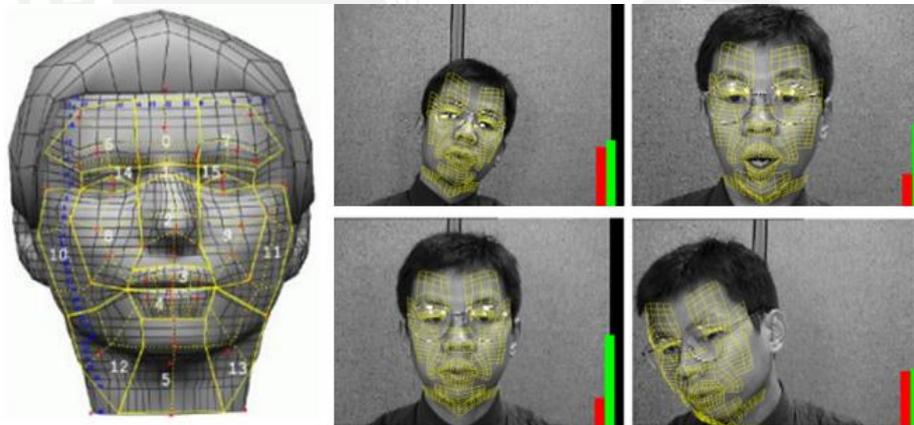


Figura 2.8: Piezas de volúmenes de Biezer en el rostro durante el seguimiento del rostro (Face Tracking) [25]

2.4 *Interacción humano-robot*

La interacción humano-robot es una ciencia que estudia la relación de los robots con las personas y como sus capacidades de interacción afectan al individuo. Tenemos que tener en cuenta no solo el objetivo y la trayectoria del robot sino también tener en cuenta la forma como lo alcanza.

En la actualidad, se busca programar al robot de forma que sus movimientos se asemejen al de un ser humano [27], haciéndolo atractivo para la interacción. El robot como dispositivo programable tiene la capacidad de desarrollar funciones preestablecidas. Algunos tienen algoritmos muy complejos que le permiten aprender de su medio con capacidades sensoriales (tacto, visión, sonido, etc.) y a la vez decidir cómo realizarla. La forma de interacción es un factor muy importante para entablar una relación eficaz humano-máquina por lo cual esta área, junto a la educación, está en constante crecimiento. Los países del continente asiático, tales como Japón, Corea, Taiwán, China, etc., como referentes de la robótica y la interacción social, vislumbran toda esta ciencia que está impulsando muchos sectores, por ejemplo, en la educación, psicología, sociología, etc.

La robótica social también está permitiendo descubrir nuevas maneras de transmitir información eficazmente [28] a través de la interacción robótica en niños desde edades tempranas [29]. Otras áreas como la psicología está estudiando el impacto de los robots en las personas debido a que en un futuro no muy lejano la interacción humano-robot será más frecuente [30].

Las ciencias políticas tampoco se quedan atrás debido a que con la masificación de los robots, en el futuro, cada uno podrá tener un asistente robótico con características cada vez más humanas. Esto impulsará a los gobiernos a crear reglas y políticas en el ámbito social que ayuden a llevar una sociedad más organizada [31]. Estas normas y leyes son ya una realidad en Japón y Corea del Sur en donde ya tienen borradores sobre leyes aplicadas a robots.

CAPÍTULO 3

Desarrollo del sistema de visión para la asistencia de personas con discapacidad

El presente capítulo describe el sistema de visión desarrollado y los requerimientos de diseño que enmarcan el mismo. Además se describirá y mostrará el proceso de implementación de cada parte del mencionado sistema.

El diseño de un dispositivo de asistencia para personas con discapacidad debe cumplir ciertos requerimientos que satisfagan sus necesidades. Las personas con alteraciones en la columna producidas por algún accidente, por lo general, pierden la movilidad de uno o más miembros, por tanto para este estudio se buscará asistir a personas que tengan dominio de sus expresiones faciales y los movimientos articulares de la cabeza.

3.1 *Diseño conceptual del funcionamiento del sistema*

El sistema de visión que se plantea desarrollar pretende servir como ayuda a personas con discapacidad o que hayan sufrido la pérdida de sus miembros superiores, inferiores o ambos. Para el apoyo de estas personas el sistema de visión a desarrollar a continuación contempla el control de un brazo robótico mediante reconocimiento de patrones. Como el sistema tiene en cuenta las personas que no pueden valerse por sí mismas mediante sus extremidades, este sistema reconoce expresiones faciales y movimientos articulares de la cabeza como patrones de comando.

Para probar la utilidad de este sistema se considerarán gestos y movimientos de la cabeza intuitivos para las personas [17], [32] los cuales están en relación al control del brazo robótico de forma incremental (Primer modo de operación). Este movimiento será desarrollado en un plano cartesiano, salvando una zona de seguridad entre el usuario y el robot, además el robot podrá tener una función pre-establecida (Segundo modo de operación) para que realice un movimiento complejo compuesto de varios movimientos articulados. Para probar este movimiento pre-establecido el brazo se programó para coger una botella de agua que estará en una mesa, en una posición fija, y luego se la alcanzará al usuario. Para lograr el objetivo se implementó un sistema de visión, que consta de una cámara que enfoca al rostro del usuario. Luego, mediante un algoritmo se hizo un seguimiento del rostro para extraer imágenes, las cuales son procesadas y clasificadas para decidir la acción a ejecutar con el robot. Con este sistema se conseguirá dar apoyo a las personas que tengan una discapacidad y no puedan valerse por sí mismas.

3.2 Base de datos

3.2.1 Consideraciones técnicas

Teniendo en cuenta a los usuarios, que son pacientes con discapacidad que no pueden realizar movimientos voluntarios ni involuntarios de sus extremidades superiores e inferiores, se ha establecido que sean las expresiones faciales y los movimientos articulares de la cabeza que determinen el manejo de un dispositivo robótico para su propia asistencia. Con estas condiciones se plantea diseñar una base de datos de gestos que serán utilizados en un algoritmo que realizará el seguimiento del rostro y el reconocimiento de ciertas expresiones faciales y movimientos de la cabeza.

La base de datos es un factor importante dentro del reconocimiento de gestos, ya que permite homogenizar los resultados y la comparación con otros métodos algorítmicos. Así también la comparación de los resultados con respecto a otros investigadores puede hacerse más fácil [21].

Actualmente existe una gran variedad de bases de datos que son muy útiles. Según la aplicación una u otra base de datos será la ideal, pero otras veces será mejor implementar una base de datos local que se ajuste a las herramientas de desarrollo. Para ello se necesita tener en cuenta diferentes factores, en este caso, posición e iluminación.

Con respecto a la posición se definen casos en donde la imagen del rostro esta escalada y rotada con respecto a los ejes vertical, anteroposterior y transversal (Ver figura 3.1). Toda esta información será útil si queremos entrenar correctamente al algoritmo que usará un aprendizaje supervisado.

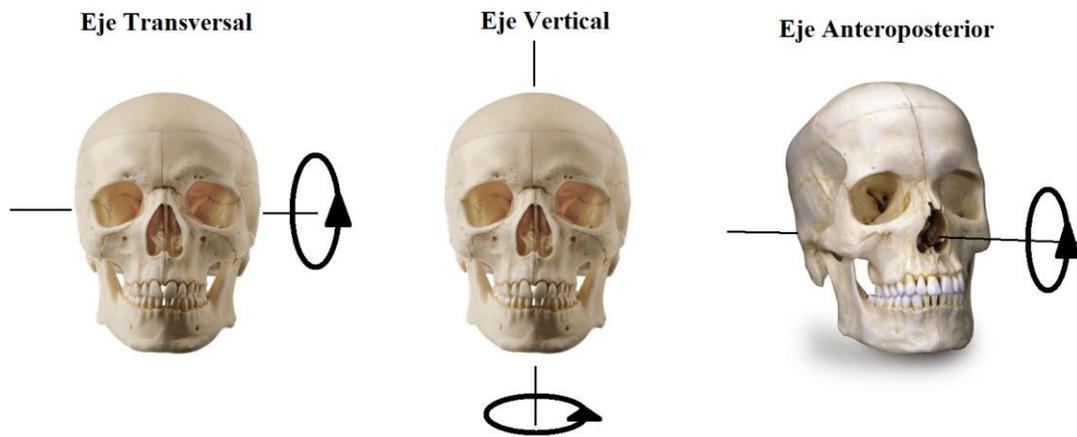


Figura 3.1: Ejes de rotación del cuerpo humano

La Iluminación es otro factor que interfiere con la tarea del reconocimiento de patrones. Por lo que se establecerán condiciones controladas para probar correctamente la utilidad del sistema.

Para la adquisición se deberá seleccionar una cámara que facilite y optimice el reconocimiento de patrones. La cámara debe tener en cuenta las siguientes características a fin de obtener un mejor resultado:

- Tamaño de la imagen entregada.
- Cantidad de cuadros por segundo.
- Protocolo de comunicación.

3.2.2 Herramientas y consideraciones metodológicas

A continuación se definirán los instrumentos, su ubicación, la iluminación, las expresiones del rostro y los movimientos articulares de la cabeza que se usarán para la creación de la base de datos.

Herramientas

- 1 Cámara VGA
- 1 Luxómetro
- 1 Pedestal de cámara
- 1 Proyector multimedia
- 1 Ecran
- 2 Linternas led
- 1 Laptop

Ubicación de la cámara

Según la proxémica, que ha medido las distancias de interacción humana. La distancia adecuada para la interacción amical entre individuos se da entre el espacio personal y espacio social, esta distancia es 120cm como máximo. Se escogió esta distancia para conseguir una interacción de confianza y generar una sensación de seguridad en el usuario.

Iluminación

Se trabajará en una habitación cerrada con iluminación general para trabajo en interiores según la Organización Internacional del Trabajo (OIT) [33] para tareas prolongadas que requieren precisión, esto es 3000 LUX, lo que será corroborado con un luxómetro. Las linternas led serán colocadas una al lado izquierdo y otra al lado derecho para tener una iluminación uniforme sobre el rostro. Las linternas estarán en pedestales a la altura del centro del rostro.

Gestos y movimientos

Los movimientos y gestos a utilizarse serán: movimientos rotacionales izquierda y derecha, inclinación hacia adelante y atrás, neutro (sin expresión) y boca abierta, los cuales son movimientos y/o gestos intuitivos para el control del dispositivo de

asistencia. Para la realización correcta de los movimientos y gestos a los participantes se les presentó diapositivas de dichas expresiones sobre un écran.

