DEPARTAMENTO DE AUTOMÁTICA, INGENIERÍA ELECTRÓNICA E INFORMÁTICA INDUSTRIAL

Escuela Técnica Superior de Ingenieros Industriales

SISTEMA DE VISIÓN PARA UN ROBOT SOCIAL

Autora: Mª Guadalupe Sánchez Escribano

Director: D. Ramón Galán López

Agradecimientos

El mayor esfuerzo de este trabajo están siendo estas líneas. Tengo tantas personas a quienes agradecer, que no encuentro motivo que justifique el orden. Así que haré un recorrido en el que pueda ir expresando mi gratitud por igual a todos: Es posible que, sin alguno de los impulsos que me habéis dado, este trabajo no hubiera salido adelante.

Comenzaré por dar las gracias a mi tutor, D. Ramón Galán, y no sólo por su labor como director. Quiero hacerlo especialmente por confiar siempre en mí, y por entender el silencio en el que trabajo.

A Miguel Hernando, por tantas horas dedicadas a mis ilusiones, a mis proyectos y a tranquilizarme en mis peores ratos. Por ofrecerme sus ideas y ayudarme personal y profesionalmente. Y, por supuesto, por compartir conmigo la pasión por la música y el piano.

A Paloma, que ya se ha convertido en una amiga a la que admiro, respeto y quiero. Gracias por ser tan buena conmigo, por creer en mí y por apoyarme siempre.

A Iñaki, porque no me ha faltado nunca su confianza en lo que hago y su defensa a mi trabajo. Gracias porque nunca me ha faltado tu llamada cuando me has visto decaída y por la alegría que nos aportas siempre a todos.

A José Emilio, porque sin darse cuenta y con pequeños consejos, en muchas ocasiones ha hecho que cambie de opinión, y me ha dado el impulso que he necesitado para reaccionar.

A Carlos, como amigo y como investigador. En un punto, a mitad de camino entre ambas cosas, me ha dedicado todo el tiempo que he necesitado para entender esa perspectiva desde la que ven las cosas en su equipo. Gracias por tanto como me has aportado en este trabajo, por las horas empleadas en explicarme lo que te he preguntado, por ilusionarte y por compartir mis dudas. Una parte de este trabajo está dedicada a tu esfuerzo.

A Ignacio López: Su tesis constituye una base fundamental en mi trabajo. Gracias no sólo por dedicar parte de tu tiempo a explicarme lo que no comprendía bien de tu estudio. Te agradezco especialmente tu comprensión y tus palabras de ánimo cuando has notado que las necesitaba.

A Ricardo Sanz, porque sentir su apoyo es una razón para continuar.

A todos vosotros, y al resto de compañeros del departamento, por ofrecerme las mejores y más divertidas tertulias que haya podido compartir.

A mis amigos:

A David, por comprender mis reservas, y por darme tiempo, fuerza y apoyo durante estos últimos meses.

A Silvia, por toda la dedicación que tuvo conmigo cuando lo necesitaba y no nos conocíamos. Y por el inmenso apoyo que me da casi a diario. A Félix, uno de mis pilares. Gracias por estar pendiente de mí en todo momento, por ofrecerme tu ayuda y por interesarte en todo lo que hago. A Agustín Gajate, porque desde el primer día que nos conocimos me dio su confianza, me animó a seguir investigando y me impulsó en cada cosa que hice. Y a José Luís, mi rubito, como yo le llamo. Cada paso que doy se lo tengo que dedicar. Todo lo que tengo ahora lo debo a la fuerza con que me empujó a conseguirlo.

Aprendo mucho de vosotros, y os sigo necesitando.

A mis padres, a mi hermana y a mi familia. Se que a veces les cuesta entender los motivos por los que sigo estudiando, como ellos dicen. Por eso les agradezco especialmente tanto apoyo y tanta confianza como me dan. Les dedico especialmente este trabajo.

Al Grupo de Control Inteligente, y a todos los que de alguna manera, personal o profesionalmente, me habéis ayudado a realizar este proyecto,

Gracias.

Prólogo

En el trayecto de las últimas décadas, la ciencia ha buscado de forma incesante las bases de una robótica con capacidades cognitivas, y continúa en su intento de construir máquinas que puedan interactuar con los humanos de forma intuitiva y consciente. Sin embargo, es algo que actualmente se acerca más a la ciencia ficción que a la capacidad real de consecución humana.

La realidad de este objetivo, es que no podemos asociarlo a los hechos que habitualmente consideramos como éxitos en este campo. Se han desarrollado modelos que emulan determinadas conductas, procedimientos y reacciones. Pero casi siempre se anula cualquier otra interacción cognitiva, presente en la realidad humana e influyente en nuestro comportamiento. Los modelos, generalmente no reflejan de forma correcta ni real nuestras pautas de actuación, de toma de decisiones e interacción.

Sin embargo, nos impulsa una creencia apasionada en la posibilidad de transmitir inteligencia a una máquina, de comprender mejor nuestro cerebro y su funcionamiento, y desarrollar una emulación. Aunque nos lleve a plantearnos muchas cuestiones acerca de la convivencia con una nueva familia de seres artificiales capaces de pensar, reaccionar y tomar decisiones por sí mismos.

El sueño de una máquina consciente está determinado en gran medida por ciertos obstáculos que cierran el camino: un gran número de características esenciales del comportamiento y cognición humanos que son aún desconocidos o, en el mejor de los casos, que apenas se comprenden.

Desde el siglo pasado hasta ahora, multitud de científicos en áreas como la ingeniería, la psicología y la neurociencia, han trabajado en diferentes campos del conocimiento, para hacer encajar todas las piezas en un modelo más completo. Un modelo con el que entender el funcionamiento del cerebro humano y, al fin, poder implementarlo en una máquina. Sin embargo, las soluciones más o menos complejas a que se ha llegado no han conseguido este objetivo, generalmente por quedarse ancladas en interrogantes que aún no tienen respuesta.

Esta inquietud supone el punto de partida de este trabajo.

CAPÍTULO 1

Introducción

1.1 Motivación del Proyecto Fin de Master

ROBOT SOCIAL

Un robot social debe tener capacidades de detección y entendimiento de comportamientos e indicaciones humanas, convenciones sociales elementales tales como expresiones faciales, movimientos de manos o miradas, e interacción sin necesidad de instrucciones o entrenamiento especiales. Asimismo deben ser capaces de emplear estos convenios para llevar a cabo intercambios interactivos humano-robot [128].

En este apartado se analizan tres puntos importantes que justifican el estudio realizado en este trabajo. En primer lugar una reflexión general acerca del aprendizaje en un robot social. En un segundo punto se presentan algunas ideas del aprendizaje humano y de la simulación de este comportamiento en una máquina. Por último, mostramos la influencia de las capacidades visuales en el aprendizaje y, de la misma manera que antes, la posibilidad de trasladar estas características a los sistemas artificiales.

Con todo ello se pretende justificar el estudio del aprendizaje en máquinas y robots a través de la visión, tema del que versa este trabajo.

1.1.1 Aprendizaje en un robot social

Los primeros pasos que se dieron en este campo, allá por los años cuarenta del siglo pasado, estaban dirigidos a construir **robots inspirados biológicamente**. Desde los primeros robots sociales conocidos, las tortugas de Grey Walter (Fig. 1), hasta hoy, las mayores fuerzas de impulso y cimientos de su desarrollo han sido la fascinación por conseguir capacidades de interacción y de decisión. A medida que han ido evolucionando las técnicas de Inteligencia Artificial, lo han hecho también los modelos de comportamiento de estos robots siendo cada vez más complejos.

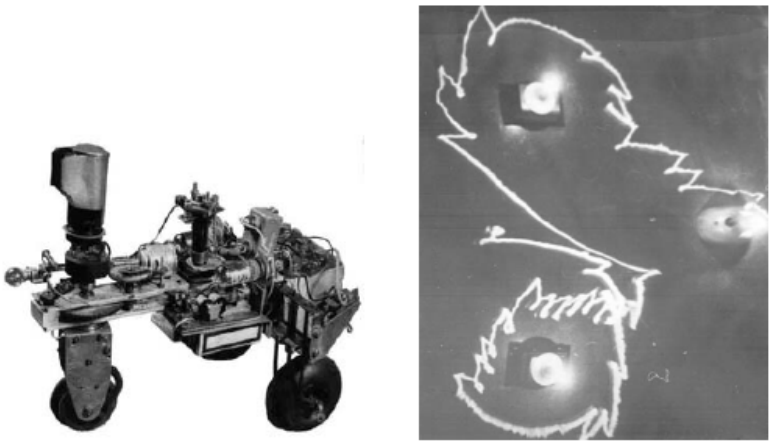


Fig. 1 Precursores de los Robot Sociales: Las Tortugas de Grey Walter (Elmer y Elsie) en 1940. La imagen de la izquierda muestra a Elsie y su concha en el año 1950 y en la de la derecha, un comportamiento que Gray Walter calificó de gran importancia: Elsie toma decisiones entre dos alternativas

Los robots implementados para trabajar de forma colectiva requieren modelos y técnicas diferentes al tema que nos ocupa. Centraremos nuestra atención en los robots que trabajan individualmente: es fácil darnos cuenta que la importancia de estos robots aumenta en áreas donde es deseable la interacción social. Por poner algunos ejemplos clásicos, podemos nombrar: (1) Robots utilizados como

"máquinas persuasivas", B. Fogg [2], para cambiar un comportamiento, sentimiento o actitud en una persona, (2) Los *"robots mediadores"* utilizados en terapias de autismo, I. Weery [3], o (3) Robots empleados como *"avatares"*, E. Paulos et. [4], en representación de humanos.

Sherry Turkle [130], desarrolladora del *Instituto de Tecnología de Massachussets* (MIT Cambridge, USA) y psicoanalista, escribió en 1984 *"The Second Self: Computers and the Human Spirit"* [129], libro de culto en el que reflexiona acerca de la posibilidad de que un robot pueda llegar a "*ser*" como un humano. Las bases de esta reflexión se sustentan en el trabajo de otros científicos que, como ella, estudiaron la posibilidad de implementar sentimientos en computadores y robots. Tras un largo periodo de investigación tanto con profesionales informáticos como con usuarios comunes, llegaron a una reveladora conclusión: El cerebro humano no tiene las condiciones psicológicas adecuadas para diferenciar un sentimiento provocado en nosotros por otro ser humano, de uno simulado por una computadora. No podemos distinguir -en este sentido- entre una mente natural y una artificial.

Esta conclusión supuso el comienzo de una serie de consecuencias significativas en los campos de la Inteligencia Artificial y la Robótica: Quedaba demostrada la capacidad de modificar el comportamiento humano "*engañando*" al cerebro con simulaciones de sus propias facultades. Como ejemplo, basta recordar la reacción del campeón mundial de ajedrez Garry Kasparov (Fig. 2) cuando perdió la partida ante Deep Blue de IBM.

Fig. 2 Kasparov contra Deep Blue, Nueva Cork 1997

Para lograr que un robot sea capaz de "*equivocar*" de esta manera a las personas es necesario que él mismo pueda modificar sus propias habilidades de forma autónoma. Los humanos modificamos nuestro comportamiento mediante

experiencias, relaciones o una interacción entre ambas, a través de una mecánica de aprendizaje que nos da la posibilidad de adquirir ese conocimiento.

Si se desea que un robot sea capaz de modificar su comportamiento, es necesario que lleve implementado un sistema de aprendizaje, K. Dautenhahn [5] J.Zlatev [6], que mejore la ejecución de las tareas que tenga programadas inicialmente, e incluso que adquiera otras nuevas.

1.1.2 Aprendizaje

W. Grey Walter [131] dijo que el verdadero futuro de los *robots inteligentes* está en la conjunción de la *Inteligencia Artificial* y la *Robótica*.

Nuestro cerebro es capaz de realizar correctamente cualquier tarea cotidiana. En base a los numerosos estudios médicos y psicológicos que se han realizado en este campo, conocemos algunas características de nuestro sistema neuronal que nos permiten destacar algunos puntos importantes de su funcionamiento:

(1) Las redes neuronales de nuestro cerebro son capaces de adaptar su configuración, es decir, son capaces de *aprender* a partir de una experiencia y modificar su estructura en un proceso de aprendizaje.

(2) Somos capaces de *generalizar* y resolver con alto grado de variabilidad.

(3) Nuestro entretejido neuronal puede *reconocer patrones*, visuales o de cualquier otro tipo, siempre que la información llegue a nuestro cerebro adecuadamente.

Hasta ahora, el diseño ha estado ligado al estudio de simples patrones de comportamiento sin más reto que la construcción de robots con nociones sociales intrínsecas, capaces de crear vínculos y habilidades con las personas y de mostrar empatía y entendimiento. Además, en la mayoría de los casos es necesario incrementar su efectividad.

Por ejemplo, en relación a implementar pautas de comportamiento "*naturales*" o *habilidades sociales sofisticadas*, tales como la capacidad de reconocer contextos sociales, entornos o personas, S. Restivo [7].

Tampoco debemos dejar de lado el hecho de que los usuarios con quienes interactúa el robot van a tener rangos y dominios muy diferentes, tales como la edad, la cultura o un marco social determinado. Actualmente están limitados a entornos de acción definidos, como museos o ferias, y un trato idéntico para todas las personas, sin tener en cuenta rango social, raza, etc. Tan pronto como estos robots comiencen a formar parte de la vida de las personas será necesaria la capacidad de ofrecer un trato individualizado, K. Dautenhahn [8].

La mayor parte de los modelos de comportamiento actuales están dirigidos al desarrollo de aplicaciones que necesitan algún tipo de inteligencia para concluir en alguna solución. Están orientados a la adquisición de conocimiento procedente del dominio de interacción, para facilitar una inferencia con el mínimo coste computacional.

Una de las primeras observaciones que obtenemos de estas aplicaciones, es que sus límites están impuestos por el conocimiento que hayamos integrado en el sistema. No van a poder resolver problemas para los que no hayan sido programados. Sin embargo, si lo que queremos son robots *inteligentes*, deberían ser capaces de ir más allá de éste límite. De hecho nos "*parece"* más inteligente un robot con capacidad de adaptación e integración de conocimiento, que pueda resolver problemas que se van presentando dinámicamente, que uno que realice perfectamente tareas fijas y programadas.

Entre las numerosas utilidades que podemos dar a un sistema de aprendizaje artificial, nos centramos en las *Aplicaciones Autoadaptables*. Muchos sistemas trabajan mejor si son capaces de adaptarse a unas circunstancias diferentes a las consideradas en su desarrollo inicial.

1.1.3 Vision para el aprendizaje

Como análisis último y, para dar paso al siguiente punto, volveremos de nuevo a los humanos. Ya hemos dicho anteriormente que el aprendizaje es un medio de adquisición de conocimiento a través de nuestra experiencia e interacción con el

entorno. Esta interacción la realizamos a través de nuestros sentidos, el medio por el que llegan las señales con la información necesaria para que nuestro sistema cerebral trabaje.

La visión es un elemento clave dentro del aprendizaje humano. En este campo, aprender depende directamente de la capacidad de ver e interpretar. Percibir correctamente y analizar lo que vemos, depende directamente de nuestra compleja serie de conexiones neuronales y de una maduración adecuada del sistema nervioso central.

En un sistema artificial la percepción de la señal no va a suponer un problema. La complejidad de la solución va a radicar en lo que se refiere a *la interpretación de la imagen*, y es aquí donde nos vamos a centrar. Es cierto que conocemos el proceso mental mediante el que decodificamos y guardamos –a corto y largo plazo- toda la información visual. Igualmente, disponemos de modelos cognitivos que describen y emulan correctamente la gestión del conocimiento en un proceso de aprendizaje. Pero esto no siempre es suficiente y las soluciones no siempre ofrecen los resultados deseados.

Queda mucho camino por recorrer. Sin embargo, podemos tomar como objetivo la construcción de un sistema que interactúe con su entorno más directo, que conozca su dominio de actuación y que además pueda aprenderlo. Y todo ello, con la información que reciba a través de un sistema de *Visión por Computador*, gestionada a través de un adecuado modelo de *Aprendizaje Automático*.

1.2 Marco de Trabajo

Uno de los objetivos del proyecto ROBONAUTA, realizado por el grupo de investigación en Control Inteligente de la UPM, es el de implementar un sistema cognitivo que permita la toma de decisiones basada en tres aspectos*:* (1) *Conocimiento Estructurado*, (2) *comportamiento por defecto* y (3) *aprendizaje*.

En la actualidad se dispone de una versión de arquitectura software basada en agentes para el control de un robot guía. Fue desarrollada dentro del proyecto URBANO. Sin embargo, durante su proceso de operación, se puso de manifiesto la necesidad de estructurar el conocimiento, más organizado.

Algunas de las pretensiones iniciales, eran que la plataforma pudiera interactuar con el público de forma inteligente, respondiendo un determinado rango de cuestiones que se le pudieran plantear. O la generación automática de las presentaciones que, en cada entorno y situación, tuviera que realizar.

El diseño de los agentes software tiene que ser tal, que permita trabajar de manera coordinada para realizar las siguientes tareas:

(1) Desarrollar una ontología que permita almacenar, consultar y abstraer el conocimiento, ofreciendo al sistema robot un comportamiento por defecto.

(2) Obtener un modelo emocional para conseguir una interacción más cercana con el público y, en el sentido contrario, que se vea afectado por la reacción de este público e influya en su comportamiento.

(3) Definir un sistema de toma de decisiones, para que la plataforma elija la tarea adecuada en cada situación, siendo posible el escalado, el incremento de la información y la inclusión de índices de calidad, así como la modificación de los mismos.

(4) Conseguir un mecanismo de aprendizaje de nuevos conceptos que afecten a las tareas, al modelo emocional, al conocimiento sobre el dominio y a la valoración de los objetivos vitales del robot. Internet va a ser la fuente fundamental de información.

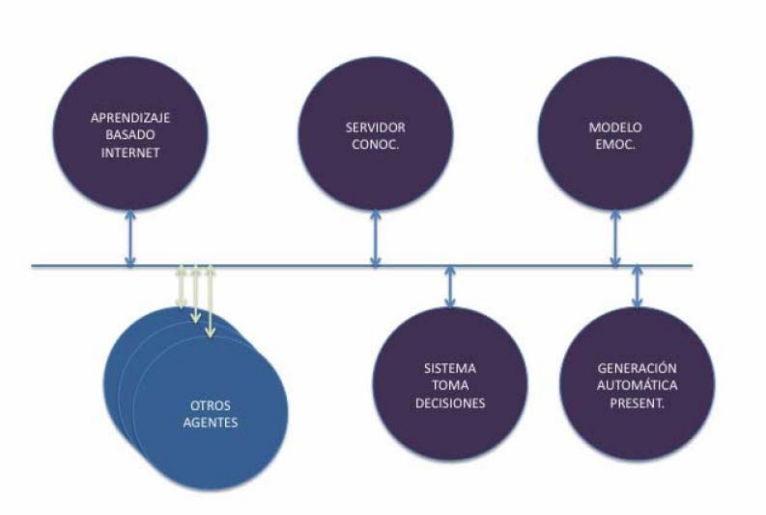


Fig. 3 Agentes del sistema cognitivo de ROBONAUTA.

(5) Un sistema de generación automática de presentaciones, que tenga en cuenta la información disponible, las características del público, el tiempo y los criterios de calidad de la propia presentación.

Dada la amplitud de este objetivo, se ha dividido en objetivos parciales, y se ha dedicado personal específico para cada uno de ellos.

Se esta diseñando una nueva arquitectura software basada en agentes, que utilizan SOAP como mecanismo de integración. Diferentes tesis doctorales y proyectos fin de carrera cubren estos objetivos. La Fig. 3 muestra un esquema de los agentes relacionados con el Sistema Cognitivo

Objetivos de los agentes software

Definimos brevemente los objetivos de cada uno de los agentes implicados en el desarrollo:

A) Ontología para Formalización del Conocimiento

Tesis doctoral en desarrollo por D. Jaime Gómez, entre cuyos objetivos esta la propuesta de una ontología, que permita tanto la representación de conceptos y de sus relaciones, como la obtención de categorías nuevas.

El Proyecto Fin de Carrera de D. Carlos Florit, dio lugar a un prototipo que demuestra viabilidad y potencia en este tipo de estructuras. Se construyó un agente que se integra en la arquitectura SOAP y que actúa como un *servidor de conocimiento*.

B) Sistema Cognitivo para la toma de decisiones

Tesis doctoral en desarrollo por D. Ignacio Chang, cuyo objetivo fundamental es el diseño de un sistema cognitivo que permita al robot una toma de decisiones.

Las decisiones afectan a cada una de las tareas de medio y alto nivel que el robot puede realizar. Cada tarea, necesita para su realización una serie de recursos que deben estar disponibles. Asimismo, realizada la tarea, los recursos disponibles habrán cambiado.

La elección de la tarea se basa en criterios que hacen más recomendable realizar una u otra de acuerdo a un *índice de calidad*.

En un entorno no estructurado, tanto los criterios de valoración del índice como las tareas pueden cambiar, y el sistema debe aprender a tomar decisiones teniendo en cuenta esta dinámica de cambio.

La Tesis propone la utilización de *lógica borrosa* como medio para la *selección de tareas*, así como para la valoración de la calidad del trabajo realizado. Se usan *algoritmos genéticos* como mecanismo de adaptación a nuevas situaciones.

C) Modelo emocional del robot

Tesis doctoral en desarrollo por Marta Álvarez, que pretende el diseño de un modelo dinámico que dote al robot de una serie de emociones que, por otra parte, van a influir en la realización de las tareas.

El objetivo de este modelo es analizar la influencia de estas emociones en la interacción *público vs. Robot-guía*, y verificar el aprendizaje de un carácter *emotivo* valorado positivamente por el público.

Como índice de calidad se ha propuesto la búsqueda de la felicidad, e irá modificando el modo de guiar las visitas por parte del robot.

D. Mariano Aranguez, realizó dentro de su Proyecto Fin de Carrera un agente integrado en la arquitectura SOAP. Proporcionaba al robot un modelo emocional, fácil de modelar y de modificar. Ha demostrado su potencial para conseguir una mejor interacción con el público.

D) Generación Automática de Presentaciones

Tesis doctoral en desarrollo por D. Javier Rainer. La pretensión es identificar las características que el público valora positivamente en una presentación, durante una visita a un museo o una feria.

Será necesario por tanto, diseñar y mantener un modelo de opinión del público sobre la calidad de las presentaciones. Dos aspectos fundamentales para la tesis los constituyen el modelo de patrón que va a seguir la presentación y el conocimiento disponible sobre la información que va a ser utilizada.

El sistema será diseñado de modo que permita reutilizar antiguas presentaciones.

E) Aprendizaje desde Internet

D. Rafael León esta desarrollando su tesis doctoral teniendo, como primer objetivo, el estudio de la viabilidad para realizar un aprendizaje de nuevos conceptos y sus relaciones, a partir de la información disponible en Internet.

Dos objetivos fundamentales son la búsqueda de frases sintácticamente correctas y semánticamente entendibles. Dado que la aplicación básica del robot es la de guía en ferias y museos, se ha querido dotar a la plataforma de un sistema de aprendizaje automático sobre el dominio de conocimiento.

En este marco de trabajo, se quiere inducir al robot un sistema de aprendizaje automático, basado en visión por computador.

1.3 Objetivos de Proyecto

El objetivo final de este proyecto es la realización de un sistema de visión por computador para una plataforma de robot social, y el estudio de las necesidades que tendría un sistema de reconocimiento y aprendizaje basado en visión.

El prototipo implementado deberá reconocer e identificar personas dentro de un grupo reducido de identidades, así como gestos para la interpretación de órdenes sencillas. Por otra parte, deberán ser estudiadas las necesidades de un sistema de aprendizaje en lo referente a la percepción, la interpretación de la escena y la capacidad de identificar objetos no conocidos.

La idea es plantear el sistema de aprendizaje desde una perspectiva cognitiva donde se tengan en cuenta los diferentes factores de la percepción, y no sólo la imagen y

su procesamiento tradicional. El propósito es utilizar no sólo la información intrínseca al robot, sino los factores extrínsecos que afectan igualmente a la percepción y la interpretación de la imagen.

Las técnicas de visión por computador tradicionales obtienen un relativo grado de éxito siempre que sean mantenidas las condiciones de entorno e iluminación para las que fueron diseñadas. Los desarrollos que se presentan en el estado del arte, se han realizado casi siempre en base a procedimientos matemáticos, estadísticos, técnicas de búsqueda, métodos heurísticos, inteligencia artificial, sistemas expertos, etc. Todos ellos válidos y con soluciones notablemente satisfactorias en las soluciones para las que fueron desarrollados. Pero con baja capacidad de modificación autónoma necesaria, por otra parte, para plantear un sistema de aprendizaje.

El enfoque de este trabajo es un estudio, tanto de los algoritmos de visión por computador que procesen la información intrínseca al sistema, como de las técnicas de visión cognitiva y modelos de percepción, que incluyan la información extrínseca necesaria para lograr un mejor acercamiento a la interpretación humana.

Adicionalmente, se requiere un prototipo en el que se muestren y estudien los resultados del procesamiento de la imagen para el reconocimiento e identificación de un grupo reducido de personas y de gestos manuales.

En una primera fase se realizará un estudio del estado del arte de la robótica y la influencia de la visión en sus aplicaciones, de los algoritmos de procesamiento de la imagen y de los modelos de percepción. Asimismo se estudiarán las claves perceptivas de la visión para obtener información tridimensional de la escena, utilizando información bidimensional.

En una segunda fase de trabajo se plantea el estudio de extracción de características con visión por computador, enfocadas a la idea de imprimir capacidades de búsqueda y aprendizaje a través de la visión. El objetivo de esta fase, es desarrollar algoritmos matemáticos que permitan reconocer un grupo reducido de personas del entorno, y la segmentación de las características necesarias para definir posteriormente órdenes sencillas a través de gestos de la mano. Es decir, técnicas de reconocimiento, identificación, seguimiento facial y gestual. Estas capacidades se extenderán posteriormente a su entrenamiento en todo lo referente a expresiones faciales y movimiento de manos y brazos, dejando esta parte como posibles desarrollos futuros.

Sabemos que ninguno de estos algoritmos resuelve problemas tales como la conciencia del desconocimiento o las capacidades de aprendizaje. Los humanos

somos capaces de percibir el contorno de una figura y delimitarlo frente al resto del campo visual para obtener información sobre su forma. Parte de esta capacidad está ligada a un conocimiento previo, y podemos resolver sin necesidad de percibir toda la información en un momento dado. Sin embargo, es información necesaria.

Este fenómeno, a través del que un perfil visual se diferencia del campo total de la percepción, emergiendo como figura destacada de un fondo, resulta de vital importancia para la estructuración del campo visual en unidades perceptuales coherentes.

Por lo tanto, en una tercera fase del proyecto se realizará el estudio de las técnicas actuales de aprendizaje con ontologías y la propuesta de utilizar los modelos de percepción y claves perceptivas estudiados en la primera parte. El objetivo es plantear un nuevo concepto de visión por computador con las posibilidades que tenemos hasta el momento. Un modelo que integre la capacidad cognitiva con los modelos matemáticos de inferencia.

1.4 Estructura del documento

El documento se estructura en los siguientes capítulos:

(1) Capítulo 1 Introducción

(2) Capítulo 2 Estado del Arte

(3) Capítulo 3 Modelo de Percepción Visual

(4) Capítulo 4 Identificación facial con Visión

(5) Capítulo 5 Extracción de características gestuales

(6) Capítulo 6 Aprendizaje Basado en Visión

(7) Capítulo 7 Conclusiones y Trabajos Futuros

En el primer capítulo se presenta la motivación del proyecto y la justificación del aprendizaje en un robot social. Se describen las líneas generales del aprendizaje y la influencia de la visión en él. Los últimos apartados del capítulo tienen que ver con el marco de trabajo en el que se realiza, los objetivos y la estructura del documento.

En el segundo capítulo, presentamos el estado de la técnica en los campos que afectan al proyecto. Comienza con una revisión acerca de los robots socialmente interactivos, de su morfología y las consecuencias para la interacción con los humanos, y de las aplicaciones de visión más utilizadas en estos desarrollos. En una segunda línea de estudio se presenta el estado del arte de las técnicas de visión por computador y procesamiento de la imagen, enfocando posteriormente estos desarrollos al reconocimiento facial, gestual y de movimiento.

El tercer capítulo realiza un estudio de los modelos de percepción, utilizando uno de ellos como ejemplo de aplicación. Asimismo, se estudian las claves perceptivas de la visión, que transmiten información tridimensional de la escena utilizando únicamente información bidimensional. Se describe el modelo de percepción (elegido por su cercanía en cuanto a objetivos) sobre el que se realiza un ejemplo de implementación de una parte de nuestro sistema.

En el cuarto capítulo, se describen completamente los algoritmos de visión implementados para el reconocimiento facial. El primer apartado está dedicado a la detección genérica de caras con plantillas *Haar-Training,* detección por distribución estadística con el algoritmo *CamShift* y el algoritmo propuesto para el trabajo conjunto de ambos. El segundo apartado describe la técnica implementada de identificación de personas con el *Análisis de Componentes Principales (PCA)*. Por último, el tercer apartado se dedica a la descripción de las clases y funciones realizadas en la implementación.

El capítulo quinto, es el resultado de los estudios e implementaciones de la extracción de características, para el futuro desarrollo de un sistema de análisis y aprendizaje gestual. En el primer apartado se describe el algoritmo de análisis de movimiento implementado. El segundo está dedicado al método propuesto para la identificación de gestos manuales, y los algoritmos para aislar la mano del resto de la escena en tiempo real, seguirla e identificar el movimiento. Del mismo modo que en el capítulo anterior, el último apartado está dedicado a la descripción de las clases y funciones implementadas.

El capítulo sexto estudia el aprendizaje y la interpretación semántica de la imagen desde la perspectiva de las ontologías visuales, intentando eliminar los niveles más bajos del procesamiento. El primer apartado describe el concepto de ontología, agente y las ventajas de su uso. El segundo está dedicado a la interpretación

semántica de la imagen. Está enfocado a dar una idea del significado de *concepto semántico* de una imagen y una discusión acerca de las necesidades de los sistemas de percepción artificiales. El último apartado realiza un estudio dentro del marco teórico de la percepción, planteando la justificación de su uso con un ejemplo de aplicación.

Por último, el capítulo séptimo se dedica a las conclusiones que se obtienen de la realización de este trabajo, y a los desarrollos futuros que se pueden derivar del mismo.

CAPÍTULO 2

Estado del Arte

En el presente trabajo se abordan una gran variedad de temas, desde la robótica social y su interacción con las personas, los sistemas de visión y su influencia en los robots sociales, hasta el aprendizaje y la percepción. Ha sido necesario el estudio de cada uno de ellos aunque el resultado final del proyecto, lo hayan marcado en mayor medida la visión, el aprendizaje y la percepción.

El capítulo tercero está dedicado por completo a la percepción en general, y a la percepción visual en particular, estudiando un modelo propuesto por el Dr. Ignacio López dentro de su tesis, [104]. Se presentan también una serie de estudios acerca de las claves perceptivas visuales que nos afectan a la hora de interpretar el espacio y la tercera dimensión. El epígrafe completo supone un estudio del estado de la técnica en esta área, y por ese motivo no se han incluido en este capítulo.

Se ha decidido recorrer el estado de la técnica en lo referente a las capacidades de interacción y aprendizaje en los robots sociales, y la influencia en ellos de las técnicas de visión por computador.

Comenzaremos con un breve repaso a los robots socialmente interactivos y a los diferentes enfoques en sus aplicaciones y en su desarrollo, dedicando una segunda parte a los robots con sistemas de visión implementados. En tercer y último lugar, se realiza un recorrido por la metodología y cuestiones importantes en el campo de la percepción visual, algoritmos de procesamiento y técnicas de detección.

2.1 Robots Socialmente Interactivos

En los *robots socialmente interactivos*, la interacción social juega el rol más importante de su funcionamiento.

Un robot controlado de forma remota no puede ser considerado social, dado que no es capaz de tomar decisiones por sí mismo. Es la mera extensión de un humano. Esto no quiere decir que, para ser considerado social, tenga que ser completamente autónomo. Una autonomía parcial también puede ser aceptada.

Por otra parte, esta autonomía viene de la capacidad obligada de tener unas habilidades, tales como recoger información de su entorno, trabajar durante periodos de tiempo aceptables sin necesidad de intervención humana, moverse en parte o completamente a través de su dominio de actuación sin necesidad de asistencia, y evitar situaciones de peligro, tanto para las personas como para él mismo, a no ser que esté programado para dichos fines.

La robótica social tiene sus comienzos en las décadas de los años cuarenta y cincuenta del siglo pasado de la mano de William Grey Walter (Fig. 1), aunque los primeros desarrollos se realizaron durante los primeros años de la década de los noventa por investigadores del área de la Inteligencia Artificial: Kerstin Dautenhahn, Maja Mataric, Cynthia Breazeal, Aude Billard, Yiannis Demiris, y Brian Duffy entre otros.

Las características que definen un robot social son:

(1) Expresan y/o perciben emociones.

(2) Tienen una comunicación con alto nivel de diálogo.

(3) Sostienen un aprendizaje y reconocimiento de modelos de otros agentes.

(4) Establecen y mantienen relaciones sociales.

(5) Usan acciones de entrada/salida naturales (gestos, pestañeo, etc.)

(6) Exhiben una personalidad y carácter particulares y distintivos.

(7) Pueden aprender y desarrollar competencias sociales.

Son muchos los usos a los que se pueden destinar estos robots, dando lugar a una diversidad que abarca desde plataformas de desarrollo, hasta juguetes, herramientas didácticas o sistemas de asistencia médica entre otros. Está demostrado que los humanos preferimos interactuar con las máquinas de la misma forma en que lo haríamos con otras personas. De ahí el hecho de implementar características en ellos que den lugar a una interacción todo lo natural que sea posible, T. Fong [9]. Se hace pues imprescindible un cierto grado de adaptabilidad y flexibilidad para dirigir cada operación a un rango del entorno suficientemente amplio: C. Breazeal [10], I. Nourbakhsh [11], J. Pineau [12] y B. Sacallellati [13].

Algunos usan modelos extendidos de interacción humanos y otros, sin embargo, muestran sus capacidades sociales sólo como reacción a determinadas conductas, dependiendo a su vez de los mismos humanos para la atribución de estos estados y emociones así como para su implementación, K. Dautenhahan [14][15], B. Duffy [16] y P. Persson [17].

Es evidente que van a tener limitaciones perceptuales, cognitivas y de comportamiento con respecto a nosotros. Sin embargo, pueden llegar a ser altamente sofisticados en determinadas áreas de socialización.

En cuanto a la metodología de diseño se pueden clasificar desde dos perspectivas diferentes:

2.1.1 Robots biológicamente inspirados

Intentan simular internamente la inteligencia de criaturas vivas. El diseño está inspirado en teorías del campo de las ciencias sociales y naturales, incluyendo la antropología, las ciencias cognitivas, la psicología, etología, sociología, estructura de interacción y teoría de la mente. Un buen ejemplo es *Kismet*, un robot diseñado con aspectos preceptúales y de comportamiento (Fig. 4).

De forma general, la *etología* estudia los animales en su entorno natural, M. Landsdale [19], la *estructura de interacción* analiza características de interacción tales como instrucciones y cooperación, y la *teoría de la mente* estudia la habilidad humana de atribuir correctamente creencias, objetivos, percepción, sentimientos y deseo hacia ellos mismos o hacia otros, A. Whiten [20].

Hablando de forma genérica, estas teorías ayudan al diseño de las capacidades cognitivas, motoras, perceptuales, emocionales y de comportamiento del robot.

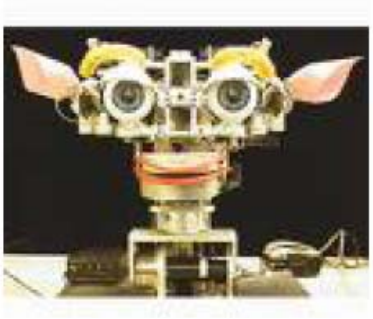


Fig. 4 KISMET (Synthetic nervous system)

Los argumentos que subyacen a esta inspiración biológica son, primeramente, que sus desarrolladores consideran a la naturaleza el mejor modelo de vida existente. Es decir, en lo concerniente al entendimiento con los humanos se convierte en un entorno propio, dado que podemos tener una interacción dentro de los mismos límites.

La segunda es que nos permite examinar directamente aquellas teorías en las que se apoya. COG es una plataforma humanoide desarrollada dentro del MIT para la exploración de teorías de modelos de comportamiento y aprendizaje inteligente. Los detalles de este proyecto se pueden ver en la página Web enlazada desde [21].

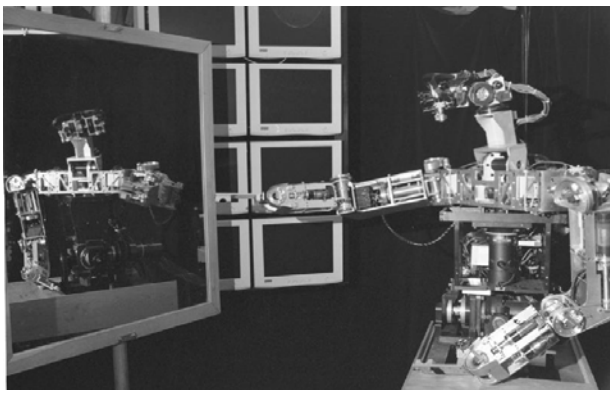


Fig. 5 COG (humanoide del MIT) reaccionando ante el estímulo visual que supone verse a sí mismo en un espejo. Tiene 22 grados de libertad para aproximar su movimiento al del cuerpo humano, y sensores visuales, vestibulares, auditivos y táctiles.

Otro ejemplo lo constituye LAZLO, una plataforma de desarrollo visual construida a partir de COG. La arquitectura de la cabeza y del cuello es la misma con una serie de puntos de movimiento adicionales y una forma de cara distinta, ya que se añadieron cualidades estéticas para permitir una interacción más natural.

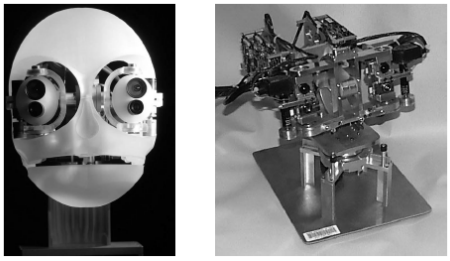


Fig. 6 Segunda plataforma de desarrollo realizado a partir de la primera, COG, para conseguir una apariencia más antropomórfica.

O el desarrollo de la cara que se está realizando para la plataforma URBANO dentro de nuestro departamento dirigido por D. Rodríguez-Losada, que tiene el propósito de transmitir emociones durante su discurso y gesticular para hacerlo más cercano al público.

Fig. 7 Desarrollo de cara con expresiones faciales, para transmitir emociones durante su discurso.

2.1.2 Robots de diseño funcional

El objetivo es el diseño de robots que en apariencia parezcan sociales e inteligentes, incluso si su desarrollo interno no tiene base en la ciencia o en la naturaleza. Esta

teoría asume que no tenemos por qué conocer el funcionamiento de una habilidad de comportamiento -deseo, sentimiento, etc.- para poder llegar a construirlo. Supone que es suficiente con conocer el mecanismo por el que se deduce ese comportamiento, P. Person [22].

Como en el caso anterior, quienes se decantan por este método, están guiados por una serie de motivaciones que podemos resumir en los siguientes puntos.

(1) El robot puede necesitar sólo una sociabilidad parcial, cuando el tiempo de interacción o la cualidad de la misma están limitados, o estar acotado por el propio entorno. No hay necesidad de implementar una habilidad en toda su acción.

(2) En ocasiones puede ser suficiente una grabación o un diálogo programado para interactuar con las personas.

(3) Los diseños artificiales pueden dar lugar a interacciones bastante convincentes. Muchos video juegos o juguetes electrónicos captan nuestra atención incluso cuando no tienen ningún equivalente en el mundo real, como es el caso del *Tamagotchi*, una mascota virtual creada en 1996 por Aki Maita y comercializada por *Bandai*.

Fig. 8 Tamagotchi: Mascota virtual creada en 1996 por Aki Maita

2.2 Características comunes de diseño

Todos los sistemas robot, ya sean sociales o no, deben resolver una serie de problemas comunes de diseño. Esto incluye

(1) Cognición: Planificación y decisión.

(2) Percepción: Navegación y sensado del entorno.

(3) Acción: Movilidad y manipulación.

(4) Interacción humano-robot: Interfaz de usuario, display.

(5) Arquitectura: Control y sistemas electromecánicos.

Los robots sociales además, deben dirigir estas cuestiones a la interacción social C. Breazeal [23] y K. Dautenhahn [24], y tener en cuenta una serie de necesidades añadidas:

(1) La percepción debe estar orientada a humanos.

Capacidad de percibir e interpretar actividades y comportamientos humanos. Incluye detectar y reconocer gestos, observar y clasificar actividades, discernir entre indicaciones y medir la realimentación con humanos.

(2) Debe existir una interacción natural humano-robot.

Humano y robot deberían de comunicarse como iguales que se conocen, como lo harían por ejemplo dos músicos tocando un dueto, T. Sheridan [25]. Para ello, el robot debe manifestar un comportamiento creíble: debe establecer una expectación social apropiada, regular la interacción social (usando diálogo y acción) y seguir normas y convencionalismos sociales con cierta naturalidad.

(3) Las indicaciones sociales deben ser inteligibles.

El robot debe enviar señales al humano con objeto de: (1) proveer un punto de realimentación acerca de su estado interno al usuario; (2) permitir al humano interactuar de manera fácil y transparente. Los canales para la expresión de las emociones, incluyen la expresión facial, cuerpo y voz.

(4) El desarrollo de sus capacidades debe darse en Tiempo Real.

Como es lógico, su rango de operación temporal debe estar dentro de los rangos de actuación humanos. Así que el robot deberá exhibir simultáneamente comportamiento, atención e intencionalidad.

2.3 Morfología e interacción con los humanos

La base estructural para una percepción mutua entre el robot y su entorno, está fundada en la relación recíproca entre dichos sistemas. En la medida que el robot perturbe su dominio de acción y éste entorno modificado altere al sistema robot, así será la realización de su estructura.

Los robots sociales no siempre necesitan un cuerpo físico tal y como lo entendemos las personas. Por ejemplo, los agentes de conversación pueden ser la realización del sistema en la misma extensión que un robot con capacidad limitada, T. Sheridan [26]. Es evidente que unos robots tienen una realización mucho más compleja que otros, véase si no la diferencia entre *Aibo* (Sony) y *Kephera* (K-Team).

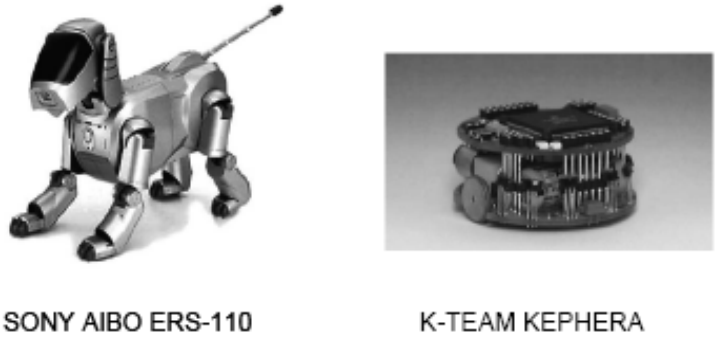


Fig. 9 Aibo tiene aproximadamente 20 actuadores (repartidos por la cabeza, boca, orejas, rabo y patas) y una variedad de sensores (táctiles, auditivos, de visión y propioceptivos). Como contraste, Kephera tiene dos actuadores (control independiente de ruedas) y un array de infrarrojos de proximidad.

La forma y estructura del robot es importante también en la medida que va a ayudar a establecer expectativas sociales ya que la apariencia física predispone la interacción. Un robot con forma de perro, va a ser tratado de forma diferente, al menos a priori, que uno con apariencia de humano.

En general la morfología de un robot puede ocasionar profundos efectos en su accesibilidad y expresividad. Por ejemplo, *Kismet* (Fig. 4) es un robot altamente expresivo. Sin embargo sus limitaciones aparecen en el momento que se requiere una manipulación o un desplazamiento.

Fig. 10 CERO (KTH), pequeño mecanismo de robot representativo.

Sheeff et al. [27] evalúa cómo las técnicas de animación tradicionales pueden ser usadas en el diseño de un robot social. Schulte et al. [28] describe cómo una caricatura de una cara humana puede llegar a ser el punto de atención y, de forma similar, Severinson-Eklind et al. [29] describe el uso de un pequeño mecanismo, CERO, simplemente como robot representativo (Fig. 10).

Algunos desarrolladores piensan que las características físicas que muestran los robots sociales deben ir acorde a sus objetivos operacionales. Este tipo de realización aparece a menudo en los robots dedicados al cuidado de las personas. Por ejemplo, en asistencia o traslado de pacientes. Así, características como un asiento o asideros de agarre, son fundamentales en su diseño para su posterior funcionalidad.

El diseño de los juguetes robot también tiende a reflejar sus requerimientos funcionales. Los juguetes deben minimizar sus costes de producción, resultar atractivos a los niños y ser capaces de orientar su comportamiento a toda la variedad de situaciones que se puedan dar durante el juego, F. Michaud [30].

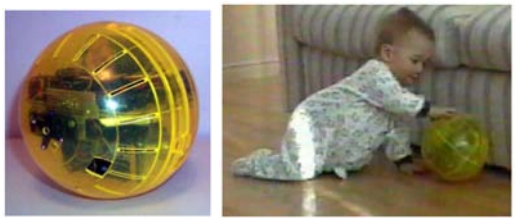


Fig. 11 Roball [30], prototipo de robot de juguete diseñado en 1998 en la Universidad de Sherbrook (Québec, Canadá) que interacciona con bebés que aún no tienen capacidad de hablar mediante movimientos, velocidades y música. Su diseño esférico le permite movimientos y protección contra choques que con otra forma no serían posibles.

Es evidente que los robots sociales comienzan a jugar un papel importante en nuestro mundo, trabajando por, para y en cooperación con los humanos. En ellos es

importante un grado de empatía con los humanos, aprovechando la predisposición que tiene nuestro cerebro, como estudió Sherry Turkle, a ser "*engañado*" por una simulación.

Resulta pues interesante el estudio y la implementación de emociones, percepción, diálogo y, en general, capacidades cognitivas que emulen con el mayor grado de parecido posible el comportamiento humano.

2.4 El valle inexplicable de Mori

Según el principio de la robótica descrito por M. Mori, el *Valle inexplicable* es una reacción emocional de rechazo, por parte de los humanos, contra aquellas entidades no humanas que guardan un parecido tanto físico como de conducta al del hombre.

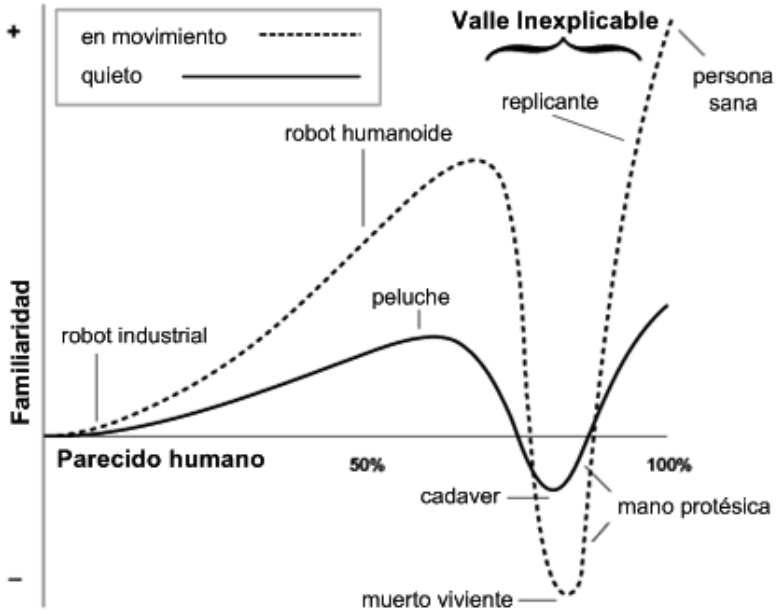


Fig. 12 Valle inexplicable de More

Una explicación a este fenómeno puede ser que, en el caso que la entidad se parezca bastante a un humano, sus características "no humanas" resaltarán más, creando así una sensación de lejanía o extrañeza, y provocando un sentimiento de rechazo. Otra

indica que este comportamiento se debe a que algunas acciones y ciertas características de los robots, se asemejen a las de los enfermos y moribundos, pero que al no tener causa concreta del motivo de este comportamiento, pueda crear en nuestra mente la sensación de riesgo contra nuestra propia integridad, así como poner en una paradoja nuestra lógica inconsciente.

