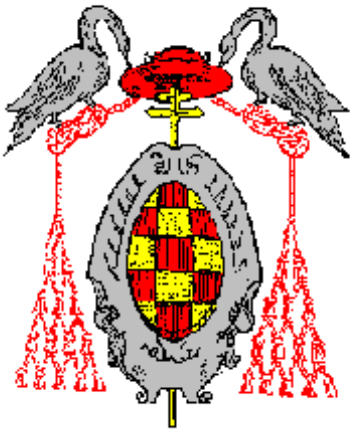
UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA

DEPARTAMENTO DE ELECTRÓNICA



SEGUIMIENTO FACIAL, MEDIANTE VISIÓN ARTIFICIAL, ORIENTADO A LA AYUDA A LA MOVILIDAD

TESIS DOCTORAL

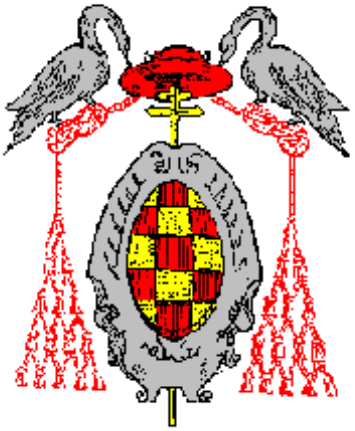
LUIS MIGUEL BERGASA PASCUAL

1999

UNIVERSIDAD DE ALCALÁ

ESCUELA POLITÉCNICA

DEPARTAMENTO DE ELECTRÓNICA



SEGUIMIENTO FACIAL, MEDIANTE VISIÓN ARTIFICIAL, ORIENTADO A LA AYUDA A LA MOVILIDAD

Autor: D. LUIS MIGUEL BERGASA PASCUAL

Director: Dr. D. MANUEL MAZO QUINTAS

1999

TESIS DOCTORAL

Resumen:

En esta tesis se propone un sistema de guiado de una silla de ruedas motorizada en entornos interiores, mediante seguimiento facial, para ayudar a la movilidad de personas minusválidas y aplicando técnicas de visión artificial. Se trata de un sistema de control en lazo cerrado en el que el usuario genera una serie de comandos a alto nivel, mediante los movimientos de cabeza, ojos y boca, que permiten mover la silla a voluntad. El sistema es no intrusivo, fácil de manejar, adaptativo a cambios de iluminación y entorno, y aplicable a cualquier persona, independientemente de su sexo y raza, siempre que no tenga problemas de movilidad de cabeza.

Para llevar a cabo estos objetivos, se ha utilizado como sensor una micro cámara de vídeo en color, en una configuración de cámara fija solidaria con la silla, que captura continuamente un plano de la cabeza del usuario. Sobre estas imágenes, se aplica un segmentador del color de la piel en tiempo real, basado en un modelo estocástico gaussiano adaptativo y no supervisado. A dicho segmentador se le ha denominado UASGM (modelo gaussiano de la piel adaptativo y no supervisado) y supone una de las principales aportaciones de la tesis. Se ha realizado un estudio de diferentes espacios de color concluyendo que el óptimo para esta aplicación era el RG (rojo, verde) normalizados. Asimismo se ha demostrado que el empleo de un modelo gaussiano es ajustado al problema. Para inicializar el modelo se ha usado un método de "clustering" por aprendizaje competitivo mediante el algoritmo VQ (Vector Quantization), con ello se consigue un ajuste del modelo a cada usuario. El número de clases óptimo para el "clustering" se ha resuelto aplicando una modificación del ratio generalizado de Fisher. El modelo se adapta mediante una combinación lineal de los parámetros ya conocidos del mismo, siguiendo el criterio de máxima probabilidad

Se ha demostrado que el método de "clustering" utilizado es una simplificación del método de mezcla de múltiples gaussianas en el modelado de histogramas y de su resolución mediante el algoritmo EM (Expectation-Maximization). Experimentalmente también se ha demostrado que los resultados obtenidos, respecto al número de clases óptimo del proceso de "clustering", mejoran o igualan a otros métodos basados en modelos de múltiples gaussianas. Por otro lado, con el método propuesto, se han logrado mejores resultados de segmentación que aplicando el GLVQ-F (Fuzzy Generalized Learning Vector Quantization), que utiliza el conocido algoritmo de "clustering" FCM (Fuzzy C-Means).

Sobre el objeto segmentado piel se realiza un seguimiento mediante estimación de parámetros por filtro de Kalman. El vector de estados estimado se introduce a una máquina de estados, previamente ajustada para el usuario, que genera los comandos a alto nivel de la silla. Estos comandos se envían a otra máquina de estados que genera las consignas de velocidad lineal y angular de la silla. Aplicando el modelo cinemático de la misma, estas velocidades se transforman en velocidades angulares para cada rueda y se envían a un módulo de control a bajo nivel, a través de una red de comunicaciones ECHELON, donde hay implementados dos controladores PI. También se ha añadido otro módulo de localización de ojos y boca que permite la activación de los comandos especiales de "on/off" y sentido, y que se sumarán a los creados mediante los movimientos de cabeza.

Para ayudar al usuario a adaptarse al sistema de guiado propuesto, se ha diseñado un entorno de simulación en 3D donde se reproduce la primera planta de la antigua Escuela Politécnica de la Universidad de Alcalá. Con este sistema se pueden simular los movimientos de la silla sin peligro alguno para el usuario.

Para demostrar las prestaciones del sistema propuesto, se han realizado pruebas sobre una de las sillas prototipo del proyecto SIAMO (que debió de ser convenientemente adaptada para permitir este tipo de guiado) con distintos usuarios. Finalmente, se presenta los resultados de una encuesta realizada a los mismos sobre la controlabilidad y rendimiento del sistema.

Abstract:

In this thesis, a guidance system is proposed for an autonomous wheelchair in outdoor environments, by means of face tracking, to aid the mobility of disabled people using computer vision. In the close loop system, the user provides high level commands,

by head, eyes and mouth movements, that allow to move the wheelchair at will. The system is not intrusive, easy to handle, adaptive to lighting conditions and applicable to any person, regardless of sex and race, if no head mobility problems are exhibited.

To carry out these objectives, a micro colour video camera has been placed on a fixed position in the chair to continuosly capture an image containing the user head. A new real time skin clustering process is applied on these images, based on an Unsupervised Adaptive Sthocastic Gaussian Model. This process denoted as UASGM constitutes the greatest contribution of this work. Different colour spaces have been studied concluding that the normalized RG space is the optimum one for skin clustering purposes. Likewise, the use of a Gaussian model has been proved to be appropriate to solve this problem. A competitive learning method called VQ (Vector Quantization) has been used to initialize the model, achieving a particular model adjustment for each user. A modification of the Fisher generalized ratio was utilized to determine the optimum number of classes to perform the clustering algorithm. The model is adapted by a linear combination of the already known parameters, following the Maximum Likelihood criterium.

It has been proved that the clustering method developed in this thesis is a simplification of the multiple Gaussians mixture, used in histogram modelling, and the Expectation-Maximization (EM). Experimental results achieved in this work have proved to be equal or better than those yielded by other methods based on multiple Gaussian functions. On the other hand, this method provides better results than the GLVQ-F (Fuzzy Generalized Learning Vector Quantization), that uses the popular FCM (Fuzzy C-Means) clustering algorithm.

A Kalman filter has been developed to track the skin blob on the image plane, once the clustering process finishes. The system state vector is introduced to a finite state machine, previously adjusted for each particular user, to provide high level commands for the wheelchair. These commands are sent to another finite state machine that issues the linear and angular velocities of the chair. Using the kinematic model, these velocities are translated to angular velocities for each wheel, and sent to the low level controllers via a LonWorks bus (by ECHELON) to perform a traditional PI control. Additional vision based modules provide the system with the ability to locate eyes and mouth on the image plane, to activate special commands like, On/Off, Forward/Backward, that work in conjuction with the commands generated by head movements.

The user adaptation to the proposed guidance system is aided by a 3D simulator, where the first floor of the former Polytechnic School of the University of Alcalá is emulated. Wheelchair movements can be simulated using this 3D environment as a sort of training, without putting the user in danger.

To demonstrate the proposed system capabilities, several tests have been performed on a prototype wheelchair (specially adapted for this kind of guidance) in the frame of the SIAMO project. Finally, the results of a quiz evaluated on different users about the system controlability and performance, are presented.

A mis padres, hermanos y a mi novia, Cristina

AGRADECIMIENTOS

Quiero mostrar mi agradecimiento a todas aquellas personas que, de manera directa o indirecta, han contribuido al desarrollo y finalización de esta tesis. Especialmente, a mi director de tesis, Manuel Mazo, por su completa disponibilidad para atenderme y asesorarme a lo largo del camino recorrido. A Alfredo Gardel, por su desinteresada ayuda en la programación de los algoritmos, y por todos los momentos compartidos que han forjado una amistad sincera. A Alfredo Ortuño, por haber programado el simulador 3D y haber estado siempre dispuesto a asumir todas las modificaciones pedidas. A Miguel Ángel Sotelo por sus indicaciones certeras y por haber estado siempre dispuesto a echar una mano. A Luciano Boquete, y nuevamente a Manuel Mazo y Miguel Ángel Sotelo, por haber revisado el borrador de esta tesis. A Juan Carlos García, Eduardo Sebastián, Marta Marrón y Elena López por haberme ayudado en mis pinitos con el ECHELON.

A todos los alumnos a los que he dirigido el TFC en temas relacionados con visión artificial aplicada al seguimiento facial, especialmente a Miguel Ángel García y Ricardo Asensio, por ser los primeros que se embarcaron conmigo en esta aventura.

A todos los compañeros del pasillo 4 por haber creado el clima de trabajo propicio y por haberme alentado siempre, en esta difícil tarea. A los demás compañeros del Departamento de Electrónica que han seguido este trabajo, escuchando mis presentaciones y realizando múltiples sugerencias.

A mis compañeros del Coro Universitario por haberme animado en mi empeño y por haber sacado de mi una sonrisa cuando las cosas no iban bien.

A mi novia, Cristina, por sus meticulosas revisiones del idioma, por haber entendido mis ausencias y por su apoyo incondicional en todo momento.

1. INTRODUCCIÓN

1.1.- PROBLEMÁTICA DE LA AYUDA A LA MOVILIDAD

En las últimas décadas la tecnología de robots móviles ha sufrido un gran avance en laboratorios de investigación, de forma que están empezando a ser utilizados masivamente en un amplio rango de aplicaciones que comprende desde la industria hasta los hogares. Sin embargo, tradicionalmente la mayoría de estas aplicaciones han ido destinadas a solventar problemas industriales, mientras que un gran número de personas mayores y minusválidas se encuentran todavía esperando soluciones que la robótica pueda dar a algunos de sus problemas.

Sin tener en consideración las connotaciones sociales y humanas que pueden intervenir en la realización de sistemas de ayuda para este sector de la población, según los datos proporcionados por el Instituto de Migraciones y Servicios Sociales (IMSERSO), la población española mayor de 65 años es de 6.182.899 personas, lo que representa el 15,72% del total de la población española (39.323.320). Se trata de un porcentaje bastante elevado que seguirá aumentando en los próximos años. Se estima que dicho sector de la población superará el 21% dentro de 30 años y que una décima parte de los mismos necesitará de ayudas técnicas para su movilidad.

El informe estadístico más importante sobre población discapacitada en España proviene de la Encuesta sobre Discapacidades, Deficiencias y Minusvalías que, en 1986, fue realizada por el Instituto

Nacional de Estadística (INE). Este estudio fue complementado en 1988 por un análisis cualitativo de las interrelaciones entre la minusvalía y el contexto social llevado a cabo por el INSERSO (Instituto Nacional de Servicios Sociales). De éste se deduce que la discapacidad para andar engloba a cerca de 870.000 personas, de las cuales 120.000 precisa constantemente una silla de ruedas para desplazarse (0,32% de la población total) y 750.000 la asistencia de otra persona o de una ayuda técnica (1,95% sobre la población total) [Poveda et al., 98].

En la tabla 1.1. se muestra la etiología habitual del colectivo permanente de usuarios de sillas de ruedas.

Diagnóstico	Abreviatura	Incidencia (%)
Amputación del miembro inferior	AMP	3,4%
Artritis	ART	7,2%
Ataxia	ATX	9,5%
Esclerosis lateral amiotrófica	ELA	3,6%
Esclerosis múltiple	EM	8,1%
Espina bífida	EB	7,9%
Hemiplejía	HEM	5,2%
Miopatía	MIO	8,8%
Parálisis cerebral	PC	11,3%
Paraplejía	PAR	12,0%
Secuelas poliomelitis	POL	7,2%
Tetraplejía	TET	14,9%
Otros	OTR	0,7%

Tabla 1.1. Etiología de usuarios de sillas de ruedas

En un análisis de la movilidad en extremidades superiores e inferiores de la población encuestada, se valoró ésta en cuatro niveles: conservada, limitada, parálisis y amputación, asignando a cada nivel 3, 2, 1 y 0 puntos respectivamente. En la figura 1.1 se muestran los resultados obtenidos tras promediar las puntuaciones de movilidad de cada sujeto para sus dos brazos y sus dos piernas. Se aprecia que los lesionados medulares exhiben una movilidad muy cercana a 1, que corresponde a una parálisis de miembros inferiores, tal y como cabe esperar. Por contra, son los indivuduos con hemiplejía,

miopatías, ataxias y artritis los que presentan una mayor movilidad en sus piernas, con un nivel 2 equivalente a movilidad limitada.

En cuanto a la movilidad de miembros superiores, son los grupos de espina bífida, amputados y parapléjicos los que mejor movilidad conservan, mientras que los enfermos de ELA destacan por tener la movilidad de brazos más reducida, con una media apenas superior a 1.5. El sistema propuesto en esta tesis podría ser utilizado por todas aquellas personas con patologías que, aún así, puedan mover su cabeza voluntariamente.

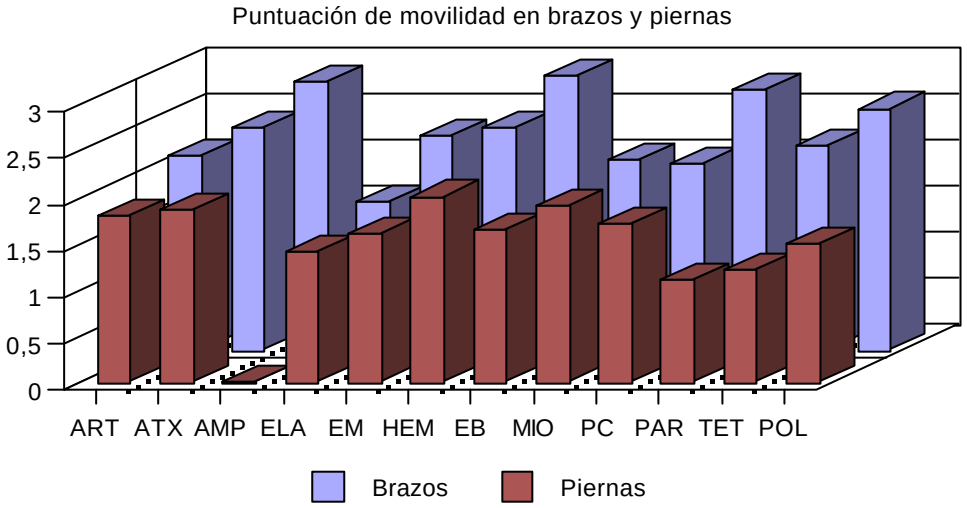


Figura 1.1. Estudio de movilidad de brazos y piernas para usuarios de sillas de ruedas

Pese a estos datos, en nuestro país apenas se desarrolla tecnología en este campo, lo que dificulta aún más el acceso a la misma. Una buena prueba de ello es que España importa el 70% de la tecnología para ayudar a las personas con algún tipo de discapacidad manifiesta.

Conscientes del vacío existente en aplicaciones destinadas a este sector de la población, en los últimos años los gobiernos y las instituciones públicas han promovido investigaciones en esta línea, siendo varios los grupos de investigación a nivel mundial que se han embarcado en el desarrollo de proyectos de cooperación y ayuda a la movilidad de personas mayores y/o con algún tipo de minusvalía. Así, cuando en 1995 los presidentes Jacques Santer y Bill Clinton decidieron impulsar acuerdos de

colaboración transatlántica, señalaron como una de las prioridades en el campo de la acción social la integración laboral de la población discapacitada a través de las nuevas tecnologías.

El objetivo de estos proyectos es el aumento de la autonomía de la persona en sus movimientos y en la realización de tareas diarias. Una silla de ruedas motorizada estándar ayuda a la movilidad de personas minusválidas que no pueden andar, siempre y cuando su discapacidad les permita el manejo del joystick de una manera segura. Sin embargo, su uso puede resultar costoso o imposible para personas con gran minusvalía. Por ejemplo, algunas personas con tetraplegia sólo son capaces de gobernar un sensor de on-off. Esto hace que el control de la silla sea particularmente difícil, sobre todo en maniobras delicadas. Para estos casos es necesario desarrollar interfaces hombre-silla más complejas y adaptadas a la minusvalía del usuario, que les permita introducir consignas de movimiento de una manera segura y sencilla.

Por otro lado, se pueden introducir automatizaciones en las sillas de ruedas que permitan realizar movimientos de una manera semiautomática o automática, sin necesidad de que el usuario esté supervisando el movimiento (guiado autónomo). Todos estos sistemas pueden también ser utilizados por usuarios habituales de sillas de ruedas y/o personas mayores que, aunque sean capaces de moverse por medios más convencionales, ven mejorado su confort en los movimientos mediante el empleo de estas técnicas, especialmente en maniobras delicadas (como por ejemplo el paso por puertas estrechas). Esta última contribución sería similar a la instalación de un sistema de asistencia en la conducción de un coche.

El diseño de este tipo de sistemas agrupa gran cantidad de disciplinas del campo de la electrónica y la robótica, que pueden ser clasificadas en los siguientes grupos: percepción, planificación, control e interacción hombre-máquina [Matía et al., 98][Mazo et al.,95a]. Mediante el empleo de estas técnicas se generan una serie de primitivas que permiten el movimiento del usuario. En [Bourhis&Agostini,98] se hace una división de las mismas, en función de la cantidad de información que necesitan del entorno, en dos grandes grupos:

Primitivas locales:

• *Detección de obstáculos*: consiste en una simple prevención de colisión y de caída si existen escaleras no protegidas. Para ello se introducen sensores de colisión ('bumpers'), sensores de ultrasonidos, telémetros láser, sistemas de visión con una cámara CCD (o dos para sistemas en estéreo), sistemas de visión con luz estructurada (compuesto por una cámara fija acoplada con un emisor láser de infrarrojo), etc.

• *Evitación de obstáculos*: los obstáculos son detectados y el robot gira alrededor de ellos siguiendo una estrategia basada únicamente en la información de los sensores. Se usan los mismos sensores que en la primitiva anterior.

• *Seguimiento de muros*: esta primitiva puede ser particularmente interesante en pasillos largos para personas que tienen dificultades para mantener un camino recto durante largo tiempo. Se suelen usar los mismos sensores que para detectar obstáculos.

• *Seguimiento de personas*: la silla de ruedas motorizada sigue a una persona que le guía sin ningún esfuerzo por parte de ésta última. Los sensores más utilizados son los de visión.

• *Paso a través de puertas*: esta primitiva es una asistencia a maniobras delicadas en pasos estrechos. Fundamentalmente utilizan telémetros, visión y ultrasonidos.

• *Maniobras de atraque*: es un sistema de control de final de movimiento que ayuda a realizar algunas maniobras típicas de acercamiento como son: posicionamiento en una mesa, en un baño, en la cama, etc. Utiliza los mismos sensores que en la etapa anterior.

Primitivas globales

• *Seguimiento de caminos automáticamente*: pueden ser grabados mediante aprendizaje o planificados mediante un sistema de posicionamiento global. Se emplean sensores odométricos, láseres rotatorios que localizan marcas sobre posiciones fijas, sistemas de visión que igualmente localizan marcas o sistemas de posicionamiento absoluto por satélites (GPSs).

• *Ayudas a la localización*: algunas personas sufren de pérdida de memoria y no son capaces

de planificar su camino. Esta primitiva es la más ambiciosa de todas, ya que plantea una localización automática de un usuario y la posibilidad de planificar un camino hasta un destino dado por éste y de ejecutarlo de forma automática. En este caso es imprescindible el empleo de GPSs y la localización de marcas mediante los sensores ya nombrados.

Las funcionalidades descritas constituyen una lista exhaustiva de primitivas que podrían implementarse en sistemas de asistencia a la movilidad. Sin embargo, cuando se intenta materializar alguna de ellas, el diseñador encuentra que estos sistemas son muy variantes en función de la discapacidad del usuario potencial y de su edad. Al no existir estudios en profundidad sobre las diferentes minusvalías y las afecciones que éstas introducen sobre los sistemas motrices del cuerpo, en la mayoría de los casos es el propio diseñador el que debe hacer un estudio de campo y extraer el modo de guiado y el interfaz hombre-silla más idóneo para cada caso. No obstante, para este tipo de aplicaciones, se deberían seguir una serie de especificaciones como las que se detallan a continuación.

1.2.- ESPECIFICACIONES GENERALES PARA EL DISEÑO DE UNA APLICACIÓN DE AYUDA A LA MOVILIDAD.

Las sillas de ruedas "inteligentes" pertenecen a lo que se conoce como *robótica de rehabilitación*. Esta línea de la robótica tiene las siguientes particularidades:

- El robot comparte su espacio con personas y otros robots.
- El usuario tiene algún tipo de minusvalía que le impide tener el 100% de sus facultades.
- Los robots se suelen mover en entornos interiores particularmente estructurados por muros y puertas que constituyen marcas naturales. Sin embargo, en situaciones reales pueden aparecer numerosas modificaciones del modelo del entorno: puertas cerradas, objetos no contemplados o personas con otras sillas que pueden entorpecer o prohibir el paso.

Sistema modular y versátil

Una característica de la aplicación es la gran diversidad de usuarios y situaciones para las que va dirigida. El entorno puede ir desde la casa del usuario hasta un edificio público, como puede ser un

centro de rehabilitación o un hospital, por lo que las distancias a recorrer son muy variantes y de características diferentes. Por otro lado, los usuarios tienen una gran variedad de posibilidades físicas y psíquicas.

El interfaz hombre-máquina debe ser modular y configurable al igual que los sensores. Estos últimos se pueden clasificar, desde un punto de vista funcional, en tres categorías:

• *Sensores proporcionales*: son generalmente sensores controlados por la mano (joysticks) y menos frecuentemente controlados por la barbilla. Otros lugares anatómicos utilizados son, por ejemplo, la nuca [Ferrario&Lodola, 92] o los pies [Audet et al., 95].

• *Sensor de on-off:* es un simple conmutador controlado, por ejemplo, con movimientos de cabeza, emisión de sonidos, soplidos, etc. Para poder generar consignas de movimiento mediante este método es necesario asociar la activación del conmutador con un conjunto de opciones elegibles en una pantalla.

• *Dos (o más) sensores de on-off*: se genera un conjunto limitado de consignas. Dentro de este grupo se encuentran: sensores de soplo [García et al, 97] (cuando se sopla hacia afuera se activan unas consignas y cuando se sopla hacia adentro otras), reconocedores de voz [Mazo et al.,95b] (son sistemas que reconocen un conjunto limitado de palabras y por lo tanto se les puede incluir dentro de esta categoría), sensores por movimientos oculares [Bergasa et al., 96] (en función de la posición de los ojos se activan diferentes comandos).

Los datos obtenidos de los diferentes tipos de sensores deben ser filtrados, sobre todo en personas con grandes minusvalías, para poder distinguir entre movimientos voluntarios y gestos involuntarios.

Compromiso coste-funcionalidad

En una aplicación destinada a personas con minusvalías hay que tener en consideración la tecnología a utilizar debido a que no es un producto de gran público. Siempre que sea posible es preferible utilizar equipos no específicos para la aplicación. Así, por ejemplo, a no ser que se necesite una funcionalidad mecánica suplementaria, como la característica de la omnidireccionalidad, es preferible utilizar sillas de

ruedas comerciales como base para la aplicación. Aparte de los problemas prácticos de la implementación de sensores, estas aplicaciones introducen otros relativos a la navegación de robots autónomos: por un lado, la planta tiene características no holonómicas y, por otro, las medidas de los sistemas odométricos son poco fiables debido a diversas causas: ruedas hinchadas desigualmente, distribución variable y no uniforme del peso de la persona sobre la máquina, superficie, etc.

La elección de un sensor también depende de un criterio de coste. La utilización de sensores de ultrasonidos, infrarrojos y cámaras CCD estándares, abaratan grandemente el coste del producto al emplear elementos de consumo masivo. El número de sensores a utilizar y el tipo dependerá de la aplicación, pero procurando seguir el criterio de emplear el más barato siempre que cumpla los requisitos que se le piden.

Aceptabilidad psicológica

El empleo de sistemas robóticos de ayuda a personas también introduce un problema psicológico de aceptabilidad. Hay que tener especial cuidado en la realización final del diseño procurando que su implementación sea discreta. El confort de la silla es un parámetro a cuidar; así, por ejemplo, si el usuario tiene problemas para mantener su cabeza recta, controlará la silla con dificultad, lo que puede originar cambios bruscos de velocidad y/o dirección que pueden ser peligrosos. Es por ello que el sistema de control no debe permitir cambios bruscos en estas variables para evitar estos posibles casos.

La concepción técnica de la silla también tiene que tener en consideración factores psicológicos. En primer lugar el interfaz entre hombre-silla tiene que ser muy amigable para el usuario teniendo en cuenta que éste, a priori, no es una persona técnica. Es necesario establecer un diálogo con la silla, por muy reducidas que sean las capacidades físicas o psíquicas del usuario. Para ello, habrá que definir un mínimo número de consignas que deberán ser generadas por el usuario aprovechando sus capacidades. Por otro lado, los movimientos automáticos de la silla tienen que ser naturales para el usuario. Los cambios de dirección deben ser predecibles, bien por ser naturales, bien porque la silla previamente disminuye la velocidad o porque es una orden dada por el usuario a través de su interfaz hombre-silla.

Para su total aceptabilidad es necesario que éste se sienta continuamente “dueño” del sistema. Hay

que darle la posibilidad de que realice la navegación por sí mismo. Incluso en los movimientos automáticos de la silla tiene que ser capaz de intervenir si aparecen problemas, parándola en el peor de los casos. Además las órdenes tienen que ser lo suficientemente intuitivas como para inspirar la confianza del operador.

Interfaz de usuario

Los requerimientos esenciales de una interfaz de usuario son capturar las intenciones del mismo de una manera intuitiva y producir una realimentación sensorial del proceso. Debería comenzar permitiendo solamente un nivel básico de funcionalidad. Cuando los usuarios estuvieran confortables con este nivel, entonces se podrían introducir nuevas características de forma incremental. Un ejemplo típico es el control de velocidad de un robot para ayudar a caminar. El usuario comienza con una velocidad baja y fija y después de un tiempo quiere tener control sobre la velocidad.

La complejidad del sistema robot está asociada a la velocidad y al ancho de banda de la generación de comandos, entendiendo este último como el número de consignas diferentes que se pueden generar, ya que determina la suavidad de los movimientos del robot. Pueden ser muy suaves, como ocurre en un control directo por joystick, o más abrupto, como ocurre en un control basado en voz. A mayor número de comandos la suavidad del movimiento será mayor y viceversa. Hay que hacer un estudio del número de comandos necesarios para obtener la velocidad y la suavidad deseadas.

El usuario tiene que tener una realimentación sensorial de los movimientos que se ejecutan teniendo en cuenta que se necesita una mayor realimentación cuando se conduce hacia atrás que cuando se hace hacia adelante. Una buena forma de realimentación es la sonora ya que la información visual puede ocasionar el "despiste" del conductor.

Robustez y seguridad

La robustez y la seguridad son particularmente importantes en aquellas aplicaciones donde el operador está situado encima del robot a gobernar, y además tiene reducidas sus capacidades físicas y/o psíquicas. Es esencial excluir todas las posibles situaciones de bloqueo aunque esto suponga una

degradación en el funcionamiento del robot. Por ejemplo, en el modo manual con parada de seguridad el piloto debe poder reanudar el control en cualquier momento. Puede ser el caso de que la silla esté bloqueada debido a un área de seguridad que podemos haber establecido alrededor de ella o bien una pérdida de localización en un movimiento planificado.

En cualquier situación la parada de la silla se debe ejecutar por una simple acción generada por un comando. Teniendo en cuenta la gran diversidad de posibles usuarios, este comando de seguridad debe ser adaptado a cada caso particular. Sin embargo, como ya han manifestado varios autores [Harwin&Rahman, 92][Van Der Loos et al., 92] la total precisión y seguridad es imposible. Por lo tanto, es bueno conocer la máxima relación aceptable de coste/riesgo de acuerdo con el beneficio esperado.

1.3.-ENTORNO DE DESARROLLO DE LA TESIS

Esta tesis se ha desarrollado dentro del proyecto de investigación TER96-1957-C03-01 financiado por la Comisión Interministerial de Ciencia y Tecnología (CICYT), una de cuyas partes se ha desarrollado en el Departamento de Electrónica durante el período 1996-99.

El grupo de investigación de asistencia a la movilidad del Departamento de Electrónica lleva trabajando en este campo más de 6 años. Como resultado de estos trabajos, se desarrollaron varios prototipos de sillas de ruedas motorizadas con diferentes alternativas de guiado [Mazo et al., 95a][Mazo et al., 95b]. Después de la experiencia adquirida en estos proyectos, se decidió abordar un proyecto más amplio en la misma línea para diseñar una silla de ruedas motorizada que contemplara el diseño de un sistema modular, la incorporación de guiados alternativos para casos de personas con severa discapacidad (guiado por el movimiento de la cabeza, movimiento de ojos y expulsión de aire), un completo interface usuario-máquina, un módulo de control de motores y sistemas de seguridad y de guiado autónomo (paso a través de puertas, seguimientos de paredes, ejecución de caminos pregrabados, etc). Dicho proyecto recibió el nombre de SIAMO (Sistema Integral de Ayuda a la MOvilidad).

Dentro de los objetivos planteados en este proyecto, el estudio, diseño y desarrollo de la interfaz

hombre-silla que permitiera el guiado de la misma mediante los movimientos de la cabeza ha motivado la realización de esta tesis. Los objetivos planteados en la misma se detallarán en el capítulo tres.

1.4.-ESTRUCTURA DE LA TESIS

En la redacción de la tesis se ha realizado una división en capítulos atendiendo a los principales temas acometidos en el desarrollo de la misma. En el capítulo 2 se realiza una descripción del estado del arte relativo a los proyectos más significativos de ayuda a la movilidad mediante sillas de ruedas motorizadas. Asimismo se realiza un estudio de diferentes aplicaciones que emplean seguimiento facial y de la mirada y se analizan las diferentes técnicas usadas para ello. Por último se revisa el estado del arte respecto a la técnica de visión artificial aplicada al seguimiento facial, al ser ésta la utilizada en el desarrollo de esta tesis.

El capítulo 3 presenta la configuración general del sistema SIAMO y ubica el módulo de guiado mediante movimientos de cabeza dentro del proyecto. Plantea la problemática del empleo de esta técnica, indicando los objetivos inicialmente marcados para la tesis. Se da una descripción general del sistema desarrollado y por último se indican las aportaciones. En el capítulo 4 se presenta el sistema segmentador de piel en color que permite obtener los pixels de piel de una imagen en color con fondo aleatorio, de forma adaptativa e independientemente del usuario. En el capítulo 5 se plantea el método utilizado para realizar el guiado mediante seguimiento facial, empleando para ello los movimientos de la cabeza, ojos y boca. En el capítulo 6 se da una descripción de la plataforma de pruebas utilizada para probar el método de guiado, así como del sistema de visión empleado. También se incluye un simulador en 3D desarrollado para que el usuario pueda entrenarse en este tipo de guiado. Por último se aportan los resultados de diferentes pruebas de guiado tanto para el simulador como para la plataforma real. Finalmente, el capítulo 7 aporta las conclusiones finales e indica las futuras líneas de actuación que se derivan del trabajo aquí realizado.

2. ESTADO DEL ARTE

2.1.- PROYECTOS MÁS SIGNIFICATIVOS DE AYUDA A LA MOVILIDAD

La mayoría de las aplicaciones, descritas en la literatura, desarrolladas para ayudar a la movilidad de personas discapacitadas y/o mayores, mediante sillas de ruedas motorizadas, implementan primitivas locales de control. La más simple consiste en el seguimiento de una línea pintada en el suelo. En [Wakaumi et al, 92] se realiza un guiado por seguimiento de líneas mediante sensores magnéticos situados bajo el soporte de los pies. El sistema se detiene en lugares predeterminados que son localizados mediante sensores de infrarrojos. Si un obstáculo se cruza en el camino preestablecido la silla para y espera a que el camino quede libre. Otro ejemplo de este tipo es el proyecto "**Slingan**"[Melén, 98] desarrollado en el "Stockholm Snoezelen Centre". El proyecto **"TetraNauta"** desarrollado entre la Universidad de Sevilla y la del País Vasco en colaboración con el hospital nacional de parapléjicos de Toledo [Civit&Abascal, 98] está desarrollando un prototipo de silla capaz de seguir marcas en el suelo (líneas) así como marcas de posicionamiento global en las paredes, mediante cámaras.

El sistema "Navchair" desarrollado en la Universidad de Michigan [Bell et al.,94] es una silla eléctrica semiautomática que está equipada con un anillo de sensores de ultrasonidos. Para su navegación utiliza dos algoritmos basados en la información recibida de los sensores de ultrasonidos. El algoritmo EERUF ("Error Eliminating Rapid Ultrasonic Firing") permite una lectura rápida de los sensores y no se ve afectado por el fenómeno de "crosstalk". Para un modelo local del entorno, el algoritmo VFH ("Vector Field Histogram") da la mejor dirección a seguir en función de los obstáculos

y la dirección apuntada por el joystick. El control de la silla es realizado por el operador humano y por la máquina. Los autores enfatizan el bajo coste de su sistema y el diseño de sus interfaces con el estándar M3S, el cual permite que varios equipos de rehabilitación puedan ser conectados juntos.

El estándar **M3S (Multiple Master Multiple Slave),** fue desarrollado en el "TNO Institute of Applied Physics, Delft, Netherlands" [Nelisse, 98]. Se trata de una estrategia de integración usando una arquitectura modular aplicada a robots móviles en el campo de la rehabilitación. Este grupo planteó el problema de que los sistemas de ayuda aplicados a la mejora de la movilidad que se estaban desarrollando en el mundo se hacían de forma no coordinada, lo que daba lugar a la creación de productos incompatibles entre sí, muy limitados y poco flexibles que ocasionaban muy frecuentemente la introducción de modificaciones con un considerable coste adicional para el producto. Un ejemplo de este problema era el diseño de una silla de ruedas para personas con gran discapacidad (espina dorsal dañada, distrofia muscular, parálisis cerebral, etc). Normalmente estas personas suelen tener una silla de ruedas gobernada por un joystick. La utilización de sistemas que permitan un guiado alternativo por otras técnicas implica la introducción de interfaces usuario-máquina muy sofisticados. Asimismo, el empleo de dispositivos de asistencia adicional como: manipuladores (para ponerse gafas, pasar páginas, insertar un floppy en un ordenador, etc); controladores remotos por infrarrojos (para abrir puertas, cerrar las cortinas, etc), requieren entradas para dispositivos adicionales como: pequeños teclados y conmutadores específicos.

Mediante esta estrategia de integración basada en una arquitectura modular se permite al usuario compilar un paquete específico de ayudas técnicas de un sistema integral completo que puede ser extendido o modificado a posteriori por deseos, necesidades y/o cambios de entorno del usuario. M3S permite la conexión de diversos dispositivos de una manera segura, fácil de usar y sin adaptaciones especiales usando las capacidades "plug-and-play" del sistema. Produce interfaces estándar entre dispositivos de entrada y actuadores, permitiendo que dispositivos de diferentes fabricantes sean ensamblados en el mismo sistema. Por otro lado, se da la posibilidad de que un usuario pueda utilizar más de un actuador usando un sólo dispositivo de entrada. Las especificaciones del M3S es un estándar abierto y disponible de forma gratuita y ha sido propuesto al ISO para su estandarización formal. (Mas información sobre el proyecto puede ser adquirida en el servidor web: http://www.tno.nl/m3s).

El **"Smart Wheelchair"** desarrollado en el "CALL Center" de Edimburgo integra varios sistemas de

ayuda a la movilidad (detección de obstáculos, evitación automática de éstos, seguimiento de líneas) y comunicaciones (distintas interfaces hombre-máquina, sintetizador de voz, dispositivos de scanner, etc) [Craig&Nisbet, 93], consiguiendo una silla altamente configurable. Además de la asistencia a la movilidad el sistema es una herramienta que permite desarrollar las habilidades cognitivas de niños jóvenes con problemas motrices (aprendizaje del entorno, comportamiento social).

El proyecto **OMNI (Office Wheelchair with High Manoeuvrability and Navigational Intelligence for People with Severe Handicap)** se encuentra dentro del programa europeo TIDE (Technological Initiative for Disable and Elderly People) y ha sido desarrollado por varias instituciones y empresas alemanas capitaneadas por la cátedra "Chair of Process Control (PRT) de la Universidad de Hagen, junto con el "CALL-Center" de la Universidad de Edimburgo y la "Scuola Superiore S. Anna" de Pisa en Italia [Hoyer et al., 95][Borgolte et al.,98]. El objetivo del proyecto fue el desarrollo de una avanzada silla de ruedas con maniobrabilidad omnidireccional e "inteligencia" en la navegación. Permite el control de la silla a personas con múltiples y severas discapacidades tanto físicas como mentales.

La silla tiene dos módulos especiales:

- Interfaz con el entorno (CEI), donde se pueden conectar una gran cantidad de dispositivos de entrada/salida digitales (múltiples conmutadores) y analógicos (joysticks, ratones, etc)

- Interfaz hombre-máquina (HMI) es un dispositivo de selección de tareas altamente configurable. Los menús consisten en una serie de iconos que son mostrados en una pantalla LCD. El usuario selecciona los iconos mediante un puntero-ratón controlado por el joystick de la silla o en un modo "scan" usando uno o varios interruptores. Cada icono lleva asociada una tarea (guiar la silla manualmente, ajustar la altura del asiento o seleccionar sensores para el guiado asistido) o bien pasar a un submenú.

La silla está dotada con sensores de ultrasonidos e infrarrojos e implementa primitivas locales y globales. El usuario puede elegir tres modos de funcionamiento: no supervisado (no hay interacción), parcialmente supervisado (la silla se para ante los obstáculos pero no cambia su dirección) y totalmente supervisado (se evitan obstáculos mediante el sistema sensorial). Además, añade un módulo de navegación por caminos pregrabados (por ejemplo entrar en el baño). En el modo de grabación, el

usuario habitual o un supervisor más fiable puede grabar el camino. En el modo de ejecución, la silla sigue el camino grabado y para que funcione correctamente debe comenzar en el mismo punto donde la grabación se inició.

En cuanto al interfaz de usuario diferencian entre tres grupos: usuarios que pueden controlar un joystick 3D, otros que pueden operar con uno 2D y personas que no pueden controlar ningún tipo de joystick pero que pueden accionar una serie de conmutadores. Mediante el joystick 3D se puede guiar la silla o bien mover un puntero ratón sobre la pantalla LCD. Los sistemas de control y de asistencia a la comunicación entre el usuario y la máquina pueden ser conectados a través de un bus estándar M3S. El sistema fue testeado por 8 personas comprendidas entre los 8 y los 56 años con diferentes grados de minusvalía obteniendo resultados satisfactorios, si bien fue necesario familiarizarse con el entorno y fundamentalmente con el manejo del joystick 3D.

En cuanto a los trabajos futuros del proyecto se engloban en tres líneas:

- Mejora de ajuste de parámetros: algunos parámetros como la velocidad hacia atrás del móvil se eligió constante; sin embargo, este parámetro puede ser cambiado en función de la minusvalía del usuario.

- Mayor velocidad en la conmutación del joystick: la conmutación entre los dos modos de funcionamiento del joystick requiere tres acciones: bajar, mover y bajar. La conmutación entre los dos modos ocurre muy frecuentemente, por lo que sería mejor realizar la conmutación de modos con un botón aparte.

- Mejora de las operaciones con un conmutador: en la operación de búsqueda y selección, las operaciones que más se ejecutan son las de arranque y parada. Con el sistema actual hay que ir buscándolas cada vez en la secuencia de consignas. Anticipando las nuevas acciones que se estima que el usuario va a realizar y colocando éstas en un menú más pequeño se haría el proceso más rápido.

Otro proyecto europeo dentro del programa TIDE es el **SENARIO**, desarrollado en el "Departament of Cibernetics of The University of Reading (UK)" [Baettie&Bishop,98][Katevas et al., 95].

El proyecto consistía en el desarrollo de una silla de ruedas motorizada para personas con discapacidad física que les permitiera moverse dentro de un entorno conocido (hospital o institución similar) de forma autónoma. El proyecto pretendía aplicar la tecnología desarrollada para el guiado de robots autónomos en el área de la tecnología asistida. Utiliza el bus estándar M3S para comunicar los drivers de entrada de datos con los actuadores.

El interfaz de usuario es un reconocedor de voz que identifica diferentes consignas de movimiento(giro derecha, giro izquierda, adelante, atrás, etc) y localizaciones finales (ir al comedor, ir al jardín, ir al baño, etc). Dispone de dos modos de control de la silla: semiautomático, movimiento controlado por el usuario con prevención de colisión y evitación de obstáculos; y automático, que permite a la silla ejecutar caminos pregrabados mediante una orden de localización final dada por el usuario. En este último modo también está activo el módulo de prevención de colisión y evitación de obstáculos.

Dispone de: un sensor "dead-reckoning" basado en inclinómetros y odómetros; dos sensores láser que rastrean 180º cada uno y "bumpers", para detectar obstáculos y evitarlos; y por último un sistema de balizamiento fijo por radio. Mediante la información odométrica y de marcas recibidas vía radio se realiza la localización absoluta del móvil. Para calcular los caminos a seguir en la evitación de obstáculos utilizan un robusto sistema basado en redes neuronales conocido como "Focused Stochastic Diffusion Network (FSDN)".