Cámara

Para la adquisición de imágenes se usara una cámara VGA que maximice el costo computacional. La comunicación de la cámara será USB debido a que es un protocolo estándar que muchas plataformas de desarrollo tienen.

Otra característica importante que permitirá obtener una mayor velocidad será la cantidad de cuadros por segundo que la cámara podrá entregar por lo que se usará una cámara de 30 cuadros por segundo.

3.2.3 Implementación

La adquisición se hizo en un espacio de trabajo cerrado que cumplieran las condiciones y consideraciones previamente mencionadas. En la figura 3.2 se ve el espacio de trabajo en donde los colaboradores realizaban los gestos y movimientos articulares de la cabeza. La prueba duraba 10 minutos por persona y esta consistía en imitar los movimientos y gestos mostrados a través de una proyección hecha por un cañón multimedia en un écran. Esto facilitaba a los participantes la realización de los gestos y movimientos. El nivel de iluminación era corroborado a través del luxómetro. En la figura 3.3 se algunos resultados de la base de datos.

Para realizar una buena adquisición, se implementó un algoritmo con el cual se pudo capturar solo el rostro de los participantes. Esto para tener la base de datos homogénea. El algoritmo detallado en el **Anexo** devuelve las imágenes en escala de grises y hace una detección automática del rostro. El mismo algoritmo de captura se empleará para la implementación del sistema de visión y la descripción del mismo se verá en el capítulo 3.3.

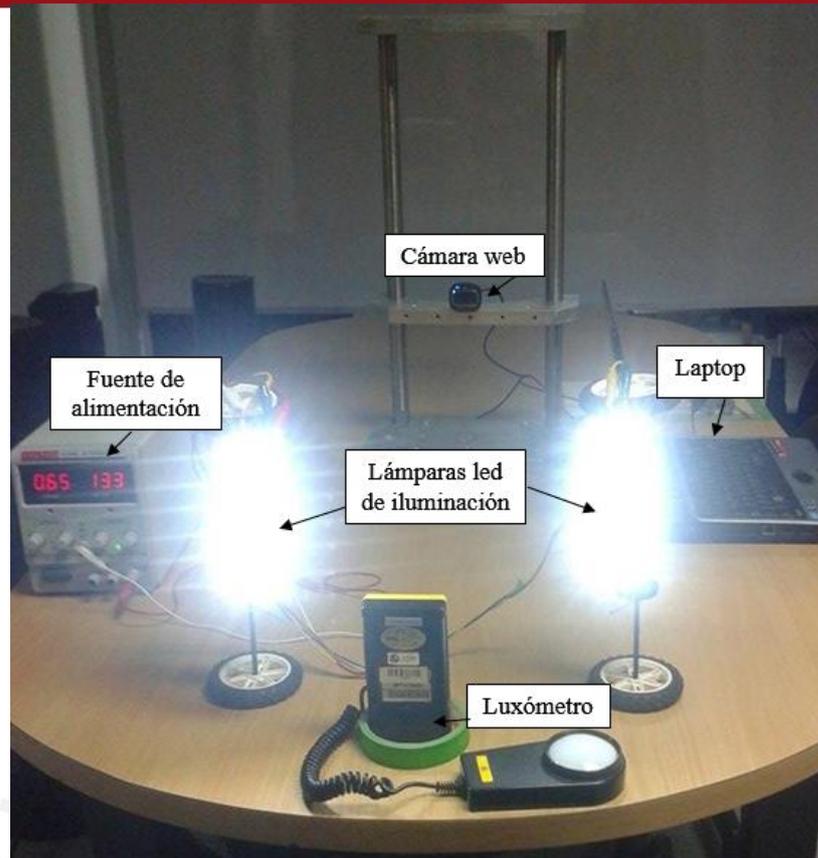


Figura 3.2: Espacio de trabajo y herramientas para la recopilación de la base de datos

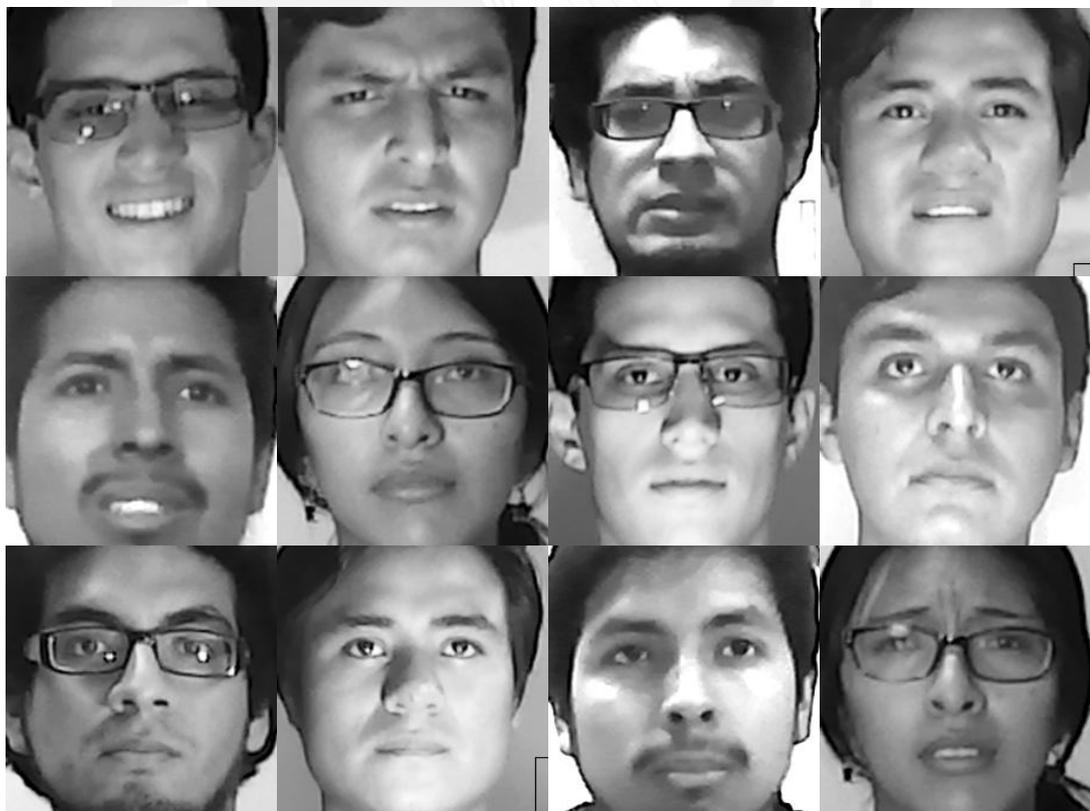


Figura 3.3: Algunas imágenes de la base de datos en posición neutra de la cabeza

3.3 *Algoritmo del sistema de visión*

3.3.1 Selección del algoritmo

Existen diversos algoritmos de reconocimientos de rostros pero para realizar una correcta selección debemos definir las características con las cuales nuestro sistema funcione adecuadamente. Según el capítulo anterior, el artículo de Bettadapura hace el estado del arte de varios algoritmos de procesamiento de imágenes aplicados al reconocimiento de rostros y mostrando los resultados en un cuadro comparativo [21]. Estos resultados son comparados en base a la velocidad de procesamiento y la precisión para detectar correctamente un rostro, diferenciándolo de falsos positivos y negativos. Estas características son deseables para el correcto funcionamiento de este sistema. En dicho artículo de Bettadapura resaltan, según los resultados, el trabajo de Tian, Kanade y Cohn del 2001 [16] y el de Aleksic & Katsaggelos del 2005 [34]. El primero tiene un 93% de precisión en el reconocimiento de gestos. Este algoritmo usa primero filtros de Gabor [35] y el detector de bordes de Canny [36] para obtener las características del rostro las que luego son clasificadas con Redes Neuronales Artificiales (RNA). El otro trabajo utiliza Análisis de las Componentes Principales (ACP) para caracterizar los ojos y los labios logrando un 93% de precisión [34]. Un trabajo de Belhumeur, Hespanha & Kriegman de 1997 presentado en el “Transaction On Pattern Analysis And Machine Intelligence” en el que se compara dos técnicas para determinar cuál de ellas tiene un menor error de detección de rostros. Los resultados indicaron que “EigenFaces”(ACP) posee mayor error en la detección de características que el algoritmo “FisherFaces”[37]. Este último tiene mayor robustez con respecto a la variación de iluminación y expresiones que ACP, además que posee menor tasa de error como se observa en la figura 3.4. Por tales motivos se usó para este trabajo el algoritmo de Viola & Jones que posee una gran velocidad de procesamiento gracias a su algoritmo “Integral image”, además que usa para el reconocimiento de gestos el algoritmo de “FisherFaces” permitiendo la ejecución del programa en tiempo real. En la figura 3.5 podemos observar el diagrama de bloques del mencionado algoritmo a implementar.

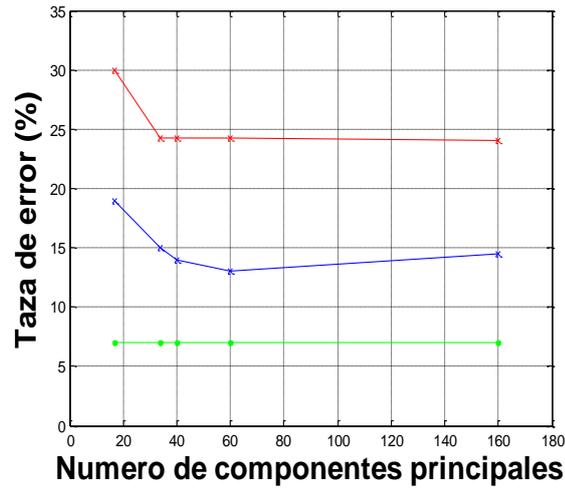


Figura 3.4: Taza de error de Eigenfaces (Rojo), Eigenfaces sin sus tres primeras componentes principales (Azul) y Fisherfaces (Verde) [37]