Un muy buen ejemplo de este fenómeno fuera del campo de la robótica es el de la bestia de Frankenstein. Su comportamiento, rasgos y origen eran claramente humanos pero hacían destacar los "no humanos". Las cicatrices y su origen necrótico alejan a la criatura de la emoción humana de afinidad.

2.5 Visión en los robots sociales

Para interaccionar de un modo significativo con personas, los robots sociales deberían ser capaces de ver el mundo tal y como lo hacemos nosotros, es decir, percibiendo las señales e interpretándolas en el mismo modo que los humanos. Esto significa que, además de la percepción requerida para las funciones convencionales de localización, navegación y evasión de obstáculos, deben tener habilidades perceptuales parecidas a las nuestras.

Entrando en el campo de la visión, hablaremos de una percepción orientada a los humanos y dentro de los mismos límites. Inicialmente deben ser capaces de: (1) seguir características tales como cara, cuerpo o manos, (2) interpretar diálogos ayudándose de la imagen, discerniendo una palabra de afecto de una orden por ejemplo y (3) reconocer personas, expresiones faciales, gestos, órdenes, etc.

Sin embargo, una percepción similar a la de los humanos requiere algo más que un dispositivo sensible que emule el sentido de la vista en su labor de captura de señal. Es decir, algo más que una cámara que aporte imágenes.

Es muy importante que humanos y robots sean capaces de encontrar el mismo tipo de estímulo y el mismo contexto, C. Breazeal [31]. Un robot puede llevar implementado un sistema de mímica asociado a su diálogo como método de expresión, aumentando así la empatía e interacción con el usuario. O llevar implementado un procedimiento gestual para comunicar un sentimiento o estado de ánimo a su interlocutor: El sistema motor del ojo humano puede expresar

determinados comportamientos con los movimientos rápidos del ojo, el pestañeo lento o la caída de párpados, C. Breazeal [32].

2.5.1 Aplicaciones generales

Entre las soluciones más utilizadas en robots sociales nos encontramos los sistemas de seguimiento de personas y objetos, denominado también con el anglicismo *Tracking*. El desafío está en encontrar métodos eficientes para localizar y seguir personas en presencia de ruido y variabilidad en la iluminación, teniendo en cuenta posibles oclusiones, movimiento de cámaras o variación del fondo en la escena.

En [33], D. Gavrilla realiza un recorrido por los sistemas de rastreo con visión y en [34], R. Tanawongsuwan presenta un método de seguimiento de personas en entornos dinámicos con un dispositivo consistente en una cámara montada sobre un robot móvil (Fig. 13).

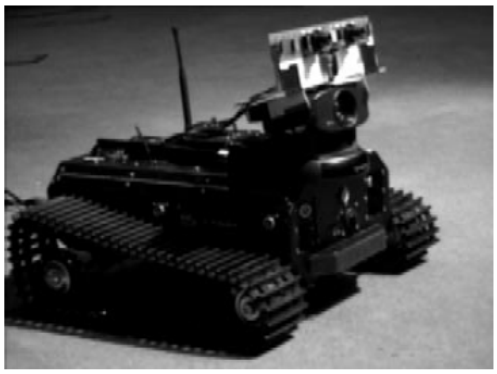


Fig. 13 Robot propuesto por R. Tanawongsuwan [34] con una cámara PTZ y un sistema estéreo SRI

El reconocimiento de gestos es otra de las aplicaciones habituales de la visión por computador en la robótica. Las personas cuando conversamos, usamos gestos para clarificar nuestro discurso y expresar de una forma más concisa la información geométrica, como la localización o la dirección. En muchas ocasiones, el locutor usa sus manos o el movimiento de éstas para, ya sea con velocidad, o con un rango de movimientos adecuado, indicar urgencia o clarificar ambigüedades al hablar. Por ejemplo: *"…he aparcado el coche más allá"*.

Aunque hay muchas formas de reconocer gestos, el *Reconocimiento Basado en Visión* tiene muchas ventajas sobre cualquier otro. La visión no requiere que el usuario acarree elementos adicionales para que el sistema opere correctamente, tales como acelerómetros o unidades inerciales.

Entre los trabajos de mayor relevancia en esta área destacamos el de V. Pavlovic [35], un sistema capaz de dar pequeñas ordenes a un robot mediante la detección de movimientos del brazo (ver Fig. 14).

Fig. 14 Tracking de personas mediante una cámara montada sobre un robot. Las personas realizan un gesto a modo de orden para que el robot traiga una pelota. De izquierda a derecha: a) Imagen original, b) Segmentación con el color de piel, c) Imagen binaria de energía en función del movimiento, d) MHI (Imagen Histórico de Movimiento) Muestra la dirección del movimiento, e) imagen estéreo

En [36], Y. Wu realiza un pequeño estudio del estado del arte en esta área. Otras aportaciones interesantes las realiza D. Kortenkamp [37], proponiendo un sistema de reconocimiento e interpretación de gestos en un robot, S. Waldherr [39] con una interfaz de reconocimiento de gestos para interacción hombre-máquina y G. Xu [40] con una propuesta para guiar un robot móvil mediante visión 2D.

Dentro del campo de la percepción facial nos encontramos varias áreas de estudio. Se tratan más detalladamente en el siguiente apartado.

- *Detección y Reconocimiento de caras*

  Hay desarrollados una amplia variedad de sistemas de detección y reconocimiento de caras que funcionan con alto grado de satisfacción, como los presentados por R. Chellappa [41] y T. Fromherz [42]. Existen detectores y sistemas de seguimiento facial en tiempo real cuyo funcionamiento ofrece unos resultados muy notables y un rango de aplicación muy amplio, como el presentado por K. Toyama [43]: un sistema que mueve un cursor sin necesidad de usar el ratón o las manos. Estos sistemas resultan muy válidos en entornos con personal discapacitado.

- *Expresión Facial*

  Desde Darwin [44] la expresión facial ha sido considerada como una vía de expresión de emociones. En la actualidad los gestos de la cara se están utilizando para dar a conocer intencionalidad mediante señales. C. Lisetti [45] presenta una revisión bastante completa del análisis de las expresiones faciales, incluyendo una reflexión en lo que respecta a la ética y la psicología. En este mismo trabajo se exponen los tres desarrollos básicos en lo referente al reconocimiento de gestos faciales: (1) *Técnicas de detección de movimiento en imágenes* que identifican los músculos faciales en acción analizando la variación durante una determinada secuencia. (2) *Modelos anatómicos de seguimiento de características* de la cara tales como la distancia entre ojos, triángulo ojos-nariz, proporcionalidad entre características, etc. (3) *Análisis de Componentes Principales* (PCA) para reducir el espacio vectorial de representación a otro construido a partir de las componentes principales: EigenFaces u hologramas. (4) *Seguimiento de la mirada* ya que constituye un buen indicador del punto de atención de persona o de la pérdida de la misma.

  Aunque son muchos los sistemas de visión que realizan el seguimiento a partir del movimiento de la cabeza, pocos han intentado conseguir el rastreo del ojo usando únicamente Visión por Computador de forma pasiva. Por otra parte, en la mayor parte de los casos carecen de la precisión deseable, K. Toyama [43] y R. Stiefelhagen [46].

2.5.2 Robots con aprendizaje visual

La construcción de robots que resuelvan el cubo de Rubik utilizando únicamente un sensor de detección de colores es una tarea inicialmente compleja. *Tilted Twister* (Fig. 15), es un robot casero relativamente sencillo construido por Hans Anderson con Lego Mindstorms, el juego de robótica de la compañía Lego. Este robot es capaz de completar en pocos movimientos un reto que a muchos humanos les resulta totalmente imposible: resolver el cubo de Rubik. Actualmente lleva un sensor HiTechnic capaz de detectar los colores originales, eliminando así los problemas iniciales que se produjeron en éste sentido.

Fig. 15 Tilted Twister, robot desarrollado por Hans Anderson y que resuelve en pocos movimientos el problema del cubo de Rubik.

Sin embargo, los científicos de la Universidad de Massachussets han querido llegar más allá, creando el *UMass Mobile Manipulator*, UMan de nombre de pila, un robot que aprende a manejar objetos de la misma forma que lo haría un niño pequeño. Consta de un brazo articulado de un metro, acabado en una mano de tres dedos e instalado en una plataforma con ruedas. Una cámara Web le permite ver los objetos colocados ante él sobre una mesa. Los empuja uno a uno, para estudiar y aprender cómo se mueven. Si localiza una parte rígida, entenderá que se trata de una articulación, y averiguará por sí mismo cuál es la manera más adecuada de manipularlo. No deja de manipularlo hasta que considera que ha comprendido su funcionamiento.

Fig. 16 UMan, un robot que aprende a identificar objetos

HOAP-3 es el primer robot capaz de aprender y, hasta cierto punto, comprender sus acciones, como aseguran sus desarrolladores. No memoriza el movimiento cuando se le muestra, como la mayoría de las máquinas. Aprende poco a poco a generalizar la acción y a adaptarla para usarla oportunamente. De esta manera, es capaz de identificar el mejor momento para utilizar cada postura o gesto.

Además habla mientras aprende, explicando a los investigadores lo que está haciendo. Estos, usan un control remoto para comunicarse con el robot. De este modo, se le transmite el momento en que se le está explicando algo nuevo.

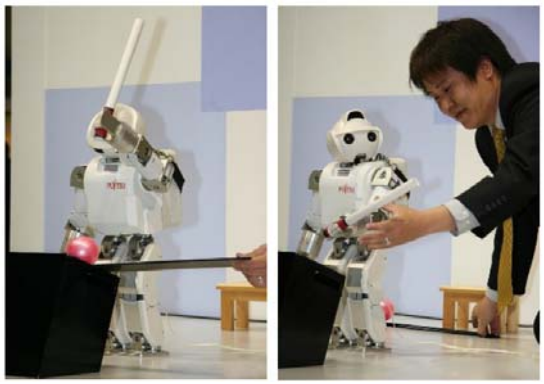


Fig. 17 En la fotografía de la izquierda, HOAP-3 está haciendo lo que ha aprendido a hacer. En la derecha, aprendiendo a golpear una bola.

2.6 Procesamiento de la Imagen

El área de estudio de la Visión por Computador y el Procesamiento de Imagen está evolucionando continuamente. Durante las décadas de los 80 y los 90 del siglo pasado, aumentó el interés por la morfología de la imagen, las redes neuronales, el procesado, reconocimiento y compresión de imagen y el análisis de sistemas basados en conocimiento. Estos esfuerzos construyeron un primer núcleo de modernización y descubrimiento de nuevas técnicas que derivaron en un auge comercial de los sistemas de visión y el procesamiento digital.

El interés por el procesamiento de la imagen surgió de dos motivaciones: mejorar la información de las imágenes para la interpretación humana y el procesado de datos para sistemas de percepción artificial.

Una de las primeras aplicaciones se implementó para mejorar la calidad de las fotografías que eran enviadas por cable submarino desde Londres a un periódico neoyorquino. Durante la década de los años 20 del siglo pasado se introdujo el

sistema Bartlane[1] de transmisión de fotografías por cable. El objetivo era reducir el tiempo requerido para transportar una fotografía a través del Atlántico, que en aquel momento era de algo más de una semana. Fue un éxito que consiguió la transmisión en menos de tres horas.

El ingenio consistía en el uso de una máquina de escritura de texto telegráfico, para transferir las imágenes en valores, de modo que pudieran ser transmitidos con los medios de comunicación del lenguaje y escritura. La máquina era una extensión del sistema Korn y su base residía en la llamada *Cinta Baudot* (ver Fig. 18).

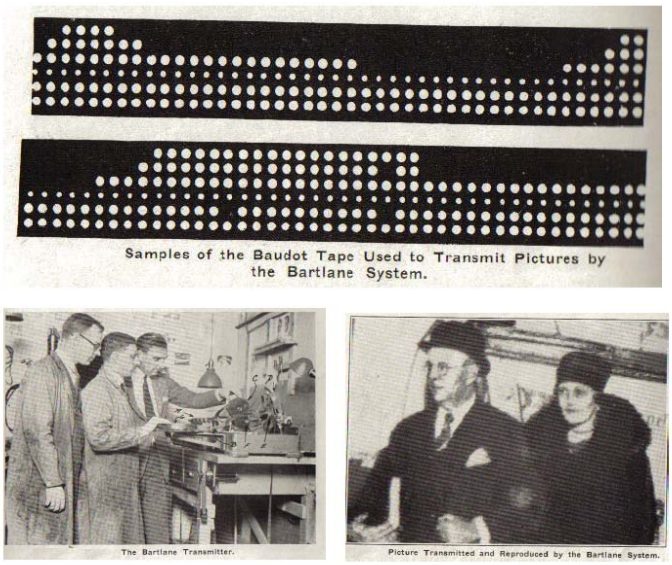


Fig. 18 En la imagen superior, muestra de la cinta Baudot usada para transmitir fotografías con el sistema Bartlane. En la parte inferior a la izquierda el transmisor Bartlane y, a la derecha, una fotografía transmitida con el ingenio.

Esta tipo de cinta era una de las más usadas en comunicaciones telegráficas automáticas a través del Atlántico. Estaba constituida por un agujero central que se establece como "guía", tres aberturas más en uno de sus lados y dos más en el lado opuesto. La combinación de ciertos huecos transmite determinados impulsos asociados, que actúan de señales de escritura en el lugar de recepción.

Los primeros problemas con que se encontraron durante estos primeros años estuvieron relacionados con las técnicas de impresión y la distribución de los niveles de brillo. Siempre buscando mejorar la calidad de la imagen. Durante la

[1] El nombre de Bartlane se debe a los nombres de sus dos inventores, Mr. Bartholomew y MacFarlane, ambos del Daily Mirror de Londres, Inglaterra.

década de los años 20 del siglo XX, el *sistema Bartlane* fue evolucionando, y la transmisión pasó de los 5 niveles de intensidad iniciales a los 15 en 1929. Durante los siguientes 35 años se siguió estudiando en esta línea, a la vez que, paralelamente, comenzaban a aparecer los primeros computadores y con ellos la potencia de cálculo y el espacio para almacenar datos.

Los primeros trabajos en visión por computador se realizaron en 1964, en el *Jet Propulsión Laboratory* (Pasadena, California) al procesar en un computador las imágenes de la luna procedentes de la cámara de la nave Ranger 7, para eliminar las distorsiones con que llegaba la transmisión. Desde entonces hasta nuestros días el campo de la visión por computador ha evolucionado incesantemente. Ya no se usa únicamente para misiones espaciales, sino que está dedicado a la resolución de multitud de problemas cotidianos.

En medicina por ejemplo, los niveles de gris se intensifican o se convierten a espacios de color para facilitar la interpretación de las imágenes de las máquinas de Rayos-X. Los geógrafos utilizan técnicas similares para analizar patrones de contaminación e imágenes de satélite. En arqueología se han podido recuperar imágenes de objetos que desaparecieron una vez que fueron fotografiados. En áreas relacionadas con la física se utilizan técnicas de visión rutinariamente en experimentos con plasmas o microscopios electrónicos. Y lo mismo ocurre en el campo de la astronomía, la biología, medicina nuclear, defensa, industria y robótica entre otros.

2.6.1 Técnicas de reconocimiento facial

La investigación en esta área está motivada no sólo por el reto que supone este problema, sino por las numerosas aplicaciones prácticas en las que puede llegar a ser aplicado. El reconocimiento facial se vuelve cada vez más importante ante los continuos avances de la tecnología digital, la comunicación por red, la telefonía móvil y el incremento de demanda de seguridad. Es la técnica biométrica primaria y con mayor interés en su demanda por las ventajas que supone frente al resto: es natural, no intrusiva y fácil de utilizar.

Aunque su estudio comenzó en la década de los 60 del siglo pasado, continúa siendo un problema sin una solución definitiva. Durante los últimos años se han conseguido avances significativos en el modelado y las técnicas de análisis, y se han

logrado sistemas para detectar y seguir caras con resultados relativamente satisfactorios.

Los sistemas de reconocimiento facial tuvieron un avance importante desde el primer sistema propuesto y desarrollado por T. Kanade [38]. Los recientes resultados en el análisis de características, reconocimiento de patrones y técnicas de aprendizaje están ofreciendo las capacidades necesarias para atender a todas estas nuevas aplicaciones. Sin embargo es especialmente difícil intentar conseguir soluciones genéricas, especialmente en entornos y tareas sin condiciones fijas o impuestas, sin limitaciones de iluminación, puntos de vista, oclusión, expresión, accesorios y una gran variedad de elementos a considerar.

El problema de una detección robusta en entornos no controlados puede ser formulado con el siguiente ejemplo: Dada una secuencia de fotografías de gente utilizando un determinado servicio, mantener el listado de todos los conectados y reconocer si pertenecen al nivel de usuarios. Identificar a quienes están en la lista e insertar a quienes no han sido reconocidos. Los dos problemas fundamentales que vienen asociados son:

(1) Descartar una cara de un listado: Es una tarea más complicada que reconocerla, puesto que el segundo caso es conocida y pertenece ya a una base de datos.

(2) Reconocer caras con oclusiones parciales: Es más difícil que reconocerlas frontalmente, ya que en este último caso se pueden apreciar todas las características, S. Gutta y H. Wechsler [108].

El proceso de identificación pasa generalmente por uno o varios de los siguientes pasos de procesamiento previo de la imagen:

(1) Traslación, rotación y escalado de la imagen para fijar un número de filas y columnas de píxeles, de modo que el centro de los ojos esté localizado.

(2) Aplicar una máscara para eliminar el fondo de la escena y el pelo.

(3) Implementar una ecualización a partir de un histograma.

(4) Normalización de los datos faciales para tener una media cero y una desviación estándar unitaria.

En este apartado realizamos un recorrido por los métodos de reconocimiento facial para aplicaciones en que el entorno no está controlado, puesto que son las que van a afectar a nuestro sistema. B. Gong et al.[109] presentan una notable introducción al reconocimiento facial, y Zhao et al.[110] enfocan este mismo problema para aplicaciones de vídeo. Khong et al.[111] estudian los métodos de reconocimiento con infrarrojos.

Los métodos de reconocimiento facial se pueden clasificar en tres grupos a los que prestaremos una mayor atención en este estado del arte:

(1) *Métodos basados en características*, donde las singularidades como ojos, nariz y boca son extraídas de su posición inicial e introducidas en clasificadores estructurales.

(2) *Métodos basados en apariencia*, que utilizan la región facial completa como una entrada al sistema de reconocimiento.

(3) *Métodos combinados*, que en principio son los que obtienen mejores resultados.

Desde el momento en que las condiciones para el reconocimiento facial dependen de la posición, de las expresiones faciales y del entorno, no habrá, en general, un método mejor que otro.

2.6.1.1 Métodos basados en características

Matching estructural

Los primeros estudios de reconocimiento facial, como por ejemplo el de I. Cox et al.[112] detectaban un conjunto de singularidades de la cara (ojos, ojos marrones, nariz y boca). Las propiedades y relaciones tales como el área, las distancias y los ángulos entre características eran utilizadas por los descriptores de reconocimiento facial. Sin embargo el resultado en estos métodos está muy ligado a la precisión del algoritmo de localización de las singularidades.

Otro sistema consiste en tomar como características un mapa de contornos, LEM (*Line Edge Map*), basado en una combinación de técnicas de matching geométrico y matching de patrones, Y. Gao et al.[113].

Las caras se codifican en mapas binarios utilizando el detector de bordes Sobel, y la similitud de las caras se mide con un esquema de características faciales.

Matching con grafos

Denominado también *Elastic bunch graph matching* (EBGM), Wiskott et al.[114], realiza primero una rejilla sobre la imagen de la cara, y los nodos se ajustan a un conjunto de puntos definidos. Después se evalúa la convolución de un conjunto de *wavelets 2D de Gabor* sobre cada nodo y la salida representa un vector de características de un punto particular de la cara. Por último, se realiza un algoritmo de matching con grafos.

Es un método que da muy buenos resultados, pero requiere que la cara esté muy próxima a la cámara y de una resolución mínima de [128x128].

Modelos de apariencia de Markov (HHMs)

Los primeros trabajos fueron propuestos por Samaria et al.[115], montando una estructura vertical de una dimensión compuesta por una especie de superestados que contenían cadenas de Markov horizontales.

Nefian et al.[116] propuso algo similar pero esta vez utilizando la *Transformada discreta del coseno* para observar los vectores de características. Kohir et al. [117] propone el escaneo de la imagen en zig-zag para definir secuencias de observación.

En los últimos años, Othman et al.[118] han propuesto una estructura de dos dimensiones y de baja complejidad. Esta estructura resulta de asumir como cierta la independencia condicional entre los entornos de vecindad de píxeles de los bloques observados dentro de la imagen.

2.6.1.2 Modelos basados en apariencia

Análisis de Componentes Principales (PCA)

Este método se describe en toda su complejidad en capítulos posteriores. Dada una serie de imágenes de tamaño definido, son convertidas a vectores. A partir de estos vectores se construye un espacio vectorial de características faciales a partir del análisis de sus componentes principales. Es un método de reducción dimensional mediante la traslación de los vectores del espacio real, a un subespacio de características de menor dimensión. Este nuevo espacio vectorial está definido por los vectores ortogonales propios de la matriz de covarianza del conjunto de vectores iniciales. Se les ha dado el nombre de *eigenfaces* por su aplicación final, y se corresponden con los vectores propios con mayores valores propios asociados, que precisamente son los que capturan mayor número de variaciones en el conjunto de vectores de entrenamiento.

Las ventajas de éste método son:

(1) Reducen la sensibilidad al ruido

(2) Reducen los requerimientos de memoria del sistema

(3) El espacio vectorial de menor dimensión es mucho más eficiente.

Y como desventajas:

(1) Depende de una adecuada posición de la imagen

(2) Iluminación constante.

Lo cierto es que si una de estas condiciones no se cumple, puede dar lugar a errores en el resultado puesto que los primeros autovectores codifican las variaciones de localización e iluminación.

ICA (Independent component análisis), Bartlett [119], genera características de localización espacial dando lugar a bases vectoriales estadísticamente independientes. La comparación con PCA es complicada porque tienen que tenerse en cuenta las diferencias tanto en tareas como en arquitectura de ambos algoritmos.

Fig. 19 Ejemplo de una distribución bidimensional y los correspondientes componentes principales y componentes independientes.

Para una mejor identificación, LDA (*Linear Discriminant Analisis*) puede ser aplicado para detectar variaciones en la imagen debidas a factores externos tales como la iluminación o el cambio en la expresión facial. LDA, Swets et al.[120] y Bellhumeour et al.[121], realiza el análisis principal, *eigenanalisis*, de un producto de dos matrices donde una debe estar invertida. Los autovectores que se obtienen se usan como bases de representación llamadas *fisherfaces*. A diferencia con PCA e ICA, es una técnica de aprendizaje supervisado.

PCA e ICA dan lugar a vectores no nulos para al menos todas las dimensiones, implicando con ello que cualquier cambio en un píxel de entrada, alterará cada dimensión de su proyección sobre el subespacio.

Modelos deformables tridimensionales

Brevemente hacemos una referencia a los modelos deformables propuestos por Blanz y Vetter [122][123]. Se trata de modelos faciales 3D que son aprendidos a partir de un conjunto de 200 ejemplares obtenidos con un escáner láser. Cada ejemplar tiene una media de alrededor de 70.000 vértices con los que se construyen conjuntos de correspondencias a todas las otras caras analizadas.

La forma se representa con un vector S que contiene las coordenadas de cada uno de los vértices y la textura con un vector T que contiene sus valores RGB. Las nuevas formas se pueden construir a partir de envolventes convexas de estos vectores.

Este método requiere muchos recursos computacionales, de memoria, tiempo de procesado, etc. motivo por el que son poco usados para el reconocimiento facial.

2.6.1.3 Métodos combinados

Tratan de mejorar el reconocimiento mediante la combinación de métodos basados en características y los basados en apariencia.

Modelos faciales estadísticos

Denominados generalmente *Active Statistical Face Models*, fueron introducidos inicialmente por Cootes et al.[124]. Son modelos de forma estadísticos que combinan:

(3) Un modelo de distribución global de puntos (PDM) modelando la forma del objeto y sus variaciones usando un conjunto de marcas.

(4) Un conjunto modelos de niveles de gris locales (LGL), que capturan las variaciones de los niveles de gris observados en cada marca.

Un modelo PDM se puede usar para representar la forma de una cara como un conjunto de marcas etiquetadas, recogidas en un vector x. Los modelos de variación de la forma de la cara son definidos a partir de los PCA a partir de la media de todas las muestras faciales que se tengan para entrenar. Un modelo LGL es entrenado para cada marca y junto con PDM se construye un *Active Shape Model* (ASM).

Mediante un proceso de búsqueda iterativo se compara el ASM de la nueva imagen con los que ya se tienen, resultado del entrenamiento previo.

Otros modelos combinados

Chen et al. [125] usan HMMs para modelar clases de imágenes faciales y un conjunto de operadores de Fisher se evalúa a partir del análisis derivativo parcial de los parámetros estimados en cada HMM. Estos operadores son también combinados con los clásicos modelos de vecindad y modelos basados en apariencia para lograr vectores de características que exploten las ventajas de ambos métodos. Se aplica posteriormente un LDA para analizar estos vectores de características y conseguir el reconocimiento facial.

2.6.2 Análisis de detección de movimiento

El análisis de movimiento es otra de las cuestiones más estudiadas actualmente en visión por computador. El interés que suscita está guiado en cierta medida por el amplio espectro de prometedoras aplicaciones en áreas como la realidad virtual, seguridad y defensa, interfaces perceptuales, etc.

Este análisis de movimiento engloba la detección de personas, el rastreo y el reconocimiento y, de forma más general, la comprensión del comportamiento humano a partir de las secuencias de imágenes.

El movimiento humano ha sido investigado a lo largo de muchos años en una amplia variedad de proyectos. DARPA, *Defense Adavanced Research Projects Agency*, (una agencia del Departamento de Defensa de los Estados Unidos responsable de multitud de desarrollos con nuevas tecnologías de gran impacto en el mundo[2]) consolidó un proyecto de vídeo seguridad y monitorización (VSAM), R. T. Collins [64], cuyo propósito fue el desarrollo de un sistema de video automático que permitía a un operador monitorizar actividades en entornos complejos. El sistema de seguridad de video W^4, I. Haritaoglu [65], utilizaba un análisis combinado de características de forma y rastreo para construir modelos de apariencia humanos. De este modo se obtenía la capacidad de detectar y seguir a varias personas y, a la vez, rastrear su actividad en entornos abiertos aún en presencia de oclusiones.

En los últimos años, el análisis de movimiento ha sido una de las características con mayor presencia internacional en publicaciones de tanto renombre como IJCV (Internacional Journal of Computer Vision), CVIU (Computer Vision and Image Understanding), PAMI (IEEE Transactions on Pattern Recognition and Machine Intelligence) e IVC (Image and Vision Computing), así como en prestigiosos workshops como ICCV (Internacional Conference on Computer Vision), CVPR (IEEE International Conference on Computer Vision and Pattern Recognition), ECCV (European Conference on Computer Vision), WACV (Workshop on Applications of Computer Vision) y IWVS (IEEE International Workshop on Visual Surveillance). En todos ellos se puede encontrar una amplia bibliografía sobre la detección del movimiento así como de sus aplicaciones.

[2] Responsible del desarrollo de ARPANET, que después se convirtió en INTERNET, y de NLS, un sistema de hipertexto y precursor de la interfaz gráfica de usuario contemporánea.

2.6.2.1 Aplicaciones potenciales del análisis de movimiento

Aunque existe un amplísimo rango de aplicaciones potenciales, centramos la atención en aquellas más extendidas.

Seguridad y vigilancia

Actualmente existe una demanda muy grande en sistemas de seguridad para bancos, grandes almacenes, aparcamientos de vehículos, vigilancia policial, etc. Es habitual encontrar una cámara de seguridad en cualquier establecimiento o edificio público e incluso en la calle. Las grabaciones de seguridad son utilizadas como herramienta forense, como prueba en contiendas legales e investigaciones policiales, en sistemas de seguridad antirrobo, identificación de matrículas, personas, etc.

La idea de obtener beneficios de estos sistemas en tiempo real, es notificar por ejemplo acciones en curso: robos que se puedan estar ejecutando en el momento o presencia de personas en lugares con acceso restringido. Este es uno de los marcos justificados de los sistemas de detección de movimiento, aunque existen muchos más.

Por ejemplo, los sistemas de detección facial, J. Steffens [66] y C. Wang [67] y de peatones, J. J. Little [68] y D. Cunado [69] han estado fuertemente ligados a propósitos de control de accesos. O las aplicaciones de seguridad en espacios abiertos, control de tráfico, monitorización de peatones en espacios con alta densidad de público, etc. suponen otras de las muchas áreas de aplicación de la visión a sistemas de seguridad y vigilancia.

Interfaces de usuario avanzadas

Otra de las importantes aplicaciones de los sistemas de detección de movimiento con visión. En este caso, el movimiento se utiliza para operaciones de control y mando. De forma general, se puede decir que la comunicación entre personas está principalmente asentada en el discurso y es el motivo por el que ha supuesto un ítem ampliamente utilizado en la comunicación hombre-máquina. Sin embargo, está sujeto a importantes restricciones medioambientales como ruido y alejamientos.

La visión es un buen complemento del reconocimiento del habla y del lenguaje natural, entendiendo esto para sistemas de comunicación entre agentes con

inteligencia incluida o implementada. En el discurso, hay una serie de detalles adicionales que se analizan mejor a través de la imagen, tales como gestos, posturas, expresiones faciales, etc. Yi Li [70], J. Segen [71], M-H Yang [72] y Y. Cui [73]. De ahí que el futuro inmediato de las máquinas esté ampliamente ligado a la habilitación de estas capacidades, M. Turk [74].

Otras aplicaciones de la detección de movimiento en el dominio de la interfaz de usuario incluyen la traducción del lenguaje de signos, control dirigido por gestos y señalización en entornos con ruido tales como fábricas o aeropuertos.

Diagnóstico e identificación basados en movimiento

La segmentación de alguna parte del cuerpo humano en una imagen, el rastreo de su movimiento a lo largo de una secuencia de imágenes y la recuperación de la estructura 3D de la figura humana para el análisis y entrenamiento físicos, supone una de las aplicaciones de mayor utilidad del estudio del movimiento.

El tradicional análisis del modo de caminar de las personas ha supuesto un buen procedimiento de diagnosis y tratamiento médico, pudiendo ser usado también como una nueva singularidad biométrica para la identificación de personas, J. J. Little [68] y D. Cunado [69].

Añadido a esto, el análisis de movimiento muestra importancia relevante en otras áreas. Por ejemplo, aplicaciones típicas en realidad virtual, juegos, estudios de vídeo, animación, teleconferencia, etc. Muchas personas pueden quedar asombradas ante el realismo en el movimiento y gesto de las películas de animación y de los juegos de computador y consola. En realidad, sus movimientos están basados en el conocimiento del movimiento del cuerpo humano y en modelos elaborados a partir de este conocimiento.

2.6.2.2 Primeros trabajos realizados

Uno de los primeros desarrollos con relevancia fue probablemente el trabajo de Aggarlwal et al.[75]. Recorrió varios métodos usados en movimiento articulado y elástico previo al año 1994, para describir modelos de formas.

En [76] Alex Pentland presenta un repaso bastante completo centrado en la identificación de personas, monitorización en seguridad y vigilancia, metodología 3D e interfaces de usuario perceptuales como parte del estado del arte del análisis visual del movimiento. No se trata de un artículo dedicado plenamente a la detección del movimiento de personas con visión, pero sí es cierto que referencia varios modelos de análisis y aplicaciones de interés muy relevante.

2.6.2.3 Detección del movimiento

Casi todos los sistemas de visión para la detección del movimiento de personas, comienzan con la detección de humanos. La localización de la figura humana conlleva la búsqueda de regiones de interés en la imagen donde se descubre la figura y su segmentación para separarla del resto de la escena. Este proceso generalmente envuelve técnicas de:

(1) Segmentación

(2) Clasificación de objetos.

Segmentación del movimiento

Es un problema significativo y de difícil solución que requiere la detección de regiones correspondientes a objetos en movimiento en escenarios naturales. De este modo se suministra un foco de atención hacia los últimos procesos o últimos cambios realizados en la imagen, considerando únicamente unos pocos píxeles (aquellos sujetos a cambios) y no todos los de la secuencia. No obstante, los cambios de iluminación, sombras o movimientos repetitivos dificultan sin lugar a duda el proceso.

En la actualidad, muchos de los métodos de segmentación usan información temporal o espacial de las imágenes.

A) Sustracción de fondo

La sustracción de fondo es una técnica muy empleada para eliminar el fondo de la escena en una imagen, I. Haritaoglu [65], K. P. Karmann [77], C. R. Wren [78], C. Stauffer [79], S. J. McKenna [80], S. Arseneau [81], H. Z. Sun [82] y A. Elgammal [83]. Se trata de un método muy particular y utilizado par la segmentación del movimiento, especialmente bajo aquellas situaciones de en que se tenga un fondo de escena estático. La intención es detectar regiones en movimiento en la imagen por medio de la diferenciación entre la secuencia actual y una imagen del fondo de escena que sirve de referencia. Es un método extremadamente sensible a cualquier cambio dinámico en la escena.

Las numerosas aproximaciones que se han dado a este problema difieren en el modelo del fondo de la escena que se utiliza y al procedimiento para actualizarlo. El más simple es una media temporal de la imagen, una aproximación similar a la de estimar una escena fija. Basándose en la observación de que el valor de la mediana de la escena era más robusto que el valor de la media, Yang y Levine [84] propusieron un algoritmo para la construcción de un primer esbozo del fondo de escena tomando este valor de la mediana para un número determinado de secuencias de imagen. Tal y como ocurría con los valores umbral definidos para los histogramas, este valor se usa para crear diferencias en la imagen. Como en otros muchos algoritmos, conlleva muchas inconsistencias debidas a los cambios de iluminación.

En muchos casos los desarrollos se dirigen a la construcción de modelos adaptativos para reducir estas influencias dinámicas en la escena. Por ejemplo Karmann y Brandt [77] y Kilger [85] propusieron un modelo de escena adaptativo basado en el filtro de Kalman para adecuar y ajustar los cambios de iluminación en la imagen.

B) Métodos estadísticos

Muchos están inspirados en los métodos de extracción de fondo descritos en el apartado anterior. Utilizan las características de los píxeles o de grupos de píxeles para la construcción de modelos avanzados, pudiendo adaptar la estadística del fondo de la escena dinámicamente durante el procesamiento de la imagen. Cada píxel en la secuencia presente es clasificado como perteneciente al fondo de escena o al primer plano mediante la comparación del modelo estadístico estimado. Es una aproximación que se está convirtiendo en una metodología en auge por su robustez ante el ruido, las sombras y las condiciones de iluminación.

Un ejemplo de estos métodos estadísticos lo proponen Stauffer y Grimson [79] con un modelo de fondo de escena combinado con un rastreo en tiempo real. Modelan

cada píxel como un conjunto de Gaussianas y lo utilizan en aproximaciones en línea para actualizarlo.

Un reciente estudio de Haritaoglu et al. [65] realizó un modelo estadístico mediante la representación de cada píxel por tres valores: (1) su mínimo valor de intensidad, (2) su máximo y (3) la diferencia entre secuencias consecutivas durante la etapa de entrenamiento. Los parámetros del modelo se actualizan periódicamente.

Las singularidades que se caracterizan estadísticamente son típicamente colores y bordes. Por ejemplo, McKenna et al. [80] usa un modelo adaptativo combinando color e información de gradiente donde, cada cromaticidad de píxel se modela usando medias y varianzas. El gradiente en las direcciones X e Y se modela utilizando medias y magnitudes de varianzas. El modelo de C. R. Wren [78] se basa en una serie de manchas con unas características estadísticas de color y forma determinados.

C) Diferenciación temporal

Las aproximaciones de las propuestas realizadas en R. T. Collins [64], I. Haritaoglu [67], A. J. Lipton [86], C. Anderson [87], J. R. Bergen [88] y Y. Kameda [89] se basan en la diferencia píxel a píxel entre dos o tres secuencias consecutivas de la imagen para la extracción de regiones con movimiento. La diferenciación temporal es un método muy adaptable a los entornos dinámicos, pero generalmente presenta soluciones más pobres en la extracción de los píxeles de una característica relevante completa.

Por ejemplo, Lipton [86] usa un umbral para determinar cambios en la imagen resultado de realizar una diferenciación en valor absoluto entre las secuencias actual y anterior. Usando en *Análisis de Componentes Conectados*, extrae las secciones con movimiento en forma de particiones o regiones incluidas en la imagen.

CAPÍTULO 3

Modelo de percepción visual

Este capítulo ofrece una visión conceptual de la percepción en general primero, y aplicada a la visión después, con la intención de que pueda ser aplicada en un futuro análisis y diseño de un sistema artificial de aprendizaje. Todo lo que aquí se expone es el resultado de dos etapas de estudio:

(1) Por un lado la percepción del color, forma y espacio en los humanos desde los diferentes puntos de vista que propone el estado del arte: neuronal, biológico y psicológico.

(2) Por otro lado el estudio de un marco teórico de percepción presentado por el Dr. I. López [104] dentro de su tesis doctoral.

Por este motivo, los primeros apartados están dedicados a la percepción visual en el contexto de los sistemas biológicos. El objetivo es presentar los principios bajo los que percibimos e interpretamos una imagen, y las claves perceptivas que influyen para crear un concepto.

En una segunda parte, que constituye el último apartado de este capítulo, se presenta el marco conceptual estudiado y una aplicación sencilla a nuestro sistema de visión para analizar la percepción desde dos puntos de vista:

(1) Las partes que intervienen en la percepción y la manera en que están relacionadas.

(2) Los flujos de información del sistema.

El objetivo último es explicar la percepción visual desde un punto de vista general para intentar establecer una ontología común a sistemas artificiales y biológicos.

Este capítulo supone un estudio en base a los siguientes puntos: (1) Interpretación de una imagen, (2) Percepción visual humana, (3) Claves perceptivas de la visión y (4) Aplicación teórica de un modelo de percepción.

El primer punto constituye un pequeño análisis acerca de la imagen, su significado y el modo en que lo percibimos los humanos. Después, en el segundo, se exponen algunas características y teorías que han ido apareciendo en el estado del arte en lo referente a la percepción visual.

El tercer punto se refiere a las claves perceptivas bajo las que descubrimos el espacio tridimensional que nos rodea, pero sin recurrir a la matemática de la visión 3D. Es decir, aunque las matemáticas de la visión 3D tradicional están ampliamente estudiadas y sus resultados demostrados, lo que vamos a analizar es esta percepción del espacio con las claves perceptivas de la visión 2D, partiendo del hecho de que somos capaces de sentir esta misma sensación de espacio ante un cuadro correctamente proyectado.

Por último se analiza uno de los sistemas realizados en este proyecto con el marco de una teoría general de la percepción, la del Dr. Ignacio López [104], lo que permite extraer conclusiones interesantes sobre el sistema y sus posibles mejoras.

3.1 Vacíos cognitivos de la visión

La visión por computador se ha centrado en el desarrollo de algoritmos para tareas perceptivas muy concretas. En estos algoritmos los ingenieros, para hacerlos funcionar, incluyen mucho conocimiento de manera implícita. Conocimiento del mundo y del entorno que el sistema artificial por tanto "desconoce". Por ello, cuando se quiere ampliar el rango de aplicación de esos algoritmos, pese a que a

priori parecería que el sistema podría hacerlo, ya que el ingeniero le ha "dado" el conocimiento para ello, el resultado no es exactamente el esperado debido a que el conocimiento necesario no es accesible para el sistema.

La idea de evaluar una imagen con modelos y técnicas basados en conocimiento e inteligencia artificial, es conseguir prototipos de comportamiento semejantes a la realidad. Urbano lleva la cámara a bordo y las imágenes que envía al sistema de visión proceden de entornos reales. Actualmente, con un sistema de visión, procesamos la imagen aislando primeramente el elemento que se pretende encontrar. En unos casos eliminando fondo, en otros casos extrayendo características que se espera que tenga ese objeto, etc. Cualquier proceso de segmentación y morfología de la imagen nos puede servir. Pero siempre tenemos que partir de un conocimiento previo de lo que buscamos. Nuestro objetivo es construir un sistema de aprendizaje, y si esto es así, entramos en una contradicción: ¿Cómo vamos a ser capaces de evaluar lo que no conocemos si para detectarlo tenemos que aislarlo?

Si únicamente tenemos un sistema de visión por computador, tendría que tener un algoritmo tan extenso como para poder llevar implementada toda la secuencia de búsqueda, de procesamiento y de decisión. Y en el momento de aprender, de incluir un nuevo concepto, sería necesario que el propio sistema de visión tuviera conocimiento de su carencia e incluyera el resto de información de forma automática.

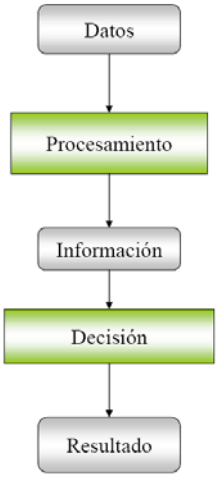


Fig. 20 Esquema de un sistema de visión por computador tradicional

De forma general, se puede decir que el conocimiento se adquiere mediante una variedad de procesos cognitivos: percepción, memoria, experiencia, razonamiento o aprendizaje entre otros.

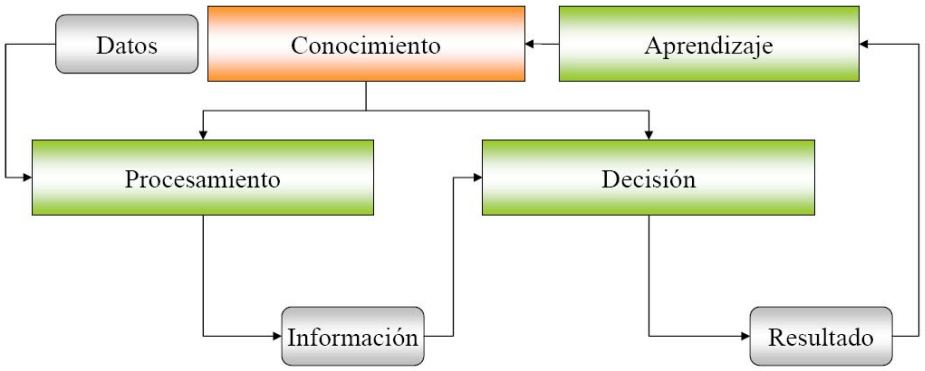


Fig. 21 Esquema de adquisición de conocimiento, generación y aplicación, desde un punto de vista computacional.

En un entorno científico y técnico, el conocimiento se reduce a un conjunto organizado de datos e información que permiten resolver un determinado problema o tomar una decisión. El módulo del conocimiento influye tanto en el procesamiento de la información como en la toma de decisión. Y admite un incremento durante un proceso de aprendizaje.

La percepción es el proceso en que obtenemos la información que interpretamos, aunque realmente esté distorsionada por nuestra propia influencia. Y el resultado de la percepción puede no ser el mismo en dos momentos diferentes ni ante cambios del entorno o de cualquier otra índole. La percepción es variable. Y si es variable no siempre concluye en el mismo resultado. Resulta difícil emular un sistema que aparentemente es estocástico.

Si el mundo es el conjunto de entidades reales y nuestro sistema cognitivo el conjunto de conceptos percibidos e interpretados, ¿qué tipo de aplicación, función o correspondencia los une? ¿Cambia la ley de correspondencia entre ambos y por ese motivo cambia el resultado de la percepción? ¿O es la misma ley con uno o varios factores de evolución? Construimos modelos que emulan las características más notables del sistema de percepción biológico, y de ésta forma podemos evaluar su comportamiento ante tareas de índole cognitivo tales como aprender, responder, razonar, etc.

Según el diccionario publicado por la RAE la *interpretación* es la explicación de acciones, dichos o sucesos que pueden ser entendidos de diferentes modos. El mismo diccionario define *percepción* como el acto de recibir por uno de los sentidos las imágenes, impresiones o sensaciones externas. La combinación de ambas es la proyección en nuestro cerebro.

Nos interesan los factores invariantes, si es que existen, que afectan a este resultado. El capítulo reflexiona acerca estas primitivas, conceptos básicos que comentábamos parecen existir en el ser humano. Esto constituye una base para comenzar el estudio de la visión por computador desde los cimientos de la percepción, evaluando todo el sistema perceptivo y no sólo una parte de él.

3.2 Percepción visual biológica

Nuestro objetivo en este apartado es exponer las diferencias conceptuales y perceptivas a la hora de recibir la información visual e interpretarla, y las claves principales de esta percepción.

Muchos de los estudios realizados en este campo afirman que la visión es una de nuestras vías principales de adquisición de conocimiento, y como consecuencia de aprendizaje. Son muchos los autores que han descrito la percepción visual como el sentido que recoge la mayor parte de la información de lo que pasa a nuestro alrededor [Devore y Devore, 1981] [Kerr, 1982] [Gregg, 1987] [MacLeod, 1991] [Magill, 1980] [Mayoral, 1982] [Revien y Gabor, 1981] [Schmidt, 1988].

[MacLeod 1991] expresa además que la visión es el sistema receptor más exacto por el cual recibimos información del movimiento consciente, de los objetos y de las características espacio-temporales del entorno.

[Kerr 1982] afirma que la mejor forma de conocer la realidad es por medio de la percepción visual. A su vez, [Roncagli 1992], asegura que a través de la visión, el hombre recibe más de dos tercios de la información sensorial que le llega al cerebro, y que esta experiencia puede ser analizada, entrenada, mejorada, orientada y educada para mejorar su rendimiento.

3.2.1 Interpretación de una imagen

En raras ocasiones vemos el mismo objeto desde la misma perspectiva dos veces. Esto supone un problema en los mecanismos de reconocimiento de patrones, ya que no experimentan cambios ante las alteraciones que se puedan producir en los estímulos de entrada. Son plantillas estáticas sobre las que se realiza una comparación. Por otra parte, la misma imagen puede dar lugar a interpretaciones muy dispares en observadores diferentes. Interpretar y reconocer la imagen es una tarea que nuestro cerebro realiza con relativa sencillez. Sin embargo es un proceso difícil de modelar por el desconocimiento que tenemos acerca de su funcionamiento.

Para dar una idea de la dificultad que supone el proceso de percepción e interpretación, proponemos los dos ejemplos siguientes:

Ejemplo 1 Tomamos una persona que tiene suspendido un objeto tras una pantalla de papel. Y otra persona que tiene una lámpara con la que ilumina el objeto, de modo que en dicha pantalla sólo se refleja su sombra. Esta es la imagen del observador. El observador supone que es un objeto e intenta identificarlo. Pero la persona que lo sostiene al otro lado lo gira. ¿Es más fácil reconocer ahora el objeto? ¿Más difícil? ¿Por qué? ¿Qué perspectiva es mejor para reconocer la forma?

Ejemplo 2 En el segundo ejemplo imaginemos un grupo de amigas que se sientan en la mesa de una confitería y, mientras esperan al camarero, advierten la presencia de una imagen impresa en el papel: dos flechas combadas que forman una circunferencia en cuyo centro hay una espiral. Todas ven la misma imagen, pero cada una interpreta algo distinto.