El sistema fue testeado por 10 personas durante mayo de 1997, siendo el grado de satisfacción de las mismas muy elevado. Las personas con menor nivel de minusvalía criticaron la falta de control de la velocidad del móvil en el modo semiautomático. El proyecto finalizó en mayo del 97 pero todavía tiene una fuerte demanda. Es por ello que en la actualidad se está realizando una variante sobre el mismo consistente en reemplazar las ruedas frontales pasivas por un sistema de dirección activa que permita controlar de forma independiente la dirección y la tracción.

El proyecto **PAM-AID** está siendo desarrollado por el "Trinity College of Dublin and the National Council for the Blind of Irland" en colaboración con otras Universidades de Suecia, Grecia y Reino Unido. Es un sistema basado en un andador convencional. El robot PAM-AID soporta a la persona que anda detrás y le proporciona ayuda en el movimiento mediante la introducción de dos motores en las ruedas del andador que se manejan desde el manillar.

El sistema dispone de sensores de ultrasonidos e infrarrojos para detectar y evitar obstáculos y permite los siguientes procesos en paralelo:

•*Avisos al usuario*: avisa al usuario de la presencia de obstáculos en las proximidades del robot mediante mensajes acústicos (tonos o frases).

•*Control directo:* permite al usuario tener control directo sobre la dirección del robot mediante la generación de comandos desde el interface de usuario.

•*Evitación de obstáculos:* previene de la colisión y ejecuta un camino alternativo alrededor del objeto usando campos potenciales artificiales.

•*Seguimiento de muros:* encuentra y sigue el muro más cercano al robot.

La planificación de estos procesos es activada mediante un arbitrador consistente en una máquina de estados que determina qué nodo o conjunto de nodos pueden tener control sobre los motores en un momento dado. Las transiciones de los estados se determinan por la prioridad de las tareas junto con la información registrada por los sensores.

El interfaz de usuario posee tres modos de funcionamiento: Joystick, interruptores para los dedos y manillar automatizado. El joystick se usó en los primeros prototipos pero su empleo originaba oscilaciones en el movimiento. Los interruptores activados con los dedos permiten seleccionar la dirección del robot (atrás, adelante) y el giro (derecha, izquierda). Se encuentran situados en el manillar, lo cual es confortable para el usuario. La automatización del manillar consiste en un microinterruptor en cada agarradero del manillar que se activa de forma automática al girar el agarradero unos cinco grados.

El sistema dispone de una realimentación acústica de mensajes de voz. Los mensajes de voz se han usado con éxito en numerosos sistemas de ayuda para ciegos. Esta utilidad se puede activar en la configuración inicial y proporciona información del tipo "obstáculo a la derecha".

Las primeras pruebas del sistema se hicieron en Londres en junio de 1997 con ocho usuarios. Los usuarios prefirieron el manillar automatizado para el control directo pero lo encontraron confuso en el

seguimiento de muros y evitación de obstáculos. Esta reacción fue debida a que, en el modo automático, los agarradores se podían mover pero no generaban comandos válidos ya que el control lo tenía el robot, lo que causaba desconcierto en el usuario hasta tal punto que en este modo preferían el control por interruptores que por manillar.

Como futuros trabajos están introduciendo un "scan laser" y sensores de infrarrojos para mejorar la seguridad del equipo y el reconocimiento de características como puertas, uniones de pasillos, etc. Están investigando la introducción de un sistema que alterne acciones de control directo y automático. Para ampliar los modos de interacción hombre-robot se plantean introducir un reconocedor de voz. Por último, pretenden mejorar la automatización del manillar desarrollando uno más sensible.

Otro proyecto destacable es el **UMIDAM** (Unidad Móvil Inteligente Destinada a la Ayuda a la Movilidad) desarrollado en el Departamento de Electrónica de la Universidad de Alcalá [Mazo et al, 95] con la financiación de la fundación ONCE y cuyo objetivo global era la puesta en funcionamiento de una silla de ruedas capaz de ser guiada mediante órdenes vocales (además de joystick) y que integraba un sistema sensorial de seguridad y ayuda a la navegación en entornos estructurados (hospitales, por ejemplo).

El sistema dispone de tres modos de guiado:

- *Control por joystick*. La silla se controla moviendo el joystick de forma manual.

- *Control por voz.* En este modo, la silla se gobierna mediante la emisión de varios comandos vocales. Así, los sensores externos actúan como sistemas de seguridad ante posibles errores en la emisión de comandos por parte del usuario o ante la presencia de obstáculos inesperados. El beneficiario tiene la posibilidad de activar o desactivar estos sensores.

- *Guiado autónomo*. Este modo permite a la silla seguir muros o bordear obstáculos de forma autónoma. De esta manera, es posible el movimiento a través de largos pasillos, como los existentes en hospitales o centros de rehabilitación, sin necesidad de estar continuamente comandando la silla.

El sistema consta de los siguientes bloques: control de motores a bajo nivel, reconocedor de voz,

sistema sensorial externo (compuesto de sensores de ultrasonidos e infrarrojos), tarjeta de memoria (para grabar los parámetros vocales de los comandos del usuario) e interfaz con el usuario (formado por un display y un teclado). Cada uno de ellos fue implementado en una tarjeta basada en el microcontrolador 8051 de la familia Intel y todas ellas se comunican a través de un bus serie. El sistema es flexible y modular ya que permite poner o quitar tarjetas en función de los requerimientos del usuario.

En cuanto a los sensores de ultrasonidos están formados por ocho unidades distribuidas alrededor de la silla y que permiten tener una buena estimación de las distancias a las que se encuentran los obstáculos en todo el entorno de la misma. También dispone de sensores de infrarrojos para detectar la presencia de discontinuidades pronunciadas en el suelo (por ejemplo escaleras) con suficiente antelación. Para ello se desarrolló un sistema basado en la utilización de un emisor de infrarrojos y un sensor sensible a la posición (PSD), capaz de medir con precisión distancias en un rango aproximado de 0,5 a 6 m. A partir del sistema sensorial desarrollado se diseñaron las estrategias de detección y evitación de obstáculos así como de guiado autónomo.

En cuanto al control por voz, dispone de nueve comandos activados mediante la emisión de nueve palabras diferentes, cuyos patrones deben ser previamente grabados por el usuario en una tarjeta de memoria. Los comandos son: "Parar", "Adelante", "Atrás", "Izquierda", "Derecha", "Mas", "Menos", "Contraseña", "Seguir". Cada comando desencadenaba una acción, así con: "Adelante", "Atrás", "Izquierda", "Derecha" se elige el tipo de movimiento a realizar y con "Mas", "Menos" se acelera o decelera. Con "Parar" la silla se parará, mediante "Contraseña" se inicia un proceso de reconocimiento, parando previamente la silla, y con "Seguir" se conmuta entre un control por voz y un control autónomo.

Dentro del proyecto **INRO** (Intelligent wheelchair) [Schilling et al.,98] se ha desarrollado un prototipo de silla de ruedas capaz de realizar las siguientes tareas: evitar obstáculos; seguir a otra silla que va delante, lo que permite hacer "convoys" de sillas que pueden ser gobernadas por una enfermera que va en la primera silla; repetición de rutas pregrabadas en interiores y exteriores, localización en todo momento de la posición del usuario. La arquitectura del sistema está compuesta de un PC conectado vía serie a los drivers de control de motores, a los sensores externos y al interfaz de usuario (compuesto de un joystick y un LCD). Los sensores de que dispone son: anillo formado por cinco sensores de ultrasonidos utilizados para evitar obstáculos, navegación en interiores y en convoy; sistema de

marcación activo para detectar agujeros y obstáculos cóncavos, formado por un laser que proyecta tres líneas verticales en la dirección de guiado y una cámara CCD para detectar dichas líneas; sistema de posicionamiento global diferencial para la navegación en exteriores. Permite tres modos de funcionamiento: manual (con joystick); seguimiento de la silla que va delante y guiado semiautónomo con trayectorias pregrabadas y evitación de obstáculos, tanto en interiores como en exteriores.

Además de los proyectos comentados existen otros realizados sobre una plataforma base consistente en un robot comercial Robuter^{TM} de la empresa ROBOSOFT sobre el que se ha colocado una silla fija adaptada para minusválidos. Uno de ellos es el **VAHM** ("French acronym for autonomous vehicle for people with disabilities"), desarrollado por el Laboratorio de automática de sistemas de la Universidad de Metz en Francia [Bourhis&Agostini, 98]. Otro es el proyecto desarrollado en el DISAM de la UPM, donde han realizado un prototipo basado en un Robuter^{TM} dentro del proyecto **MobiNet** ("MoBile technology for health care services NETwork") y se ha testeado el comportamiento de la arquitectura **AFREB** ("Adaptive Fusion of Reactive Behaviours") [Matía et al., 98].

Por último tiene especial interés destacar un proyecto de guiado de una silla de ruedas para personas con gran discapacidad mediante movimientos oculares. En dicho proyecto se ha aplicado el sistema **"EagleEyes",** desarrollada en el "Computer Science Departament of Boston College", a una silla de ruedas llamada "**Wheelesley**" diseñada en el MIT [Yanco&Gibs, 97]. El sistema EagleEyes obtiene la posición de los ojos respecto a la cabeza mediante la medida de los potenciales electro-oculográficos (EOG) obtenidos a través de 5 electrodos colocados en la cabeza del usuario. Dispone de dos niveles de control: alto y bajo nivel. Mediante el alto nivel el usuario activa una serie de comandos en función de la posición a la que esté mirando en una pantalla LCD. Estos comandos son del tipo: adelante, atrás, giro derechas, giro izquierdas, etc, y son enviados al bajo nivel para ser ejecutados. En este nivel se pueden realizar correcciones de las trayectorias generadas mediante el alto nivel para evitar obstáculos y pasar por lugares estrechos. Para ello "Wheelesley" dispone de doce sensores de infrarrojos y cuatro de ultrasonidos.

Después de este análisis sobre el estado del arte en este área de la investigación se pueden extraer las siguientes conclusiones:

- Los proyectos de investigación de ayuda a la movilidad de personas discapacitadas han

perdido en los últimos años su carácter marginal y son ya numerosos los estudios realizados al respecto e incluso existen algunos prototipos comercializados. Cabe destacar que es Europa el continente que parece más sensibilizado con este tema, quedando Estados Unidos y Japón en un segundo plano.

•No existe todavía una evaluación a largo plazo de la bondad del funcionamiento de los sistemas descritos en la literatura; en algunos de ellos aparecen pruebas realizadas con un limitado grupo de usuarios pero no se ha realizado un estudio en profundidad sobre el grado de satisfacción de los mismos.

•Los interfaces hombre-máquina son ligeramente abordados en la descripción de los sistemas, estando en la mayoría de los casos muy poco evolucionados lo que implica que no son instrumentos óptimos en la comunicación entre los usuarios y las máquinas. La causa de ello hay que buscarla en el hecho de que las otras partes de los sistemas se han alimentado de un amplio "background" existente en el guiado de robots autónomos mientras que los interfaces hombre-máquina son específicos para este tipo de aplicaciones.

•Los modos de guiado son fundamentalmente dos: *manual*, controlado por joystick , alguna vez por interruptores, en raras ocasiones por voz y excepcionalmente por movimientos oculares; y *semiautomático,* donde se sigue alguna marca en el suelo o se reproducen trayectorias pregrabadas.

2.2.- APLICACIONES DEL SEGUIMIENTO FACIAL Y DE LA DIRECCIÓN DE LA MIRADA

Dado que en la bibliografía consultada no aparece ningún sistema como el que se pretende desarrollar en esta tesis, se va a realizar un estudio de diferentes aplicaciones que utilizan la técnica propuesta en la misma. En los últimos años hemos asistido a un gran avance en los sistemas de comunicación entre el hombre y las máquinas. Principalmente estos avances se han producido en la comunicación desde las máquinas hacia el usuario mediante la introducción de sistemas de presentación gráficos, sistemas con ventanas y sintetizadores de voz. Sin embargo, en la comunicación desde el hombre a las máquinas, los avances han sido más discretos ya que, en la actualidad, los sistemas utilizados por la

mayoría de los usuarios para introducir información son los teclados, los ratones o los joysticks, mayoritariamente manejados con la mano. No obstante, actualmente, se están desarrollando en diversos laboratorios de investigación sistemas de comunicación más sofisticados basados en la voz y en información visual que serán introducidos en el mercado en los próximos años. Día a día los métodos de comunicación entre las personas se van trasladando al ambito de las máquinas, si bien la complejidad de los mecanismos que intervienen hacen que éstos se vayan introduciendo poco a poco, siendo en la actualidad un tema candente de investigación.

La investigación en sistemas reconocedores de voz ha sufrido un gran avance en los últimos años, existiendo en nuestros días diversos productos comercializados. La comunicación a través de información visual es la que se encuentra menos desarrollada en la actualidad, debido a la gran cantidad de datos que se deben procesar. Una forma de comunicación visual que resulta de gran interés es la detección y seguimiento de la dirección en la cual está mirando. Ésta se puede descomponer en dos partes: por un lado la dirección de la cabeza, y por otro la dirección que forman los ojos con respecto a la cabeza.

Cada persona realiza a diario gran cantidad de movimientos de ojos que le ayudan a realizar diferentes tareas como: leer, escribir, percepción y aprendizaje de cosas nuevas, toma de información del entorno para guiarse, manipulación de objetos, comunicación con otras personas, etc. Normalmente la persona no es consciente del gran esfuerzo que los ojos realizan en el proceso de percepción visual y de la gran cantidad de información que se maneja. Pese a que es una tarea transparente a la persona, no es nada trivial ya que nuestros ojos están continuamente moviéndose (al igual que nuestra cabeza) para obtener la información visual requerida de la mejor forma posible. El sistema visual humano es bidireccional; por un lado, se comporta como un órgano de entrada de datos de la información del entorno hacia el cerebro y, por otro, como un órgano de salida ya que es capaz de apuntar a la dirección de la cual se quiere extraer la información. Esta capacidad que tienen las personas de poder dirigir su mirada a un punto del campo visual puede ser utilizada para comunicarse con las máquinas.

Hace varias décadas que existen sistemas de seguimiento de la mirada de personas. Hasta hace pocos años la inmensa mayoría de estos sistemas se utilizaban en investigaciones de psicología aplicadas al estudio de la percepción y del conocimiento en tareas como conducir o leer textos.
Así, en 1936, Mowrer desarrolló el primer sistema de seguimiento de los ojos de una persona cuando ésta lee [Scott&Findlay, 93].

Inicialmente, estas técnicas se usaban sólo en experimentos de laboratorio, los equipos eran voluminosos y caros y requerían que la cabeza del usuario permaneciera fija, para lo cual se construían unos soportes metálicos donde se encajaba la cabeza. Estas técnicas eran intrusivas (empleo de lentes de contacto con espejos planos) y no podían ser usadas en grandes períodos de tiempo ni fuera del laboratorio. Los datos eran registrados durante el experimento y se analizaban off-line una vez que había acabado. En nuestros días existen sistemas comercializados que se aplican en investigaciones en psicología y percepción empleando un método no intrusivo como es la video-oculografía (ver el punto siguiente) [User's Manual of SMI, 1995].

En los últimos años, se han multiplicado las aplicaciones que utilizan técnicas de seguimiento de la mirada, dejando el ámbito del laboratorio para desarrollar sistemas comerciales para el gran público. Entre ellos podemos destacar:

•*Sistema de seguimiento de misiles para pilotos* [Smyth et al, 94]. Es un display integrado en el casco del piloto donde se muestra un punto de mira que es gobernado por los ojos, lo que le permite tener las manos libres y poder utilizarlas para otras tareas. El piloto utiliza sus ojos como un manipulador extra que activa cuando cree oportuno y el cálculo de la posición se realiza on-line con gran precisión.

•*Enfoque automático*. La cámara Canon EOS 5 fue la primera en introducir el enfoque automático mediante la mirada. Ésta es calculada mientras el usuario mira por el visor, y puede seleccionar cinco zonas de enfoque en la imagen [Canon, 95]. El propio usuario puede calibrar la cámara mirando a unas posiciones fijas del visor. La Canon UC-X1 Hi fue la primera videocámara controlada por la mirada. El sistema de autoenfoque está basado en el usado por la Canon EOS 5, aunque no está limitado a cinco puntos de enfoque sino que enfoca el objeto que se mira. La cámara usa cuatro fuentes de luz infrarroja, dos se utilizan para personas con gafas y están situados un poco más lejos que los otros dos (la cámara detecta si el usuario lleva o no lleva gafas). La calibración se hace mirando a dos puntos en el display del visor . En el propio manual de usuario se cita que el sistema de seguimiento puede fallar de vez en cuando, lo que obliga al usuario a realizar rápidos movimientos con sus ojos para volver a enganchar el sistema. En la pantalla del ocular aparece continuamente un cuadrado blanco donde el usuario está mirando, esto es muy molesto y suele despistarle.

•*Activación selectiva de planos en teleconferencias con varias cámaras* [De Silva et al., 95]. Utiliza un sistema llamado MPEC ("multiple person eye contact") que se emplea en teleconferencias y que permite seguir con una cámara a la persona con la que se desea hablar simplemente mirándola en un plano general. Con la mirada se controla la elevación, el azimut y el zoom de la cámara que está monitorizando al interlocutor del hablante.

•*Detección e identificación de caras en sistemas de monitorización y seguridad*. Se distinguen dos aplicaciones: las que tienen como objetivo la detección de caras humanas sin reconocimiento de las mismas [Rowley et al., 98][Sung&Poggio, 98]. En este caso lo que se busca es detectar la presencia de una cara o bien contar el número de caras que pueden aparecer en una imagen compleja con multitud de usuarios. Por otro lado, están las aplicaciones destinadas a reconocer caras [Zhang et al.,97][S-H Lin et al.,97][Lawrence at et, 97], En este otro caso lo que se busca es identificar si el rostro a analizar se corresponde con alguno de los almacenados en una base de datos.

•*Desplazamiento del entorno en sistemas de realidad virtual (RV).* En [Stiefelhagen et al., 97] se presenta el control de un visualizador de imagen panorámico. El sistema permite hacer el control de "scroll" sobre una imagen panorámica en 360 º mediante un seguidor de mirada y permite actuar sobre el "zoom" mediante un reconocedor de voz. La empresa alemana SMI ("Senso Motoric Instruments") comercializa, desde 1995, un sistema de seguimiento ocular sobre unas gafas de RV llamado H.M.D. ("Head Mounted Display"). El sistema permite moverse sobre un mundo virtual y realizar teleoperaciones [User´s Manual of SMI, 95].

•*Lectores de labios*. Cada vez son más las aplicaciones que, además de calcular la dirección de la mirada, realizan el seguimiento de la forma de características faciales (boca, cejas, ojos, etc) con el fin de hacer reconocedores gesticulares destinados a desarrollar lectores de labios que puedan complementar a los reconocedores de voz [Meier et al., 97].

•*Ayuda a personas minusválidas*. En [Bergasa et al., 96] se desarrolló un sistema para mover un robot mediante los movimientos oculares para ser utilizado por personas con grandes minusvalías que no pudieran hacerlo por medios más convencionales. En [Heinzmann&Zelinsky, 97] están trabajando en un reconocedor de gestos realizados con la cabeza que permita la generación de consignas para favorecer la comunicación con un robot.

Mediante la técnica "EagleEyes" se puede posicionar un cursor mediante los movimientos oculares [Yanco&Gibs, 97], como ya se explicó en el punto anterior. [Chapman, 91] desarrolló una aplicación que permitía controlar varios sistemas sobre una pantalla como: procesador de textos, llamadas telefónicas, control de TV, etc. Todas estas opciones eran activadas manteniendo la mirada sobre ellas durante un período de latencia. [Frey et al, 92] desarrolló una máquina de escribir controlada por la mirada llamada "Erica". Para hacer más fiable y manejable el sistema sustituyó el teclado por seis grandes teclas a lo que añadió un acceso a frases comunes y un algoritmo de predicción de cadenas basado en los modelos de Markov lo que hacía que el teclado virtual cambiara dinámicamente decrementando el tiempo de escritura en un 25 %. En [Clarke et al.,98] se presenta un sistema de posicionamiento de un cursor en una pantalla mediante la dirección de la mirada.

2.3.-TÉCNICAS EMPLEADAS EN EL SEGUIMIENTO DE LA DIRECCIÓN DE LA MIRADA

En el punto anterior se vio cómo el seguimiento facial, o más concretamente el de la mirada, tiene múltiples aplicaciones entre las que se encuentran algunas de ayuda a personas discapacitadas. En este punto se va a realizar un estudio de las diferentes técnicas empleadas para este fin y que, según [Glenstrup&Engell-Nielsen,95], se pueden dividir en cuatro grandes grupos:

2.3.1.- Electrooculografía (EOG)

Consiste en medir el dipolo característico que aparece en el globo ocular ya que la retina es electronegativa con respecto a la córnea. La medición de este dipolo se hace detectando pequeñas diferencias de potencial con unos electrodos adheridos a la piel alrededor de los ojos. Este método no es suficientemente sensible como para medir pequeños movimientos de los ojos (precisión de 5º) y no puede ser utilizado para determinar movimientos verticales puesto que los músculos de los párpados introducen corrientes eléctricas. En horizontal puede medir un amplio rango dinámico de ±70º y no requiere gafas especiales. Necesita de calibración cada vez que se colocan los electrodos y deben estar perfectamente adheridos ya que cualquier movimiento de los mismos introduce errores de cálculo. El sistema es intrusivo ya que el usuario necesita llevar electrodos y cables para registrar las señales, lo que puede resultar molesto.

2.3.2.- Lentes de contacto

Se coloca al usuario unas lentes de contacto especiales que permiten hacer una medida precisa de la dirección de la mirada. Existen dos técnicas: la primera introduce *uno o varios micro espejos en la lente y mide la reflexión de los rayos de luz que inciden sobre ella*, y a partir de esta información calcula la posición de los ojos. La segunda consiste en *incrustar una microbobina en la lente (en algunos animales se les implanta quirúrgicamente en la esclerótica)*. La cabeza del usuario se somete a un campo electromagnético de alta frecuencia, como consecuencia, se crea una señal eléctrica en la bobina que es amplificada y registrada. La señal se descompone en sus componentes vertical y horizontal, reflejando los movimientos en estas direcciones. Si, además, se añade una segunda micro bobina en forma de "8" también se podrán detectar los movimientos de torsión del ojo. Añadiendo una tercera bobina en la frente se pueden medir los movimientos de la cabeza.

Este método se suele emplear dentro de un rango de medida de ± 40º, tanto en horizontal como en vertical, siendo altamente preciso (inferior a 1º) y exacto. Sin embargo es un método muy invasivo y molesto (necesita de una lente especial y un casco para crear el campo electromagnético o para iluminar el ojo con luz infrarroja) y puede ser usado únicamente durante 20 a 30 minutos seguidos. Por otro lado calcula la dirección de los ojos respecto a la cabeza, por lo que no da la dirección absoluta de la mirada (ojos+cabeza).

2.3.3.- Reflejo de infrarrojos (IR)

Se iluminan los ojos con fototransistores de IR y se detecta la reflexión de la luz sobre los mismos mediante fotoreceptores. Existen diferentes formas de colocación que dan lugar a distintos tipos de reflexiones.

a) *Reflexión del "limbus"* (borde entre el blanco del ojo (esclerótica) y el iris). Un fototransistor apunta al borde del iris y otro a la esclerótica. Se colocan dos fotoreceptores en estas mismas posiciones, de forma que sus salidas son amplificadas diferencialmente y el resultado de esta amplificación ajustado a cero cuando el ojo se encuentre en la posición de cero grados. Debido a que la parte superior e inferior del "limbus" suele estar cubierta por los párpados, únicamente se utiliza para seguimiento horizontal.

b) *Reflexión de la pupila.* En este caso hay un fototransistor apuntando a la pupila y fotoreceptores alrededor en forma de círculo. Como la pupila no está cubierta por los párpados es posible un seguimiento en vertical. La zona a seguir es mayor que la anterior pero está menos contrastada.

c) *Reflexión de la córnea y la pupila*. Es igual que el anterior pero tiene en cuenta que, debido a la estructura del ojo, el rayo de luz primeramente se reflejará en la córnea (parte externa) y después en la pupila (para ser más exactos en el cristalino dado que la pupila es un orificio).

d) *Reflexión de las imágenes de Purkinje*. Esta técnica se basa en que los rayos de luz que inciden sobre un ojo pueden tener cuatro diferentes reflexiones como se puede ver en la figura 2.1. La primera se produce en la cara externa de la córnea y se llama *primera imagen de Purkinje*. La segunda se produce en la cara interna de la córnea (entre ésta y el humor acuoso del ojo) y se llama *segunda imagen de Purkinje*. La tercera es la reflexión con la cara exterior al cristalino y se llama *tercera imagen*. Por último, la cuarta es la reflexión producida en la cara interior del cristalino y recibe el nombre de *cuarta imagen de Purkinje*. La técnica "Dual-Purkinje Image" utiliza la primera y la cuarta imagen para calcular la posición. Es más exacta que las anteriores pero tiene la desventaja de que para poder detectar la cuarta imagen es necesario emitir con mayor potencia que en las anteriores técnicas lo que puede ser más peligroso para el usuario.

Este método requiere de una estructura que suele ir montada sobre unas gafas y que puede ser ajustada al usuario mediante tornillos. El método es muy estable, preciso y exacto, es lineal dentro de ±20º en horizontal y ±10º en vertical y con una precisión inferior a 1º. Es no invasivo, ya que no necesita ningún elemento en contacto con el usuario, pero obliga a llevar gafas especiales. Al utilizar luz IR no molesta al usuario. Hay que calibrarlo cada vez que se utiliza y tiene el riesgo de someter al ojo a radiaciones IR que, aunque de baja potencia, pueden dañarlo.

Únicamente calcula la posición de los ojos respecto a la cabeza y, por lo tanto, no da la posición absoluta de la mirada (ojos+cabeza).

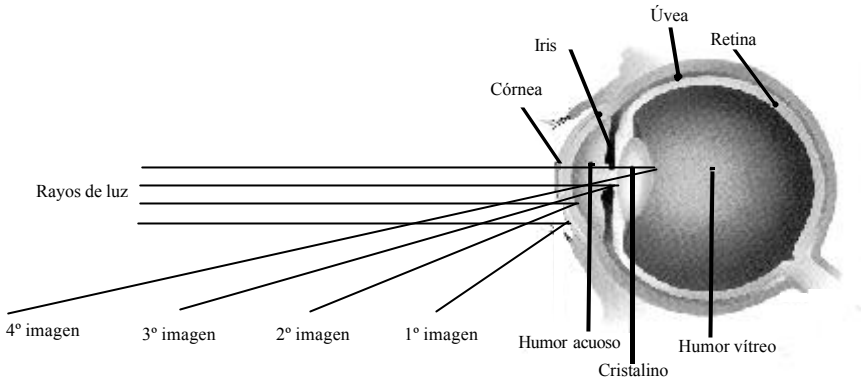


Figura 2.1. Imágenes de Purkinje

2.3.4.- Vídeo-oculografía (VOG)

Consiste en utilizar cámaras de vídeo para calcular la dirección de la mirada. Se coloca una o varias cámaras CCD de reducidas dimensiones que está continuamente enfocando el ojo (o los ojos) sobre el que se desea hacer el seguimiento. La señal de vídeo de la cámara es digitalizada mediante un "frame-grabber" y posteriormente procesada empleando técnicas de visión artificial. En función de la posición de la cámara se distinguen diversas configuraciones:

• *Cámara solidaria con la cabeza*. En estos sistemas la cámara se coloca sobre un casco a una corta distancia del ojo. Para evitar perder el campo visual en el ojo donde está ubicada la cámara se puede colocar un cristal semitransparente que permita ver a través de él y además tomar un primer plano del ojo mediante una cámara situada encima (ver figura 2.2.). En sistemas de realidad virtual (RV), la cámara puede ir colocada en el interior de las gafas utilizando también un cristal semitransparente.

El sistema es muy preciso (precisiones menores a 1º) dentro de un rango de medida medio (±30º en horizontal y ±25 en vertical). El sistema es intrusivo ya que requiere que el usuario se coloque un casco. Suelen requerir iluminación adicional y, salvo en las gafas de RV (en donde la iluminación está controlada), pueden aparecer problemas de reflejos en función de la iluminación ambiente. Es un sistema referenciado a la cabeza y por lo tanto los movimientos

que detecta son de los ojos con respecto a la cabeza, no calcula la dirección absoluta de la mirada. En la figura 2.2. se presenta la estructura de un sistema comercial llamado H.E.D. ("Headband/Helmet-mounted Eye tracking Device") que realiza un seguimiento de un ojo a través de un cristal semitransparente, y otro llamado H.M.D. (ya explicado) desarrollados por la empresa SMI.

• *Cámara sobre soporte fijo.* Se coloca la cámara sobre un soporte fijo a una corta distancia (menor a un metro.) pero sin que moleste a la visibilidad del usuario. En la figura 2.3. se muestra un sistema comercial de este tipo llamado R.E.D. ("Remote Eye tracking Device") desarrollado por la empresa SMI. Se pueden dar diferentes opciones:

a) Se añade una óptica con gran zoom para que obtenga un primer plano del ojo y así poder analizar los movimientos con una resolución similar a la configuración anterior, pero sin ser intrusiva, ya que el usuario no tiene que ponerse ningún elemento adicional. El principal problema de este método es el desencuadre de la escena ya que, como se está capturando un primer plano del ojo, los movimientos de la cabeza pueden hacer que el ojo se salga de la escena. Aunque se reduzca algo el zoom, este sistema requiere que el usuario mantenga la cabeza inmóvil y además para cada uno de ellos hay que ajustar la cámara y calibrar el sistema. Las medidas se hacen con respecto a la cabeza y es muy sensible a la iluminación ambiente debiendo de utilizar algún tipo de iluminación especial (como es el caso del sistema R.E.D. que usa un foco de luz infrarroja).

b) Cámara que captura un plano de la cabeza. En este caso se utilizan algoritmos para calcular la dirección de la mirada de forma absoluta (ojos+cabeza). La posición de los ojos se captura con menor resolución que en los casos anteriores y, por lo tanto, la precisión del sistema es menor. Al tomar una imagen de toda la cara permite utilizar otras variables para la comunicación, como son los gestos. Es no intrusivo y permite movimientos de la cabeza. Es sensible a la iluminación ya que las imágenes de la cara se toman con iluminación ambiente. Puede ser aplicable en casos donde no se requiera gran exactitud y donde la comodidad y la seguridad del usuario sea primordial. Además, puede ser utilizado en sistemas estándar de videoconferencia ya que la posición de la cámara y el plano que toman es similar que el empleado en este caso

[Yang et al.,98].

Figura 2.2. Configuraciones de cámara solidaria con la cabeza

Figura 2.3. Configuraciones de cámara sobre soporte fijo

c) Una cámara toma un plano de la cabeza y otra, dotada de movimiento de elevación, azimut y zoom, toma un primer plano del ojo. Este sistema es una mezcla de los dos anteriores y agrupa las ventajas de ambos eliminando sus problemas. La cámara que toma un plano general de la cabeza calcula la dirección de la misma y la posición de los ojos. Una segunda cámara toma un primer plano del ojo en la posición calculada por la anterior. Para solucionar el problema de desencuadre se dota a esta última cámara de movimiento de forma que continuamente esté siguiendo la posición de un ojo. El único inconveniente es la complejidad del método al tener que usar dos

cámaras, una de ellas situada sobre una torreta en la que hay que controlar tres movimientos en tiempo real. Esta alternativa es aplicable en aquellos casos en los que prima la seguridad y comodidad del usuario y además se busca gran precisión.

d) Sistema esteroscópico con dos cámaras que toman un plano de la cabeza. La idea es utilizar técnicas de esteroscopia para hacer un modelo 3D de una cabeza que permita calcular fácilmente la posición de la mirada. Existen algunos estudios al respecto pero todavía no trabajan en tiempo real. El sistema permite hacer un análisis no sólo de la dirección de la mirada sino también de los gestos de la cara [Essa&Pentland, 97]. Cuando el hardware permita que funcione en tiempo real será, sin duda, el más interesante y cómodo de los analizados.

2.3.5. Estudio comparativo de las técnicas estudiadas

En la tabla 2.1 se presenta una comparativa de los distintos métodos explicados. En el caso de VOG se ha omitido la configuración c), ya que tiene las ventajas de a) y b), y la configuración d) por la imposibilidad de llevarla a la práctica con los medios actuales.

Técnica Parámetro	EOG	Lentes	Infrarrojos	VOG cám.cabeza	VOG cám. fija ojo	VOG cám. fija cabeza
Resolución	Media	Muy alta	Muy alta	Alta	Alta	Media
Rango	Muy alto	Alto	Medio	Alto	Medio	Muy alto
Intrusividad	Electrodos	Lentes y casco	Casco o gafas	Gafas	Nula	Nula
Peligrosidad	Baja	Media	Media	Nula	Nula	Nula
Comodidad	Baja	Baja	Baja	Baja	Media	Muy alta
Libertad de movimientos	Media	Media	Media	Media	Baja	Muy alta
Dirección calculada	Ojos	Ojos	Ojos	Ojos	Ojos	Ojos + cabeza

Tabla 2.1. Estudio comparativo de los diferentes métodos

Se han tomado diferentes parámetros para evaluar los sistemas relativos a la exactitud de sus medidas

y a la comodidad y peligrosidad de su uso. Se ha hecho una clasificación cualitativa ya que dentro de cada método existen diferentes sistemas que pueden dar distintos valores, aunque dentro de unos rangos.

En esta tesis se ha elegido la técnica de video-oculografía en su configuración de cámara fija enfocando a la cabeza y empleando los movimientos de la cabeza, ojos y boca para su aplicación al guiado de una silla de ruedas para minusválidos. Las razones que han llevado a esta decisión han sido:

•*Búsqueda de la máxima seguridad y comodidad del usuario*. Por ello se ha optado por un método no intrusivo, utilizando un sensor pasivo (cámara CCD) que evite cualquier peligro y molestia para el beneficiario. Asimismo, al emplear una microcámara que se coloca en el chasis de la silla a unos 80 cm del usuario, no interfiere en los movimientos ni molesta a la visibilidad, con lo que el usuario se puede comportar como si no llevara implantado este control.

• *Funcionamiento en tiempo real*. Como el objetivo último es el guiado de una silla de ruedas, es necesario trabajar en tiempo real. Por otro lado, con generar un número de consignas finitas limitadas se puede realizar un guiado óptimo, por lo que no es necesaria una gran precisión en las medidas. Trabajando con imágenes a baja resolución se consiguen las dos premisas comentadas.

• *Robustez de la aplicación*. Se ha optado por un control de la velocidad de la silla con los movimientos de la cabeza y de una serie de comandos especiales (adelante/atrás y activo/no activo) mediante el guiño de un ojo y la ocultación de los labios. Se comprobó experimentalmente que los algoritmos de seguimiento de la cabeza eran más robustos que los de seguimiento de ojos, al ser aplicados en entornos con iluminación y fondos cambiantes. Por otro lado, el hecho de utilizar la dirección de la cabeza para activar los comandos da más libertad al usuario, al poder mover sus ojos libremente sin que se active ningún comando, y evita el problema del parpadeo. Para la activación de los comandos especiales se han elegido dos acciones que son fácilmente localizables en imágenes frontales.

•*Utilización de sistemas estándar*. Con el objetivo de abaratar los costes de la aplicación se ha buscado que los elementos fueran lo más estándar posibles. Por ello, se ha tomado la configuración típica de una videoconferencia (campo en gran auge y con gran cantidad de

clientes potenciales), es decir: una cámara de vídeo, un "frame-grabber" y un PC.

2.4.-ESTADO DEL ARTE EN TÉCNICAS DE VISIÓN ARTIFICIAL APLICADAS AL SEGUIMIENTO FACIAL Y CÁLCULO DE LA DIRECCIÓN DE LA MIRADA

La realización del seguimiento facial y de la dirección de la mirada, con gran precisión, mediante técnicas de visión artificial (VA) lleva asociado una complejidad tan elevada que no ha podido ser abordado hasta hace unos pocos años. Ello ha sido debido a la gran cantidad de información a tratar, lo cual implicaba el disponer de un hardware suficientemente rápido que permitiera el poder realizar un procesado en tiempo real.

De la bibliografía consultada sobre el tema propuesto en esta tesis se concluye que existen diversos sistemas de seguimiento de la dirección de la mirada mediante VA, pero muy pocos lo aplican al guiado de robots móviles. Algunas aplicaciones donde se ha empleado esta técnica son: control de "scroll" en realidad virtual; sensación visual mejorada en videoconferencias; extracción de descriptores faciales que permitan disminuir la cantidad de información a transmitir en sistemas de videoconferencias, de forma que con los descriptores se realiza una síntesis de la cara en recepción; control de la posición del cursor en la pantalla con la mirada; detección y seguimiento de personas; interfaz hombre-computadora mediante lenguaje gestual; estudio de los movimientos de la boca como sistema de ayuda a los reconocedores de voz, etc. Asimismo, existen muchos trabajos destinados únicamente a la detección y reconocimiento de rostros, sin realizar seguimiento, con vistas a sistemas de identificación para la policía. Otro área de gran auge en los últimos años ha sido el estudio de sistemas de seguimiento de manos con fines muy variados como: control de un robot mediante apuntamiento con el dedo, control de un CD y una TV con la mano en sistemas de realidad virtual, squash virtual, etc. Evidentemente, estas últimas aplicaciones no emplean la dirección de la mirada pero las técnicas utilizadas pueden ser aplicadas para este fin.

A continuación se van a describir algunos trabajos previos realizados por otros autores que servirán para fijar el estado del arte sobre los sistemas de seguimiento de la dirección de la mirada mediante VA.

El origen de estos sistemas hay que buscarlos en aquéllos que realizaban identificación de

características faciales. En éstos no era imprescindible la velocidad de ejecución ya que iban destinados a reconocer a una persona dentro de una base de datos mediante la obtención de ciertas características faciales. Los primeros trabajos en esta línea realizaban una simple **correlación entre la imagen facial y una plantilla de la característica a identificar** (ojos y boca principalmente) [Ballard& Brown,82]. Este método tenía una serie de problemas: las características faciales no son objetos rígidos, y por lo tanto, varían su forma con relación a la de la plantilla; la característica tiene que tener el mismo tamaño que la plantilla, lo que obliga a tomar imágenes siempre a la misma distancia; una adaptación de la plantilla implica un tiempo de proceso elevado; por último, la iluminación de la escena podía variar y no coincidía con la que había cuando se formó la plantilla. Todos estos problemas hacían que con cierta frecuencia las identificaciones fueran incorrectas.

Para solucionarlos aparece una técnica conocida como **plantillas deformables** [Yuille&Hallinan,92]. Este método se basa en tres elementos básicos:(1) la creación de un modelo parametrizado de la forma de la característica a obtener, (2) la creación de una serie de imágenes transformadas que resalten ciertas características de la imagen a analizar (imagen de bordes, imagen de picos, imagen de valles, etc) y (3) un algoritmo que ajuste la plantilla a la característica mediante la aplicación de una función de energía sobre la imagen original y las imágenes transformadas. Este método es relativamente invariante a las distorsiones geométricas y a las variaciones de la luz, necesita de una plantilla de la forma de la característica a detectar y de su localización aproximada a priori. Se adaptan muy bien a los objetos a identificar pero con un gran tiempo de cómputo. Además de detectar características, suministra una descripción exacta de la forma de éstas, lo que puede ser utilizado en tareas de clasificación y reconocimiento. En [Yuille et al.,92] se describe la aplicación del método de plantillas deformables para la detección de ojos y bocas utilizando plantillas a base de círculos y parábolas. Los parámetros que controlan la forma de la plantilla son el centro y el radio del círculo y los parámetros característicos de las parábolas. Utiliza una función de energía definida a partir de varias imágenes transformadas obtenidas con operadores morfológicos como son: imagen de bordes, imagen de picos (destaca las zonas claras),imagen de valles (destaca las zonas oscuras) y de la propia imagen a nivel de gris. Para ajustar la plantilla busca un mínimo en la función de energía adaptando los parámetros mediante la técnica de descenso por el gradiente. Necesita inicialización manual y el tiempo de ejecución en una WorkStation SUN4 estaba comprendido entre 5 y 10 minutos para detectar un solo ojo. En [Shackleton&Welsh,91] emplea los parámetros de las plantillas ajustadas de ojos y boca como entrada a un clasificador para realizar reconocimiento entre los patrones almacenados en una base de datos. En [Hallinan,1992] plantea algunos problemas del método de Yuille y propone una mejora del

mismo denominándolo **plantillas deformables robustas**. Las imágenes transformadas las obtiene mediante plantillas deformables de baja resolución en lugar de mediante operadores morfológicos, lo que disminuye el tiempo de cómputo. Robustece las plantillas sustituyendo las líneas por áreas y hace una inicialización automática de la plantilla. No obstante el tiempo de cómputo sigue siendo de varios minutos para cada característica. En [Yuille&Hallinan,92] se hace un estudio del empleo de las plantillas deformables para hacer seguimiento, concluyendo que funcionan perfectamente cuando la deformación entre una imagen y la siguiente es muy baja, lo que supone tiempos de proceso máximos de 1 sg., con movimientos lentos, muy por debajo de los obtenidos por el autor. Plantea, asimismo, la posibilidad de relacionar diferentes plantillas (ojo derecho, ojo izquierdo y boca), con lo que se robustece la detección y permite calcular otros parámetros como, por ejemplo, la dirección de la mirada de una persona.

Trabajos más recientes han mejorado la técnica pero todavía no es aplicable en tiempo real. Así en [Jain et al., 96] permite una definición de la plantilla dibujada a mano en formato "bitmap". El modelo se utiliza como una probabilidad a priori en un modelo bayesiano y la función de energía se interpreta como una función de densidad de probabilidad que se ajusta únicamente con una imagen de bordes. El objeto se localiza maximizando la probabilidad a posteriori. El método es general y puede ser usado para formas arbitrarias. A pesar de trabajar sólo con la imagen de bordes el tiempo de proceso en la localización de una mano es de 5 sg. en una Sun Sparc 20.

En [Dubuisson et al., 96] se emplea un método similar al anterior para diferenciar entre 5 tipos de vehículos en imágenes de coches circulando por una calle. El acierto del método es del 92% para una base de datos de 405 imágenes, aunque no trabaja en tiempo real.