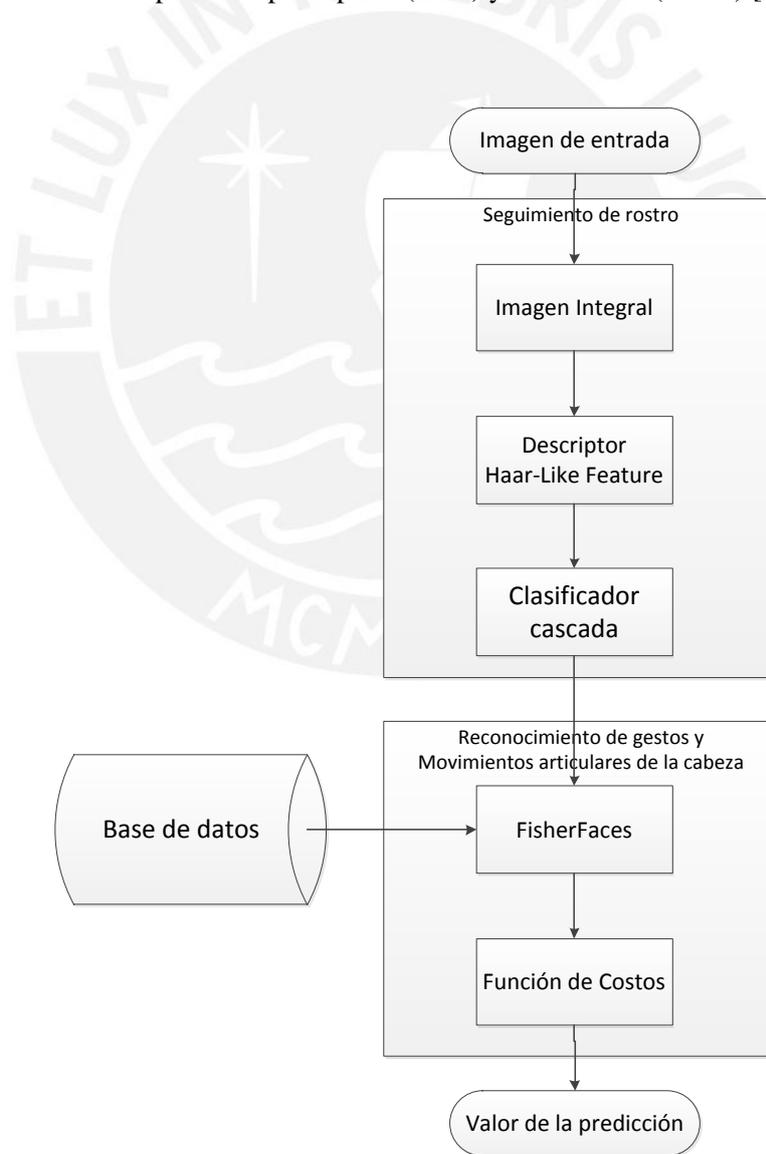


Figura 3.5: Diagrama de bloques del algoritmo de reconocimiento de patrones

3.3.2 Descripción del algoritmo

3.3.2.1 Haar-like features

Para representar una imagen o características de ella “Haar-like features” aplica diferentes parches en la imagen. Estos parches permiten detectar características de borde, línea y centro envolvente, todas ellas en la misma imagen. Luego corresponde almacenarlas con una etiqueta para usarlas en la etapa de clasificación [38].

La figura 3.6 se muestra los tipos de parches que se aplican a la imagen, estos tienen una zona blanca y otra negra. La zona blanca indica que se debe hacer una suma positiva de todos los pixeles que ahí existen. La zona negra hace lo mismo pero la suma es negativa. Los valores resultantes de la zona donde el parche fue aplicado muestra la existencia de espacios simétricos, la cual facilita la detección del rostro que tiene características simétricas (Ver figura 3.7).

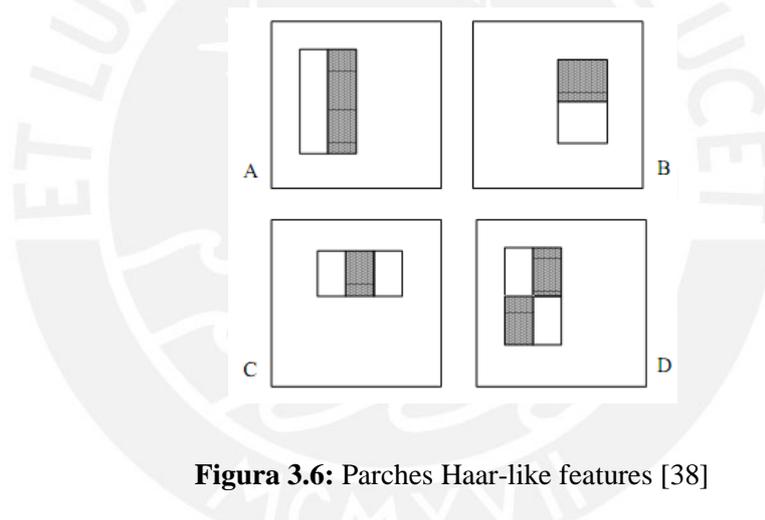


Figura 3.6: Parches Haar-like features [38]

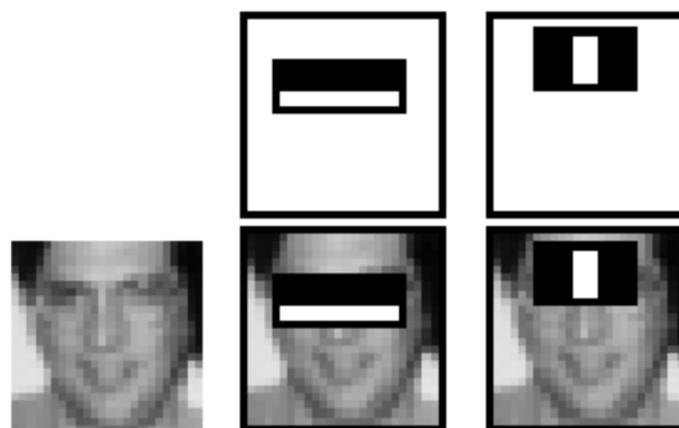


Figura 3.7: Parche aplicado en la imagen del rostro de una persona [38]

3.3.2.2 Imagen integral (“Integral image”)

Para reducir el coste computacional del algoritmo y lograr mayor velocidad de procesamiento el algoritmo propuesto por Viola & Jones propone una conversión de la imagen original [38]. Esta conversión se da en función a la facilitación de la obtención de las características a extraer, la cual se explicará más adelante a detalle. Imagen integral devuelve una imagen con valores que representan la suma acumulada, mientras recorremos la imagen de izquierda a derecha y de arriba hacia abajo [38] como se observa en la figura 3.8. El valor del pixel en la ubicación de la nueva imagen integral se rige mediante la ecuación 3.1 en donde $ii(x,y)$ es la imagen integral y $i(x',y')$ es la imagen original.

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

Para tal objetivo se usa las siguientes ecuaciones recurrentes (3.2) y (3.3).

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

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

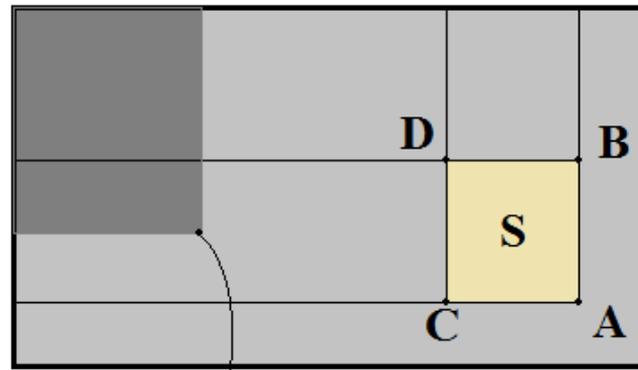
En la figura 3.8 se observa un ejemplo de la conversión de una imagen.

Original					Integral				
5	2	3	4	1	5	7	10	14	15
3	2	4	2	3	8	12	19	25	29
2	2	1	3	4	10	16	24	33	41
3	5	6	4	5	13	24	38	51	64
4	1	3	2	6	17	29	46	61	80

Figura 3.8: Ejemplo de la obtención de la integral de una imagen

Fuente propia

Con la imagen integral la metodología de la aplicación de los parches se vuelve mucho más rápida debido a que ya no se hace la suma de todos los píxeles sometidos al parche sino que todas esas sumas se remplazan por 3 operaciones, la cual es mostrada en la figura 3.9.



$$ii(x, y) = \sum_{x' < x, y' < y} i(x', y')$$

Figura 3.9: Calculo de la suma de píxeles en un área en una imagen integral

Fuente propia

Para obtener la superficie deseada, en este caso **S**, realizamos la operación de la ecuación 3.4 con las cuatro arreglos **D**, **C**, **B**, **A**.

$$S = A - B - C + D \quad (3.4)$$

De esta manera podremos calcular rápidamente los valores de las zonas blancas y negras en los diferentes tipos de parches aplicados.

3.3.2.3 Clasificador cascada

Para la detección de características, el clasificador cascada hace una comparación entre la imagen de entrada y todos los descriptores pertenecientes a una imagen de la base de datos.

La cantidad de descriptores por cada imagen es muy alta (alrededor de 180,000) por lo que para disminuir el tiempo computacional se procede a evaluarlos de manera que

se seleccionen aquellos que describen mejor a la imagen. Para dicha selección se usa la ecuación 3.5 en donde h_j es el descriptor clasificado, f_j es el descriptor a clasificar, θ_j es el umbral seleccionado para la clasificación del descriptor y p_j es el valor paridad que indica la dirección de la desigualdad [38]. Para este caso se ha colocado un umbral que detecta en un 40% los falsos positivos.

$$h_j(x) = \begin{cases} 1 & \text{si } p_j f_j(x) < p_j \theta_j \\ 0 & \text{cualquier otro caso} \end{cases} \quad (3.5)$$

En esta etapa se determina si el descriptor encontrado pertenece a un rostro como se ve en la figura 3.10.

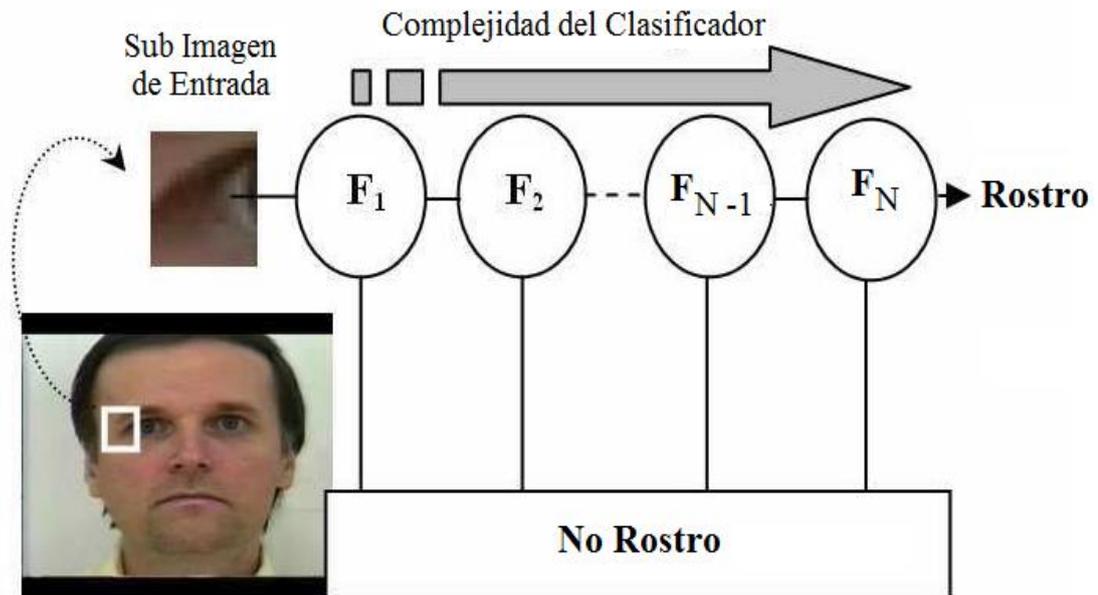


Figura 3.10: Esquemático de la detección cascada [38]

3.3.2.4 FisherFaces

Fisherfaces es un algoritmo que usa una proyección para determinar la semejanza entre una o más clases. Dicha proyección viene dada mediante una matriz de pesos representativa que se hace mediante un conjunto de datos de entrenamiento. La base de datos que se usa para obtener dicha matriz debe representar todos los casos que se quieran reconocer. Es decir son la cantidad de clases que se quieren tener [37].