Fig. 22 Flechas combadas del ejemplo de las amigas propuesto

Una de ellas, tal vez guiada por la dirección que marcan las flechas, entiende la representación de un movimiento circular continuo, a lo que otra de ellas, añade que alude al carácter cíclico de algún fenómeno. Alguna puede ver un simple ornamento

sin significado y sin embargo, una interpreta que la espiral central es el fuego y el conjunto simboliza a los Lares, divinidades menores de la antigua Roma. Esta última ha puesto la imagen en relación con el nombre local “Lares”, pero además cuenta con un conocimiento que las demás, o bien no poseen, o bien no activaron.

De una imagen podemos obtener diferentes interpretaciones dependiendo del observador, del mismo modo que para un mismo objeto interpretado, se pueden dar varias imágenes en función del punto de vista de dicho observador.

Las situaciones representadas en estas líneas, dan un ejemplo de una singularidad esencial de la imagen: su polisemia. Esta apertura semántica supera con mucho a la que tiene nuestro lenguaje verbal, y los efectos en cuanto a significado o sentido que le puede atribuir un observador dependen de muchos factores: expectativas, deseos, estado de ánimo, tipo de imagen, cultura, conocimiento, etc.

3.2.2 Revisión de algunos estudios sobre percepción visual

J. C. Ruiz [105], en su libro publicado para alumnos de Bellas Artes, define la *percepción visual* como la sensación interna de conocimiento aparente, que resulta de un estímulo o impresión luminosa registrada en nuestros ojos.

Existe una larga discusión sobre el origen de las percepciones. Sin embargo, se cree que la percepción visual, al menos, requiere un aprendizaje que se va realizando durante toda la vida, aunque casi siempre de modo casual e inconsciente, motivo por el que sufre grandes alteraciones y está ampliamente condicionada por el entorno.

De la larga discusión sobre el origen de las percepciones mantenidas por los filósofos, unos mantienen el *Nativismo*, percepción como reacción intuitiva e innata, y otros el *Empirismo*, percepción como resultado de un aprendizaje y acumulación de experiencias. Hay una tercera postura mantenida por los filósofos de la *Gestalt*, sugiriendo que es producida por una realización característica y espontánea del sistema nervioso central, que pudiera llamarse *"organización sensorial"*. Si bien los últimos experimentos llevados a cabo por Gibson y Walk, con su *"risco visual"*, reafirma la tesis de una percepción innata del espacio,

también se puede pensar que el perceptor establece, de modo inconsciente, un cuadro de comparaciones entre sus impresiones de experiencias anteriores y las sensaciones presentes.

En la percepción visual de las formas hay un acto óptico-físico que funciona mecánicamente de modo parecido en todos los hombres. Las diferencias fisiológicas de los órganos visuales apenas afectan al resultado de la percepción, y eso que, tamaño, separación, pigmentación y otras muchas características de los ojos, hacen captaciones diferenciadas de los modelos.

Las diferencias comienzan con la interpretación de la información recibida; las desigualdades de cultura, educación, edad, memoria, inteligencia, y hasta el estado emocional, pueden alterar el resultado. Es una lectura, una interpretación inteligente de señales cuyo código no está en los ojos sino en el cerebro. Estas formas o imágenes se "leen" a semejanza de un texto literario, y de igual manera requiere un aprendizaje, una gramática que explique sus leyes y profundice en su sentido.

Philip N. Jhonson-Laird [51] realiza un análisis sobre tres de los puntos de opinión más destacados en la concepción de los procesos visuales y su funcionamiento, que describimos brevemente en los siguientes párrafos:

- *Idea: La escena se proyecta como una fotografía: El ojo es como una cámara. Captas la imagen de una escena, la registras y la proyectas en el interior de tu cabeza como una pintura o fotografía.*

  Este anticuado argumento de la psicología no es válido para Jhonson-Laird. El resultado de la percepción visual no puede ser una fotografía dado que a la vez y, de forma sucesiva, esta debería de ser percibida también. Un cuadro colgado de la pared de un museo no tiene ningún sentido hasta que no ha sido percibido.

- *Idea: La visión es imposible: Del mismo modo que antes, consideramos al ojo como una cámara de televisión capaz de captar una escena. Pero ahora consideramos el hecho de que la misma escena puede dar lugar a diferentes interpretaciones, todas ellas válidas.*

  Este segundo argumento, evidentemente, también falla dado que el cerebro no es capaz de seleccionar qué proyección es la adecuada con respecto a la imagen real que se está viendo. Una respuesta natural a este escepticismo sería decir que, teniendo dos ojos, podemos tener percepción de cada uno de ellos y como consecuencia usar esta disparidad para conocer la orientación real de lo que estamos viendo (en el caso del ejemplo nuestras tres barras).

Pero podríamos contestar igualmente que dicha percepción sería imposible si permanecemos sin movimiento o con uno de los dos ojos cerrado. Y añadido a esto, aún nos deberían dar explicaciones acerca del modo en que retina y cerebro ofrecen esta información estereoscópica

- Idea: E*l proceso de la visión es muy sencillo para nuestro cerebro, y muy difícil de comprender para nosotros.*

  Este planteamiento lo considera más cercano a la realidad. Nosotros vemos e identificamos sin esfuerzo alguno y de forma automática. Es evidente que, tras de sí, este funcionamiento tiene una larga experiencia evolutiva. Si vemos un tigre, evitamos acercarnos. No paramos a examinar nuestro proceso visual para ver si funciona correctamente. Sin embargo, a la ciencia cognitiva se le plantea un serio problema en la búsqueda de aquello que es ventajoso para cada especie. Es difícil descubrir cómo trabaja la visión.

Sugiere que, para resolver el problema sea cual fuere el punto de vista empleado, necesitamos como mínimo tres niveles de explicación: (1) Una teoría de *"Qué"* es procesado, (2) *"cómo"* procesa el sistema, y por último (3) la neurofisiología, o dicho de otra forma, el sistema nervioso y su estructura.

3.2.3 Percepción de la imagen

La cantidad de energía y las diversas composiciones espectrales de la luz que llega al ojo son los estímulos físicos correspondientes a las diferencias de color que observamos en el campo visual. Los rayos de luz penetran en el ojo y son enfocados sobre la superficie del globo ocular. Sobre esta superficie se extiende la retina, una estructura sumamente compleja de terminaciones nerviosas. La luz llega a estos receptores, los excita según su energía y longitud de onda, y provoca descargas eléctricas que son transmitidas al cerebro. A base de señales como estas, que provienen de multitud de localizaciones diferentes sobre la retina, construimos nuestra representación visual de la realidad.

3.2.3.1 La Forma.

Los psicólogos *gestaltistas* estudiaron las leyes que operan en la transformación de ese mosaico de señales que emana de la retina en imágenes de objetos, en formas. Puede demostrarse experimentalmente que, de acuerdo al color de los puntos que la componen, las distintas partes de la imagen retiniana tenderán a unificarse o a segregarse en unidades perceptuales. Como explicación a este fenómeno la escuela de la *Gestalt* ha definido la llamada *"Ley perceptual de la semejanza"*. A esta ley corresponderá como corolario, una ley de la desemejanza en el sentido de que las partes diversamente coloreadas del campo visual tienden a segregarse, apareciendo como unidades perceptuales diferentes.

Las diferencias o contrastes que provocan la segregación pueden ser de diversos tipos. Las tres categorías en que agrupamos las diferencias percibidas entre los colores son el brillo, el matiz y la saturación. De estas posibles diferencias, la más común es la del contraste por el brillo. La fotografía en blanco y negro forma parte de nuestro paisaje cotidiano. El contraste de claridades es también el más efectivo. Los contrastes de matiz y saturación rara vez se encuentran en estado puro.

Lo normal es que ambos aporten información cualitativa adicional a la imagen básica obtenida por contraste. Gracias al matiz podemos saber si una fruta está verde o madura. Si desaparece el contraste por claridad la percepción de la forma se debilita. Cuando una figura y su fondo son equivalentes en claridad, ambos tienden a fusionarse aunque tengan diferente matiz. Este efecto se conoce en psicología como el *efecto Liebmann*. En estos casos se hace difícil o hasta imposible el precisar el contorno, dando origen a un efecto de inestabilidad o vibración óptica que ha sido con frecuencia empleado en la gráfica contemporánea.

En la tendencia a la agrupación o segregación dentro de la imagen retiniana influye de manera decisiva la segunda variable señalada por Gibson [126]: el ordenamiento de los puntos luminosos de propiedades afines sobre el mosaico de terminaciones nerviosas de la retina. La proximidad influye. Los puntos vecinos se agrupan entre sí con preferencia a los distantes. Las transiciones entre superficies diversamente coloreadas dan lugar a la percepción del contorno, límite y elemento característico de la forma. La manera en que los distintos contornos se articulan en la imagen es decisiva para la estructuración de unidades formales en virtud de la simetría, el cierre, la continuidad y la simplicidad o buena forma. Estos factores han sido bien estudiados en los textos clásicos de psicología de la percepción.

3.2.3.2 Estructura del Campo Visual.

El contorno da forma a una figura, delimitándola frente al resto del campo visual que aparece como fondo. Este fenómeno a través del cual una forma visual se diferencia del campo total de la percepción, emergiendo como figura que se destaca contra un fondo, resulta de vital importancia para la estructuración del campo visual en unidades perceptuales coherentes. La relación entre la figura y el fondo tiene siempre un carácter espacial. El fondo no está limitado por el contorno de la figura, dando, al contrario, la impresión de que continúa por detrás. El fondo se localiza usualmente a una distancia indefinida detrás de la figura. La influencia de los colores sobre la percepción de las relaciones espaciales no puede desligarse del papel de las regiones coloreadas como figuras o como fondo. En la perspectiva atmosférica clásica, los objetos cercanos son más oscuros que los alejados, pero esto tiene sentido sólo si consideramos que en ese caso el fondo es claro. La relación seguramente se invertiría para un fondo oscuro.

En general, las superficies de color más parecido al color del fondo parecen estar más cerca del fondo, lo que usualmente equivale a estar más alejadas del observador. La percepción del espacio inducida a partir de la relación figura-fondo puede entraren conflicto con la tendencia a percibir algunos matices como si estuvieran más próximos al observador que otros, es decir, colores que avanzan o retroceden en el espacio. En este caso, el color pudiera llegar a invertir el campo, apareciendo lo que era figura como un hueco.

3.2.3.3 Interacción del color.

No se puede aislar el estudio del color de su función dentro del campo visual estructurado. Es especialmente importante en el estudio de los efectos de interacción donde el color percibido es el resultado de la influencia recíproca de todas las regiones coloreadas del campo.

La apariencia de un color en un campo visual de estructura compleja está condicionada por una serie de fenómenos entre los que se encuentran las dimensiones absoluta y relativa de las regiones coloreadas.

3.2.4 Percepción de espacio y profundidad

Percepción es la impresión del mundo exterior alcanzada exclusivamente por medio de los sentidos. Es una interpretación significativa de las sensaciones. Limitando el estudio de las percepciones sólo al campo visual, diremos que es la sensación interior de conocimiento aparente que resulta de un estímulo o impresión luminosa registrada en nuestros ojos.

La función de un ojo se puede resumir en dos fases: una recepción del estímulo de luz sobre la retina, el elemento sensitivo, y un envío de impulsos nerviosos al cerebro que dependen de la física de la luz y de las lentes, así como de la bioquímica de las células nerviosas del ojo.

Así bien, la cognición no es meramente un problema computacional de cómo transformar símbolos mentales. Estos símbolos pueden ser creados perfectamente por la interacción física con el entorno, una interacción física ampliamente desconocida. Los robots llevan cámaras electrónicas que realizarán la tarea sensitiva con algo más de crudeza que el ojo humano. Sin embargo, no nos vamos a centrar en esta parte sino en lo que ocurre después.

Para el robot, lo realmente interesante es la representación simbólica de su entorno, de su espacio de movimiento y de las cosas, objetos y personas que puedan ir encontrándose en su camino. Por eso distinguiremos varios tipos de representación simbólica en función del nivel de computación exigido y la forma en que se capta la imagen.

3.2.4.1 Claves perceptivas de la profundidad

Nuestra percepción del mundo real es tridimensional. Sin embargo, la información que nos da la cámara es una representación plana, dando lugar a una percepción de un espacio óptico, fingido o simulado.

Podemos considerar el mundo real un espacio euclidiano en tres dimensiones, al menos en los límites de la visión humana, pero los modelos y las imágenes que recibimos en la retina son bidimensionales. En este sentido, son importantes las

aportaciones que hicieron James J. Gibson, Eliane Vurpillot, D. M. Armstrong, Gyorgy Kepes y, aunque ya obsoleta en algún punto, la obra de Berkeley.

Según sus aportaciones, percibimos el espacio, la profundidad y en general la tercera dimensión en función de dos grupos de factores clave:

- Claves primarias:

  (3) Disparidad binocular

  (4) Convergencia ocular

  (5) Ajuste o acomodación

  (6) Paralaje de movimiento

  (7) Desplazamiento del observador

- Claves secundarias:

  (8) Tamaño

  (9) Oclusiones parciales

  (10) Sombras

  (11) Texturas y detalles

  (12) Llenos y vacíos

  (13) Borrosidad y desenfoque

  (14) Horizontalidad y borde inferior de la imagen

  (15) Perspectiva lineal

  (16) Color

(17) Perspectiva aérea.

La representación de la imagen queda sin embargo completamente definida en nuestro cerebro con la ausencia de los factores primarios. En realidad, la percepción de un cuadro nos permite conocer los factores relativos al punto de vista, fondo, formas, etc. sin necesidad de conocer la realidad tridimensional de quien lo pintó. Sólo con las claves perceptivas secundarias nuestro cerebro es capaz de interpretar la escena.

Fig. 23 En estos cuadros se puede apreciar el fondo de escena, la lejanía y cercanía de los objetos, su forma, etc. sin necesidad de conocer su información tridimensional. El pintor nos lo ha mostrado únicamente con combinaciones de colores, texturas, sombras y textura.

Es por este motivo que vamos a intentar describir primeramente, los aspectos más generales de la interpretación de imágenes dejando de lado cualquier aspecto emocional y teniendo en cuenta únicamente aspectos de naturaleza cognitiva: *¿Qué necesitamos conocer como observadores para interpretar una determinada imagen?*

Por su importancia, el siguiente apartado está dedicado a una exposición más amplia de las *Claves secundarias* y de las características tridimensionales que aporta cada una de ellas. Aunque el apartado se denomine *Espacio y profundidad: Claves perceptivas*, nos estamos refiriendo a las secundarias, puesto que queremos interpretar espacio con una única imagen de dos dimensiones.

3.3 Espacio y profundidad: Claves perceptivas

La base de conocimiento con la que trabajará nuestro sistema de visión, necesita conceptos visuales, relaciones entre ellos y axiomas que nos permitan concluir en

decisiones adecuadas acerca de lo que esta procesando la cámara. Por su importancia en los siguientes apartados, exponemos brevemente cada concepto, es decir, cada una de las claves secundarias para la percepción tridimensional.

3.3.1 Tamaño

Tamaño y distancia son las formas de dos conceptos empíricos base de la percepción de profundidad. Existe una ley denominada *"la consistencia del tamaño"*, cuya teoría está aún es discusión, que implica la relación tamaño-distancia de un objeto y el tamaño real de su impresión en la retina. En realidad, la impresión depende del ángulo visual o superficie de retina impresionada, único elemento estable y objetivamente fiable en este problema.

El tamaño que "tenemos aprendido" de las cosas lo comparamos con el que realmente percibimos. Esto está ampliamente estudiado por la perspectiva geométrica. Durante una proyección cinematográfica en una sala a oscuras, perdemos la escala comparativa del tamaño de las imágenes, y aceptamos que un primer plano de una cabeza humana tiene tamaño real. Esta ilusión se destruye al aparecer la silueta viviente de un espectador que casualmente se levanta. Está imponiendo la distancia perceptiva con la realidad asumida de su tamaño.

Fig. 24 Tres figuras geométricas sin valor asociativo y como consecuencia, sin referencias para conocer su magnitud real

En la figura 58 se exponen tres figuras geométricas sin valor asociativo alguno. No podemos conocer sus magnitudes reales. Al observarlos podemos tener dos situaciones válidas.

- Si los consideramos en el mismo plano de la imagen, por lo tanto a la misma distancia del observador, nos parecerá que tienen tres tamaños diferentes.

- Si pensamos que los tres son del mismo tamaño, empezaremos a ver unos más alejados que otros.

Esta alteración entre el tamaño y la distancia se produce en este caso porque no interviene la experiencia, que al fin y al cabo, es la persistencia del tamaño preestablecido en las figuras.

Sin embargo, en la siguiente imagen la profundidad se resuelve con varias claves concordantes. Sin embargo, de todas las utilizadas, es el tamaño de las figuras humanas, que se reducen de forma progresiva, el que ofrece una clave eficaz y sirve de ejemplo ilustrativo.

Fig. 25 Imagen de una obra de Degas donde la profundidad queda resuelta con varias claves concordantes

3.3.2 Oclusiones parciales

Cuando un objeto se interpone entre la cámara o el observador y la figura de la escena oculta parcialmente, se produce otra de las claves secundarias más eficaces de la percepción tridimensional.

En este caso, el problema vendría de la mano del procesamiento de la imagen: De los algoritmos utilizados para la percepción de la forma, para los cuales no existe el concepto de oclusión. Sólo el concepto de rotura de continuidad, que supone un límite o borde final del objeto, aunque realmente no lo sea.

No obstante, si la forma que da lugar a esa oclusión se puede percibir en su forma y tamaño completos, tendremos de nuevo la posibilidad de percibir espacio entre ambas. Cercanía y lejanía.

En esta clave inciden dos propiedades:

- La opacidad y corporeidad por defecto de los cuerpos, exceptuando los cuerpos traslúcidos que de nuevo, nuestra experiencia y aprendizaje ordenan adecuadamente.

- El *"efecto pragnante"* o de *"buena forma"*, que tienen las figuras completas con respecto a las que quedan fragmentadas. Una figura plena siempre se adelanta en la captación receptiva porque ofrece menos duda en su interpretación, y esta anticipación en el tiempo respecto de la más complicada, acarrea una anticipación en el espacio.

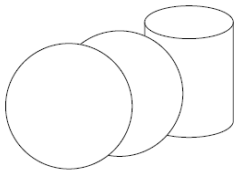


Fig. 26 La interposición del primer círculo, ocultando parte del segundo, y la parcial oclusión del cilindro, crean un caso típico de percepción tridimensional de la profundidad. Generalmente, nunca se piensa en el segundo círculo como en una luna en cuarto creciente, ni en el cilindro como un volumen irregular mixtilíneo.

Sin embargo, es un concepto que requiere un análisis más detallado puesto que no está exento de errores de interpretación.

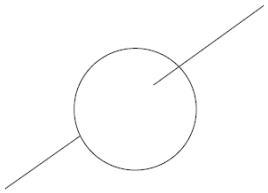


Fig. 27 Representación de un círculo y dos segmentos de recta. Se nos presenta como si se tratase de una única línea que atraviesa un volumen esférico, y no como dos segmentos radiales. Se ha creado de esta manera una sensación de cuerpo o volumen, sin otro recurso que la interrupción de una recta y la "*buena forma"* de los dos elementos.

3.3.3 Sombra

Otro de los factores de percepción de espacio es la iluminación. La luz se muestra a nuestros ojos por contrastes y efectos de sombras. Es conveniente recordar que la mayor o menor cantidad de luz reflectante en los objetos, no se mide física o matemáticamente, sino por comparación o contraste entre las partes claras y oscuras. En realidad no importa el valor absoluto de esta intensidad, sino los relativos de proporcionalidad de luminancia.

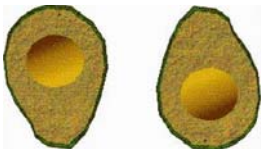


Fig. 28 Curioso fenómeno óptico donde la misma figura girada, provoca percepciones diferentes en cuanto a la profundidad y el relieve.

En visión por computador se deben buscar claves unidas a singularidades adicionales para poder realizar el estudio de la imagen correctamente. En realidad, en un sistema de visión no importa la percepción de una zona más oscura en la imagen, si no tenemos los axiomas y relaciones necesarios para determinar que se trata de una sombra. Por ejemplo, la misma circunferencia con los mismos colores, puede dar sensación tridimensional o bidimensional, si en el interior de la misma se pueden detectar bordes durante la etapa del procesamiento.

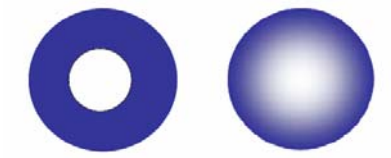


Fig. 29 Efecto volumétrico esférico entre los límites difuminados de la imagen de la derecha que contrasta con el efecto planar de los límites lineales de la imagen de la izquierda.

En un sistema de visión por computador con una base de conocimiento y decisión asociados, tendríamos la oportunidad de realizar un ejercicio como el que se muestra en la siguiente imagen, donde se pueden combinar conceptos visuales iguales como el color o la detección de contornos, en axiomas que permitan concluir resultados distintos.

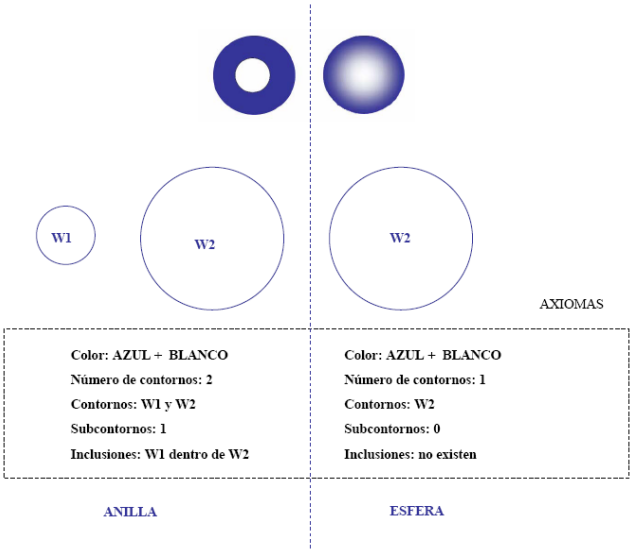


Fig. 30 Construcción de un conjunto de axiomas combinando los mismos conceptos visuales que concluyen en percepciones tridimensionales diferentes.

3.3.4 Textura

La aportación de claves de profundidad tan importantes como las de gradientes de texturas, se deben a las investigaciones de James J. Gibson, realizadas durante la segunda guerra mundial. El concepto de *gradiente* lo utiliza como el aumento o la disminución de algo a lo largo de un eje o una dimensión determinada, y lo relaciona con las curvas de la geometría analítica.

La textura no precisa formas limitadas u objetos concretos para producir la profundidad de espacio. Se fundamenta en realidades fisiológicas como la capacidad de nuestro órgano visual o de las cámaras para captar pequeños detalles según las distancias.

La capacidad de las cámaras, lo mismo que la de nuestra retina, es limitada, por lo que la nitidez se establece en función de la distancia a los objetos observables y al ángulo visual de la óptica. La textura no es más que la percepción por tamaño, nitidez, detalle, colores límpidos o por sombras más o menos contrastadas.

En la propia naturaleza es donde mejor podemos observar el fenómeno de la textura. En la siguiente imagen se muestra con bastante claridad de detalle cómo la lejanía se hace borrosa hasta la confusión, en contraste con la nitidez de las primeras formas que corresponden al borde inferior de la imagen.

Fig. 31 Efecto óptico de nitidez en las zonas cercanas, y difuminado en las más alejadas a la cámara.

Como en el caso de las sombras, en un sistema de visión por computador un operador Canny podría ofrecer muchos bordes en la zona inferior de la imagen, en contraste con la disminución de los mismos en la zona superior.

Sin embargo, el problema no es tan sencillo como se presenta en este momento. Cualquier cámara, lo mismo que el ojo humano, puede desenfocar los objetos cercanos para dirigir el objetivo a los más alejados. Su estudio va a depender del entorno de trabajo del robot.

3.3.5 Llenos y vacíos

Es una clave perceptiva de menor eficacia que las anteriores pero, aún teniendo mucho que ver con la textura y la distancia, merece una mención aparte. Tomando dos superficies iguales de tamaño y forma, pero una llena y la otra vacía, la sensación es que la llena está más cerca mientras que la vacía se distancia del observador.

Probablemente sea una de las claves perceptivas con menos aplicaciones en un sistema de visión enfocado al reconocimiento de objetos, puesto que se utilizaría más como método de evasión de obstáculos que como medio para percibir el espacio.

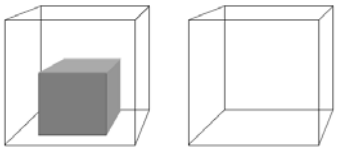


Fig. 32 El objeto de la izquierda da la impresión de estar más cercano que el de la derecha, aún estando ambos en el mismo punto de distancia con respecto al observador.

3.3.6 Borrosidad y desenfoque

La falta de ajuste en la convergencia de la visión binocular produce borrosidad o falta de nitidez en la imagen. Una sola cámara tiene la propiedad de concentrarse en un punto del espacio, o por el contrario, dispersar su campo de visión por el efecto de la distancia focal.

Sin embargo, tanto el ojo como la cámara, tienen una incapacidad notable para ver con nitidez tanto los objetos muy lejanos como los muy cercanos. Este fenómeno lo observó y estudió Berkeley. Si esta clave de percepción de profundidad sólo funcionase con objetos lejanos, podría haberse incluido como parte integrante de la textura. Pero no es así. Los hechos experimentados a diario de no percibir objetos demasiado cercanos al ojo y en su caso a la cámara, son experiencias acumuladas para la eficacia de la borrosidad. Por ejemplo, las gafas no son visibles para quien mira por ellas, y la vaga sombra que proyecta la nariz en nuestra visión es casi completamente ignorada.

Hay otro factor importante que se puede observar en los objetos que se mueven. Los más cercanos se desplazan con mayor rapidez por la retina o por la lente de la cámara que otros que se mueven con igual velocidad pero su posición es más lejana. Esta velocidad relativa de los objetos cercanos produce imprecisión y borrosidad, mientras que las cosas estáticas o lentas, por su mayor distancia, se perciben mejor en sus detalles.

3.3.7 Horizontalidad

El horizonte constituye una referencia básica para la posición vertical. Lo lejano se asienta en el horizonte y lo cercano se aleja de él. Si los ejes vertical y horizontal se materializan con el entorno inmediato en una imagen, tomaremos por válidos elementos coordenados como el ángulo de la habitación o los marcos de las ventanas y cuadros, las aristas del suelo, etc. Si se pierden estas referencias es prácticamente imposible distinguir estos conceptos menores.

Las leyes de percepción visual, por medio de sus claves de profundidad, sólo pueden mostrarnos las interpretaciones correctas del espacio. Nos fijan con rigor científico el lugar que ocupan las cosas en la tercera dimensión y la estructura organizativa de interdependencias de las formas y objetos en el espacio euclidiano. Lo que no pueden mostrarnos jamás, de manera espontánea, es el significado de esas formas y objetos. Esto requiere un aprendizaje diferente en campos como la iconología, la simbología y la semiótica o semántica de la imagen.

.

En un sistema de visión podemos sin embargo utilizar estas características en muchos sentidos. Por ejemplo, supongamos que la *Transformada de Hugh*, un detector lineal muy utilizado en sistemas de visión por computador, detecta en la imagen diez líneas: R1 y R2, las cuatro líneas P1 a P4, y las otras cuatro C1 a C4. Supongamos también que nuestro sistema de conocimiento debe elaborar unos axiomas de decisión así como relaciones que permitan identificar objetos con ellas.

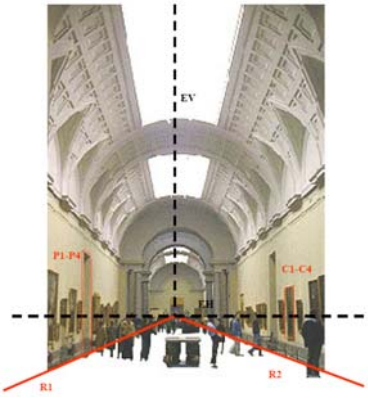


Fig. 33 Referencias a partir de la horizontalidad. En esta imagen además de ofrecer información acerca de la lejanía, la horizontal nos ofrece un punto de referencia a partir del cual podemos comparar otras líneas detectadas en la imagen.

Así por ejemplo podemos tomas las líneas P1 a P4 y comprobar que dos de ellas son paralelas al eje EV y que además se cruzan con R1. Es la diferencia que tienen frente a las líneas C1 a C4, ya que éstas no cruzan con R2. En el primer caso, estando en el entorno de un museo, podemos concluir que es una puerta, mientras que en el segundo, es muy probable que la figura que estamos viendo pueda ser un cuatro.

3.3.8 Perspectiva lineal

Por las leyes de la geometría proyectiva, las líneas rectas contenidas en planos paralelos al observador, mantienen sus propiedades métricas de posición y dirección, mientras que las perpendiculares a estos planos convergen en un punto y se transforman en múltiples oblicuidades. Sus fundamentos matemáticos ponen en juego un lenguaje coherente y lógico, de análisis y razonamiento de formas, que permite manejarlas, transformarlas y reconstruirlas con un rigor científico que tolera la transferencia a otros espacios.

Es uno de los medios utilizados para examinar el espacio y las formas tridimensionales, a la vez que ayuda a comprender la naturaleza y geometría estructural del objeto y su entorno. El análisis de una composición de escena en una imagen según el método de la perspectiva lineal, revela algunas características esenciales de la visión:

- Todas las líneas paralelas de la naturaleza convergen en un punto llamado *"punto de fuga"*.

  (18) Para cada grupo de paralelas hay un punto de fuga diferente. Sin embargo, los puntos utilizados para construir objetos paralelos a la superficie existen en la misma horizontal, llamada *"línea horizonte"*.

  (19) Las líneas redondas se dibujan como si estuvieran inscritas en rectángulos cuyos lados fuesen tangentes a la curva

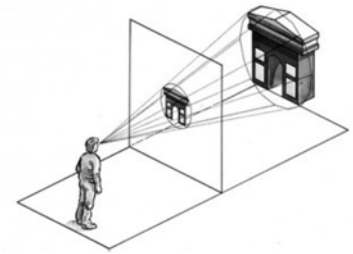


Fig. 34 Perspectiva lineal

3.3.9 El color

Es uno de los temas más complejos y sin embargo más utilizados en la visión por computador. El planteamiento del color como clave perceptiva nos obliga a mencionar las características que afectan dicha clave.

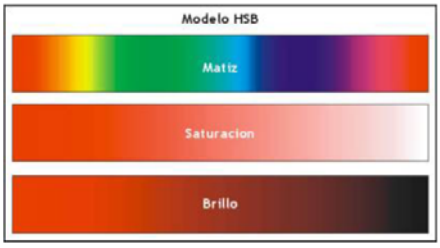


Fig. 35 Modelo de color HSB

El color tiene una serie de propiedades que no se relacionan directamente con la longitud de onda, sino con el receptor que lo capta, ya sea célula o sensor.

- *Tono o matiz:* Es lo que llamamos diferencialmente "color", con nombres como rojo, verde, amarillo, etc. El término "tono" está sacado de la terminología musical, y expresa la mayor o menor intensidad, mientras que "matiz" se refiere más a la cualidad diferencial con los demás.

(20) *Brillo o Valor:* Cualidad del color por la que se puede equiparar a la familia de los grises, que van del blanco al negro. En este sentido un azul siempre será más negro que un amarillo.

(21) *Saturación o Intensidad:* Es la sensación de fuerza o debilidad de un color, o su mayor o menor participación del blanco. Un blanco siempre será un color menos saturado que un rosa.

Es una de las propiedades más ampliamente estudiadas y utilizadas en el estado del arte de la visión por computador, por lo que no nos vamos a parar en este punto.

3.3.10 Perspectiva aérea

Se diferencia de la perspectiva lineal en que la perspectiva aérea se altera con los valores de iluminación, y no tiene valores matemáticos ya que no se funda en la óptica geométrica. Es de menor rigor y control.

Las formas distintas pierden la tajante definición de su contorno y aparecen más claras y menos diferentes que los objetos próximos al observador.

La perspectiva aérea muestra formas distantes en colores más claros y menos brillantes que los empleados en figuras situadas en el primer plano.

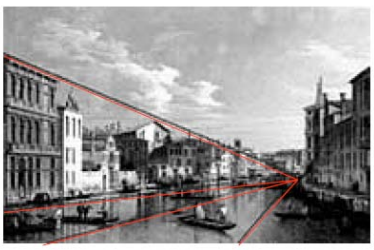


Fig. 36 Imagen con las líneas de la perspectiva aérea marcadas en rojo

3.3.11 Discusión final

El hombre aprende constantemente almacenando experiencias, casi siempre de modo inconsciente. Establece de forma automática relaciones y vinculaciones entre *formas y objetos*, y más adelante entre *objetos y función*, con asociaciones a otras experiencias semejantes. Esta asociación continua que el hombre hace entre las *formas y las cosas*, las *cosas y su uso*, el *uso y la historia*, forman una cadena de *adquisición de conocimiento*. Una cadena de aprendizaje.

El conocimiento de las leyes descritas en este apartado, las claves perceptuales, constituyen el alfabeto básico del lenguaje visual. Estas leyes que nos ofrece la psicología contemporánea pueden ser ignoradas pero ello no impide que se sigan cumpliendo. La mayor parte de los trabajos de visión por computador 2D no tienen referencias entre las claves perceptivas como para poder relacionar espacialmente los objetos y poder obtener conclusiones más complejas. Siempre que se necesita información espacial se recurre a las claves de la visión tridimensional: disparidad binocular, convergencia, paralaje o desplazamiento.

Las claves perceptivas que se han expuesto en este apartado nos revelan el mundo real, el mundo tridimensional sobre un papel o un lienzo. Pero del mismo modo que estas claves funcionan en un sentido, ¿podrían funcionar en la misma dirección y sentido contrario? Es decir, ¿qué claves pueden servir para evaluar el papel y darnos conocimiento del mundo?

3.4 Análisis en un marco teórico de percepción

El hombre adquiere conciencia de si mismo y del mundo que le rodea por medio de los sentidos y de los estímulos que éstos recogen. A través de ellos descubrimos, organizamos y recreamos la realidad, adquiriendo conciencia de ella por medio de la percepción.

Conviene diferenciar correctamente *estímulo* y *percepción*. El *estímulo* pertenece al mundo exterior y produce un primer efecto en la cadena del conocimiento. Se refiere a todo aquello que active un receptor sensorial: calor, frío, rojo, etc. La *percepción* pertenece al mundo interno e individual de cada persona, al proceso psicológico de la interpretación y conocimiento de las cosas y de los hechos, es

decir, depende del observador y de su entorno por medio de estímulos que recibe de este pero además, recibe influencia del resto de los procesos del propio sistema.

Si nos quedáramos aquí, estaríamos teniendo en cuenta sólo un sentido de la interacción, la de entradas al sistema observador. Sin embargo el propio sistema ejerce influencia sobre el entorno.

El acto perceptivo, aunque cotidiano y realizado de forma automática, no es nada simple y tiene múltiples implicaciones. La identificación de la realidad por las impresiones que se producen en nuestros sentidos es una de las evidencias más firmes de la misteriosa perfección de la mente humana

3.4.1 Marco teórico de análisis

El hecho de que querer implementar sistemas basados en la percepción biológica, nos lleva a la necesidad de trabajar dentro de un marco teórico que pueda explicar, tanto la percepción de los sistemas biológicos, como la de los sistemas artificiales. Ese marco/teoría general es el presentado por el Dr. Ignacio López [104].

El motivo por el que se ha elegido este modelo y no otro, es por la cercanía tanto de su propuesta como de los ejemplos de aplicación, a las necesidades que plantea nuestra línea de investigación. Este modelo se enmarca dentro de la *Teoría General de Sistemas*. En concreto sigue la formulación realizada por Klir [107].

3.4.1.1 Generalidades del modelo de percepción

La mayor parte de los estudios realizados acerca de la percepción se basan en modelos biológicos. De estos modelos y de la revisión de las tendencias fundamentales en este campo, podemos distinguir dos categorías de percepción principales:

(1) *Percepción indirecta o mediada*: Asume que la percepción es un proceso de inferencia, que depende no sólo de la naturaleza de los sentidos, sino también de aspectos propios del sistema como su conocimiento, experiencias o emociones. El propio sistema intercede en él mismo.

(2) *Percepción directa*: Se trata de una teoría propuesta por J. J. Gibson [106] y propone que la percepción se lleva a cabo de forma espontánea sin procesos de inferencia por los cuales se busquen explicaciones a las lecturas de los sentidos. Afirma que las lecturas sensoriales contienen un significado coherente para el sistema por sí mismas.

El estudio realizado por I. López [104] afirma que todo proceso perceptivo implica tres aspectos:

(1) *Estímulo cercano*: magnitud medida por el sistema sensorial sin sufrir modificación alguna del entorno de acción.

(2) *Singularidades*: patrones atribuidos a una configuración concreta de los objetos del entorno. Es la que se representa como conjunto de objetos percibidos.

(3) *Objeto*: concepto, idea o entidad conceptual

En la siguiente figura se muestra el esquema de la secuencia fundamental de percepción según el modelo elegido. SP y DP representan las dos fases de la secuencia fundamental. SP se encarga de extraer las singularidades y DP establece las relaciones de equivalencia entre los objetos y una parte del entorno que interpretará como objetos. Es decir, analiza las singularidades que ha separado SP.

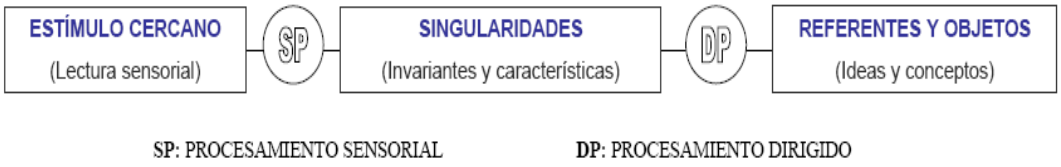


Fig. 37 Secuencia fundamental de percepción según modelo de I. López

El proceso está orientado a reconocer objetos específicos del entorno y estos objetos se denominan *referentes* del proceso.

De acuerdo con estos conceptos, la *percepción* se concibe como un proceso que produce cambios en el sistema, relacionados sin aleatoriedad con el estado del entorno percibido. Podemos distinguir en ella tres fases:

(4) El entorno produce ciertos cambios en las variables del perceptor.

(5) Las dependencias mutuas entre estas variables modificarán las variables* del perceptor. Los consideramos *cambios implícitos*.

(6) La variación en las variables del perceptor modificará las variables en la frontera perceptor-sistema. Son la representación que se denominó en líneas anteriores *objeto percibido*.

Estos son los conceptos fundamentales que necesitamos para entender el ejemplo propuesto en este trabajo. Se remite al lector a la tesis doctoral de I. López [104], para una descripción detallada tanto del modelo, como de los conceptos de percepción, perceptor, dinámica perceptiva, memoria perceptiva y sistemas de percepción.

3.4.1.2 Descripción del modelo de percepción

Entendemos biológicamente que un sentido es un proceso fisiológico de recepción y reconocimiento de sensaciones y estímulos que se produce a través de la vista, el oído, el olfato, el gusto o el tacto, o la situación de su propio cuerpo (RAE).

Un sentido realiza el análisis necesario del entorno perceptivo desde un punto de vista determinado, el que corresponda dentro de la estructura funcional del sistema. Formalmente, un punto de vista consiste según G. J. Klir [107] en:

(7) Nivel de resolución, que define la operación del sentido en el espacio y en el tiempo. Relativamente análogo a la calidad de la señal que contiene la información.

(8) Un conjunto de cantidades consideradas.

(9) Relaciones de comportamiento entre las cantidades.

(10) Las propiedades que determinan estas relaciones.

Un perceptor puede integrar, en su caso general, múltiples sentidos y percibir un conjunto de *referentes*.

Procesamiento de información cercana, SP

Simplificando la idea propuesta en la tesis, el procesamiento de información cercana se refiere a todos los procesos que intervienen en calcular *singularidades* a partir del *estímulo cercano*.

Distingue dos tipos de proceso:

(11) *Equivalencia de singularidad:* Es el cálculo de un conjunto de *singularidades* a partir de los valores de las cantidades del sistema sensorial.

Conjunto de singularidades considerado:

$$\Psi = \{\psi_j, \quad j = 1...n_\Psi\}$$

[Ec. 1]

Conjunto de cantidades de sistema sensorial empleadas para calcular cada una:

$$Q_k \xrightarrow{\sigma_k} \psi_k \qquad \psi_k = \sigma_k(Q_k)$$

[Ec. 2]

Por lo tanto, el proceso de información cercana calculará n_ψ relaciones de equivalencia de singularidad (σ_k)

(12) *Ecualización:* Modificación del valor de una cantidad de entrada a un proceso de *equivalencia de singularidad*.

Si indicamos con q_k^{ss} el valor de la cantidad k-ésima del sistema sensorial, la ecualización equivale a:

$$q_k^{ss} \xrightarrow{eq_k} q_k^{ss*}, \qquad q_k^{ss*} = eq_k(q_k^{ss})$$
$$eq_k \rightarrow \textit{proceso de ecualización de la cantidad } k-\textit{esima}$$

[Ec. 3]

El valor ecualizado q_k^{ss*} sería la entrada al proceso de *equivalencia de singularidad*.

El proceso de ecualización se hace necesario en los casos en que deben corregirse desviaciones en la lectura del dispositivo sensorial, y en aquellos en que la lectura sea correcta, pero sea necesario adaptar la señal al punto de vista del sentido.

Procesamiento de información cognitiva, DP

El proceso que lleva a cabo un sentido se ajusta a la secuencia fundamental de percepción propuesta en el trabajo de referencia [104]:

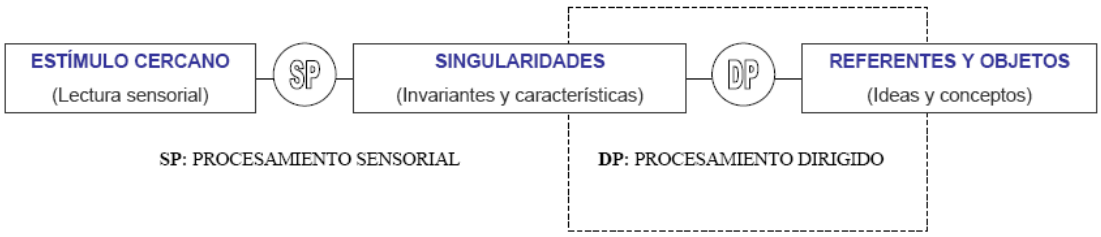


Fig. 38 Procesamiento de información cognitiva, DP

En uno de los sentidos el DP procesa *singularidades* para inferir un cierto estado del entorno perceptivo, en concreto el estado de los objetos del entorno que se corresponden con el *referente*. El resultado es el *objeto percibido*.

El proceso de inferencia consiste en la deducción de un estado del entorno. Es decir, establece una serie de equivalencias entre el conjunto de singularidades observado,

$\Psi = \{\psi_j, \quad j = 1...n_\Psi\}$, y el referente sentido, ρ. Esta equivalencia la podemos representar como una relación ξ, de forma que llamaremos *objeto percibido* a esto:

$$\rho^R = \xi(\Psi)$$
$$\xi \rightarrow \textit{equivalencia cognitiva}$$

[Ec. 4]

El superíndice R indica que se trata de una representación de una instancia del referente. Ya hemos dicho que el perceptor puede integrar múltiples sentidos y percibir un conjunto de referentes:

$$V = \{\rho^i, \quad i = 1...n_\rho\}$$

[Ec. 5]

La operación del perceptor entonces se puede expresar con un número igual de relaciones de equivalencia:

$$\rho^{iR} = \xi^i(\Psi^i) \qquad i = 1...n_\rho$$

[Ec. 6]

La formulación sólo es válida conceptualmente, ya que pueden existir relaciones de equivalencia que dependan de otros objetos percibidos y no de singularidades directamente, y casos mixtos en que dependa de ambas cosas a la vez.

3.4.2 Sistema de localización y rastreo facial

El método de detección facial que analizamos está diseñado para detectar la presencia de una cara en la imagen y no perder su localización ante pérdidas eventuales en la detección.

En la imagen siguiente, Fig. 39, se muestra el diagrama de distribución física de configuración del sistema que se va a emplear durante el desarrollo de los algoritmos y donde estará funcionando la solución final.

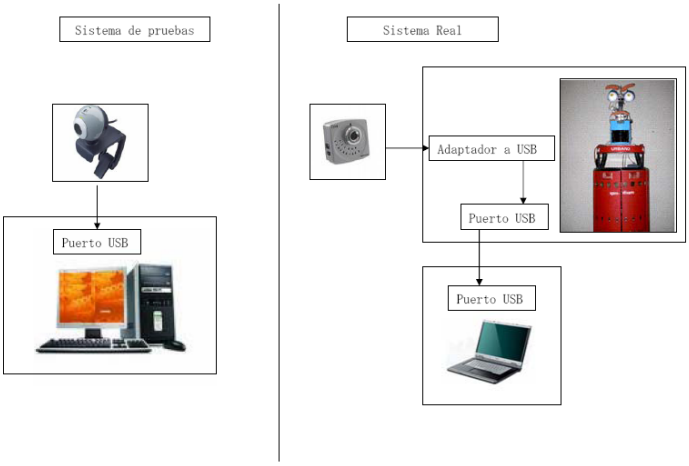


Fig. 39 Diagrama de distribución. A la izquierda el sistema utilizado para implementar los algoritmos de visión. A la derecha el sistema real donde irá implementado el resultado final

En este apartado aplicamos el marco conceptual descrito, al sistema de detección facial incluido en el desarrollo realizado para este proyecto. En primer lugar debemos identificar cada parte del sistema de visión por computador con el elemento adecuado del marco teórico.

3.4.2.1 Descripción del sistema

Para la realización de éste proyecto se ha utilizado una estación de trabajo PC con OS Windows XP. En él se han realizado el desarrollo y la ejecución de los algoritmos de visión por computador y el procesamiento de la imagen. De acuerdo con la línea de trabajo que viene teniendo el grupo de Control Inteligente, este programa se ha realizado en lenguaje C++ sobre los entornos de desarrollo MS Visual Studio .NET 2005 (Visual Studio 8). De esta manera se consigue compatibilidad con los demás trabajos del grupo y una adecuada portabilidad de los desarrollos implementados.

El lenguaje de programación C++ es uno de los más empleados en la actualidad. Se puede decir que es un lenguaje híbrido que permite programar tanto en estilo procedimental, en estilo orientado a objetos o ambos a la vez. Permite además ser empleado mediante programación basada en eventos para crear programas que necesitan interfaz gráfico de usuario.

Algunas de las principales ventajas que presenta el lenguaje C++ son primeramente su difusión (es uno de los lenguajes más empleados en la actualidad) y las ventajas en cuanto a bibliografía y generalidad de uso que esto aporta. Es un lenguaje versátil, de propósito general, por lo que se puede resolver una muy amplia variedad de problemas. Otra de sus ventajas fundamentales es la portabilidad: está estandarizado y el mismo código fuente puede ser compilado en diversas plataformas. Adicionalmente a todo esto, es uno de los lenguajes más rápidos en cuanto a ejecución y actualmente hay una gran cantidad de compiladores, depuradores, librerías disponibles que facilitan aún más el trabajo de programar dado que se puede reducir el tamaño y complejidad del código. Al tratarse de un lenguaje compilado, presenta muy buena eficiencia en los tiempos de ejecución (imprescindible en los programas de procesamiento de imagen y visión por computador por la carga computacional de los algoritmos que se emplean)

El uso del entorno Visual simplifica la creación y compilación de código y, sobre todo, permite el tratamiento de los cuadros de diálogo e interfaces gráficas para aplicaciones de Windows de un modo sencillo. Se ha optado por aplicaciones de tipo MFC (*Microsoft Foundation Classes*).

Actualmente se dispone de una enorme cantidad de cámaras y lentes adecuados para capturar imágenes dependiendo de cada aplicación. Sin embargo, desde que en los años 90 se comenzaron a popularizar las cámaras de videoconferencia (webcam) se empezaron a obtener resultados muy interesantes. Comparables incluso a los de las cámaras CCD. Nuestro sistema captura la imagen directamente y en tiempo real con una webcam *Logitech Quickcam Connect* con resolución VGA de hasta 640x480 pixel. Se trata de una cámara barata, robusta, muy eficiente, y la forma más sencilla de obtener las imágenes que necesitamos para obtener un primer prototipo.