En [Bergasa et al., 97] se implementa un método de plantillas deformables para detectar los ojos de una cara. La localización inicial es automática, se trabaja con las imágenes transformadas (bordes, picos y valles) y se reduce las etapas de ajuste de la función de energía. Sobre una base de datos de 50 personas se obtuvo un acierto de localización del 92% con un tiempo de cómputo de 25 sg. en un Pentium 100 Mhz. En el Anexo 3 se presenta el trabajo realizado por el autor empleando este método. Esta técnica no fué utilizada debido al excesivo tiempo de cálculo consumido.

En un intento de hacer sistemas más rápidos, **L.C. De Silva** aplica técnicas de **proyección de pixels de borde** para hacer una localización a "grosso modo" de las características faciales, ojos y boca sobre

una imagen, coloca unas ventanas sobre ellas y en el interior de las ventanas de los ojos aplica una plantilla deformable circular [De Silva et al., 95a] [De Silva et al., 95b]. A continuación busca la boca entre las posiciones de los ojos detectados empleando proyección de pixels de borde y, con las tres posiciones, calcula la dirección de la mirada. En las siguientes imágenes realiza un seguimiento de los ojos moviendo las ventanas de localización y buscando la posición de los mismos dentro de ellas, mediante plantillas deformables; y de la de la boca mediante proyección.

El sistema funcionaba con imágenes en blanco y negro (b/n) y hay que asumir una serie de condiciones a priori: 1) el sujeto debe estar frente a un fondo plano y uniforme, 2) el sujeto no debe llevar gafas ni sombrero, 3) la imagen facial inicial tiene que estar inclinada menos de 5º. Utiliza un algoritmo para calcular bordes llamado "edge focusing", que obtiene una imagen de bordes muy buena pero con gran tiempo de cálculo (15 sg.) El algoritmo fue implementado por el autor en esta tesis y comparado con un filtro Sobel, obteniendo unos resultados similares con un tiempo de proceso muy inferior (200 ms en un Pentium 100 Mhz) [Bergasa et al., 98a]. El problema del método, aparte de sus restricciones iniciales, es que es muy sensible a la iluminación, a giros de la cabeza y a la posición del pelo, lo que da lugar a errores en la localización de la plantilla y obliga a ejecutar nuevamente el algoritmo de proyección de pixels de borde. Por otro lado, el tiempo de proceso en una Sun Sparc 20 WS es de unos 20 minutos para la localización de las características en toda la imagen y ocho segundos en el seguimiento. El objetivo de De Silva fue desarrollar un sistema de videoconferencia con varias cámaras que se pudieran ir activando en función de la dirección de la mirada [De Silva et al., 95c]. El autor de esta tesis implementó este método introduciendo algunas modificaciones como: la ya comentada de la obtención de los bordes, una localización inicial en función de la imagen analizada en lugar de con datos estadísticos a priori, el empleo de características geométricas para encontrar los ojos dentro de la ventana mediante características geométricas [Bergasa et al.,98b] y mediante una plantilla del iris posicionada por el método de la roseta [Bergasa et al.,98a]. Los dos métodos se probaron con varias secuencias de imágenes obteniendo mejores resultados con el segundo. El tiempo de cómputo para éste fue de 2 imagenes/sg en un Pentium a 100 Mhz. En el Anexo 4 se describen los métodos empleados y los resultados obtenidos. La velocidad del proceso no era suficiente para realizar una aplicación en tiempo real y además obligaba a tener un fondo uniforme por lo que fué abandonada.

Otro grupo de investigación que ha diseñado un sistema para detectar la dirección de la mirada es el capitaneado por **Alexander Zelinsky, de la Universidad de Camberra en Australia**. Ellos utilizan la técnica más básica, ya comentada, de las correspondencias con una plantilla pero, a diferencia de

Barrard&Brown logran un sistema que funciona en tiempo real (30 Hz) y con una gran cantidad de pequeñas plantillas relacionadas, lo que evita los problemas que aparecían en el método anterior puesto que la variación que se puede producir entre una imagen y la siguiente es despreciable. Emplean un algoritmo en b/n sencillo pero con un hardware específico para realizar seguimiento, el MEP de Fujitsu, que es capaz de seguir más de 100 plantillas de tamaño 16x16 o 8x8, realizando en cada una de ellas 256 correlaciones, en tiempo real [Zelinsky& Heinzmann,96] [Heinzmann&Zelinsky,97]. Para robustecer el seguimiento los datos de las plantillas se fusionan con los obtenidos de un modelo geométrico 2D de la cara. Mediante un filtro de Kalman se estiman las posiciones de las plantillas a partir del modelo, de forma que aquéllas que se alejen de la estimación se les da menor peso en la actualización del modelo. A partir de estos datos se calculan lo que llaman "acciones atómicas" (moverse a la derecha, izquierda, arriba, abajo, lateral) y a partir de éstas es capaz de reconocer hasta doce gestos diferentes (Si, No, Puede Ser, mirada arriba, mirada derecha, etc). Su objetivo es diseñar un sistema en tiempo real que permita controlar la navegación de un robot móvil mediante gestos [Cheng& Zelinsky, 97]. El sistema es un seguidor de la mirada y un reconocedor de gestos. Funciona con fondo complejo, sin iluminación especial y en tiempo real. Los problemas que plantea este método son: las condiciones de iluminación y que la posición no debe variar entre cuando se graban las plantillas y cuando se ejecuta el algoritmo; el maquillaje puede generar reflejos en función de la posición de la cabeza pudiendo ocasionar fallos; el sistema se puede enganchar con el fondo si no existe una cara en la imagen; la potencia se la da el hardware específico que emplea, que no se encontrará disponible comercialmente hasta mediados de 1999.

En la Universidad de Cambridge un grupo de investigación dirigido por **R. Cipolla** lleva varios años investigando en el seguimiento de la dirección de la mirada para su aplicación a un sistema de realidad virtual [Gee&Cipolla, 96] [Yow&Cipolla, 95] [Yow&Cipolla, 96]. Han diseñado un algoritmo que localiza zonas de pixels oscuros de una imagen en entornos claros empleando filtros escalables y orientables basados en la segunda derivada del gaussiano. Agrupa estos puntos en características faciales básicas y éstas en caras candidatas, que serán evaluadas por una red bayesiana que determinará la cara que tiene una mayor probabilidad. El sistema es robusto y funciona para cualquier persona, fondo, iluminación y es capaz de detectar caras para diferentes escalas, orientaciones y puntos de vista. El problema radica en que la cantidad de cómputo es tan elevada que no funciona en tiempo real (del orden de minutos). Curiosamente este autor tiene otras referencias donde analiza la problemática del seguimiento de la mirada pero sin utilizar su método (obviamente por el tiempo de cálculo), sino usando un simple método de detección de pixels oscuros para detectar los ojos, la nariz

y la boca. Así en [Gee&Cipolla, 94a, 94b] plantea un modelo 3D para estimar la dirección de la mirada a partir de las posiciones de ojos, nariz y boca. En [Gee&Cipolla, 95] presenta el algoritmo RANSAC (Random Sample Consensus) como la mejor técnica para estimar las posiciones de las características faciales. Este método funciona a velocidad de vídeo (25 ima/sg.) en una Sun Sparc 10 WS, debe inicializarse manualmente y se probó sobre secuencias de imágenes preparadas.

Otra técnica utilizada es el empleo de **redes neuronales (RN)**. Sin embargo, en la mayoría de los casos se utilizan para identificar caras y hacer reconocimiento pero no para hacer seguimiento. Algunos ejemplos de ello son:[Zhang et al.,97][Rowley et al.,98][Sung&Poggio, 98][S-H Lin et al, 97][Lawrence et al.,97]. Una excepción es el trabajo realizado por Baluja&Pomerleau para hacer un seguidor de la mirada [Baluja&Pomerleau, 94], donde aplicaron una red ALVINN que fue diseñada para conducir el vehículo NAVLAB 1 durante 21,2 millas con velocidades de hasta 55 millas/hora. Hace una localización inicial del ojo derecho, en una imagen b/n, mediante la búsqueda de pixels oscuros rodeados de pixels claros y coloca una retina, en esta posición, de tamaño 15x30. Utiliza un perceptrón multicapa con 16 neuronas ocultas. Las salidas se organizan con 50 neuronas para especificar la coordenada "x" y otras 50 para la "y". Utiliza una representación gaussiana a la salida. Para entrenar la red se utiliza una etapa de calibración donde el usuario tiene que seguir un cursor que aparece en la pantalla durante unos tres minutos. Se toma un conjunto de 2000 posiciones de imágenes y se entrena la red con 250 posiciones de las 2000 capturadas. El sistema trabaja a 15 Hz en una Sun Sparc 10 y da un error máximo de 2º con un movimiento de unos 30 cm. Los problemas del método son: el usuario no puede moverse más que en 30 cm en horizontal; es sensible a los cambios de iluminación; se necesita hacer una ardua calibración; y por último, se calcula la dirección únicamente con la información de un punto, lo que es muy poco robusto.

En la "**Carnegie Mellon University**" se han desarrollado varios trabajos que utilizan seguimiento de la mirada trabajando en imágenes en color. Martin Hunke desarrolló un sistema de seguimiento de caras humanas basado en **redes neuronales y segmentación en color** [Hunke, 94]. Primeramente, realizaba una segmentación de la piel de la cara utilizando un histograma en un espacio de color RG normalizado. Para ello calculaba off-line el histograma del color de la piel, fijaba de forma supervisada los niveles de umbralización y con ellos segmentaba las imágenes. Los umbrales los establecía mediante un rectángulo en el espacio RG (Rojo, Verde) normalizado y no se adaptaba correctamente al espacio real de la piel. Por ello, en la segmentación, había colores del fondo de tonos similares a la piel que eran segmentados como piel. Para solucionar este problema introdujo la variable de movimiento. Consideró

que, como quería hacer seguimiento de los objetos segmentados, tomaría únicamente aquellos que se habían movido. Para ello era necesario que el fondo de la escena no cambiara (gran limitación). Mezcla los dos métodos (color+movimiento) y realiza una segmentación más robusta (con las limitaciones dichas). Con ello tiene localizada la cara. Para calcular la dirección de ésta dentro de la zona segmentada aplica la misma estructura de red empleada en el proyecto ALVINN (retina de perceptrón multicapa entrenado por back-propagation).Utiliza dos redes: una para calcular las posiciones "x" e "y" de la cara y otra para el tamaño. Ambas usan las componentes de color normalizadas como entrada de una retina de tamaño 16x16, para la localización de (x,y). Una vez localizada la cara se toma una subimagen de tamaño 12x12 de los pixels del centro de la cara y se introducen a la segunda red para calcular el tamaño. Utiliza un método de generación de imágenes para el entrenamiento de la red basado en tomar varias secuencias de varias personas y a partir de ellas hacer imágenes artificiales cambiando el tamaño, la orientación y el fondo. El sistema trabaja a 15 imágenes/sg en una Sun Sparc 10 WS, da unos aciertos de localización por encima del 90% y unos errores de localización inferiores a cinco pixels. El problema es que las secuencias de test estaban tomadas con luz preparada y a una distancia prácticamente constante. La segmentación en color que usa es muy primitiva y no adaptativa y requiere un fondo uniforme. No utiliza ninguna técnica de predicción ni ningún modelo para el movimiento, lo que hace que no sea capaz de detectar fallos en el seguimiento. En [Hunke&Waibel,94] plantea la aplicación del método para compensar la señal de ruido en micrófonos direccionales, como lector de labios (aunque obviamente empleando más técnicas) y en videoconferencias.

Otro sistema más sofisticado que el anterior que utiliza **segmentación en color** es el descrito en [Stiefelhagen et al., 97a], donde se muestra un sistema que funciona a 15 imágenes/sg. usando una HP9000 WS, un "framegrabber" y una cámara VC-C1 de Canon. Es utilizado para controlar un "scroll" de 360º en imágenes panorámicas en un "Apple's Quick Time Movie Player", empleando comandos de voz para controlar el zoom. Para calcular la dirección de la mirada primeramente calcula la dirección de la cabeza mediante una segmentación en color de la piel empleando un modelo estocástico del color de la piel humana en el espacio RG normalizado. Posteriormente localiza las características faciales: ojos, extremos de la boca y nariz para realizar el seguimiento dentro de la zona segmentada como piel. Para localizar los ojos busca los pixels más oscuros en una ventana en la que se estima que deben estar. La localización de los agujeros de la nariz se hace de la misma forma. Para localizar la boca se utiliza una proyección horizontal de pixels en una ventana calculada a partir de las posiciones anteriores y se busca un mínimo global. Utiliza un modelo 3D de la cabeza a partir de los seis puntos localizados y hace una estimación de los mismos usando una modificación del algoritmo

RANSAC consistente en tomar todos los posibles subconjuntos de puntos para calcular la posición y elegir el mejor. Esta modificación es válida cuando el número de puntos a considerar sea pequeño, como en este caso (6). De forma general, el método establece que se tomen subconjuntos aleatorios de datos hasta que se encuentre uno bueno.

El sistema es robusto a los cambios de iluminación, y a la orientación de la cabeza en la imagen. El problema que tiene es que el modelo de piel utilizado es un modelo calculado a priori para todos los usuarios, lo que puede originar que para algunas personas cuyo color de piel difiera del modelo, el sistema se puede enganchar con otro objeto en lugar de con el de la piel. Por otro lado la localización de ojos, narices y boca puede fallar si se crean sombras que hagan que existan otras zonas del mismo nivel de gris que las anteriores. En [Stiefelhagen et al., 97b] se utiliza una **localización de los ojos basada en un perceptrón multicapa** que tiene como entrada una ventana de (20x10) pixels. Los errores que obtiene son algo menores a los del método anterior, si bien en este caso utiliza un modelo más simple para calcular la dirección de la mirada, realizando únicamente movimientos horizontales y necesitando un proceso de aprendizaje supervisado. Se aplica el sistema descrito primeramente para hacer un seguimiento de los labios con el objeto de poder hacer un lector de los mismos que pueda ayudar a los sistemas reconocedores de voz. Al seguimiento descrito habría que añadir alguna otra técnica, como las plantillas deformables, para obtener parámetros sobre la forma de la boca en cada imagen.

En [Yang et al., 98a] se presenta un sistema que combina los gestos faciales reconocedores de voz para aumentar sus prestaciones. Yang y Waibel añadieron al sistema de seguimiento descrito una cámara con control de elevación, azimut y zoom [Yang et al., 98b] para hacer seguimiento de caras. Utilizaron tres modelos: el primero, un modelo estocástico para caracterizar el color de la piel de las caras humanas. El segundo, un modelo de movimiento para estimar la ventana de búsqueda de la cara dentro de la imagen. El tercero, un modelo de la cámara usado para predecir y compensar los movimientos de ésta. Lograron incrementar la velocidad del algoritmo a 30 imágenes/sg. con el mismo hardware.

La segmentación en color ha sido utilizada en los últimos años por gran cantidad de autores. Destaquemos por ejemplo el **proyecto "See-Eagle**" [Littmann&Ritter, 97], donde una persona apunta a un objeto en una mesa. El objeto es reconocido calculando el punto de corte entre la línea que define la dirección apuntada y el plano de la mesa. La posición del objeto se utiliza para guiar un brazo robot

a esta posición para recogerlo. El sistema permite calcular la dirección apuntada con una exactitud de 1±0,4 cm en un área de 50x50 cm. Realiza una segmentación en un espacio RGB empleando una red neuronal LLM (Local Linear Maps) cuya estructura es similar a una RBF generalizada. La crítica que se puede hacer a este sistema es que es dependiente de la iluminación, no es adaptativo y necesita un proceso de aprendizaje.

Otro grupo de investigación que utiliza segmentación en color es el del **IMAG-GRAVIR de Grenoble (Francia).** Han diseñado un seguidor de la mirada aplicando tres métodos: detección de guiño, matching con un histograma en color normalizado y correlación cruzada (SSD y NCC) [Crowley et al., 95]. Cada método forma un proceso y el sistema utiliza en cada caso el proceso que le de más confianza. El supervisor conmuta a diferentes estados en función de eventos detectados en los procesos. Estos estados son: 1) En la inicialización o cuando la confianza del seguimiento sea baja, el supervisor ejecuta una detección de guiño. En este momento se inicializa un modelo en color de la piel y una máscara de correlación para cada ojo conmutando al estado dos. 2) Si la confianza del tracking es alta se sigue usando la correlación. Cuando la confianza de la correlación baja por debajo de 0,5 se conmuta al tercer estado. 3) Re-inicialización, la cara se sigue usando el modelo de color calculado hasta que no se detecte un guiño que permita inicializarlo de nuevo. Cuando éste se produzca también se actualiza la máscara de correlación y el supervisor conmuta al estado dos. El resultado de la detección de la cara es introducido en un estimador recursivo (filtro de Kalman). El sistema trabaja a 20 imágenes/sg. en una WS a 150 Mhz. y se ha aplicado al control de elevación/azimut/zoom de una cámara. Los problemas de este método son: se hace totalmente dependiente del guiño, de forma que si está perdido y no se da el guiño el sistema no tiene capacidad de recuperación. El modelo de color se actualiza cada vez que hay un guiño pero no es adaptativo. Para el cálculo de la dirección no se utiliza un modelo facial 3D sino la posición de dos puntos con lo que la predicción no es muy robusta. En [Crowley&Coutaz, 95] se plantea que mediante el seguimiento de la dirección de la mirada y el control activo de los parámetros de una cámara se pueden reducir grandemente el ancho de banda necesario en las comunicaciones por videoconferencia.

En **"Departament of Electronics in University College of Dublin**" han desarrollado un sistema llamado AAC ("Augmentative and Alternative Communication") [Clarke et al.,98] consistente en controlar la posición del cursor en la pantalla mediante la mirada. Utiliza una cámara de vídeo estándar colocada encima del monitor y una tarjeta digitalizadora Creative Labs Video Blaster RT-300. Emplea un modelo en color de la piel Gaussiano en un espacio RGB para localizar la cara. Para poder

generalizarlo a cualquier usuario, inicialmente detecta un guiño de un ojo y a partir de esta posición define una ventana para tomar una muestra de pixels con el fin de inicializar el modelo. Una vez que ha localizado la cara hace una búsqueda de los ojos y boca en dos zonas restringidas, dentro del objeto cara, y empleando correspondencias. Realiza un filtrado de las coordenadas, para evitar falsas detecciones, y transforma las posiciones de las características filtradas en la posición 2D del ratón. El sistema funciona a 25 imágenes/sg. en una WS a 150 Mhz. Las objeciones que se le pueden hacer a este trabajo son: emplea un método de inicialización del modelo del color de la piel que requiere guiñar un ojo y utilizar una ventana a priori para tomar la muestra, lo que puede ocasionar que se evalúen pixels de "no piel"; emplea un modelo no adaptativo; la localización de ojos y boca pierde eficacia cuando la cabeza está girada más de 45º en cualquier dirección; y por último, no utiliza un modelo facial 3D sino que calcula las coordenadas del ratón mediante una conversión de las posiciones filtradas de ojos y boca.

2.5. CONCLUSIONES

Como conclusiones cabría decir que se ha hecho una revisión de los principales proyectos de ayuda a la movilidad existentes hoy en día y que en ninguno de ellos se utiliza un guiado mediante seguimiento facial aplicando técnicas de visión artificial. No obstante, se han presentado diversas aplicaciones en las que se emplea seguimiento facial y de la dirección de la mirada, planteando diversas técnicas. Se ha realizado un estudio comparativo de las distintas técnicas y se ha razonado el empleo de la visión artificial para los objetivos de esta tesis. Por último, se ha analizado el estado del arte de la visión artificial aplicada al seguimiento facial, concluyendo que los sistemas que trabajan en tiempo real, de forma robusta y que permiten una cierta movilidad del usuario, son los basados en correspondencias y en análisis en color, necesitando de un proceso de inicialización de forma supervisada.

3. VISIÓN GENERAL DEL SISTEMA

3.1.- ARQUITECTURA DEL SISTEMA SIAMO

Previamente a la descripción de los objetivos planteados en esta tesis y del esquema general de la misma se va a realizar una descripción de la arquitectura del proyecto SIAMO. Con ello se pretende ubicar el sistema de guiado, mediante seguimiento facial, dentro del proyecto.

El objetivo del proyecto SIAMO ha consistido en diseñar una silla de ruedas motorizada (ver figura 3.1.), que contemplara un sistema de guiado básico, mediante joystick, y sistemas de guiados alternativos para casos de personas con severa discapacidad, como: guiado por voz, por expulsión de aire, por movimientos de cabeza y por movimientos oculares. Además se le ha dotado de una completa interfaz usuario-máquina, un módulo de control de motores y de guiado autónomo (paso a través de puertas, seguimientos de paredes, ejecución de caminos pregrabados, etc) y sistemas de seguridad.

La arquitectura del sistema SIAMO [Mazo et al., 98] ha sido diseñada para ser completamente versátil. Permite la incorporación o eliminación de servicios simplemente añadiendo o quitando los módulos involucrados en cada tarea. Los módulos que forman el sistema se pueden ver en la figura 3.2. y son: control de bajo nivel, interface usuario-máquina, seguridad y detección del entorno y navegación e integración multisensorial.

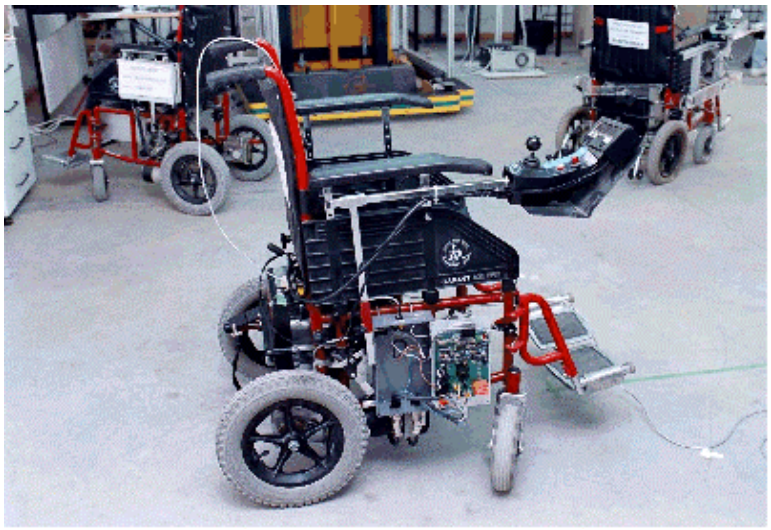


Figura 3.1. Aspecto del SIAMO

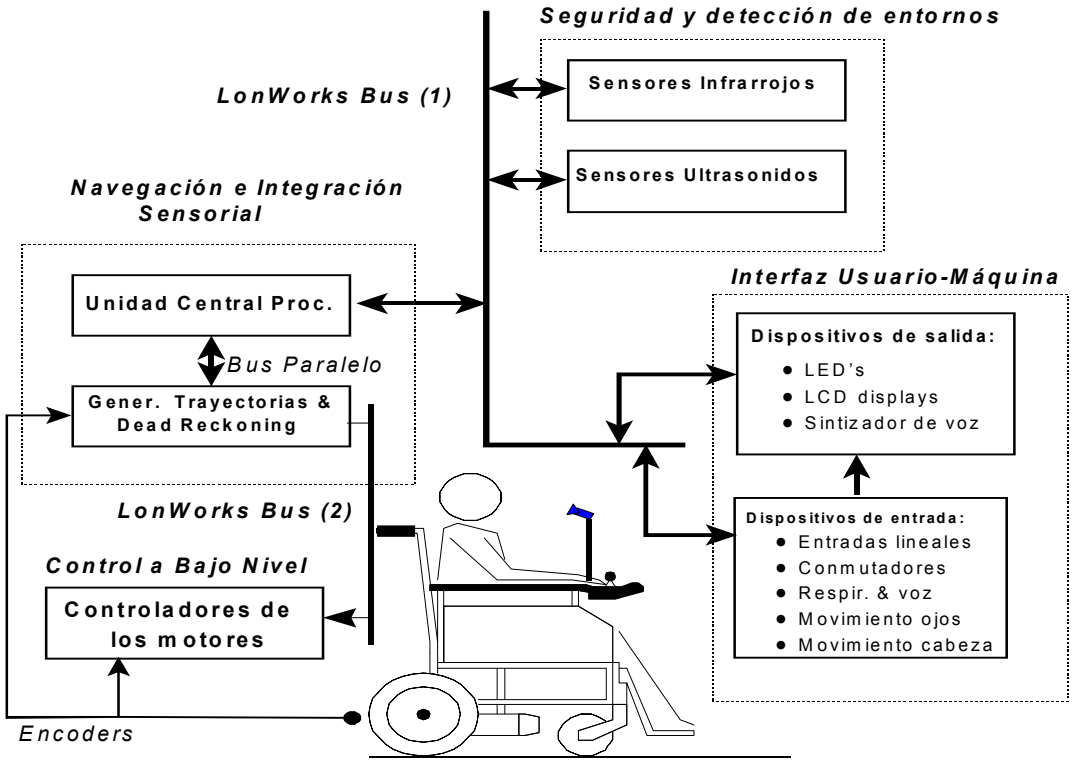


Figura 3.2. Diagrama de bloques del proyecto SIAMO

Cada módulo está formado a su vez por varios subsistemas, algunos implementan la función básica del mismo y otros son opcionales y permiten extenderlos, adaptarlos o cambiarlos. El interfaz usuario-máquina está formado por un display para mostrar el estado del sistema, seleccionar el modo de control (joystick, reconocedor de palabras, por expulsión de aire, etc) y de operación (control directo, semiautomático y automático). El tipo y número de módulos a introducir en el sistema es programable según las necesidades del usuario. En su configuración básica el proyecto SIAMO necesita: el módulo de control de bajo nivel, el más simple interfaz usuario-máquina y un joystick lineal para trabajar, como una silla de ruedas motorizada convencional.

La arquitectura general responde a un sistema de control distribuído. Los módulos han sido diseñados como entidades independientes, con suficiente inteligencia como para tomar decisiones y dar información de su estado intercambiando una limitada cantidad de información entre ellos. La información internodal está formada por tres buses: 1) Intercambia información entre los sensores y el interfaz hombre-máquina con el sistema de navegación; 2) para comandar los motores; 3) para comunicar la unidad central con el generador de caminos y el "dead-reckoning". Buscando la mayor estandarización de acceso a los buses de comunicación, se ha utilizado una red LonWorks para implementar los buses 1) y 2). Otra característica de este bus es que permite hacer nodos inteligentes facilitando y simplificando la interrelación entre la silla de ruedas y el entorno. Para la implementación de cada uno de los subsistemas se han empleado tarjetas de procesamiento basadas en DSPs con suficiente flexibilidad para adaptarlas a las necesidades de cada nodo.

El sistema SIAMO incorpora tres modos de funcionamiento: control directo, semiautomático y automático. En el modo directo la silla está bajo el control absoluto del usuario, quien genera las consignas de control a alto nivel. Éstas pueden ser obtenidas por varios módulos dependiendo del grado de discapacidad del usuario: joystick 2D lineal, entradas binarias mediante interruptores, reconocedor de voz, soplo, movimientos de la cabeza y movimientos de ojos. En cada modo se pueden activar varios sistemas de seguridad y detección de obstáculos.

En el modo semiautomático una estructura de sensores de infrarrojos y ultrasonidos proporciona facilidades adicionales al usuario en las tareas de bajo nivel, aunque las decisiones de alto nivel las sigue tomando el usuario. Las facilidades son: paso a través de puertas, entrada a ascensores, acercamiento a mesas, seguimiento de pasillos, evitación de obstáculos generando un camino alternativo entorno al mismo.

En el modo automático se seleccionan unos destinos finales por el usuario y se generan rutas para llegar a ellos. Para esto existen dos técnicas: ejecutar caminos pregrabados, lo que supone que la silla debe comenzar el camino siempre en el mismo sitio; o ejecutar algoritmos de navegación en función de la información local ofrecida por un "scan laser", por los sensores de ultrasonidos y por la información absoluta dada por un sistema de odometría y GPSs.

La estructura sensorial del proyecto SIAMO está pensada para detectar posibles situaciones de colisión, para lo cual se han integrado sensores de infrarrojos y de ultrasonidos.

a) Sensores de Ultrasonidos. Su principal función es la detección de obstáculos en el entorno de movimiento de la silla. Para ello se han estudiado varias configuraciones. La primera, utiliza ocho transductores de ultrasonidos independientes, cuya información permite adaptar la velocidad de la silla mediante un sistema borroso. Una segunda alternativa utiliza cuatro sensores de ultrasonidos en cada lado de la silla, lo que permite incorporar funciones de seguimiento de muros. Por último, se han utilizado también arrays de ocho sensores en cada esquina de la silla, cuya principal ventaja es una mayor rapidez en la medida de distancias.

b) Sensores de Infrarrojos. Su misión es detectar las irregularidades del suelo y obtener información sobre el entorno (por ejemplo, los límites de las puertas). También en este caso se han desarrollado tres alternativas. La primera, se basa en el uso de emisores IRED y detectores PSD, y permite detectar irregularidades en el terreno por delante de la silla. La segunda, se basa en un emisor láser junto con una cámara CCD y realiza el posicionamiento tridimensional de los obstáculos y la detección de los límites físicos del entorno de movimiento. La tercera alternativa utiliza un registrador láser para medir distancias en un ángulo de 180 grados en pasos de 0,5 grados y facilita el reconocimiento de la presencia de obstáculos u objetos desconocidos dentro del mapa de entorno.

3.2.- OBJETIVOS PLANTEADOS EN LA TESIS

El objetivo fundamental de esta tesis es diseñar el sistema de guiado por movimientos de cabeza, definido dentro del proyecto SIAMO, utilizando visión artificial. Se trata de un modo de operación directo en el que el usuario genera las consignas a alto nivel. Adicionalmente se puede activar un sistema de seguridad y detección de obstáculos basado en sensores de ultrasonidos e

infrarrojos. El sistema debe permitir el guiado del prototipo SIAMO en interiores a una velocidad máxima de 1 m/s. Debe ser no intrusivo para el usuario, fácil de manejar y debe funcionar con cualquier persona, independientemente de su raza y sexo. Pese a la condición de independencia del usuario no hay que olvidar que el proyecto se concibe como un sistema de guiado alternativo para aquellas personas con altos grados de minusvalía que no pueden manejar una silla de ruedas por métodos más convencionales, condición que se tendrá en cuenta en el desarrollo del sistema. Para su buen funcionamiento se impone como condición que el usuario no tenga problemas de movilidad de cabeza y se asume que se requiere un período de entrenamiento.

3.3.- ESQUEMA GENERAL DEL SISTEMA

Para conseguir los objetivos planteados se ha empleando la técnica de visión artificial en una configuración de cámara fija, solidaria a la silla, que captura continuamente un plano de la cabeza del usuario. De esta manera se logra un sistema no intrusivo para éste ya que no es necesario que se coloque ningún casco activo, gafas especiales o cualquier tipo de sensor sobre su cuerpo. Por otro lado, el empleo de esta técnica permite la utilización de componentes estándar empleados masivamente, como son: una cámara de vídeo, un "frame grabber" y un PC, lo que conlleva un abaratamiento de los costes de la aplicación. Asimismo hay que destacar que las técnicas utilizadas pueden ser directamente empleadas en otras aplicaciones como: seguimiento e identificación de personas, videoconferencias, realidad virtual, etc.

Dentro del campo de la visión artificial se ha elegido trabajar con visión en color al proporcionar una mayor información que la obtenida con una cámara en blanco y negro (b/n) y al existir en la actualidad hardware comercial suficientemente rápido como para procesar imágenes en color en tiempo real.

La condición de no intrusión sobre el usuario ha impuesto la colocación de una microcámara, solidaria con el chasis de la silla, a una distancia del mismo suficiente para poder tomar una imagen de su cabeza y que no moleste su visibilidad ni el acceso a la misma. En concreto se ha colocado enfrente del usuario a 80 cm del mismo. Se ha utilizado un CCD de ½" y una óptica de 12,5 mm. Empleando el modelo "pin-hole" de la cámara se obtiene la siguiente ecuación:

$$F_d = \frac{d_{co}}{M + 2 + \frac{1}{M}} \qquad M = \frac{Tamaño\,objeto}{Tamaño\,CCD} \tag{3.1.}$$

donde F_d es la distancia focal de la cámara y d_{oi} es la distancia del objeto a la cámara.

Esto supone la captación de un plano visual de aproximadamente 40x30 cm, como se puede ver en la figura 3.3., suficiente para que la cabeza del usuario quede dentro del plano de análisis para cualquier movimiento que realice.

El análisis de una imagen en color de la cabeza de un usuario (similar a la perspectiva de una foto de carnet), con fondo aleatorio y condiciones cambiantes, limita el tipo de técnicas a utilizar si se quiere conseguir una condición de tiempo real con un hardware convencional. Otro parámetro a tener en cuenta es la resolución de la imagen a tratar, ya que ésta fijará la cantidad de información a analizar por imagen. Siguiendo el criterio de análisis en tiempo real se ha elegido la resolución mínima necesaria para conseguir los objetivos planteados en esta tesis y que resultó ser de 128x128 pixels. A esta resolución, la característica facial más robusta a localizar es la piel al ser el objeto mayor. Esta es la razón por la que se decidió realizar un robusto segmentador de piel que supusiera la base de todo el sistema.

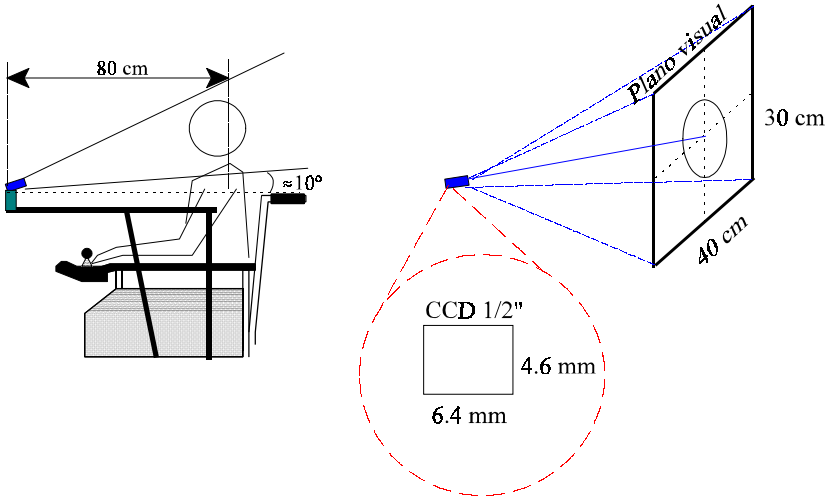


Figura 3.3. Creación del plano visual

Se ha realizado un segmentador del color de la piel basado en un modelo estocástico adaptativo y no supervisado. Es capaz de segmentar la piel de personas de cualquier raza con fondo e iluminación cambiante. Sobre el objeto segmentado piel se aplica un seguimiento mediante estimación por filtro de Kalman. El vector de estado estimado se introduce a una máquina de

estados que genera los comandos de alto nivel de la silla. También existe otro módulo de localización de ojos y boca que servirá para la generación de los comandos especiales:Activo/Inactivo y adelante/atrás, que se suman a los creados mediante el movimiento de la cabeza. Estos comandos se envían a otra máquina de estados que implementa el control a alto nivel y que genera las consignas de velocidad lineal y angular de la silla ($[V_{cmd} \ \ S_{cmd}]^T$). Aplicando el modelo cinemático de la misma, estas velocidades se transforman en velocidades angulares para cada rueda ($[T_{r,cmd} \ T_{l,cmd}]^T$) y se envian a un módulo de control a bajo nivel donde hay implementados dos controles de velocidad en lazo cerrado mediante dos encoders ópticos y dos controladores PI.

En la figura 3.4. se presenta un diagrama de bloques que recoge el funcionamiento general del sistema de guiado propuesto en esta tesis. Como se puede observar, existe una realimentación de los movimientos de la silla mediante los movimientos de la cabeza del usuario. Esto provoca un control en lazo cerrado a alto nivel que se cierra mediante el propio usuario.

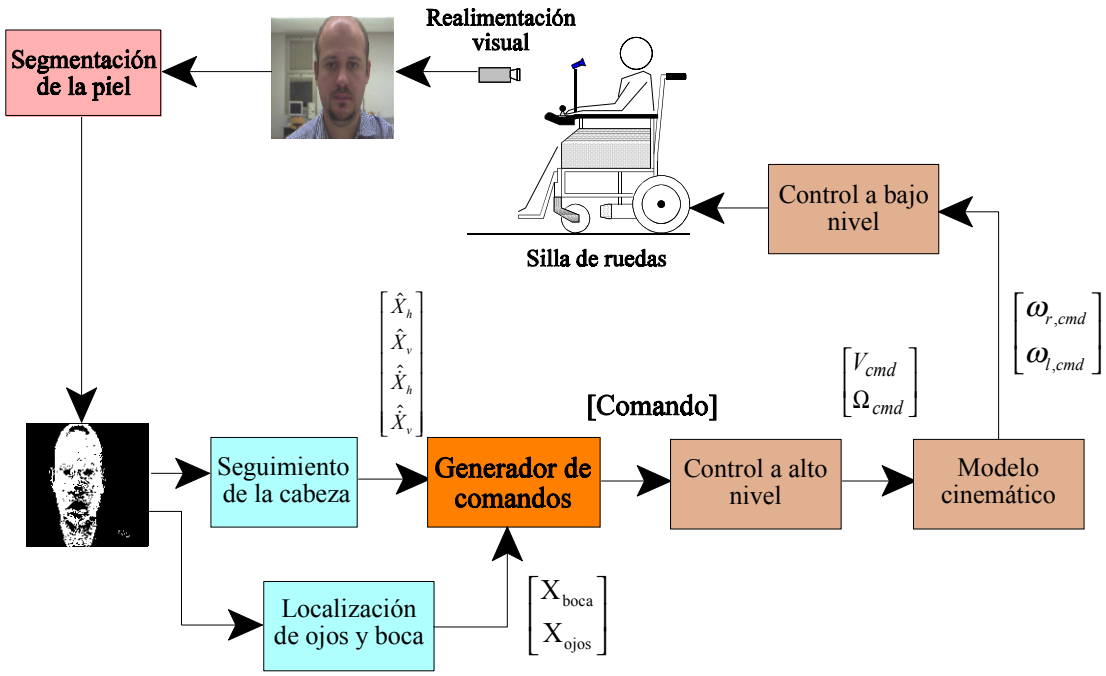


Figura 3.4. Diagrama de bloques del sistema de guiado

Dado que el sistema de guiado propuesto requiere de un período de aprendizaje, se ha diseñado un simulador en tiempo real basado en un entorno virtual 3D de la primera planta de la Escuela Politécnica y en un modelo de la silla. Este sistema permite al usuario simular los movimientos de la silla de una manera segura y real. Además puede probar este tipo de control adaptándose a él mediante un entrenamiento previo. Se ha utilizado el mismo esquema que para el control de la silla real. En este caso, los comandos no son enviados al módulo de control a bajo nivel de la silla sino a otro PC donde se ejecuta el simulador. El sistema de guiado envía las consignas ($[V_{cmd}\ \ S_{cmd}]^T$) al simulador vía RS-232, como se puede ver en la figura 3.5.

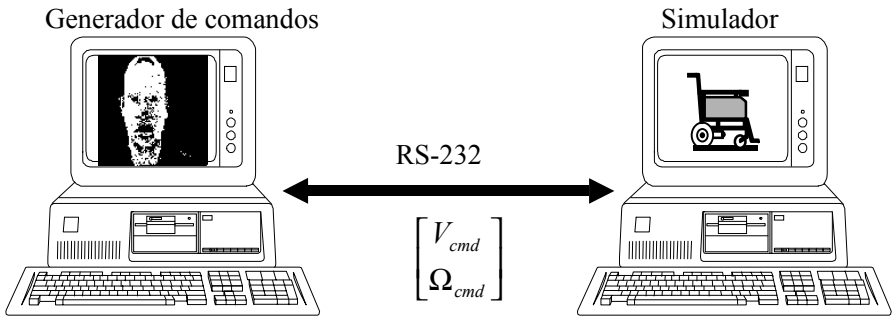


Figura 3.5. Estructura del simulador

3.4.- APORTACIONES DE LA TESIS

Las principales aportaciones que se presentan en esta tesis son:

*** Diseño de un segmentador de piel en color adaptativo y no supervisado en tiempo real basado en un modelo Gaussiano 2D (UASGM[1])**. Este segmentador inicializa un modelo gaussiano 2D mediante un proceso de "clustering" (agrupación de forma no supervisada), basado en aprendizaje competitivo, mediante el algoritmo VQ (Vector Quantization), y soluciona el problema de "validación de cluster" (número de clases existentes en una imagen) empleando una función de coste consistente en una modificación del ratio de Fisher generalizado. De esta forma el sistema inicialmente realiza una búsqueda del color de la piel del usuario y se ajusta a las

[1] A dicho segmentador se le nombrará a partir de ahora mediante las siglas UASGM (Unsupervised and Adaptive Skin Gaussian Model) al ser el nombre que se le ha asignado en las publicaciones internacionales realizadas sobre el mismo.

características de la misma, lo que le da la facultad de segmentar la piel de cualquier persona, incluso de diferentes razas. Este método se ha obtenido partiendo de la teoría general de modelado estocástico mediante mezcla de Gaussianas y asumiendo ciertas hipótesis que simplifican el proceso. De esta forma se consigue un sistema que funciona en tiempo real y que da similares prestaciones que otros más complejos existentes en la literatura [Langan et al., 98] [Roberts et al., 98] que no alcanzan esta condición. La facultad de adaptación al usuario supone una mejora respecto a sistemas similares: como [Yang et al., 98b][Stiefelhagen et al., 97b], donde se emplea un modelo a priori; o como [Crowley&Berard, 96][Clarke et al.,98], donde se emplean técnicas como guiñar un ojo para inicializar el modelo.

*** Estudio de diferentes espacios de color**, a partir de una base de datos de 120 personas de diferentes sexos y razas, con el objetivo de obtener el espacio de color óptimo para realizar un segmentador de piel robusto a cambios de iluminación y diferentes usuarios. Este estudio fue llevado a cabo debido a la no existencia de unanimidad en el empleo de espacios de color por parte de otros autores en trabajos anteriores [Littmann&Ritter, 97][Yang et al.,98a][Moreira&Fontoura, 96][Gong&Sakauchi, 95].