En la sección 1.9.3 se mencionó las características de la base de datos implementada, las cuales se trabajaron en un ambiente controlado para disminuir el ruido del sistema. Estos detalles se especifican en esa misma sección.

El algoritmo de Fisherfaces trabaja primero con la base de datos, la cual es puesta en una matriz en donde cada columna es una imagen y a la vez cada una de ellas es una clase. Luego se obtiene la media de esa matriz la cual servirá para determinar la aproximación entre cada clase y la media. Esto también permitirá obtener una matriz que represente la dispersión que existe entre las clases S_B . Donde u es la imagen media de todas las imágenes para el entrenamiento, u_i es la imagen media representativa de cada clase, N_i es el número de muestras en clase y C la cantidad de clases [37]. Esto se representa en la ecuación 3.6.

$$S_B = \sum_{i=1}^C N_i (u_i - u)(u_i - u)^T \quad (3.6)$$

Ahora es necesario obtener otra matriz de dispersión pero esta vez que represente la dispersión propia de cada clase S_W , para lo cual se hallará la media de cada clase y se restará de cada valor de la muestra. Donde x_k es cada una de las imágenes en cada clase [37]. Esto lo que apreciamos en la ecuación 3.7.

$$S_W = \sum_{i=1}^C \sum_{x_k \in X_i} (x_k - u_i)(x_k - u_i)^T \quad (3.7)$$

Como siguiente paso a las matrices de dispersión se aplicará un análisis de discriminación lineal estándar del cual se maximizará el ratio de dispersión. Extrayendo los autovectores de la matriz de dispersión entre las clases y los autovalores de la matriz de dispersión de cada clase podremos calcular la matriz de pesos W_{opt} con la que podremos clasificar los rostros [37]. Dicha operación se puede observar en la ecuación 3.8.

$$W_{opt} = \arg \max_W \frac{|W^T S_B W|}{|W^T S_W W|} = [w_1 \ w_2 \ \dots \ w_m] \quad (3.8)$$

3.4 Mecanismo de asistencia

3.4.1 Unidad robótica

Para lograr una mejor participación de las personas con discapacidad se han desarrollado dispositivos que los ayudan a conseguir autonomía en sus actividades diarias. Se plantea escoger mecanismos que sirvan a las personas con discapacidad. Este sistema de asistencia escogido es el brazo robótico Jaco de la marca *Kinova* (Ver figura 3.11), el cual interactúa de forma segura y fácil con el medio. Este cuenta con 6 grados de libertad que permiten al usuario realizar diferentes movimientos y conseguir objetos dentro de su zona de alcance (90 cm), además posee 3 dedos que se controlan individualmente para agarrar y sostener objetos. Posee también un control intuitivo para el usuario con sus tres modos: trasladar, rotar y agarrar. Estas características pueden ayudar a desarrollar un sistema especializado para personas con discapacidad.

3.4.2 Programación del sistema robótico

El brazo robótico Jaco de la marca *kinova* posee una interface de programación avanzada (API) que permite su programación para la integración con cualquier otro sistema. Además el brazo robótico tiene sus propias librerías de programación y comunicación para que controlarse mediante los sistemas operativos Windows y Ubuntu. Estas son: Kinova API Jaco y Kinova DLL. Estas librerías pueden usarse en lenguaje C++ de manera que el algoritmo de visión y el programa del brazo robótico sean uno solo código. La figura 3.12 muestra el diagrama de flujo de las tareas programadas en el brazo robótico. Por seguridad para los usuarios se estableció una zona de seguridad, mediante software, para limitar el rango de movimiento de las articulaciones del brazo robótico.



Figura 3.11: Brazo Jaco de Kinova y sus partes [39]

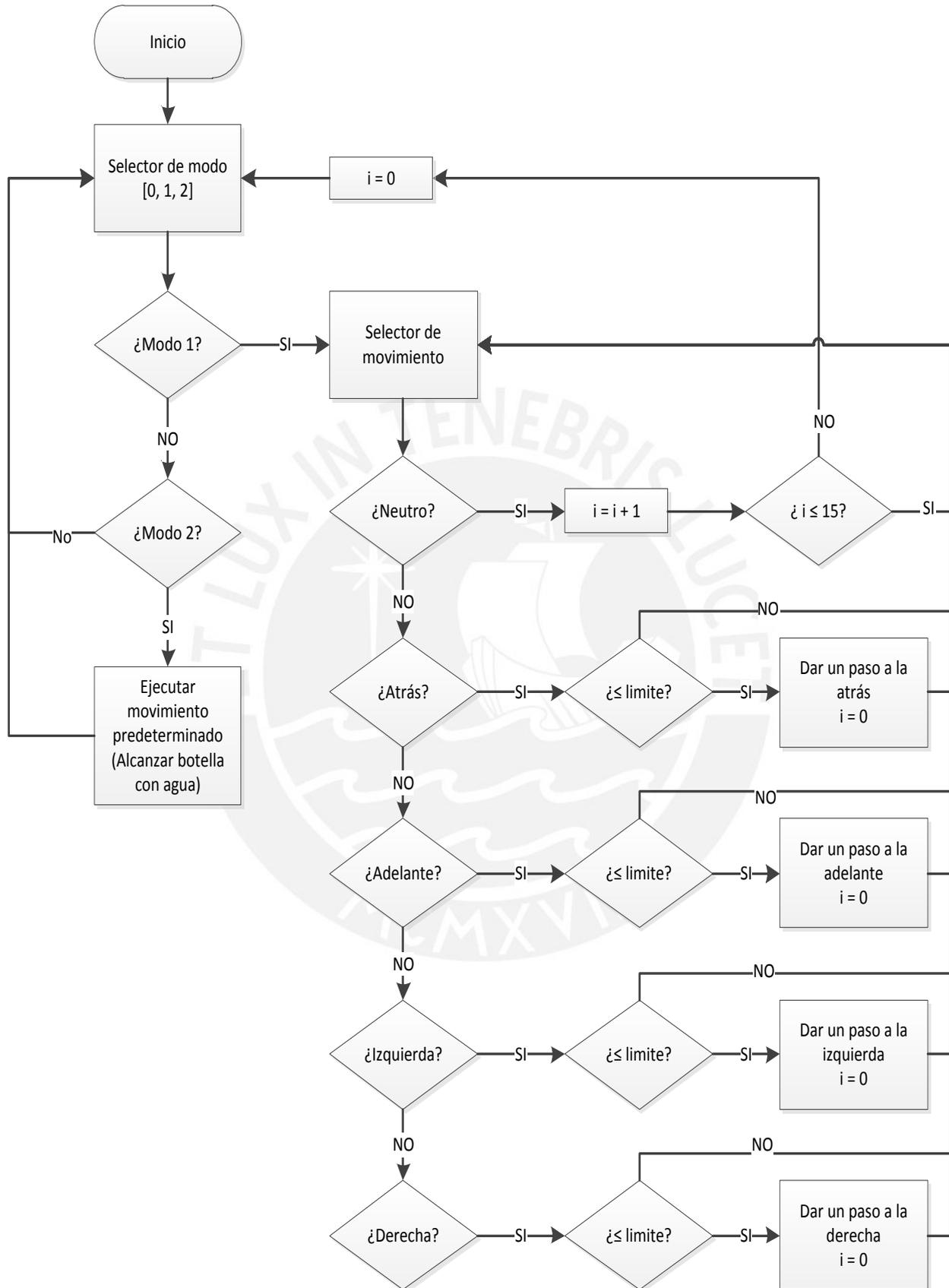


Figura 3.12: Diagrama de flujo del funcionamiento del brazo robótico

Las librerías que permiten realizar movimientos, ajustar parámetros en el brazo robótico y habilitar la comunicación serial son:

```
#include "KinovaTypes.h"
#include "Kinova.API.UsbCommandLayerUbuntu.h"
```

Preparación del Joystick virtual que emula la data que envía un Joystick real.

```
JoystickCommand virtualCommand;
```

Inicialización de los valores iniciales del brazo robótico.

```
virtualCommand.InclineForwardBackward = 0;
virtualCommand.InclineLeftRight = 0;
virtualCommand.MoveForwardBackward = 0;
virtualCommand.MoveLeftRight = 0;
virtualCommand.PushPull = 0;
virtualCommand.Rotate = 0;
```

Actualización de data para mover la posición del brazo a una posición específica.

```
(*MySendJoystickCommand) (virtualCommand);
```

Ejemplo de código para los movimientos cartesianos del efector final según el gesto detectado

```
if(prediction == 1){
virtualCommand.InclineForwardBackward = 0;
virtualCommand.InclineLeftRight = 0;
virtualCommand.PushPull = 0;
(*MySendJoystickCommand) (virtualCommand);
usleep(1000);}
if(prediction == 2){
virtualCommand.InclineForwardBackward = 1;
virtualCommand.PushPull = 0;
virtualCommand.InclineLeftRight = 0;
(*MySendJoystickCommand) (virtualCommand);
usleep(1000);}
if(prediction == 3){
virtualCommand.InclineForwardBackward = -1;
virtualCommand.PushPull = 0;
virtualCommand.InclineLeftRight = 0;
(*MySendJoystickCommand) (virtualCommand);
usleep(1000);}
if(prediction == 4){
virtualCommand.InclineForwardBackward = 0;
virtualCommand.PushPull = 0;
virtualCommand.InclineLeftRight = 1;
(*MySendJoystickCommand) (virtualCommand);
usleep(1000);}
if(prediction == 5){
virtualCommand.InclineForwardBackward = 0;
virtualCommand.PushPull = 0;
virtualCommand.InclineLeftRight = -1;
(*MySendJoystickCommand) (virtualCommand);
usleep(1000);}
```