Cada 30ms aproximadamente se recibe una secuencia de la cámara. El sistema considera la imagen completa con todos sus canales de color y la convierte a una *imagen integral* (en apartados posteriores se explica su formación), sobre la que realiza la estimación de localización de una cara con una *plantilla Haar-Training de detección facial frontal*. Si se produce respuesta positiva por parte de esta función, el sistema analiza el área central del polígono cuadrado que rodea la zona facial y construye un histograma con la distribución de color que constituya la moda probabilística. Se realiza una reproyección de este histograma sobre la secuencia y se desestiman aquellas zonas que no coincidan con el valor de esta moda

probabilística de color. Es decir, puesto que el valor del histograma ha sido escogido del centro de una zona facial, todo aquello que no sea desestimado en este momento, corresponde a colores cercanos al color de la piel. El rastreo termina dibujando alrededor de estas zonas una elipse que marca el grado de inclinación de los ejes de máxima y mínima variación espacial.

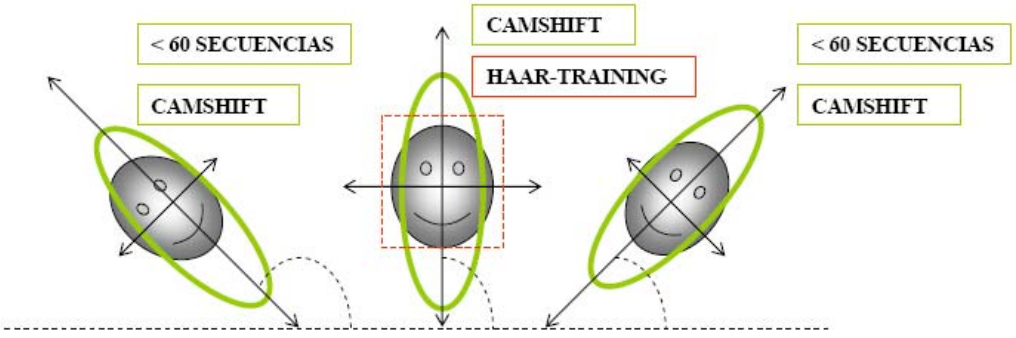


Fig. 40 El sistema detecta una cara frontal colocada en una posición vertical y sin giro, tal y como se muestra en la figura central. El objetivo es no perder la localización en los momentos de pérdida de detección. El algoritmo de detección se denomina HAAR-TRAINING y el de rastreo CAMSHIFT. El sistema tiene un umbral máximo de 60 secuencias de rastreo sin detección y una distancia entre centros detección que no supere la distancia de medio lado del cuadrado de detección facial.

Durante el ciclo normal de operación, el sistema detecta la cara y permite al algoritmo de rastreo continuar su trabajo durante un número determinado de secuencia y una distancia máxima permitida entre centros de detección.

Una situación de presencia facial es detectada por:

(1) La respuesta positiva de ambos algoritmos y una distancia entre centros menor que un umbral $U_{CENTROS}$.

(2) La respuesta positiva del algoritmo de rastreo y un número de secuencias de respuesta negativa del algoritmo de detección menor que un umbral $U_{PERDIDA}$.

Existen dos situaciones en que el sistema se sale de su ciclo normal de funcionamiento:

(1) Detección facial falsa, debido a que un determinado rango de sombras puedan dar lugar a confusión para la plantilla de detección.

(2) Cambio momentáneo de las condiciones de iluminación

3.4.2.2 Análisis perceptivo del sistema

Como se ha expuesto en apartados anteriores, los *referentes* son conceptos cuyas instancias dentro del entorno son representadas por la percepción, es decir, los que producen el estímulo. La percepción se incluye dentro de la estructura funcional del sistema. La *ley de representación* y el *referente*, que definen el proceso perceptivo, son los *objetivos* del sistema. Esto implica que las *representaciones* y los *cambios implícitos* derivados de la percepción mantienen una correspondencia con los *objetivos del sistema.* En la siguiente figura pueden verse estas relaciones de una forma gráfica:

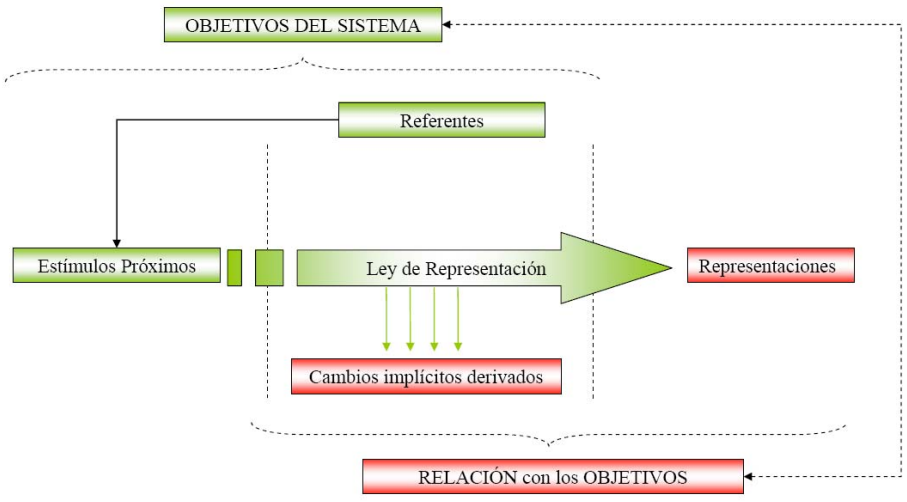


Fig. 41 Esquema del proceso perceptivo del modelo

Dicho de otra forma, el *referente* fija qué entidades pueden ser reconocidas en el *entorno perceptivo*, las relaciones que se espera que mantengan y las propiedades que las causan. La *Ley de representación* analizará el entorno acorde a estas

definiciones. Por lo tanto, un *referente* consiste en una serie de entidades conceptuales que pueden ser observadas en el *entorno perceptivo*: (1) *Quantities*, (2) relaciones entre *Quantities*, (3) propiedades que las describen.

$$V = \left\{ \rho^i, i = 1...n_r \right\}$$
$$i = referente\ "i"$$
$$n_r = N^o\ referentes$$
$$V = Sense\_referent$$

[Ec. 7]

Un *referente* particular "i" es percibido tras analizar una serie de *singularidades* que están asociadas a este referente. Formalmente se puede expresar de la siguiente forma:

$$\rho^i = \varepsilon_i(\Psi^i)$$
$$\Psi^i = \left\{ \psi_k^i, \quad k = 1...n_i \right\}$$
$$\Psi^i = Cjto.Singularidades$$
$$\varepsilon_i = Equivalencia\_cognitiva$$

[Ec. 8]

La información próxima está basada en los valores del conjunto de cantidades que forman el sistema sensorial (Q^{SS}).

$$Q^{SS} = \left\{ q_j^{SS}, \quad j = 1...n_{qSS} \right\}$$
$$q_j^{SS} = Quantitie \quad j$$
$$n_{qSS} = total\_quantities\_del\_sistema$$

[Ec. 9]

Un proceso de *ecualización* puede ser representado por una función eq_j y se refiere a la posible variación del dominio de actuación para la mejora en el procesamiento de la imagen.

Un ejemplo de proceso de ecualización puede ser la umbralización binaria de la imagen. De la misma manera, la ecualización se representa formalmente según la siguiente expresión:

$$eq_j : D_j^{SS} \rightarrow D_j^{SS*}$$
$$q_j^{SS} \rightarrow q_j^{SS*}$$

[Ec. 10]

La singularidad es el resultado de un cierto conjunto de valores del sistema sensorial que indicaremos Q_K^i:

$$Q_K^i \xrightarrow{\sigma_K^i} \psi_K^i$$
$$Q_K^i \subset Q^{SS}$$
$$Q^i = \bigcup_{k=1...n_i} Q_K^i$$
$$Q_K^i = Cjto\_cantidades\_de\_Singularidad\_K(referente\_i)$$
$$\sigma_K^i = Equivalencia\_Singularidades(Singularidad\_k\_refenrente\_i)$$
$$Q^i = referente(i)\_sistema\_sensorial$$

[Ec. 11]

Los conjuntos de singularidades del sistema de detección facial que se van a definir en este supuesto son:

$$Conjunto\ \ \Psi^1 = región\_cuadrada\_HAAR-TRAINING(\exists)$$
$$Conjunto\ \ \Psi^2 = región\_elipse\_CAMSHIFT(\exists)$$
$$Conjunto\ \ \Psi^3 = UMBRAL\_CENTROS$$
$$Conjunto\ \ \Psi^4 = UMBRAL\_FRAMES$$
$$Conjunto\ \ \Psi^5 = UMBRAL\_BRILLOS$$
$$Conjunto\ \ \Psi^6 = UMBRAL\_EXISTENCIA(Ausencia\_Cara)$$

[Ec. 12]

Estos conjuntos, están formados por una serie de singularidades y referentes que los conforman. Se han definido de la siguiente forma:

$$
\begin{aligned}
&Conjunto\ \ \Psi^{1}\\
&\quad Singularidad\,\psi_{1}^{1} = Plantilla\_Positiva\\
&\quad Singularidad\,\psi_{2}^{1} = Lado\_mínimo\\
&\quad Singularidad\,\psi_{3}^{1} = Haar\_Token\\
&Conjunto\ \ \Psi^{2}\\
&\quad Singularidad\,\psi_{1}^{2} = Histograma\_Construido\\
&\quad Singularidad\,\psi_{2}^{2} = Area\_mínima\\
&\quad Singularidad\,\psi_{3}^{2} = CamShift\_Token\\
&Conjunto\ \ \Psi^{3}\\
&\quad Singularidad\,\psi_{1}^{3} = Centro\_Cuadrado\\
&\quad Singularidad\,\psi_{2}^{3} = Centro\_Elipse\\
&Conjunto\ \ \Psi^{4}\\
&\quad Singularidad\,\psi_{1}^{4} = Plantilla\_Negativa\\
&\quad Singularidad\,\psi_{2}^{4} = Número\_\ frames\\
&Conjunto\ \ \Psi^{5} = Umbral\_Brillo\\
&\quad Singularidad\,\psi_{1}^{5} = N^{o}\ pixel\ /\ RGB > 200\ (muy\_claro)\\
&\quad Singularidad\,\psi_{2}^{5} = N^{o}\ pixel\ /\ RGB < 25\ (muy\_obscuro)\\
&\quad Singularidad\,\psi_{3}^{5} = 25 < (N^{a}\ pixel\_RGB) < 200\ (normal)\\
&Conjunto\ \ \Psi^{6}\\
&\quad Singularidad\,\psi_{1}^{6} = Plantilla\_Negativa\\
&\quad Singularidad\,\psi_{2}^{6} = Elipse\_Negativa
\end{aligned}
$$

[Ec. 13]

$$
\begin{aligned}
&referente\ \ \rho^{1} : Detección\_cara\\
&referente\ \ \rho^{2} : Rastreo\_cara\\
&referente\ \ \rho^{3} : Umbral\_Centros\\
&referente\ \ \rho^{4} : Umbral\_Tiempos\\
&referente\ \ \rho^{5} : Umbral\_Brillo\\
&referente\ \ \rho^{6} : Umbral\_Existencia
\end{aligned}
$$

[Ec. 14]

Las *funciones de equivalencia* entre *singularidades* establecen la correspondencia entre las *cantidades* y *singularidades* del sistema sensorial. Tal y como se propone en el modelo original, los nombres de las *singularidades* han sido elegidos para representar sus correspondientes *funciones de equivalencia*.

En este sistema las funciones examinan el valor de salida de dos funciones algorítmicas, de dos umbrales de detección y del número de píxeles que contienen valores de intensidad relativamente altos.

$$\textit{Dependencia}\ \ \varepsilon_1^1\left(\psi_1^1,\psi_2^1\right)\ \ y\ \ \varepsilon_2^1\left(\psi_1^1,\psi_2^1,\psi_1^2,\rho^3,\rho^5\right)$$

$$\rho^1 = \textit{Detección\_cara} \Leftrightarrow \begin{cases} \varepsilon_1^1 = \psi_1^1 \wedge \psi_2^1 \\ \varepsilon_2^1 = \psi_1^1 \wedge \psi_2^1 \wedge \psi_1^2 \wedge \psi_2^2 \wedge \rho^3 \wedge \rho^5 \end{cases}$$

$$\textit{Dependencia}\ \ \varepsilon_1^2\left(\varepsilon_1^2\right)\ \ y\ \ \varepsilon_2^2\left(\psi_1^1,\psi_2^1,\psi_1^2,\psi_2^2,\rho^4,\rho^5\right)$$

$$\rho^2 = \textit{Rastreo\_cara} \Leftrightarrow \begin{cases} \varepsilon_1^2 = \varepsilon_2^1 \\ \varepsilon_2^2 = \psi_3^1 \wedge \psi_1^2 \wedge \psi_2^2 \wedge \rho^4 \wedge \rho^5 \end{cases}$$

$$\textit{Dependencia}\ \ \varepsilon_1^3\left(\psi_2^1,\psi_1^3,\psi_2^3\right)$$

$$\rho^3 = \textit{Cumple\_Umbral\_Centros} \Leftrightarrow \left\|\psi_1^3 - \psi_2^3\right\| \le \left(\psi_2^1 / 2\right)$$

$$\textit{Dependencia}\ \ \varepsilon_1^4\left(\psi_1^4,\psi_2^4\right)$$

$$\rho^4 = \textit{Cumple\_Umbral\_Tiempos} \Leftrightarrow \begin{cases} \psi_1^4 = \neg\,\psi_1^1 \\ \psi_2^4 < 60 \end{cases}$$

$$\textit{Dependencia}\ \ \varepsilon_1^5\left(\psi_3^5\right)$$

$$\rho^5 = \textit{Cumple\_Umbral\_brillo} \Leftrightarrow \psi_3^5 > \left(\psi_1^5 + \psi_2^5\right)$$

$$\textit{Dependencia}\ \ \varepsilon_1^6\left(\psi_1^6,\psi_2^6\right)$$

$$\rho^6 = \textit{Ausencia\_cara} \Leftrightarrow \begin{cases} \psi_1^6 \wedge \psi_2^6 \rightarrow \begin{cases} \psi_1^6 = \neg\,\psi_3^1 \\ \psi_2^6 = \neg\,\psi_3^2 \end{cases} \\ \rho^1 \wedge \rho^2 \wedge \rho^5 \end{cases}$$

[Ec. 15]

Determinamos que el cumplimiento de los umbrales supone estar en el rango de valores en que la detección es posible.

(1) Umbral de centros: la distancia Euclídea entre centros es menor que el umbral marcado.

(2) Umbral de tiempos: el número de secuencias es menor que el umbral marcado

(3) Umbral de brillos: el brillo de la imagen no es demasiado claro ni demasiado oscuro. Está dentro de los umbrales máximo y mínimo marcados.

(4) Umbral de ausencia de caras: Los dos testigos están libres. Esto supone que no han sido activados y que, como consecuencia, no hay caras.

El mapa perceptual es el que se muestra en la siguiente figura:

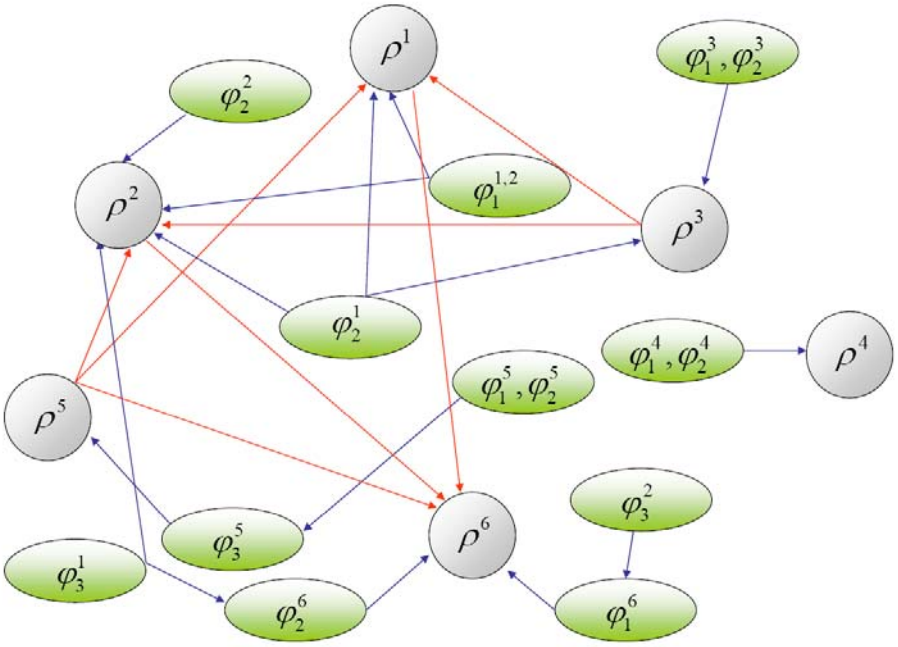


Fig. 42 Mapa perceptual del sistema de detección facial

Este mapa muestra gráficamente en color azul las dependencias entre singularidades, y las relaciones entre singularidad y referente, y en color naranja las dependencias entre referentes.

CAPÍTULO 4

Identificación Facial con Visión

En los años sesenta surgen los primeros algoritmos para la detección de caras. Estos algoritmos estaban basados en técnicas heurísticas y antropométricas. Sin embargo fallaban muy a menudo y eran muy sensibles a cualquier tipo de cambio (fondo, raza, gafas, etc.). Los desarrollos en esta área se abandonaron por falta de aplicación. Los años noventa es el momento en el que la tecnología experimentó una gran evolución debida principalmente al interés por las videoconferencias, reproducción de vídeos, aplicaciones de seguridad con cámaras y aplicaciones interactivas hombre-máquina. Y se recuperó este interés por identificar a las personas a través de la imagen.

No obstante, aún con los avances tecnológicos conseguidos en procesamiento, la detección de caras en una imagen puede llevar desde unas milésimas hasta varios segundos, dependiendo de la fiabilidad, precisión y robustez del algoritmo, del tamaño de la imagen y factores de capacidad hardware del sistema.

El primer paso para detectar una cara por medio de técnicas de visión por computador es su localización dentro de la imagen. Es necesario distinguir en primer lugar qué zonas de la escena pertenecen al entorno (lo que se llama generalmente fondo o background) y qué parte es zona de interés.

La decisión de la técnica o algoritmo a emplear dependerá fundamentalmente de la naturaleza de los datos de partida, y del tiempo de procesado máximo del sistema.

La exigencia de funcionamiento en tiempo real limitará inevitablemente este tiempo máximo de procesamiento, así como los algoritmos y filtros a emplear.

En este epígrafe se muestran las opciones que, una vez estudiadas como parte del estado del arte, se han considerado coherentes y válidas para realizar una implementación. Posteriormente se han mejorado aquellos factores que inicialmente no ofrecían los resultados esperados para las condiciones de trabajo a las que se va a someter al sistema. Todo ello queda debidamente explicado en los siguientes apartados siguiendo la siguiente estructura:

- Reconocimiento facial genérico para la localización de una cara en la imagen
    - Detección de caras por el algoritmo de P. Viola[1] y M. Jones[2] y las Plantillas Haar-Training
    - Detección de caras con Momentos Espaciales y el algoritmo CAMSHIFT[53]
    - Propuesta de un algoritmo de trabajo conjunto y redundante con el fin de evitar las posibles pérdidas en la detección.
- Identificación facial de la persona que está siendo localizada si se encuentra dentro de un grupo de personas conocido.
    - Método de identificación de caras mediante el Análisis de Componentes Principales (PCA) [54][55]
    - Optimización en la elección del conjunto de imágenes de entrenamiento.

Asimismo se reflejan los resultados alcanzados con cada uno de ellos y una estimación de su comportamiento ante cambios de iluminación, posición y movimiento.

[1] Microsoft Research, One Microsoft Way, Redmond, WA 98052,USA
[2] Mitsubishi Electric Research Laboratory, 201 Broadway, Cambridge, MA 02139, USA

4.1 Reconocimiento facial

En este apartado se exponen los fundamentos teóricos y los resultados de la implementación realizada para localizar una cara en una escena. Esta localización facial supone la base para el sistema que se va a implementar posteriormente en el robot.

La interacción entre el robot y el usuario estará guiada por una identificación inicial de su tutor y de las personas habituales de su entorno. A partir de esta identificación, se van a implementar técnicas de reconocimiento de gestos para recepción de pequeñas órdenes de movimiento, aprendizaje de gestos manuales para su propia expresión en discursos y presentaciones y aprendizaje de nuevas personas en caso de no ser reconocidas. Si falla la localización inicial de la cara dentro de la imagen, fallan también en cadena todas las capacidades posteriores.

Por lo tanto, es fundamental que la capacidad de reconocer y localizar una cara en el sistema sea, además de rápido (para su correcto funcionamiento en tiempo real), fiable. Capaz de obtener unos rangos de precisión muy altos.

Es por ello que se han propuesto dos métodos trabajando de forma redundante. El primero de ellos se basa en una búsqueda sin dependencia de rasgos de color, dada la varianza de brillo que se produce en las sucesivas imágenes captadas en tiempo real. El segundo, por el contrario, basa su búsqueda en estos rasgos de color, pero realizando la búsqueda dentro del entorno de detección del primero. De este modo, proporcionamos un método dinámico de adquirir las características iniciales para el funcionamiento del segundo algoritmo, sin necesidad de aportar unos datos estáticos que, evidentemente y tratándose de un sistema de visión, van a cambiar dependiendo de la luz del entorno donde esté funcionando el sistema.

Según la literatura, el método de detección de caras con plantillas Haar-Training es uno de los más importantes y efectivos para tiempo real y vídeo. De hecho, la mayor parte de las cámaras de fotos, webcam y demás sistemas de detección de caras que se están vendiendo en el mercado en la actualidad, llevan implementado este algoritmo.

MODOS DE OPERACIÓN EN RECONOCIMIENTO FACIAL

De un sistema de reconocimiento facial se espera que sea capaz de identificar caras presentes en imágenes de forma automática. Tienen dos modos de operación: (1) autenticación facial (detección o localización), y (2) identificación (reconocimiento de identidad).

El primer caso requiere un estudio de similitud punto por punto. Una comparación entre la nueva imagen y un patrón o plantilla genéricos de aquella que se quiere identificar. En el segundo caso se requiere un examen de semejanza con un conjunto de plantillas incluidas en una base de datos para conocer a cual de ellos corresponde la nueva cara.

PROCESAMIENTO PARA EL RECONOCIMIENTO FACIAL

Se trata ante todo de un problema de reconocimiento de patrones. La cara es un ítem tridimensional sujeto a variaciones de iluminación, postura y expresión, y debe ser identificada con imágenes de dos dimensiones (que son las que nos proporcionan actualmente los equipos de captura de imagen).

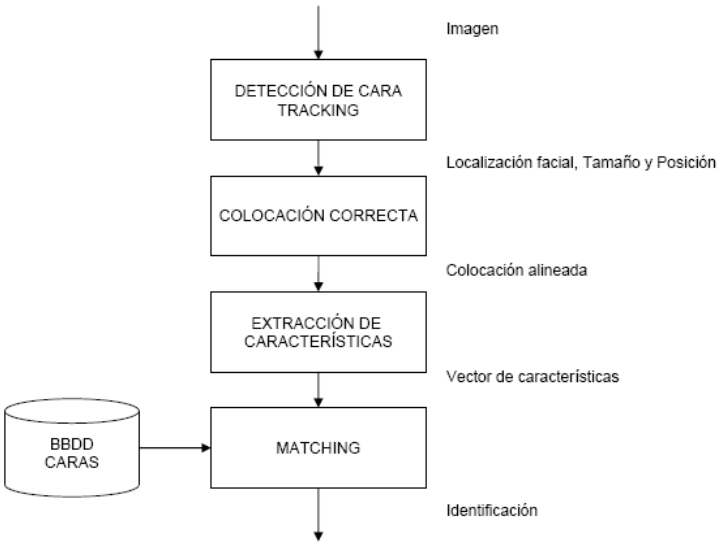


Fig. 43 Diagrama de flujo genérico de un sistema de reconocimiento facial

Un sistema de reconocimiento facial genérico se puede estructurar en cuatro módulos: detección, colocación, extracción de características y estudio de similitudes punto a punto[3]. Los dos primeros suponen una adaptación y preparación previa del sistema, siendo el reconocimiento propiamente dicho el que realizan los dos segundos módulos.

La detección consiste en separar las áreas continentes de caras del resto de la imagen. En el caso de vídeo puede ser necesario además un seguimiento, siendo

[3] El término correcto -por ser el que se usa siempre- sería el anglicismo “matching”.

necesario un componente algorítmico adicional que realice esta tarea. El módulo de colocación no tiene por qué estar ceñido a una posición fija. Es un concepto de normalización de posición, tamaño, iluminación y escalas de gris para dar la posibilidad de una comparación fidedigna.

Posteriormente la fase de extracción de características provee la posibilidad de hacer efectiva la información que es apropiada para distinguir entre caras de diferentes personas teniendo en cuenta las continuas variaciones fotométricas y geométricas de la imagen. Finalmente, la comparación de similitudes depende del vector de características extraído en el módulo anterior. Es el ítem que se compara por ser de menor dimensión que la imagen en sí misma. Los elementos incluidos en la base de datos tendrán el mismo formato, indispensable para comprobar la semejanza.

4.1.1 Detección de Caras con Plantillas Haar-Training

Se trata de un algoritmo de detección de caras capaz de procesar las imágenes extremadamente rápido, alcanzando a la vez rangos de precisión muy altos.

Este desarrollo tiene tres contribuciones clave: (1) La introducción de una nueva representación de la imagen, llamada *"Imagen Integral"*, para disminuir la carga computacional en el procesamiento de la imagen, (2) un clasificador simple y eficaz, construido a partir del algoritmo *AdaBoost* [56], en el que se seleccionan un pequeño número de características críticas de la imagen de un conjunto muy amplio de ellas y, (3) en último lugar, un método de combinación de clasificadores en cascada, que permite ir discriminando regiones de la escena donde se tiene la seguridad de no encontrar estas características, dejando únicamente el procesamiento fuerte en las pequeñas regiones donde es más seguro que se van a encontrar.

4.1.2 Características base

La razón de utilizar características y no píxeles directamente es que los sistemas basados en características operan mucho más rápido que los sistemas basados en el procesamiento píxel por píxel.

En concreto, las características que usa este procedimiento, son una reminiscencia de las *Funciones base Haar* usadas por Papageorgiou et al. (1998). Más específicamente, se usan tres:

- El valor de la característica *rectángulo-doble* como diferencia entre la suma de píxeles entre dos regiones rectangulares consecutivas. Las regiones tienen el mismo tamaño y forma y son vertical u horizontalmente adyacentes (ver Fig. 44)

- La característica *triple-rectángulo* que se calcula como la suma de los píxeles de las dos regiones extremas y la diferencia de este resultado con la suma del valor de los píxeles de la región central.

- La característica *cuatro rectángulos* que será la diferencia entre los valores de los píxeles de las dos regiones diagonales.

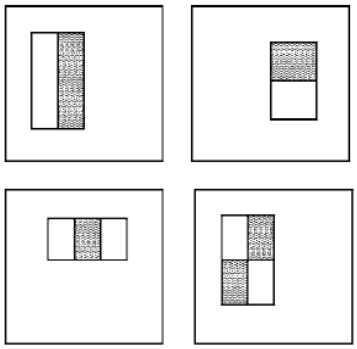


Fig. 44 Ejemplo de las características descritas para aproximar la ventana de detección. Las dos superiores son *"características de doble-rectángulo"* y las de la parte inferior de *"triple"* y *"cuatro-rectángulos"*

4.1.2.1 Imagen Integral

Las características descritas anteriormente pueden ser procesadas muy rápidamente usando una representación intermedia llamada *"Imagen Integral"*. En el punto (x,y), la imagen integral contiene la suma de los píxeles por encima y por la derecha del mismo (éste inclusive):

$$ii(x,y) = \sum_{x' \le x, y' \le y} i(x',y') \xrightarrow{donde} ii(x,y) \equiv IMAGEN\ INTEGRAL$$

[Ec. 16]

Usando el siguiente par de recurrencias podemos calcular la imagen integral en un único paso a partir de la original:

$$\begin{aligned} s(x,y) &= s(x,y-1) + i(x,y) \\ ii(x,y) &= ii(x-1,y) + s(x,y) \\ s(x,-1) &= 0 \\ ii(-1,y) &= 0 \end{aligned}$$

[Ec. 17]

Usando esta imagen integral, cualquier suma rectangular se puede calcular en cuatro referencias y, como consecuencia, la diferencia entre dos áreas rectangulares en ocho referencias (ver Fig. 45). Si son adyacentes serán seis..

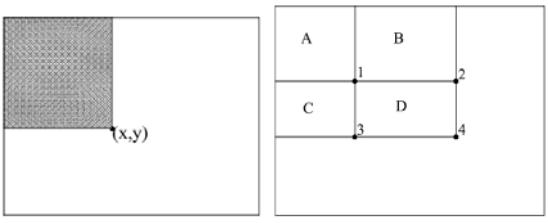


Fig. 45 El valor de la imagen integral en el punto [x y] es la suma de todos los píxeles por encima y a la izquierda de él mismo (éste incluido).

En la imagen de la derecha se muestra un ejemplo de cómo la suma de píxeles dentro del rectángulo D puede ser calculada con cuatro referencias. El valor de la imagen integral en la localización 1 es la suma de los píxeles del rectángulo A. En el punto 2, el valor de A+B, en la localización 3 es A+C y en la cuatro A+B+C+D.

Entonces, la suma en D puede ser calculada como 4+1-(2+3). Una alternativa a la idea de imagen integral pueden ser los boxlets, trabajo propuesto por Simard et al. 1999.

En [53] se puede ver la discusión correspondiente del procedimiento que se muestra, así como la justificación de la rapidez que ofrece el uso de las características frente a cualquier reductor piramidal.

4.1.2.2 Aprendizaje y entrenamiento de las Funciones Clasificadoras

Dado un conjunto de características y un conjunto de imágenes (positivas y negativas)[4] de entrenamiento, se puede emplear cualquier algoritmo de aprendizaje para obtener una *Función Clasificadora*. Sung y Poggio usan una especie de modelo Gaussiano (Sun and Poggio, 1998). Rowley et al. (1998) un pequeño conjunto de simples características de la imagen y una red neuronal. Más recientemente Roth et al. (2000) han propuesto un método inusual de representación de la imagen y usan lo que denominan *Winnow learning procedure* (por traducirlo de alguna forma, un procedimiento de aprendizaje por reducción).

En general, las propuestas para realizar el entrenamiento y calcular las funciones adecuadas para clasificar son muchas y dependen de cada autor. En su forma original, el algoritmo de aprendizaje AdaBoost es usado para desarrollar otro algoritmo simple (por ejemplo, puede ser usado para desarrollar un simple perceptrón). Es decir, combinando una colección de *clasificadores débiles* (llamados así por su bajo coste computacional a la hora de procesarlos, dado que analizan un número pequeño de condiciones) para formar un *clasificador fuerte* (que por ende, se trata de un filtro más potente y con mayor carga computacional al procesarlo)[5]. El clasificador fuerte final toma la forma de un perceptrón, una combinación ponderada de clasificadores débiles seguido de un umbral.

AdaBoost es un mecanismo agresivo de selección para encontrar un número reducido de *Funciones Clasificadoras* que, sin embargo, tiene una variedad

[4] Se entiende imagen positiva, aquella que tiene la característica buscada entre otras. Análogamente, la imagen negativa será aquella que no contenga dicha característica.
[5] Por ejemplo, un algoritmo de aprendizaje para un perceptrón, realiza la búsqueda entre todos los posibles y devuelve aquel con menor error de clasificación (ya que la evaluación a que se ha sometido es la de clasificar un determinado item)

significativa. Para cada característica, el entrenador débil determina una función clasificadora *"umbral óptimo"* como aquella con la que se han desclasificado un menor número de muestras.

$$\begin{aligned} h(x, f, p, \theta) &\rightarrow Clasificador\ debil \\ f &\rightarrow Caracteristica \\ \theta &\rightarrow Umbral \\ p &\rightarrow Polaridad \end{aligned}$$

$$h(x, f, p, \theta) = \begin{cases} 1 & si \quad pf(x) < p\theta \\ 0 & otros \end{cases}$$

[Ec. 18]

La polaridad indica la dirección de desigualdad. Y en este contexto, la x es una imagen de 24x24 píxeles.

El algoritmo de aprendizaje es el siguiente:

- Dado un conjunto de imágenes: $(x_1 , y_1)\ldots (x_n , y_n)$ donde $y_i=0/1$ dependiendo de si es una muestra negativa o positiva respectivamente.

- Se inicializan los pesos:

$$\begin{cases} \omega_{1,i} = \dfrac{1}{2m} \quad para \quad y = 0 \rightarrow m = N^{o}\, muestras\, negativas \\ \omega_{1,i} = \dfrac{1}{2l} \quad para \quad y = 1 \rightarrow l = N^{o}\, muestras\, positivas \end{cases}$$

[Ec. 19]

- Suponemos un conjunto de T hipótesis para el aprendizaje. Entonces, para cada hipótesis t, siento t =1…T, se realizan los siguientes pasos:

  (1) Se normalizan los pesos:

$$\left\{ \omega_{t,i} \leftarrow \frac{\omega_{t,i}}{\sum_{j=1}^{n} \omega_{t,j}} \right.$$

[Ec. 20]

(2) Se selecciona el mejor clasificador débil con respecto al error ponderado:

$$\varepsilon_t = \min_{f,p,\theta} \sum_i \omega_i \left| h(x_i, f, p, \theta) - y_i \right|$$

[Ec. 21]

(3) Se actualizan los pesos:

$$\omega_{t+1,i} = \omega_{t,i} \bullet \beta_t^{1-e_i}$$

$$ei = 0 \quad si \quad xi \rightarrow clasificado \quad correctamente$$

$$ei = 1 \quad si \quad xi \rightarrow clasificado \quad incorrectamente$$

$$\beta_t = \frac{\varepsilon_t}{1-\varepsilon_t}$$

[Ec. 22]

(4) Se obtiene el clasificador final:

$$\begin{cases} 1 & \sum_{t=1}^{T} \alpha_t h_t(x) \geq \frac{1}{2} \sum_{t=1}^{T} \alpha_t \\ 0 & otros\ casos \end{cases}$$

[Ec. 23]

$$\alpha_t = \log \frac{1}{\beta_t}$$

[Ec. 24]

En la siguiente figura se pueden ver dos posibles características seleccionadas por el AdaBoost (en la figura aparecen solas, y en la parte inferior superpuestas en las imágenes). La primera de las dos características evalúa la intensidad entre la región de los ojos y la parte superior de las mejillas (basándose en el hecho de que las regiones oculares son generalmente más oscuras que las mejillas). La segunda compara las intensidades entre las zonas oculares y el puente de la nariz.

Fig. 46 Ejemplo de dos características resultado del AdaBoost. Se trata de dos clasificadores, de dos y tres características (izquierda y derecha superior respectivamente)

4.1.2.3 Construcción de la cascada de clasificadores

Una vez que tenemos los clasificadores, tenemos que ir evaluando la imagen con ellos, desde el más débil al más fuerte. El reto ahora es un método eficaz y que tenga el menor coste computacional que sea posible.

La idea es no procesar toda la imagen con todos los clasificadores, sino ir desestimando zonas de la misma e ir disminuyendo el área de búsqueda. Esto supone un primer ahorro computacional. En segundo lugar, la idea es que los clasificadores débiles (que requieren un menor coste computacional) sean utilizados para procesar la mayor parte de los sectores grandes de la imagen. Y los clasificadores fuertes, cuya evaluación va a invertir más recursos, sean pasados por las regiones (más pequeñas) que resulten del proceso anterior. Estas regiones son denominadas *"ventanas"* o *"subventanas"*.

Así que el objetivo es colocar adecuadamente el orden en el que se van a utilizar estos clasificadores para optimizar –ya en un tercer proceso- al máximo la búsqueda. Y esta colocación será una secuencia o cascada que deberá ser óptima para la eficiencia en precisión y tiempo de procesado.

En esta sección se describe el algoritmo que proponen los autores para construir esta cascada de clasificadores (construidos con el AdaBoost). La justificación de la estructura en cascada tiene que ver con el hecho de que dentro de una imagen, una amplia mayoría de subventanas son negativas. La cascada intenta rechazar el máximo número de negativos en el menor tiempo posible.

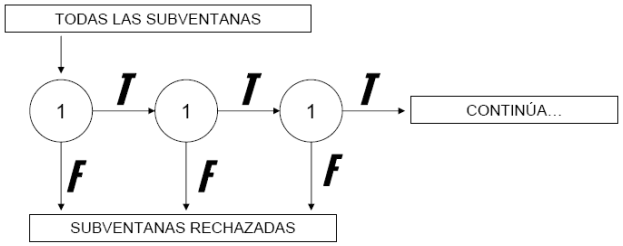


Fig. 47 Descripción de un detector en cascada. El primer clasificador de la serie, elimina un número muy amplio de negativos con muy poco procesamiento. Los siguientes niveles eliminan negativos adicionales, pero necesitan un mayor coste computacional. Tras realizar este proceso varias veces, el número de subventanas ha disminuido radicalmente, a medida que va aumentando el gasto de memoria de procesamiento.

Como aclaración adicional, rechazar negativos se entiende eliminar zonas de la imagen donde no vamos a encontrar la característica buscada.

ENTRENAMIENTO DE LOS CLASIFICADORES EN CASCADA

El proceso de diseño de la secuencia está dirigido por un conjunto de objetivos de detección y eficiencia. Es decir, se deben conseguir ratios de detección entre el 85% y el 90%, disminuyendo al máximo los falsos positivos y el tiempo de procesamiento.

Suponiendo que tenemos entrenada una cascada de clasificadores, con un ratio de falsos positivos de:

$$F = \prod_{i=1}^{K} f_i$$

$$F \rightarrow Ratio\ \ falsos\ \ positivos\ de\ la\ cascada$$

$$K \rightarrow N^{o}\ clasificadores$$

$$f_i \rightarrow Falso\ \ positivo\ del\ (i-th)\ clasificador$$

[Ec. 25]

El ratio de detección es:

$$D = \prod_{i=1}^{K} d_i$$

$$D \rightarrow \textit{Ratio Detección de la cascada}$$

$$K \rightarrow \textit{Nº clasificadores}$$

$$d_i \rightarrow \textit{Detección del } (i-th) \textit{ clasificador}$$

[Ec. 26]

Dados unos objetivos concretos de ratios (de falsos positivos y de detección), los ratios objetivo pueden ser determinados para cada etapa en el proceso de la cascada.

Por ejemplo, un ratio de detección de 0.9 puede ser logrado por un clasificador de la etapa 10, si cada etapa ha tenido un ratio de detección de 0.99 (dado que $0.9 \sim 0.99^{10}$). El factor clave de medida de cada clasificador es su *"ratio positivo"*, la proporción de ventanas que han sido etiquetadas como potenciales contenedores de una cara. El número de características que se espera evaluar es:

$$N = n_o + \sum_{i=1}^{K} \left(n_i \prod_{j<i} p_j \right)$$

$$N \rightarrow \textit{Características que se espera evaluar}$$

$$K \rightarrow \textit{Nº clasificadores}$$

$$p_i \rightarrow \textit{Ratio positivo del } (i-th) \textit{ clasificador}$$

$$ni \rightarrow \textit{Nº características en el } (i-th) \textit{ clasificador}$$

[Ec. 27]

El proceso debe ser realizado con cuidado: el aprendizaje con AdaBoost está dirigido únicamente a minimizar errores y no precisamente para alcanzar ratios de detección específicos. Un esquema tradicional y simple de intercambiar estos errores es ajustar el umbral del perceptrón a que dio lugar el AdaBoost. Los umbrales muy altos conllevan mayor número de falsos positivos y sin embargo, la contrapartida es que también aumentan los rangos de detección.

En la práctica se una un simple marco para dar lugar a un clasificador efectivo y con alta eficacia. El usuario selecciona el máximo aceptable para los falsos positivos y las detecciones de calda clasificador (f_i y d_i). Cada nivel de la cascada se entrena con el AdaBoost hasta que dicho nivel alcanza los ratios fijados (cuyo valor se ha

determinado previamente por test). Si no se han alcanzado todos, se añade otro nivel.

Los resultados, las conclusiones y los estudios realizados acerca de lo que aquí se ha expuesto se pueden ver en [57].

4.1.2.4 Detección de caras con OpenCV

El detector de objetos que se ha expuesto previamente es una de las capacidades que vienen incluidas dentro de las librerías OpenCV usadas.

Recordamos que el clasificador (*Cascada de clasificadores boost trabajando con características Haar*) tiene que estar entrenado con unos pocos de cientos de muestras (imágenes) del objeto que se desea buscar posteriormente en la escena. Por ejemplo una cara. Estos son las llamadas muestras positivas. Cuando estos clasificadores son entrenados, pueden ser aplicados a las regiones de interés para localizar el lugar de la escena en que se encuentra el objeto buscado.

OpenCv tiene entrenadas algunas de estas plantillas para detectar caras frontalmente y, adicionalmente, tres plantillas para detección del cuerpo (completo, parte superior y parte inferior). Es pues interesante el uso de esta característica para evitar realizar un trabajo que ya está hecho. Están localizadas en *OpenCV/data/haarcascades/*.

4.1.3 Detección de caras con CAMSHIFT

Tenemos localizada la cara dentro de la escena de la imagen con un muy buen resultado. Hasta este punto todo parece funcionar correctamente y, en principio, podríamos permitir que el sistema siguiera creciendo con esta primera base implementada.

Sin embargo, hemos realizado esta implementación a partir de unas plantillas que realizan la búsqueda de un determinado objeto y en una determinada posición. Es decir, en nuestro caso, realizamos la búsqueda frontal de una cara. No vamos a entrar en la posibilidad de que detecte una persona cuando la realidad corresponde a cualquier elemento material, juguete o dibujo que, sin embargo, al tener apariencia de cara, es detectado como tal. Pero sí en el que se expone a continuación.

Si Urbano, nuestro robot social, interactúa con *alguien que ve* y le impedimos (es nuestra decisión como diseñadores en este supuesto) que *obedezca* a nadie que no sea su tutor, nos podemos encontrar con el siguiente caso: En un momento de nuestro discurso y, al gesticular, movemos la cabeza de un lado a otro. Esto provoca que el sistema no detecte la cara puesto que alguna de las características que busca la plantilla, pueden no estar visibles en ese momento. Probablemente en nuestro giro, ocultemos a la cámara un ojo, la nariz, o las propias distancias que intenta comprobar el patrón entrenado, lo que provoca una respuesta con un falso negativo. En ese momento Urbano *cree* que no hay nadie, e interrumpe su atención hacia nosotros.

Esto, que en principio es únicamente una de las pequeñas limitaciones del algoritmo *Haar-training*, puede suponer un problema en la finalidad que intentamos dar a nuestra implementación.

Es pues necesario que tengamos un algoritmo redundante, que permita al sistema tener una seguridad adicional de detección cuando el primero falle. No vamos a conseguir un resultado perfecto con el 100% de acierto y 0% de error. Sin embargo, si aportamos un seguimiento de la característica encontrada durante un determinado rango de tiempo, a partir de otras particularidades que no dependan de la posición, impediremos la pérdida instantánea de la detección de la cara por este motivo. El sistema será “ciego” al fallo del primer algoritmo y Urbano seguirá realizando su tarea con normalidad.

Con esta premisa, buscamos una característica que no dependa de la posición de la cara, puesto que es el punto de mal funcionamiento que queremos evitar. Una particularidad que tiene esta independencia a la posición que necesitamos, es el color de la piel. Por este motivo se decidió su utilización para el algoritmo redundante.

Se va a realizar un seguimiento de la cara a partir del color de la piel. Como en el caso anterior, necesitamos algoritmos rápidos y eficientes, capaces de ser procesados en tiempo real, y sin consumir demasiados recursos de sistema.

El algoritmo que se describe en los siguientes epígrafes, es un método robusto y no paramétrico, que va incrementando el gradiente de densidad hasta encontrar la *moda probabilística* dominante. Este algoritmo se denomina *MeanShift*.

MeanShift [58] es un algoritmo que opera sobre distribuciones de probabilidad. En [59], Comaniciu afirma que, para rastrear objetos coloreados en tiempo real, estos colores deben estar representados como distribuciones de probabilidad (histogramas de color en nuestro caso).

En una secuencia de imágenes en tiempo real, el color y su distribución cambian continuamente, por lo que el algoritmo *MeanShift* deberá ser cambiado dinámicamente en relación a la distribución que se esté rastreando. El algoritmo resultante es el llamado *Camshift (Continously Adaptative Mean Shift*) [53].

Está basado en técnicas de *estadística robusta* y *distribuciones de probabilidad*. La Estadística Robusta es aquella que ignora cualquier dato que esté lejos de una región de interés. Y esto supone una ayuda para la compensación de ruido.

4.1.3.1 Descripción del algoritmo MEANSHIFT

Es el algoritmo del cual deriva *CamShift*. Como ya se ha dicho anteriormente, se trata de una técnica no paramétrica que incrementa el gradiente de una distribución de probabilidad para encontrar la moda probabilística dominante más cercana. El modo de operación que sigue es el siguiente:

- Elección de un tamaño de ventana de búsqueda.
- Elección de la localización inicial de dicha ventana.
- Cálculo de la localización de la media dentro de la ventana de búsqueda.
- Centrado de la ventana en ese punto.
- Iteración de los pasos c.) y d.) hasta que el movimiento para realizar el centrado no sea mayor que un umbral definido: existencia de convergencia.

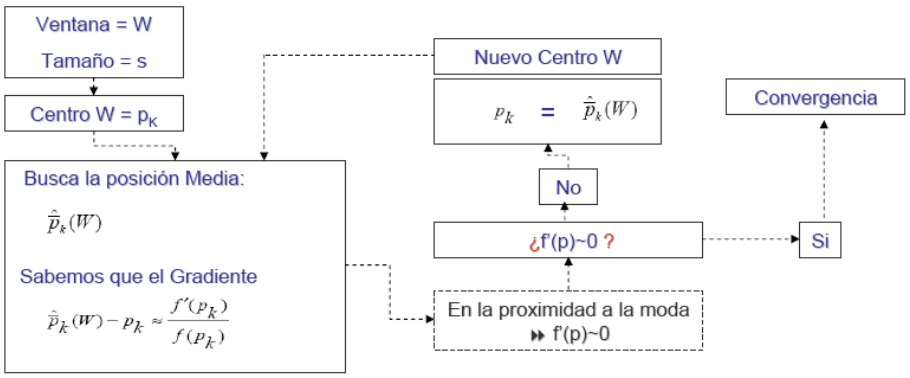


Fig. 48 Esquema de búsqueda y convergencia del algoritmo *MeanShift*

Asumiendo un espacio de distribución euclídea que contiene la distribución f, la convergencia queda demostrada según sigue:

(1) Se toma un tamaño *s* de ventana *W*

(2) La ventana inicial estará centrada en el punto p_k

(3) Se calcula la localización media dentro de la ventana:

$$\hat{\bar{p}}_k(W) = \frac{1}{|W|}\sum_{j \in W} p_j$$

[Ec. 28]

(4) El gradiente cambia según:

$$\hat{\bar{p}}_k(W) - p_k = \frac{f'(p_k)}{f(p_k)}$$

[Ec. 29]

(5) Centra la ventana en $\hat{\bar{p}}_k(W)$

(6) Se repiten las dos operaciones anteriores hasta que *f'(p)~0*, es decir, hasta conseguir la convergencia.