*** Sistema de generación de comandos a alto nivel para controlar los movimientos de una silla, de una manera sencilla y robusta, mediante movimientos de cabeza, ojos y boca.** En la bibliografía consultada únicamente existe un trabajo previo en esta línea, el de A. Zelinsky [Zelinsky&Heinzmann, 96][Heinzmann&Zelinsky, 97] que permite controlar la navegación de un robot móvil mediante doce gestos diferentes en tiempo real. Este investigador, como ya se comentó, emplea la técnica de correspondencias de plantillas con un hardware específico de Fujitsu. No obstante, la introducción de las videoconferencias como un sistema de comunicación de masas ha provocado el desarrollo de los sistemas reconocedores de caras y de gestos, en los últimos años, en un intento de conseguir en un futuro cercano una eficiente codificación y síntesis de gestos humanos y expresiones faciales en la transmisión de datos de vídeo. De esta forma se pretende conseguir una reducción del ancho de banda de transmisión de vídeo de unos 10MHz a unos 10 Kbaudios [Daugman, 97]. Otra aplicación en boga en nuestros días son los lectores de labios como ayuda a los sistemas reconocedores de voz [Meier et al., 99]. Estas técnicas están emergiendo con fuerza, pero todavía no se encuentran en aplicación debido a la complejidad del problema, al intentar dar una solución general para cualquier usuario y cualquier fondo. En esta tesis, en una búsqueda de la solución más simple y robusta para el problema planteado, teniendo en cuenta el "handicap" de la minusvalía, se han codificado 6 comandos diferentes: cuatro mediante los movimientos de la cabeza, empleando un modelo 2D de la misma

y los otros dos escondiendo los labios y guiñando un ojo respectivamente. Para obtener estos últimos comandos se han empleado características de color y de la geometría 2D de las características faciales.

*** Guiado de un prototipo de silla de ruedas real en entornos interiores mediante el sistema de generación de comandos creado.** Se ha aplicado el sistema de generación de comandos al prototipo SIAMO, logrando un óptimo control en el sistema de guiado. El prototipo ha sido probado por una serie de usuarios proporcionándose los resultados de la aceptación del sistema. Asimismo se ha diseñado un simulador de guiado, que permite al usuario entrenarse, utilizando herramientas de representación gráfica en 3D.

4. SEGMENTADOR DE PIEL EN COLOR

4.1.- INTRODUCIÓN

El diseño de un sistema de seguimiento facial en una secuencia de imágenes implica la localización de la cara en cada una de ellas y la estimación de sus parámetros, en la imagen a analizar, a partir de la información calculada en imágenes anteriores. Para localizar el objeto "cara" se utilizan técnicas de segmentación de imágenes. Dichas técnicas asignan cada pixel a una región, dentro de un número finito de regiones "prototipo", caracterizadas por tener propiedades comunes entre los elementos de la región y diferentes con las de otras regiones. A las regiones "prototipo" habitualmente se les denomina *clases o estados*. Dentro de las técnicas de segmentación, suficientemente explicadas en bibliografía especializada en VA, destacan: las orientadas a regiones, uso de umbrales analizando histogramas en b/n o en color, unión de bordes y detección de fronteras, análisis de texturas y uso del movimiento [Niemann, 1990].

Un enfoque que resulta muy interesante en los métodos de segmentación es el empleo de técnicas estadísticas, mediante las cuales las clases son caracterizadas por medidas estadísticas de bajo orden como: media, varianza, correlación de funciones o densidad espectral de potencia. De esta forma, el problema de segmentación de una imagen se convierte en un problema de optimización estadística, lo que produce una mayor precisión en la caracterización de las clases de la imagen.

Las técnicas de segmentación de imágenes basadas en modelos estocásticos pueden ser supervisadas o no supervisadas. En las supervisadas, los parámetros del modelo se obtienen de un conjunto de datos de entrenamiento, mientras que en las no supervisadas los parámetros del modelo se estiman directamente de la imagen a analizar. El diseño de un sistema de segmentación autónomo implica el empleo de técnicas no supervisadas; sin embargo, la baja fiabilidad que ofrecen algunos métodos o la alta complejidad de otros hace que no sean muy utilizados en la práctica, siendo un tema actual de investigación [Langan et al.,98][Roberts,98]. El principal problema de la segmentación no supervisada es el ajuste de un modelo, a priori aleatorio, con la imagen, estando todavía sin resolver hasta la fecha de una manera metódica y generalista.

En esta tesis se presenta un método original para el cálculo de un modelo estocástico adaptativo de las distribuciones de color de la piel sobre un espacio de color 2D normalizado de forma no supervisada.

El empleo de un espacio de color para segmentar la piel de una cara tiene una serie de ventajas. Por un lado, el procesamiento en color es más rápido que usar otras características faciales (texturas, filtros de Gabor, etc) y es más robusto que emplear análisis en b/n o de bordes ya que maneja más información. Bajo ciertas condiciones de luz el color es invariante a la posición y orientación de la cara. Esta propiedad hace que la estimación del movimiento sea más fácil ya que solamente se necesita una traslación del modelo para estimar el movimiento. Sin embargo, "el color no es un fenómeno físico sino una percepción de las características espectrales de una radiación electromagnética en el espectro visible tomada por la retina" [Umbaugh, 98]. El uso del color como una característica para seguir caras humanas presenta varios problemas: 1) el color de la cara obtenido por la cámara está influenciado por factores como luz ambiente, movimiento de objetos, etc; 2) cámaras diferentes producen diferentes colores para una misma persona y bajo las mismas condiciones de iluminación; 3)el color de la piel difiere de una persona a otra. Para poder utilizar el color como una característica para seguir caras será necesario solventar estos problemas.

En esta tesis se hace un estudio de la distribución del color de la piel humana en diferentes espacios de color y se concluye que el mejor para la aplicación es el RG normalizado. Además se demostrará que: el color de la piel humana forma una clase compacta en un espacio de color; las diferencias de color entre varias personas se reduce trabajando con cromaticidades y eliminando la intensidad, y bajo ciertas condiciones de luz, una distribución del color de la piel puede ser modelada mediante una función normal en el espacio de color.

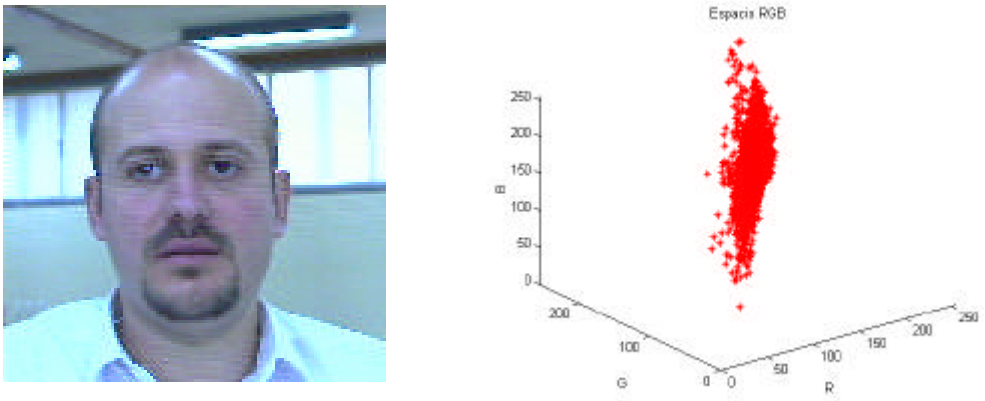
Pese a que las diferencias de color entre diferentes personas se reducen trabajando en el espacio (rg), se demostrará que la segmentación puede dar errores apreciables si se trabaja con un modelo universal a priori. Para solucionarlo se propone hacer un modelo personalizado para cada usuario, obtenido de forma no supervisada y con fondos aleatorios. El problema consiste en inicializar el modelo dentro del histograma. Para ello se propone modelar el histograma de color con K funciones gaussianas. Introducción de una serie de hipótesis sobre el modelo se demostrará que éste puede ser calculado empleando técnicas de "clustering" con aprendizaje competitivo y empleando una función de coste para evaluar el ajuste entre el modelo y la distribución de color. Analizando las clases de la imagen se localizará la clase piel y se calcularán sus estadísticos.

Todo lo dicho da lugar a la creación de un modelo estocástico ajustado al usuario, para caracterizar la distribución de color de la piel humana en un espacio (rg). Para hacerlo más robusto además se le hará adaptativo. Para ello se propone usar una combinación lineal de los parámetros conocidos para predecir los nuevos parámetros del modelo. Esto es posible, ya que una transformación lineal de funciones normales es una función normal, con lo que las distribuciones de color en un momento dado se pueden considerar como distribuciones transformadas de otras anteriores. Se ha usado el método de máxima probabilidad para estimar los coeficientes de la transformación lineal, estudiando dos casos: estimación del vector media sólamente y estimación del vector media y de la matriz de covarianza.

4.2.- DISTRIBUCIÓN DEL COLOR DE LA PIEL

El color es el resultado de la percepción de los rayos de luz de la región del espectro visible (400 nm a 700 nm) que inciden sobre la retina. La retina está formada por dos tipos de células: conos (responsables de la visión en color) y bastoncillos (reponsables de la visión en b/n). A su vez los conos están especializados en tres bandas de color: R-roja (400 a 500 nm), G-verde (500 a 600 nm.) y B-azul (600 a 700 nm). La potencia física (o radiación) se expresa como una distribución espectral de potencia. Un histograma de color es una distribución de colores en el espacio de color y da el número de pixels de la imagen que tienen un color discreto en el espacio [Swain, 90]. Los colores del espacio dependen de la resolución con la que se esté trabajando. Normalmente se trabaja en el espacio RGB, por analogía con el funcionamiento de nuestra retina, y se digitaliza cada canal con ocho bits (esto es debido a que se ha demostrado que el hombre no es capaz de distinguir colores en una resolución mayor). Por lo tanto, para este caso, el histograma en color sería un array 3D de (256x256x256).

Un histograma en color representa la distribución de colores de una imagen de una manera estable, ya que no se ve afectado por oclusión y cambios de vista. Es por ello que se utiliza con gran frecuencia en la segmentación de objetos. Está demostrado que los colores que una persona ve como uniformes no caen aleatoriamente en el histograma sino que se encuentran agrupados. El color de la piel de una persona cumple estas observaciones. En la figura 4.1. se muestra una imagen con la cara de una persona y su histograma de color en un espacio RGB. Obsérvese cómo el color de piel se encuentra agrupado en una pequeña zona del espacio de color.



(a) (b)

Figura 4.1.(a) Imagen facial (b) Histograma del color de la piel

4.2.1. Espacio de color

Diferentes distribuciones espectrales de luz pueden provocar percepciones de color indistinguibles para una persona. Como ya se ha comentado, la retina humana tiene tres tipos diferentes de conos, los cuales responden a la radiación incidente con curvas espectrales diferentes (R,G,B). Basándose en el sistema perceptor humano, tres únicas componentes son necesarias y suficientes para describir un color. Los diferentes colores se forman en función del peso que tienen las tres componentes primarias. Teóricamente las componentes se definen como la integral del producto entre el estímulo luminoso (E(f)) y las tres funciones de color linealmente independientes (r(f),g(f),b(f)), donde f representa la frecuencia del estímulo.

$$R = \int_{f_1}^{f_2} r(f)E(f)df \tag{4.1}$$

$$G = \int_{f_1}^{f_2} g(f)E(f)df \tag{4.2}$$

$$B = \int_{f_1}^{f_2} b(f)E(f)df \tag{4.3}$$

Es sabido que las personas tienen diferentes apariencias de color de piel. Incluso para la misma persona su apariencia es diferente si se viste con diferentes ropas o se encuentra bajo diferentes condiciones de luz. En otras palabras son varios los factores que contribuyen a la apariencia del color de la piel. La percepción de color humana se realiza en un espacio tridimensional RGB. La mayor parte de las cámaras de vídeo trabajan en el espacio RGB o en otro espacio fácilmente convertible a éste. Sin embargo el espacio RGB no es el mejor en todas las aplicaciones. En el problema de localizar "caras" humanas la intensidad no es importante. La apariencia de la piel difiere más en el brillo que en el color mismo, como se ha podido comprobar experimentalmente. Por lo tanto, eliminando la intensidad de la representación del color se reduce grandemente la diferencia de color de piel humana. Es por ello que en muchas aplicaciones el espacio RGB es transformado matemáticamente en un espacio que desacopla la información de brillo de la información de color.

En la literatura existen referencias que utilizan diferentes espacios transformados. En [Littmann&Ritter, 97] dentro del proyecto "See Eagle", se hace un estudio de diferentes espacios de color para segmentar piel y concluye que obtiene mejores resultados en el espacio RGB. Esto es debido a que utiliza 80 imágenes de una misma mano con diferentes puntos de vista y con una iluminación uniforme. Sin embargo si estos parámetros varían los resultados con RGB empeoran considerablemente. [Yang&Waibel,97] utilizan un espacio RG normalizado para realizar segmentación de piel y aseguran que es robusto ante diferentes personas y cambios de iluminación.[Moreira&Fontoura, 96] también utilizan el espacio RG normalizado para segmentar objetos en color. [Gong&Sakauchi,95] utilizan las componentes HS del espacio HSI en segmentación en color.

Debido a la existencia de varios estudios en que se emplean diferentes espacios de color en la segmentación y a que en ninguno se justifica el por qué se ha utilizado un espacio y no otro de una manera convincente, se va a realizar un estudio comparativo de los siguientes espacios transformados: HSI, SCT, RGB normalizado e YQQ.

4.2.2. Estudio comparativo de espacios de color transformados

La ciencia del color y su percepción por el sistema visual humano han sido estudiados ampliamente por "The Commission Internationale l'Eclairage" (CIE). A continuación se describirán los espacios transformados en estudio.

El espacio HSI se obtiene mediante una transformación no lineal del RGB (ver ecuación 4.4) que transforma el cubo de color RGB en triángulos de crominancia para una intensidad dada. La luminancia aparece como un eje perpendicular a los triángulos y, por lo tanto, desacoplada con la crominancia, como se observa en la figura 4.2.

$$\begin{bmatrix} I \\ V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} \frac{1}{3} & \frac{1}{3} & \frac{1}{3} \\ -\frac{1}{\sqrt{6}} & -\frac{1}{\sqrt{6}} & \frac{2}{\sqrt{6}} \\ \frac{1}{\sqrt{6}} & -\frac{2}{\sqrt{6}} & 0 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \tag{4.4}$$

$$H = \tan^{-1}\left(\frac{V_2}{V_1}\right) \tag{4.5}$$

$$S = \left[(V_1)^2 + (V_2)^2\right]^{\frac{1}{2}} \tag{4.6}$$

Está formado por las componentes: *tono* ("Hue") especificado por el ángulo que forma la línea que une el centro del triángulo y el color con la horizontal, representa la longitud de onda del color puro (azul, naranja, etc); *saturación* ("Saturation") o componente radial, indica en qué grado un color puro está diluido con el blanco; *intensidad* ("Intensity") o componente vertical al plano HS, indica el brillo del color.

El espacio SCT ("Spherical Coordinate Transform") es una representación del espacio RGB en coordenadas esféricas (I, ángulo A y ángulo B), como se puede ver en la figura 4.3. Las ecuaciones que relacionan los dos espacios son:

$$I = \sqrt{R^2 + G^2 + B^2} \tag{4.7}$$

$$angA = \cos^{-1}\left[\frac{B}{I}\right] \tag{4.8}$$

$$angB = \cos^{-1}\left[\frac{R}{Isen(angA)}\right] \tag{4.9}$$

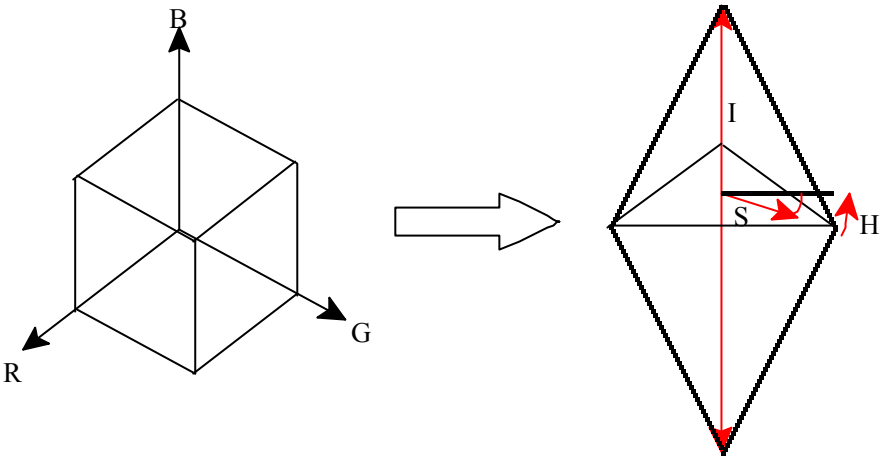


Figura 4.2. Transformación de RGB a HSI

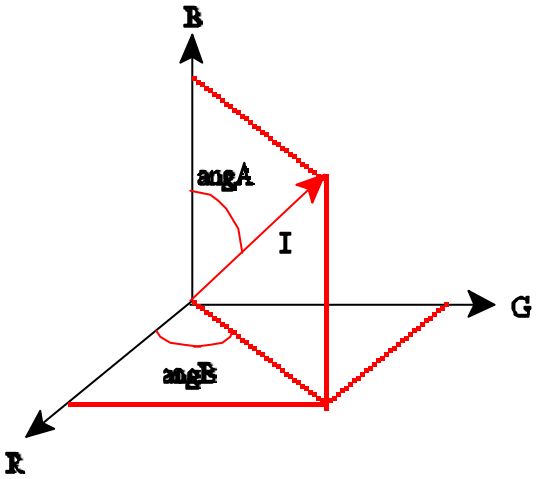


Figura 4.3. Transformación de RGB a SCT

Otro espacio es el RGB normalizado. El CIE ha definido a las componentes de este espacio como cromaticidades y se forman como sigue:

$$r' \frac{R}{R\%G\%B} \qquad g' \frac{G}{R\%G\%B} \qquad b' \frac{B}{R\%G\%B} \tag{4.10}$$

Las ecuaciones normalizan las componentes de color individuales por la suma de las tres, que, como se ha visto, representa la información de intensidad luminosa. Con esta normalización se elimina la dependencia del espacio con la luminancia. En este espacio la luminancia no viene dada como una componente explícita pero será suficiente tomar dos cromaticidades para definir un espacio de color independiente de la luminancia ya que la tercera componente es linealmente dependiente de la suma de las otras dos. Así por ejemplo, si se elije un espacio (rg) se cumplirá:

$$r\%g\%b' 1 \quad 6 \quad b' 1\&(r\%g) \tag{4.11}$$

El último espacio transformado estudiado es el propuesto en [Littmann&Ritter,97], conocido como YQQ, en el que se distinguen las componentes Y, Q-RG y Q-RB que vienen dadas por:

$$Y' \frac{R\%G\%B}{3} \qquad Q\&RG' \frac{R}{(R\%G)} \qquad Q\&RB' \frac{R}{(R\%B)} \tag{4.12}$$

El estudio comparativo entre los espacios se hará tomando dos componentes de cada uno de ellos al ser eliminada la luminancia. En el caso del (rgb), al no venir dada de forma explícita, se tomarán todas las posibles combinaciones entre sus elementos, es decir: rg, rb y gb.

La información de entrada estará formada por una base de datos facial de120 imágenes de rostros de distintas personas, de distinto sexo, edad y raza, tomadas con fondos e iluminación diferentes. Se realizarán tres agrupaciones: raza blanca (formada por 80 imágenes de personas de raza blanca); raza negra (formada por 20 imágenes de personas de raza negra) y raza amarilla (formada por 20 imágenes de asiáticos). Las imágenes fueron seleccionadas de dos fuentes: por un lado, se tomaron aleatoriamente de la base de datos facial disponible en Internet en la dirección http://pics.psych.stir.ac.uk/, que contiene una amplia colección de imágenes para su uso en investigaciones de psicología y visión artificial; por otro, se tomaron al azar imágenes de alumnos y alumnas de la Escuela Politécnica.

En la figura 4.4. se muestran los histogramas de color, para los distintos espacios en estudio, de la imagen de la figura 4.1.(a). Se han clasificado los pixels de la imagen entre *pixels de piel* (en color rojo) y de *no piel* (en color verde). Como se puede observar, en todos ellos la clase piel tiene una forma compacta más o menos alargada y existe solapamiento entre clases, siendo éste más acusado en el HSI. No obstante esta última apreciación hay que tomarla con reservas ya que se trata del análisis de una única imagen y, por lo tanto, puede cambiar para otro usuario u otro fondo.

Para probar la bondad de los espacios se ha diseñado un clasificador bayesiano biclase, para cada espacio de color, que tendrá como entradas un vector de características (**X**) formado por las dos componentes cromáticas de cada espacio. A continuación se ha estudiado el error de clasificación cometido en cada uno de ellos. El empleo de un clasificador estadístico está justificado por su capacidad de mejorar el reconocimiento para clases solapadas.

El teorema de Bayes constituye el soporte matemático sobre el que se apoya este clasificador:

$$p(\omega_i/X) = \frac{p(X/\omega_i)p(\omega_i)}{p(X)} \tag{4.13}$$

donde:

- p(X) opera como un factor de escala ya que aparece en todas las clases
- ω_i representa la clase i-ésima
- X es el vector de características que se pretende clasificar.

En nuestro caso, habrá dos clases (piel, no piel) Para clasificar el vector de características **X** se calcula la probabilidad de que dicho vector pertenezca a cada una de las clases posibles, resultando elegida aquella cuya probabilidad obtenida resulte mayor, prescindiendo del factor de escala. Para ello es necesario calcular la función densidad de probabilidad de las muestras, que se ha supuesto normal y que viene dada por la ecuación 4.14.

$$f(X) = \frac{1}{(2\pi)^{\frac{n}{2}}|C_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(X-m_i)^T C_i^{-1}(X-m_i)} \tag{4.14}$$

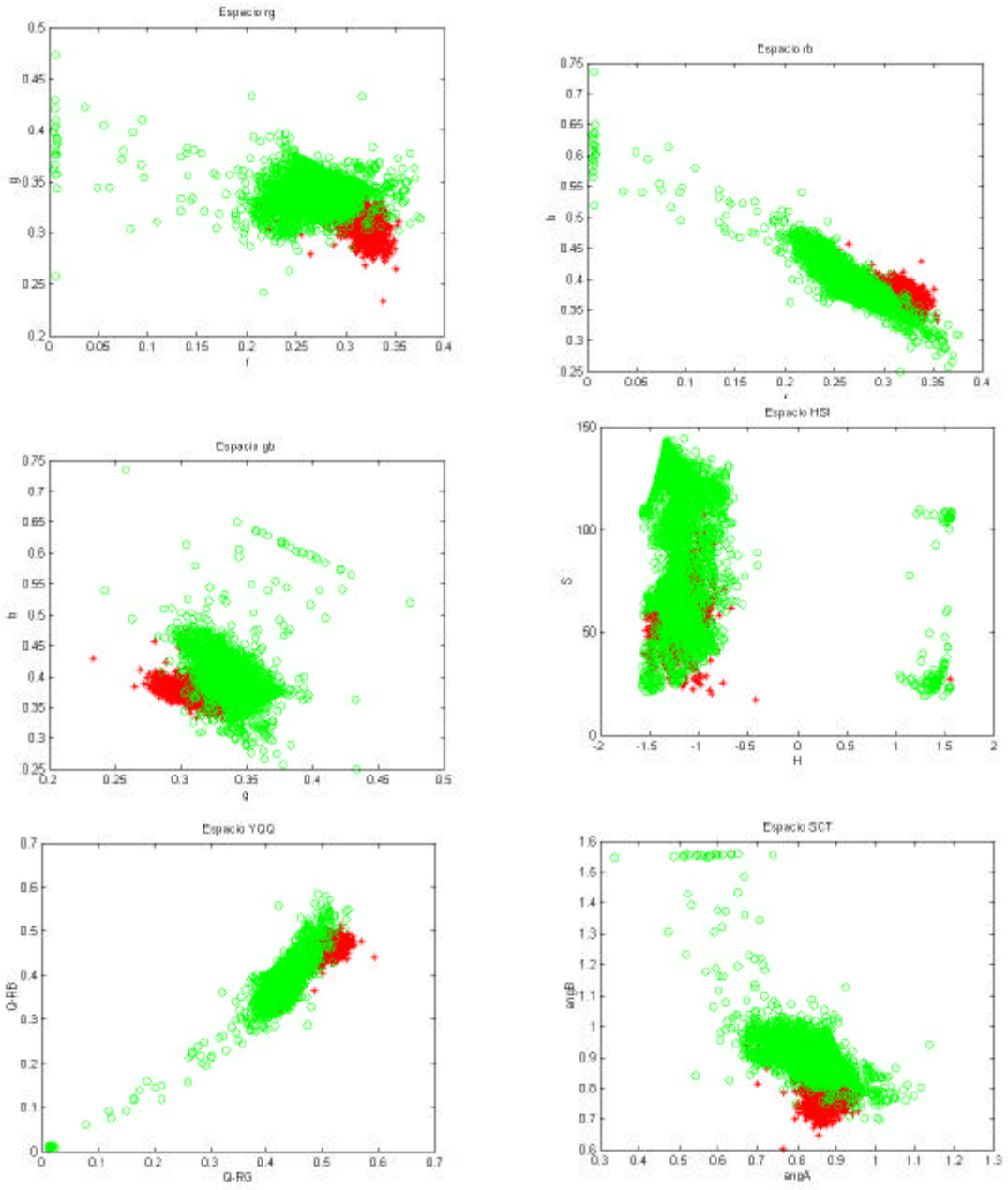


Figura 4.4. Distribución de pixels en los distintos espacios de color para la imagen 4.1

donde C_i es la matriz de covarianza de todas las muestras de una clase dada y m_i es el vector de medias de las mismas. Se trata de un problema de estimación estadística, donde la hipótesis de modelar las variables como normales es aceptada en la práctica totalidad de los casos reales [Rodríguez,97].

No se puede asumir la hipótesis de que las clases sean equiprobables y que las matrices de covarianza

de las muestras que la forman sean iguales, por lo que no se puede hacer ninguna simplificación [Maravall, 93].

Por lo tanto los pasos a seguir para construir el clasificador bayesiano serán:

•*Entrenamiento*. Se calculan las matrices de covarianza (C_i) y media (m_i) para ambas clases de forma supervisada. Se calculan las probabilidades a priori de cada una de las clases, en nuestro caso como la distancia del usuario a la cámara se mantiene constante, experimentalmente se han obtenido los siguientes valores:

$$p(piel) = 0.4 \qquad p(no\ \ piel) = 0.6 \tag{4.15}$$

•*Clasificación*. Se calcula el valor que toma el vector de características de una muestra desconocida sobre las dos funciones densidad de probabilidad y se asigna a la clase cuyo valor sea máximo.

$$\begin{array}{l} p(piel)f_{piel}(X) > p(no\ piel)f_{no\ piel}(X) \quad \mathtt{6} \quad p(piel/X) > p(no\ piel/X) \quad ; \quad X \mathtt{0} Piel \\ p(piel)f_{piel}(X) < p(no\ piel)f_{no\ piel}(X) \quad \mathtt{6} \quad p(piel/X) < p(no\ piel/X) \quad ; \quad X \mathtt{0} No\ piel \end{array} \tag{4.16}$$

En cuanto a la forma de entrenamiento y la medida de error de clasificación existen diferentes estrategias [Rodríguez,97]:

a) *Resustitución*. Emplea el universo de muestras como conjunto de entrenamiento para construir el clasificador y como conjunto de prueba para medir el error. Si el clasificador está bien entrenado los resultados deben ser muy buenos.

b) División del número de muestras en dos conjuntos. Uno se emplea para construir el clasificador y el otro para probarlo. Se repiten aleatoriamente las divisiones y se mide el error como la media de los errores de clasificación.

c) *Uno para probar. Se separa una de las muestras y se construye el clasificador con las restantes*. Para probarlo se emplea la muestra que se separó. Se repite el proceso dejando fuera, sucesivamente, cada una de las muestras de que consta el universo. El error final es la media de los obtenidos en cada iteración.

d) *Se divide el universo de muestras en G grupos con M muestras cada uno*. Se clasifica un grupo en base a un clasificador construído según las muestras del resto de grupos. Se repite el proceso para cada uno de los grupos y se toma la media del error.

e) *Siendo N el número de muestras disponibles, se generan aleatoriamente N números comprendidos entre 1 y N*. Los números que aparecen más de una vez indican las muestras que se emplean para construir el clasificador. El resto se emplean para la fase de pruebas. Se repite el proceso varias veces y se toma la media de los errores de cada ciclo.

En esta tesis se ha calculado el error dividiendo el universo de muestras en dos grupos: uno para entrenar el clasificador y otro para probarlo. Se ha realizado un clasificador para cada raza, el 50% de los *pixels de piel* de cada raza se utilizan para entrenamiento y el otro 50% para prueba. En cuanto a los *pixels de no piel*, se ha realizado un solo grupo con los pixels de fondo de las imágenes de todas las razas, usando un 50 % para entrenar el clasificador y otro 50 % para probarlo.

En las tablas 4.1,4.2, 4.3, 4.4 se indican los valores obtenidos para los estadísticos de la clase piel y no piel en los distintos espacios de color y para las razas blanca, negra y amarilla respectivamente. En todas ellas se ha seguido la siguiente nomenclatura:

$$X' \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} \qquad m_i' \begin{bmatrix} m_{p_1} \\ m_{p_2} \end{bmatrix} \qquad C_i' \begin{bmatrix} F^2_{p_1} & F^2_{p_1 p_2} \\ F^2_{p_2 p_1} & F^2_{p_2} \end{bmatrix} \tag{4.17}$$

Como se puede apreciar, los valores de covarianza de la clase no piel son muy superiores a los de la clase piel, debido a que esta última representa un único color, mientras que la primera está formada por todos los fondos aleatorios de las imágenes. Por otro lado, las covarianzas de la clase piel son relativamente pequeñas si se tiene en cuenta que se han obtenido con diferentes usuarios y con diferente iluminación, lo que demuestra que los espacios analizados son muy invariantes a las personas de una misma raza y a los cambios de iluminación. Además los estadísticos entre las diferentes razas son muy parecidos, con una diferencia máxima de 5 centésimas.

Clase Piel - Raza blanca Espacio color \Estadísticos	m_{p1}	m_{p2}	F_{p1}^2	F_{p2}^2	$F_{p1p2}^2 = F_{p2p1}^2$
p_1=r ,p_2=g	0,3048	0,3248	$0,2246*10^{-3}$	$0,2850*10^{-3}$	$-0,0465*10^{-3}$
p_1=r, p_2=b	0,3048	0,3704	$0,2246*10^{-3}$	$0,4165*10^{-3}$	$-0,1781*10^{-3}$
p_1=g, p_2=b	0,3248	0,3704	$0,2850*10^{-3}$	$0,4165*10^{-3}$	$-0,2385*10^{-3}$
p_1=H, p_2=S	-1,1495	65,3854	0,2736	231,0846	0,0652
p_1=angA, p_2=angB	0,8776	0,8171	0,0018	0,0015	0,0002
p_1=Q_RG, p_2= Q_RB	0,4841	0,4516	$0,3824*10^{-3}$	$0,5333*10^{-3}$	$0,1558*10^{-3}$

Tabla 4.1.Estadísticos de la clase piel para la raza blanca

Clase Piel - Raza negra Espacio color \Estadísticos	m_{p1}	m_{p2}	F_{p1}^2	F_{p2}^2	$F_{p1p2}^2 = F_{p2p1}^2$
p_1=r ,p_2=g	0,2908	0,3223	$0,2891*10^{-3}$	$0,0570*10^{-3}$	$-0,0016*10^{-3}$
p_1=r, p_2=b	0,2908	0,3869	$0,2891*10^{-3}$	$0,3429*10^{-3}$	$-0,2875*10^{-3}$
p_1=g, p_2=b	0,3248	0,3704	$0,0570*10^{-3}$	$0,3429*10^{-3}$	$-0,0554*10^{-3}$
p_1=H, p_2=S	-1,1525	65,7933	0,0139	170,8465	0,7945
p_1=angA, p_2=angB	0,8432	0,8373	0,0014	0,0010	-0,0008
p_1=Q_RG, p_2= Q_RB	0,4740	0,4291	$0,2564*10^{-3}$	$0,6485*10^{-3}$	$0,3455*10^{-3}$

Tabla 4.2.Estadísticos de la clase piel para la raza negra

Clase Piel - Raza amarilla Espacio color \Estadísticos	m_{p1}	m_{p2}	F_{p1}^2	F_{p2}^2	$F_{p1p2}^2 = F_{p2p1}^2$
p_1=r ,p_2=g	0,2911	0,3309	$0,2290*10^{-3}$	$0,1978*10^{-3}$	$0,0545*10^{-3}$
p_1=r, p_2=b	0,2911	0,3780	$0,2290 *10^{-3}$	$0,3358*10^{-3}$	$-0,1835*10^{-3}$
p_1=g, p_2=b	0,3309	0,3780	$0,1978*10^{-3}$	$0,3358*10^{-3}$	$-0,1523*10^{-3}$
p_1=H, p_2=S	-1,2110	78,6037	0,0583	191,2119	1,0813
p_1=angA, p_2=angB	0,8622	0,8494	0,0015	0,0013	-0,0002
p_1=Q_RG, p_2= Q_RB	0,4679	0,4353	$0,2821*10^{-3}$	$0,5147*10^{-3}$	$0,1666*10^{-3}$

Tabla 4.3.Estadísticos de la clase piel para la raza amarilla

Clase No Piel Espacio color \Estadísticos	m_{p1}	m_{p2}	F_{p1}^2	F_{p2}^2	$F_{p1p2}^2 = F_{p2p1}^2$
p_1=r ,p_2=g	0,2630	0,3419	0,0024	0,0007	-0,0002
p_1=r, p_2=b	0,2630	0,3951	0,0024	0,0026	-0,0022
p_1=g, p_2=b	0,3419	0,3951	0,0007	0,0026	-0,0004
p_1=H, p_2=S	-0,9568	86,1897	5	1545,9	3,3
p_1=angA, p_2=angB	0,8335	0,9191	0,0010	0,0108	-0,0061
p_1=Q_RG, p_2= Q_RB	0,4319	0,3998	0,0030	0,0055	0,0034

Tabla 4.4. Estadísticos de la clase no piel

A continuación se presenta una tabla de los resultados de la clasificación para las distintas razas y los distintos espacios de color. Los resultados se presentan en tantos por ciento, han sido evaluados sobre el conjunto de imágenes de prueba y se engloban en cuatro grupos: p-p (pixels de piel clasificados como piel), p-np (pixels de piel clasificados como no piel), np-p (pixels de no piel clasificados como piel) y np-np (pixels de no piel clasificados como no piel).

	Raza blanca				Raza negra				Raza amarilla			
	p-p	p-np	np-p	np-np	p-p	p-np	np-p	np-np	p-p	p-np	np-p	np-np
rg	93,84	6,16	15,78	84,22	92,15	7,85	13,96	86,04	92,35	7,65	14,44	85,56
rb	92,64	7,36	16,80	83,20	92,09	7,91	15,98	84,02	92,05	7,95	14,98	85,02
gb	91,35	8,65	16,76	83,24	91,64	8,36	15,55	84,45	91,37	8,63	15,67	84,33
HSI	87,36	12,64	16,33	83,67	85,88	14,12	14,22	85,78	86,55	13,45	14,54	85,46
SCT	91,45	8,55	23,44	76,56	92,05	7,95	24,67	75,33	91,98	8,02	25,05	74,95
YQQ	90,37	9,63	23,89	76,11	91,87	8,13	24,10	75,9	92,15	7,85	24,98	75,02

Tabla 4.5 Errores de clasificación

Los errores cometidos en cada espacio de color para las distintas razas son muy parecidos. Si se suman los errores de los pixels que han sido mal clasificados (p-np, np-p) y para todas las razas, se obtienen los errores globales que se presentan en la tabla 4.6.

Como se puede observar, el espacioen el que se comete menos error de clasificación es el (rg). Sin embargo, hay que resaltar que la diferencia con el (rb) es tan solo del 1,72% y con el (gb) del 2,6%, aumentando para el resto. La causa hay que buscarla en que las tres cromaticidades tienen expresiones

muy similares y por lo tanto la diferencia entre ellas no puede ser muy acusada, siendo la (rg) la que tiene un mejor comportamiento a la hora se segmentar la piel humana.

Espacio	Error (%)
rg	21,94
rb	23,66
gb	24,54
HSI	28,43
SCT	32,56
YQQ	32,86

Tabla 4.6. Errores globales de clasificación

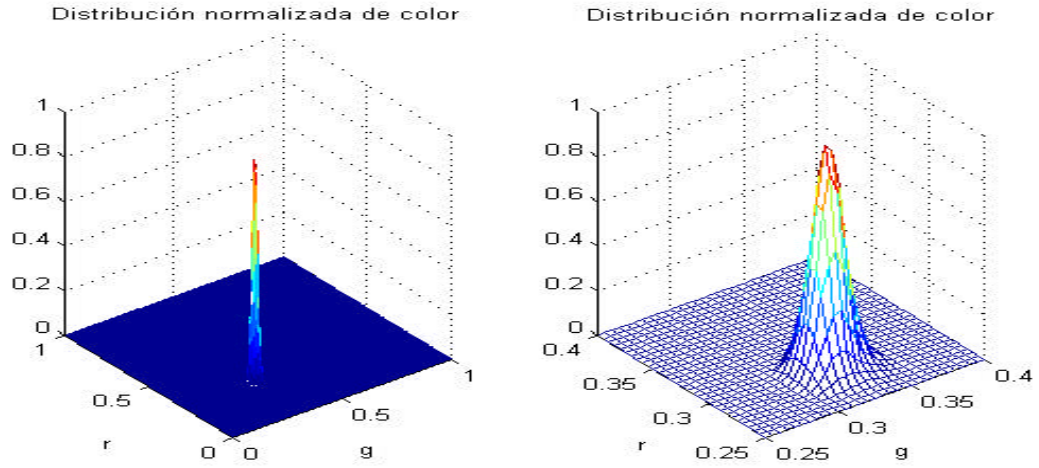
Por lo tanto, se puede concluir que después de hacer un estudio comparativo de distintos espacios de color para distintas razas, el espacio en el que se obtienen mejores resultados para segmentar la piel humana es el (rg).

4.3.- MODELO ESTOCÁSTICO DE LA PIEL

Una vez encontrado el espacio de trabajo, se va a estudiar la forma de la distribución de colores de la piel en este espacio y se va a modelar dicha distribución matemáticamente. Se hará un estudio para personas de diferentes razas y se demostrará que el modelo es aplicable independientemente de la misma. Por otro lado, hay que recordar que la distribución de colores no sólo depende del color de la piel sino también de la iluminación. Los cambios de luz se manifiestan como un desplazamiento en el espacio de las distribuciones de color. Caracterizar todas las distribuciones, independientemente de la iluminación, con un modelo fijo es prácticamente imposible. Sin embargo, bajo ciertas condiciones de luz, la forma de la distribución del color de la piel se mantiene constante, produciéndose únicamente un desplazamiento de la misma cuando la iluminación cambia.

En la figura 4.5 se muestran las distribuciones normalizadas del color de la piel para la persona de raza blanca de la figura 4.1.(a). Para obtener los pixels de piel se ha extraído la careta de la persona, a partir de la imagen original, de forma manual y empleando el programa comercial

"Paint Shop Pro". Este método de obtener los pixels de piel mejora al empleado en [Yang&Waibel, 97], donde se toma una ventana rectangular de pixels de la zona de la boca, con lo que utiliza una pequeña muestra de los mismos, y además se ven contaminados por ruido ya que consideran los pixels de labios y dientes como si fueran de piel.



(a) Vista general (b) Vista local

Figura 4.5. Distribución de pixels de piel de una persona de raza blanca en el espacio rg.

La pregunta que se plantea es qué función estadística aproxima mejor los datos. Como se puede ver gráficamente en la figura anterior, se puede asemejar a una distribución Gaussiana bidimensional. En la inmensa mayoría de los métodos estadísticos usados en ingeniería se asume una distribución normal de los datos; en esta tesis se va a demostrar esta hipótesis calculando el error cometido entre el modelo matemático Gaussiano de los datos y la distribución real de los mismos. Este método mejora el de "Quantile-Quantile plot (Q-Q plot)" usado en [Yang&Waibel,97] al no ser necesaria ninguna interpretación geométrica, ya que se va a disponer de la curva de error generada en cada punto.

El método seguido para el cálculo del error es el siguiente:

1.- Se calcula el histograma (H) de la clase piel en el espacio (rg). Como lo que se desea es analizar la forma de una distribución de datos se ha realizado una normalización de los mismos (H_n).

$$H_n(X) = \frac{H(X)}{\max\{H(X)\}} \tag{4.18}$$

2.- Se calculan los estadísticos (media y covarianza) de estos datos y con ellos se construye la función normal que mejor aproxime a los mismos. Como se está trabajando en espacios normalizados se aplicarán funciones como las de la ecuación 4.19. Esta expresión da la probabilidad de cada color de pertenecer a la clase piel.

$$f(X) \propto e^{-\frac{1}{2}(X-m_i)^T C_i^{-1}(X-m_i)} \tag{4.19}$$

3.- Se calcula el error cometido (E), en valor absoluto, entre el histograma real y la función de pertenencia:

$$E = \int_{r=0}^{r=1} \int_{g=0}^{g=1} |H_n(X) - f(X)| dX \tag{4.20}$$

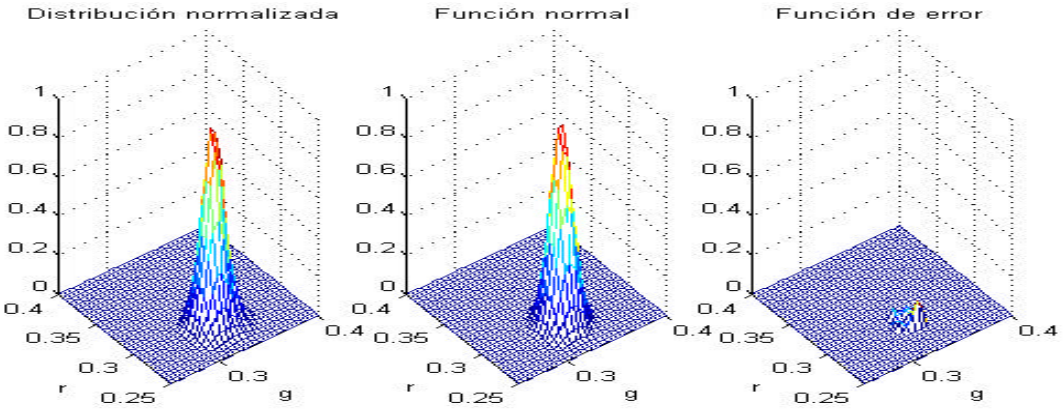
Se ha trabajado con un histograma muestreado de 200x200 celdillas, lo que supone una resolución de 0,005 en cada eje. Se han calculado tres funciones de error, una para cada raza, como se observa en la figura 4.6. Se ha evaluado el error para la parte representativa de la gaussiana, es decir hasta dos desviaciones típicas respecto a la media. En la tabla 4.7. se presentan los errores globales cometidos en tantos por ciento.