Ejemplo de movimientos pre-definidos para recoger objetos

```
//Inicializamos el metodo API
cout << "Calling the method InitAPI()" << endl;
result = (*MyInitAPI)();
cout << "result of InitAPI() = " << result << endl << endl;

//Inicializamos el uso del brazo robotico
cout << "We take control of the robotic arm." << endl;
result = (*MyStartControlAPI)();

//Declaramos la variable tipo "TrajectoryPoint"
TrajectoryPoint trajectoryPoint;

//Inicialización variable "trajectoryPoint"
trajectoryPoint.InitStruct();

//Carga de valores en la variable
trajectoryPoint.Position.HandMode = HAND_NOMOVEMENT;
trajectoryPoint.Position.Type = CARTESIAN_POSITION;

trajectoryPoint.Position.CartesianPosition.X = 0.21;
trajectoryPoint.Position.CartesianPosition.Y = -0.39f;
trajectoryPoint.Position.CartesianPosition.Z = 0.5f;

trajectoryPoint.Position.CartesianPosition.ThetaX = 1.52f;
trajectoryPoint.Position.CartesianPosition.ThetaY = 0.48f;
trajectoryPoint.Position.CartesianPosition.ThetaZ = -3.17f;

//Envío de la variable con la trayectoria deseada
(*MySendBasicTrajectory)(trajectoryPoint);
```

Zona de seguridad (Ver figura 3.13)

```
//Declaramos la variable tipo "zoneList"
zoneList zones;

//Inicialización variable "zones"
zones.NbZones = 1;

//Carga de valores en la variable
zones.Zones[0].zoneShape.shapeType = PrismSquareBase_Z;
zones.Zones[0].zoneShape.Points[0].X = 0.1f;
zones.Zones[0].zoneShape.Points[0].Y = -0.52f;
zones.Zones[0].zoneShape.Points[0].Z = 0.23f;
zones.Zones[0].zoneShape.Points[0].ThetaX = 0.0f;
zones.Zones[0].zoneShape.Points[0].ThetaY = 0.0f;
zones.Zones[0].zoneShape.Points[0].ThetaZ = 0.0f;
zones.Zones[0].zoneShape.Points[1].X = 0.1f;
zones.Zones[0].zoneShape.Points[1].Y = -0.32f;
zones.Zones[0].zoneShape.Points[1].Z = 0.23f;
zones.Zones[0].zoneShape.Points[2].X = 0.3f;
zones.Zones[0].zoneShape.Points[2].Y = -0.32f;
zones.Zones[0].zoneShape.Points[2].Z = 0.23f;
zones.Zones[0].zoneShape.Points[3].X = 0.3f;
zones.Zones[0].zoneShape.Points[3].Y = -0.54f;
zones.Zones[0].zoneShape.Points[3].Z = 0.23f;
zones.Zones[0].zoneShape.Points[4].Z = 0.63f;

//Ajuste de la zona de seguridad
result = (*MySetProtectionZone)(zones);
cout << "La zona de protección ha sido configurada con la variable result = " << result << endl;
```

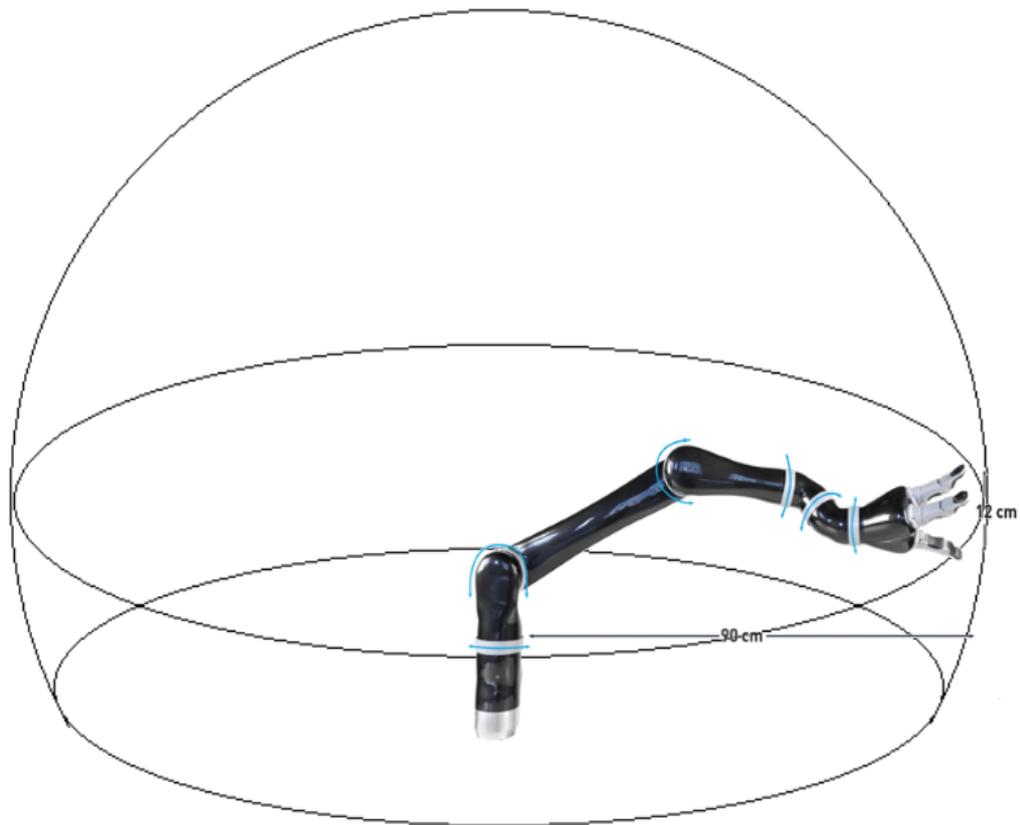


Figura 3.13: Brazo Jaco de Kinova [39] en su zona de seguridad

3.5 Sistema electrónico

3.5.1 Requerimientos de la plataforma de desarrollo

El algoritmo de Viola & Jones [38] menciona que para obtener un procesamiento a 15 cuadros por segundo en imágenes de 384 x 288 en tiempo real se necesita como mínimo un procesador de 700 MHz, además 500 GB de memoria RAM. Esos son los requerimientos mínimos para que la cámara funcione correctamente para poder capturar las imágenes con la calidad y velocidad deseadas. Para seleccionar el Hardware adecuado hay que tener en cuenta diferentes características como la portabilidad del sistema de visión para que acompañen al usuario discapacitado. También un sistema operativo liviano y libre son características importantes a tener en cuenta. Por eso se escogerá Linux debido a que cumple satisfactoriamente estos requerimientos, además por su robustez ante ataques de virus que puedan dañar el sistema.

Con respecto al algoritmo de visión según la figura 3.14 los lenguajes de programación que ocupan menor memoria y procesan más rápido son C y C++. Debido a estas ventajas y por la facilidad de librerías de desarrollo se programará el sistema en lenguaje C++.

La selección del procesador será parte fundamental para el desarrollo del sistema de visión. En el mercado existe una gran cantidad de plataformas de desarrollo pero se deben tener en cuenta los requerimientos antes mencionados sin exigir tampoco al máximo al procesador ni sobredimensionarlo. En la tabla 3.1 se muestra la comparación de las características de los distintos tipos de hardware que se pueden usar para el procesamiento de imágenes.

Tabla 3.1: Comparación entre algunas plataformas de desarrollo en el mercado

Plataforma	CPU	RAM	Precio (\$)
Jetson tk1	4-Plus-1 quad-core 2.3GHz ARM Cortex A15 CPU	2GB	400
PcDuino V3	AllWinner A20 SoC 1GHz ARM Cortex A7 Dual Core	1GB	140
Raspberry Laptop	700MHz quad-core ARM Cortex-A7 CPU 1.7GHz Intel Core i3	1GB 4GB	70 500

Al final se escogió dos plataformas que tengan al menos los mínimos requerimientos de velocidades de procesamiento y memorias RAM según el artículo de Viola & Jones. Se implementó el algoritmo en una laptop en su primera etapa y luego en un PcDuino V3 en su segunda etapa para analizar y comparar las velocidades de procesamiento.

3.5.2 Selección de cámara

Para la elección de la cámara tenemos que tener en cuenta la velocidad de transmisión, la resolución y los cuadros por segundo que puede entregar. Los requerimientos mínimos del software son: usar una cámara VGA con 15 cuadros por segundo y una conexión compatible a la plataforma de desarrollo, en este caso comunicación USB ya que se podrá usar una laptop o un PcDuino V3. La cámara adquirida se muestra en la figura 3.15.

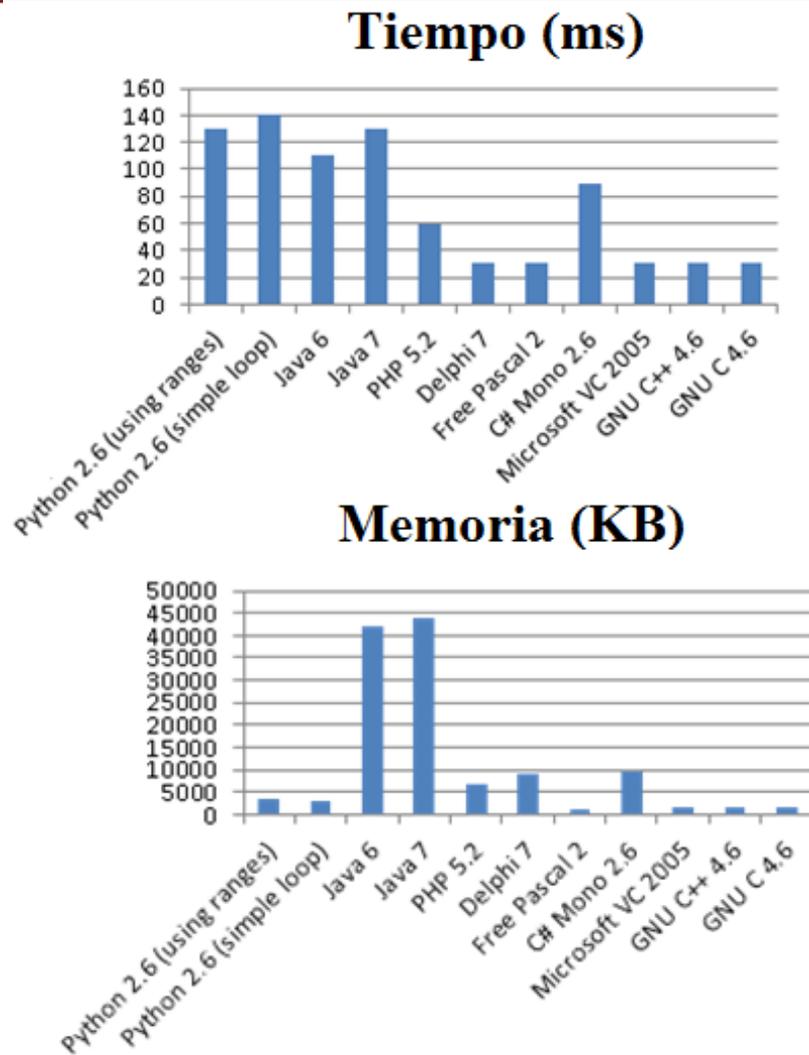


Figura 3.14: Comparación de performance entre los lenguajes de programación [40]