En procesamiento 2D, la localización media, también llamada *centroide* o *centro de gravedad* se calcula de la siguiente manera:

(1) El momento de orden cero[6]: $M_{00} = \sum_x \sum_y I(x, y)$, donde *I(x,y)* es el valor probabilística del píxel *(x,y)*

(2) El momento de orden uno para x e y: $\begin{cases} M_{10} = \sum_x \sum_y x \cdot I(x, y) \\ M_{01} = \sum_x \sum_y y \cdot I(x, y) \end{cases}$

(3) El centroide será: $\begin{cases} x_c = \dfrac{M_{10}}{M_{00}} \\ y_c = \dfrac{M_{01}}{M_{00}} \end{cases}$

Sin embargo *MeanShift* está diseñado para distribuciones estáticas, y no resulta válido para nuestras pretensiones de trabajo en tiempo real.

4.1.3.2 Descripción del algoritmo CAMSHIFT

Al contrario que *MeanShift*, este algoritmo está pensado para adaptarse a cambios dinámicos de la distribución probabilística que se rastrea.

Estos cambios ocurren por ejemplo, cuando los objetos que están siendo rastreados, en secuencias de imágenes en tiempo real, se mueven y provocan un cambio en dicha distribución. Tanto en su localización como en el tamaño.

CamShift ajusta el tamaño de la ventana inicial durante el procesado del algoritmo. El tamaño inicial se fija en un valor razonable y empírico.

[6] Se puede pensar en el Momento de Orden Cero como el "área" de la distribución encontrada sobre la ventana de búsqueda.

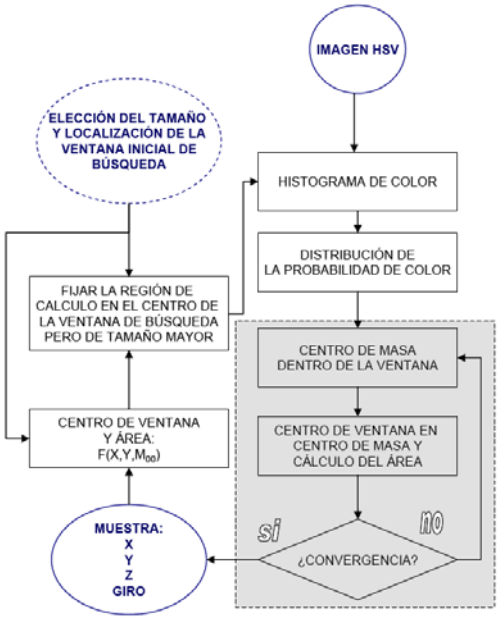


Fig. 49 Esquema de búsqueda, convergencia y adaptación del tamaño de la ventana del algoritmo CAMSHIFT

El cálculo del algoritmo *CamShift* (ver Fig. 49), se puede resumir en los siguientes pasos:

- Elección de la localización de la ventana
- Se calcula *MeanShift* (una o varias iteraciones en función de la convergencia). Se almacena el Momento de Orden Cero (área de la distribución de probabilidad).
- Se toma como nuevo tamaño de ventana uno tal que sea función del Momento de Orden Cero (área)[7].
- Se repiten los dos pasos anteriores hasta llegar a la convergencia (la localización media está separada del centro actual de la ventana en un valor que no supera un umbral definido)

[7] En [60] se puede leer una discusión acerca del valor del nuevo tamaño de la ventana y de la función utilizada para su cálculo.

4.1.3.3 Cálculo de la orientación de la distribución de probabilidad

Se usan los *Momentos de Segundo Orden*: $\begin{cases} M_{20} = \sum_x \sum_y x^2 \cdot I(x,y) \\ M_{20} = \sum_x \sum_y y^2 \cdot I(x,y) \end{cases}$

El área de la distribución de probabilidad, será rodeada por una elipse. La orientación viene dada por el ángulo que forma el eje mayor con respecto a la horizontal:

$$\theta = \frac{\arctan\left(\frac{2\left(\frac{M_{11}}{M_{00}} - x_c y_c \right)}{\left(\frac{M_{20}}{M_{00}} - {x_c}^2 \right) - \left(\frac{M_{02}}{M_{00}} - {y_c}^2 \right)} \right)}{2}$$

[Ec. 30]

Los dos primeros autovalores de la distribución de probabilidad pueden además ser calculados según [61] de la siguiente forma:

$$\text{Sean } \begin{cases} a = \frac{M_{20}}{M_{00}} \\ b = 2\left(\frac{M_{11}}{M_{00}} - x_c y_c \right) - {x_c}^2 \\ c = \frac{M_{02}}{M_{00}} - {y_c}^2 \end{cases}$$

El largo (l) y ancho (w) pueden entonces ser calculados así:

$$l = \sqrt{\frac{(a+c) + \sqrt{b^2 + (a-c)^2}}{2}}$$

$$w = \sqrt{\frac{(a+c) + \sqrt{b^2 - (a-c)^2}}{2}}$$

[Ec. 31]

Cuando este resultado se aplica al seguimiento facial, aporta el ángulo de giro de la cabeza tal y como se muestra en la siguiente figura:

Fig. 50 Orientación de la distribución de probabilidad

4.1.4 Algoritmo redundante propuesto

Primeramente mostramos el esquema del algoritmo propuesto:

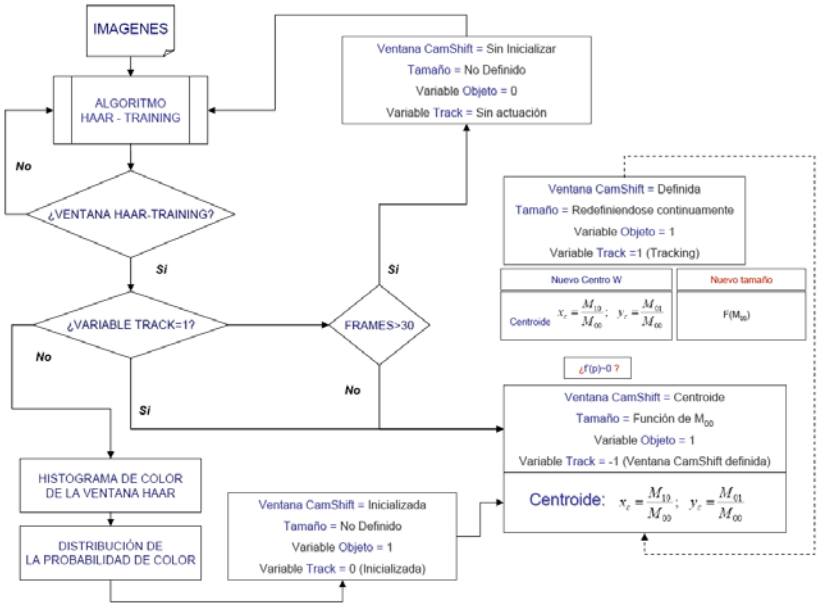


Fig. 51 Algoritmo redundante CAMSHIFT y Haar-Training para evitar las pérdidas temporales en la localización

Como ya se comentó al comienzo de este epígrafe, la idea es proteger al máximo al sistema de pérdidas en la localización facial. *Haar-Training* nos da la capacidad de situar la cara frontalmente en función de una búsqueda con plantilla:

(1) Depende de la visibilidad de las características faciales y de su orientación

(2) Es invariante a los cambios de color

CAMSHIFT ofrece un seguimiento de una determinada distribución de probabilidad, de un color:

(1) Depende de los cambios de color

(2) Es invariante a características faciales y a cambios de orientación

Los cuatro puntos anteriores dan una idea inicial de la propuesta que se realiza: un algoritmo invariante es siempre robusto, dado que no se ve afectado por cambios dinámicos en la detección. La no varianza a características faciales y cambios de orientación de la propuesta probabilística, puede compensar esta dependencia que se muestra como desventaja en *Haar-Training*.

Por otra parte, *CamShift* depende de una *moda probabilística* que se calcula a partir de un color determinado: El que se quiere rastrear. La imagen será analizada en el espacio de color HSV para que no se vea afectada de las características cromáticas de la luz. Es la particularidad de este espacio de color.

Los sistemas clásicos de localización facial basados en color, establecían que, en este espacio, las zonas pertenecientes a regiones de piel tienen los siguientes valores empíricos:

$$0.32 \leq S \leq 0.51$$
$$28.94^{o} \leq H \leq 360^{o}$$
$$0.55 \leq V \leq 0.93$$

[Ec. 32]

Estudiar la detección con estos valores, u otros en función de nuestras condiciones de entorno, da lugar que la variación que se pueda dar entre los colores de piel de cada persona. Del mismo modo que los cambios de iluminación que se puedan

producir, pueden modificar los valores impuestos para cada canal, y con ello la detección.

La segunda propuesta consiste en no definir de forma estática estos umbrales, y permitir al sistema la elección de los mismos en función de la persona que esté detectando en ese momento. Dado que tenemos una detección inicial con plantillas, se puede utilizar el color de esta área de interés, que siempre se corresponde con una cara, como dato inicial para calcular la moda probabilística y determinar la distribución a seguir.

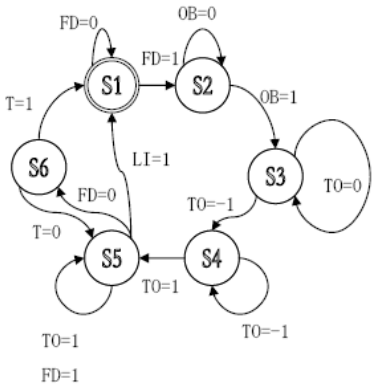
El algoritmo debe seleccionar el dato a rastrear, es decir el color, segmentar la imagen y reproyectarla para calcular el área de la distribución probabilística. Posteriormente deberá seguirla.

Si los valores de esta característica se salen de los rangos de localización definidos, en función de la lejanía entre las dos detecciones, deberá inicializarse de nuevo. Al mismo tiempo, debe verificar el estado de detección del algoritmo por plantillas, así como la no detección y su intervalo. Si este lapso supera un umbral de tiempo, avisa al resto del sistema para que pare su detección y comience de nuevo la selección de la característica.

Consideramos esta propuesta como un sistema que cambia de un estado a otro en función de unas variables booleanas que marcan su transición según las especificaciones impuestas.

4.1.4.1 Máquina de Estados del Algoritmo

En Fig. 52 se puede ver el modelo de comportamiento del algoritmo. La salida será un modo de actuación en función del valor de una serie de variables booleanas (que serán las entradas). El siguiente diagrama de estados representa gráficamente el modelo propuesto:



VALOR DE LA VARIABLE	DESCRIPCIÓN
FD=0	Cara no detectada (HAAR)
FD=1	Cara detectada (HAAR)
0B=1	Seleccionando área de color
0B=0	Color seleccionado
T0=0	Inicia cálculo de distribución
T0=-1	Calculando Distribución
T0=1	Trabajo normal CAMSHIFT
T=0	Tiempo < 30 secuencias
T=1	Tiempo > 30 secuencias
LI=1	Sobrepasa limitaciones definidas

VARIABLE	SIGLA
m_FaceDetected	FD
m_select_object	0B
m_track_object	T0
m_ContHaar	T
m_limitations	LI

DESCRIPCIÓN DEL ESTADO	ESTADO
Sistema en Espera	S1
Detección de cara	S2
Solicitud de selección de área	S3
Cálculo Distribución Probabilidad	S4
Detección y Seguimiento normal	S5
Estado de Comprobación	S6

Fig. 52 Máquina de estados del algoritmo redundante para detección y localización facial.

El sistema funciona libremente mientras no detecte una cara. En este momento, el proceso que vigila es el algoritmo *Haar-Training*.

Cuando las plantillas localizan una cara dentro de la imagen, el algoritmo redundante deja el estado de reposo para pasar a un estado de detección. Captura un área de la imagen que se encuentre centrada en el centro de la localización facial, y de un tamaño ligeramente menor. Este tamaño está definido estáticamente con valores empíricos. En el caso de nuestro sistema, se ha elegido

$$A_{Seleccion} = \left(A_{Haar} / 16 \right)$$

[Ec. 33]

En Fig. 53 se ve el proceso automático que sigue la implementación realizada. Una vez seleccionado, se toma el color, se calcula su moda probabilística y la superficie que tiene su distribución de probabilidad.

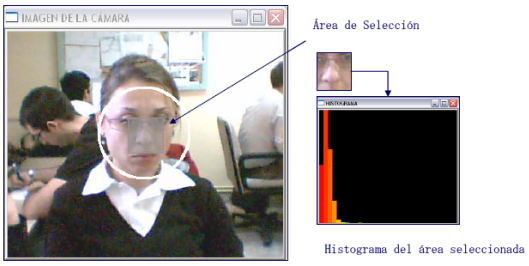


Fig. 53 Proceso de selección del área que marca el color para la distribución de probabilidad

Este algoritmo, no reduce el tiempo de procesado de forma significativa con respecto al funcionamiento por separado de cada uno de los dos que lo componen. Si bien es cierto que se necesitan unas condiciones mínimas para su correcto funcionamiento, una vez que se dan, la propuesta realizada ofrece unos resultados altamente satisfactorios, eliminando algunas de las limitaciones que conlleva el funcionamiento independiente de cada uno de ellos. Todo esto se analiza con detalle en el siguiente apartado.

4.1.5 Análisis de resultados

La implementación y pruebas han sido realizadas en un sistema WindowsXP sobre Intel® Pentium® 4 CPU 3.20GHz con imágenes de 320x240 y un área mínima de detección facial de 40x40 píxeles. Los resultados están estadísticamente calculados durante todos los ensayos realizados.

Se ha analizado el sistema con los algoritmos *Haar-Training* y *Camshift* funcionando de forma independiente y posteriormente trabajando según el algoritmo redundante propuesto.

Asimismo, se han ido variando las condiciones iniciales del entorno durante las pruebas: Condiciones de normalidad, entendiendo tal concepto como aquellas bajo las cuales fue diseñado.

Estas condiciones son:

(1) Ruido (afectando al color de la piel)

(2) Oclusiones

(3) Personas diferentes (colores de piel distintos)

(4) Aumento/Disminución de intensidad (cambios de brillo)

Es indiscutible que *Haar-training* es muy robusto en la detección de caras ante cambios de iluminación y presencia de ruido. Sin embargo sí se han detectado situaciones en que se producen fallos en el algoritmo.

El detector o plantilla ha sido entrenado para una posición determinada del objeto. En nuestro caso, para una posición frontal de la cara. Esto provoca que un giro por encima de un valor determinado, impida el funcionamiento correcto del algoritmo.

Se han evaluado experimentalmente estos giros y el resultado es que, para ángulos mayores de 15º-20º en plano y 35º-º45º en otras direcciones (posición de perfil por ejemplo), el detector se vuelve poco fidedigno.

También se han detectado errores cuando el fondo de la escena presenta colores muy claros frente a caras muy oscuras. Lo mismo que en la situación inversa donde se tiene mucha oscuridad de fondo e iluminación excesiva en la región facial.

En estas situaciones se producen falsas detecciones, en muchas ocasiones con un alto grado de oscilación. Según Viola [57], la inserción de una varianza no lineal de normalización basada en estadística robusta, podría dar lugar a una buena corrección de este efecto. Aunque no sin el coste computacional adicional que, por otra parte, en muchos casos no conviene.

Por último, uno de los fallos más visibles y evidentes que tiene el detector se produce ante oclusión de alguna característica importante. Por ejemplo, si los ojos no están visibles en la imagen, el algoritmo falla. Sin embargo, una oclusión momentánea de la boca no produce este fallo, dado que no es una de las consideradas propiedades importantes para la evaluación de la plantilla, como se puede apreciar en la Fig. 54.

Fig. 54 Tres momentos en la detección con el algoritmo Haar-Training. Se puede ver la detección normal en la imagen central. En la imagen de la derecha se sigue detectando la cara, aún estando oculta una de sus características. En la imagen de la izquierda la oclusión se produce en el ojo, considerado fundamental en la evaluación de las plantillas. En esta ocasión el algoritmo falla.

En una segunda parte de análisis, el rastreo con *Camshift* también produce una serie de errores muy ligados a la imagen, al ruido, a la perspectiva y a oclusiones de características de interés.

Es un algoritmo que realiza un re-escalado continuo de sí mismo para ajustarse a la estructura de los datos (ver 4.1.3.2). Cuando la cara está demasiado cerca, su distribución de probabilidad ocupa un área muy grande.

En esta situación, el tamaño de la ventana del algoritmo es también mayor y puede provocar distorsiones en el resultado, incluyendo falsos positivos. Del mismo modo que si la cara está muy distante: en este caso el algoritmo oscila. El resultado visual de este error es una detección oscilante y muy dinámica, de modo que se van dibujando elipses pequeñas en diferentes lugares de la imagen sin llegar a estabilizarse.

Otra situación habitual es que generalmente, cuando *Camshift* está rastreando y siguiendo una cara, la presencia de otras caras o movimientos de manos en la escena, no causa una pérdida de detección. A no ser que alguna de estas distracciones de lugar a oclusiones importantes.

El rastreo lo realiza correctamente. El problema está en que la zona de interés ya no estará formada por el área que representa la cara dentro de la imagen. Sino por toda el área con la misma moda de color, y eso supone que va a englobar todas las caras cercanas que aparezcan, las manos, o cualquier otro color que conduzca a equívocos.

Esto se puede apreciar en la imagen de la izquierda: al colocar la mano cerca de la zona facial, el algoritmo engloba su área dentro de la elipse (Fig. 55).

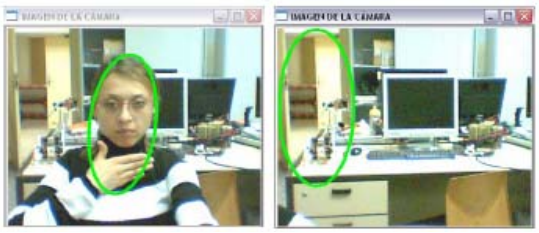


Fig. 55 En la imagen de la izquierda se ve la modificación debida a la presencia cercana de otro objeto (en este caso una mano) con la misma moda de color. La ventana se reajusta y el área de distribución de color no se corresponde únicamente con el área de la cara, sino con la suma de éste área y el de la mano. En la imagen de la derecha, se ha producido un efecto del mismo orden: no aparece una cara en la imagen, pero sí que hay zonas de la escena con un color similar, lo que provoca que el algoritmo las detecte.

En el mismo orden de error, se puede dar una situación en la que, durante la búsqueda de un color cercano al color de la piel, sean detectados objetos que no tienen nada que ver con la cara y que -sin embargo- sí entran en el rango de la moda de color buscada por el algoritmo (ver imagen derecha en Fig. 55). Si podemos disponer de un segundo algoritmo que no dependa de esta característica de color, se puede llegar a modificar el resultado final de este comportamiento.

Es decir, con dos algoritmos funcionando de forma redundante, se pueden compensar los efectos negativos de cada uno de ellos por separado. La propuesta de la máquina de estados pretende un método de trabajo en paralelo de modo que, en condiciones de detección correctas el resultado sea redundante y, en caso contrario, sean capaces de compensarse mutuamente.

Si ponemos a trabajar los dos algoritmos dentro de sus límites de actuación, las dos detecciones son correctas (como se puede apreciar en Fig. 56). Tenemos un resultado redundante (dos detecciones de la misma cara).

Fig. 56 Funcionamiento normal del algoritmo redundante. El color blanco se corresponde con el algoritmo Haar-Training y la elipse verde con el algoritmo Camshift.

Sin embargo, como se puede ver en la siguiente ilustración, ante errores de uno de los dos métodos, el otro puede continuar su detección (evitando la pérdida del interlocutor de Urbano como última consecuencia). Recordamos que *Camshift* podía provocar una variación del área de distribución que rastrea (por presencia de objetos adicionales en la escena con colores similares), dando lugar a una zona de interés que no tiene por qué ser exactamente la cara. Ni siquiera tiene por qué estar contenida completamente en la misma (ver Fig. 57 izquierda). En algoritmos de identificación facial (se tratarán en epígrafes posteriores), esto puede ser un problema dado que, para reconocer a un interlocutor, el sistema debe de analizar únicamente la zona facial. Como se puede apreciar en la figura, Haar-training define exactamente esta localización, corrigiendo de este modo la variación de *Camshift*.

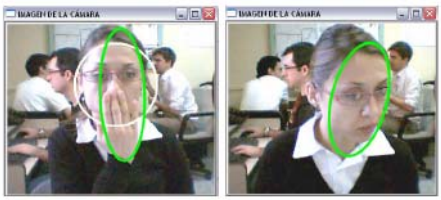


Fig. 57 Imagen de la izquierda: Haar-Training mantiene el área real de la escena que corresponde a una cara aunque Camshift esté determinando otra. En la imagen de la derecha, es Camshift el que mantiene la detección ante la pérdida a que dan lugar las plantillas ante giros de la cabeza

Como se puede observar, con el algoritmo propuesto los errores de detección quedan ampliamente compensados. La característica buscada se extrae y se sigue correctamente sin obligar al sistema a una carga computacional alta. Este método está pensado para ser la base de un sistema de interacción entre un humano y un robot a través de la imagen. Si el algoritmo inicial con el que se detecta una presencia de alguien conlleva un gasto de memoria importante, los siguientes no van a disponer de un rango adecuado de recursos para poder trabajar.

4.2 Identificación facial

La cara es el foco primario de atención en nuestras relaciones sociales, jugando un papel aún más importante a la hora de expresar identidad o sentimientos. Los humanos –inicialmente- no tenemos capacidad de inferir inteligencia o carácter a

partir de una cara, pero sí de reconocer esa cara [55]. De hecho, somos capaces de reconocer cientos de caras aprendidas durante el transcurso de nuestra vida e incluso reconocer aquellas que no hemos visto durante largos periodos de tiempo. Nuestra capacidad visual es tan robusta que realiza su función de idéntica forma aún cambiando el estímulo (cambios en la iluminación, expresión, edad, distracciones como gafas, peinados, etc.).

Los modelos computacionales de reconocimiento facial no sólo son interesantes por la capacidad de comprensión de nuestro sistema perceptivo que de ellos obtenemos, sino también por la consecuente aplicación práctica que podemos adquirir de su implementación. Puede ser empleado para resolver una amplia variedad de problemas en sistemas de seguridad, identificación criminal, procesamiento de imagen e interacción hombre-máquina.

En nuestro sistema, una vez que tenemos la localización de una cara dentro de la escena, pasamos a un segundo proceso en el que se intenta conocer la identidad de la misma. Como se ha expuesto en párrafos anteriores, el objetivo es que Urbano conozca su entorno, lo aprenda y lo identifique. En este caso, el entorno serán las personas más cercanas con quienes va a trabajar y, más concretamente, un tutor a quién obedecerá las órdenes más complicadas.

El método de identificación que se presenta está ampliamente estudiado e implementado en el estado del arte. Nuestra propuesta añade una pequeña variación. Generalmente, las detecciones a que son sometidos los sistemas, parten de fotografías que se reconocen dentro de un dominio definido de personas. El usuario inserta en el sistema una nueva imagen y solicita el reconocimiento. Nosotros queremos el mismo proceso, pero con ejecución automática y en tiempo real. Veamos los motivos.

Urbano se mueve de forma autónoma por su entorno y debe conservar esta autonomía en su interacción con las personas e identificación del medio. Es decir, aunque el sistema se va a entrenar de la misma manera que la propuesta elegida, la diferencia va a estar en el momento de la identificación, ya que la solicitud de identificación será automática.

Urbano permanecerá ejecutando el algoritmo de detección de presencia (4.1.4) hasta que se produzca una entrada en la escena de una cara. En ese momento, comenzará a ejecutarse el algoritmo de identificación facial sin necesidad de que sea el usuario quien lo solicite.

4.2.1 Introducción

Las mayores dificultades con la que nos encontramos al intentar desarrollar un modelo de identificación facial, es la complejidad de las características faciales, la multidimensionalidad y nuestra falta de conocimiento acerca del funcionamiento de nuestro cerebro para dar significado al estímulo visual.

Durante el estudio de las diferentes técnicas utilizadas actualmente para este fin, nos encontramos con esta idea de utilizar el *Análisis de Componentes Principales.* La razón por la que se tomó la decisión de implementarlo es primero, la posibilidad de reducir la dimensionalidad de los datos y segundo, poder obtener un subespacio vectorial de características más relevantes donde proyectar las imágenes. En otras palabras, un subespacio vectorial de menor dimensión donde realizar el trabajo de aproximación y reconocimiento.

La idea es pues, trabajar sólo con aquellas características principales, las más notables entre todas las que se detecten en las caras y no con el espacio completo. En [54] y [55] se exponen las ideas principales del método y son los artículos que han servido de base a nuestra solución.

En líneas generales, se parte de un conjunto de imágenes faciales (aquellas que queremos que después sean conocidas). Con este conjunto de imágenes se va a entrenar el sistema. El esquema está basado en una aproximación teórica que descompone las imágenes en pequeños conjuntos de características llamadas *"eigenfaces"* y que van a ser consideradas los componentes principales del conjunto de entrenamiento. El reconocimiento está basado en la proyección de la nueva imagen dentro del subespacio vectorial para clasificarla, comparando su posición con la posición de las otras caras conocidas dentro de este espacio.

ANÁLISIS EN ESPACIOS DE CARACTERÍSTICAS DE MENOR DIMENSIÓN

El análisis está basado en el hecho de que una clase de patrones de interés, como son las caras, residen en una parte del espacio de características de la imagen y no en todo él al completo. Las caras tienen una configuración de sus características muy similar, de modo que todas van a estar dentro del mismo rango de posición dentro del espacio de características. La imagen original supone una representación altamente redundante y de una dimensión que excede a las necesidades de proyección de estos patrones.

Con el Análisis de Componentes Principales (PCA) [52], un pequeño número de EigenFaces [62] son calculados a partir de un conjunto de imágenes de

entrenamiento usando la Transformada de Karhunen-Loeve o PCA. De este modo, una imagen queda representada por un vector de características de pequeña dimensión: un vector de pesos. Con esta información vectorial, tenemos capacidad de construir un espacio vectorial de menor dimensión, incluido en el espacio de características de la imagen original, en el que proyectar el trabajo de identificación con menor carga de comprobación matemática y menos exigencias computacionales.

4.2.2 Identificación facial con Eigenfaces

Muchos de los trabajos realizados en este campo han dejado a un lado el estudio de las características que son verdaderamente importantes para identificar una cara. No tanto desde la perspectiva de la percepción humana, cuyos principios desconocemos en gran medida, sino desde un punto de vista computacional. Singularidades que ofrezcan unas pautas claras, eficaces y precisas para la identificación. Esta y otras carencias, llevaron a M. Turk y A. Pentland [54] a implementar un sistema de codificación y descodificación de la imagen, en función de su información intrínseca.

El planteamiento consistía en enfatizar las características, locales y globales, basándose en un método de reducción de dimensionalidad cuya idea inicial fue desarrollada por Sirovich y kirby [62] y [63] entre 1987 y 1990. El propósito final es realizar las comparaciones entre las transformaciones de las imágenes y no entre ellas en sí mismas, pudiendo tener así un sistema de identificación menos pesado (en cuanto a gasto computacional) y más eficaz (por comparar únicamente singularidades importantes para la toma de decisión).

En términos matemáticos, lo que queremos es encontrar los componentes principales de una determinada distribución de píxeles. Es decir, los vectores principales de la matriz de covarianza de un grupo de imágenes faciales, tratando estas imágenes como puntos o vectores de un espacio vectorial de gran dimensión.

Estos vectores principales pueden ser entendidos como peculiaridades que caracterizan la variación entre las imágenes de las caras, y van a ser utilizados para construir un espacio vectorial de menor dimensión donde poder proyectarlas. De este modo, cada imagen tendrá una aportación de cada vector principal que, además, constituye una característica. Por eso reciben el nombre de *"eigenfaces"* o

"caras principales". En la Fig. 58 se pueden ver tres de las imágenes utilizadas en la etapa de entrenamiento y sus correspondientes *eigenfaces*.

En un proceso inverso, cada imagen se puede sustituir exactamente por una combinación lineal de los *eigenfaces* calculados. De este modo puede quedar representada sólo por los mejores –aquellos que tienen los mayores autovalores asociados- que son los que aportan las mayores varianzas entre el conjunto de entrenamiento. No tienen por qué ser todos, porque el objetivo no es la reconstrucción y, por lo tanto, la pérdida de información al prescindir de uno de los vectores principales queda plenamente compensada por el aumento de la capacidad computacional. Esto nos da la oportunidad de incrementar el rango del entorno y reconocer a un grupo mayor de personas. Todo esto se verá más adelante.

Fig. 58 Fila superior: tres de las imágenes usadas para el entrenamiento del sistema. Fila inferior: tres de los eigenfaces calculados a partir de las imágenes de entrenamiento.

4.2.2.1 Operaciones de inicialización

El procedimiento de identificación facial con *Análisis de Componentes Principales* requiere una serie de puntos de inicialización previos y supone una etapa precedente imprescindible. Los enumeramos:

(1) Adquisición de un conjunto inicial de imágenes para el entrenamiento (ver Fig. 59).

(2) Cálculo de las *caras principales* del conjunto de entrenamiento, también llamadas *eigenfaces*, manteniendo únicamente aquellos con mayores valores propios asociados. Estos M vectores propios constituirán el nuevo espacio vectorial de trabajo: El *facespace*.

Siempre que se tengan que incluir nuevas imágenes al conjunto, se deberá repetir esta operación.

(3) Cálculo de la correspondiente distribución de pesos en el espacio M-dimensional, proyectando cada imagen en dicho espacio.

Las operaciones de reconocimiento serán siempre posteriores a esta fase de inicialización.

4.2.2.2 Secuencia de operaciones para el reconocimiento

El proceso comienza con la recepción de una nueva imagen sobre la que se solicita la identificación. Si nuestro espacio de trabajo está codificado, tendremos que transformar igualmente esta nueva imagen para poder trabajar con ella bajo la misma referencia. La sucesión de puntos a seguir es la siguiente:

(1) Cálculo del conjunto de pesos de la imagen de entrada en el *facespace*. Es decir, la proyección[8] de dicha imagen para conocer la contribución de cada *eigenface* en ella.

(2) Determinación de si la imagen se corresponde con la de una cara o no, evaluando su cercanía al *facespace*.

(3) Determinación de si la cara es conocida o no, evaluando su cercanía a las clases del *facespace*, definidas durante el entrenamiento.

Adicionalmente, se puede obtener un proceso de aprendizaje incluyendo aquellas imágenes clasificadas como no conocidas repetidamente en el trayecto de una serie de evaluaciones, y realizando de nuevo las operaciones de inicialización, ver sección 4.2.2.1 .

[8] La proyección ortogonal de un vector **a** sobre otro vector **b** es la sombra perpendicular que ejerce el primero sobre el segundo. Se calcula matemáticamente: $P = \frac{\vec{a} \bullet \vec{b}}{\vec{b} \bullet \vec{b}}$

4.2.2.3 Cálculo de los Eigenfaces

Tal y como se ha dicho en el apartado anterior, partimos de una imagen de entrada. Esta imagen $I(x, y)$ (con una profundidad de color de 8 bpp o 256 colores) la vamos a considerar una matriz de datos de dimensión $N \times N$. Esta matriz puede será transformada en un vector de dimensión N^2 encadenando todas las filas:

$$\Gamma i = \begin{bmatrix} P_{ix_{11}} \\ \\ P_{ix_{NN}} \end{bmatrix}_{\left[N^2 \times 1\right]}$$

[Ec. 34]

Con esta transformación, una imagen típica de tamaño 256×256 viene a ser un vector de dimensión 65.536, o de forma equivalente, un punto en un espacio $65.536 - Dimensional$. Ya tenemos expresada la imagen en términos matemáticos para poder procesarla. Evaluamos ahora su información intrínseca.

Si analizamos una cara, podemos afirmar rápidamente que no tienen una distribución de características aleatoria. Muy al contrario suelen tener una configuración parecida, por lo que en principio es lógico pensar que podrían ser representadas en un espacio vectorial de menor dimensión. Esta lógica tiene su base en algo tan sencillo como eliminar la información común, las características comunes a todas ellas, y trabajar únicamente con sus variaciones. Al eliminar las características comunes, eliminamos dimensiones del espacio vectorial de características donde inicialmente están representadas. Y al trabajar con las variaciones y no con las características en si mismas, eliminamos otra parte dimensional. El objetivo final es encontrar un conjunto de direcciones de variación de características que no cambien al someter la imagen a transformaciones, y las direcciones en álgebra lineal vienen definidas por vectores.

Esta es la idea de los componentes principales[9], encontrar el conjunto de vectores que mejor definan la distribución de características y formar con ellos un nuevo espacio vectorial de menor dimensión que será nuestro nuevo espacio de trabajo: el *facespace*. Estos vectores deben ser linealmente independientes y ortogonales. Cada uno de estos vectores (de dimensión N^2) es una combinación lineal del conjunto de imágenes inicial. Uno de los motivos por el que se han dado en denominar

[9] También llamada extensión de Karhunen-Loeve

eigenfaces o *caras principales* es por que el resultado de la implementación revela una cara *fantasmagórica* (ver Fig. 60).

Otro de los motivos tiene que ver con el hecho de que las particularidades que definen pertenecen a características faciales.

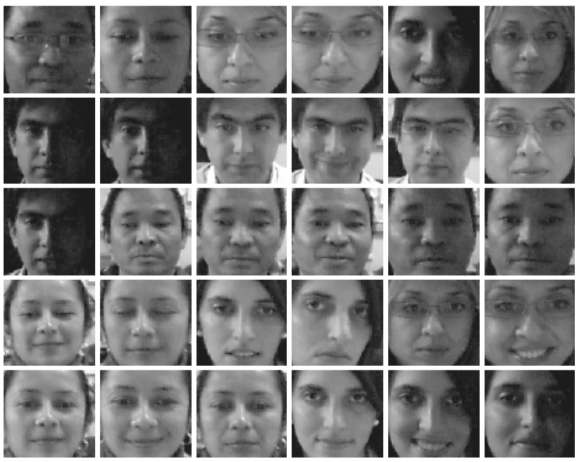


Fig. 59 Muestra de un posible conjunto inicial de imágenes de entrenamiento.

Consideramos un conjunto de M imágenes (ver Fig. 59) convertidas en vectores $\Gamma_1, \Gamma_2, ..., \Gamma_M$. Tal y como se hace con las series de Fourier, tenemos un valor medio que no nos aporta información distintiva entre ellas. Si a cada imagen le sustraemos este valor, nos quedamos sólo con las características adicionales a la media (análogo a eliminar el valor medio de una señal y realizar la codificación con los armónicos de interés).

Una vez que tenemos las imágenes convertidas en vectores, organizamos esta información en una matriz de la siguiente manera:

$$\Gamma = \begin{bmatrix} \Gamma_1 \\ ... \\ \Gamma_M \end{bmatrix}^T_{[MxN]} \quad donde \quad \Gamma_i = \begin{bmatrix} P_{ix_{11}} \\ ... \\ P_{ix_{NN}} \end{bmatrix}_{[N^2 \times 1]} \qquad \forall i = 1...M$$

[Ec. 35]

Siendo M el número de imágenes tomadas para el entrenamiento.

Se halla la media:

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\Gamma_n$$

[Ec. 36]

Restando dicho valor a cada una de las imágenes iniciales, se forman los vectores diferencia:

$$\Phi_i = \Gamma_i - \Psi$$

[Ec. 37]

Estos vectores diferencia se someten al cálculo de los componentes principales, que busca los M vectores propios ortonormales, u_n, que mejor definan la distribución de los datos.

El criterio para la elección de estos vectores estará ligado a sus valores propios asociados:

$$\lambda_K = \frac{1}{M}\sum_{n=1}^{M}\left(u_k^T\Phi_n\right)^2 \rightarrow \lambda_K = Maximo$$

[Ec. 38]

Sujeto a:

$$u_l^T u_k = \delta_{lk} = \begin{cases} 1, & l = k \\ 0, & otros \end{cases}$$

[Ec. 39]

Siendo u_k, λ_k los vectores y valores propios –respectivamente- de la matriz de covarianza C:

$$C_{\left[N^2 \times N^2\right]} = \frac{1}{M}\sum_{n=1}^{M} \Phi_n \Phi_n^T = A \cdot A^T$$
$$A = \left[\Phi_1 \quad ... \quad \Phi_M\right]$$

[Ec. 40]

La dimensión de la matriz de covarianza, determina un número de vectores y valores propios del orden de N^2, algo que es prácticamente imposible de procesar. Necesitamos un método para poder tratar el sistema computacionalmente.

Si el número de puntos en el espacio vectorial (número de imágenes) es mucho menor que la dimensión del mismo $\left(M << N^2\right)$, únicamente habrá $\left(M-1\right)$ vectores propios distintos de cero. Es decir, $\left(M-1\right)$ vectores propios significativos.

Afortunadamente podemos hacer el cálculo de los vectores propios (de dimensión N^2) determinando primeramente los de una matriz $\left[M \times M\right]$ y después estimando una combinación lineal adecuada de las imágenes diferencia de las caras (Φ_i). Consideramos entonces que v_i es un vector propio de $A^T A_{[M \times M]}$.

$$A^T A \cdot v_i = \mu_i \cdot v_i$$

[Ec. 41]

Multiplicando ambos factores por A:

$$AA^T A \cdot v_i = \mu_i \cdot A \cdot v_i$$
$$AA^T \cdot (A \cdot v_i) = \mu_i \cdot (A \cdot v_i)$$

[Ec. 42]

Como podemos ver en la [Ec. 42], $A \cdot v_i$ constituyen los vectores propios de $C = AA^T$. Es decir, los vectores propios no son los v_i como correspondería si la matriz de covarianza fuera $C = A^T A$. Los vectores propios son las proyecciones de cada imagen (sin su valor medio) en la base ortogonal que forman los v_i. La parte invariante la establece $A \cdot v_i$.

En base a este resultado, construiremos una matriz $L_{[M\times M]} = A^T A$ donde cada elemento sea $L_{mn} = \Phi_m^T \Phi_n$. Hallamos los vectores propios de L, que van a formar la primera base ortogonal sobre la que proyectar las imágenes. Estos vectores propios determinan una combinación lineal del conjunto de las imágenes de entrenamiento para formar los e*igenfaces* u_l:

$$u_l = \sum_{k=1}^{M} v_{lk} \cdot \Phi_k \quad \forall l = 1...M$$

[Ec. 43]

Y con estos *eigenfaces* calculados, formamos el nuevo espacio de trabajo: el subespacio vectorial denominado *facespace*. Con esta transformación matemática, el cálculo queda ampliamente reducido de un orden N^2 a otro correspondiente al número de imágenes de entrenamiento (el conjunto M).

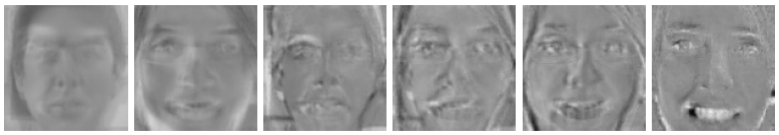


Fig. 60 Seis de los eigenfaces calculados a partir de las imágenes de entrada de Fig. 59

4.2.2.4 Clasificación de una imagen e identificación

Los *eigenfaces* calculados como se expone en el apartado anterior, parecen adecuados para describir imágenes de caras e identificarlas. Sin embargo, aunque el número de vectores principales que obtengamos sea M, casi siempre podemos alcanzar una buena solución con una cantidad menor M'<M. No necesitamos todos los vectores propios puesto que la reconstrucción de imágenes no es uno de los requerimientos del sistema. Sólo con estos M' vectores podemos construir un subespacio vectorial M'-dimensional del espacio vectorial original N^2-dimensional válido para nuestros propósitos.

Suponemos entonces que entra una imagen nueva (Γ) a nuestro sistema ya entrenado. Debe ser transformada en sus componentes *eigenfaces* para obtener la misma codificación que aquellas con las que se va a comparar. Dado que al entrenar el sistema se substrajo previamente el valor medio a cada imagen, a esta nueva se le debe aplicar el mismo proceso, y después proyectarla dentro del *facespace*.

$$\omega_k = u_K^T \left(\Gamma - \Psi\right) \quad k = 1...M'$$

$\Psi \rightarrow$ *Media*
$M' \rightarrow$ *Nºeigenfaces*
$\omega \rightarrow$ *Peso*

[Ec. 44]

Cada (ω_k) es un peso, que viene a ser la coordenada en un sistema cartesiano tradicional. Es decir, la contribución que hace el *eigenface* k en la representación de la imagen. Todos los pesos correspondientes a una imagen forman el vector de pesos de la misma:

$$\Omega^T = \left[\omega_1 \quad \omega_2 \quad ... \quad \omega_{M'}\right]$$

[Ec. 45]

Esto constituye un marco de trabajo típico de identificación de patrones. El vector de pesos va a ser usado por un algoritmo sencillo de búsqueda para determinar si alguna clase de las definidas se acerca a la que estamos evaluando. El método más simple para decidir qué clase facilita la mejor descripción de la nueva imagen, es encontrar la clase k que minimice la distancia Euclídea:

$$\varepsilon_k^2 = \left\|\Omega - \Omega_k\right\|^2$$

[Ec. 46]

Es decir:

$$\varepsilon_k = \min\left(d\left(\Phi, \Phi_i\right)\right) = \min\left(\left\|\Phi, \Phi_i\right\|\right) \quad \forall i = 1...M$$

[Ec. 47]

La obtención del vector de pesos es el proceso equivalente a proyectar la nueva imagen sobre el espacio vectorial de menor dimensión definido. Cualquier imagen, facial o no, puede ser proyectada sobre este *facespace* obteniendo así un vector de pesos. Vamos a evaluar primero la proyección de la imagen de una cara ya que, los resultados que se obtienen, ayudan a explicar qué ocurriría si proyectamos algo diferente a una imagen facial.

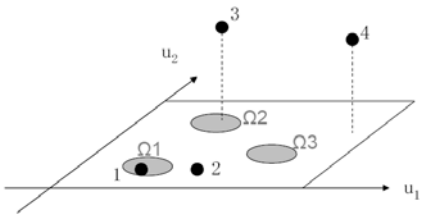


Fig. 61 Versión simplificada del facespace. Dos eigenfaces (u_1,u_2), tres personas conocidas (Ω1, Ω2, Ω3) y cuatro situaciones: 1) Cercano al facespace y a la clase 1; Persona conocida. 2) Cercano al facespace pero lejos de cualquiera de las clases definidas; Persona no conocida. 3) Lejos del facespace y dentro de una clase; Falso positivo. 4) Lejanía del espacio y de la clase; No es una imagen facial.

Definiendo la distancia ε (siendo $\varepsilon_k \neq \varepsilon$) como el alejamiento entre la nueva imagen y las que pertenecen al *facespace*, podemos obtener un criterio discriminante de pertenencia al mismo. Inicialmente es posible un proceso inverso de reconstrucción de la imagen a partir de los *eigenfaces*:

$$\Phi_{reconstruida} = \sum_{p=1}^{M'} \omega_p \cdot u_p$$

[Ec. 48]

Por otro lado, la imagen original estará siendo evaluada sin su valor medio:

$$\Phi = \left(\Gamma - \Psi\right)$$

[Ec. 49]

Estas dos estimaciones, en una situación ideal de reconstrucción perfecta, deberían de ser iguales. Por lo tanto y, ya que estamos en un espacio Euclídeo, podemos usar su métrica para calcular la distancia entre ambos y proponer umbrales de pertenencia al *facespace*.

$$\varepsilon^2 = \left\| \Phi - \Phi_{reconstruida} \right\|^2$$

[Ec. 50]

En la implementación que se ha desarrollado para nuestro robot Urbano, se ha calculado el umbral de esta distancia estimando que son las propias imágenes (con las que se ha construido este espacio vectorial) las que deben definir en qué cantidad. Es decir, las M imágenes que se han utilizado, tienen una distancia entre ellas y alguna debe ser máxima. Por lo tanto, es lógico pensar que cualquier imagen que esté entre las entrenadas, debería tener una distancia con las demás que no supere este alejamiento. Por ese motivo se ha elegido como umbral la máxima distancia entre las imágenes del conjunto de entrenamiento.

$$\varepsilon = \max\left(d\left(\Phi_i, \Phi_j \right) \right) = \max\left(\left\| \Phi_i, \Phi_j \right\| \right) \quad \forall i, j = 1...M$$

[Ec. 51]

En la Fig. 61 se pueden apreciar gráficamente las cuatro posibilidades de pertenencia para una imagen de entrada a evaluar. En el primer caso se reconoce a una persona y se identifica. En el segundo caso se reconoce la existencia de una persona pero no se puede identificar: permanece desconocida al sistema. El tercer caso es una cuestión típica de falso positivo. En nuestro sistema sin embargo puede ser detectado este error puesto que tenemos un algoritmo redundante (4.1.4) funcionando en la detección de presencia facial en la escena.

Ahora nos preguntamos ¿qué ocurre si proyectamos en el *facespace* una imagen que no contenga una cara? Vamos a analizar esto en el siguiente apartado.

4.2.2.5 Localización y detección de caras

El análisis realizado en las secciones anteriores asume que tenemos centrada una imagen facial en la escena del mismo tamaño que las utilizadas para el entrenamiento. De alguna forma, necesitamos localizar la cara en la escena para realizar la identificación. En nuestro desarrollo esta parte queda completamente cubierta por los algoritmos presentados en la sección anterior. Son éstos los que realizan las labores de detección y localización de la cara.

Sin embargo, la información que nos ofrece el *facespace* puede utilizarse como alternativa para localizar caras dentro de la imagen. Nos permite reconocer una presencia facial aparte de la tarea de identificación.

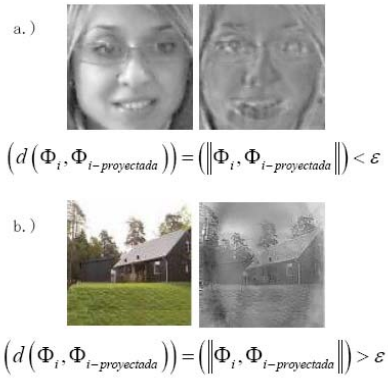


Fig. 62 Imágenes y sus proyecciones en el facespace definido por los eigenfaces. Vemos que en el caso a.) la distancia entre la proyección de la cara no supera el umbral de pertenencia, son parecidas. Sin embargo en el caso b.) la imagen y su proyección son muy diferentes.

Como podemos ver en la Fig. 62, las imágenes de caras no cambian de forma radical cuando son proyectadas en el *facespace*, mientras que la proyección de otro tipo de imagen se presenta muy diferente.

Esta idea básica es la que se utiliza para detectar la presencia de cara en la escena: se crea una ventana (de dimensiones idénticas a las imágenes utilizadas en el entrenamiento) que irá evaluando para cada localización la distancia ε calculada según el apartado anterior. Esta distancia al *facespace* es utilizada como medida de ausencia de caras dentro de la imagen:

$$\varepsilon = \max\left(d\left(\Phi_i, \Phi_j\right)\right) = \max\left(\left\|\Phi_i, \Phi_j\right\|\right)$$
$$\forall i, j = 1...M$$
$$\forall p \quad cualquiera$$
$$\begin{cases} d\left(\Phi_p, \Phi_{p-proyectada}\right) < \varepsilon \rightarrow cara \\ d\left(\Phi_p, \Phi_{p-proyectada}\right) > \varepsilon \rightarrow \neg cara \end{cases}$$

[Ec. 52]

Sin embargo, la aplicación directa de la [Ec. 51] tiene un gasto computacional muy fuerte y se necesitan algoritmos de desarrollo adicionales. En [55] se expone un método de simplificación para hacer frente a esta tarea.

4.2.2.6 Conclusiones

El *análisis de los componentes principales (PCA)* es un método que basa el reconocimiento facial en la idea de encontrar un pequeño conjunto de características que aproximen de forma óptima un grupo de imágenes conocidas.

Las singularidades que pueda ofrecer esta metodología, no tienen por qué ser las mismas que nuestra percepción valoraría como necesarias para el mismo fin.

No es una solución elegante pero, en general, proporciona un procedimiento para la identificación facial rápido, sencillo y con unos resultados aceptables en entornos adecuados al sistema. Se han detectado fallos importantes para los cambios de iluminación fuertes y en circunstancias en las que la fisonomía de dos personas sea parecida: Mismo peinado, mismo ancho de cara, etc. No obstante se están estudiando las posibilidades de reducir los errores y aumentar la capacidad de aprendizaje del sistema.