Raza	Error(%)
Blanca	4 %
Negra	12 %
Amarilla	7 %

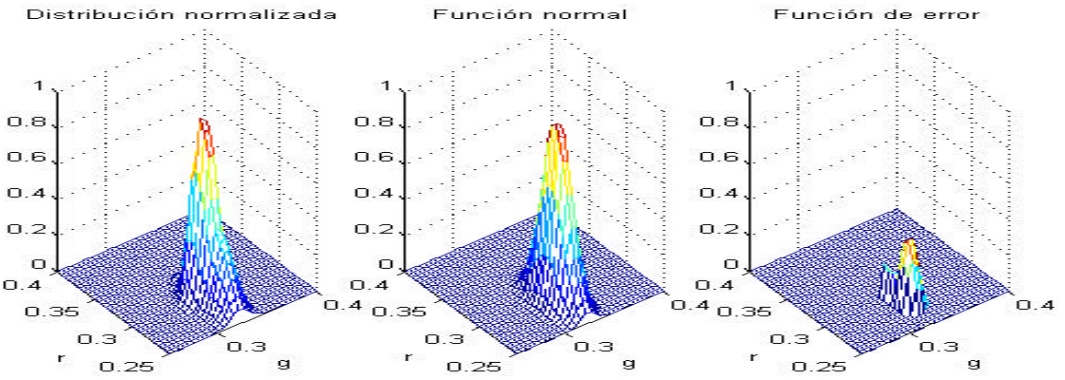
Tabla 4.7. Errores entre los modelos normales y las distribuciones de color

Se puede concluir que la aproximación de la distribución de color de piel, para cualquier raza, mediante una función normal es perfectamente asumible, ya que los errores cometidos para la raza blanca y amarilla se encuentran por debajo del 10% aumentando hasta el 12 % para la raza negra. Parte de estos errores son debidos a causas como: inexactitud en la toma de datos manual; redondeo que se

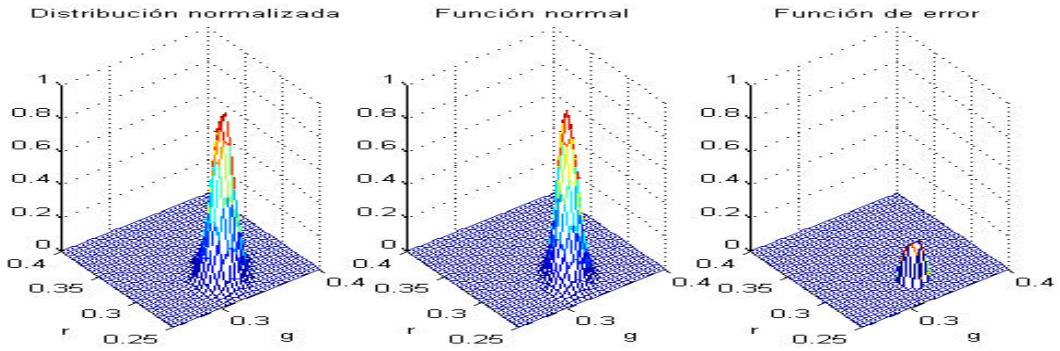
produce en el muestreo del histograma y en los colores, al ser codificados en 8 bits (entre 0 y 255). Otra causa de error es la iluminación no uniforme, que hace que la distribución se aleje más de la función normal que si no lo es.



(a) Raza blanca



(b) Raza negra



(b) Raza amarilla

Figura 4.6. Errores de los modelos normales para las distintas razas

4.3.1. Invarianzas del modelo de la piel

Se ha demostrado que se puede hacer una aproximación de la distribución de los colores de la piel en el espacio (rg) mediante un modelo normal. A lo largo de este capítulo se ha aludido a que el espacio (rg) es muy invariante a los distintos usuarios y a los cambios de iluminación. La pregunta que cabe plantearse ahora es: ¿es suficiente con utilizar un único modelo para caracterizar las distribuciones de color de cualquier persona y ante cualquier iluminación? Para responder a esta pregunta en el Anexo 1 se analiza la invarianza del modelo ante los siguientes parámetros: diferentes usuarios, traslaciones y giros, zoom y cambios de iluminación. Para ello se calculan las variaciones de los estadísticos respecto a los valores patrones obtenidos en el punto 4.2.

Las conclusiones que se extraen de este estudio son que el modelo gaussiano se puede considerar invariante a traslaciones, giros y zoom. En cuanto a diferentes usuarios ha quedado demostrado que las diferencias del grupo de prueba con el patrón es pequeña, e incluso las diferencias entre los patrones de distintas razas son también reducidas. Por último, con los cambios de iluminación las distribuciones mantienen su forma, produciéndose un pequeño desplazamiento de las medias de las distribuciones y manteniéndose prácticamente constantes sus varianzas. Con estas conclusiones quedan solventados los problemas 1) y 3) del uso del color en un segmentador de piel, planteados en la introducción. En cuanto al problema 2) de que cámaras diferentes producen colores diferentes, no se tiene en esta aplicación al disponer únicamente de una cámara. No obstante, aunque se cambiara, no supondría un problema en el método desarrollado en esta tesis, ya que el modelo es ajustado al usuario en un proceso de búsqueda inicial.

Con lo dicho, podría pensarse que el empleo de un modelo universal calculado a priori, que caracterice el color de piel, sería suficiente para segmentar la piel de cualquier persona, en cualquier posición y ante cualquier iluminación al estar sus variaciones muy limitadas. Sin embargo, esto no es así ya que en el espacio de trabajo una pequeña variación de parámetros puede suponer un gran error de clasificación. El efecto de trabajar con un espacio normalizado es que se minimizan las variaciones de color sobre este espacio; pero esto lleva consigo otro problema y es que las diferencias entre los colores de los distintos objetos de la imagen también disminuyen, produciéndose una compresión de los colores en una pequeña zona que hace difícil su clasificación al haber un gran solapamiento entre clases. En la figura 4.7. se muestran las proyecciones de las distribuciones de color de la figura 4.1, donde se puede apreciar el efecto comentado (el rojo representa el color de la piel y el verde el del fondo).

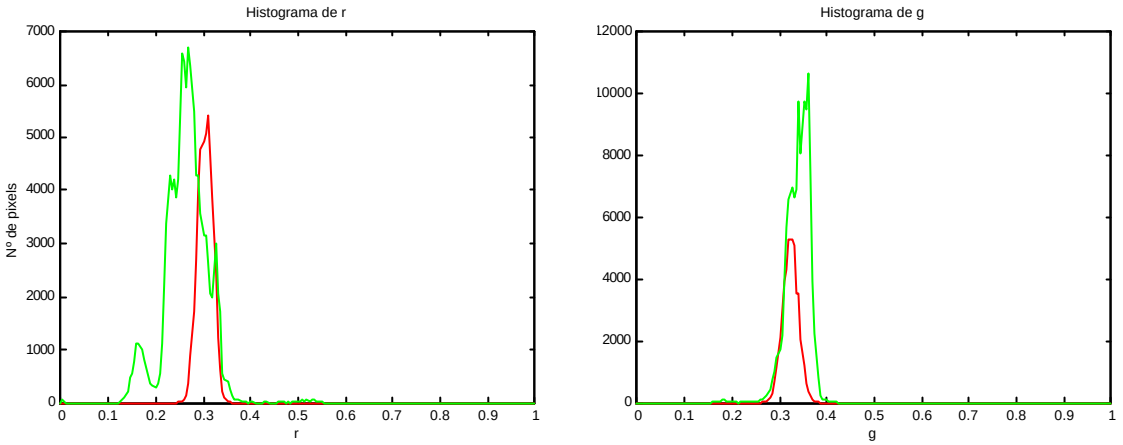


Figura 4.7. Solapamiento de la clase piel y el fondo en el histograma

Por lo tanto, el resultado de la segmentación con un modelo a priori permite localizar una parte de la piel del usuario que es inversamente proporcional a la distancia entre su distribución y el modelo. Pero además, colores de fondo similares a los de la piel humana pueden ser segmentados como piel y por lo tanto constituir un error de clasificación. Este error será dependiente del fondo que se tenga en cada momento y puede dar lugar a grandes errores.

En la figura 4.8 se muestran diferentes ejemplos de segmentación tomando como modelo de clase piel el patrón de piel blanca. En el ejemplo (a) se observa que la segmentación de la piel es bastante acertada pero se introducen gran cantidad de pixels de fondo. En (b) el error de clasificación es muy grande, considerando piel prácticamente todos los pixels de la imagen. En (c) se segmenta la piel de una persona de raza negra, consiguiéndose un gran grado de acierto en la piel pero también se introducen una gran cantidad de pixels de fondo como piel.

Estos ejemplos demuestran que el empleo de un modelo universal no es aplicable para realizar un buen segmentador de piel, a pesar de la gran invarianza existente en las distribuciones de color de la misma. Por lo tanto, es preciso crear un modelo para cada usuario que se adapte perfectamente a su distribución y que minimice los errores de clasificación. El problema que se plantea es: ¿cómo se localiza el modelo de la piel dentro del histograma? Para responder a esta pregunta se va a analizar el problema más general del modelado estocástico no supervisado de un histograma, que se tratará en el siguiente punto.

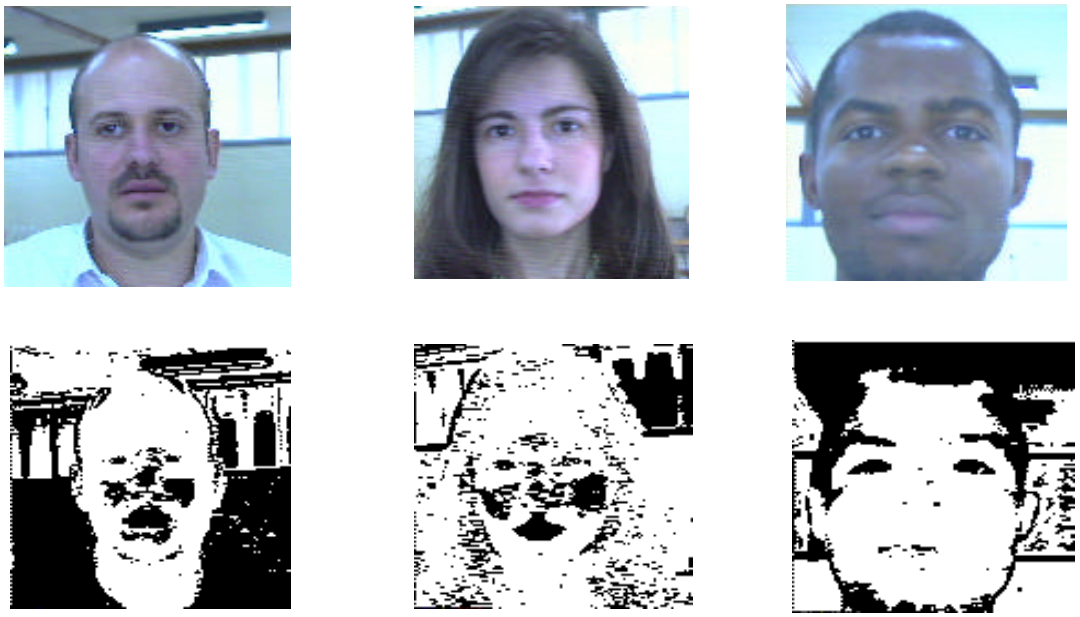


Figura 4.8. Solapamiento de la clase piel y el fondo en imágenes

4.4.- MODELO ESTOCÁSTICO NO SUPERVISADO DE UN HISTOGRAMA

El problema de segmentación de una imagen se puede abordar de una forma genérica como un problema de optimización estadística. La segmentación del color de la piel de una imagen se puede resolver modelando estocásticamente los distintos colores que aparecen en una imagen y localizando entre ellos el modelo de la piel. Para la comprensión de este procedimiento se van a describir los fundamentos teóricos del modelado estocástico no supervisado.

Sean un conjunto finito de pixels de una imagen $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$, donde cada pixel viene definido por sus componentes de color "rg" $\mathbf{x}_j=(x_{jr},x_{jg})$. Por simplicidad se ha indexado una imagen 2-D como un array 1-D de longitud N. Supóngase que existen K clases y que se quieren clasificar los N componentes de **X**. Consideremos un modelo de K componentes (M_K) donde cada modelo está definido por un vector de parámetros 2_K, U^d y, por lo tanto, tiene d estadísticos. Supóngase conocida la probabilidad a priori $P(T_i)$ de cada clase así como la estructura probabilística de cada una, que consideraremos normal. Dentro de un enfoque bayesiano, si $P(X \mid T_{i,} 2_i)$ es la probabilidad de que un patrón tome en la clase i el valor **X**. Para los estadísticos de todas las clases $2=(2_{1,} 2_2, ... 2_K)$ la probabilidad de **X** será:

$$P(X|\theta) = \sum_{i=1}^{K} P(X|\omega_i, \theta_i) P(\omega_i) \tag{4.21}$$

Se denomina P(X | 2) a la probabilidad total ("mixture density"), las probabilidades condicionales P(X | T_i, 2_i) serán las componentes de la probabilidad total ("component densities") y las probabilidades a priori P(T_i) serán los parámetros de la mezcla ("mixing parameters"). El objetivo de la clasificación no supervisada será el estimar el vector 2. Una vez conocido el vector 2 se descompone la probabilidad total en sus componentes.

Para la estimación de 2 se va a aplicar el método de máxima probabilidad ("maximum likelihood") consistente en estimar el vector de parámetros $\hat{\theta} = (\hat{\theta}_1, \hat{\theta}_2, ..., \hat{\theta}_K)$ que maximice la probabilidad P(X | 2), tal que $\hat{\theta}_i = (\hat{\theta}_{1i}, \hat{\theta}_{2i}, ..., \hat{\theta}_{di})$ da la estimación de los estadísticos que definen P(X | T_i, 2_i).

Siendo la probabilidad para un pixel:

$$P(x_j|\theta) = \sum_{i=1}^{K} P(x_j|\omega_i, \theta_i) P(\omega_i) \tag{4.22}$$

la probabilidad de que de la imagen se haya extraído la muestra **X** será la probabilidad conjunta de cada pixel:

$$P(X|\theta) = \prod_{j=1}^{N} P(x_j|\theta) \tag{4.23}$$

Desde un punto de vista analítico es preferible trabajar con el logaritmo neperiano de (4.23) llamado logaritmo de la probabilidad ("log-likelihood"), de forma que, al ser monótonamente creciente, resulta que el vector $\hat{\theta}$ que maximiza el logaritmo también maximiza la función.

$$L(X|\theta) = \ln(P(X|\theta)) = \ln \prod_{j=1}^{N} P(x_j|\theta) = \sum_{j=1}^{N} \ln P(x_j|\theta) \tag{4.24}$$

Por lo tanto, el vector $\hat{\theta}$ será aquel para el cual las primeras derivadas del logaritmo de la función (4.24) sean nulas.

$$\frac{\partial L(X|\theta)}{\partial \theta_{li}} = \sum_{j=1}^{N} \frac{1}{P(x_j|\hat{\theta})} \frac{\partial(\sum_{k=1}^{K} P(x_j|\omega_k,\hat{\theta}_k)P(\omega_k))}{\partial \theta_{li}} = 0 \qquad (4.25)$$

Introduciendo la hipótesis de que los vectores de los estadísticos 2_i y 2_k para i,k =1,2,...,K, i..k, son independientes, y considerando la probabilidad a posteriori:

$$P(\omega_i|x_j,\hat{\theta}_i) = \frac{P(x_j|\omega_i,\hat{\theta}_i)P(\omega_i)}{P(x_j|\hat{\theta})} \qquad (4.26)$$

la derivada de la ecuación (4.25) se puede escribir de la siguiente forma:

$$\begin{aligned}
\frac{\partial L(X|\theta)}{\partial \theta_{li}} &= \sum_{j=1}^{N} \frac{P(\omega_i|x_j,\hat{\theta}_i)}{P(x_j|\omega_i,\hat{\theta}_i)P(\omega_i)} \frac{\partial(P(x_j|\omega_i,\hat{\theta}_i)P(\omega_i) + \sum_{\substack{k=1 \\ i\neq k}}^{K} P(x_j|\omega_k,\hat{\theta}_k)P(\omega_k))}{\partial \theta_{li}} \\
&= \sum_{j=1}^{N} \frac{P(\omega_i|x_j,\hat{\theta}_i)}{P(x_j|\omega_i,\hat{\theta}_i)P(\omega_i)} P(\omega_i) \frac{\partial(P(x_j|\omega_i,\hat{\theta}_i))}{\partial \theta_{li}} \\
&= \sum_{j=1}^{N} \frac{P(\omega_i|x_j,\hat{\theta}_i)}{P(x_j|\omega_i,\hat{\theta}_i)} P(x_j|\omega_i,\hat{\theta}_i) \frac{\partial(\ln P(x_j|\omega_i,\hat{\theta}_i))}{\partial \theta_{li}} \\
&= \sum_{j=1}^{N} P(\omega_i|x_j,\hat{\theta}_i) \frac{\partial(\ln P(x_j|\omega_i,\hat{\theta}_i))}{\partial \theta_{li}} = 0
\end{aligned} \qquad (4.27)$$

Los valores $\hat{\theta}_{li}$ l 0 [1,d], i 0 [1,K] que satisfacen la ecuación anterior son la solución del problema ya que maximizan la función log-probabilidad.

Si las probabilidades a priori P(T_i) son también desconocidas, las estimaciones $\hat{P}(\omega_i)$ se obtienen a partir de la ecuación (4.24) con las condiciones adicionales de que:

$$\hat{P}(\omega_i) > 0 \qquad \text{y} \qquad \sum_{k=1}^{K} \hat{P}(\omega_k) = 1 \qquad (4.28)$$

Por lo tanto, la ecuación (4.24) se puede transformar en:

$$L(P(\omega)) = \ln P(X|\theta) + \lambda((\sum_{k=1}^{K} P(\omega_k) - 1) \qquad (4.29)$$

De donde:

$$\frac{\partial(P(\omega))}{\partial P(\omega_i)} = \sum_{j=1}^{N} \frac{\partial(\ln P(x_j|\hat{\theta}))}{\partial P(\omega_i)} + \frac{\partial(\lambda(\sum_{k=1}^{K} \hat{P}(\omega_k) - 1))}{\partial P(\omega_i)}$$

$$= \sum_{j=1}^{N} \frac{1}{P(x_j|\hat{\theta})} \frac{\partial(P(x_j|\omega_i, \hat{\theta}_i)P(\omega_i) + \sum_{\substack{k=1 \\ i \neq k}}^{K} P(x_j|\omega_k, \hat{\theta}_k)\hat{P}(\omega_k))}{\partial P(\omega_i)} \tag{4.30}$$

$$+ \frac{\partial(\lambda(\hat{P}(\omega_i) - 1) + \lambda \sum_{\substack{k=1 \\ i \neq k}}^{K} \hat{P}(\omega_k))}{\partial P(\omega_i)}$$

$$= \sum_{j=1}^{N} \frac{P(\omega_i|x_j, \hat{\theta}_i)}{P(x_j|\omega_i, \hat{\theta}_i)P(\omega_i)} P(x_j|\omega_i, \hat{\theta}_i) + \lambda = 0$$

sumando en i las ecuaciones anteriores resulta:

$$\sum_{i=1}^{K} \sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i) = -\lambda \sum_{i=1}^{K} \hat{P}(\omega_i) \tag{4.31a}$$

$$\sum_{j=1}^{N} (\sum_{i=1}^{K} P(\omega_i|x_j, \hat{\theta}_i) = -\lambda \tag{4.31b}$$

$$N = -\lambda \tag{4.31c}$$

Por lo tanto, de la ecuación (4.30) se obtiene que la probabilidad a priori $\hat{P}(\omega_i)$ debe satisfacer la condición:

$$\hat{P}(\omega_i) = \frac{1}{N} \sum_{j=1}^{N} P(\omega_i|x_j, \hat{\theta}_i) \qquad \forall\, i = 1,2,\ldots,K \tag{4.32a}$$

además de las condiciones:

$$\hat{P}(\omega_i) > 0 \quad \text{y} \quad 1 = \sum_{i=1}^{K} \hat{P}(\omega_i) \tag{4.32b}$$

y el vector $\hat{\theta}_i$, para i=1,2,...,K debe satisfacer la condición:

$$\sum_{j=1}^{N} P(\omega_i|x_j,\hat{\theta}_i)\frac{\partial(\ln(P(x_j|\omega_i,\hat{\theta}_i))}{\partial\theta_{li}} = 0 \qquad (4.32c)$$

donde:

$$P(\omega_i|x_j,\hat{\theta}_i) = \frac{P(x_j|\omega_i,\hat{\theta}_i)\hat{P}(\omega_i)}{\sum_{i=1}^{K} P(x_j|\omega_i,\hat{\theta}_i)\hat{P}(\omega_i)} \qquad (4.32d)$$

Supóngase que las funciones de probabilidad condicional $P(X|\Theta_i,2_i)$ y componentes de la probabilidad total siguen una ley normal multivariable $N(\mathbf{m_i},\mathbf{C_i})$, donde $\mathbf{m_i}$ representa el vector de medias y $\mathbf{C_i}$ el de covarianzas de la clase i. Por lo tanto el log-probabilidad condicionada será:

$$L(x_j|\hat{\theta}_i) = \ln(P(x_j|\omega_i,\hat{\theta}_i)) = \ln\left\{\frac{1}{2\pi|\hat{C}_i|^{1/2}}\exp\left\{-\frac{1}{2}(x_j-\hat{m}_i)^T\hat{C}_i^{-1}(x_j-\hat{m}_i)\right\}\right\}$$
$$= -\ln(2\pi|\hat{C}_i|^{1/2}) - \frac{1}{2}(x_j-\hat{m}_i)^T\hat{C}_i^{-1}(x_j-\hat{m}_i) \qquad (4.33)$$

En un problema de segmentación no supervisada se desconocen los vectores media ($\mathbf{m_i}$), las matrices de covarianza ($\mathbf{C_i}$) , las probabilidades a priori de cada clase ($P(\Theta_i)$) y el número de clases (K). Utilizando las ecuaciones (4.32), partiendo de una función de probabilidad como la de la ecuación (4.33), se calculan los parámetros anteriores empleando el método de máxima probabilidad [Oliver et al., 96], según las siguientes ecuaciones:

$$P(\omega_i|x_j,\hat{\theta}_i) = \frac{P(x_j|\omega_i,\hat{\theta}_i)\hat{P}(\omega_i)}{\sum_{i=1}^{K} P(x_j|\omega_i,\hat{\theta}_i)\hat{P}(\omega_i)}$$
$$= \frac{|\hat{C}_i|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x_j-\hat{m}_i)^T\hat{C}_i^{-1}(x_j-\hat{m}_i)\right\}\hat{P}(\omega_i)}{\sum_{i=1}^{K}|\hat{C}_i|^{-\frac{1}{2}}\exp\left\{-\frac{1}{2}(x_j-\hat{m}_i)^T\hat{C}_i^{-1}(x_j-\hat{m}_i)\right\}\hat{P}(\omega_i)} \qquad (4.34)$$

$$\hat{P}(\omega_i) = \frac{1}{n}\sum_{j=1}^{N} P(\omega_i|x_j,\hat{\theta}_i) \qquad \forall i = 1,2,\ldots,K \qquad (4.35)$$

$$\hat{m}_i = \frac{\sum_{j=1}^{N} P(\omega_i | x_j, \hat{\theta}_i) x_j}{\sum_{j=1}^{N} P(\omega_i | x_j, \hat{\theta}_i)} \tag{4.36}$$

$$\hat{C}_i = \frac{\sum_{j=1}^{N} P(\omega_i | x_j, \hat{\theta}_i)(x_j - \hat{m}_i)(x_j - \hat{m}_i)^T}{\sum_{j=1}^{N} P(\omega_i | x_j, \hat{\theta}_i)} \tag{4.37}$$

para i $\in$ [1,K] y j $\in$ [1,N].

Aunque la notación de las ecuaciones anteriores es aparatosa, su interpretación es sencilla. La estimación de la probabilidad a priori $\hat{P}(\omega_i)$ es el promedio de las probabilidades a posteriori que tiene la muestra **x** de pertenecer a la clase i, $\hat{m}_i$ es el vector media de la clase i ponderado por las probabilidades a posteriori de pertenecer a esta clase y $\hat{C}_i$ es la estimación ponderada de la matriz de covarianzas de la clase i. Nótese que los parámetros anteriores se calculan para un valor de K dado, que en principio es un parámetro desconocido y, por lo tanto, supone otra variable a estimar ($\hat{K}$). Su estimación se suele abordar a parte y por lo tanto será explicada una vez se hayan ajustado los demás parámetros.

Estas ecuaciones no obtienen explícitamente los valores $\hat{P}(\omega_i), \hat{m}_i, \hat{C}_i$. Es un conjunto de ecuaciones no lineales que no producen una solución única y para resolverlas hay que emplear un procedimiento iterativo.

Un algoritmo muy utilizado para este problema es el conocido como EM (Expectation-Maximization) [Langan et al.,98], que sugiere el procedimiento iterativo de la figura 4.9. Para una iteración p en el paso E se obtiene la distribución normal multivariable de la probabilidad condicional $P(X|\omega_i, \hat{\theta}_i)^{(p)}$ y la probabilidad a posteriori $P(\omega_i|X, \hat{\theta}_i)^{(p)}$. Con estos datos en el paso M se obtiene una nueva estimación de los parámetros $\hat{\theta}_i^{(p+1)}$ aplicando el criterio de máxima probabilidad. El algoritmo se aplica de forma iterativa hasta que la diferencia de la función a maximizar $L(X|\hat{\theta}_i)$ entre las iteraciones p y p+1 no sea superior a un valor predeterminado *.

En cuanto a la estimación de K, es conocida como el problema de validación de cluster ("cluster validation"). La solución lógica a este problema, según el planteamiento expuesto, sería considerar K como un nuevo parámetro a ser estimado dentro del modelo 2. Sin embargo se ha demostrado que esta solución, empleando un método de máxima probabilidad, no da buenos resultados, al obtener un

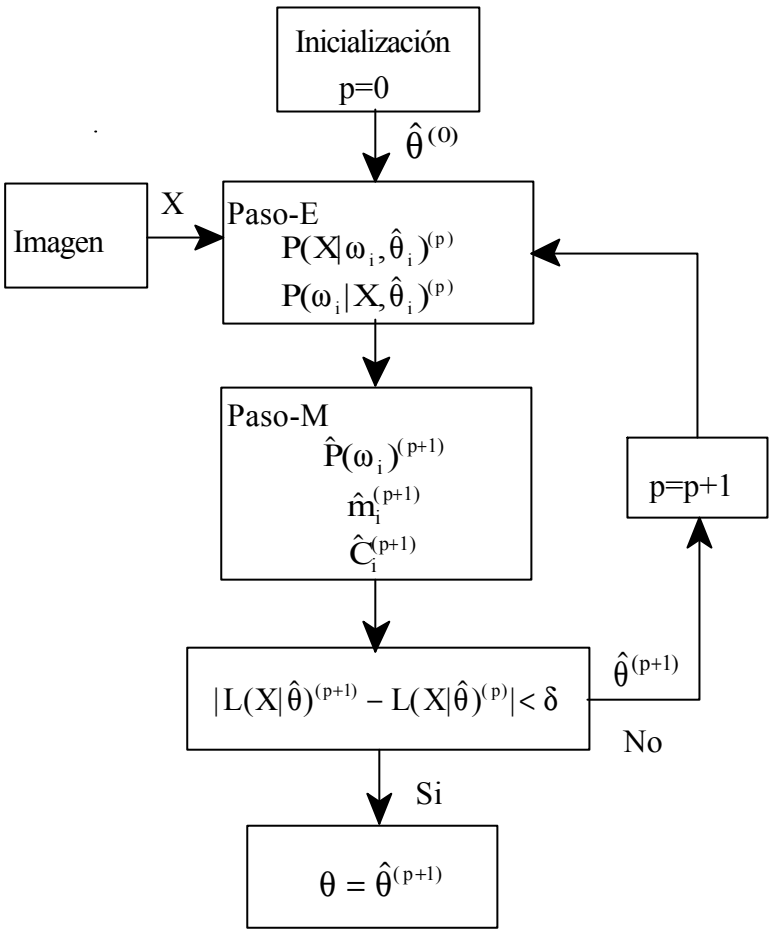


Figura 4.9. Diagrama de bloques del procedimiento EM

valor monotonamente creciente para K [Langan et al., 98][1].

Una alternativa para la solución del problema consiste en realizar una segmentación para diferentes clases (K) candidatas, según el método empleado, y fijar una función de coste que permita determinar el K óptimo. Dentro de esta línea se encuentran distintos métodos que difieren en la función de coste empleada. Entre ellos se destacan los siguientes:

Esta es la razón por la que se le ha considerado como un parámetro independiente que se calcula una vez ajustados el resto de los parámetros.

1) Fuzzy Hypervolume (FHV). Se explica en [Gath et al.,89] y ajusta el modelo en función del mínimo volumen total definido como:

$$V(k) = \sum_{k=1}^{K} \sqrt{|C_k|} \tag{4.38}$$

2) Evidence density. Esta técnica se desarrolla en [Roberts, 97] y utiliza el parámetro anterior para penalizar la medida de log-probabilidad.

$$(k) = \frac{L(X|q^*)}{V(K)} \tag{4.39}$$

3) Minimum Description Length (MDL). Desarrollado por Rissanen [Rissanen, 78] selecciona el orden del modelo que minimiza una función de longitud formada por una mezcla de datos y parámetros del modelo.

$$MDL(k) = -L(X|q) + \frac{1}{2} N_p(K) \ln N \tag{4.40}$$

Donde $N_p(K)$ es el número de parámetros en el modelo Gaussiano K.

4) Minimum Message Length (MML). Creado por Wallace y Freeman y posteriormente ampliado por [Oliver et al.,96]. Utiliza la siguiente función de ajuste.

$$\begin{aligned} \mathrm{MML(K)} \approx\ & \mathrm{Kd}\ln(2\sigma_{\mathrm{pop}}^2) - \ln(\mathrm{K}-1)! + \frac{\mathrm{N_p}}{2}\ln\kappa(\mathrm{N_p}) - \ln \mathrm{K}! \\ & + \sum_{\mathrm{i=1}}^{\mathrm{d}} \sum_{\mathrm{k=1}}^{\mathrm{K}} \ln\frac{\sqrt{2}\mathrm{N_k}}{\sigma_{\mathrm{k,i}}^2} + \frac{1}{2}\ln \mathrm{N} - \frac{1}{2}\sum_{\mathrm{k=1}}^{\mathrm{K}} \ln \mathrm{P(k)} - \mathrm{L(X|\hat{\theta})} + \frac{\mathrm{N_p}}{2} \end{aligned} \tag{4.41}$$

El valor de $\mathsf{F}_{\mathrm{pop}}$ proviene de asumir que a priori cada componente del vector media para cada una de las K gaussianas tiene una distribución plana en el rango $(-\mathsf{F}_{\mathrm{pop}}, \mathsf{F}_{\mathrm{pop}})$. Asimismo, los elementos de la diagonal de la matriz de covarianza de cada gaussiana se toman como $\mathsf{F}_{\mathrm{pop}}^2$. $\mathsf{6}(\mathrm{N_p})$ es la constante de cuantificación de "lattice" óptima en un espacio $\mathrm{N_p}$ dimensional. Estas constantes se obtienen de una tabla de constantes "lattice" aunque hay que resaltar que para alguna dimensión no existen (por ejemplo D=9) y que para otras están muy poco especificadas.

5) Gaussian Mixture Modeling (GMM). Desarrollado en [Roberts et al.,98]. Plantea que la evidencia de una muestra (lnP(X)) depende de tres factores:

$$\ln \mathrm{P(X)} = \mathrm{L(X|\hat{\theta})} + \mathrm{f_{post}(H)} + \mathrm{f_{prior}}(\hat{\theta}, \mathrm{X}) \tag{4.42}$$

a) del log-probabilidad bajo un enfoque bayesiano (L(X|2)); b)de una función a priori (f_{prior}); c) de una función a posteriori (f_{post}) que depende de la matriz Hessiana de los parámetros del GMM. Con lo cual la evidencia estimada de X será:

$$\ln P(X) = L(X|\hat{\theta}) - Kd\ln(2\sigma^2_{pop}) + \ln(K-1)! + \frac{N_p}{2}\ln(2\pi)$$
$$-\frac{1}{2}\left(\sum_{k=1}^{K-1}\ln\sum_{j=1}^{N}\left(\frac{P(\omega_k|x_j,\hat{\theta}_k)}{\hat{P}(\omega_k)} - \frac{P(\omega_K|x_j,\hat{\theta}_K)}{\hat{P}(\omega_K)}\right)^2 + 2d\sum_{k=1}^{K}\ln(\sqrt{2}N\hat{P}(\omega_k)) - 2\sum_{k=1}^{K}\sum_{i=1}^{d}\ln\lambda_{k,i}\right) \quad (4.43)$$

donde $8_{k,l}$ son los autovalores de la matriz de covarianza. Por simplicidad se suele considerar las matrices de covarianza como diagonales y entonces: $8_{k,l} = F^2_{k.l.}$ Nótese que la función a priori es similar a la del método anterior, donde la constante de lattice ha sido sustituída por el límite inferior de la constante y para aquellos valores que no existe se usa interpolación lineal.

Todos estos métodos introducen términos de penalización cada vez más sofisticados sobre la función L(X|2), para intentar obtener un método general aplicable en cualquier caso. Sin embargo, este objetivo no se ha alcanzado ya que aparecen los siguientes problemas:

- •Cuando aumenta el número de clases, el log-probabilidad evaluada en la imagen segmentada también aumenta.

- •Los términos de penalización son dependientes del tamaño de la imagen.

- • No se garantiza el mínimo global ya que la función a minimizar depende de una gran cantidad de parámetros inter-relacionados y, por lo tanto, existen muchos mínimos locales que hacen que la solución sea muy dependiente de la inicialización.

- •La velocidad de convergencia del método es muy baja al existir una gran cantidad de parámetros a estimar.

Todos estos problemas llevan a autores como David A. Langan a considerar que el problema de "validación de clusters" está todavía sin resolver en nuestros días [Langan et al.,98].

Una alternativa al modelado estocástico de un histograma para realizar segmentación es el empleo de técnicas de agrupación o "clustering". Estas técnicas plantean que, dado **X**, el cual está formado por

un conjunto K clases: $\mathbf{X}=\{\mathbf{X_1}, \mathbf{X_2}, ...,\mathbf{X_K}\}$, se pueden encontrar un conjunto de subregiones $(T_1,T_2,...,T_K)$ tal que cada patrón $\mathbf{X_j}$ se agrupe en una y sólo una de las regiones:

$$\omega_1 \cup \omega_2 \cup ... \cup \omega_K = X \qquad (4.44.a)$$

$$\omega_i \cap \omega_k = \varnothing \quad \text{para} \quad i \neq k \qquad (4.44.b)$$

Simplifican considerablemente el proceso de segmentación, aunque no son tan exactas como las técnicas estocásticas. Entre ellas destacan el algoritmo de las k-medias y el ISODATA.

El algoritmo de las k-medias es sencillo pero poco eficiente cuando no se conoce el número de clases a priori con exactitud [Duda&Hart,73]. Partiendo de una muestra a clasificar, **X,** se aplica un proceso iterativo con las siguientes operaciones:

1º Se toman al azar entre los elementos a agrupar K vectores, de forma que inicialmente constituyen los centroides (al ser los únicos elementos) de las K clases.

$$a_1 : Z_1^{(0)}; a_2 : Z_2^{(0)}; ...; a_K : Z_K^{(0)}; \qquad (4.45)$$

Donde se ha introducido entre paréntesis el índice iterativo del algoritmo.

2º En la iteración p se distribuyen todas las muestras entre las K clases según la siguiente regla:

$$\begin{aligned} &X \in \alpha_i^{(p)} \quad \text{si} \quad ||X - Z_i^{(p)}|| < ||X - Z_k^{(p)}|| \\ &\forall k = 1,2,...,K / k \neq i \end{aligned} \qquad (4.46)$$

3º Una vez redistribuidos los elementos a agrupar entre las diferentes clases, se actualizan los centroides de las clases utilizando la media aritmética de $\omega_k^{(p)}$.

$$Z_k^{(p+1)} = \frac{1}{N_k^{(p)}} \sum_{X \in a_k^{(p)}} X \qquad ; k = 1,2,...,K \qquad (4.47)$$

Siendo $N_i^{(p)}$ el número de elementos de la clase ω_i en la iteración (p).

4º Se comprueba si el algoritmo ha alcanzado una posición estable, es decir, si se cumple:

$$Z_k^{(p+1)} = Z_k^{(p)} \qquad (4.48)$$

5º Cuando esto ocurre se calcula una función de coste en función de la distancia de las muestras con

su centroide según la siguiente ecuación:

$$F = \sum_{k=1}^{K} \sum_{X \in a_k^{(p)}} || X - Z_k^{(p)} ||^2 \tag{4.49}$$

Si esta función se encuentra por debajo de un umbral establecido de forma heurística se considera el ajuste como bueno; en caso contrario, se vuelve a repetir el proceso.

Este algoritmo es muy eficiente si se sabe con exactitud el número de clases, pero en caso contrario surgen los siguientes problemas:

- Se necesita saber el número de clases a priori.

- El método que emplea para calcular K no es muy óptimo ya que considera que todas las clases tienen la misma dispersión estadística para cada dimensión (d) de **X**.

- Como trabaja con distancia euclídea para cada muestra se mueve el centroide para el que la distancia euclídea es menor. Este hecho hace que el aprendizaje del sistema sea muy local y dependiente de la inicialización, de manera que puede dar lugar a clusters erróneos.

El algoritmo ISODATA (Iterative Self-organizing Data Analysis Techniques) es un método de agrupación de datos basado en el algoritmo de las K-medias, al que se le ha añadido una serie de parámetros y operaciones con un elevado contenido heurístico [Maravall, 93]. Se definen una serie de parámetros como: número de clases, número mínimo de miembros de una clase, desviación típica máxima, distancia euclídea mínima entre dos clusters, número de fusiones que se pueden llevar a cabo en una iteración, número de iteraciones máximo, etc. Una vez realizado un primer "clustering", realizan un proceso de unión y división de clases, evaluando las distancias medias de cada una de ellas con las distancias medias de todas las clases y las varianzas de cada una, e imponiendo una serie de umbrales de forma heurística dados por el usuario a priori.

Los problemas que plantea este algoritmo son:

•Empleo de parámetros heurísticos, que deben ser introducidos por el usuario, para calcular el número de clases.

•Al igual que en las K-medias, utiliza distancia euclídea, lo que hace que la separación de las clases sea lineal y además tenga un aprendizaje local y dependiente de las condiciones iniciales del sistema.

4.5.- SEGMENTADOR PROPUESTO

Según todo lo explicado hasta ahora, el método más generalista de realizar un proceso de segmentación de la piel es el basado en modelos estocásticos, dentro de un enfoque bayesiano y usando funciones normales. El empleo de un modelo calculado a priori de forma supervisada y "off-line" funciona correctamente si la clase piel no se aleja de sus valores patrones y si los objetos del fondo no se encuentran muy solapados con el objeto piel en el espacio de color elegido. La utilización de un espacio normalizado de color "rg" da una gran invarianza al modelo respecto a personas de la misma o distinta raza, cambios de iluminación, traslaciones y giros. Sin embargo, la normalización hace que la distribución de colores de los objetos del fondo queden muy concentradas en torno a la de la piel y por, lo tanto, existe muy poca distancia entre unas distribuciones de color y otras. Este hecho hace que pequeñas variaciones de posición de la función normal, que modela la distribución de la piel, den lugar a errores considerables en la segmentación.

Todo lo anterior lleva a considerar que un modelo definido a priori únicamente funcionará con imágenes preparadas, donde los parámetros de color de las caras no difieran de los parámetros patrones. Para poder generalizar el segmentador se ha recurrido a definir un modelo de color de piel adaptado al usuario. Para ello se realiza un proceso de "clustering" de la imagen, con lo que se consigue hacerlo totalmente autónomo. La idea consiste en hacer una segmentación de las principales distribuciones de color presentes en la imagen y posteriormente localizar entre ellas la que corresponde a la clase piel.

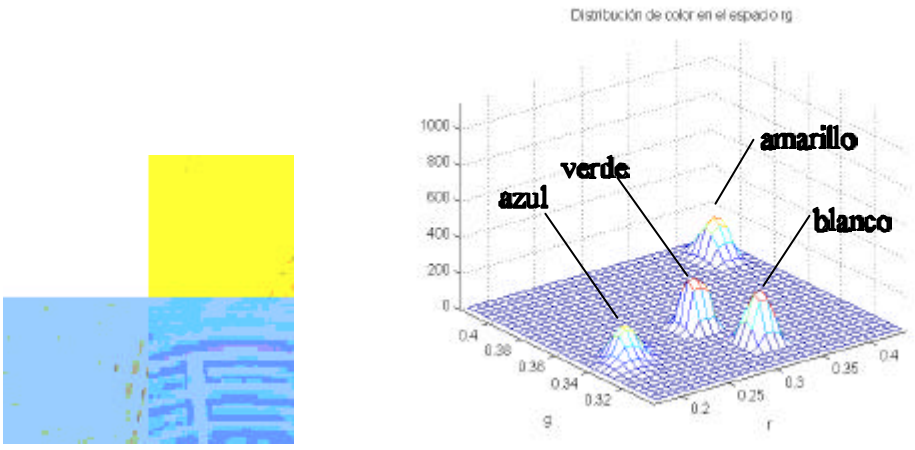
Las dos ventajas fundamentales que introduce este método son:

1.- Para cada usuario se calcula su propio modelo

2.- Su cálculo se realiza de forma no supervisada y, por lo tanto, se evita la etapa de obtención de parámetros "off-line".