3.5.3 Alimentación del sistema

Según las condiciones de funcionamiento del sistema se plantea que el sistema de visión tenga su propio sistema de alimentación para que se pueda adaptar a cualquier otro dispositivo de asistencia. Para que pueda mantenerse en funcionamiento el sistema, al menos 12 horas, se tiene que hacer un cuadro de potencias (Tabla 3.2) para estimar las características de la batería a seleccionar. En el cuadro de potencia figurara la plataforma de desarrollo (PcDuino V3) y el medidor de voltaje y corriente que indicaran el estado de carga. El medidor indicará el estado de la carga para prevenir que la carga de la batería sea menor al mínimo recomendado por el fabricante. La carga de la cámara del sistema será asumida por el PcDuino por lo cual no se contemplara en la tabla de potencias.



Figura 3.15: Cámara seleccionada (Microsoft LifeCam HD-3000)

Tabla 3.2: Tabla de potencias del sistema de visión implementado en PcDuino V3

Componentes	Cantidad	Voltaje (V)	Corriente Unitaria(A)	Corriente (A)	Potencia(W)
PcDuino V3	1	5	2	2	10
Medidor de voltaje y corriente	1	5	0.3	0.3	1.5
TOTAL				2.3	11.5

Si se tiene 11.5 Watts de potencia de consumo por parte del sistema y se quiere usar el sistema por 12 horas entonces una batería que ofrezca 138 Watts/h. En el mercado tenemos varias posibilidades de las cuales las más clásicas son las de Níquel-Cadmio (NiCd) y entre las más recientes tecnológicas tenemos a las de Litio-Polímero (LiPo). La diferencia fundamental entre estas es su capacidad de carga/descarga siendo el ratio de carga/descarga de las baterías de LiPo mayor que las de NiCd. A mayor ratio más rápido se carga la batería y para la descarga puede entregar más corriente. Por tal motivo se escogerá las de LiPo.

De acuerdo a los cálculos mostrados en la tabla 3.2 la autonomía necesaria para alimentar al sistema por 12 horas corresponde a 138 Watts/h, por lo tanto dicho requerimiento se ajusta a 2 baterías comerciales de Litio-Polímero de 8000mA/h con

3 celdas que ofrecen 177,6 Watts/h. Si consideramos que por sugerencia del fabricante se debe salvar aproximadamente el 20% de la carga de la batería para extender su vida útil por mayor tiempo. Al final se obtendría una potencia de $177,6 \text{ Watts/h} \times 0.8 = 142 \text{ Watts/h}$ con lo que cubrimos con los 138 Watts/h requeridos por el sistema. La batería se muestra en la figura 3.16.



Figura 3.16: Batería seleccionada (Batería LiPo 8000mA/h - 3 celdas)

3.6 *Interface de usuario y modos de operación*

En la computadora el usuario tendrá la posibilidad de observar los movimientos que ha realizado y verificar si estos han sido reconocidos correctamente. En este caso se reconocerán los gestos de la cara y los movimientos de la cabeza los cuales tendrán una numeración dependiendo del modo. Los modos son dos: uno cartesiano, el cual desplaza el efector final en un plano de coordenadas cartesianas, y el segundo modo llamado “asistencia de objetos”. En este último, el objeto a alcanzar se encontrará en una posición específica, es decir todo objeto a ser alcanzado y agarrado tendrá que estar en un lugar fijo.

Se escogerá el modo mediante los movimientos de la cabeza, donde el movimiento de la cabeza hacia la derecha representa la selección del modo 1 y el movimiento de la cabeza hacia la izquierda seleccionará el modo 2.

Para el primero modo que será de los movimientos articulares de la cabeza se ha seleccionado los siguientes números:

- Centro (1) = Cabeza centrada
- Abajo (2) = Mirada hacia abajo
- Arriba (3) = Mirada hacia arriba
- Izquierda (4) = Mirada hacia la izquierda
- Derecha (5) = Mirada hacia la derecha

Estos movimientos pueden ser usados para dirigir el efector final hacia un interruptor de luz, empujar una puerta, abrir una ventana, etc.

El segundo modo realizará trayectorias y movimientos complejos para coger una botella con agua y dar de beber al usuario, luego el brazo regresará a su posición inicial a esperar una nueva selección de modo.

Con estos modos el usuario podrá controlar al brazo **Jaco** sin inconveniente con movimientos cartesianos y movimientos pre-establecidos para un mayor confort. En la figura 3.17 se observa como el sistema de visión detecta un gesto y lo etiqueta con un número.



Figura 3.17: Seguimiento del rostro y predicción de un gesto por parte del sistema de visión

3.7 Integración del sistema

La implementación del sistema se dará mediante la configuración mostrada en la figura 3.17. En ella se puede observar las conexiones de la cámara, el brazo robótico, el regulador de voltaje, el medidor de voltaje y las baterías.

En la figura 3.18 se puede observar la implementación física del hardware del sistema de asistencia para personas con discapacidad.

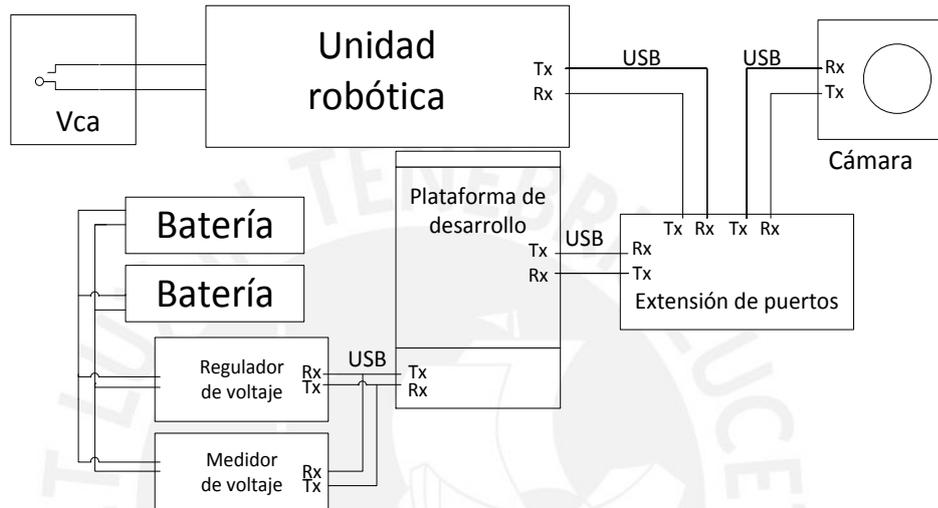


Figura 3.18: Configuración de las conexiones del hardware del sistema de visión para la asistencia de personas con discapacidad

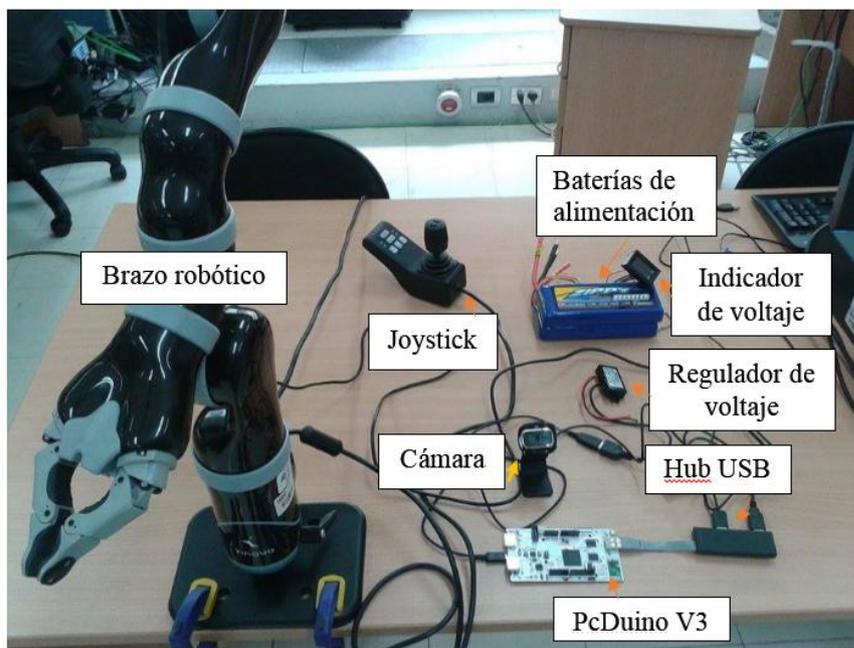


Figura 3.19: Componente para la implementación del sistema de visión

CAPÍTULO 4

Pruebas y resultados

En el siguiente capítulo se mostrarán las pruebas experimentales del sistema de visión funcionando, analizando la velocidad del procesamiento, la practicidad para el uso y el correcto reconocimiento de gestos para el control del sistema. Las pruebas se realizaron con personas con y sin discapacidad.

4.1 *Pruebas*

4.1.1 Comparación de la base de datos

Para el mejor resultado en precisión del sistema de reconocimiento de imágenes se planteó usar una base de datos propia. Dicha base de datos fue creada usando las mismas condiciones con las que se probaría el sistema; es decir: misma resolución, misma calidad y misma iluminación. Para comprobar dicha teoría se hicieron pruebas usando otra base de datos. Se usó la base de datos del Instituto Max-Planck de Cibernética Biológica [41] para entrenar el sistema nuevamente y luego se realizaron nuevamente pruebas de precisión con los mismos sujetos. De esta manera se obtuvo un valor comparable que haga objetivo los resultados.

La base de datos del Instituto Max-Planck de Cibernética Biológica contiene carpetas donde no solo se encuentran gestos y expresiones sino también videos de movimientos articulares de la cabeza.