El prototipo actual no procesa en paralelo. En próximos desarrollos, entre los que se encuentra la versión que irá implementada en el robot, la programación de hilos nos llevará a soluciones en las que se pueda distribuir el trabajo del algoritmo, de modo que sea posible aumentar la velocidad de aprendizaje, aprender e identificar al mismo tiempo y aumentar el número de imágenes del conjunto de entrenamiento. En este caso, tendríamos la posibilidad de incluir en el entrenamiento la misma cara con diferentes variaciones de iluminación y entorno, eliminando en gran medida los errores que se producen por esta causa.

Asimismo, será también necesario el estudio empírico de la cantidad de vectores propios necesarios para dar la mejor solución, puesto que, como ya se ha comentado anteriormente, al no tener como objetivo la reconstrucción de la imagen, debemos minimizar la dimensión del subespacio vectorial en la medida de lo posible.

4.3 Clases C++ dedicadas

4.3.1 Clase Process_Vision: Localización facial

Esta clase está dedicada a las principales funciones de procesamiento de la imagen que sirven como base al sistema de identificación facial. Se describe a continuación el trabajo realizado por aquellas dedicadas a la localización facial.

4.3.1.1 DetectaCara

- bool DetectaCara(IplImage *FaceImg)

Función de implementación del algoritmo *Haar-Training*. En esta función se procesa la imagen procedente de la cámara para intentar localizar presencia de caras. En el caso de que existan, rodea el área facial con un rectángulo o cuadrado y devuelve una variable booleana con valor "verdadero". En caso de detección múltiple, sólo dibujará alrededor de aquella que esté más cercana.

En esta función se da valor a las variables del algoritmo de detección Camshift: [m_CamshiftPermit], que permitirá la inicialización del mismo.

4.3.1.2 FaceReference

- Void FaceReference (void)

Función que estima el valor de la variable [m_FaceRect] a partir del círculo que ha sido detectado como contenedor de una cara. Este rectángulo va a servir de referencia para la región de interés del CamShift. Es un rectángulo dinámico: va a cambiar su tamaño en cada detección de cara.

En esta función además se calcula un rectángulo estático para una clase dedicada al Análisis de Componentes Principales.

4.3.1.3 SelectArea

– Void SelectArea (void)

Función que selecciona el rectángulo adecuado para el comienzo del algoritmo Camshift. Tiene implementada la selección bajo las condiciones de detección de cara, ausencia de selección previa o funcionamiento del algoritmo. Únicamente se selecciona el área de color a seguir en los casos de inicialización. Una vez construido el histograma éste no cambia.

Las variables que condicionan son: (1) [m_FaceDetected], para la detección de cara, (2) [m_select_object] para el control de la selección, (3) [m_track_object] que controla si el algoritmo ya está funcionando y no se necesita inicialización.

4.3.1.4 TrackFace

– Void TrackFace (void)

Función de implementación del algoritmo *Camshift*. Esta función realiza el seguimiento de la cara por distribución de color una vez detectada. El área seleccionada para proporcionar la distribución de color inicial se realiza con la función *SelectArea* (4.3.1.3).

Se ha incluido código para evitar que la elipse y la circunferencia que rodean la cara (la misma cara detectada por dos métodos) no estén cercanas en el momento de la detección y del rastreo.

Por otra parte, cuando el algoritmo *Haar-Training* pierde la detección, permanece la localización con Camshift durante un número determinado de secuencias (empíricamente se han estimado los valores entre 30 y 60 secuencias, en función del entorno que vaya a tener el sistema).

En el transcurso de la espera pueden ocurrir dos cosas: (1) En el caso de no recuperar la detección se inicializa de nuevo el algoritmo completo, (2) En caso de recuperar la detección, Camshift no se inicializa y se continúa la detección con ambos normalmente.

4.3.2 Clase PCA: Identificación facial

Esta clase está dedicada al cálculo de los componentes principales y a las funciones de estimación de identidad de una cara.

4.3.2.1 Draw_EigenFaces

- Void Draw_EigenFaces (void)

Función que calcula y guarda las imágenes de los vectores principales de cada una de las fotografías guardadas en disco. Estas fotografías son tomadas directamente por el software con la medida necesaria para el algoritmo. No hace falta realizar una adecuación fuera de línea de las imágenes.

4.3.2.2 LoadFacesToTrain

- Void LoadFacesToTrain (std::vector <IplImage*> &FacesVector)

Función que carga los vectores principales para su entrenamiento. El argumento es el vector de imágenes. Esta función únicamente lee las imágenes e inicializa el vector con las imágenes cargadas en el disco.

4.3.2.3 TrainEigenFaces

- Void TrainEigenFaces (std::vector <IplImage*> FacesVector, CvMat*AverageFace, CvMat* U, CvMat* U_Transpose, std::vector <CvMat*> FaceProjectionVector, float &Threshold, float & ThresholdClass)

Función que realiza el entrenamiento o aprendizaje del sistema. Convierte el vector de imágenes pasado por argumento en vector de vectores. Halla la media de todas las imágenes. Define un nuevo vector diferencias. Calcula las matrices de covarianza y los vectores y valores propios. Construye la proyección de las imágenes dentro del nuevo espacio vectorial (facespace) y define un umbral de acercamiento al nuevo espacio vectorial.

El threshold de acercamiento se define a partir del vector de pesos [FaceProjectionVector] de cada imagen dentro del facespace. Cada uno de los pesos de la imagen se corresponden con la influencia de cada vector principal (eigenface) en la misma. De esta manera queda constituido un método de reconstrucción de la imagen.

Si la nueva imagen es una cara se puede calcular su proyección dentro del facespace y reconstruirla. Al comparar la original con su reconstrucción, la diferencia debería ser en principio pequeña, ya que son la misma imagen (original y reconstruida).

Parece lógico pensar que el threshold debe estar construido en torno a este razonamiento. Todas las caras conocidas tienen una distancia de diferencia entre ellas. Si vemos la distancia entre una de las caras y las demás, tendríamos una distancia máxima. Cualquier nueva imagen que potencialmente fuera identificada con ella, no debería sobrepasar este valor.

Por lo tanto el valor del threshold está calculado como la distancia máxima entre las imágenes de entrenamiento proyectadas en el facespace.

4.3.2.4 FaceRecognition

- Void FaceRecognition (IplImage* Face, CvMat*AverageFace, CvMat* U, CvMat* U_Transpose, std::vector <CvMat*> FaceProjectionVector, float &Threshold, float & ThresholdClass

Función que realiza la identificación de una persona si se encuentra dentro del conjunto para el que ha sido entrenado el sistema. Toma la imagen nueva [Face], le resta el valor medio calculado, la proyecta y busca dentro del facespace: (1) primero, si está dentro de las imágenes conocidas y (2) en caso de que pertenezca al facespace, qué imagen minimiza la distancia.

CAPÍTULO 5

Extracción de características gestuales

El objetivo de este capítulo es exponer un método de selección de características que sean suficientemente concluyentes como para utilizarlas en un sistema de aprendizaje. El propósito final tiene dos vertientes de importancia:

- Selección de singularidades fuertes e invariantes a los cambios dinámicos que se puedan producir en el entorno del sistema de visión.

- Independencia dentro de la misma imagen entre la actuación de los algoritmos faciales y los algoritmos de detección de gestos.

Por lo tanto se ha optado por un primer desarrollo para detectar el movimiento en la imagen, que a la vez pueda servir para analizar el desplazamiento, tanto en dirección y sentido, como en su trayectoria. El segundo desarrollo realizado tiene que ver con la separación y extracción de la mano y de sus características para evaluar el gesto realizado.

El objetivo final es conseguir una envolvente poligonal de la mano con un centro de gravedad y sobre el que se pueda evaluar el movimiento, de modo que permita conocer trayectoria, posición, localización y desplazamiento, y utilizar estas características en el proceso de entrenamiento de la máquina de aprendizaje.

5.1 Identificación de movimiento

Uno de los objetivos fundamentales que se quieren implementar en URBANO es el aprendizaje dinámico de nuevos objetos. El objetivo no es implementar fuera de línea un sistema capaz de reconocer un conjunto definido de elementos del entorno. El interés marcado para nuestro desarrollo es crear en el robot la capacidad de reconocer aquello que aún no conoce y poder aprenderlo en ese momento. Ser capaz de modificar en tiempo real su base de conocimiento.

Para ello, el primer paso es identificar aquello que aún no es conocido. El planteamiento inicial es, que esta acción ocurra tal y como se produce en los humanos:

(1) Nos damos cuenta nosotros mismos e identificamos algo no conocido

(2) Alguien señala ese objeto y modifica nuestro punto de atención a la localización del mismo

Existe una larga tradición en el estudio de las secuencias de imagen en visión. Sin embargo en la actualidad, el núcleo de los desarrollos está menos centrado en la medida del movimiento en la imagen, y más en etiquetar la acción que tiene lugar en la escena. Este cambio está motivado no sólo por su baja carga computacional y de recursos, sino también por el interés de sus aplicaciones (como las interfaces wireless [90] o los entornos interactivos [91]).

Los humanos reconocemos el movimiento directamente, sin necesidad de realizar una reconstrucción del modelo tridimensional de las personas para después registrar la actividad. En la Fig. 63 se puede ver un ejemplo de ello. Las tres imágenes tienen una resolución muy baja, pero sí se puede distinguir un sujeto en movimiento al pasar la mirada de forma secuencial. Una de las imágenes por sí sola no representa nada reconocible, pero la secuencia de las tres pone de relieve las diferencias entre ellas, y nos transmite el movimiento. Lo que captamos es la diferencia entre las imágenes consecutivas y no en sí el análisis de singularidades en cada una.

Esta idea es la que ha servido como base a Aaron F. Bobick y James W. Davis [92] para utilizar el algoritmo MHI para la detección de movimiento en secuencias de vídeo.

Fig. 63 Tres secuencias de vídeo no consecutivas que representan un movimiento. Aunque la imagen no esté bien definida, al mirar la secuencia se puede reconocer el movimiento de alguien que se está sentando

5.1.1 Detección de movimiento con el Algoritmo MHI

El algoritmo MHI, del inglés *Motion History Images*, comienza construyendo una imagen binaria *motion-energy image* (MEI) que representa las regiones donde ha ocurrido movimiento con respecto a la secuencia anterior. Después se genera una imagen MHI que está constituida por píxeles cuyo valor es una función del movimiento sufrido. Por último, se utiliza un método de reconocimiento que desarrolla automáticamente una segmentación temporal.

5.1.1.1 Plantillas temporales para la detección del movimiento

El objetivo es construir una imagen que represente el desplazamiento como un movimiento a lo largo del tiempo. Se supone que el fondo de escena es fijo y sin cambios. Esta imagen se convierte en un vector que se puede comparar con plantillas almacenadas de movimientos conocidos, y ser utilizada como una plantilla temporal.

Motion-Energy Images (MEI)

Construimos la imagen MEI para representar dónde se produce movimiento en la imagen. Consideramos el movimiento asociado a la operación de sentarse tal y como se muestra en la figura siguiente:

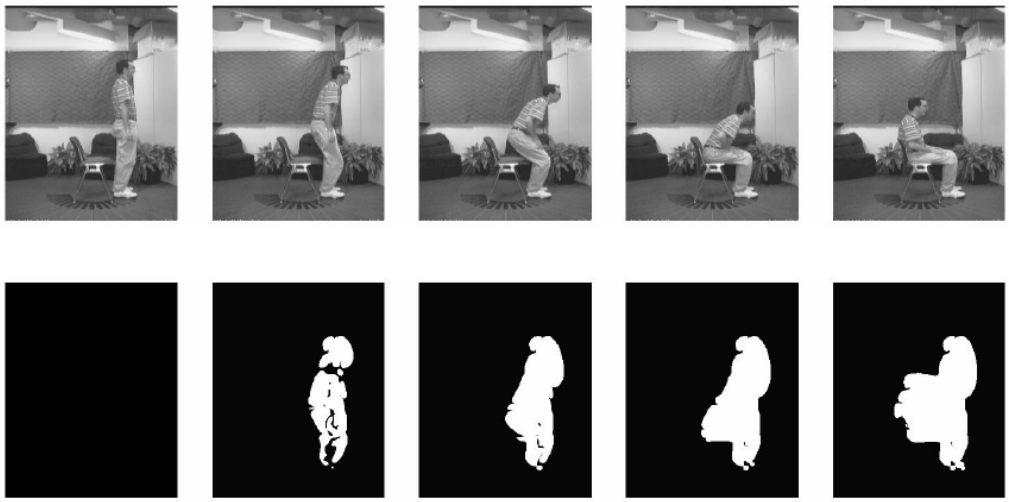


Fig. 64 Movimiento de una persona al realizar la operación de sentarse. La fila superior corresponde a cinco secuencias no consecutivas de la acción, y la de abajo son cinco imágenes de movimiento acumulado teniendo en cuenta todas las secuencias para su construcción.

Las secuencias correspondientes al movimiento realizan un barrido de una región particular de la imagen, aquella donde ha habido cambios. El objetivo es que la forma de esta región pueda ser usada tanto para detectar este movimiento, como el punto de vista o ángulo desde el que se ve.

MEI es la imagen binaria acumulativa del movimiento. Si consideramos una secuencia de imágenes $[I(x,y,t)]$ y una imagen binaria indicativa de los puntos donde ha habido cambios $D(x,y,t)$, la imagen binaria MEI estará definida de la siguiente manera:

$$E_\tau(x,y,t) = \bigcup_{i=0}^{\tau-1} D\left(x,y,t-i\right)$$

[Ec. 53]

La duración de τ es crítica en la definición del movimiento y de su extensión. Si por ejemplo está definida para 0.5 s., cualquier movimiento que se produzca con una duración menor no será detectado. Sin embargo, si la definición de este tiempo es de 2 s., , el más mínimo cambio en la secuencia da lugar a una alteración en la imagen MEI provocando que la forma de la región de movimiento sea mayor o incluso peor definida. El valor τ debe ser elegido como un compromiso entre lo que debe ser detectado, y el valor óptimo de la forma de la región de movimiento. En nuestro sistema de detección de movimiento, este valor es de $\tau = 358ms$.

Motion-History Images (MHI)

Construimos la imagen MHI para la representación de cómo es el movimiento en la imagen. La intensidad de cada píxel en su construcción será una función de la historia temporal del movimiento en ese punto. Uno de los posibles valores de esta función puede ser calculado con un operador de devaluación:

$$H_\tau(x,y,t) = \begin{cases} \tau & si \quad D(x,y,t) = 1 \\ \max\left(0, H_\tau(x,y,t-1) - 1\right) & otros \end{cases}$$

[Ec. 54]

Si un píxel tiene movimiento, es decir, la imagen $D(x,y,t)$ tiene un valor 1 en ese punto, el valor de la imagen MHI será el valor τ . En otro caso, tomará el valor máximo entre 0 y el valor anterior de esta imagen MHI menos un valor de devaluación. Esta devaluación se introduce para eliminar los movimientos antiguos. Visualmente genera algo parecido a un cerco de movimiento con una intensidad que se rebaja en función de su antigüedad. En nuestro sistema no cambia la intensidad: el cambio se mantiene en el píxel mientras permanezca dentro del umbral de historia permitido. En ese momento convierte su color directamente a cero (ver Fig. 65)

Fig. 65 Movimiento simple de una mano y las imágenes MEI (central) y MHI (derecha) que se generan.

5.1.1.2 Dirección y sentido del movimiento realizado

Adicionalmente, MHI lleva implícita la dirección y el sentido del movimiento. En la imagen anterior (Fig. 65) se puede apreciar el flujo del desplazamiento hacia la izquierda de una mano.

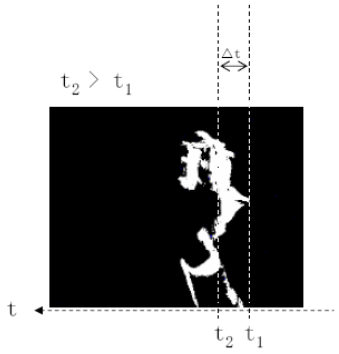


Fig. 66 La región blanca representa los cambios en una secuencia de imágenes durante un determinado tiempo. En un movimiento a la izquierda como el representado en la figura, los píxeles blancos más a la derecha representan una antigüedad mayor que los situados más a la izquierda.

Dado que el desplazamiento es hacia la izquierda, el movimiento será más "antiguo" en un píxel blanco cuanto más a la derecha se encuentra en la imagen MHI, como se puede ver en la Fig. 66. Para este tipo de movimientos individuales, sin interacción del fondo de la escena y cuando las oclusiones no son significativas, se puede representar la dirección del movimiento con la propia imagen MHI. Cuando los desplazamientos en la imagen son más complicados, es conveniente el uso de algoritmos dedicados tales como el flujo óptico. Son más complicados de implementar, pero la pérdida de la dirección del movimiento es menor.

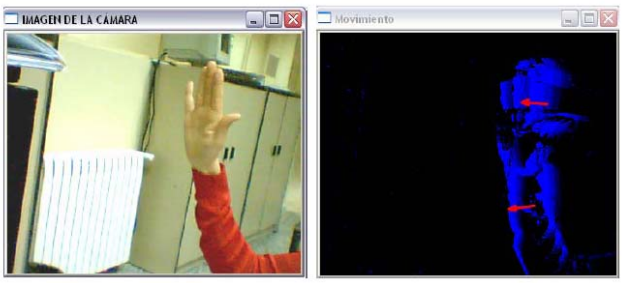


Fig. 67 Dirección del movimiento calculada con MHI

5.2 Método de identificación de gestos manuales

Para construir un sistema que identifique determinados movimientos conocidos, necesitamos determinar un algoritmo de comparación de píxeles, denominado *matching,* para la plantilla temporal MHI.

Las señales de la mano y del brazo constituyen otra forma de comunicación entre los humanos. Para que exista un entendimiento entre interlocutores, el sistema debe estar estandarizado. El código debe ser conocido dentro del grupo donde se van a utilizar las señales, de otro modo no es posible la transmisión del mensaje. Por lo tanto es necesario realizar un trabajo de definición previo de las trayectorias fijas identificadas para cada indicación gestual.

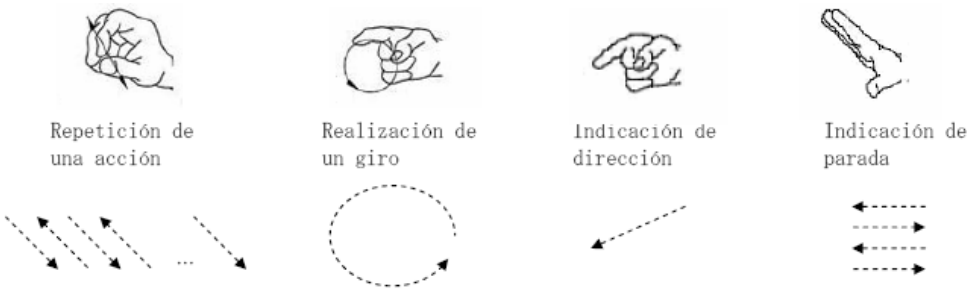


Fig. 68 Ejemplos de trayectorias fijas que definen indicaciones realizadas con las manos. En la parte inferior, se pueden ver las trayectorias de movimiento de cada mensaje. Estas trayectorias se almacenan y son usadas posteriormente para realizar el matching para la identificación del gesto.

Nuestro sistema en particular debe reconocer órdenes sencillas en el movimiento de la mano de su tutor. Estos movimientos ya estarán definidos para cada mandato que se van a poder dar al robot. Una vez segmentada la información referente a la trayectoria seguida en la imagen, se realizará la comparación mediante una aproximación basada en la apariencia. Es decir, una identificación de las partes invariantes de la plantilla de movimiento actual, con las trayectorias invariantes almacenadas.

5.2.1 Metodología para la extracción de trayectorias

Como es habitual en los sistemas de visión, el primer punto de trabajo consiste en separar la escena del elemento a evaluar.

En este caso nos interesan dos cosas: (1) la detección del movimiento y, (2) la trayectoria seguida. De momento nos vamos a centrar en ello sin determinar las señales –por otra parte necesarias- que indicarán comienzo y fin de la orden.

Es esencial el rastreo de una singularidad que se vea poco afectada ante los cambios dinámicos de la imagen. Si observamos el polígono imaginario que rodea la mano, se puede detectar una simetría.

Nos centramos en el polígono y no en el contorno de la mano porque nuestra necesidad es estudiar las trayectorias, y con esta aproximación tenemos la información que necesitamos. Así que podemos utilizar el centro de gravedad del poliedro que circunscribe la mano para rastrear su movimiento y convertir esa información en trayectoria.

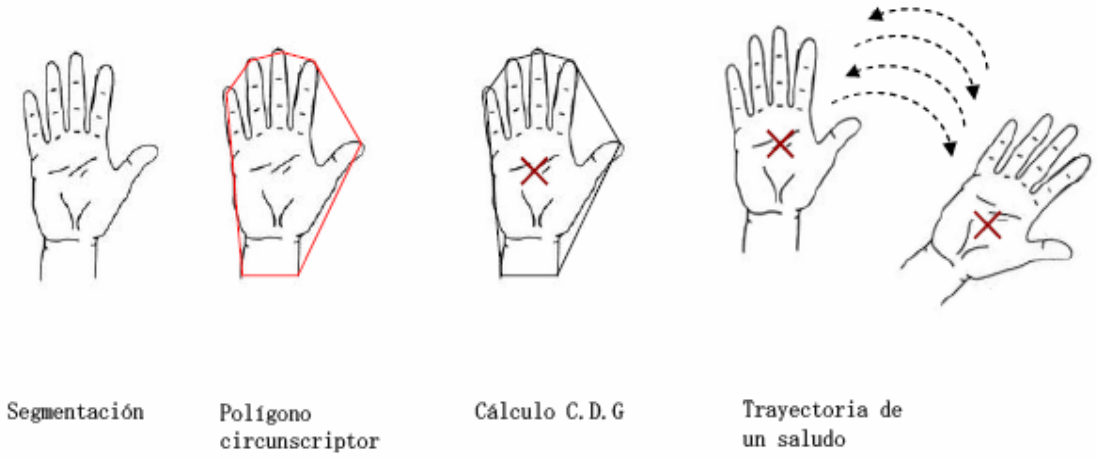


Fig. 69 Metodología de segmentación de la trayectoria del movimiento

5.2.2 Vector de características del gesto

En este epígrafe se expone el método de extracción de características para construir los vectores de características para el entrenamiento del sistema y la posterior identificación de gestos manuales. La máquina de aprendizaje utilizará estos vectores para implementar el conjunto de funciones que realizarán la toma de decisión en el proceso de identificación. A este tipo de aprendizaje se le denomina *aprendizaje a partir de muestras*.

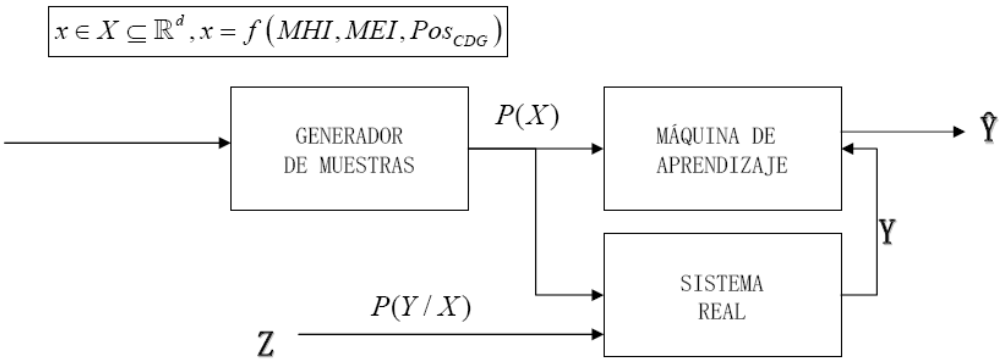


Fig. 70 Bloques de un sistema general de aprendizaje

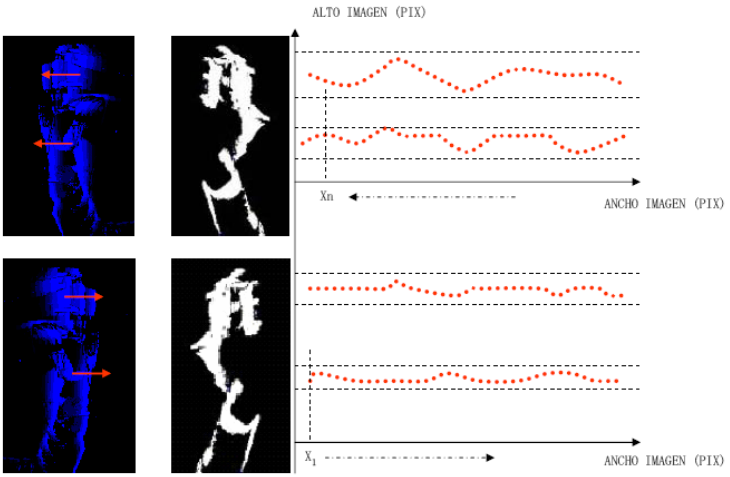
Tomando los trabajos de Cherkassky y Mulier [94] como referencia, se entiende por *aprendizaje a partir de muestras*, el proceso de estimar una dependencia desconocida entrada-salida de un sistema utilizando un número limitado de observaciones.

El modelo general se desarrolla a partir de un determinado número de vectores de entrada con una distribución probabilística determinada y un sistema de aprendizaje del que se obtiene un espacio de funciones. El sistema observa el conjunto de vectores de entrenamiento conociendo además la respuesta que debe dar el sistema, para construir a partir de ellos algún operador que sirva de vaticinador de respuestas del sistema ante entradas nuevas para las que no ha sido entrenado.

Desde esta perspectiva, se debe construir un vector de atributos o patrón, a partir de singularidades conocidas del movimiento. Se ejecuta el algoritmo de detección y se generan las imágenes MEI y MHI para cada movimiento. A partir de los datos obtenidos de las plantillas temporales se obtienen una serie de descripciones estadísticas basadas en características invariantes (por ejemplo los siete momentos

invariantes de Hu, M.Hu [93]). Esto, junto con los puntos de la trayectoria realizada por la mano constituirá el vector de características.

En la siguiente figura se pueden apreciar las singularidades de las trayectorias en un movimiento de la mano hacia ambos lados. La imagen MHI, segmentada en color azul, muestra el histórico del movimiento, dejando un color más intenso en los píxeles donde éste ha sido más reciente. En la imagen central, segmentada en color blanco, se percibe la forma completa que ha dibujado este movimiento durante el espacio temporal medido. En las gráficas que aparecen a la derecha, se muestran las trayectorias, medidas en píxeles sobre la imagen, que ha tenido el objeto en movimiento.



$$x \in X \subseteq \mathbb{R}^d, x = f\left(MHI, MEI, Posicion_{CDG}\right)$$

Fig. 71 Ejemplo de captura de puntos para el vector de entrenamiento. El movimiento de la mano genera una serie de posiciones en el sistema cartesiano de la imagen. La distribución estadística que generan es guardada en el vector **x** de entrenamiento, junto con otras características referentes a las plantillas MEI y MHI.

El problema reside en elegir la máquina de aprendizaje adecuada. Aquella que genere un conjunto de funciones con discrepancia mínima o que mejor aproxime la respuesta del sistema.

5.3 Algoritmos para la extracción de la trayectoria

En un problema general de segmentación de imágenes, el objetivo es localizar los diferentes objetos que están presentes en la escena. En un sistema de visión genérico, la fase de segmentación se encuadra entre la fase de *preprocesamiento*, en que las imágenes son sometidas a diferentes operaciones de filtrado que ayuden a mejorar la calidad de la imagen o a destacar los objetos que posteriormente queremos segmentar, y la de *identificación o reconocimiento*, que es la fase en que se identifican los distintos objetos y se etiquetan.

En general, la segmentación de la imagen supone una de las tareas más difíciles en el procesamiento visual. Este proceso condiciona el resultado del análisis posterior, por lo que la elección del método es fundamental para la consecución de los objetivos finales.

Existen diferentes tipos de segmentación asociados a cada problema a resolver. En nuestro caso se ha aplicado un método de segmentación del movimiento (sección 5.1 con el que se han obtenido muy buenos resultados cuando el objeto a segmentar se encuentra en movimiento y el resto de la escena y la cámara permanecen fijos.

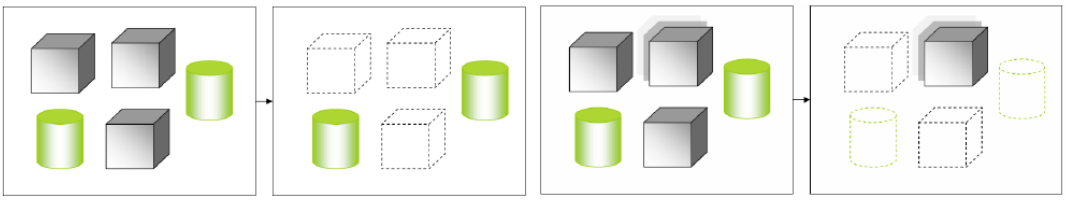


Fig. 72 Problema general de segmentación, a.) segmentación por color donde se elimina cualquier elemento de la imagen que no presente las características marcadas, b.) segmentación de los elementos que experimentan cambios de localización en una secuencia de imágenes.

Sin embargo tenemos situaciones que impiden que las características del entorno se mantengan fijas dando lugar a resultados en el algoritmo de la detección de movimiento, que no son válidos para los fines a que se quiere dedicar:

- Urbano lleva integrada la cámara a bordo, lo que provoca que el sistema detecte movimiento en cualquier situación, ya que la imagen cambia su estado continuamente.

- Las órdenes se van a expresar con la mano y sin embargo, al detectar el movimiento, se va a segmentar elementos adicionales como el brazo, el cuerpo y el movimiento de la cabeza como mínimo. En el momento de evaluar la trayectoria para identificar el gesto, esta situación no es válida.

- Aunque Urbano permanezca quieto, es posible que haya movimiento de personas en el fondo de la escena. Los algoritmos de detección facial tienen limitada su búsqueda sólo al objeto o cara más cercanos. Sin embargo, la detección del movimiento no desestima en función de la cercanía o lejanía del objeto, sino en función del cambio espacial, por lo que los obstáculos dinámicos ocasionan ruido y resultados no válidos para nuestro propósito.

El vector de características necesita unos patrones de movimiento de la mano para construir las plantillas MHI y MEI. Por otra parte, los puntos de la trayectoria van a ser los generados por el centro de gravedad del objeto en movimiento. Por lo tanto, necesitamos segmentar la mano en la imagen para poder extraer las características necesarias.

5.3.1 Segmentación automática por color de piel en HSV

En la sección 4.1.3 se utiliza el algoritmo Camshift para detectar las regiones de piel y realizar una segmentación redundante de la región facial. Recordamos que se construía un histograma a partir de la distribución de color de la región facial señalada por el algoritmo AdaBoost. Con este histograma se realizaba una reproyección sobre la imagen, dando lugar a una segmentación de las regiones correspondientes al color de la piel. En términos estadísticos, el valor de cada píxel en la imagen de salida caracteriza la probabilidad de que el correspondiente píxel en la imagen de entrada pertenezca al objeto cuyo histograma está siendo usado.

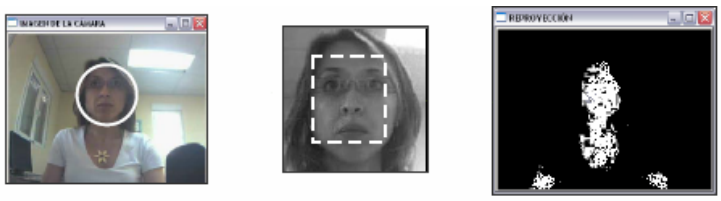


Fig. 73 Resultado de la segmentación del color de piel con el método de la reproyección del histograma construido a partir de la distribución de color de la región facial.

Las manos pertenecen al mismo grupo de distribución de color, con lo que se puede extender este resultado para añadir esta nueva región segmentada a la imagen. La diferenciación de las zonas de interés de cara y mano es trivial, puesto que en todo momento tenemos localizada la elipse facial.

5.3.2 Identificación del contorno de la mano

La mayor parte de los algoritmos de vectorización, es decir, algoritmos para encontrar contornos en la imagen, parten de imágenes binarias. Una imagen binaria contiene únicamente píxeles con valores 0 y 1. El conjunto de valores *0-conectados* y *1-conectados* forman el *0-(1-)componente conectado*.

Existen dos tipos de conectividad: *Conectividad-4* y *Conectividad-8*. Dos píxeles de coordenadas (x',y') y (x'',y'') se dice que tienen *Conectividad-4* si:

$$|x'-x''|+|y'-y''|=1$$

[Ec. 55]

Y se dice que tienen *Conectividad-8* si:

$$\max(|x'-x''|+|y'-y''|)=1$$

[Ec. 56]

Estas relaciones se pueden ver gráficamente en la Fig. 74

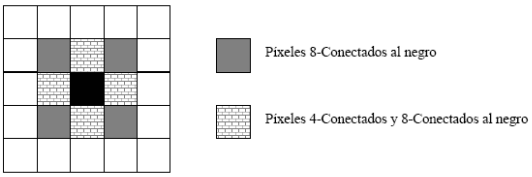


Fig. 74 Patrones de conectividad de píxeles

Utilizando este método de relación, la imagen se fracciona en varios componentes no superpuestos. Cada conjunto consiste en un grupo de píxeles con el mismo valor, 1 o 0, y cada par de píxeles dentro de un componente determinado puede quedar unido por una secuencia de píxeles con *conectividad-4* o *conectividad-8*. En otras palabras, rutas de 4-8 píxeles entre ambos puntos.

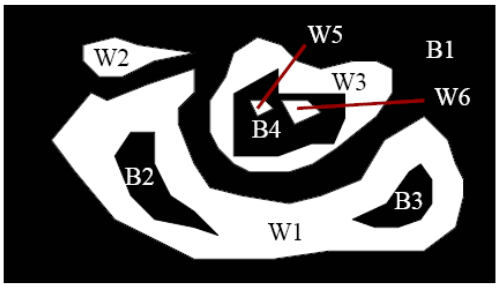


Fig. 75 Componentes jerárquicamente conectados

Por otra parte, estos componentes pueden tener sus propias interrelaciones:

- Los componentes-1 de las regiones W1, W2 y W3 están dentro del marco B1 (componente-0). Esta completamente rodeado por B1.

- Los componentes-0 de las regiones B2 y B3 están dentro de W1.

- Los componentes-1 de las regiones W5 y W6 están dentro de B4, y ésta dentro de W3, así que W5 y W6 están también dentro de W3. Sin embargo ni W5 ni W6 se cierran entre sí, lo que quiere decir que se encuentran al mismo nivel.

Para evitar posibles contradicciones topológicas se asigna *conectividad-8* a los componentes-1 y *conectividad-4* a los componentes-0. Este es el convenio adoptado

por las librerías *OpenCV* utilizadas en este trabajo. Otra de los ajustes que se dan en esta librería es adoptar como fondo de escena los componentes-0, asumiendo que los componentes-1 son la estructura topológica de trabajo. Un componente-0 rodeado completamente por un componente-1 se denomina *"agujero"* del componente-1. Un *"punto frontera"* de un componente-1 podría ser cualquier píxel que pertenezca al entorno de otro componente-0 cercano. Muchos *"puntos frontera"* dan lugar a un *"borde"*.

El conjunto de *"bordes"* definidos también como *"contornos"* de todos los componentes almacenados con información sobre su jerarquía dan lugar a una representación comprimida de la imagen binaria original.

La función de la librería OpenCV utilizada para la búsqueda de contornos dentro de la imagen es *cvFindContours* con la opción de búsqueda de listar todos los contornos sin tener en cuenta jerarquías o propiedades límite. Esta parte queda indicada con la macro CV_RETR_LIST preparada para tal fin. Esta macro realiza las tareas de conexión explicadas en párrafos anteriores.

5.3.3 Método Freeman para representar el contorno

Es el método de representación elegido para representar el contorno, también llamado *"código de cadena"*. Se enumera la vecindad de cada píxel del 0 al 7, tal y como se ve en (Fig. 76). El píxel número cero denota el que tiene la posición más a la derecha del píxel considerado y así se forma la secuencia. Como una secuencia de 8 puntos conectados, el borde puede ser almacenado como las coordenadas del punto inicial seguido de un código formado con los caracteres del 0 al 7 que representan la siguiente localización relativa al contorno.

Se trata de una representación compacta de curvas digitales que facilita la representación poligonal posterior.

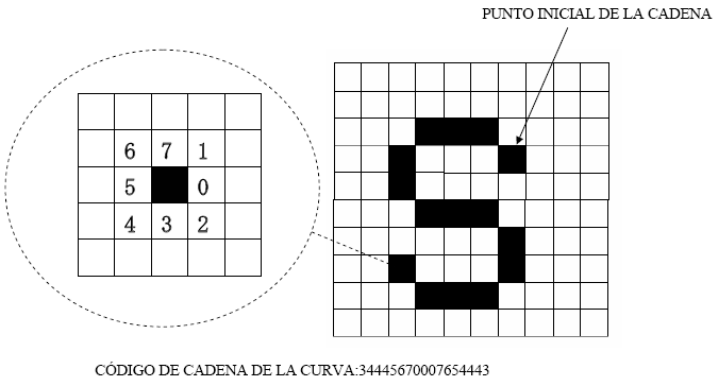


Fig. 76 Representación de contornos por el método Freeman

La aproximación poligonal del código de cadena es una opción en que la curva se codifica como una secuencia de puntos, vértices y múltiples líneas para poder analizar y manipular convenientemente los contornos. En nuestro caso se ha optado por una aproximación simple consistente en una compresión vertical, horizontal y diagonal de modo que se queda únicamente con los extremos finales.

El resultado que se obtiene de aplicar esta propuesta es el que se muestra en la siguiente figura, donde se puede apreciar la zona sementada de la mano rodeada por el contorno calculado.

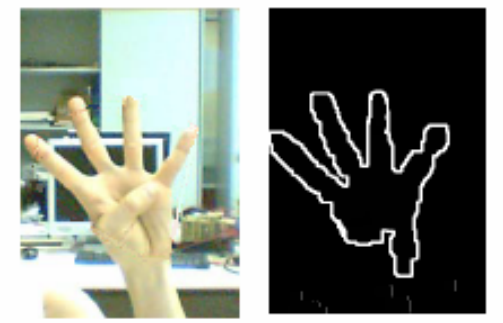


Fig. 77 Resultado de la segmentación realizada para sobre la mano

No obstante, este resultado está muy condicionado por los cambios dinámicos referentes a la iluminación de la escena.

5.3.4 Características de interés dentro del contorno de la mano

El objetivo de este procedimiento es la búsqueda de un polinomio convexo que enmarque completamente la región de la mano y que permita evaluar cambios en su forma dependiendo de la posición que tengan los dedos. Se podría trabajar con un algoritmo más sencillo, como la obtención del rectángulo que circunscriba el contorno. Pero eliminaríamos la capacidad de analizar los casos en que los dedos, o la mano en sí misma, adopten posiciones que puedan dar lugar a diferentes órdenes gestuales.

El método por el que se ha optado en este proyecto es la búsqueda de singularidades fuertes tipo "*esquina*" en el área de interés, es decir, un método en el que se realiza la búsqueda de esquinas con mayores valores propios asociados. Para ello se utiliza la función de la librería OpenCV, *GoodFeaturesToTrack* que calcula el mínimo valor propio para cada píxel de la imagen de entrada y desestima aquellos que no superen un valor de calidad mínimo que se utiliza de umbral. La función asegura que todas las esquinas encontradas tienen entre ellas una distancia mínima de al menos dos características fuertes. Los puntos que no cumplen con esta condición de distancia se desestiman. Posteriormente se incluye un algoritmo de localización subpíxel para las características estimadas.

La precisión subpíxel está basada en la observación de que cualquier vector desde el centro de un píxel a cualquier punto de su entorno de vecindad es ortogonal al gradiente de la imagen en ese punto del entorno de vecindad. El método consiste en calcular el centro de la ventana del entorno de vecindad en un punto que cumpla con esta característica.

Matemáticamente:

$$\varepsilon_i = \nabla I_{p_i}^T \cdot (q - p_i)$$

$$\nabla I_{p_i}^T \rightarrow \textit{gradiente de la imagen}$$

$$\varepsilon_i \rightarrow \textit{factor a optimizar}(\min)$$

[Ec. 57]

El valor de q se obtiene de tal forma que el factor a optimizar se minimice:

$$(\sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T)q - (\sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T \cdot p_i) = 0$$
$$\sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T \rightarrow G$$
$$\sum_i \nabla I_{p_i} \cdot \nabla I_{p_i}^T \cdot p_i \rightarrow b$$

[Ec. 58]

Los gradientes se suman en un entorno de vecindad o ventana de búsqueda de q:

$$q = G^{-1} \cdot b$$

[Ec. 59]

De este modo se obtiene la precisión para cada uno de los puntos esquina que se han detectado en la imagen

Fig. 78 Resultado de la búsqueda de las características de interés en la región de la mano.

5.3.5 Aproximación polinómica con el método Convex hull

Según la enciclopedia Wikipedia, el término matemático *convex hull* también denominado *convex envelope,* de un conjunto de puntos X en un espacio vectorial real V, es el mínimo conjunto convexo que contiene a X.

En geometría computacional es muy común utilizar el término para denominar los límites de un conjunto convexo mínimo que contiene a un grupo finito de puntos en el plano. A menos que estos puntos sean lineales, *convex hull* tiene el aspecto de una simple cadena poligonal cerrada.

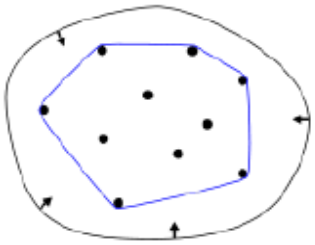


Fig. 79 Analogía de la goma elástica para el *convex hull*

En sistemas planares como las imágenes, se puede visualizar imaginando una goma elástica que engloba todos los puntos.

El *convex hull* supone la forma con área mínima enmarcada por la goma y que recoge todos los puntos del conjunto X. Para que exista el *convex hull* de un conjunto de puntos en un espacio vectorial V, se debe verificar que X esté contenido en al menos un conjunto convexo, aunque este conjunto convexo sea el propio espacio V, y cualquier intersección de otros conjuntos convexos que contengan a X sea también un conjunto convexo que contiene a X.

Por lo tanto, *convex hull* es la intersección de todos los conjuntos convexos contenidos en X. Más formalmente, el *convex hull* de un conjunto X se define como el conjunto de combinaciones de subconjuntos convexos de puntos de X, es decir, el conjunto de puntos de la forma:

$$\sum_{j=1}^{n} t_j \cdot x_j$$

[Ec. 60]

donde n es un número natural arbitrario, los t_j son números no negativos de suma total igual a 1 y los x_j los puntos del conjunto X.

Entonces:

$$H_{convex}(X) = \left\{ \sum_{i=1}^{k} \alpha_i x_i \quad \middle| x_i \in X, \alpha_i \in \Re, \alpha_i \geq 0, \sum_{i=1}^{k} \alpha_i = 1 \quad k = 1,2,... \right\}$$

[Ec. 61]

Si X es un subconjunto de un espacio vectorial N-dimensional, las combinaciones convexas de al menos N+1 puntos son suficientes en la definición anterior. Es lo mismo que afirmar que *convex hull* es la unión de de todos los *Simplex* con al menos N+1 vértices de X.

Esto es conocido también como el *Teorema de Carathéodory*.

En nuestro desarrollo se han utilizado las características de interés encontradas dentro de la región de la mano, como los puntos del conjunto X dentro del espacio vectorial V de los puntos de la imagen.

Por lo tanto, el algoritmo trabaja con los puntos que han sido obtenidos en la detección de singularidades fuertes dentro del área de interés marcada por el color que se ha desarrollado previamente, y serán los responsables de formar la envolvente convexa de la mano.

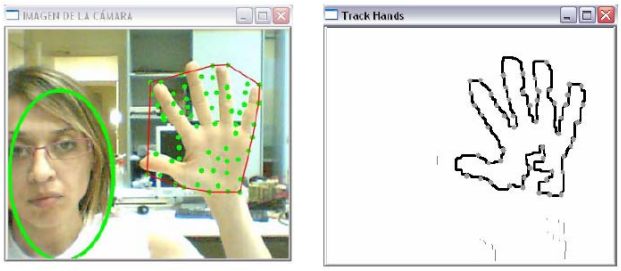


Fig. 80 Aplicación del método *convex hull*

En la imagen se presenta la forma de la envolvente convexa que define un volumen del objeto de interés. Como ya se ha comentado anteriormente, existen métodos para definir áreas y bordes en objetos, pero no dan un resultado tan contundente ni tan flexible en la aportación de características como este.

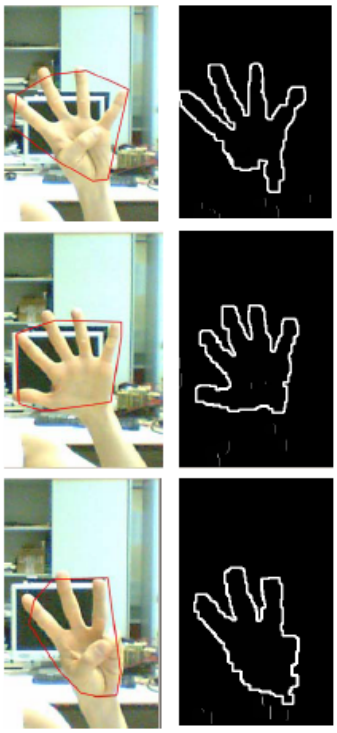


Fig. 81 Resultados del algoritmo *convex hull* para diferentes posiciones

En la Fig. 81 se puede apreciar el cambio que se produce en la envolvente para cada posición adoptada. Esto nos permite una diferenciación importante en la definición del gesto. Adicionalmente se propone el cálculo del centro de masas para una localización mejor del polígono.

5.3.6 Centro de masas de la envolvente convex hull

El conjunto de puntos que han servido para el cálculo de la envolvente, son una distribución discreta de probabilidad en una imagen 2D. Por lo tanto la localización media o *centroide* del polígono se calcula a partir de los momentos de orden cero y uno.

Momento de orden cero:

$$M_{00} = \sum_x \sum_y I(x,y)$$

[Ec. 62]

Momentos de orden uno:

$$M_{01} = \sum_x \sum_y y \cdot I(x,y)$$
$$M_{10} = \sum_x \sum_y x \cdot I(x,y)$$

[Ec. 63]

El centro de masas está localizado en un punto (x_c,y_c) tal que:

$$x_c = \frac{M_{10}}{M_{00}} \quad ; \quad y_c = \frac{M_{01}}{M_{00}}$$

[Ec. 64]

Y el resultado en la imagen de salida es el que se presenta en la siguiente figura.

Fig. 82 Centro de gravedad del polígono *convex hull*

5.4 Síntesis del método propuesto

El sistema aprovecha la circunstancia del cálculo dinámico de la distribución de color de la piel para segmentar inicialmente la mano. Se extraen las características de interés del área segmentada, con el objetivo de tener un subconjunto de puntos que puedan servir para el cálculo de una envolvente de la mano, que constituye una figura geométrica de mejor análisis en cuanto a movimiento y posición que la segmentación en sí misma. Se puede ver que los algoritmos de detección facial y los de detección de gestos trabajan de forma paralela sin interacción de uno sobre otro.

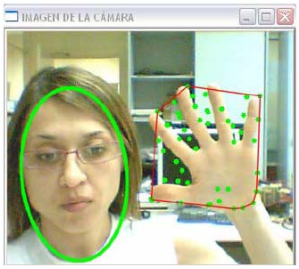


Fig. 83 Trabajo conjunto de los algoritmos de detección facial y de extracción de características de la mano.

El diagrama de flujo que se muestra en la página siguiente, presenta el algoritmo de extracción de características propuesto (línea continua), incluido en el algoritmo general de detección del sistema (línea discontinua).