El proceso de "clustering" empleando modelos estocásticos se fundamenta en que las distribuciones de los pixels de una imagen en su histograma de color pueden ser consideradas como variables estocásticas y modeladas matemáticamente mediante funciones gaussianas.

El color es una percepción visual que depende de la resolución con la que se esté trabajando. Así por ejemplo, un color para una determinada resolución se puede descomponer en varios cuando ésta aumente. Teóricamente si se toma una imagen de un objeto con un único color, éste se debería proyectar en una delta en el espacio de color "rg". Sin embargo esto no ocurre así, sino que los pixels se distribuyen adoptando una forma gaussiana. En la figura 4.10. se presentan una imagen formada por cuatro colores uniformes[2] y su distribución en el espacio "rg", donde se puede comprobar la afirmación hecha. El efecto de dispersión se debe, entre otros, a la falta de uniformidad de la iluminación y la distorsión introducida por la óptica y la respuesta del CCD.



(a) (b)

Figura 4.10.(a) Imagen de test en color (b)Distribución de los pixels en el espacio rg

Estas imágenes corresponden a trozos de piezas de plástico tomadas en el "Computer Science Departament of The Trinity College in Dublin (Irland)" donde el autor estuvo colaborando en un proyecto de detección de defectos en piezas industriales.

Como se puede observar, el espacio de color "rg" tiene una topología que puede ser modelada por un número de funciones gaussianas igual al de los colores de la imagen. Cada una de estas funciones representará una función de pertenencia de los pixels a un determinado color. Analizando el grado de pertenencia de cada pixel a las funciones, se puede realizar la segmentación. Un proceso de "clustering" se plantea en sentido contrario al explicado anteriormente: "A partir de la topología del histograma en el espacio "rg" se deben calcular el número de funciones normales que forman el modelo y los parámetros de cada una de ellas".

Dentro de un enfoque bayesiano, y empleando el método de máxima probabilidad habrá que estimar el número de funciones del modelo ($\hat{K}$) y los estadísticos de cada una de ellas ($\hat{P}(\omega_i), \hat{m}_i, \hat{C}_i$). Como se ha explicado en el punto anterior, el empleo de un método de ajuste de todos los parámetros trae consigo una gran complejidad de cálculo y problemas en la convergencia de los mismos, lo que lo hace inviable para una aplicación como la planteada en esta tesis, donde el algoritmo debe ejecutarse en un corto período de tiempo y además se puede tolerar un cierto error de clasificación.

No obstante, se pueden asumir una serie de hipótesis sobre el problema planteado, que conlleva una simplificación de la probabilidad a posteriori $P(\omega_i | x_j, \hat{\theta}_i)$ del modelo, entendida como la función discriminante de la clasificación.

Hipótesis 1. Desnormalización de la probabilidad. El objetivo consiste en clasificar muestras en función del valor de las probabilidades a posteriori de las distintas clases. Por lo tanto, como se trata de comparar varios valores, todos ellos normalizados por el mismo factor, se puede prescindir de la normalización, con lo que la ecuación (4.34) dará lugar a la siguiente función discriminante (fd):

$$fd(\omega_i | x_j, \hat{\theta}_i) = |\hat{C}_i|^{-\frac{1}{2}} \exp\left\{-\frac{1}{2}(x_j - \hat{m}_i)^T \hat{C}_i^{-1}(x_j - \hat{m}_i)\right\} P(\omega_i) \qquad (4.50)$$

Hipótesis 2. Equiprobabilidad de todas las clases a priori. Se supone que todas las clases tienen a priori el mismo valor de $P(T_i)$. Dado el desconocimiento inicial del tipo de fondo de la imagen y de la persona a analizar se puede asumir esta hipótesis. Aplicando logaritmos neperianos a la expresión anterior y tras una sencilla manipulación queda:

$$fd(\omega_i | x_j, \hat{\theta}_i) = \frac{1}{2}\ln|\hat{C}_i| + \frac{1}{2}(x_j - \hat{m}_i)^T \hat{C}_i^{-1}(x_j - \hat{m}_i) \qquad (4.51)$$

que se corresponde con lo que se conoce como distancia de Mahalanobis:

$$d_M(x_j,\hat{\theta}_i) = (x_j - \hat{m}_i)^T \hat{C}_i^{-1}(x_j - \hat{m}_i) \tag{4.52}$$

afectada por el elemento corrector: $\frac{1}{2}\ln|\hat{C}_i|$.

Hipótesis 3. Igualdad en las matrices de covarianza. Consiste en considerar que las matrices de covarianza de todas las clases son iguales ($C_1=C_2=...=C_K=C$). Esta hipótesis es asumible ya que lo que se pretende es particionar el espacio de color en los principales colores presentes en la imagen, entre los que se encuentra el de la piel, y una buena estimación de ellos será considerar que todos tienen la misma covarianza. No hay que olvidar que la percepción de un color depende de la resolución con la que se trabaje. En este caso se puede eliminar el elemento corrector ya que será el mismo en todas ellas y la función discriminante quedará:

$$fd(\omega_i|x_j,\hat{\theta}_i) = (x_j - \hat{m}_i)^T \hat{C}^{-1}(x_j - \hat{m}_i) \tag{4.53}$$

donde se ha eliminado el factor de proporcionalidad 1/2. Obsérvese que, bajo esta hipótesis, sí se corresponde con la distancia de Mahalanobis.

Hipótesis 4. Igualdad de varianzas para todas las componentes de las clases e independencia entre ellas. Consiste en asumir que todas las clases tienen la misma matriz de covarianza y además esta matriz es de la forma:

$$C_1 = C_2 = ... = C_K = \begin{bmatrix} s^2 & 0 \\ 0 & s^2 \end{bmatrix} = s^2 I \tag{4.54}$$

En el apartado 4.2. se ha demostrado que la dependencia entre las componentes de color "r" y "g" son muy bajas y que las diferencias entre las varianzas para "r" y "g" son también pequeñas. Bajo esta hipótesis, la función discriminante será la distancia euclídea entre la muestra y el centro de la clase:

$$fd(\omega_i|x_j,\hat{\theta}_i) = (x_j - \hat{m}_i)^T(x_j - \hat{m}_i) = ||x_j - \hat{m}_i||^2 \tag{4.55}$$

De esta forma el problema de modelado estocástico del histograma mediante funciones gaussianas queda reducido a un problema de aproximación del mismo mediante vectores usando distancia

euclídea; donde hay que estimar los estadísticos ($\hat{m}_i$) de cada clase y el número de clases o colores ($\hat{K}$) presentes en una imagen. Aunque la precisión de la segmentación disminuye al aplicar las hipótesis propuestas, es suficiente para detectar los principales colores de una imagen y, por lo tanto, dar una buena estimación del color de la piel.

Para estimar ($\hat{m}_i$) se aplica aprendizaje competitivo local basado en distancia euclídea y empleando el método VQ (Vector Quantization) propuesto por Kohonen [Kohonen, 97]. Para paliar el problema del aprendizaje local se emplea un método de inicialización basado en un histograma aproximado. Para estimar el número de clases ($\hat{K}$) se evalúa el grado de ajuste de las clases a probar con la topología de la distribución de colores, mediante una función de coste que intenta minimizar la desviación interna entre los pixels pertenecientes a una misma clase y maximizar la distancia entre las distintas clases. Se prueban un número de clases entre dos y un máximo $K_{máx}$ introducido por el usuario, de forma que el valor de K que dé un máximo en la función de coste será el número de colores estimado de la imagen ($\hat{K}$). Las posiciones de los centroides de las clases representarán las medias estimadas de las mismas $\hat{m} = (\hat{m}_1, \hat{m}_2, ..., \hat{m}_K)$.

Entre estas clases se localiza la clase piel, calculando la distancia a un patrón, y se aplica sobre ella un modelo gaussiano. Se segmentan como piel todos aquellos pixels para los que la función densidad de probabilidad del modelo sea un valor mayor que un umbral adaptativo (Th). Por último, se realiza una adaptación del modelo mediante la estimación de sus parámetros usando una combinación lineal de los ya conocidos y empleando el criterio de máxima probabilidad. En la figura 4.11 se muestra un esquema con las distintas fases del proceso.

A continuación se describirán cada una de las fases del proceso de clustering.

4.5.1. Localización inicial

El objetivo de esta fase es dar una estimación inicial de los vectores media de las clases que mejor aproximan las distribuciones de color del histograma, para a partir de estas posiciones aplicar un algoritmo de aprendizaje competitivo local. Este tipo de aprendizaje implica una dependencia de las posiciones finales de los vectores con la estimación inicial, de forma que si ésta no es buena se puede

caer en mínimos locales o puede haber vectores que "no aprendan", es decir, que no modifiquen su posición.

Existen tres formas básicas de inicializar un algoritmo con aprendizaje no supervisado:

1.- Aleatoria. Las posiciones iniciales estimadas de los vectores se sitúan en el espacio "rg" de forma aleatoria:

$$\hat{m}_i^{(0)} = (\text{rand}(r), \text{rand}(g)) \quad i = 1,2,...,K \quad \text{rand}(c) \in [\min(c), \max(c)] \tag{4.56}$$

Puede caer en mínimos locales si los valores iniciales se separan mucho de los valores óptimos. Al situarse los vectores de forma aleatoria puede ocurrir que un vector se sitúe en una zona alejada de los datos donde "no aprende" y, en consecuencia, es como si no estuviera.

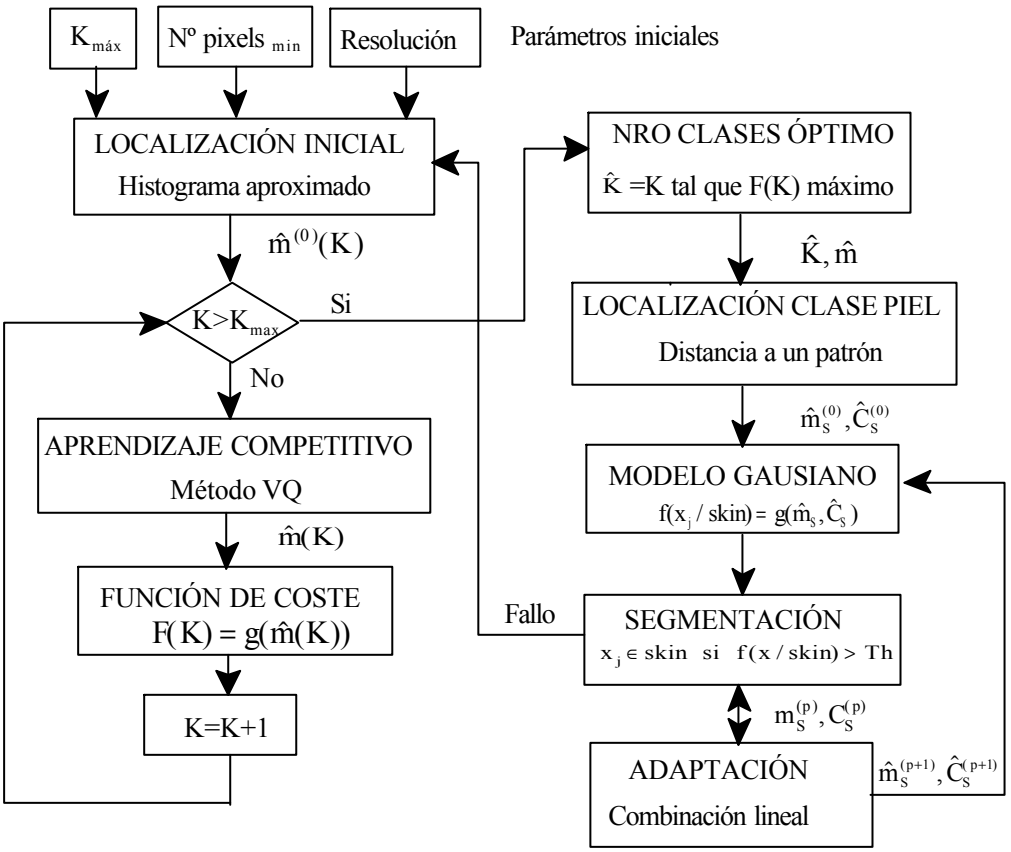


Figura 4.11. Fases del algoritmo UASGM

2.- Equiespaciada. Los vectores se distribuyen en el espacio de datos de entrada de forma uniforme.

$$\hat{m}_i^{(0)} = \left(\left(\min(r) + (i-1)\left(\frac{\max(r) - \min(r)}{K} \right) \right), \left(\min(g) + (i-1)\left(\frac{\max(g) - \min(g)}{K} \right) \right) \right) \qquad (4.57)$$
$$i = 1,2,\ldots,K$$

Se asegura que queda evaluado todo el espacio, pero si hay muchos datos en una zona y pocos en otra, puede caer en mínimos locales. Puede ocurrir que un vector caiga en una zona donde no hay muestras y por lo tanto "no aprenda", ya que la forma de las distribuciones de color en los histogramas son impredecibles.

3.- *Quasiespaciadas*. Los vectores se distribuyen de forma aleatoria en zonas equiespaciadas. Este caso es una mezcla de los dos anteriores pero no soluciona sus problemas.

El método propuesto consiste en hacer una estimación inicial de los vectores empleando un histograma aproximado del espacio "rg". La idea consiste en situar los vectores en las posiciones con mayores concentraciones de pixels en el histograma aproximado. Con ello se consigue inicializar los vectores en las posiciones de los colores más probables a una baja resolución. Como ya se ha visto, los colores producen distribuciones Gaussianas en el histograma a la resolución con la que se está trabajando (256x256 colores); sin embargo estas distribuciones se pueden transformar en deltas eligiendo una resolución menor de NxN cromaticidades. De esta forma se evitan los problemas de mínimos locales planteados.

El usuario puede elegir la resolución que desea del histograma aproximado, H, especificando el número de acumuladores, N, para cada componente de color. Por lo tanto, cada componente se divide en N intervalos de tamaño S igual al rango dinámico del componente de color, P, dividido por el número de acumuladores deseados,N, (S=P/N). Es decir, el histograma aproximado será una matriz de NxN, donde cada acumulador es inicializado a cero. Para cada pixel de la imagen, $\mathbf{x}=(x_r, x_g)$ se tendrá:

$$H(f_{acum}(x)) = H(f_{acum}(x)) + 1 \qquad (4.58)$$

$$f_{acum}(x) = (truc(\frac{x_r}{S}), truc(\frac{x_g}{S})) \qquad (4.59)$$

donde truc( ) indica el valor entero más próximo por defecto de la división.

La aproximación del histograma implica tener una resolución de colores de tamaño SxS, de manera que colores que estén separados en el histograma menos de este valor caerán en el mismo acumulador y serán considerados un único color. De esta forma se obtienen los principales colores de la imagen. Para esta aplicación se ha utilizado una matriz de 50x50 acumuladores que, para un rango dinámico de las componentes de color de 1 (r,g 0 [0,1]), da una resolución de 0.02, suficiente para nuestros propósitos ya que supone detectar 2500 cromaticidades diferentes independientes de la luminancia. Por otro lado, los colores de la clase piel tienen unas varianzas evaluadas por exceso de 0.0175, por lo que esta clase caerá en su mayor parte en una celdilla. Se han evaluado las varianzas de color de otros objetos y se han obtenido valores como éstos o menores.

El algoritmo sitúa las posiciones iniciales de los K vectores a ser evaluados en las posiciones de los K mayores acumuladores del histograma aproximado. Por supuesto que estas posiciones no son los centros de las distribuciones de color pero dan una buena aproximación de las mismas con un error máximo de ±P/2N.

Un problema que puede aparecer es el de la dispersión de una distribución de color en varios acumuladores, dependiendo de donde caiga la media de la distribución. En la figura 4.12 (a) se aprecia el citado efecto, en el caso de que mayoritariamente las distribuciones caigan en un acumulador y por lo tanto el efecto de dispersión es despreciable.

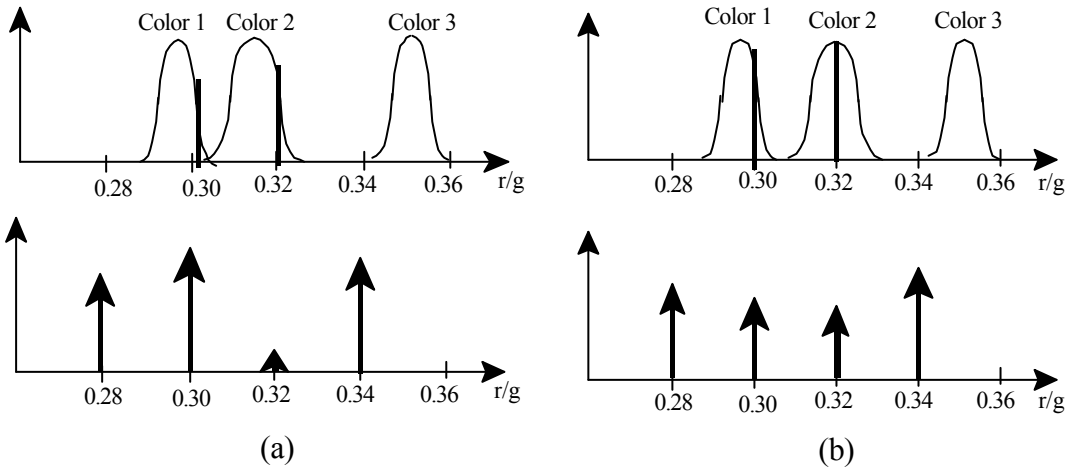
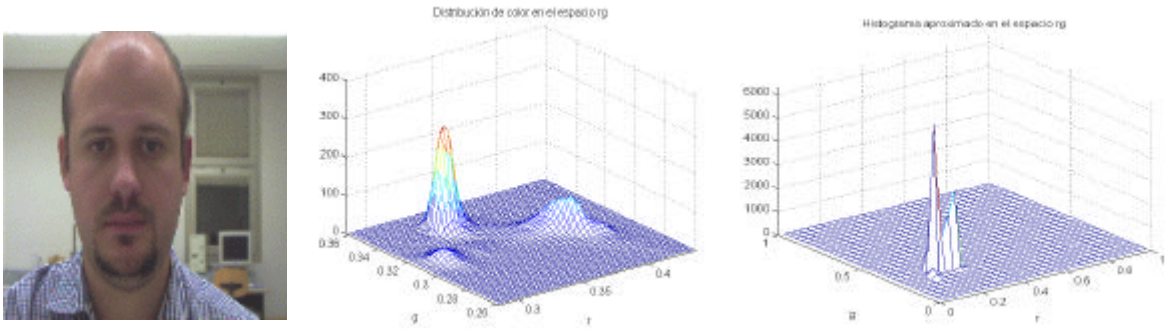


Figura 4.12. Efecto de dispersión de distribuciones en varios acumuladores

En la figura 4.12 (b) se observa una gran dispersión del Color 2 que ocasiona que se distribuya entre dos acumuladores. No obstante, este efecto se corrige en la fase de entrenamiento y determinación del número óptimo de objetos, ya que entonces un número de vectores igual al de principales distribuciones de color se colocarán en los centros reales de las gaussianas.

En la figura 4.13. se presenta una imagen en color a analizar, el histograma en el espacio "rg" y el histograma aproximado de la misma. En los acumuladores con mayor número de pixels (máximos del histograma aproximado) se colocan inicialmente los vectores a ser entrenados mediante aprendizaje competitivo.



(a) (b) (c)

Figura 4.13. Ejemplo de obtención del histograma aproximado. (a) Imagen original, (b) Histograma en el espacio "rg", (c) Histograma aproximado

4.5.2. Aprendizaje competitivo

Para estimar los vectores media de las clases del modelo se utiliza un algoritmo de aprendizaje competitivo basado en distancia euclídea propuesto por Kohonen y llamado VQ ("Vector Quantization"). Este algoritmo utiliza un conjunto reducido de vectores (llamados neuronas) para modelar una gran cantidad de muestras. En esta aplicación, las muestras son los valores de las componentes normalizadas de color de los pixels de una imagen, y las neuronas estiman las posiciones de las medias de las funciones Gaussianas que modelan el histograma. Para aplicar el algoritmo VQ es necesario conocer el número de vectores de la aproximación (K) y una estimación de las posiciones iniciales de estos vectores. El primer parámetro será evaluado entre un mínimo de dos (ya que se

supone que al menos en la imagen habrá una cara y un fondo) y un máximo introducido por el usuario ($K_{máx}$). Para cada valor de K se realiza el aprendizaje y posteriormente se evalúa el "clustering" al que da lugar mediante una función de coste.

Las posiciones iniciales de los vectores son muy importantes en este método ya que pueden influir grandemente en la exactitud del resultado final. En [González et al., 95] y [Karayiannis et at., 96] se presenta una discusión sobre este tema. A. I. Gonzalez destaca la dependencia del algoritmo VQ (también llamado SCL(Simple Competitive Learning)) con las posiciones iniciales de los vectores, ya que para cada muestra de entrada únicamente se ajusta la neurona ganadora lo que hace que el aprendizaje sea muy local. Analiza una mejora del algoritmo conocida como GLVQ (Generalized Learning Vector Quantization), donde para cada muestra de entrada se mueven todas las neuronas y no sólo la ganadora, lo que hace que el aprendizaje sea más general que en el caso anterior y, por lo tanto, lo hace insensible a las posiciones iniciales de los vectores. Sin embargo, en dicho artículo se ponen de manifiesto algunos problemas de este método, como son:

1.- Inconsistencia de las reglas de aprendizaje cuando el espacio de muestras de entrada es pequeño (como es el caso que nos ocupa), provocando fluctuaciones en el aprendizaje.

2.- Sensibilidad a los cambios de escala de los datos de entrada.

3.- Si el número de vectores es grande (K>10) o el espacio de las muestras de entrada es también grande (por ej. [-3, 6]) este algoritmo se comporta como el SCL.

N. B. Karayiannis también destaca los problemas del algoritmo GLVQ y presenta una modificación Fuzzy del GLVQ que llama GLVQ-F y que soluciona sus problemas. La modificación consiste en introducir la fórmula FCM (Fuzzy C-Means) para controlar el movimiento de los vectores en el aprendizaje. Con ello se consigue un aprendizaje independiente de las posiciones iniciales de los vectores, insensible a los cambios de escala y que funciona para cualquier rango dinámico de las variables de entrada.

En nuestro caso, los problemas de aprendizaje local del método SCL se solventan con la inicialización empleada, que asegura una buena estimación para las neuronas. Con el objeto de aplicar el algoritmo óptimo para la aplicación, también se ha programado el método GLVQ-F y se han comparado los

resultados obtenidos con los dos métodos (ver punto 4.6.), concluyendo que, con el algoritmo propuesto, se obtienen mejores resultados y con menor coste computacional que con el GLVQ-F.

A continuación, se pasará a describir los fundamentos teóricos del algoritmo VQ empleado.

Dados los colores normalizados de los pixels de una imagen $\mathbf{X}=\{\mathbf{x}_1,\mathbf{x}_2,...,\mathbf{x}_N\}$ y un número de vectores media situados en las posiciones definidas por $\hat{m}$:

$$\hat{m} = (\hat{m}_1, \hat{m}_2, ..., \hat{m}_K) \quad ; \quad \hat{m}_i = (\hat{m}_{ir}, \hat{m}_{ig}) \tag{4.60}$$

El método VQ da la mejor aproximación de la función densidad de probabilidad, f($\mathbf{x}$), de la variable estocástica $\mathbf{x}$ 0 U^2 usando un número finito de vectores $\hat{m}$, llamados neuronas. Para ello utiliza un sistema de dos capas: una capa de entrada y una capa competitiva, como se puede ver en la figura 4.14.

El índice i se obtiene de forma implícita por un proceso de decisión de la forma:

$$i = \arg\min_k \{||x - \hat{m}_k||\} \tag{4.61}$$

donde la norma es euclídea.

Para calcular la mejor aproximación de **X,** Kohonen define una función de error cuadrático medio de cuantificación, como la mostrada en la ecuación 4.62. El mínimo de esta función dará el conjunto de vectores $\hat{m}$ que mejor aproximan a **X**. Para la búsqueda del mínimo se emplea la técnica de descenso por el gradiente.

$$E = \int ||x - \hat{m}_i||^2 f(x)dx \tag{4.62}$$

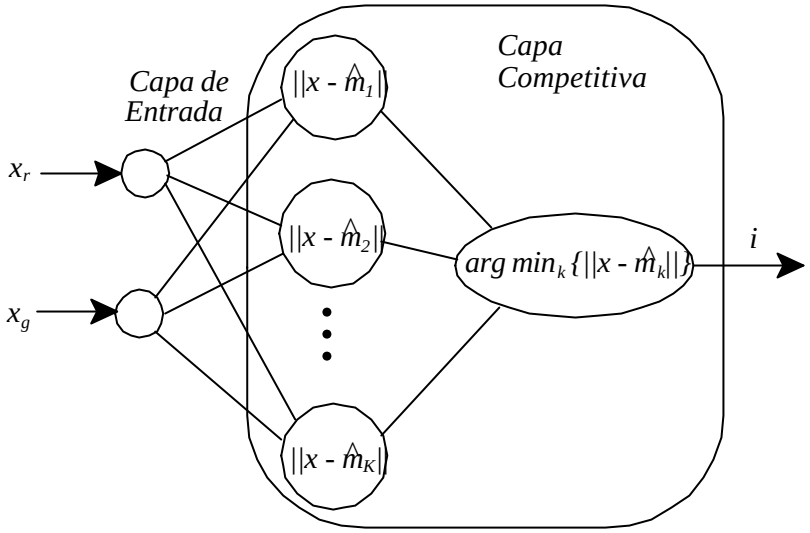


Figura 4.14. Aprendizaje competitivo

Obsérvese que el subíndice i es una función de **x** y de todos los $\hat{m}_k$. El gradiente de E con respecto a $\hat{m}_k$, k=1,2,...,K, no se puede calcular fácilmente ya que si $\hat{m}_k$ varía, el índice i puede tener discontinuidades (paso de un valor discreto a otro debido a cambios de la neurona ganadora) y en estos puntos no existe derivada. Por lo tanto, el cálculo de la derivada de E con respecto a $\hat{m}_k$ no tiene una solución para cualquier f(**x**), pero introduciendo la restricción de que f(**x**) sea continua y definiendo dos teoremas se puede llegar a definir:

1.- El problema causado por la discontinuidad de i puede ser acotado si se cumple la siguiente identidad:

$$\min_k(a_k) = \lim_{\square} \left[\sum a_k^r \right]^{\frac{1}{r}} \tag{4.63}$$

Donde a_k es un conjunto de números reales positivos

2.- Dada una función:

$$q(x,r) = (1+|x|^r)^{\frac{1}{r}} \tag{4.64}$$

excluyendo los valores de x en los cuales g o $\lim_{r64} g$ no son derivables, por ejemplo x 0 {-1,0,1}, se cumple:

$$\lim_{r6\&4} \frac{Mg}{Mx} = \frac{M}{Mx}\left(\lim_{r6\&4} g\right) \tag{4.65}$$

Con las restricciones dadas es posible aplicar el gradiente $\nabla_{\hat{m}_j} E$, ya que ahora una función de la forma $\left(\sum_k ||x - \hat{m}_k||^r\right)^{\frac{2}{r}}$ es continua, simplemente evaluada y continuamente diferenciable en sus argumentos, excepto cuando **x** sea exactamente igual a algún $\hat{m}_k$. Se han excluído todos los casos en los que uno de los términos $||x - \hat{m}_i||^2$ es exactamente igual a la suma de los otros términos, aplicando el teorema 2. Considerando una variable estocástica **x** con f(**x**) continua, ahora todos los casos singulares tienen probabilidad cero. Bajo estas condiciones el gradiente puede ser calculado como sigue:

$$||x - \hat{m}_i||^2 = \left(\min_k\left(||x - \hat{m}_k||\right)\right)^2 = \lim_{r\to-\infty}\left(\sum_k ||x - \hat{m}_k||^r\right)^{\frac{2}{r}} \tag{4.66}$$

llamando:

$$\nabla_{\hat{m}_j} E = \int \lim_{r\to-\infty} \nabla_{\hat{m}_j}\left(\sum_k ||x - \hat{m}_k||^r\right)^{\frac{2}{r}} f(x)dx \tag{4.67}$$

$$\sum_k ||x - \hat{m}_k||^r = A \tag{4.68}$$

se puede hacer la siguiente igualdad:

$$\nabla_{\hat{m}_j} A^{\frac{2}{r}} = \frac{2}{r} A^{(\frac{2}{r}-1)} \nabla_{\hat{m}_j}\left(||x - \hat{m}_j||^r\right) = \frac{2}{r} A^{(\frac{2}{r}-1)} \nabla_{\hat{m}_j}\left(||x - \hat{m}_j||^2\right)^{\frac{r}{2}} \tag{4.69}$$

Realizando operaciones simples se obtiene:

$$\nabla_{\hat{m}_j} A^{\frac{2}{r}} = -2\left(A^{\frac{2}{r}}\right)\left(\frac{\left(||x - \hat{m}_j||^2\right)^{\left(\frac{r}{2}-1\right)}}{A}\right)\left(x - \hat{m}_j\right) \tag{4.70}$$

De la ecuación (4.68) se tiene:

$$\lim_{r\to-\infty} A^{\frac{r}{2}} = ||x - \hat{m}_i||^2 \tag{4.71}$$

llamando:

$$B = \frac{\left(||x - \hat{m}_j||^2\right)^{\left(\frac{r}{2}-1\right)}}{A} = \frac{||x - \hat{m}_j||^r}{\sum_k ||x - m_k||^r} ||x - \hat{m}_j||^{-2} = \left(\sum_k \frac{||x - \hat{m}_k||^r}{||x - \hat{m}_j||^r}\right)^{-1} ||x - \hat{m}_j||^{-2} \tag{4.72}$$

Obsérvese que cuando r 6-4 el término $||x - \hat{m}_k||^r / ||x - \hat{m}_j||^r$ para un valor dado de $\hat{m}_j$, es máximo cuando $\hat{m}_j = \hat{m}_k$ y predomina sobre los otros términos. Por lo tanto:

$$\lim_{r\to-\infty} B = \lim_{r\to-\infty}\left(\frac{||x - \hat{m}_j||}{||x - \hat{m}_i||}\right)^r ||x - m_j||^{-2} = \delta_{ij}||x - m_j||^{-2} \tag{4.73}$$

Donde δ_{ij} es la delta de Kronecker (=1 para i=j y 0 resto). Combinando los resultados parciales se obtiene:

$$\lim_{r\to-\infty} \nabla_{\hat{m}_j} A^{\frac{2}{r}} = -2||x - \hat{m}_i||^2 \delta_{ij}||x - \hat{m}_j||^{-2}\left(x - \hat{m}_j\right) = -2\delta_{ij}\left(x - \hat{m}_j\right) \tag{4.74}$$

y

$$\nabla_{\hat{m}_j} E = \int \lim_{r\to-\infty} \nabla_{\hat{m}_j} A^{\frac{2}{r}} f(x)dx = -2\int \delta_{ij}\left(x - \hat{m}_j\right) f(x)dx \tag{4.75}$$

donde en un instante t el gradiente será:

$$\nabla_{\hat{m}_j} E(t) = -2\delta_{ij}\left[x(t) - m_j(t)\right] \tag{4.76}$$

de forma que aplicando descenso por el gradiente y haciendo un cambio de índice, habrá que mover la variable $\hat{m}_i$ en la dirección de $-\nabla_{\hat{m}_i} E(t)$ con lo que queda:

$$\hat{m}_i(t+1) = \hat{m}_i(t) + \alpha(t)[x(t) - \hat{m}_i(t)] \tag{4.77}$$

Aquí, "(t) indica el paso de aprendizaje que ya incluye la constante -2. Para que el sistema sea estable

tiene que estar comprendido entre (0,1). Existen dos alternativas: que sea fijo o que vaya disminuyendo en función del tiempo.

Se toma un subconjunto de entrenamiento $\mathbf{X_L}=\{x_1, x_2,...,x_L\}$ de entre las muestras de entrada $\mathbf{X}$ de forma aleatoria, y para cada muestra $\mathbf{x_j}$, se calcula su distancia euclídea con los vectores $\hat{m}$, de forma que para el que se obtenga la mínima distancia será el ganador. El vector ganador se mueve una distancia proporcional a la distancia que separa el pixel del vector. Este proceso se repite hasta que los vectores se muevan menos que un umbral definido a priori por el usuario. La cantidad que se mueven los pixels es controlada por el parámetro "(t) y se decrementa con el tiempo. Por lo tanto, el método consta de los siguientes pasos:

1.- Dado un subconjunto de muestras $\mathbf{X_L}$ 0 U^2 se toma un conjunto de vectores $\hat{m}$ con un número de componentes K entre 2 y K_{max}, un parámetro T que indica el número máximo de pasos hasta que el sistema converja (calculado de forma experimental) y un error máximo de convergencia ,.

2.- Se inicializa $\hat{m} = \{\hat{m}_1^{(0)}, \hat{m}_2^{(0)}, ..., \hat{m}_L^{(0)}\}$ 0 U^2 empleando el histograma aproximado

Se inicializa "$_0$, (0,1)

Para t=1,2,..., T

Para j=1,2,...,L

a.- Se encuentra $i = \arg\min_k \{||x_j - \hat{m}_k||\}$

b.- Se actualiza el ganador

$$\hat{m}_i(t) = \hat{m}_i(t-1) + \alpha(t-1)[x(t) - \hat{m}_i(t-1)]$$

Nuevo j

Se computa:

$$E(t) = ||V(t) - V(t-1)|| = \sum_{j=1}^{L}\sum_{k=1}^{K} |\hat{m}_{kj}(t) - \hat{m}_{kj}(t-1)|$$

Si E(t) #0 se para;

Si no "(t)="(t-1) (1 - t/T)

Nuevo t.

Se puede dar una simple interpretación geométrica del aprendizaje como la que aparece en la figura 4.15.,consistente en considerar que la neurona ganadora se moverá hacia el dato de entrada, $\mathbf{x_j}$, sobre

el vector $(x_j(t) - \hat{m}_i(t-1))$ una cantidad proporcional a la diferencia entre el dato de entrada y la posición de la neurona ganadora.

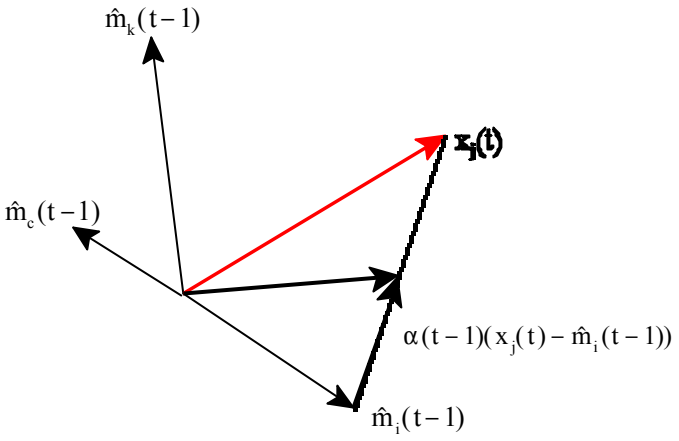


Figura 4.15. Interpretación geométrica del aprendizaje

Las objeciones que se pueden poner a este tipo de aprendizaje son que considera que las componentes de color son independientes y además variantes a los cambios de escala y transformaciones de las componentes de color. Sin embargo, en nuestro caso, se puede asumir independencia de las componentes "r" y "g" y, por otro lado, las variables de color tienen un margen dinámico fijo entre [0,1] y no se van a realizar transformaciones geométricas sobre ellas.

En la figura 4.16 se muestran las posiciones de las neuronas sin entrenar (círculos) y de las neuronas después del entrenamiento (cruces) para un número comprendido entre 2 y 5 y para el histograma de la figura 4.13. Como se puede observar los vectores se distribuyen sobre las zonas de mayor densidad de pixels. Para el entrenamiento se han utilizado un 10% de los pixels de la imagen ya que se ha demostrado experimentalmente que los resultados eran similares a los obtenidos con un número mayor de muestras y, sin embargo, el tiempo de cálculo se reducía grandemente. Para demostrar esta afirmación en la tabla 4.8 se muestran las posiciones de las neuronas para un número de 2, 3 y 4 y para una cantidad de pixels del 100%, 10% y 1%. Se observa que las diferencias se encuentran en la tercera cifra decimal.

$\hat{m}_i^T = \begin{bmatrix} \hat{m}_{ir} \\ \hat{m}_{ig} \end{bmatrix}$	2 Neuronas	3 Neuronas	4 Neuronas
100%	0,2253 0,3052 0,3209 0,3252	0,3054 0,1896 0,2276 0,3252 0,2709 0,3239	0,3132 0,2885 0,1896 0,2272 0,3246 0,3256 0,2708 0,3239
10%	0,2253 0,3044 0,3205 0,3244	0,3046 0,1893 0,2277 0,3244 0,2712 0,3236	0,3128 0,2881 0,1893 0,2273 0,3241 0,3250 0,2712 0,3235
1%	0,2250 0,3033 0,3200 0,3248	0,3042 0,1890 0,2281 0,3245 0,2702 0,3234	0,3100 0,2800 0,1890 0,2253 0,3241 0,3295 0,2702 0,3224

Tabla 4.8. Posiciones de las neuronas en función del número de muestras

Una vez explicados los fundamentos teóricos del algoritmo VQ, basado en *aprendizaje local*, se va a hacer lo propio con el algoritmo GLVQ-F, basado en *aprendizaje global*, para posteriormente hacer una comparación entre ellos.

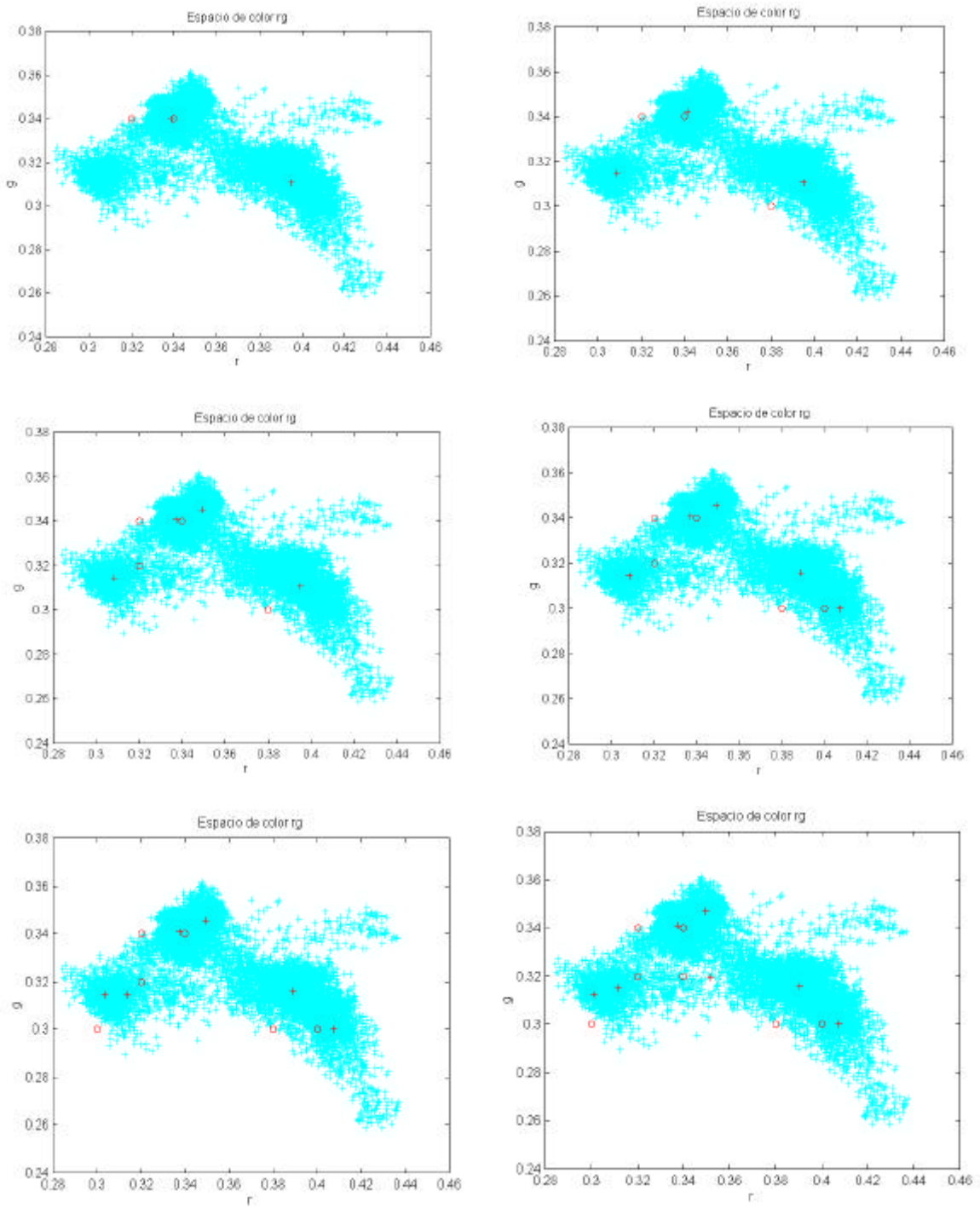


Figura 4.16. Aprendizaje para un número de neuronas entre 2 y 7 y un 10% de muestras.