Las cuales cuenta con 4 directorios cada una. En total la base de datos del Instituto Max-Planck de Cibernética Biológica contiene 42 categorías donde encontramos rostros con las que se puede entrenar algoritmos de procesamiento de imágenes para el reconocimiento de gestos y movimientos de la cabeza.

4.1.2 Análisis de la curva de aprendizaje

La curva de aprendizaje es una herramienta que muestra en el tiempo la evolución del aprendizaje a medida que se repita un ejercicio. Normalmente el tiempo con que se desempeña un trabajo tiene que ir disminuyendo cada vez que se repita debido a que el operario se adapta a la actividad propuesta. Los errores también disminuyen con cada prueba que se haga. Esta herramienta se utiliza en muchas áreas para analizar la productividad de trabajadores, evaluar diseño de mandos, etc. Matemáticamente la curva de aprendizaje en una curva que está representada por la siguiente ecuación [42]:

$$Y_n = Kn^{\log_2 b}$$

Donde:

n : Número de la actividad

K : Horas para realizar dicha actividad

Y_n : Horas para producir la n -ésima actividad

b : Porcentaje de aprendizaje

Por lo visto anteriormente, la evaluación de la funcionalidad y eficacia será demostrada mediante el análisis de la gráfica de la curva de aprendizaje que se obtengan por parte de los usuarios (Ver figura 4.1). A la vez resulta útil estimar cuánto tiempo y cuántas pruebas son necesarias para realizar un conjunto de gestos preestablecidos, los cuales se traduzcan en movimientos del brazo robótico. Como prototipo implementado estos datos serán un indicador del buen funcionamiento del sistema.

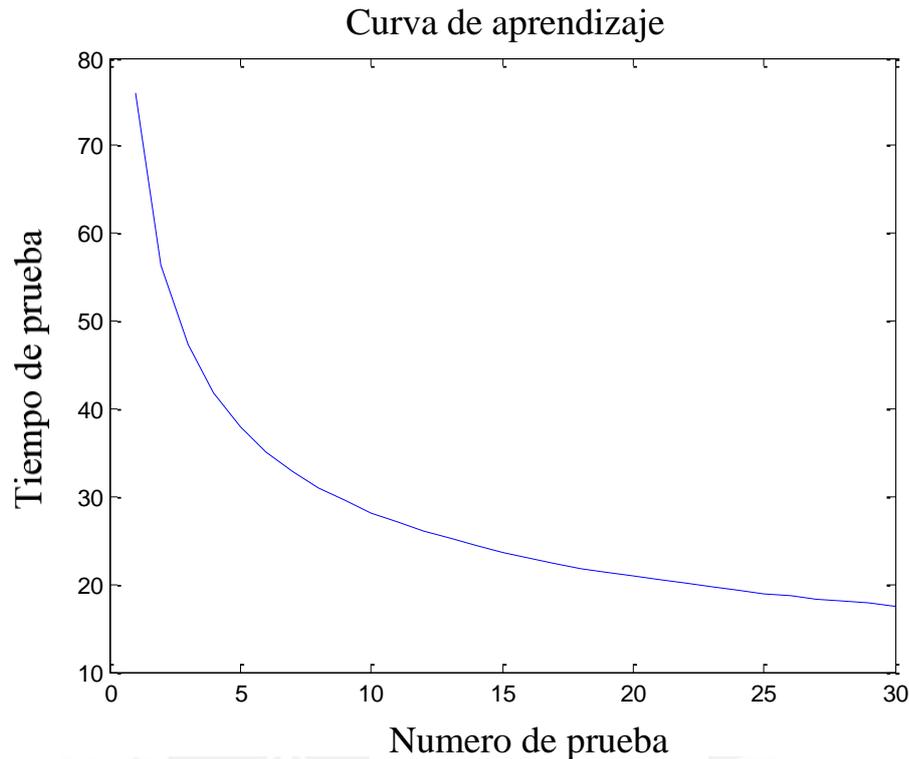


Figura 4.1: Curva de aprendizaje por parte de los usuarios donde T (tiempo de duración de la prueba) y P (# de prueba). Fuente propia

Se ha establecido una prueba que consiste en realizar una secuencia de movimientos para controlar la posición cartesiana del efector final del brazo robótico, que en este caso son: centro, abajo, neutro, arriba, neutro, derecha, neutro e izquierda. El comando del brazo robótico se muestra en las siguientes imágenes de la figura 4.2. Se han llegado a realizar pruebas en 15 usuarios (2 de ellos personas con discapacidad) para controlar el brazo robótico Jaco de *Kinova* las cuales deberán mostrar como el usuario se adapta al sistema a medida que lo utiliza. Cada usuario realizo el control durante 10 veces con lo cual se pudo analizar también las curvas de aprendizaje. Lo cual permitió observar que los usuarios podían controlar todos los movimientos preestablecidos del brazo en orden aleatorio a partir del 5to intento, logrando así una tasa de aprendizaje de 79%. Dichas pruebas se realizarán dentro de un ambiente controlado, el mismo en donde se recopiló la base de datos. Para la captura de data se ha programado al sistema de forma que devuelva el valor proyectado para cada cuadro que es procesado. Estos valores son devueltos en una hoja de cálculos (OpenOffice Calc). Dicha data será analizada y contrastada con el patrón de datos establecidos y del cual se obtendrá la precisión con la que trabaja el sistema.



(a)

(b)



(c)

(d)



(e)

Figura 4.2: Control del brazo robótico mediante movimientos articulares de la cabeza. (a) arriba que indica hacia adelante, (b) neutro que detiene el brazo, (c) izquierda que indica desplazamiento hacia la izquierda, (d) derecha que indica desplazamiento hacia la derecha, (e) abajo que significa atrás

4.1.3 Iluminación del sistema

A la vez también se realizaron pruebas con el luxómetro para determinar cuál debe ser nuestro rango de trabajo con respecto a la iluminación que debe existir para que funcione correctamente el sistema. La iluminación fue medida con respecto a la superficie del rostro, específicamente en las mejillas debido a que es la superficie que más homogénea y en mayor proporción del rostro. El luxómetro fue puesto a la altura de las mejillas mientras el sistema funcionaba y así se iban tomando nota en que valores funcionaba el sistema de visión, como se muestra en la figura 4.3.



Figura 4.3: Pruebas de iluminación para el funcionamiento del sistema

4.2 Resultados

4.2.1 Curva de aprendizaje

De las pruebas se ha llegado a concluir que a medida que se vaya repitiendo la prueba el usuario se vuelve más diestro para controlar el brazo Jaco, y según la figura 4 los 5 movimientos hechos en forma ordenada o desordena tienden a demorar unos 12 segundos a medida que se usa más veces el sistema como se observa en la figura 4.4. Este análisis de la curva de aprendizaje permitió confirmar la viabilidad y la precisión para conseguir el control adecuando de un dispositivo de asistencia mediante este sistema de visión.

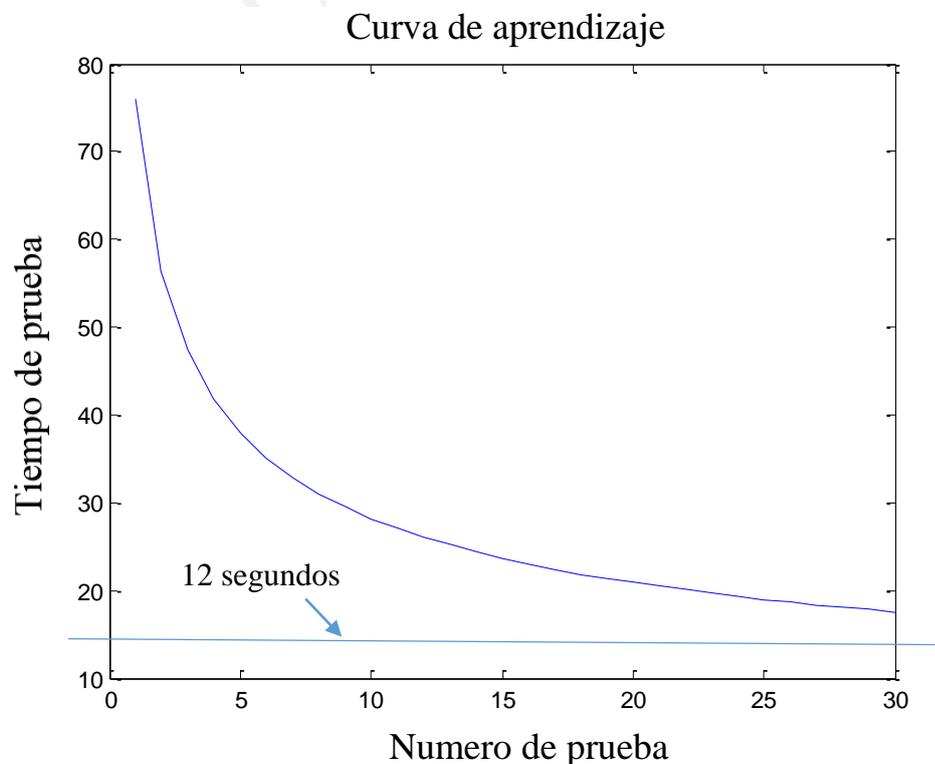


Figura 4.4: Tendencia de la curva de aprendizaje. Fuente propia

4.2.2 Retardo y precisión del sistema

En las pruebas del sistema se logró obtener el retardo que se genera entre la captación del gesto por el computador y el envío del comando de movimiento al brazo. Esto se evaluó mediante software. En promedio todo el sistema tiene un retardo en la respuesta de 790 milisegundos. También se evaluó el tiempo de procesamiento, por cada cuadro, en la laptop y en el PcDuino V3. Los resultados fueron 440 y 690 milisegundos en promedio respectivamente.

La precisión del reconocimiento por parte del sistema se evaluó cuadro por cuadro. Se propuso a los usuarios seguir un conjunto de movimientos los cuales iban siendo registrados por el sistema de manera que luego podrían ser corroborados. Los resultados mostraron que se pudo acertar al 89.2% con respecto a todos los casos. De los cuales el 2.2% se da por el tiempo que le toma a la cámara captar la iluminación del ambiente apenas se enciende y el 8.6% restante son los falsos positivos que se dan por el ruido generado por el fondo de la imagen. El fondo en la imagen puede contener figuras simétricas que son reconocidas por el algoritmo debido a que este trabaja en base a formas simétricas las cuales son características del rostro.

15 usuarios, de las cuales 2 fueron personas con discapacidad, probaron el sistema con lo cual se corrobora que este tipo de sistemas puede ser usado tanto en personas con y sin discapacidad.