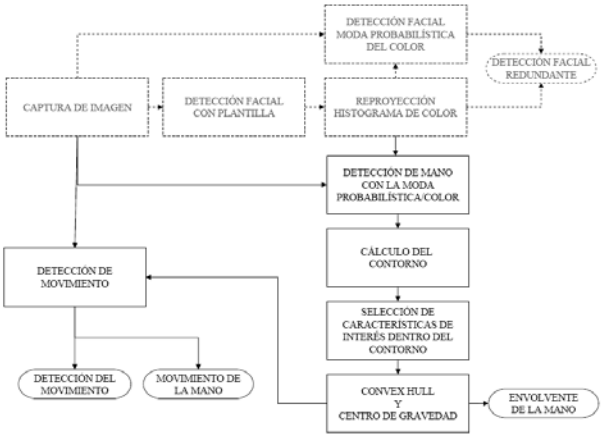


Fig. 84 Diagrama de flujo del sistema de extracción de características. La línea discontinua corresponde al algoritmo inicial de detección facial y extracción de la moda probabilística que posteriormente utiliza el algoritmo de extracción de características propuesto, con línea continua.

5.5 Clases C++ dedicadas

5.5.1 Clase Process_Vision: Detección de movimiento

Esta clase está dedicada a las principales funciones de procesamiento de la imagen que sirven como base al sistema de visión. Se describe a continuación el trabajo realizado por aquellas dedicadas a la detección del movimiento en la imagen y a la función que llama al algoritmo de extracción de características gestuales.

5.5.1.1 DetectaMovimiento

– void DetectaMovimiento (IplImage* ImgMov)

Función que realiza las llamadas al algoritmo de detección de movimiento dentro de la imagen. El objetivo es mostrar los elementos que han sufrido algún cambio en el transcurso de las cuatro últimas secuencias.

Esta función comprueba que la cámara está enviando imágenes y llama al algoritmo MHI mediante la función *update_mhi*

5.5.1.2 update_mhi

– void FaceReference (IplImage* img, IplImage* dst, int dic_threshold)

Algoritmo de detección de movimiento en la imagen mediante la diferenciación entre las cuatro últimas secuencias enviadas por la cámara. La variable *timestamp(τ)* del algoritmo está definida para la obtención del tiempo actual en segundos.

Se ha optado por trabajar con un número de imágenes de cuatro, pero este número puede ser cambiado en la variable global *N* del programa. Esto siempre tiene que ser resultado de un estudio del movimiento a detectar y del tiempo en su transcurso por parte de los desarrolladores. En nuestro caso, cuatro secuencias de vídeo nos ofrecen buenos resultados y no sobrecarga el sistema de cálculo.

Además, en esta función se ha incluido la creación de las imágenes con fondo negro y píxeles en diferentes intensidades de azul dependiendo de la antigüedad del cambio. En el procedimiento calcula la diferencia entre las dos últimas secuencias recibidas, umbraliza el resultado para binarizar la imagen y posteriormente analiza el cambio y su antigüedad.

Por último calcula los gradientes en la imagen resultado para dar el centro de movimiento de cada segmento de la imagen y su dirección.

5.5.1.3 TrackHands

– void TrackHands (void)

Función que llama al algoritmo de seguimiento de la mano. Comienza evaluando las variables de detección facial. Sólo cuando la detección está realizándose, se obtiene el histograma necesario para esta función, y es cuando se llama al algoritmo.

La función separa los canales de la imagen para trabajar con el canal H del espacio de color HSV. Posteriormente realiza el cálculo de los lugares de la imagen donde va a ser localizada la mano, para evitar errores ante posibles oclusiones con la cara. Se realiza un filtrado piramidal para disminuir el ruido en la imagen y se llama a la función que crea la secuencia de puntos para el algoritmo *convex hull*. Esta función pertenece a la clase *ConvexHullClass*.

5.5.2 Clase ConvexHullClass: Envolvente de la mano

Esta clase está dedicada al cálculo del subespacio vectorial de puntos X dentro del espacio vectorial V de la imagen, para la obtención de la envolvente convexa de la mano.

5.5.2.1 m_CreateSequence

- Void m_CreateSequence(IplImage* Img, IplImage* ImgOriginal)

Función que determina los contornos pertenecientes al área definida para la búsqueda de una mano, en la imagen reproyectada a partir del histograma de color. Posteriormente determina las características de interés que se encuentran dentro del contorno hallado y elimina los que se quedan fuera. Una vez que tiene determinada la secuencia de puntos llama a la función *ConvexHullFunction*.

5.5.2.2 ConvexHullFunction

- void ConvexHullFunction (CvSeq* ptseq, IplImage* Imagen, IplImage* ImgOriginal)

Se determina la envolvente convex hull para los datos recibidos en la secuencia de puntos. Saca por pantalla el resultado dibujado sobre la mano de la imagen inicial de la cámara. Posteriormente calcula del centro de gravedad de la distribución de puntos.

CAPÍTULO 6

Interpretación semántica y aprendizaje

En la actualidad tenemos una variedad muy amplia de algoritmos de visión por computador que ofrecen soluciones más o menos satisfactorias en condiciones de entorno muy restringidas. De unos años a esta parte ha surgido, sin embargo, el anhelo de interpretar la imagen desde una perspectiva cognitiva. Analizar una imagen desde el punto de vista de la algoritmia, no supone más que descifrar y evaluar singularidades en la escena de elementos que previamente han sido analizados en el entorno real y en su proyección. Pero dar sentido a los resultados de ésta búsqueda es otra cuestión. Si tenemos una imagen de fondo negro con un centelleo de puntos blancos, y tenemos conocimientos de procesamiento de la imagen, podríamos pensar que se trata de una segmentación de algún tipo de objeto que se presenta en grupos, como células por ejemplo. Si esta misma imagen la ve un astrónomo, puede identificar formas adicionales en la colocación del serpenteo de puntos blancos e identificar una galaxia, con lo que las células dejarían de ser células para convertirse en estrellas.

Esto que a primera vista puede resultar un desatino, es la realidad de lo que ocurre al interpretar una imagen. De forma inconsciente incluimos nuestro propio conocimiento para dar solución a lo que percibimos.

Una de las soluciones que se han comenzado a imponer en el estado del arte es el uso de ontologías orientadas a conceptos visuales. Un intento de adquirir y utilizar el conocimiento sin la necesidad de recurrir continuamente a niveles matemáticos.

Una propuesta de utilizar ontologías para eliminar de algún modo el acceso continuo a los niveles más básicos de la visión por computador, y guiar al experto en la descripción de los elementos de su dominio.

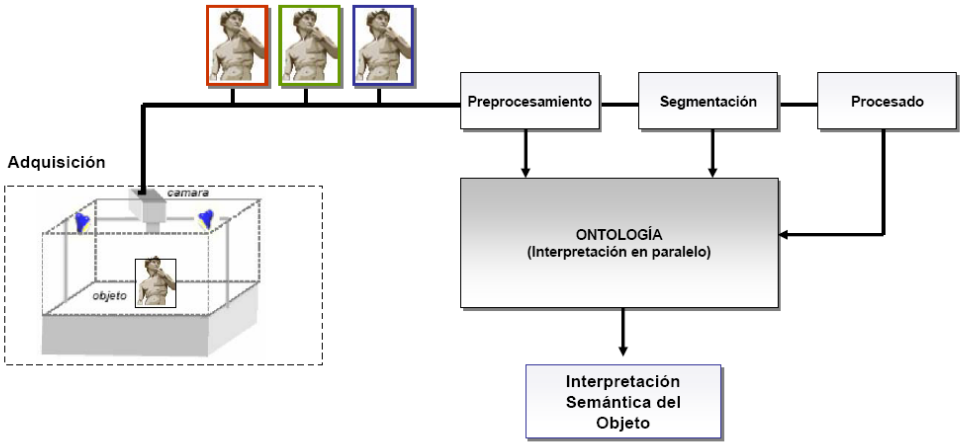


Fig. 85 Esquema general de un sistema de visión utilizando ontologías de conceptos visuales, para sustituir la decisión a nivel puramente matemático.

Desde esta nueva perspectiva de adquisición de conocimiento se trabaja con la realidad percibida y sus singularidades a la vez que con su interpretación cognitiva.

En otro orden de aplicación, un rasgo que generalmente está asociado a la inteligencia es la capacidad de adquirir conocimiento, W. Fritz [132]. Esto se manifiesta en procesos de aprendizaje que aceptan ser descritos en términos de entendimiento e incorporación de la información que se extrae de un determinado contexto. Según W. Fritz, un *sistema inteligente autónomo* puede definirse como aquél capaz de descubrir y registrar si una acción efectuada sobre una situación dada fue beneficiosa para lograr su objetivo.

El objetivo futuro del sistema de visión de la plataforma Urbano es el desarrollo de un aprendizaje automático basado en visión. Para aprender en un dominio real, necesitamos formular alguna teoría acerca de los efectos que producen las acciones sobre el entorno. Es decir, construir planes y monitorizar la ejecución de dichos planes para detectar expectativas no cumplidas y diagnosticar o rectificar errores.

El desarrollo de las ontologías permite compartir el entendimiento común de las estructuras de información (entre personas o entre agentes software), la reutilización de estos conocimientos, manifestar suposiciones sobre el dominio, separar el conocimiento de dicho dominio del conocimiento operacional y analizar dicho conocimiento.

6.1 Ontología de conocimiento

Las ontologías inicialmente estuvieron ligadas a conceptos filosóficos y metafísicos donde adquirieron el significado de "*filosofía del ser"* (ontos=being y logos=treatise). El término comienza a ser relevante en el campo de estudio de la *Ingeniería del Conocimiento* (KE) entendiéndose como una explicación sistemática de la existencia.

En este contexto fue definida por Gruber: "*ontología es especificación explícita de una conceptualización* [96],[97]". Una definición que la *Inteligencia Artificial* (AI) adoptó muy rápidamente.

N. Guarino [99], define la ontología como una teoría lógica que justifica el significado planteado para un vocabulario, es decir, un compromiso ideológico con una concepción particular del mundo.

La finalidad de la ontología es definir qué primitivas, con su semántica asociada, son necesarias para la representación del conocimiento en un contexto dado. Está formada por varias entidades:

- Un conjunto de conceptos C (por ejemplo geométricos)
- Un conjunto de relaciones R (espaciales por ejemplo)
- Un grupo de axiomas (transitividad, simetría, etc)

Se dispone también de dos operadores de relación [$\leq, \geq$] que definen la jerarquía de los conceptos y de las relaciones. La ontología va a ser el soporte de los mecanismos de razonamiento.

Una comunicación eficiente entre el usuario y la máquina exige un entendimiento, y este entendimiento exige un lenguaje compartido. La dificultad de un lenguaje compartido sin embargo, está en la identificación de requerimientos, y en los límites de operatividad. Las ontologías son una base sobre la que construir una referencia compartida que se obtiene a partir de un consenso denominado *compromiso ontológico*.

Tal y como se explica en [102] el proceso de desarrollo de una ontología conlleva varias fases:

- *Especificación:* Motivación por la que se construye la ontología y quiénes son los usuarios finales.

- *Conceptualización:* Es el modelo abstracto de algún fenómeno del mundo construido a partir de conceptos relevantes identificados en dicho fenómeno. Se puede decir que supone el dominio de conocimiento.

- *Formalización:* Implica que la ontología debe ser legible para las máquinas. Transforma el modelo conceptual en un modelo formal.

- *Implementación:* Parte de la base de que la ontología captura el conocimiento aceptado por un grupo, entendiendo que el grupo es un conjunto de agentes software, o más generalmente, una colección de agentes basados en conocimiento (humanos o computacionales) que utilizan este conocimiento compartido durante el funcionamiento del sistema o en base a un proceso de construcción de agentes. En la implementación se transforma el modelo formal en un modelo computacional.

El término "*conceptualización*" fue definido originalmente por M.R. Genesereth y L. Nilsson [98], como un conjunto de relaciones extensibles que describen un estado particular. Para precisar esta *conceptualización*, concluyeron que la ontología debía estar formada por clases, instancias, funciones, relaciones y axiomas.

Las ontologías se encuentran aún sumidas en un proceso de controversia y debate tanto en el campo de la *Inteligencia Artificial* (A*rtificial Intelligence, AI*) como en el de la *Ingeniería del Conocimiento* (*Knowledge Engineering, KE*). Cada definición atiende un punto de vista diferente.

Sin embargo, pueden ser vistas como un vocabulario descrito en términos de un dominio o de una tarea. La clave, sin embargo, no es la ontología en sí misma como vocabulario, sino el significado y compromisos subyacentes.

El significado, las relaciones, restricciones y axiomas de los términos de la ontología deben construir un conjunto de términos pertenecientes al mismo campo semántico, y convertirlo en una ontología.

6.1.1 Concepto de Agente

Un *agente inteligente* lo define M. Wooldridge y N. R. Jennings [100] como un sistema informático que presenta una serie de propiedades como autonomía, habilidades sociales, comportamientos reactivos y adaptación, y que está implementado utilizando conceptos que generalmente se atribuyen a los humanos.

Los agentes pueden ser:

(1) *Racionales*, actuando sobre su propio entorno y tomando decisiones para su propio bienestar

(2) A*utónomos*, que deben conseguir por sí mismos los objetivos y la toma de decisiones, de forma libre e independiente.

(3) *Móviles*, que se desaplazan libremente por las redes electrónicas, comunicando con otros objetos del entorno tales como recursos de información u otros agentes

(4) *Cognitivos*, cualquier agente que explote conocimiento explícito. Los agentes, por lo general, son heterogéneos en cuanto a capacidades y objetivos.

6.1.2 Motivación del uso de ontologías y agentes

Según J. Bermejo, R. Sanz e I. López [95], durante los últimos años las *ontologías* y los *agentes* vienen siendo dos áreas de estudio entrelazadas. Se han comenzado a desarrollar *ontologías* enfocadas a aplicaciones basadas en *agentes*. Y por su lado, los *agentes* se benefician de las *ontologías* en lo referente a la obtención de procesos basados en información, en las tareas orientadas al conocimiento y especialmente cuando el conocimiento involucrado constituye el fundamento del sistema, como ocurre por ejemplo con los sistemas distribuidos de control.

La *Ingeniería del Conocimiento* (KE) y *la Inteligencia Artificial* (AI) han comenzado a dirigir sus pasos al uso y desarrollo de ontologías para mejorar los sistemas basados en conocimiento, ya que permiten definir los conceptos y las relaciones entre ellos dentro de un dominio de interés. El objetivo no es determinar qué es y qué no es, sino concretar estos conceptos para los sistemas de conocimiento y los computadores.

Las ventajas del uso conjunto son varias, pero se pueden destacar algunas de ellas extraídas del artículo de J. Bermejo, R. Sanz e I. López [95].

- Las ontologías clarifican los sistemas de conocimiento: realizan un análisis de un determinado dominio, y permiten definir un vocabulario efectivo, unas suposiciones, y la conceptualización subyacente. Asimismo, permiten separar el dominio de conocimiento, del problema operacional del sistema.

- Ayudan en la escalabilidad del conocimiento: el análisis del conocimiento puede dar lugar a una base fundamental muy amplia. Las ontologías codifican y manejan este conocimiento de un modo escalable, facilitando así la posibilidad de aprendizaje.

- Permiten, tanto a usuarios como a otros agentes, compartir y reutilizar el conocimiento mediante la asociación de términos, relaciones y conceptos, así como por la sintaxis para codificar el conocimiento.

- Incrementan la robustez de los sistemas basados en agentes, dado que los estos agentes pueden describir las relaciones ontológicas y razonar ante eventos imprevistos de su dominio. Es decir, ofrecen la posibilidad de obtener autonomía en la modificación del sistema implementado.

- Las ontologías proveen unas bases de interoperabilidad entre agentes.

- Las que se centran en el dominio de la ingeniería del software para sistemas basados en agentes, proporcionan ayuda tanto en los equipos de desarrollo como en procesos software, e incluso durante las etapas de explotación se comportan como soporte de entendimiento cognitivo e integración de agentes, incluyendo capacidades reflejas de cognición.

6.2 Interpretación semántica de la imagen

En ésta sección se presenta una solución para categorizar los objetos involucrados en los aspectos de aprendizaje, reconocimiento y representación del conocimiento dentro de un sistema de visión cognitiva. Los fundamentos que se presentan tienen su base en la propuesta realizada por Nicolas Eric Maillot y Monique Thonnat [101].

En un sistema basado en visión se pueden identificar varios tipos de conocimiento:

(1) El dominio del conocimiento.

(2) La relación entre el dominio del conocimiento y el conocimiento del procesamiento de la imagen, denominado también *conocimiento de mapeo*.

(3) El conocimiento derivado del procesamiento de la imagen.

6.2.1 Ontología basada en visión

El objetivo de la ontología visual tiene que ver con la relación entre el dominio de conocimiento y el resultado de procesar la imagen, también denominado *mapeo*. Como se ha comentado en apartados anteriores, la extracción del dominio supone la construcción de una estructura jerárquica de clases que representan a los objetos del entorno de trabajo con sus subpartes (Fig. 86).

Es importante recalcar que el dominio de conocimiento es independiente de los niveles involucrados en la visión y que, en consecuencia, puede ser reutilizado para otros fines. Pertenece al escenario de trabajo y lo comparten los especialistas en ese contexto. La ontología va a suponer una directriz que proporcione el vocabulario necesario para la descripción del dominio. En nuestro caso concreto, la descripción visual o la interpretación semántica de la imagen.

En torno a este objetivo, la estrategia va a consistir en aproximar el modelo a determinados procedimientos semánticos donde, los conceptos visuales y sus

etiquetas, estén asociados. Esto, en el sentido de tener definidas varias formas de acceder a ellos, con los mismos procesos de segmentación y extracción de características habituales.

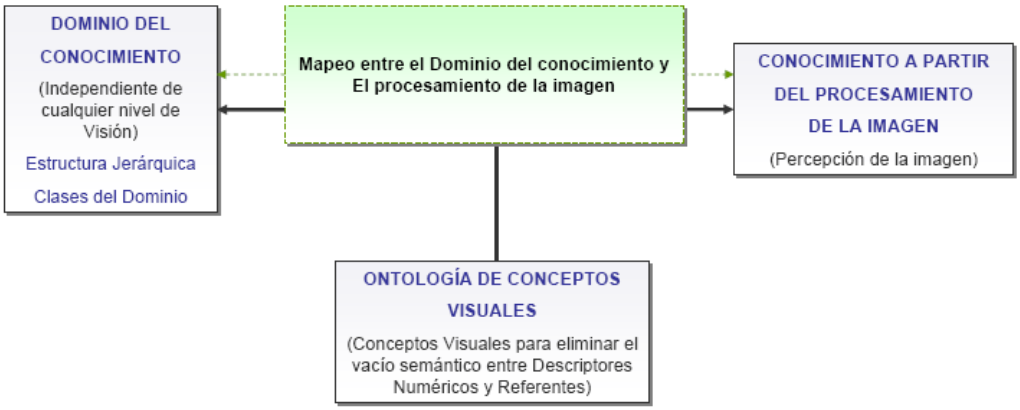


Fig. 86 Mapeo entre el dominio del conocimiento y el sistema de visión por computador con una ontología basada en visión cognitiva.

El proceso de construcción lo constituyen tres fases:

(1) Construcción de un sistema de categorización de objetos mediante la adquisición de conocimiento, obteniendo así un método concluyente e inequívoco para reconocerlos posteriormente.

(2) Un proceso de aprendizaje. Generalmente se trata de un entrenamiento fuera de línea, no un aprendizaje automático que, por otra parte, sería lo deseable.

(3) Utilización de la ontología para inferir el reconocimiento.

6.2.1.1 Adquisición del conocimiento

El experto comienza creando un dominio del conocimiento que se estructura como una jerarquía de dominio de clases.

Posteriormente se hace frente a una fase de descripción, donde se especifica el modo en que la ontología va a dirigir el proceso de adquisición. Es decir, el experto utiliza el vocabulario que provee la ontología, para denominar a los elementos del dominio. El resultado es una base de conocimiento formada por conceptos visuales provenientes de la ontología asociada al dominio de clases.

El resultado es la obtención de una *base de conocimiento* y un *etiquetado* de la imagen, es decir, una marca asociada a un conjunto de imágenes previamente procesadas. Es un procedimiento que generalmente se realiza a mano.

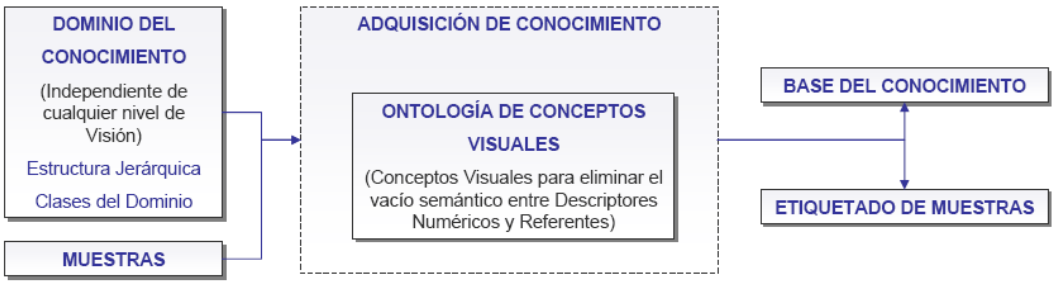


Fig. 87 Proceso de adquisición del conocimiento: La ontología de conceptos visuales guía el proceso. Durante su desarrollo se provee al sistema de muestras representativas de las imágenes que van a ilustrar el dominio del conocimiento.

6.2.1.2 Aprendizaje de conceptos visuales

Durante el proceso de adquisición de conocimiento, fue necesaria la figura del experto y un conjunto muestral de imágenes. La función del proceso de aprendizaje es completar el vacío que se queda entre la fase de adquisición y la fase de procesamiento y etiquetado manual.

El aprendizaje de un objeto es inicializado por una *petición de aprendizaje*, que contiene una lista de las clases del dominio. Es importante reconocer qué clases no van a ser relevantes para la aplicación en cuestión, y reducir el proceso a la parte del dominio que realmente esté implicada.

La extracción de características es iniciada, del mismo modo que antes, por *solicitudes de extracción* enviadas durante el proceso de aprendizaje, y evalúa estas características sobre las muestras segmentadas (imágenes con un proceso previo de segmentación en este caso).

El *aprendizaje visual* consiste en el entrenamiento de un conjunto de clasificadores, construidos a partir de una selección de características. Estos clasificadores son entrenados para reconocer los conceptos visuales que, previamente, han sido definidos para la descripción del dominio. La salida del proceso de entrenamiento es una *base de conocimiento* ampliada con los clasificadores entrenados.

Modo formal

En esta fase de aprendizaje, cada *conjunto de entrenamiento* (T_i), se asocia con un *concepto visual* (C_i), entendiendo que el conjunto de entrenamiento es un grupo de N vectores etiquetados, y definidos durante la fase de extracción de características. Por ejemplo, los conceptos pueden ser geométricos, de tono, de saturación, de brillo, etc.

$$T_i = \begin{cases} X_1 \\ \ldots \quad X_i \in \mathbb{R}^N \rightarrow Vector \\ X_n \end{cases}$$
$$C_i \rightarrow concepto$$
$$T_i \rightarrow Cjto.entrenamiento\ de\ C_i$$

[Ec. 65]

Estos vectores de características están marcados con la etiqueta (Y_i), de valores entre 1 y -1, siendo el valor 1 indicador de que el vector correspondiente (X_i) es una *muestra representativa* del concepto (C_i), y el valor -1 que se trata de una *muestra negativa*.

$$T_i = \begin{cases} X_1 \\ \ldots \quad X_i \in \mathbb{R}^N \rightarrow Etiqueta = \begin{cases} Y_1 \\ \ldots \quad Yi = \{-1,1\} \in \mathbb{N} \\ Y_n \end{cases} \\ X_n \end{cases}$$

[Ec. 66]

Los sistemas de decisión se denominan *clasificadores* (d_i), y están asociados a cada concepto C_i. Se definen durante la etapa de entrenamiento y serán los determinantes de la decisión final.

La existencia de un determinado *vector de características* condiciona que haya un determinado *concepto*. En términos probabilísticas se expresa: $P\ (C_i\ /\ X)$. Pero tenemos que tener en cuenta que el sistema debe tener una pequeña capacidad de tolerancia en el sentido que explicamos a continuación: la probabilidad de que se obtenga un concepto a partir de la existencia de un conjunto de vectores de características, será una, y de un valor determinado.

En principio, es lógico pensar que el 100% de las probabilidades será la suma de la probabilidad de tener ese concepto con la probabilidad de no tenerlo, de modo que si asignamos un 25% a la posibilidad de que sea negativo, tendríamos un 75% de posibilidades de que fuera positivo. Dar una tolerancia al sistema supone no cerrar la suma por completo, y permitir un rango de equivocación. De esta manera la probabilidad estaría afectada así:

$$P(C_i / x) + P(\neg C_i / x) + P(R_i) = 1$$

[Ec. 67]

En la etapa de entrenamiento se definen también un *umbral de confianza* sobre el resultado y un *umbral de distancia* que evalúa el nivel de pertenencia a ambas probabilidades, a la positiva y a la negativa a la vez, es decir, si pertenece a la intersección de los dos conjuntos probabilísticos.

6.2.1.3 Un ejemplo de ontología visual

Como se ha venido explicando, una ontología es una guía que suministra un lenguaje de descripción, en este caso visual, del dominio de clases.

Para mostrar un ejemplo de lo que podría ser una ontología de estas características, nos ponemos en un caso en el que tuviéramos que distinguir diferentes imágenes aéreas. Para ello tendríamos que realizar un análisis detallado de la textura y del color de fondo.

Conocemos la aplicación que tiene la *Transformada de Fourier* en la descripción de texturas (Fig. 88), por lo que los expertos de este supuesto, proponen incluir la información que infiere la transformación, en los métodos de procesado de imagen y, de esta manera, obtener conceptos.

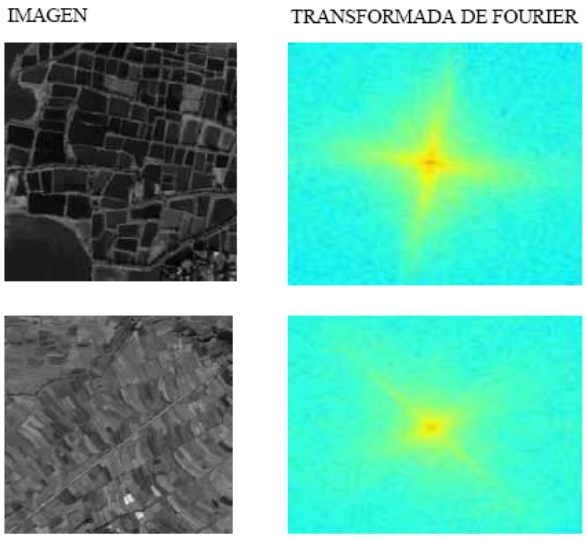


Fig. 88 Dos ejemplos de la aportación de la Transformada de Fourier al análisis de texturas.

Adicionalmente se incluyen los conceptos de color, y un análisis de la textura, teniendo en cuenta las características perceptuales.

Así que la ontología estaría estructurada en tres partes:

(1) Conceptos de textura

(2) Conceptos de color

(3) Conceptos de Fourier

A) Concepto de Textura

La percepción humana de la textura tiene mucho que ver con procesos cognitivos. Cada dimensión de la percepción de textura, se ve como una abstracción de un conjunto de conceptos visuales. El estudio del proceso cognitivo que da lugar a los

conceptos visuales que proponemos, lo realizaron Rao y Lohse [103] y de sus conclusiones extrajeron N. E. Maillot y M. Thonnat [101] la jerarquía que se propone en este ejemplo.

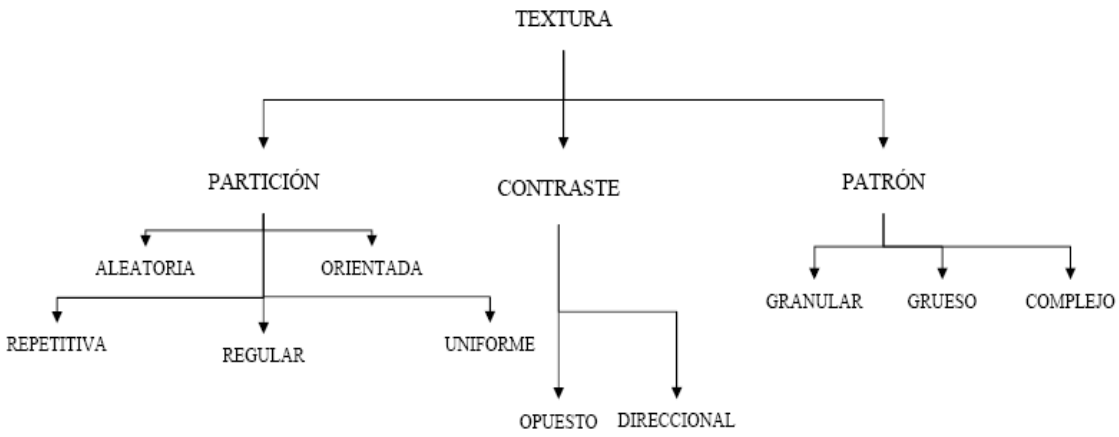


Fig. 89 Estructura cognitiva de concepto visual de la textura

B) Concepto de color

El *término color*, también denominado nombre del color, es una palabra o una frase que referencia un determinado color. Puede referirse a la percepción humana, afectada por el contexto visual, o a cualquier propiedad física subyacente, como una determinada longitud de onda de la luz visible. Existen también sistemas numéricos de especificación de color, denominados *espacios de color*.

El sistema de color Munsell [133] y el ICCS-NBS (Inter.-Society Color Council-National Buerau Standards) son léxicos de términos de color. La desventaja de estos sistemas, sin embargo, es que únicamente especifican determinadas muestras de color. Esto obliga a conseguir las combinaciones restantes a partir de la interpolación.

Para obtener el conjunto de colores normalizado, se va a utilizar por ejemplo la descripción que propone la ISCC-NBS. Con esta norma, la jerarquía queda como se muestra en la Fig. 90. Con esta referencia jerárquica, podemos llegar por ejemplo al concepto “brillante”, como combinación del término “claro” perteneciente al brillo, y el término “fuerte” perteneciente a la saturación.

Ninguno de estos términos tiene definición absoluta. Los expertos los utilizan como discriminante.

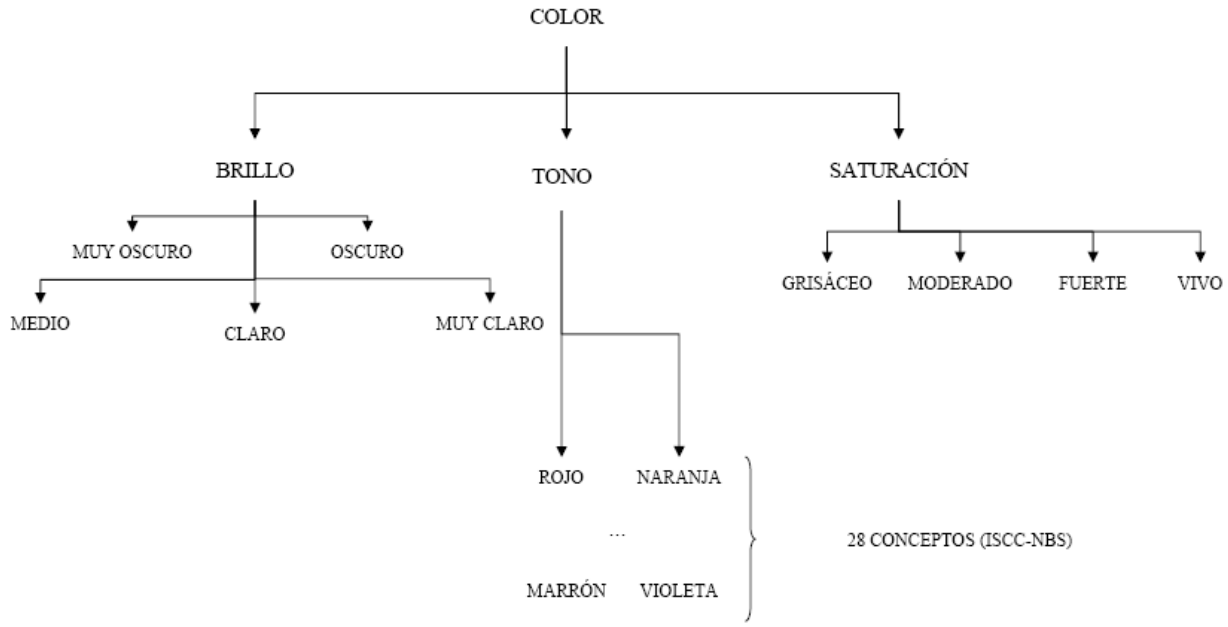


Fig. 90 Estructura del concepto visual de color.

C) Concepto Transformada de Fourier

La *Transformada 2D de Fourier,* tiene una variedad de aplicaciones muy amplia en los sistemas de visión por computador. El rango dinámico del espectro de Fourier es generalmente más extenso que cualquier otro dispositivo visualizador, como por ejemplo un display, en los que únicamente se pueden proyectar conceptos de brillo.

Por ello, proyectando debidamente este espectro, podemos obtener un rango de propiedades que determinen, de forma muy precisa, algunos de los conceptos visuales que necesitamos para evaluar algunas propiedades adicionales de las texturas.

Las tres propiedades que se han considerando más cercanas a la identificación de texturas son las que se proponen en la jerarquía que se muestra en la siguiente figura que, por otro lado, constituye la jerarquía del concepto *Transformada de Fourier* en nuestro ejemplo.

La pretensión que tenemos de este concepto visual, es que nos proporcione información para la descripción del dominio del objeto desde un punto de vista espacial.

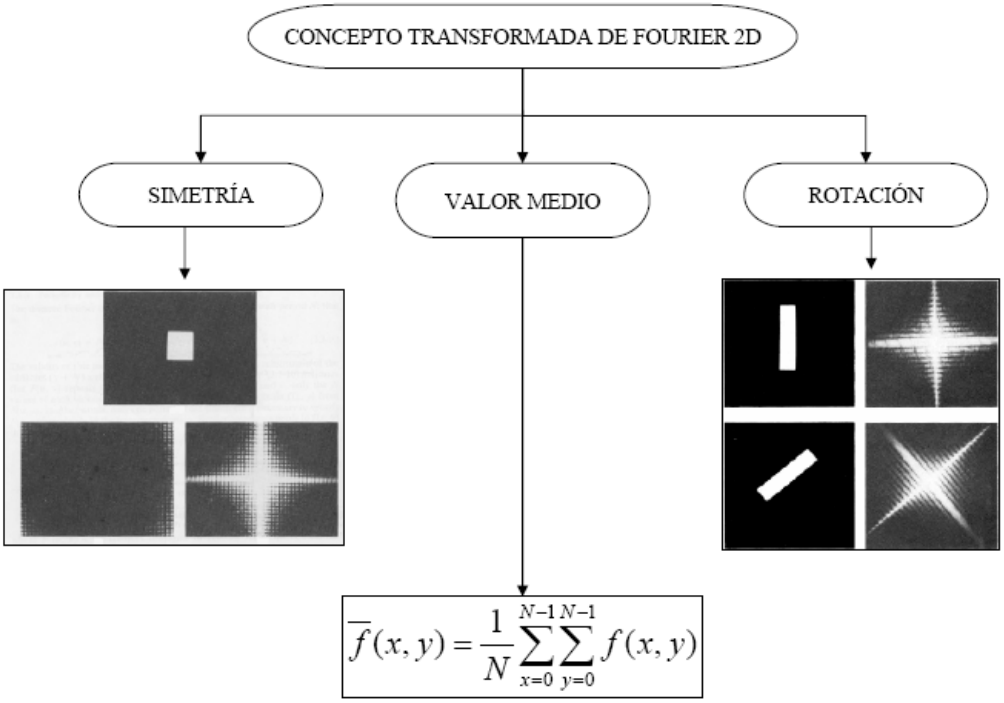


Fig. 91 Estructura del concepto visual de la transformada de Fourier

6.2.2 Semántica de la imagen

En el contexto que nos encontramos, consideramos *interpretación semántica de la imagen* a la forma lógica que se obtiene del sistema de descripción visual, basado en la ontología.

Interpretar semánticamente la imagen es combinar, y coordinar convenientemente cada una de sus singularidades, para formar conceptos y estudiar el significado de los mismos.

A) Análisis sintáctico

Es el primer proceso, y lo podemos denominar así por analogía. Obtiene la estructura de singularidades, sin reflejar el significado de las mismas. Es decir, sólo el modo en que se agrupan para dar lugar a los conjuntos de características que definen un determinado elemento.

B) Análisis semántico

Es el segundo proceso, y es el de *Interpretación Semántica* en sí mismo. Obtiene el significado de los conceptos de modo independiente al contexto, y es lo que se conoce como *Forma Lógica*. Para esta parte se utiliza la ontología visual, que constituye el lenguaje de representación o expresión de las formas lógicas.

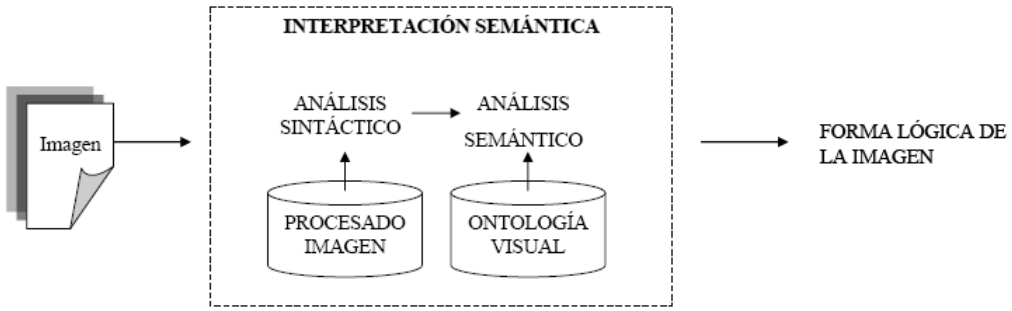


Fig. 92 Esquema del proceso de interpretación de la imagen.

C) La forma lógica

Codifica los posibles significados de cada una de las singularidades de la imagen, e identifica los conceptos y las relaciones entre ellas.

No depende de la aplicación, ni del contexto. Es una representación semántica independiente. Por este motivo se puede llegar a reutilizar en aplicaciones diferentes.

6.2.2.2 Método de interpretación semántica

El sistema intenta comparar un objeto desconocido para categorizarlo con una de las clases del dominio. La comparación se realiza, primero entre los conceptos visuales encontrados en la nueva imagen y los que ya se tienen clasificados, y en una segunda fase, entre las combinaciones de los mismos.

En general, se puede describir el proceso en cinco etapas, como las que definen N. E. Maillot y M. Thonnat [101] en su trabajo y que se exponen a continuación:

(1) Entra en el sistema una solicitud que contiene una imagen del objeto.

(2) Se realiza una segmentación para aislar el objeto del fondo de la escena.

(3) Se comparan los atributos encontrados (por ejemplo, superficie circular) y los conceptos visuales que se obtuvieron en la fase de aprendizaje. Para ello se realiza el proceso de extracción de características y se construye un nuevo vector. El resultado de esta fase, es un conjunto de probabilidades asociadas con cada atributo.

(4) Mediante la combinación de estas probabilidades se evalúa si el nuevo elemento se corresponde con el objeto buscado.

(5) Si no lo ha reconocido, realiza una serie de nuevos intentos. Si todos fallan, desestima el elemento y su clase.

En caso de un reconocimiento positivo, el resultado es el nombre asociado a la clase identificada y el valor de sus atributos.

6.2.2.3 La interpretación y la percepción

El proceso de interpretación, lleva incluido un proceso de percepción por el que se adquiere información del entorno.

Sin embargo, es un error considerar el proceso perceptivo solamente en una de sus partes, la evaluación de sus componentes, y en una de sus direcciones, desde el *entorno* hacia el *sistema perceptor*. El Dr. Ignacio López incluye una reflexión acerca de esta influencia en su tesis doctoral *"A Foundation for Perception in Autonomous Systems"*[104].

En el desarrollo de la ontología visual que se describe en este capítulo, se observa un sistema de aprendizaje dinámico, en cuanto a que puede incluir nuevas clases y nuevas cantidades. Sin embargo, no tiene en cuenta la sucesión de cambios que se pueden dar en los valores de estas cantidades, ni las características extrínsecas al sistema.

El perceptor es una parte del sistema y, por lo tanto, puede presentar acoplamientos, dependencias y restricciones con los demás elementos, I. López [104]. La influencia entre el perceptor y el sistema, ocurre a lo largo de todo el proceso perceptivo y por supuesto, en el objeto percibido. Si no analizamos esta influencia, estamos obviando el contexto en el que tiene lugar, y los resultados pueden variar en función de las interferencias que hayan ocurrido durante su ejecución.

Estas ideas dan paso al estudio realizado en los siguientes apartados, a partir de los planteamientos expuestos en el tercer capítulo.

El uso de una ontología debería de ir ligado a un *modelo de percepción* en el que, a partir de los descriptores numéricos de la imagen, denominados en la jerga cognitiva *singularidades*, se obtengan los *referentes* del sentido u objetos a percibir, pero que además incluya las equivalencias que intervienen en el proceso perceptivo.

En el siguiente diagrama (Fig. 93) se puede apreciar la idea general del uso de la ontología para el aprendizaje.

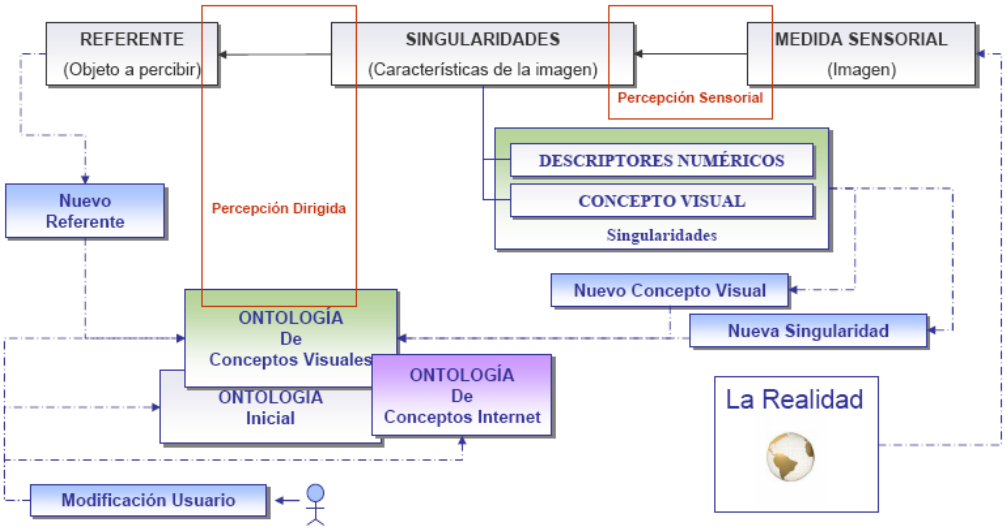


Fig. 93 Esquema de un proceso de percepción y aprendizaje cognitivo

6.3 Análisis en el marco teórico de percepción

En este último apartado, se plantea un ejemplo para la discusión sobre las diferencias en un proceso de visión, utilizando el modelo de percepción de I. López [104] y las claves perceptivas expuestas en el tercer capítulo, y la metodología tradicional. Lo que se pretende es realizar una comparativa entre los métodos tradicionales de la visión por computador, y una metodología en la que se tendría en cuenta el modelo para el entorno perceptivo que afecta al sistema.

Para ello, se va a realizar un breve análisis en el que se va a construir una ontología visual desde la perspectiva tradicional, teniendo en cuenta sólo los parámetros intrínsecos al sistema, y desde la perspectiva cognitiva, en el que se van a incluir también los parámetros extrínsecos.

Supongamos que tenemos la siguiente imagen para analizar, y que nuestro interés sobre ella es conocer la existencia de dos tipos de elementos, esferas o anillos, y su cercanía a la cámara.

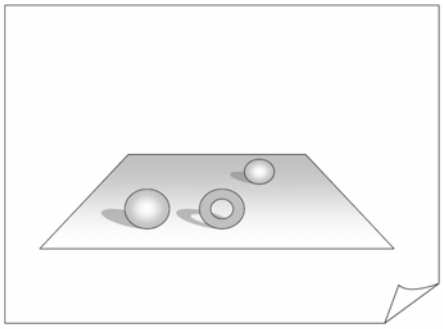


Fig. 94 Imagen original para procesar información acerca de los elementos presentes en la escena y su cercanía a la cámara.

En las secciones previas, se han explicado las bases y la estructura de una ontología basada en conceptos visuales. Cualquier base de conocimiento resultante, nos va a servir como herramienta de clasificación. Durante este proceso, en el que se ordena y encasilla un determinado objeto para obtener su identidad, se utilizan una serie de *descriptores numéricos* que, previamente, se han calculado sobre la imagen. Este proceso se realiza con las técnicas habituales de visión por computador.

6.3.1 Selección de los descriptores numéricos

Consideramos que la relación entre los conceptos visuales y los descriptores numéricos se definirán manualmente para este supuesto. El conjunto de descriptores de forma que vamos a considerar serán:

(1) *Largo (L)*: máxima proyección

(2) *Ancho (W)*: Máxima ortogonalidad con (L)

(3) *Ratio (L/W)*: L/W

(4) *Área (A)*: Número de píxeles

(5) *Factor de forma (FF):* $\frac{4\pi A}{\pi L^2}$

(6) *Circularidad (C)*: $\frac{4A}{\pi L^2}$

(7) *Perímetro (P)*: Longitud de perímetro

(8) *Area frontera (AF)*: Rectángulo circunscriptor

(9) *Ratio areas*: A/(AF)

(10) *Area Convex hull (ACH)*: Area de convex hull

(11) *Perímetro Convex hull (PCH)*: Perímetro convex hull

(12) *Solidez: A/(ACH*)

(13) *Linealidad (L)*: $\sum_{PUNTO\ A}^{PUNTO\ B} PIX \approx \sqrt{(B_X - A_X)^2 + (B_Y - A_Y)^2}$

(14) *Transformada de Hugh-líneas rectas (TH)*: Algoritmo con respuesta

El *ratio (R)*, por ejemplo, calculado sobre una región de interés, se utiliza para caracterizar el concepto visual *elongación*. La *circularidad (C)* y el *factor de forma (FF)*, para caracterizar el nivel de acercamiento a un círculo.

6.3.2 Análisis fuera del marco teórico

Primero realizamos una adaptación de la imagen para su análisis, y una extracción de fondo donde queden aislados los elementos y las regiones de interés. Durante la extracción de fondo, eliminamos todo aquello que no tenga las características que esperamos encontrar, en este caso todo aquello que no tenga forma circular, que no contenga la distribución de color deseada, etc.:

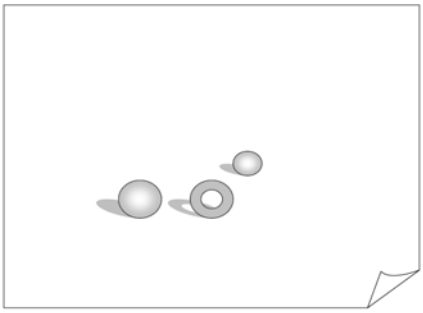


Fig. 95 Operación de extracción de fondo para aislar los objetos.

Una vez realizado esto, buscamos algoritmos que nos ofrezcan la posibilidad de encontrar elementos circulares y caracterizarlos dando lugar, en el mejor de los casos, a un resultado parecido al de la siguiente figura:

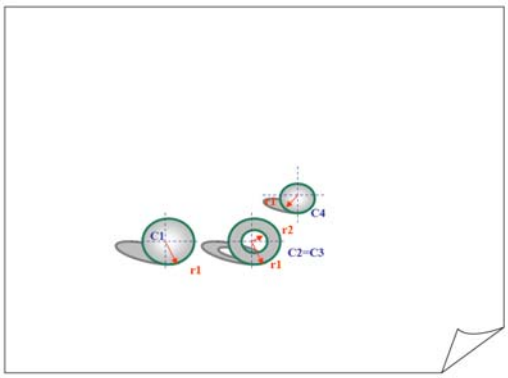


Fig. 96 Resultado de aplicar algoritmos de procesamiento de la imagen

Adicionalmente a este resultado, necesitaríamos realizar algoritmos de *Cálculo de Contornos* y de *Centro de Masas* para obtener los centros de interés. No vamos a realizar este análisis puesto que no es el tema que nos ocupa en este capítulo.