Sea el conjunto finito de pixels normalizados de una imagen $\mathbf{X}=\{\mathbf{x_1},\mathbf{x_2},...,\mathbf{x_N}\}$ tal que $\mathbf{x_c}$ 0 U^2, sean $\hat{m} = (\hat{m}_1, \hat{m}_2, \ldots, \hat{m}_K)$ los vectores a estimar, sea i el vector ganador en un proceso competitivo, sea $L_{\mathbf{x}}$ la función que mide el error de posición entre $\mathbf{x_c}$ y los vectores, ponderados en función de la distancia al pixel. El algoritmo GLVQ utiliza la siguiente función de error:

$$L_x = L(x_c; \hat{m}_1, \ldots, \hat{m}_K) = \sum_{r=1}^{K} g_r ||x_c - \hat{m}_r||^2 \tag{4.78}$$

$$g_r = \begin{cases} 1 & \text{si} \quad i = \arg\min_r \{||x_c - \hat{m}_r||\} \\ \dfrac{1}{D} & \text{para el resto} \end{cases} \tag{4.79}$$

$$D = \sum_{r=1}^{K} ||x_c - \hat{m}_r||^2 \tag{4.80}$$

La neurona más cercana a la muestra debe tener una aportación en la función de error de g_r=1, tal que r=i, mientras que el resto de neuronas tienen la misma aportación, g_r=1/D tal que r...i, inferior a la primera. Sin embargo, como se puede comprobar, si D<1 se producirá un efecto contrario al deseado. Este es el problema del algoritmo GLVQ. Para solucionarlo hay que cambiar la definición de los pesos $\{g_r\}$ en la función a optimizar. Las propiedades que deben tener dichos pesos para que se puedan aplicar sin problemas son:

1) La magnitud de g_r debe ser inversamente proporcional a $||x_c - \hat{m}_r||$
2) Cada valor de g_r debe estar comprendido entre [0 ,1]
3) La suma de los pesos $\{g_r\}$ debe ser 1.

Existen diferentes maneras de definir $\{g_r\}$ de forma que satisfagan las propiedades descritas [Karayiannis et al.,96]. Una muy utilizada consiste en aplicar la fórmula FCM (Fuzzy C-Means) (Ecuación (4.81)), ya que es un método muy robusto para realizar clustering, donde se debe asumir que n>1.

$$u_r = \left(\sum_{j=1}^{K} \left(\frac{||x_c - \hat{m}_r||^{\frac{2}{(n-1)}}}{||x_c - \hat{m}_j||^{\frac{2}{(n-1)}}} \right) \right)^{-1} \qquad r = 1,2,\ldots,K \tag{4.81}$$

Por lo tanto, sustituyendo esta familia de pesos en la ecuación (4.78) se tiene:

$$L_x = L(x_c; \hat{m}_1, \ldots, \hat{m}_K) = \sum_{r=1}^{K} u_r ||x_c - \hat{m}_r||^2 =$$
$$= \sum_{r=1}^{K} \left(\sum_{j=1}^{K} \left(\frac{||x_c - \hat{m}_r||^{\frac{2}{(n-1)}}}{||x_c - \hat{m}_j||^{\frac{2}{(n-1)}}} \right) \right)^{-1} ||x_c - \hat{m}_r||^2 \tag{4.82}$$

Assumiendo que $||x_c - \hat{m}_j|| > 0$ y n>1, se tiene que el gradiente de L_x con respecto a $\hat{m}_k$ será (ver Apéndice 2):

$$\nabla_{\hat{m}_k} L = \nabla_{\hat{m}_k} \left(\sum_{r=1}^{K} u_r ||x_c - \hat{m}_r||^2 \right) = \left(\frac{-2}{n-1} \right) (x_c - \hat{m}_k) \bullet$$
$$\left[(n-2)u_k + \left(\sum_{r=1}^{K} \left(\frac{||x_c - \hat{m}_k||^{\frac{2}{n-1}}}{||x_c - \hat{m}_r||^{\frac{2}{n-1}}} \right)^{2-n} \right) u_k^2 \right] \tag{4.83}$$

Empleando la técnica iterativa de descenso por el gradiente, se tiene que, para un instante de tiempo t, la posición de los vectores vendrá dada por:

$$\hat{m}_j(t) = \hat{m}_j(t-1) + \left(\frac{2\alpha(t-1)}{n-1} \right) \left[(n-2)u_j + \left(\sum_{r=1}^{K} \left(\frac{||x_c - \hat{m}_j(t-1)||^{\frac{2}{n-1}}}{||x_c - \hat{m}_r(t-1)||^{\frac{2}{n-1}}} \right)^{2-n} \right) u_j^2 \right] (x_c - \hat{m}_j(t-1)) \tag{4.84}$$
$$j = 1,2,\ldots,K$$

Esta expresión da una familia de ecuaciones en función de n. Para el caso de n=2 se transformará en:

$$\hat{m}_j(t) = \hat{m}_j(t-1) + \alpha(t-1)(2Ku_j^2)(x_c - \hat{m}_j(t-1)) \qquad j = 1,2,\ldots,K \tag{4.85}$$

Experimentalmente se han obtenido óptimos resultados para n=2, minimizando la carga computacional. El factor 2K en el ratio de aprendizaje es irrelevante y, por ello, se puede quitar de la ecuación.

Un estudio que resulta interesante es ver la respuesta del sistema en función de n. Para ello se va a

calcular la ecuación (4.84) para los dos casos extremos que se pueden dar.

Consideremos el caso de n tendiendo a infinito, con lo que se obtiene (ver Anexo 2):

$$\lim_{n\to\infty}\{\hat{m}_j(t)\} = \hat{m}_j(t-1)+\left(\frac{2\alpha(t-1)}{K}\right)(x_c - \hat{m}_j(t-1)) \qquad j=1,2,...,K \tag{4.86}$$

En este caso el ratio de aprendizaje para los K vectores es el mismo para todos ellos.

En el caso de que n tienda a 1 se tiene (Ver Anexo 2):

$$\lim_{m\to 1}\{\hat{m}_j(t)\} = \begin{cases} \hat{m}_j(t-1)+(2\alpha(t-1))(x_c - \hat{m}_j(t-1)) & \text{si} \quad i = \arg\min_r\{\|x_c - \hat{m}_r\|\} \\ 0 & i \neq j \end{cases} \tag{4.87}$$

Como se puede observar, en este caso la fórmula es idéntica a la del método VQ, por lo que éste se puede considerar como un caso particular del GLVQ-F.

El parámetro n controla la velocidad de aprendizaje del sistema ya que afecta al ratio de aprendizaje. Así si n=1 el aprendizaje se realiza por extricta excitación del ganador; si n es mayor que 1 la excitación será diferente en función de los vectores y para un valor de n grande se produce una excitación igual para todas las neuronas.

En el punto 4.6. se hace una comparación entre el método VQ y el GLVQ-F, aplicados al problema planteado en esta tesis, para diferentes ejemplos.

4.5.3.- Factor de calidad del "clustering"

Mediante el proceso de aprendizaje anterior se obtiene la mejor aproximación del histograma con K vectores. A continuación se realiza una clasificación de los pixels en los vectores prototipos, empleando para ello el criterio de menor distancia euclídea a los mismos.

$$x_j \in \alpha_i \text{si } i = \arg\min\{||x_j - \hat{m}_k||\} \quad 1 \le k \le K; 1 \le j \le N \tag{4.88}$$

Se necesita un factor de calidad para evaluar el ajuste entre el modelo y el histograma, de forma que para un máximo de dicho factor se obtiene el número óptimo de clases del modelo y, por lo tanto, el "clustering" óptimo. Para ello se debe seguir el criterio de distribuir los pixels entre las clases, de manera que se maximice alguna medida de similitud interna de la clase y por lo tanto se maximice la divergencia entre las distintas clases.

Los vectores prototipos o valores medios de cada clase vendrán dados por la ecuación (4.89), donde M_k es el número de pixels que pertenecen a la clase k-ésima.

$$\hat{m}_k = \frac{1}{M_k}\sum_{i=1}^{M_k} x_i \qquad 1 \le k \le K \tag{4.89}$$

El vector patrón medio de todas las clases será:

$$\hat{m}_0 = \frac{1}{M}\sum_{i=1}^{M} x_i = \sum_{k=1}^{K} \hat{m}_k \tag{4.90}$$

Donde M indica el número total de pixels a clasificar.

Se conoce como matriz de dispersión ("scatter") de una clase k a la varianza media de los pixels pertenecientes a la clase con respecto al vector media de dicha clase y vendrá dada por:

$$S_k = \frac{1}{M_k}\sum_{i=1}^{M_k} [x_i - \hat{m}_k][x_i - \hat{m}_k]^T \tag{4.91}$$

La *matriz de dispersión intraclases* ("whitin-cluster scatter matrix") se define como el valor medio de las matrices de dispersión de cada clase k, siendo, por tanto:

$$S_W = \frac{1}{K}\sum_{k=1}^{K}\frac{1}{M_k}\sum_{i=1}^{M_k}[x_i - \hat{m}_k][x_i - \hat{m}_k]^T \quad (4.92)$$

Dicha matriz representa la varianza media de todos los elementos de las clases.

La *matriz de dispersión intergrupos* ("between-cluster scatter matrix") indica la varianza media de las medias de cada clase $(\hat{m}_i)$ con respecto al vector patrón medio de todas las clases $(\hat{m}_0)$, como se indica en la siguiente ecuación:

$$S_B = \frac{1}{K}\sum_{k=1}^{K}[\hat{m}_k - \hat{m}_0][\hat{m}_k - \hat{m}_0]^T \quad (4.93)$$

En esta última matriz todas las clases tienen el mismo peso a la hora de definir la varianza media con respecto al patrón, lo que ocasiona que clases con muy pocos pixels enmascaran el efecto de clases con un gran número de pixels. Como en esta aplicación lo que se pretende es localizar los principales colores de la imagen y dado que el ajuste sobre clases poco representativas supone la división del color de la piel en diferentes subcolores, se ha introducido una modificación sobre esta matriz, de manera que se pondera el peso de la varianza de la media de cada clase con respecto al patrón global mediante el número de pixels de la clase. Por lo tanto, se ha aplicado una matriz de dispersión intergrupos modificada que viene dada por:

$$S_B = \frac{1}{KM}\sum_{k=1}^{K}M_k[\hat{m}_k - \hat{m}_0][\hat{m}_k - \hat{m}_0]^T \quad (4.94)$$

Las *matrices de dispersión intraclases e interclases* dependen de cómo se distribuyan los pixels en las diferentes clases. Así, para clases homogeneas, la matriz S_W disminuye mientras que la matriz S_B aumenta ,ya que la varianza entre las clases es mayor. De donde resulta que para lograr la mayor similitud entre los pixels de una clase, como resultado de la minimización de la matriz S_W se obtiene la maximización de la matriz S_B y viceversa.

Por lo tanto, el objetivo que se busca consiste en incrementar el ratio con el cual se incrementa la varianza entre clases con respecto a la varianza interna de cada clase. Este problema fue estudiado por Ficher [Maravall 93] para el caso biclase con funciones univariables, dando lugar al conocido "ratio de Ficher" que se define como:

$$F = \frac{S_B}{S_W} \tag{4.95}$$

donde las matrices S_W y S_B son funciones y K=2. El problema radica en la generalización de este ratio para un caso multivariable, como el que nos ocupa, donde los pixels se definen por dos componentes de color. En este caso las matrices no son funciones, lo que lleva a plantearse: ¿qué se entiende por minimizar/maximizar una matriz? Uno de los criterios más utilizados consiste en maximizar la traza de $S_w^{-1} S_B$, según la ecuación (4.96). Este criterio se conoce como de *Hotelling* [Escudero, 77] o también como *ratio generalizado de Fisher* [Maravall 93]. Para un número de clases dado, K, la función de coste (F_K) será igual a la suma de los autovalores $\lambda_1,\lambda_2,...,\lambda_K$ de la matriz $S_w^{-1} S_B$.

$$F_K = \mathrm{tr}[S_W^{-1}S_B] = \begin{bmatrix} \lambda_1 & & \neq 0 \\ & \lambda_2 & \\ & \vdots \quad \vdots & \\ \neq 0 & & \lambda_K \end{bmatrix} = \sum_{i=1}^{K} \lambda_i \qquad ;1 \le K \le K_{max} \tag{4.96}$$

Por lo tanto, el número óptimo de clases será aquel que de un máximo en la función de coste (F). El éxito de utilizar este criterio depende de la forma de agrupación de los patrones a clasificar. Si estos patrones forman grupos compactos, están separados unos de otros y las características de cada patrón son independientes, los resultados obtenidos serán buenos. En el caso que nos ocupa se puede considerar que se dan las condiciones anteriores ya que las distribuciones de color tienen formas compactas, están relativamente separadas y la dependencia entre las características (r y g) es baja. Es decir, son prácticamente matrices diagonales.

Para probar el algoritmo propuesto se van a utilizar una serie de datos aparecidos en la referencia [Roberts 98][3], donde se hace un estudio comparativo de diferentes métodos de realizar "clustering" aplicando un enfoque estadístico bayesiano y que han sido explicados en el punto 4.4. Se hará una comparación de los resultados obtenidos con los distintos métodos.

Experimento 1. Los datos se generan mediante cuatro funciones gaussianas con la misma desviación

---

IEEE Transactions on Pattern Analysis and Machine Intelligence de Noviembre de 1998.

típica (F) y con las siguientes medias:

$$\begin{aligned} m_1 &= (0,0)^T \\ m_2 &= (2,\sqrt{12})^T \\ m_3 &= (4,0)^T \\ m_4 &= (-2,-\sqrt{12})^T \\ s &= \{1.2,1.0,0.66\} \end{aligned} \tag{4.97}$$

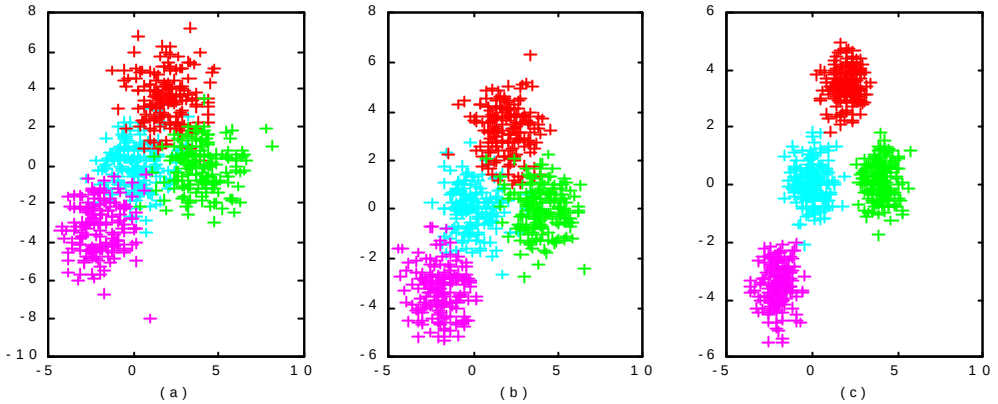


Figura 4.17. Datos para el experimento 1(a) F=1.2 (b) F=1.0 (c) F=0.66

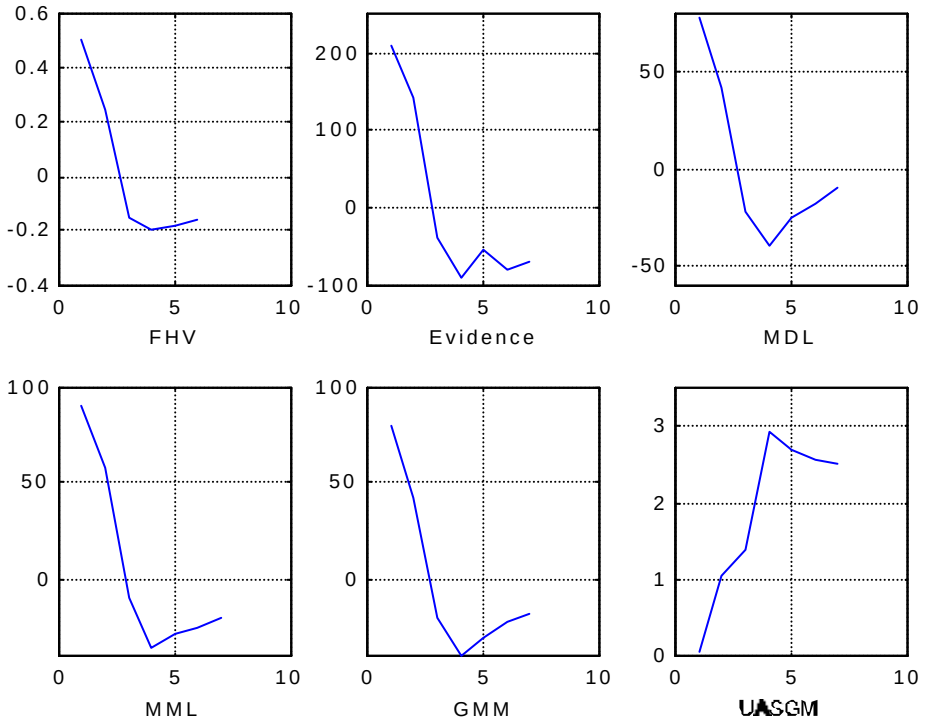


Figura 4.18. Resultados experimento 1 para F=0.66

Se toman 120 muestras por cada gaussiana, lo que hace un total de 480 muestras y se evalúan tres

casos diferentes de varianza (ver figura 4.17), presentando los resultados en las figuras 4.18, 4.19 y 4.20.

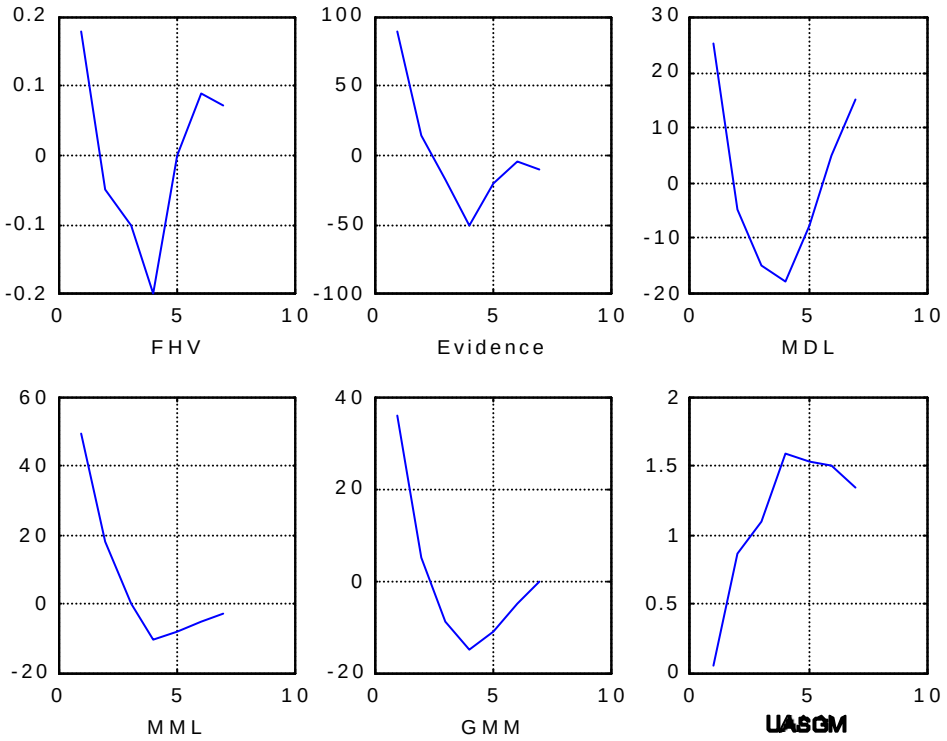


Figura 4.19. Resultados experimento 1 para F=1.0

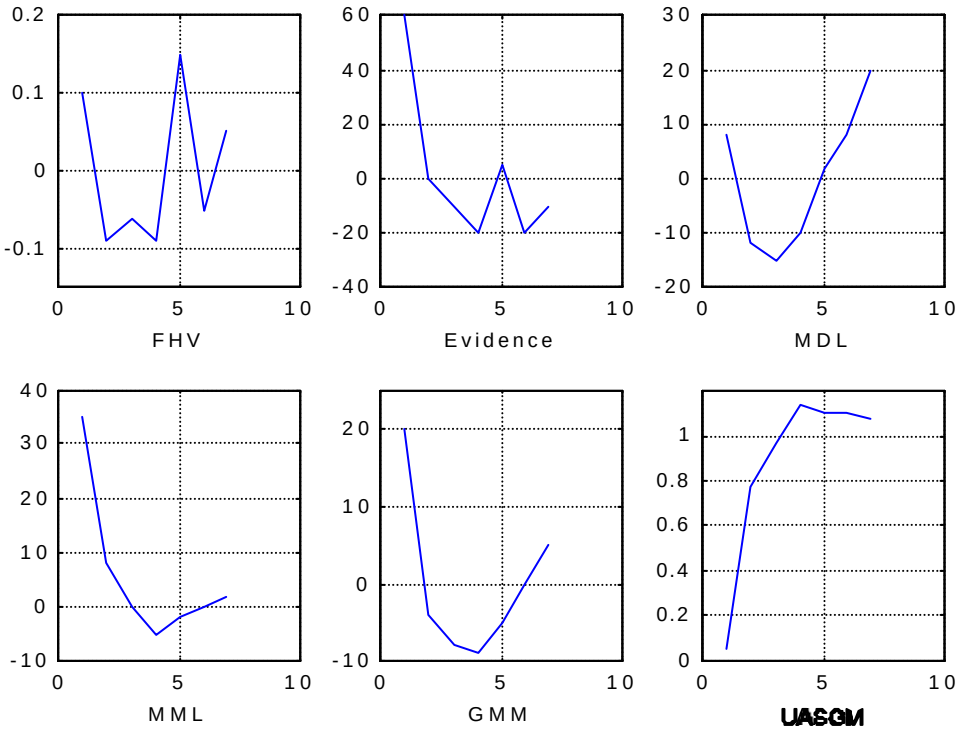


Figura 4.20. Resultados experimento 1 para F=1.2

Como se puede observar, para el caso de F=0.66 y F=1.0 todos los métodos dan un resultado correcto identificando como óptimas cuatro clases que se corresponden con las cuatro gaussianas del experimento. **Para el caso de F=1.2, únicamente los métodos MDL, MML, GMM y el propuesto en la tesis (UASGM) dan resultados correctos**. En la tabla 4.9 se muestran las posiciones finales de las neuronas para el método UASGM, donde se ve que difieren muy poco de las posiciones teóricas.

K=4	m_1	m_2	m_3	m_4
Ideal	0,0 0,0	2,0 3,4641	4,0 0,0	-2,0 -3,4641
F=1.2	0,0199 0,2093	2,1501 3,4871	4,1401 -0,0129	-2,0871 -3,6306
F=1	0,0817 0,1906	1,9003 3,3457	4,1108 0,0603	-1,8547 -3,2971
F=0.66	-0,0144 0,0514	2,0000 3,4977	4,0107 0,1548	-2,0502 -3,4848

Tabla 4.9. Posiciones de las neuronas para el experimento 1 y el método UASGM

Experimento 2. Los datos se obtienen a partir de cuatro funciones gaussianas que tienen dos a dos la misma media y la misma desviación típica. Se toman de 250 muestras de cada función (ver figura 4.21). Los resultados del experimento se muestran en la figura 4.22.

$$\begin{aligned} m_1 &= m_2 = (1,1)^T \\ m_3 &= m_4 = (-1,-1)^T \\ {}_1 &= s_3 = 1 \\ s_2 &= s_4 = 2 \end{aligned} \tag{4.98}$$

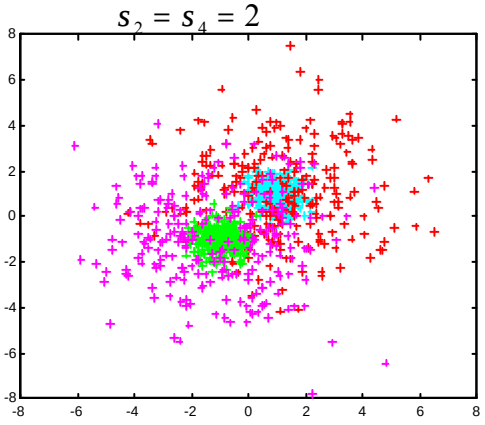


Figura 4.21. Datos para el experimento 2

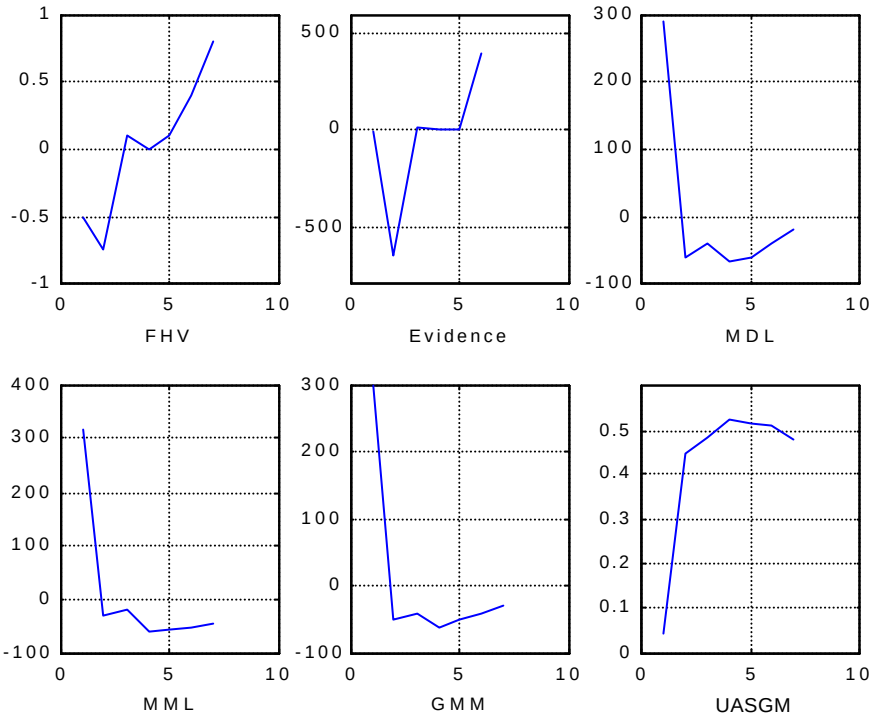


Figura 4.22. Resultados del experimento 2

o s métodos MML, MDL, GMM y el propuesto en la tesis dan un resultado correcto de cuatro clases óptimas. En la tabla 4.10. se muestran las posiciones de las neuronas y las covarianzas de cada clase para el método UASGM. Como se puede observar, a pesar de haber acertado el número de clases, las posiciones de los patrones no coinciden con las de las medias de las funciones gaussianas que los forman, sino que se ha producido una división de las muestras de entrada en cuatro grupos con covarianzas similares. No obstante, este experimento no es aplicable al problema planteado en esta tesis ya que no se pueden tener dos distribuciones distintas con la misma media, puesto que se estaría hablando del mismo color.

K=4	k_1	k_2	k_3	k_4
m_i	-2,0509 -2,2125	-1,7291 0,6772	1,5370 1,6886	0,8618 -1,2007
C_i	1,2094 -0,0182 -0,0182 1,2079	1,1217 -0,2855 -0,2855 1,2959	1,2799 -0,1143 -0,1143 1,1200	1,3619 -0,1982 -0,1982 1,0662

Tabla 4.10 Posiciones de las neuronas y covarianzas de las clases para el experimento 2

Experimento 3. "Iris Data". Los datos "Iris" de Anderson son muy conocidos en análisis de clasificadores y consisten en medidas de la morfología de las plantas. Están formados por 50 muestras de tres clases de datos: Iris Versicolor, Iris Virginica e Iris Setosa. Cada dato está formado por cuatro características. Se han tomado las características (x_2,x_3) (figura 4.23) y se han aplicado los distintos métodos, obteniéndose los resultados mostrados en la figura 4.24.

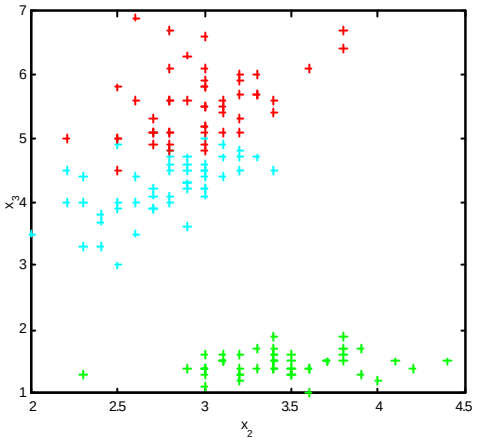


Figura 4.23. "Iris Data"

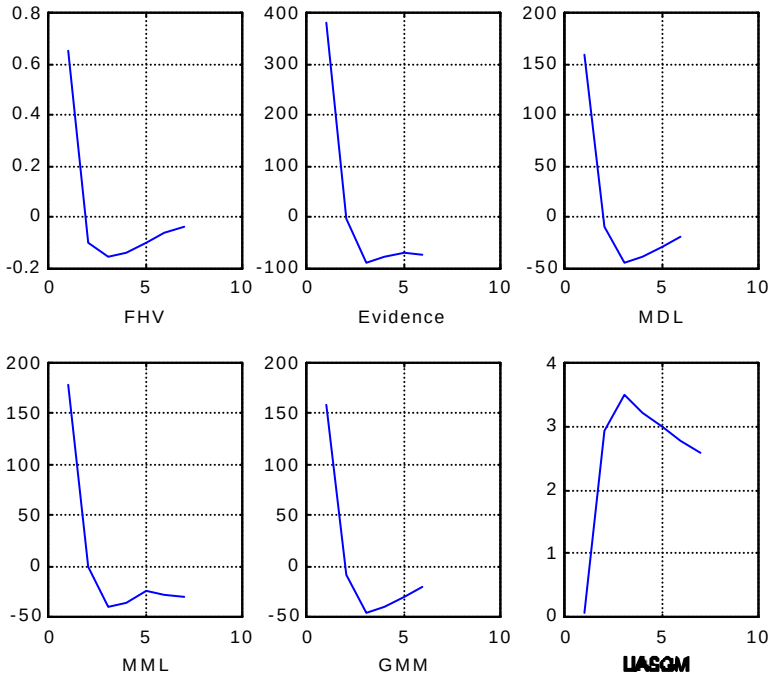


Figura 4.24. Resultados del experimento 3

Todos los métodos dan un resultado correcto de K=3. Se ha realizado una clasificación de los datos con el método UASGM obteniendo cuatro errores en 150 datos, lo que supone un 97.5% de aciertos. Estos resultados son parecidos a los obtenidos con el método GMM (98% de aciertos) de la referencia [Roberts 98].

Después de analizar estos experimentos se puede concluir que **los resultados logrados con el algoritmo UASGM son similares a los que se han conseguido empleando los métodos MML, MDL y GMM, siendo un método mucho más sencillo y con menor carga computacional, que además supera los resultados de los métodos FHV y de la Evidencia**. En la tabla 4.11. se muestra un análisis comparativo de la carga computacional de los diferentes algoritmos, en número de "flops" referenciados al método UASGM. Como se observa, todos ellos tienen una carga por encima de cinco veces la del UASGM.

Métodos	Carga Computacional ($Mflops/Mflops_{UASGM}$)
UASGM	1
FHV	5,01
Evidence	5,02
MDL	5,03
MML	5,97
GMM	6,56

Tabla 4.11. Comparación de carga computacional entre los distintos métodos

Descartados todos los métodos estudiados en favor del propuesto en esta Tesis, a continuación se va a analizar la aplicación del método UASGM sobre el ejemplo que se ha estado utilizando en la explicación y que se corresponde con la imagen de la figura 4.19.

Experimento 4. Aplicación sobre una imagen real de una cara con fondo aleatorio. En la figura 4.25 se observa que el número óptimo de clases calculado por el algoritmo es K=3. En la figura 4.26 se muestra la clasificación de los pixels de la imagen en las distintas clases para las distintas configuraciones evaluadas (2#K#7). Como se puede apreciar, para K=2 se distinguen el color de la cara y el resto. En K=3 aparece una nueva clase para el color de la camisa. En K=4 el color del fondo

se divide en dos clases, debido a que hay una parte más clara y otra más oscura en la imagen. Con K=5 el color de la piel se divide en dos, una para las partes más rojas (labios y rojeces de la cara) y otra para el resto. Con K=6, el color de la camisa se divide en dos y con K=7 aparece un nuevo color para las zonas más oscuras de la imagen, como ojos y extremos de la barba. Como el objetivo es localizar el color de la piel, la mejor aproximación de los principales colores de la imagen se obtiene para K=3 donde se localiza: una clase para el color de la piel, otra para el del fondo y otra para el de la camisa.

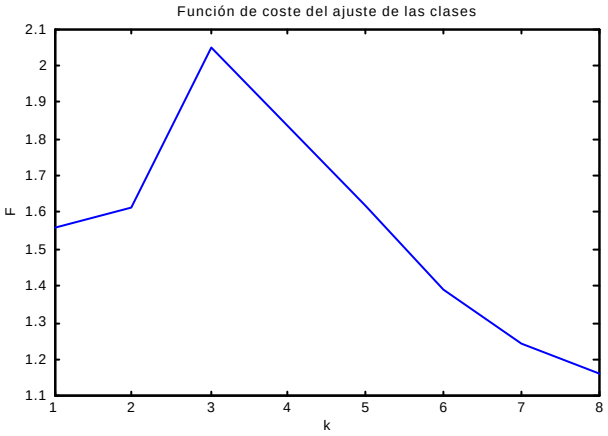


Figura 4.25. Función de coste

Los resultados obtenidos con este método son mejores que los presentados en [Moreira&Costa, 96] ya que utiliza el algoritmo de aprendizaje VQ, con un número óptimo de vectores, en lugar del SOM (Self Organizing Map). Los SOM son organizadores topológicos en el sentido de que organizan un número de neuronas P sobre un mapa, en función de la topología de los datos de entrada, pero no realizan un proceso de "clustering" por sí mismos, sino que necesitan de otro método para hacer la clasificación [Moreira&Costa, 96], es decir, es un método de reducción de la cantidad de información a clasificar pero no clasifica. Ejemplos de este tipo se tienen en el trabajo de Moreira ya referenciado donde, una vez situadas las neuronas en el mapa de Kohonen, hace una clasificación supervisada de las mismas y después clasifica los pixels empleando la técnica de los K-vecinos más cercanos. Otros ejemplos son [Campbell et al.,96] [Campbell&Thomas,97], donde aplican un SOM para organizar los datos y, posteriormente, utilizan un perceptrón multicapa con entrenamiento supervisado para hacer la segmentación.

En el método desarrollado en esta tesis se sitúan las neuronas sobre el espacio de color y sus

posiciones son las medias de las funciones gaussianas que modelan el histograma, de manera que la clasificación se realiza directamente sin emplear ningún otro método. Por otra parte, el sistema es capaz de calcular, de forma no supervisada, el número de funciones que mejor modelan el histograma.

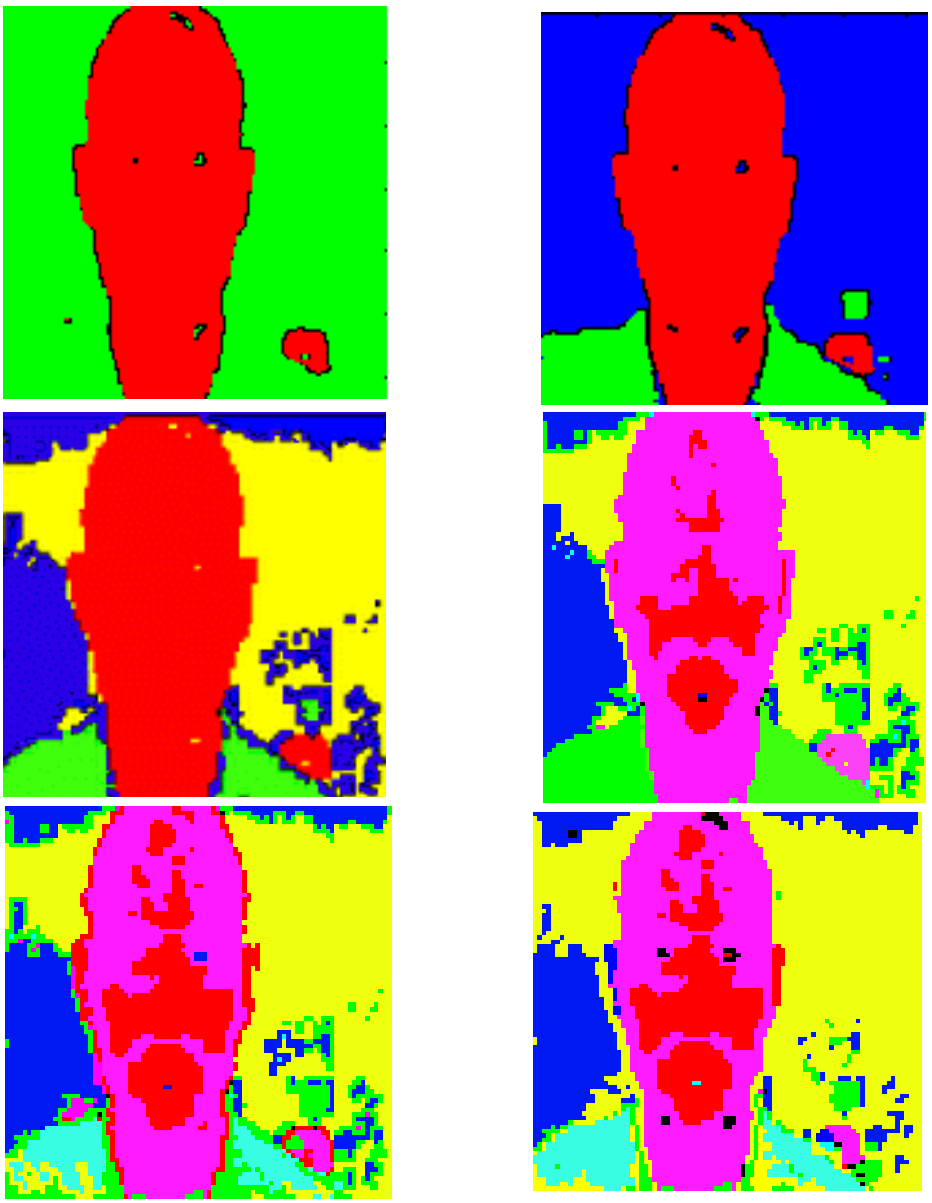


Figura 4.26. Clasificación de los pixels para las distintas clases evaluadas.

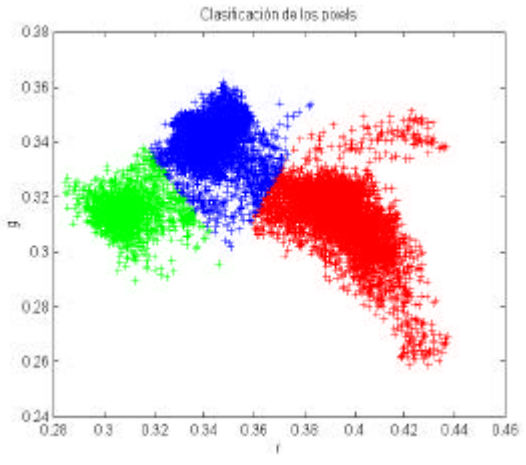
Como se puede ver en la figura 4.26 el "clustering" realizado (para K=3) no es perfecto, ya que hay colores en la imagen con un pequeño número de pixels que no son detectados por el algoritmo (la ventana) mientras que considera como color de piel la piel en sí, la boca y parte del pelo. No obstante, este método da una buena estimación de los pixels de piel de la imagen mejorando el algoritmo de [Stiefelhagen et al., 97], donde la clase piel a priori es computada
"off-line".

4.5.4. Segmentación de la piel

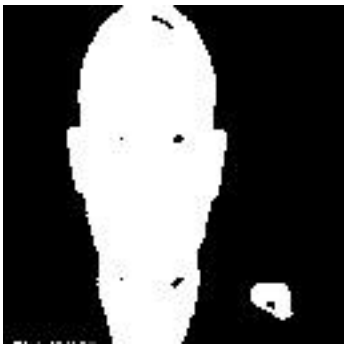
Una vez clasificada la imagen en los principales colores que la componen hay que localizar, entre ellos, la clase piel. Para ello, se utiliza un método a nivel de clases y no a nivel de pixels, consistente en computar la distancia euclídea entre los centros de los "clusters" de la imagen $\hat{\mathbf{m}}_k$ y un "cluster" patrón que representa el prototipo de color de la piel humana ($\mathbf{m}_{\text{patrón}}$). La clase que esté más cerca de este prototipo será considerada como clase piel, según la siguiente ecuación:

$$\hat{m}_S^{(0)} = \min_k \left\{ || \hat{m}_k - m_{patron} || \right\} \tag{4.99}$$

En la figura 4.27 se presenta la clasificación de los pixels en el espacio de color y la clase detectada como piel para la figura del ejemplo.



(a)



(b)

Figura 4.27. (a) Clasificación de los pixels en el espacio rg (b) Detección de la clase piel.

Las distancias entre el patrón y los centros de cada clase se muestran en la siguiente tabla:

	piel	fondo	camisa
distancia	0,0248	0,0480	0,0616

Tabla 4.12. Distancias entre las clases y el patrón

Experimentalmente se comprobó que la detección de la clase piel fallaba si las condiciones de iluminación cambiaban considerablemente. El sistema funcionaba bien en el laboratorio y en los pasillos, pero fallaba en los descansillo de los ascensores (con menor iluminación) y cuando la silla se acercaba a las cristaleras (debido al efecto de la luz directa del sol). A estos problemas hay que añadir el que la apertura del diafragma de la cámara también influye en la percepción de los colores de la imagen. Todos estos efectos provocan un desplazamiento del patrón de piel respecto al calculado a priori, tal y como se demostró en el Anexo 1. Esto puede provocar que algún otro color (no de piel) se encuentre más cerca que el patrón y, por lo tanto, sea considerado como color de piel.

Para lograr el funcionamiento del sistema en un rango más amplio de iluminación, e independientemente de la posición del diafragma, se sustituyó el patrón de piel a priori por un patrón calculado en función de la luminancia de la imagen.