4.2.3 Base de datos

Los resultados de probar el sistema de visión entrenado con otra base de datos y realizar las mismas pruebas explicadas anteriormente mostraron que: El algoritmo entrenado con la base de datos del Instituto Max-Planck de Cibernética Biológica tiene 66.05% de precisión a diferencia de la base de datos propia que tiene un 89.2%. Esto se dio debido a que las imágenes de la base de datos Max-Planck contaban con diferente calidad e iluminación.

CONCLUSIONES

- Fue posible implementar un sistema de asistencia para personas con discapacidad usando algoritmos de procesamiento de imágenes y una base de datos local para controlar un dispositivo robótico a través del reconocimiento de gestos faciales y movimientos articulares de la cabeza. Obteniendo una precisión del sistema del 89.2% para reconocer los gestos y movimientos seleccionados. Esto garantizado en ambientes controlados en iluminación constante de al menos 1600 LUX y no más de 3500 LUX, parámetros que se mantienen dentro las indicaciones de la OIT para desempeñar trabajos dentro de oficinas y habitaciones.
- La implementación de una propia base de datos con la misma cámara y calidad de imagen, ayudaron a mejorar los resultados del algoritmo implementado pasando de un 66.05% a un 89.2% de precisión. Esto se dio debido a que tanto el entrenamiento y las pruebas del algoritmo se realizaron con una misma base de datos local que están bajo las mismas condiciones de iluminación y calidad de imagen (Sensibilidad de los sensores de la cámara).
- La zona de seguridad configurada por software puede servir como protección del usuario defendiendo así su integridad y preservando también el buen estado del brazo, evitando coaliciones.

- Fue posible implementar un sistema de visión en tiempo real con un pequeño retardo de 440 milisegundos usando algoritmos complejos de procesamiento de imágenes en una laptop y un PcDuino V3 pero en otros computadores con procesamiento paralelo podría mejorar la velocidad de procesamiento y respuesta del brazo robótico.
- El uso constante de los movimientos articulares de la cabeza y expresiones faciales mejoran el control del movimiento del robot. Según los resultados, se puede lograr realizar 5 movimientos con el robot en 12 segundos al ir aumentando el número de intentos.



BIBLIOGRAFÍA

- [1] C. Serrano Martín and S. Santos del Riego, “Comunicación vocal y no vocal en la parálisis cerebral infantil,” *Rehabilitación*, vol. 35, no. 2, pp. 114–120, 2001.
- [2] T. Kahane and B. Ross-Larson, “World report on disability,” *Disabil Rehabil*, 2011.
- [3] T. Burchardt, *The education and employment of disabled young people*. 2005.
- [4] Who, “World Report on Disability 2011,” *Am. J. Phys. Med. Rehabil. Assoc. Acad. Physiatr.*, vol. 91, no. 7, p. 549, 2011.
- [5] Y. Diskin, B. Nair, A. Braun, S. Duning, and V. K. Asari, “Vision-based navigation system for obstacle avoidance in complex environments,” *2013 IEEE Appl. Imag. Pattern Recognit. Work.*, pp. 1–8, Oct. 2013.
- [6] S. Jeong, T. Kim, J. Lee, S. Lee, and J. Lee, “Design Analysis of Precision Navigation System,” 2012.
- [7] B. K. Kim, H. Tanaka, and Y. Sumi, “Robotic wheelchair using a high accuracy visual marker LentiBar and its application to door crossing navigation,” *2015 IEEE Int. Conf. Robot. Autom.*, pp. 4478–4483, 2015.
- [8] V. Maheu, P. S. Archambault, J. Frappier, and F. Routhier, “Evaluation of the JACO robotic arm: Clinico-economic study for powered wheelchair users with upper-extremity disabilities,” *IEEE Int. Conf. Rehabil. Robot.*, pp. 4–6, 2011.
- [9] B. Blankenship and A. T. Alouani, “Design and Implementation of a Vision System Using a Neural Network Classifier,” pp. 29–33, 1997.
- [10] K. Kpalma and J. Ronsin, “An Overview of Advances of Pattern Recognition Systems in Computer Vision,” *Vis. Syst. Segmentation Pattern Recognit.*, no. June, pp. 169–194, 2007.
- [11] X. Li, M. Yin, and S. Huang, “Design of Target Recognition System Based on Machine Vision,” pp. 0–3, 2010.
- [12] B. Lasher and M. Narayanan, “Vision systems-an overview,” in *Proceedings*

of *NORTHCON'93 Electrical and Electronics Convention*, pp. 118–121.

- [13] K. Onoguchi, W. Mutsumi, and Y. Okamoto, “A Visual Navigation System Using a Multi-Information Local Map,” pp. 767–774, 1990.
- [14] M. L. Della Vedova, T. Facchinetti, a. Ferrara, and a. Martinelli, “Visual Interaction for Real-Time Navigation of Autonomous Mobile Robots,” *2009 Int. Conf. CyberWorlds*, pp. 211–218, Sep. 2009.
- [15] P. Ekman and W. Friesen, “Facial action coding system,” 1977.
- [16] Y.-I. Tian, T. Kanade, and J. F. Cohn, “Recognizing action units for facial expression analysis,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 23, no. 2, pp. 97–115, 2001.
- [17] A. Youssef, H. Shimodaira, and D. Braude, “Head motion analysis and synthesis over different tasks,” *Proc. Intell. Virtual Agents*, pp. 285–294, 2013.
- [18] C. C. J. C. Burges, “A Tutorial on Support Vector Machines for Pattern Recognition,” *Data Min. Knowl. Discov.*, vol. 2, no. 2, pp. 121–167, 1998.
- [19] K. Anderson and P. W. McOwan, “A real-time automated system for the recognition of human facial expressions,” *IEEE Trans. Syst. Man Cybern. Part B*, vol. 36, no. 1, pp. 96–105, Feb. 2006.
- [20] P. H. Adaboosting, J. Guo, S. Member, C. Lin, M. Wu, C. Chang, and H. Lee, “Probability-Based Face Mask Pre filtering and,” vol. 18, no. 8, pp. 447–450, 2011.
- [21] V. Bettadapura, “Face expression recognition and analysis: the state of the art,” *arXiv preprint arXiv:1203.6722*. pp. 1–27, 2012.
- [22] K. Park, R. Park, S. Member, Y. Kim, and S. Member, “Face Detection Using the 3×3 Block Rank Patterns of Gradient Magnitude Images and a Geometrical Face Model,” no. c, pp. 793–794, 2011.
- [23] P. Zhang, “A video-based face detection and recognition system using cascade face verification modules,” *2008 37th IEEE Appl. Imag. Pattern Recognit. Work.*, pp. 1–8, 2008.
- [24] S. Z. Li, “Regularized Transfer Boosting for Face Detection Across

- Spectrum,” *IEEE Signal Process. Lett.*, vol. 19, no. 3, pp. 131–134, Mar. 2012.
- [25] H. Tao and T. S. Huang, “A Piecewise Bezier Volume Deformation Model and Its Applications in Facial Motion Capture,” *Adv. Image Process. Underst.*, vol. 52, 2002.
- [26] P. S. Aleksic and A. K. Katsaggelos, “Facial Animation Parameters and Multistream HMMs,” vol. 1, no. 1, pp. 3–11, 2006.
- [27] A. Gams, T. Petrič, J. Babič, and L. Žlajpah, “Augmenting movement imitation with reflexive stability behavior,” in *2011 IEEE International Conference on Robotics and Biomimetics, ROBIO 2011*, 2011, pp. 204–209.
- [28] T.-C. L. T.-C. Liu and M. C. M. Chang, “Human-Robot Interaction Research Issues of Educational Robots,” *2008 Second IEEE Int. Conf. Digit. Game Intell. Toy Enhanc. Learn.*, pp. 215–216, 2008.
- [29] J. H. J. Han, M. J. M. Jo, S. P. S. Park, and S. K. S. Kim, “The educational use of home robots for children,” *Rom. 2005. IEEE Int. Work. Robot Hum. Interact. Commun. 2005.*, pp. 378–383, 2005.
- [30] M. a. Goodrich and A. C. Schultz, “Human-Robot Interaction: A Survey,” *Found. Trends® Human-Computer Interact.*, vol. 1, no. 3, pp. 203–275, 2007.
- [31] S. L. Anderson, “Asimov’s ‘Three Laws of Robotics’ and machine metaethics,” *AI Soc.*, vol. 22, no. 4, pp. 477–493, 2008.
- [32] E. Z. McClave, “Linguistic functions of head movements in the context of speech,” *J. Pragmat.*, vol. 32, no. 7, pp. 855–878, Jun. 2000.
- [33] B. D. Goldstein, P. Boffetta, J. J. Hurrell, L. R. Murphy, and S. L. Sauter, “Parte VI, Riesgos Generales,” *Enciclopedia de la salud y seguridad en el trabajo*. Jeanne Mager Stellman, 2012.
- [34] P. S. Aleksic and K. Katsaggelos, “Comparison of MPEG-4 facial animation parameter groups with respect to audio-visual speech recognition performance,” *IEEE Int. Conf. Image Process. 2005*, pp. III–501, 2005.
- [35] L. Shen and L. Bai, “A review on Gabor wavelets for face recognition,” *Pattern Analysis and Applications*, vol. 9, no. 2–3, pp. 273–292, 2006.

- [36] J. Canny, “A computational approach to edge detection.” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 8, no. 6, pp. 679–698, 1986.
- [37] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, “Eigenfaces vs.~{Fisherfaces}: Recognition using class specific linear projection,” *Pami*, vol. 19, no. 7, pp. 711–720, 1997.
- [38] P. Viola and M. Jones, “Rapid object detection using a boosted cascade of simple features,” *Proc. 2001 IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognition. CVPR 2001*, vol. 1, pp. I-511–I-518, 2001.
- [39] ©. Robotnik, “Brazo robótico KINOVA JACO² Robotnik.” [Online]. Available: <http://www.robotnik.es/brazos-roboticos/kinova-jaco/>.
- [40] helloacm.com, “A Quick Performance Comparison on Languages at Codeforces.” [Online]. Available: <https://helloacm.com/a-quick-performance-comparison-on-languages-at-codeforces/>.
- [41] M. Breidt, D. W. Cunningham, and C. Wallraven, “Face Video Database of the Max-Planck Institute for Biological Cybernetics.” [Online]. Available: <http://vdb.kyb.tuebingen.mpg.de/list.php>.
- [42] L. E. Yelle, “THE LEARNING CURVE: HISTORICAL REVIEW AND COMPREHENSIVE SURVEY,” *Decis. Sci.*, vol. 10, no. 2, pp. 302–328, Apr. 1979.