A) Resultados en el procesamiento de la imagen

Como resultado, tenemos una relación de elementos parecida a la siguiente: Cuatro círculos, tres contornos, tres centros de masa (en el mejor de los casos cuatro) y cuatro áreas.

Para concluir con resultados válidos tenemos que realizar un estudio más o menos parecido a:

(1) Tenemos dos círculos del mismo radio y otros dos de radios diferentes.

(2) De los tres contornos, uno tiene un hueco en el centro.

(3) Comprobar las posiciones en coordenadas de los centros de masa y detectar si alguno de ellos coincide en posición.

(4) Detectar qué áreas pertenecen a cada uno de los círculos, teniendo en cuenta que un área va a pertenecer a dos de ellos, que precisamente son los que tienen el mismo centro.

(5) Agrupar los elementos adecuadamente para obtener el concepto de figuras.

(6) Realizar un *matching* para comprobar que dos de las figuras son de la misma forma pero con tamaños diferentes.

(7) Realizar un escalado con una calibración previa para obtener el grado de cercanía a la cámara de cada figura localizada. Todo esto, teniendo en cuenta que los elementos tienen en la realidad el mismo tamaño, puesto que el escalado a partir de la imagen va a depender del mismo. El sistema queda inhabilitado para distinguir cercanía o lejanía en varios elementos de diferente área o volumen.

B) Ontología de conceptos visuales

La ontología nos va a guiar en el proceso de la descripción visual aportando el vocabulario necesario. Como hemos dicho en secciones anteriores, la ontología no va a ser dependiente de la aplicación, y debe desarrollarse manteniendo su escalabilidad.

Partiendo de esta base, se ha estructurado la ontología en dos partes principales: la primera contiene los conceptos de color del ejemplo planteado anteriormente en este capítulo, y la segunda conceptos espaciales de forma.

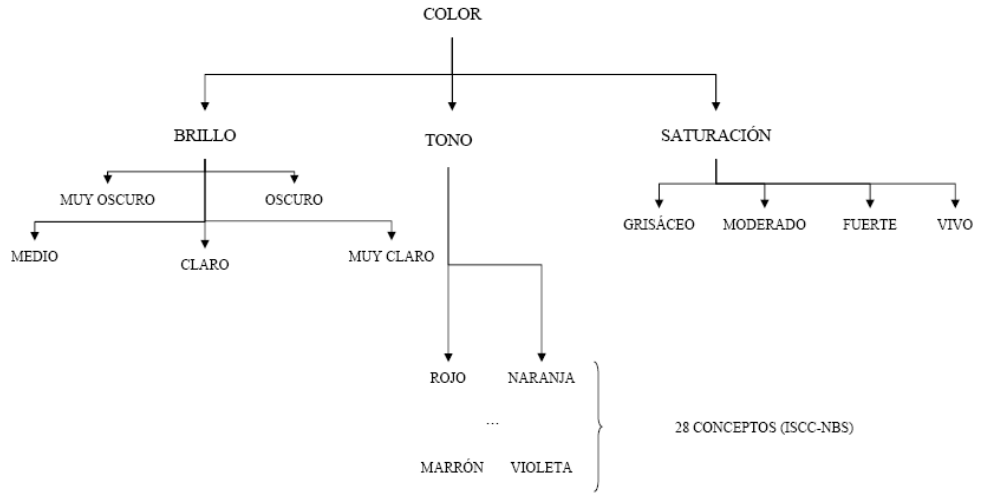


Fig. 97 Jerarquía del concepto visual Color

El concepto visual de color fue explicado en el ejemplo anterior. El concepto de forma, lo que pretende es la descripción del objeto desde un punto de vista espacial.

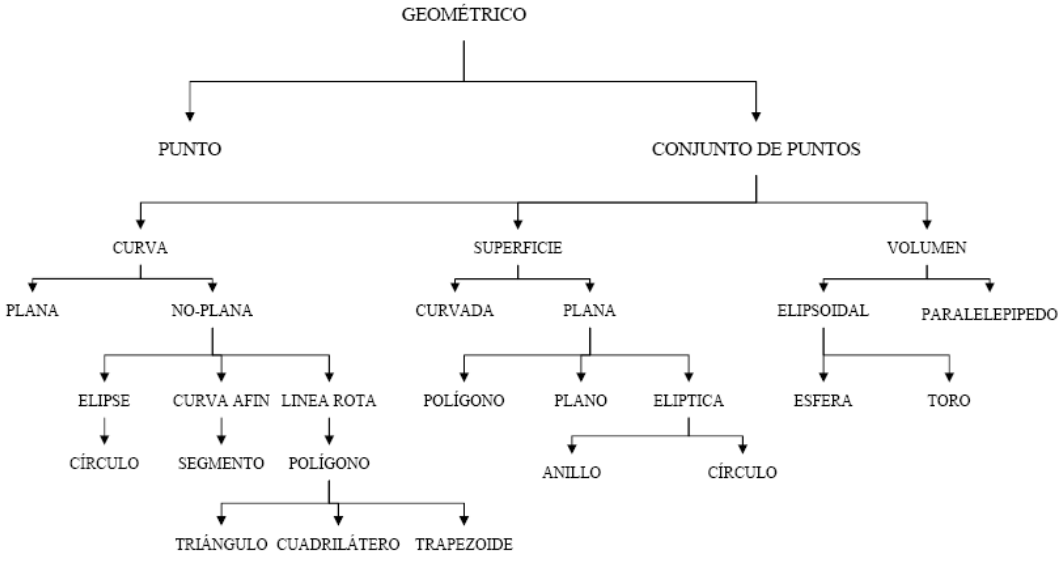


Fig. 98 Jerarquía del concepto visual Geometrico.

C) Resultados de la interpretación semántica

Pongamos que la implementación del sistema ha supuesto definir 49 conceptos visuales, una cantidad entre 20 y 30 relaciones espaciales, y una relación inicial de 14 descriptores numéricos.

La ontología utilizada como herramienta, nos proporciona un medio fácil para comunicarnos en el contexto del análisis. Aunque el conjunto de descriptores numéricos están asociados con los conceptos visuales, algunos son eliminados por el experto: cuando se elige un concepto visual, el experto implícitamente elige un conjunto de ellos. Esto se debe a que la base de conocimiento es muy cercana al nivel más bajo de la visión.

En el mejor de los casos, todas estas operaciones han salido correctamente y hemos conseguido averiguar que tenemos un anillo y dos esferas. Que una esfera y el anillo están a la misma distancia y que la otra esfera se encuentra en una posición más lejana de la cámara.

Sin embargo, no podemos distinguir la lejanía y cercanía de los elementos si no partimos de la base de que sean de igual tamaño en la realidad. Dos elementos de diferente tamaño, con la misma cercanía a la cámara, tendrían un resultado de lejanía diferente al término de la operación de interpretación de la imagen.

No tendríamos concepto de espacio, ni detectaríamos la presencia de elementos con características diferentes a las que buscamos. Impedimos al sistema desde el principio la detección de objetos desconocidos para *aprenderlos*, puesto que las relaciones para inferirlos se tienen que hacer fuera de línea y previo al funcionamiento del sistema. No tenemos en cuenta nuestro conocimiento previo, los factores extrínsecos al sistema de visión y no podemos deducir que haya presencia de elementos que “son algo” como referentes, pero que aún no conocemos.

6.3.3 Metodología bajo el marco teórico

Según el modelo del marco teórico de I. López [104] el conjunto de referencias y singularidades analizando con las claves puede ser analizado de la siguiente forma.

A) Primer análisis: Detección de formas

Según la metodología explicada en el tercer capítulo, realizamos el estudio necesario de los conjuntos de singularidades necesarios:

$$
\begin{aligned}
&Conjunto\ \ \Psi^1 \quad Condicion\_CONTORNO \\
&Singularidad\psi_1^1 = Contorno\_1 \\
&Singularidad\psi_2^1 = Contorno\_2 \\
&Conjunto\ \ \Psi^2 \quad Condicion\_COLOR \\
&Singularidad\psi_1^2 = Color\_1 \\
&Singularidad\psi_2^2 = Color\_2 \\
&Conjunto\ \ \Psi^3 \quad Condicion\_RADIOS \\
&Singularidad\psi_1^3 = Radio\_1 \\
&Singularidad\psi_2^3 = Radio\_2 \\
&Conjunto\ \ \Psi^4 \quad Condicion\_CENTROS \\
&Singularidad\psi_1^4 = Centro\_1 \\
&Singularidad\psi_2^4 = Centro\_2
\end{aligned}
$$

[Ec. 68]

Los referentes:

$$
\begin{aligned}
&referente\ \ \rho^1 : Detección\_anillo \\
&referente\ \ \rho^2 : Detección\_pelota \\
&referente\ \ \rho^3 : Condicion\_Radios \\
&referente\ \ \rho^4 : Condicion\_Centros \\
&referente\ \ \rho^5 : Condicion\_Color
\end{aligned}
$$

[Ec. 69]

Y las relaciones de equivalencia:

$$
\begin{aligned}
&Dependencia \;\; \varepsilon_1^1\left(\psi_1^1,\psi_2^1,\rho^3,\rho^4,\rho^5\right)\\
&\rho^1 = Deteccion\_anillo \Leftrightarrow \varepsilon_1^1 = \psi_1^1 \wedge \psi_2^1 \wedge \rho^3 \wedge \rho^4 \wedge \rho^5\\
&Dependencia \;\; \varepsilon_1^2\left(\psi_1^1,\psi_1^2,\rho^5\right)\\
&\rho^2 = Deteccion\_pelota \Leftrightarrow \varepsilon_1^2 = \left(\psi_1^1 \vee \psi_1^2\right) \wedge \rho^5\\
&Dependencia \;\; \varepsilon_1^3\left(\psi_2^1,\psi_1^3,\psi_2^3\right)\\
&\rho^3 = Condicion\_Radios \Leftrightarrow \left(\psi_3^1 \neq \psi_3^2\right)\\
&Dependencia \;\; \varepsilon_1^4\left(\psi_1^4,\psi_2^4\right)\\
&\rho^4 = Condicion\_Centros \Leftrightarrow \left(\psi_1^4 = \psi_2^4\right)\\
&Dependencia \;\; \varepsilon_1^5\left(\psi_{1,}^5\right)\\
&\rho^5 = Condicion\_Color \Leftrightarrow \psi_3^5 > \left(\psi_1^1 \wedge \psi_2^1\right)
\end{aligned}
$$

[Ec. 70]

Para este primer modelo del sistema tendríamos la capacidad de analizar diferentes tamaños de dos objetos diferentes sin más que obtener instancias de los referentes ρ^1, ρ^2, ρ^3.

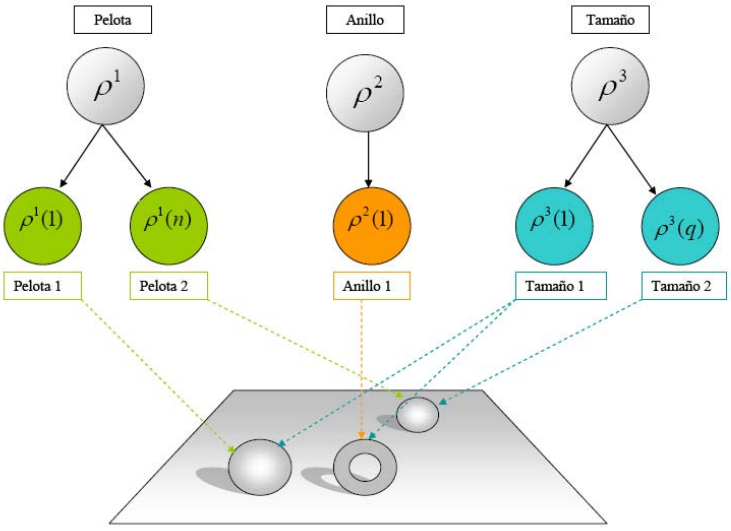


Fig. 99 Detección de las dos esferas y el anillo dentro del marco teórico del modelo de percepción considerado en el análisis.

Tal y como se muestra en la Fig. 99, la combinación de las instancias de estos tres referentes, nos da la capacidad de reconocer los mismos objetos que en el modelo tradicional.

B) Segundo análisis: Evaluación de la distancia

Ahora proponemos el mismo método que en el modelo tradicional planteado inicialmente. Sólo que en esta ocasión no realizamos una primera eliminación de fondo, sino que procesamos la imagen directamente intentando encontrar líneas rectas además de las circulares que inicialmente se ha propuesto.

Utilizamos a *Transformada de Hugh* [127] y el resultado puede ser una imagen parecida a la de la siguiente figura:

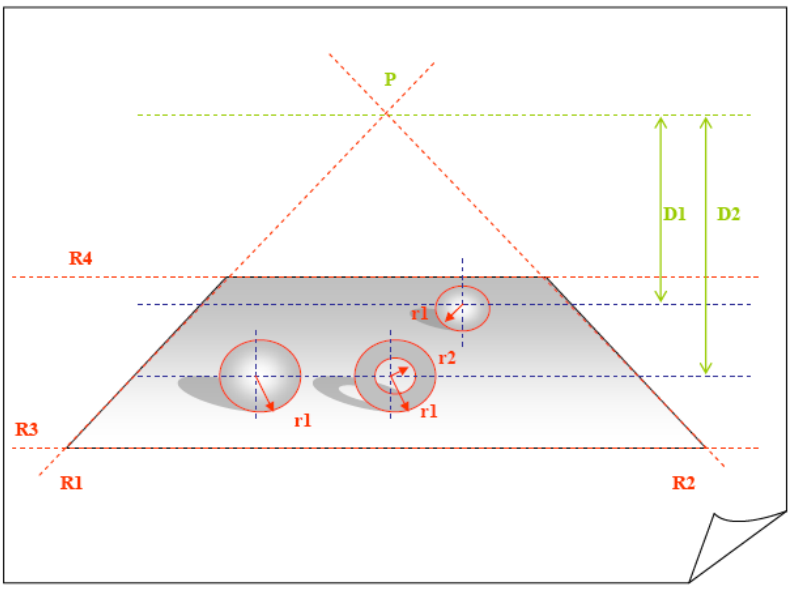


Fig. 100 Resultado de la detección de líneas rectas y líneas circulares con la Transformada de Hugh directamente sobre la imagen sin eliminación previa de fondo.

Tenemos que añadir algunos referentes adicionales que nos permitan definir los nuevos elementos que aparecen en la imagen.

$$\begin{aligned} &Conjunto \quad \Psi^6 \quad Condicion\_no\_Ejes \\ &\quad Singularidad\psi_1^6 = No\_Vertical \\ &\quad Singularidad\psi_2^6 = No\_Horizontal \\ &Conjunto \quad \Psi^7 \quad Condicion\_SECANTE \\ &\quad Singularidad\psi_1^7 = Punto\_de\_Corte \\ &\quad Singularidad\psi_2^7 = recta\_2^a \end{aligned}$$

[Ec. 71]

Los referentes:

$$referente \quad \rho^6 : Punto\_de\_Fuga$$

[Ec. 72]

Y las relaciones de equivalencia:

$$\begin{aligned} &Dependencia \quad \varepsilon_1^6\left(\psi^6, \Psi^7\right) \\ &\quad \rho 6 = Detecci\acute{o}n\_Punto\_de\_Fuga \Leftrightarrow \varepsilon_1^6 = \psi^6 \wedge \Psi^7 \end{aligned}$$

[Ec. 73]

Disponemos de un nuevo referente esta vez explícito al sistema: el *punto de fuga* de la imagen. En un sistema de proyección cónica, es el lugar donde convergen todas las rectas proyectadas paralelas a una dirección; se trata de un punto situado en el infinito y existen tantos como direcciones en el espacio.

Es un concepto intuitivo, un lugar donde veríamos confluir las dos líneas de la base sobre la que se sitúan los tres objetos de la imagen. No se ve, no es un elemento intrínseco al sistema. Está en nuestra percepción porque nuestros propios procesos de aprendizaje lo han definido y creado. Es decir, se trata de un concepto extrínseco que puede ser incluido en el sistema como referente y proporcionar parte de la información necesaria para percibir el espacio.

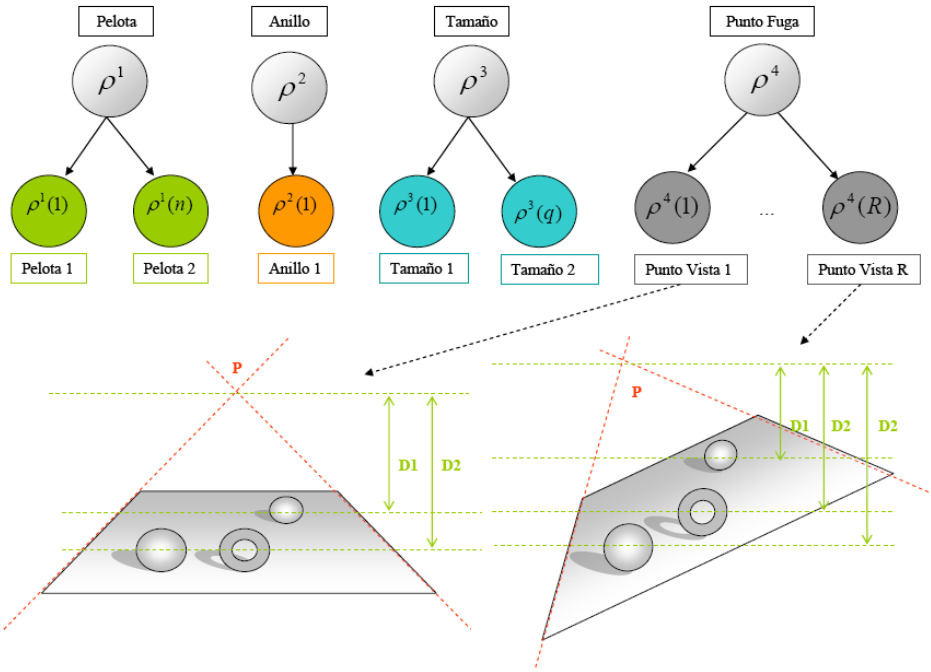


Fig. 101 Modelo de percepción que incluye el referente Punto de Fuga

El punto de fuga no sería posible considerarlo para inferir un resultado en un sistema de visión tradicional en el que únicamente se trabaja con las características intrínsecas al sistema.

Este análisis constituye una justificación teórica de lo que se quiere plantear en los trabajos futuros para la obtención de un sistema de aprendizaje basado en visión.

CAPÍTULO 7

Conclusiones y trabajos futuros

7.1 Conclusiones

A lo largo del estudio y desarrollo del presente trabajo, se ha profundizado en el análisis de las técnicas actuales de visión por computador para resolver problemas de reconocimiento facial y gestual, y en el estudio de las nuevas propuestas de visión cognitiva, donde se propone un estudio completo de la imagen teniendo en cuenta, no sólo las características intrínsecas que infieren los algoritmos matemáticos, sino también las extrínsecas que afectan a nuestro conocimiento y percepción y que, como consecuencia, modifican el significado y la interpretación final de la imagen.

El objetivo del trabajo era construir un sistema de visión por computador para una plataforma de robot social y el estudio de las necesidades que tendría un sistema de reconocimiento y aprendizaje basado en visión. Este planteamiento se ha analizado desde la perspectiva tradicional para realizar su implementación y, posteriormente, se realizado el estudio teórico desde una perspectiva cognitiva. La propuesta teórica se realiza a partir de la elección de un modelo de percepción sobre el que se ha proyectado el sistema de detección facial, con objeto de justificar la necesidad de los factores de la percepción en la interpretación de la imagen.

El proyecto se ha desarrollado, en cada una de sus fases, dedicando tiempo tanto al estudio inicial del estado del arte, a la implementación de un prototipo, y al estudio de la visión cognitiva y los modelos de percepción, según los siguientes puntos:

- Respecto al estudio del estado del arte, se han tenido en cuenta diferentes áreas que van, desde la interacción de los robots con las personas, la influencia de los sistemas de visión en dichos sistemas y los factores que más afectan al resultado final, hasta los principales criterios de la visión cognitiva, las ontologías de conceptos visuales para la interpretación semántica de la imagen y los modelos de percepción.

- Ha sido necesario un estudio detallado de los algoritmos de visión por computador en las áreas de reconocimiento e identificación facial, identificación de movimiento, segmentación y filtrado de la imagen, técnicas de color y análisis en el dominio de la frecuencia. Asimismo se han estudiado las nuevas propuestas en el campo de la visión cognitiva, los modelos de percepción con agentes y las ontologías de conocimiento para el aprendizaje.

- Se ha desarrollado un prototipo de sistema de visión por computador que detecta la presencia de caras dentro de una imagen, con un algoritmo redundante que combina la búsqueda por plantillas y por distribución probabilística de color. El sistema define de forma dinámica la moda probabilística de color a seguir, correspondiente al color de la piel, y construye a partir de este valor, el histograma adecuado para la búsqueda. El algoritmo reduce a niveles muy bajos, alrededor de un 5%, las pérdidas en la detección (siempre que se mantengan las restricciones del sistema).

- Una vez localizadas las zonas faciales, identifica a la persona de la imagen si se encuentra dentro de un rango inicial de cinco a diez identidades para las que ha sido entrenado, con un análisis de componentes principales (PCA) y una traslación del sistema a un subespacio vectorial de características faciales. Este nuevo espacio vectorial tiene una dimensión menor, lo que permite la realización del sistema en tiempo real.

- El sistema además, segmenta y aísla la mano dentro de la imagen, con objeto de analizar su movimiento y gesto a partir de un polígono convexo que la circunscribe, construido a partir de la selección de características de interés pertenecientes al borde del área de la mano, detectada a su vez con la misma distribución probabilística de color que se definió para la detección facial. Estas capacidades se extenderán posteriormente a su entrenamiento en todo lo referente a expresiones faciales y movimiento de manos y brazos.

- Se ha estudiado y analizado el sistema de detección facial con un modelo de percepción cognitiva propuesto en la tesis presentada por el Dr. Ignacio López dentro de nuestro departamento.

- Se ha analizado y estudiado la interpretación semántica de la imagen y el aprendizaje a partir de ontologías visuales que permitan inferir sin necesidad de recurrir a los niveles más básicos de la algoritmia de la visión por computador.

- Se han evaluado las técnicas de aprendizaje, percepción e interpretación con ontologías y se ha realizado la propuesta de utilizar los modelos de percepción y las claves perceptivas estudiados en este trabajo.

El sistema implementado tiene un alto grado de éxito siempre que sean mantenidas las condiciones de entorno e iluminación para las que ha sido diseñado. Los desarrollos se han realizado en base a procedimientos matemáticos, estadísticos y de probabilidad.

Este primer prototipo no tiene capacidad de modificación autónoma, necesaria para la implementación de un sistema de aprendizaje, siendo uno de los objetivos futuros de este proyecto.

7.2 Líneas futuras

Las líneas de desarrollo que se plantean a partir de la realización de este trabajo se pueden clasificar en dos partes: (1) Por un lado, las líneas futuras referentes a la plataforma y al sistema de visión implementado y, (2) por otra parte, las que tienen que ver con la visión cognitiva, los modelos de percepción y el enfoque al aprendizaje.

(1) Referentes a la plataforma y al sistema de visión:

- Puesta a punto de la solución implementada para reconocer a su tutor y recibir órdenes sencillas, con objeto de incluir la solución en la plataforma.

- Estudiar la reescritura del código para la realización de un sistema distribuido, que optimice las capacidades de la solución en lo referente a potencia de cálculo y velocidad de procesamiento. El objetivo es incrementar el número de identidades que pueda llegar a reconocer, procesar en paralelo el reconocimiento y aprendizaje de nuevas identidades, mejorar la robustez ante la variabilidad del entorno, aumentar el número de órdenes con capacidad de atención, trabajo independiente de cada algoritmo y escalabilidad del sistema.

- Extensión de las capacidades para reconocer movimientos de la mano al reconocimiento de gestos faciales y del brazo, para poder entrenar al robot en lo referente al movimiento en su discurso. Esto eliminaría la necesidad actual de incluir coordenadas de actuación a los servomotores del brazo, boca, párpados y ojos de Urbano. El entrenamiento se realizaría a través de un aprendizaje que incluyera directamente en el sistema, las coordenadas de movimiento a través del reconocimiento con visión.

- Independizar la detección de la región facial, de la detección gestual de las manos, actualmente ligadas por el cálculo de la distribución probabilística del color de piel.

- Ampliar la detección a ambas manos y no sólo a la mano derecha.

- Incorporación de información importante en lo referente a la percepción de texturas con la Transformada Discreta de Fourier.

- Estudio de un método de identificación facial para el aprendizaje, que permita la modificación automática en tiempo real sin necesidad de realizar un entrenamiento fuera de línea cada vez que se quiera incluir una nueva identidad al sistema.

- Trabajo conjunto con los sistemas de detección de voz y de habla para la identificación del tutor de forma redundante, tanto con el procesamiento con visión, como por el timbre de su interlocutor.

En este trabajo se ha realizado, además, el estudio de un posible marco general donde se contemple la percepción en todo su rango, de modo que suponga un nivel adicional de información a la hora de interpretar la imagen y de realizar un aprendizaje.

Acorde a esta idea, se ha seleccionado un modelo de percepción, y se ha llevado este modelo al sistema de visión por computador implementado (en concreto al sistema de detección facial), con objeto de justificar la cobertura que puede llegar a ofrecer en lo que tiene que ver con el entendimiento global del sistema.

(2) Referentes a la visión cognitiva, los modelos de percepción y el enfoque al aprendizaje:

- Implementación de un sistema de aprendizaje basado en visión, con capacidades para reconocer e identificar personas y objetos, así como aprender nuevos conceptos e incluirlos en su conocimiento.

- Conseguir generalidad en el sistema de percepción visual, dado que la heterogeneidad que tienen estos sistemas es directamente proporcional a los problemas que infieren en las soluciones. Es necesario un análisis unificado, justificando de esta manera el estudio que se ha realizado del modelo de percepción de I. López [104], a su vez realizado en base a la *Teoría General de Sistemas*.

- Obtener conceptos, principios y relaciones de aplicación para la ontología de conceptos visuales, teniendo en cuenta la necesidad de identificación de nociones y principios que subyacen al concepto de sistema autónomo y a sus necesidades de operación.

- Realizar un primer prototipo de ontología visual con conceptos visuales deducidos a partir de características intrínsecas y extrínsecas de la imagen, así como de las claves perceptuales de reconocimiento espacial.

- Estudiar la implementación de un posible sistema de interpretación semántica de la imagen y aprendizaje a partir de la visión.

Bibliografía

[1] http://support.sony-europe.com/aibo/

[2] B. Fogg, Introduction: Persuasive technologies, Communications of the ACM 42 (5) (1999)

[3] I. Werry, et al., Can social interaction skills be taught by a social agent? The role of a robotic mediator in autism therapy, in: Proceedings of the International Conference on Cognitive Technology, 2001.

[4] E. Paulos, J. Canny, Designing personal tele-embodiment, Autonomous Robots 11 (1) (2001).

[5] K. Dautenhahn, A. Billard, Bringing up robots or the psychology of socially intelligent robots: From theory to implementation, in: Proceedings of the Autonomous Agents, 1999.

[6] J. Zlatev, The Epigenesis of Meaning in Human Beings and Possibly in Robots, Lund University Cognitive Studies, vol. 79, Lund University, 1999.

[7] S. Restivo, Bringing up and booting up: Social theory and the emergence of socially intelligent robots, in: Proceedings of the IEEE Conference on SMC, 2001.

[8] K. Dautenhahn, The art of designing socially intelligent agents—science, fiction, and the human in the loop, Applied Artificial Intelligence Journal 12 (7–8) (1998) 573–617.

[9] T. Fong, I. Nourbakhsh, K. Dautenhahn, A survey of socially interactive robots: concepts, design, and applications, Technical Report No. CMU-RI-TR-02-29, Robotics Institute, Carnegie Mellon University, 2002.

[10] C. Breazeal, Designing Sociable Robots, MIT Press, Cambridge, MA, 2002.

[11] I. Nourbakhsh, An affective mobile robot educator with a full-time job, Artificial Intelligence 114 (1–2) (1999) 95–124.

[12] J. Pineau, M. Montemerlo, M. Pollack, N. Roy, S. Thrun, Towards robotic assistants in nursing homes: Challenges and results, Robotics and Autonomous Systems 42 (2003) 271–281 (this issue).

[13] B. Scassellati, Foundations for a theory of mind for a humanoid robot, Ph.D. Thesis, Department of Electronics Engineering and Computer Science, MIT Press, Cambridge, MA, 2001.

[14] K. Dautenhahn, I could be you-the phenomenological dimension of social understanding, Cybernetics and Systems Journal 28 (5) (1997).

[15] K. Dautenhahn, C. Nehaniv, Living with socially intelligent agents: A cognitive technology view, in: K. Dautenhahn (Ed.), Human Cognition and Social Agent Technology, Benjamin, New York, 2000.

[16] B. Duffy, Anthropomorphism and the social robot, Robotics and Autonomous Systems 42 (2003) 177–190 (this issue).

[17] P. Persson, et al., Understanding socially intelligent agents-A multilayered phenomenon, IEEE Transactions on SMC 31 (5) (2001).

[18] B. Scassellati, Foundations for a theory of mind for a humanoid robot, Ph.D. Thesis, Department of Electronics Engineering and Computer Science, MIT Press, Cambridge, MA, 2001.

[19] M. Lansdale, T. Ormerod, Understanding Interfaces, Academic Press, New York, 1994.

[20] A. Whiten, Natural Theories of Mind, Basil Blackwell, Oxford, 1991.

[21] http://www.ai.mit.edu/projects/humanoid-robotics-group/cog/

[22] P. Persson, et al., Understanding socially intelligent agents—A multilayered phenomenon, IEEE Transactions on SMC 31 (5) (2001).

[23] C. Breazeal, Designing Sociable Robots, MIT Press, Cambridge, MA, 2002.

[24] K. Dautenhahn, The art of designing socially intelligent agents—science, fiction, and the human in the loop, Applied Artificial Intelligence Journal 12 (7–8) (1998) 573–617.

[25] T. Sheridan, Eight ultimate challenges of human–robot communication, in: Proceedings of the International Workshop on Robots and Human Communication, 1997.

[26] T. Sheridan, Eight ultimate challenges of human–robot communication, in: Proceedings of the International Workshop on Robots and Human Communication, 1997.

[27] M. Scheeff, et al., Experiences with Sparky: A social robot, in: Proceedings of the Workshop on Interactive Robot Entertainment, 2000.

[28] J. Schulte, et al., Spontaneous, short-term interaction with mobile robots in public places, in: Proceedings of the International Conference on Robotics and Automation, 1999.

[29] K. Severinson-Eklund, A. Green, H. Hüttenrauch, Social and collaborative aspects of interaction with a service robot, Robotics and Autonomous Systems 42 (2003) 223–234 (this issue).

[30] F. Michaud, S. Caron, Roball—An autonomous toy-rolling robot, in: Proceedings of the Workshop on Interactive Robot Entertainment, 2000.

[31] C. Breazeal, B. Scassellati, A context-dependent attention system for a social robot, in: Proceedings of the International Joint Conference on Artificial Intelligence, Stockholm, Sweden, 1999, pp. 1146–1153.

[32] C. Breazeal, A. Edsinger, P. Fitzpatrick, B. Scassellati, Active vision systems for sociable robots, IEEE Transactions on Systems, Man and Cybernetics 31 (5) (2001).

[33] D. Gavrilla, The visual analysis of human movement: A survey, Computer Vision and Image Understanding 73 (1) (1999).

[34] R. Tanawongsuwan, et al., Robust tracking of people by a mobile robotic agent, Technical Report No. GIT-GVU- 99-19, Georgia Institute of Technology, 1999.

[35] V. Pavlovic, R. Sharma, T. Huang, Visual interpretation of hand gestures for human–computer interaction: A review, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997).

[36] Y. Wu, T. Huang, Vision-based gesture recognition: A review, in: Gesture-Based Communications in HCI, Lecture Notes in Computer Science, vol. 1739, Springer, Berlin, 1999.

[37] D. Kortenkamp, E. Huber, P. Bonasso, Recognizing and interpreting gestures on a mobile robot, in: Proceedings of the AAAI-96, Portland, OR, 1996, pp. 915–921.

[38] T. Kanade. “Picture Processing by Computer Complex and Recognition of Human Face”. Ph.D. thesis, Kyoto University, 1973.

[39] S. Waldherr, R. Romero, S. Thrun, A gesture-based interface for human–robot interaction, Autonomous Robots 9 (2000).

[40] G. Xu, et al., Toward robot guidance by hand gestures using monocular vision, in: Proceedings of the Hong Kong Symposium on Robotics Control, 1999.

[41] R. Chellappa, et al., Human and machine recognition of faces: A survey, Proceedings of the IEEE 83 (5) (1995).

[42] T. Fromherz, P. Stucki, M. Bichsel, A survey of face recognition, MML Technical Report No. 97.01, Department of Computer Science, University of Zurich, 1997.

[43] K. Toyama, Look, Ma—No hands! Hands-free cursor control with real-time 3D face tracking, in: Proceedings of the Workshop on Perceptual User Interfaces, 1998.

[44] C. Darwin, The Expression of Emotions in Man and Animals, Oxford University Press, Oxford, 1998.

[45] C. Lisetti, D. Schiano, Automatic facial expression interpretation: Where human–computer interaction, artificial intelligence, and cognitive science intersect, Pragmatics and Cognition 8 (1) (2000).

[46] R. Stiefelhagen, J. Yang, A. Waibel, Tracking focus of attention for human–robot communication, in: Proceedings of the International Conference on Humanoid Robots, 2001.

[47] J. Panksepp, Affective Neuroscience, Oxford University Press, Oxford, 1998.

[48] P. Ekman, Basic emotions, in: T. Dalgleish, M. Power (Eds.), Handbook of Cognition and Emotion, Wiley, New York, 1999

[49] H. Schlossberg, Three dimensions of emotion, Psychology Review 61 (1954).

[50] Chomsky, Noam (1959) *A Review of B. F. Skinner's Verbal Behavior.* [Journal (Paginated)]

[51] Philip N. Jhonson-Laird “The Computer and the Mind- An introduction to Cognitive Science” Hardvard University Press

[52] K. Fukunaga, “Introduction to statistical pattern recognition”. Academic Press, Boston, 2 edition,1990

[53] Gary R.Bradski, Microcomputer Research Lab, Santa Clara, CA, Intel Corporation “Computer Vision Face Tracking For Use in a Perceptual User Interface”, 1998

[54] Matthew A. Turk and Alex P. Pentland, “Face Recognition Using Eigenfaces”, Vision and Modeling Group, The Media Laboratory Massachusetts Institute of Technology. 1991 IEEE.

[55] Matthew Turk and Alex Pentland, “Eigenfaces for Recognition”, Journal of Cognitive Neuroscience Volume 3, Number1, 1991 Massachusetts Institute of Technology.

[56] Yoav Freund, Robert E.Schapire, “A short Introduction to Boosting”, Journal of Japanese Society for Artificial Intelligence, 14(5):771-780, September, 1999.

[57] Paul Viola, Michael J.Jones, “Robust Real-Time Face Detection”, International Journal of Computer Vision 57(2), 137-154, 2004.

[58] K.Fukunaga, “Introduction to Statistical Pattern Recognition”, Academic Press, Boston, 1990.

[59] D. Comaniciu and P.Meer, “Robust Analysis of Feature Spaces: Color Image Segmentation”, CVPR’97, pp. 750-755

[60] Y.Cheng, “Mean shift, mode seeking, and clustering”, IEEE Trans. Pattern Anal. Machine Intell., 17:790-799, 1995.

[61] W.T. Freeman, K.Tanaka, J.Ohta, and K.Kyuma, “Computer Vision for Computer Games”, Int. Conf. On Automatic Face and Gesture Recognition, pp. 100-105, 1996.

[62] L. Sirovich and M. Kirby. “Low Dimensional Procedure for the Characterization of Human Faces”. *Journal of the Optical Society of America*, 4(3):519-524, 1987.

[63] M. Kirby and L. Sirovich, “Application of the Karhunen–Loeve procedure for the characterization of human faces”. IEEE Trans. Pattern Anal. Machine Intell. 12 1 (1990), pp. 103–108.

[64] R.T. Collins et al., “A system for video surveillance and monitoring: VSAM final report“, CMU-RI-TR-00-12, Technical Report, Carnegie Mellon University, 2000.

[65] I.Haritaoglu, D. Harwood, L.S. Davis, “W^4: real time surveillance of people and their activities“, IEEE Trans. On Pattern Analysis and Machine Intelligence, 22(8) (2000) 809-830.

[66] J. Steffens, E. Elegain, H.Neven, “Person Spotter-fast and robust system for human detection, tracking and recognition“. Proc. Of IEEE Intl. Conf. on Automatic Face and Gesture Recognition, 1998, pp. 516-521.

[67] C. Wang, M.S. Brandstein, “A hybrid real-time face tracking system“. Proc. Of Int. Conf. on Accoustics, Speech, and Signal Processing, Seattle, WA, 1998.

[68] J.J. Little, J.E Boyd, “Recognizing people by their gait: the shape of motion, Videre: Journal of Computer Vision Research“, The MIT Press, 1 (2), 1998.

[69] D. Cunado, M.S. Nixon, “J.N. Carter, Automatic gait recognition via model-based evidence gathering“, Proc. Of Workshop on Automatic Identification Advanced Thecnologies. New Jersey, 1998, pp. 27-30.

[70] Yi Li, Sondge Ma, Hanqing Lu, “Human posture recognition using multi-scale morphological method and Kalman motion estimation“. Proc. of IEEE Intl. con Pattern Recognition, 1998, pp. 175-177.

[71] J. Segen, S. Kumar, “Shadow gestures: 3D hand pose estimation using a single camera“. Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition. 1999, pp. 479-485.

[72] M-H. Yang, N. Ahuja, “Recognizing hand gesture using motion trajectories“. Proc. of IEEE CS Conference on Computer Vision and Pattern Recognition. 1999, pp. 468- 472.

[73] Y. Cui, J.J. Weng, "Hand segmentation using learning-based prediction and verification for hand sign recognition". Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition. 1997, pp. 88-93.

[74] M. Turk, "Visual interaction with lifelike characters". Proc. of IEEE Intl. Conf. on Automatic Face and Gesture Recognition, Killington, 1996, pp. 368-373.

[75] J.K. Aggarwal, Q. Cai, W. Liao, B. Sabata, "Articulated and elastic non-rigid motion: a review". Proc. of IEEE Workshope on Motion of Non-Rigid and Articulated Objects. 1994, pp. 2-14.

[76] Alex Pentland, "Looking at people: sensing for ubiquitous and wereable computing", IEEE Trans. on Pattern Analysis and Machine Intelligence, 22 (1) (2000) 107-119.

[77] K.P.Karmann, A. Brandt, "Moving object recognition using an adaptative background memory", in V Cappellini, Time-vaying Image Processing and Moving Object Recognition, 2.Elseiver, Amsterdam, The Netherlands, 1990.

[78] C.R. Wren, A. Azarbayejani, T.Darrell, A.P.Pentland, "Pfinder: real-time tracking of the human body" IEEE Trans. on Pattern Analyisis and Machine Intelligence, 19 (7) (1997) 780-785.

[79] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking. Proc. of IEEE CS Conf. on Computer Vision and Pattern Recognition, Vol. 2, 1999, pp. 246-252.

[80] S.J. McKenna et al., Tracking groups of people, Computer Vision and Image Understanding, 80 (1) (2000) 42-56.

[81] S. Arseneau, J.R. Cooperstock, Real-time image segmentation for action recognition. Proc. of IEEE Pacific Rim Conf. on Communications, Computers and Signal Processing. 1999, pp. 86-89.

[82] H.Z. Sun, T. Feng, T.N. Tan, Robust extraction of moving objects from image sequences. Proc. of the Fourth Asian Conference on Computer Vision. Taiwan, 2000, pp. 961-964.

[83] A. Elgammal, D. Harwood, L. S David, Nonparametric background model for background subtraction. Proc. of the Sixth European Conference on Computer Vision, 2000.

[84] Y.H. Yang, M.D. Levine, The background primal sketch: an approach for tracking moving objects, Machine Vision and applications, 5 (1992) 17-34.

[85] M. Kilger, A shadow handler in a video-based real-time traffic monitoring system. Proc. of IEEE Workshop on Applications of Computer Vision. 1992, pp. 1060-1066.

[86] A.J. Lipton, H. Fujiyoshi, R. S. Patil, Moving target classification and tracking from real-time video. Proc. of IEEE Workshop on Applications of Computer Vision. 1998, pp. 8-14.

[87] C. Anderson, P. Bert, G. Vander Wal, Change detection and tracking using pyramids transformation techniques. Proc. of SPIE-Intelligent Robots and Computer Vision, Vol. 579, 1985, pp. 72-78.d

[88] J.R. Bergen et al., A three frame algorithm for estimating two-component image motion, IEEE trans. on Pattern Analysis and Machine Intelligence, 14 (9) (1992) 886-896.

[89] Y. Kameda, M. Minoh, A human motion estimation method using 3-successive video frames. Proc. of International Conference on Virtual Systems and Multimedia, 1996.

[90] W. Freeman and M. Roth, “Orientation Histogram for Hand Gesture Recognition” Proc. Int'l Workshop Automatic Face and Gesture Recognition, pp. 296-301, 1995.

[91] P. Maes, T. Darrell, B. Blumberg, and A. Pentland, “The ALIVE System: Wireless, Full-Body Interaction with Autonomous Agents” ACM Multimedia Systems, 1996.

[92] Aaron F. Bobick, James W. Davis, “The Recognition of Human Movement Using Temporal Templates”, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 23, No.3, 2001.

[93] M. Hu, “Visual Pattern Recognition by Moment Invariants”, IRE Trans. Information Theory, vol. 8, no. 2, pp. 179-187, 1962.

[94] Cherkassky, V. and F. Mulier, “Learning from Data: concepts, theory and methods”, John Wiley & Sons, Inc. 1998.

[95] J. Bermejo-Alonso, R. Sanz, I. López, “A survey on Ontologies for Agents, From theory to Practice” ASLab ASL-A-2006-XX v 1.0 Draft, June12, 2006.

[96] T.R. Gruber. “Toward principles for the design of ontologies used for knowledge sharing”, in N. Guarino and R. Poli, editos, International Workshop on Formal Ontology in Conceptual Analysis and Knowledge Representation, Padova, italy, 1993. Kluwer Academic Publishers.

[97] T.R. Gruber, “A Translation approach to portable ontologies”, Knowledge Adquisition, 5(2):199-220, 1993.

[98] M.R. Genesereth and L. Nilsson, “Logical Foundations of Artificial Intelligence”, Morgan Kaufmann, Los Altos, California, 1987.

[99] N. Guarino, “Formal ontology and information systems”, In. Guarino, editor, Formal Ontology in Information Systems, FOIS 98, pages 3-15, Trento, Italy, June 1998, IOS Press.

[100] M. Wooldridge and N. R. Jennings, “Intelligent Agents: Theory and Practice”, Knowledge Engineering Review, 10(2):115-152, 1995

[101] Nicolas Eric Maillot, Monique Thonnat, “Ontology based object recognition”, Image and Vision Computing 26 (2008) 102-113, July 2005.

[102] Blázquez M, Fernández M, García-Pinar JM, Gómez Pérez A (1998), “Building ontologies at the knowledge level using the ontology design environment”. In: Proceedings of the 11 th work-shop on knowledge acquisition, pp 18-23

[103] R.Rao and G. Lohse, “Towards a texture naming meaning system: Identifying the dimension of texture”, Visual Research 36 (11) (1993) 1649.

[104] I. López Paniagua, “A Foundation for Perception in Autonomous Systems”, Universidad Politécnica de Madrid, 2007.

[105] J.C.Ruíz,“PercepciónVisual”,http://www.scribd.com/doc/6877744/H016-Percepcion-Visual

[106] James J. Gibson, “The Ecological Approach to Visual Perception”, Lawrence Erlbaum Associates, 1987.

[107] George J. Klir, “Teoría de los Sistemas Generales”, Matemática actual. ICE Ediciones,1980.

[108] Gutta, S., Wechsler, H.:”Face Recognition using asymetric faces”, In:ICBA. (2004) 162-168

[109] Gong, S., McKenna, S.J., Psarrou, A.: “Dynamic Vision: From Images to Face Recognition” Imperial College Press, World Scientific Publishing (2000)

[110] Zhao, W., Chellappa, R., Phillips, P., Rosenfeld, A.: “Face Recognition: A literature survey”. ACM Computing Surveys 35 (2003) 399-458.

[111] Kong, S.G., Heo, J., Abidi, B. R., Paik, J., Abidi, M.A.: “Recent advances in visual and infrared face recognition” Computer Vision and Image Understanding 97 (2005) 103-135.

[112] Cox, I., Ghosn, J., Yianilos, P.: “Feature-based face recognition using mixture-distance” In: Proc. Int. Conf. Comput. Vis. Patt. Recogn. (1996) 209-216

[113] Gao, Y., Leung, M.: “Face Recognition using line edge map” IEEE Transactions on Pattern Analysis and Machine Intelligence 24 (2002) 764-779

[114] Wiskott, L., Fellous, J., Krugern, N., von der Malsburg, C.:”Face Recognition by elactic bunch graph matching” IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (1997) 775-779.

[115] Samaria, F.:”Face Recognition using Hidden Markov model”, PhD thesis, Univ. of Cambridge (1994)

[116] Nefian, A., Hayes. III, M.:"Hidden Markov Models for face recognition", In: Proc. Int'l Conf. Acoustics, Speech, and signal Processing, vol. 5(1998) 2771-2774.

[117] Kohir, V., Desai, E.,: "Face recognition" In : Proc. Image Processing. (2000) 309-312.

[118] Othman, H., Aboulnasr, T.: "A separable low complexity 2D HMM with application to face recognition" IEEE Trans. Patt. Anal. Mach. Intell. 25 (2003) 1229-1238.

[119] Bartlett, M.: "Face Image Analysis by Unsupervised Learning" Kluwer Academic publishers (2001).

[120] Swets, D., Weng, J.: "Using discriminant eigenfeatures for image retrieval" IEEE transactions on Pattern Analysis and Machine Intelligence 18 (2003) 115-137.

[121] Bellhumeour, P., Hespanha, J., D.J., K.: "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection

[122] Blanz, V., Vetter, T.: "A morphable model for the syntesis of 3PH. (1999) 187-194.

[123] Blanz, V., Vetter, T.: "Face Recognition based on fitting a 3D morphable model". IEEE Transactions on Pattern Analysis and Machine Intelligence 25 (2003) 1063-1074.

[124] Cootes, T., Taylor, C., Cooper, D., Graham, J.: "Active shape models-their training and application" Computer Vision and Image Understanding 61 (1995) 38-59.

[125] Chen, L., Man, H., Nefian, A.V.: "Face Recognition based on multi-class mapping of Fisher scores" Pattern Recognition 38 (2005) 799-811.

[126] Gibson, J. J. (1974), "The perception of the Visual World", Houhgton Mifflin Company, Boston

[127] Richard O. Duda, Peter E. Hart, "Use of Hough Transformation to detect lines and curves in pictures", Technical note 36, April 1971.

[128] Bryan Adams, Cynthia Breazeal, Rodney A. Brooks, Brian Scassallati, "Humanoid Robots: A New Kind of Tool", MIT Artificial Intelligence Laboratory

[129] http://mitpress.mit.edu/catalog/item/default.asp?ttype=2&tid=10515

[130] http://www.mit.edu/~sturkle/

[131] http://en.wikipedia.org/wiki/Walter_de_Gray

[132] Fritz, W., García Martínez, R., Rama, A., Blanqué, J., Adobatti, R, y Sarno, M. 1989 "The Autonomous Intelligent System". Robotics and Autonomous Systems, 5(2): 109-125

[133] http://en.wikipedia.org/wiki/Munsell_color_system