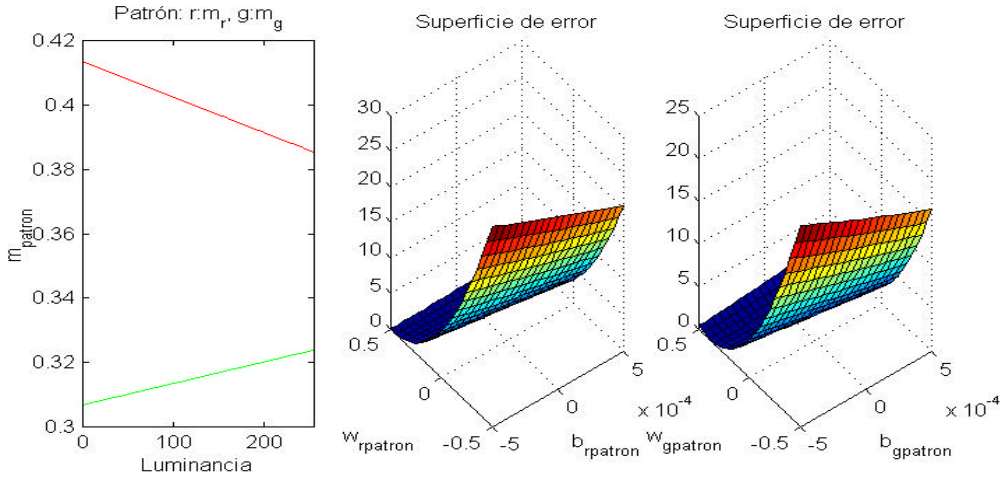
Para ello se tomaron diferentes secuencias de imágenes de distintos usuarios. Se probaron para distintas condiciones de luz de los interiores de la Escuela Politécnica y se modificó la posición del diafragma. Se observó que existía una dependencia lineal entre las posiciones del patrón y la luminancia, por lo que el problema fue resuelto aplicando la técnica de mínimos cuadrados. En la figura 4.28 se muestran las funciones obtenidas para las componentes r y g del patrón, así como las gráficas de las superficies de error en función de los parámetros de las funciones.

Para mejorar la segmentación de la clase piel se modela la distribución de colores de los pixels de piel mediante una función gaussiana 2D, $N(\hat{m}_S^{(0)}, \hat{C}_S^{(0)})$, donde $\hat{m}_S^{(0)}$ es la posición del prototipo de la clase piel y $\hat{C}_S^{(0)}$ es la matriz de covarianza de las componentes de color "rg" de los pixels clasificados como piel (M_S). La función gaussiana da la densidad de probabilidad de un pixel de pertenecer a la clase piel:

$$f(x_j / skin) = \frac{1}{2\pi \hat{C}_S^{0.5}} e^{-0.5(x_j - \hat{m}_S)^T \hat{C}_S^{-1}(x_j - \hat{m}_S)} \qquad (4.100)$$

La función de pertenencia de un pixel a la clase piel viene dada por la expresión anterior multiplicada por el factor $(2\pi \hat{C}_S^{0.5})$. Se establece un umbral (Th), de forma que, si el valor de la función de pertenencia es mayor que el umbral, se considera que el pixel pertenece a la clase piel y en caso contrario no.

En la figura 4.29 se muestran los pixels pertenecientes a la clase piel (color rojo) aplicando el proceso de clustering y el modelo para distintos umbrales. Asimismo en la figura 4.3 aparecen las imágenes



(a) (b) (c)

Figura 4.28. (a) Patrón en función de la luminancia, (b) Superficie de error para "r", (c) Superficie de error para "g"

segmentadas en el espacio (x,y) para los distintos umbrales.

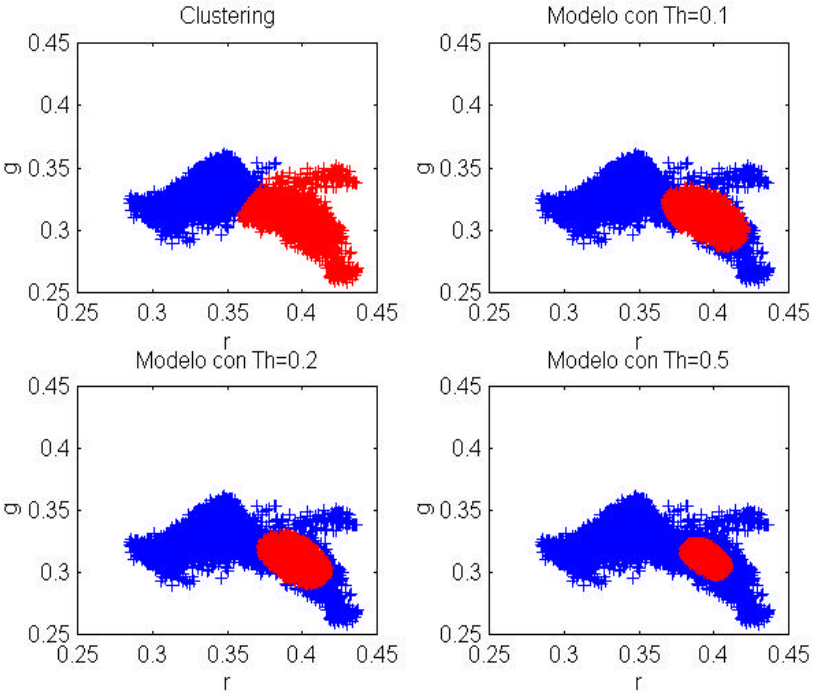


Figura 4.29. Distribución de los pixels para distintos umbrales

Th=0.1 Th=0.2 Th=0.5

Figura 4.30. Segmentación para diferentes niveles de umbral

Una vez segmentada la piel para la primera imagen de una secuencia, para las siguientes se aplica el modelo y se vuelve a hacer la segmentación. No hay que olvidar que el sistema se va a aplicar al guiado de una silla de ruedas donde el usuario se va a estar moviendo y, por lo tanto, las condiciones de la escena cambiarán. Los sistemas basados en color son sensibles a los cambios de iluminación y a cambios de escena. Para poder utilizar este método en un amplio rango de condiciones de iluminación y escenas diferentes se utilizará un modelo adaptativo que será explicado en el siguiente punto.

4.5.5. Adaptación del modelo

El modelo adaptativo usa una combinación lineal de los parámetros del modelo conocidos para predecir los nuevos parámetros. Siendo $\mathbf{X_S}$ los pixels de la clase piel, se ha demostrado que $\mathbf{X_S}$ es una distribución normal bivariable, si $\mathbf{Y_S=BX_S}$ es una transformación lineal de $\mathbf{X_S}$. Donde **B** es una matriz (m x p) de números reales con m#p y rango m, $\mathbf{Y_S}$ también es una distribución normal bivariable.

Para la segmentación de la piel de una imagen estática se ha utilizado un modelo estocástico Gaussiano definido por sus estadísticos media $(\hat{m}_S^{(0)})$ y covarianza $(\hat{C}_S^{(0)})$. Si ahora se está analizando una secuencia de imágenes se puede considerar que el valor estimado de los estadísticos serán una combinación lineal de los z últimos valores obtenidos para cada uno de ellos, esto es:

$$\hat{m}_S^{(p+1)} = \sum_{l=0}^{z-1} \alpha_l R_l \qquad (4.101)$$

donde $\hat{m}_S^{(p+1)}$ es el vector media estimado para el instante (p+1); $\mathbf{R_l}$, son los vectores media previos

$$\hat{C}_S^{(p+1)} = \sum_{l=0}^{z-1} \beta_l S_l \tag{4.102}$$

y "$_l$ #1 son los coeficientes para el cálculo de la media estimada, l=1,...,z. El término $\hat{C}_S^{(p+1)}$ es la matriz de covarianza estimada para (p+1); $\$_l$#1 son los coeficientes para la covarianza estimada y $\mathbf{S_l}$ son las matrices de covarianza previas. Los coeficientes determinan en qué medida los estadísticos pasados van a influir en la estimación de los estadísticos actuales. Se emplea el criterio de máxima probabilidad para encontrar el mejor conjunto de coeficientes para la predicción. La función de probabilidad obtenida al aplicar el modelo estimado a los M_S pixels de la clase piel en el espacio de color bivariable normalizado será el producto de las probabilidades para cada pixel:

$$L = \prod_{k=1}^{M_S} f(x_k) = \frac{1}{(2\pi)^{M_S} |\hat{C}_s|^{\frac{1}{2}M_S}} \exp\left[-\frac{1}{2} \sum_{k=1}^{M_S} (x_k - \hat{m}_s)^T \hat{C}_s^{-1} (x_k - \hat{m}_s) \right] \tag{4.103}$$

El logaritmo de la función de probabilidad será:

$$\log L = -M_S \log(2\pi) - \frac{1}{2} M_S \log|\hat{C}_S| - \frac{1}{2} \sum_{k=1}^{M_S} (x_k - \hat{m}_S)^T \hat{C}_S^{-1} (x_k - \hat{m}_S) \tag{4.104}$$

Donde logL es una función creciente de L, es decir, que un máximo en logL se corresponde con un máximo en L.

Haciendo una simple transformación matemática se tiene:

$$\sum_{k=1}^{M_S} \left(x_k - \hat{m}_S\right)\left(x_k - \hat{m}_S\right)^T = M_S C_S + M_S \left(m_S - \hat{m}_S\right)\left(m_S - \hat{m}_S\right)^T \tag{4.105}$$

Usando este resultado y las propiedades de la traza de una matriz se puede rescribir la ecuación (4.104) como:

$$\log L = -M_S \log(2\pi) - \frac{1}{2} M_S \log|\hat{C}_S| - \frac{1}{2} M_S tr\hat{C}_S^{-1} C_S - \frac{1}{2} M_S (m_S - \hat{m}_S)^T \hat{C}_S^{-1} (m_S - \hat{m}_S) \quad (4.105)$$

La ecuación anterior se utiliza para derivar las ecuaciones respecto a los factores de peso con el fin de obtener la máxima probabilidad. Se van a analizar dos casos: adaptación solamente del vector media y adaptación del vector media y la matriz de covarianza.

Adaptación del vector media

En este caso no se estima la matriz de covarianza y el vector media se estima en función de una combinación lineal de los vectores media previos.

$$\begin{aligned} \hat{m}_S^{(p+1)} &= \sum_{l=0}^{z-1} \alpha_l R_l \\ \hat{C}_S^{(p+1)} &= C_S^{(p)} \end{aligned} \quad (4.106)$$

Donde la matriz R_l viene dada por:

$$R_l = \begin{bmatrix} m_{rS}^{(p-l)} \\ m_{gS}^{(p-l)} \end{bmatrix} \qquad l = 0,\ldots,z-1 \quad (4.107)$$

siendo p el índice temporal que indica la imagen que se está analizando.

Realizando las derivadas de la función de probabilidad (4.105) con respecto a α_l, l=0,...,z-1, e igualando a cero, se obtiene la siguiente expresión:

$$\sum_{l=0}^{z-1} R_j^T \hat{C}_S^{-1} R_l \alpha_j = R_j^T \hat{C}_S^{-1} m_S, \qquad j = 0,\ldots,z-1 \quad (4.108)$$

Si $\sum_{l=0}^{z-1} R_j^T \hat{C}_S^{-1} R_l \neq 0 \qquad j = 0,\ldots,z-1,$ se tiene:

$$\alpha_j = \sum_{l=0}^{z-1} \left(R_j^T \hat{C}_S^{-1} R_l \right)^{-1} R_j^T \hat{C}_S^{-1} m_S, \qquad j = 0,\ldots,z-1 \quad (4.109)$$

sustituyendo los pesos de la ecuación anterior en la ecuación (4.106) se obtiene el valor de la media estimada que, junto con la covarianza, se aplican al modelo y se realiza la segmentación obteniendo los nuevos pixels de piel para hacer la nueva estimación.

Adaptación del vector media y matriz de covarianza

En este caso, se estiman los vectores media y matriz de covarianza como combinación lineal de los valores anteriores según las ecuaciones (4.101) y (4.102).

Como los parámetros a estimar son asintóticamante independientes, cada uno de ellos puede ser estimado asumiendo que el otro es conocido. El valor de θ_l, l=0,...,z-1 se estima aplicando la ecuación (4.109), y, $\mathbf{S_l}$ vendrá dada por la ecuación (4.110).

$$S_l = \left[C_S^{(p\&l)}\right] \qquad l = 0,...,z-1 \tag{4.110}$$

Derivando la ecuación (4.105) respecto a β_l, l=0,...,z-1, e igualando a cero se obtienen los pesos que maximizan la función de probabilidad. Como $\hat{C}_S^{-1}$ es una matriz definida positiva, la ecuación (4.105) se maximiza con respecto a la media cuando $\hat{m}_S = m_S$, con lo que la función de probabilidad se reduce a:

$$-\log(2\pi) - \log|\hat{C}_S| - tr\hat{C}_S^{-1}C_S \tag{4.111}$$

Teniendo en cuenta las siguientes derivadas con respecto a β_l l=0,...,z-1:

$$\frac{\partial}{\partial \beta_j}\hat{C}_S^{-1} = -\hat{C}_S^{-1}S_j\hat{C}^{-1} \qquad j = 0,...,z-1 \tag{4.112}$$

$$\frac{\partial}{\partial \beta_j}\log|\hat{C}_S| = tr\hat{C}_S^{-1}S_j \qquad j = 0,...,z-1 \tag{4.113}$$

La derivada de (4.111) quedará:

$$- \mathrm{tr}\hat{C}_S^{-1}S_j + \mathrm{tr}\hat{C}_S^{-1}S_j\hat{C}_S^{-1}C_S = 0 \qquad (4.114)$$

y sustituyendo la ecuación que estima la máxima probabilidad se obtiene:

$$\mathrm{tr}\left(\sum_{l=0}^{z-1}\beta_l S_l\right)^{-1} S_j = \mathrm{tr}\left(\sum_{l=0}^{z-1}\beta_l S_l\right)^{-1} S_j \left(\sum_{l=0}^{z-1}\beta_l S_l\right)^{-1} C_S \qquad j = 0,\ldots,z-1 \qquad (4.115)$$

El problema de estimar la máxima probabilidad de una función normal multivariable con estructura lineal para los vectores media y matriz de covarianza ha sido estudiada por otros autores [Yang et al.,97]. En general no existe una solución explícita para la ecuación (4.115), y por ello la estimación se realiza empleando técnicas numéricas iterativas. En nuestro caso se ha empleado el proceso de estimación propuesto por Anderson [Anderson, 73].

La idea básica consiste en calcular de forma iterativa e independientemente los coeficientes $\alpha_l^{(i)}$ y $\beta_l^{(i)}$, donde el superíndice (i) indica la inésima iteración dentro de un paso (p). Con tal fin, la ecuación (4.115) se puede reescribir de la siguiente forma:

$$\sum_{l=0}^{z-1}\mathrm{tr}\left(\hat{C}_S^{(i-1)}\right)^{-1}S_j\left(\hat{C}_S^{(i-1)}\right)^{-1}S_l\beta_l^{(i)} = \mathrm{tr}\left(\hat{C}_S^{(i-1)}\right)^{-1}S_j\left(\hat{C}_S^{(i-1)}\right)^{-1}C_S^{(i)} \qquad j = 0,\ldots,z-1 \quad (4.116)$$

El proceso iterativo implica el cálculo de parámetros en el siguiente orden: α_l, $\hat{m}_S$, C_S, β_l, $\hat{C}_S$, l=0,...,z-1. La iteración se termina si $\max(|\beta_l^{(i)} - \beta_l^{(i-1)}|, l=0,\ldots,z-1) \# \varepsilon$, donde ε es un parámetro de error definido por el usuario. En el algoritmo hay que diferenciar dos casos: inicialización e inésima iteración. A continuación se muestran las ecuaciones para cada uno de ellos:

Inicialización

$$\alpha_j^{(0)} = \left(\sum_{l=0}^{z-1} R_j^T R_l\right)^{-1} R_j^T m_S \quad j = 0,\ldots,z-1 \qquad (4.117)$$

Inésima iteración

$$\hat{m}_S^{(0)} = \sum_{l=0}^{z-1} \alpha_l^{(0)} R_l \tag{4.118}$$

$$C_S^{(0)} = \frac{1}{M_S} \sum_{l=1}^{M_S} (x_k - m_S)(x_k - m_S)^T + (x_k - \hat{m}_s^{(i)})(x_k - \hat{m}_s^{(i)})^T \tag{4.119}$$

$$\sum_{l=1}^{v} trS_j S_l \beta_l^{(0)} = trS_j C_S^{(0)} \qquad j = 0, \ldots, z-1 \tag{4.120}$$

$$\hat{C}_S^{(0)} = \sum_{l=0}^{z-1} \beta_l^{(0)} S_l \tag{4.121}$$

$$\alpha_j^{(i)} = \left(\sum_{l=0}^{z-1} R_j^T \left(\hat{C}_S^{(i-1)} \right)^{-1} R_l \right)^{-1} R_j^T \left(\hat{C}_S^{(i-1)} \right)^{-1} m_S \quad j = 0, \ldots, z-1 \tag{4.122}$$

$$\hat{m}_S^{(i)} = \sum_{l=0}^{z-1} \alpha_l^{(i)} R_l \tag{4.123}$$

$$C_S^{(i)} = \frac{1}{M_S} \sum_{l=1}^{M_S} (x_l - m_S)(x_l - m_S)^T + (x_l - \hat{m}_S^{(i)})(x_l - \hat{m}_S^{(i)})^T \tag{4.124}$$

$$\sum_{l=0}^{z-1} tr\left(\hat{C}_S^{(i-1)} \right)^{-1} S_j \left(\hat{C}_S^{(i-1)} \right)^{-1} S_l \beta_l^{(i)} = tr\left(\hat{C}_S^{(i-1)} \right)^{-1} S_j \left(\hat{C}^{(i-1)} \right)^{-1} C_S^{(i)} \qquad j = 0, \ldots, z-1 \tag{4.125}$$

Con los estadísticos estimados en el paso “p” se aplica el modelo sobre una nueva imagen “(p+1)", se realiza la segmentación y se vuelven a estimar los parámetros para la siguiente. Experimentalmente se ha tomado un valor de z=20.

Estudio comparativo entre las dos formas de adaptación

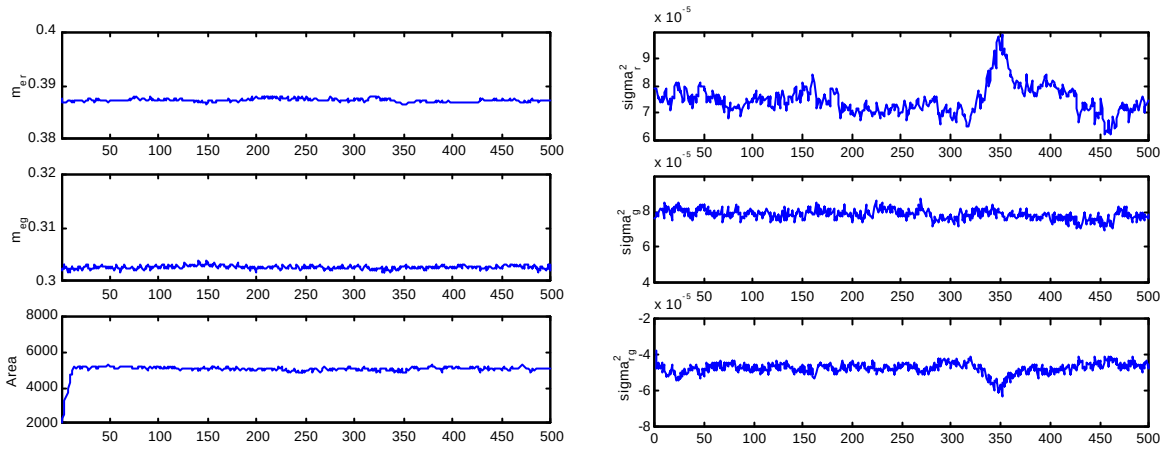


Figura 4.31. Variación de los parámetros del modelo sin estimación

Lo primero que cabría plantearse es: ¿cómo responde el sistema sin adaptación? Para responder a esta pregunta, en la figura 4.31 se muestra una secuencia de 500 imágenes en las que el usuario mantiene la cabeza fija mirando a la cámara. En ella se pueden ver la evolución de: los estadísticos $\hat{m}_S$ (que vienen dados por m_{er} y m_{eg}); $\hat{C}_S$ (que se representa por $sigma^2_r$, $sigma^2_g$, $sigma^2_{rg}$); y del área del objeto piel, en función del tiempo. Como se puede ver, inicialmente hay un transitorio en el área hasta que se estabiliza en un cierto valor. Sobre los parámetros aparece una componente de ruido debida a la falta de estimación de los mismos. El error de segmentación producido, en este caso, es de aproximadamente un 8 % de los pixels de piel, para un umbral fijo, Th=0.2. El valor dado es aproximado ya que, aunque el usuario debe estar inmóvil mirando a la cámara, en la práctica tiene un pequeño cabeceo que hace que el objeto piel pueda tener pequeñas variaciones.

En la figura 4.32 se muestran los mismos parámetros que en el caso anterior y en la misma situación pero ahora estimando el vector media del modelo. Como se observa, también existe un transitorio en el área del objeto, pero en este caso la variación del vector media y del área en función del tiempo ha diminuido como consecuencia de la estimación. Los parámetros de covarianza no sufren modificación, respecto al caso anterior, al no realizarse estimación sobre ellos. En este caso se consigue disminuir el

error de segmentación a un 2% aproximadamente de los pixels de piel, lo que supone una gran mejora respecto a no estimar.

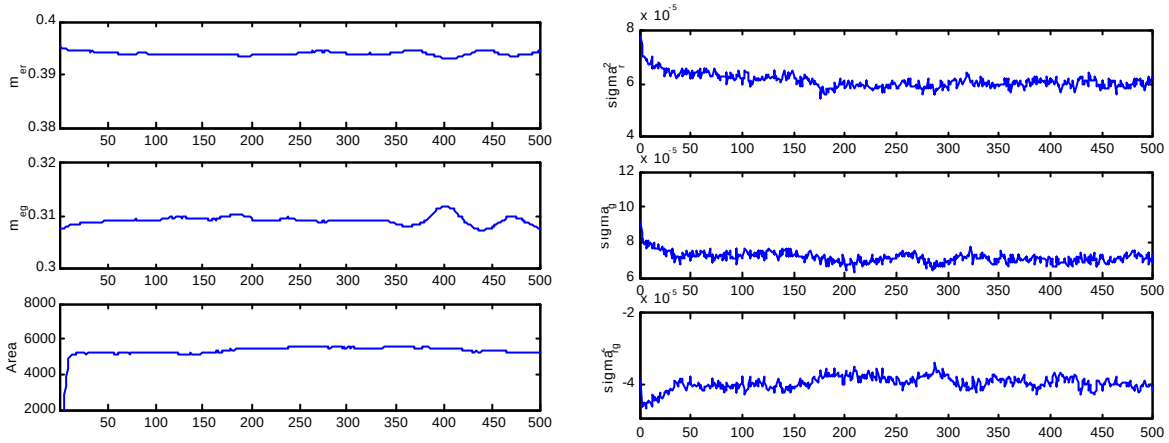


Figura 4.32. Variación de los parámetros del modelo estimando el vector media

Por último, en la figura 4.33 se muestran los parámetros para las condiciones anteriormente descritas y estimando el vector media y la matriz de covarianza. Como se ve, el ruido de todos los estadísticos se reduce respecto al obtenido sin estimación. El error de segmentación ahora es de aproximadamente un 1%, lo que supone una pequeña disminución respecto a estimar solamente el vector media.

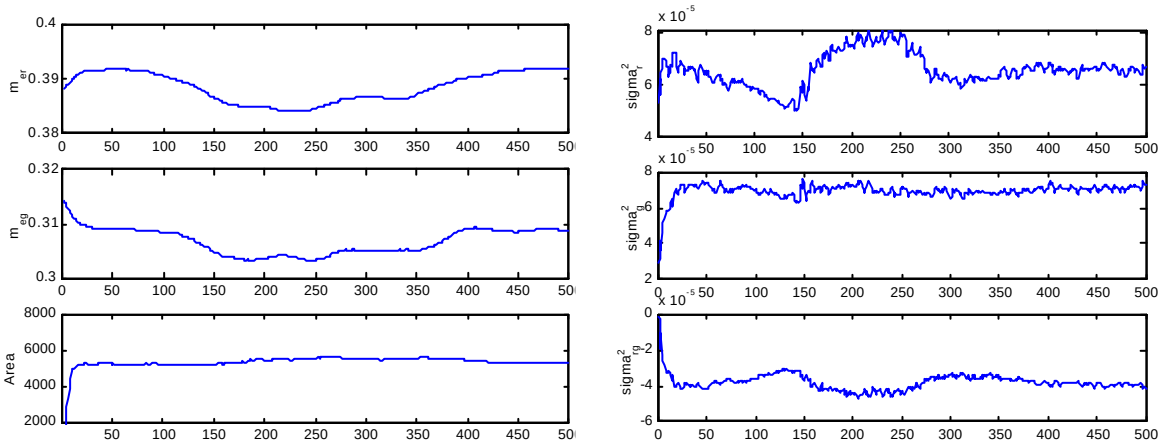
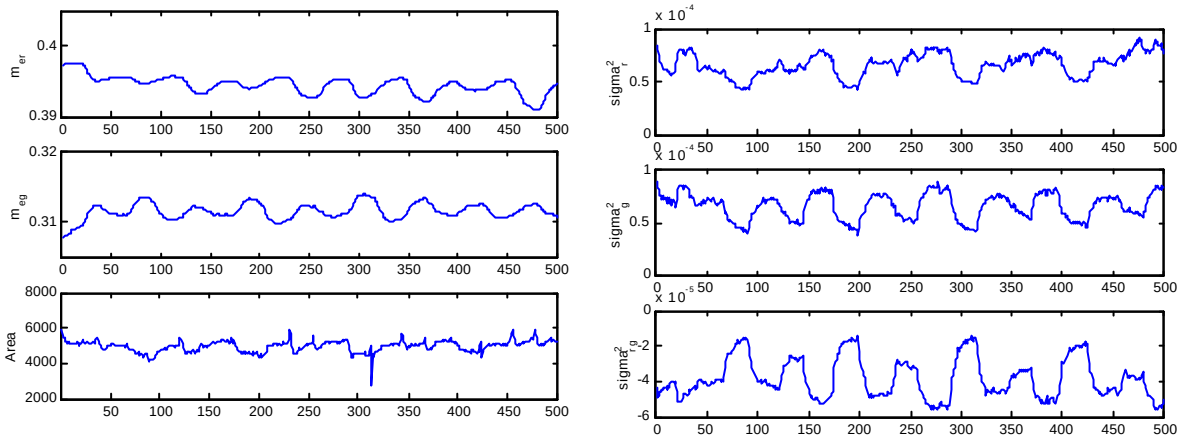


Figura 4.33. Variación de los parámetros del modelo estimando media y covarianza

Por lo tanto, se puede concluir diciendo que los mejores resultados se obtienen estimando tanto el vector media como la matriz de covarianza del modelo.

A continuación se va a presentar otra prueba consistente en ver la variación de los estadísticos para una secuencia en la que el usuario está continuamente realizando movimientos con su cabeza. En la figura 4.34 se muestran los resultados para un umbral fijo, Th=0.2.

F



igura 4.34. Variación de los parámetros del modelo para movimiento de cabeza y Th fijo

Como se observa, en algunos movimientos se producen grandes transitorios en el área del objeto que se estabilizan con el tiempo. Experimentalmente se observó que la segmentación con imágenes en movimiento mejoraba usando un umbral adaptativo proporcional a la traza de la matriz de la covarianza estimada, como se muestra en la siguiente ecuación:

$$Th = K_{Th}\, tr\left[\hat{C}_S\right] \qquad (4.127)$$

Aquí, K_{th} es la constante de proporcionalidad calculada experimentalmente.

En la figura 4.35 se muestra un ejemplo de evolución de los parámetros del modelo para las mismas condiciones que las de la figura 4.34 pero usando un umbral adaptativo. En este caso desaparecen los transitorios de área cuando se producen movimientos bruscos ya que el umbral adaptativo mejora la

respuesta del sistema.

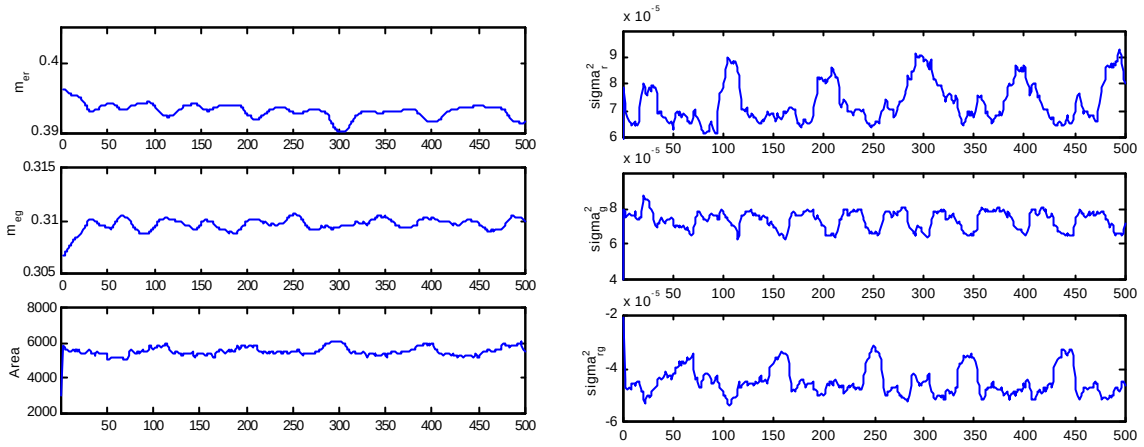


Figura 4.35. Variación de los parámetros para movimiento de cabeza y Th adaptativo

La última prueba a realizar consiste en provocar pérdidas del objeto piel en la imagen y ver el comportamiento del sistema. En la figura 4.36 se muestran los resultados de esta prueba, donde se han provocado pérdidas tanto en la muestra 160 como en la 225; en el primer caso, quitanto la cabeza del campo visual, y, en el segundo, tapando el objetivo con la mano. Se observa que el sistema se recupera rápidamente cuando el objeto piel vuelve a aparecer en la imagen.

El sistema dispone de un serie de controles, tanto en el espacio "rg" como en el (x,y),para detectar posibles fallos o pérdidas del segmentador. En "rg" se considera que ha habido un error cuando la covarianza del modelo está por debajo de un umbral mínimo o por encima de uno máximo establecidos a priori. Asimismo en (x,y) se controla que el objeto piel tenga un tamaño que debe estar comprendido entre un valor máximo y mínimo establecido por el programador.

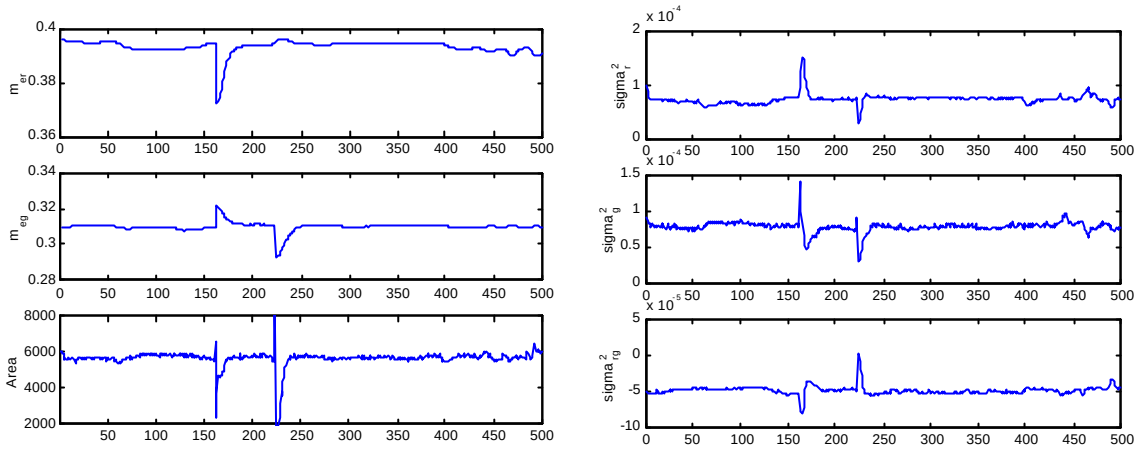


Figura 4.36. Variación de los parámetros del modelo ante pérdidas provocadas

4.6. RESULTADOS

Como colofón del método propuesto se presenta una comparativa de los resultados obtenidos aplicando el método UASGM y GLVQ-F sobre una serie de imágenes de prueba de personas de distinto sexo y raza.

Experimento 1

Inicialmente se aplicó el método GLVQ-F a la imagen de la figura 4.13, cuyos resultados para el algoritmo UAGSM se han expuesto en el punto anterior. Hay que recordar que el método GLVQ-F posiciona los K vectores que mejor aproximan el histograma sobre el espacio "rg" empleando para ello aprendizaje local. Sin embargo, no calcula el número de vectores óptimos para realizar el "clustering". Es por ello que, para que la comparación entre los dos métodos sea más exacta, se aplicó el mismo método de obtención del $K_{óptimo}$ que el utilizado en el método UASGM.

En la figura 4.37 (a) se muestran la función de coste del "clustering", donde se observa que da un número óptimo de cuatro clases, una más que con el otro método. En la figura 4.37(b) aparece el "clustering" de los pixels en el espacio (r,g) y en (e) en el espacio (x,y). En (c) y (f) se dan los pixels segmentados como piel en el "clustering" para (r,g) y (x,y) respectivamente.Por último en (d) y (g) se

pueden ver los pixels segmentados como piel por el modelo en los espacios (r,g) y (x,y) y para un umbral Th=0.2.

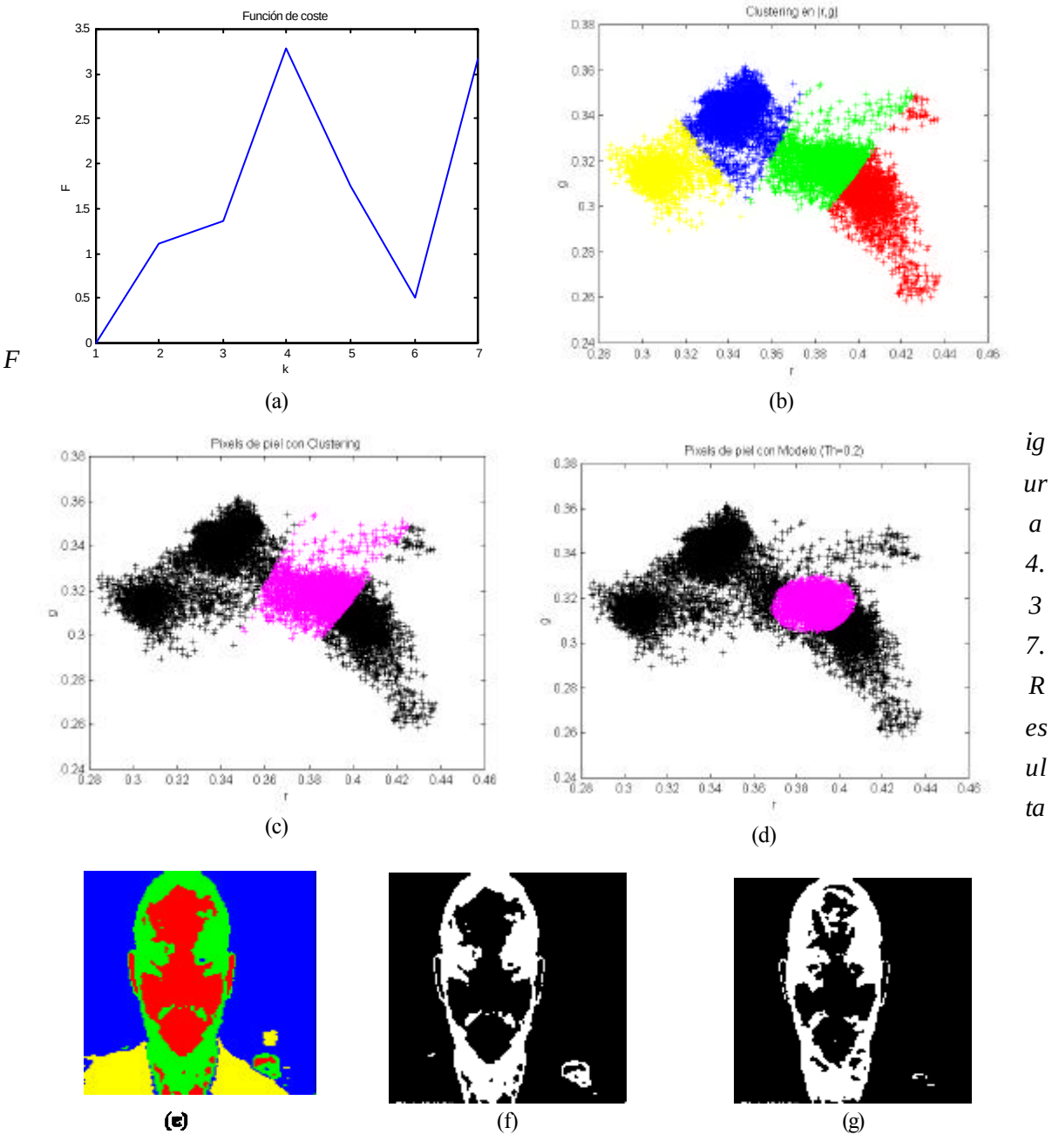


Figura 4.37. Resultados de la segmentación con GLVQ-F para la imagen de la figura 4.13. (a) Función de coste, (b) "Clustering" en (r,g), (c) Pixels de piel para el "clustering", (d) Pixels de piel con el modelo, (e) "Clustering" en (x,y), (f) Segmentación de la piel para el "clustering", (g) Segmentación de la piel para el modelo.

En la tabla 4.13 se muestran los resultados de los errores de segmentación de los métodos así como de los tiempos de cómputo de los mismos, tomando como referencia el tiempo del UASGM. Como se observa, con el método propuesto en esta Tesis se obtiene un error mucho menor y con un tiempo de cómputo también menor.

Métodos	Error de Segmentación[3] (% de pixel)	Tiempo de computación (T/T_{UASGM})
UASGM	2	1
GLVQ-F	30	1,45

Tabla 4.13. Comparación de métodos para el experimento 1

Experimento 2

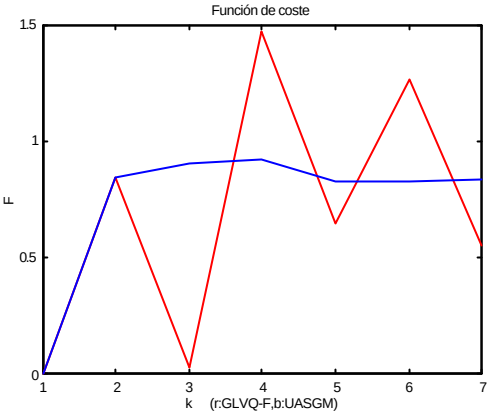
En la figura 4.8 se presentan tres casos de imágenes en las que el color del fondo y de la piel se encontraban muy solapados. En este experimento se va a comprobar el comportamiento de los dos algoritmos estudiados para dichas imágenes. En las figuras 4.38, 4.39 y 4.40 se muestran los resultados de la comparación. En la tabla 4.14 se comparan los errores de segmentación y los tiempos de computación para los dos métodos.

Métodos	Error de Segmentación (% de pixel)			Tiempo de computación (T/T_{UASGM})		
Métodos	fig 4.38	fig 4.39	fig 4.40	fig 4.38	fig 4.39	fig 4.40
UASGM	18	20	8	1	1	1
GLVQ-F	10	35	9	2,06	2,21	1,77

Tabla 4.14. Comparación de métodos para el experimento 2

No se considerará como error aquellos pixels de fondo segmentados como piel que no tengan conectividad con el objeto piel, ya que éstos no son tenidos en cuenta por el sistema.

(a)


Función de coste
F
k (r:GLVQ-F,b:UASGM)

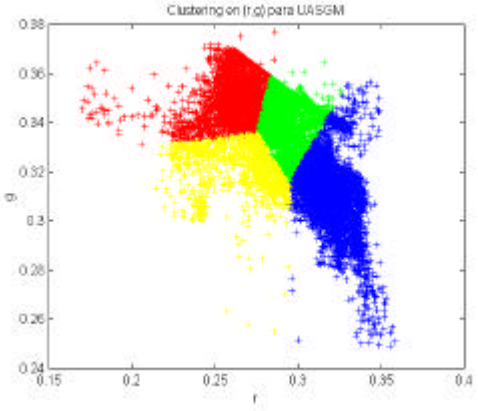
(b)


Clustering en (x,y) para UAGSM

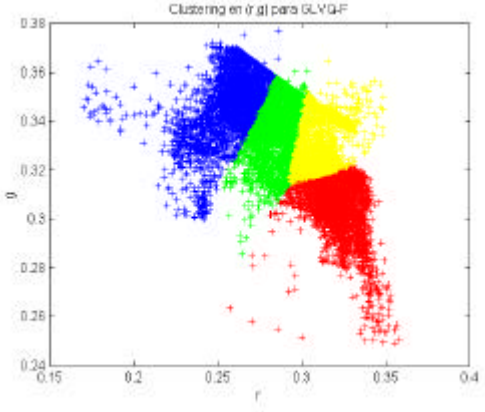
(c)


Clustering en (x,y) para GLVQ-F

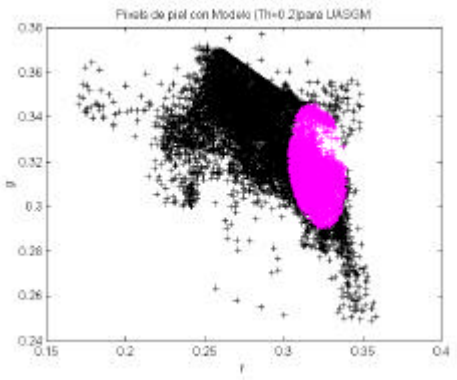
(d)


Clustering en (r,g) para UASGM
g
r

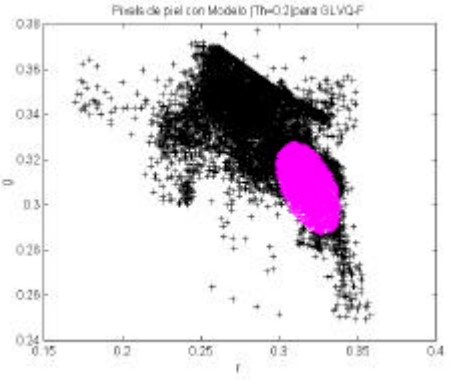
(e)


Clustering en (r,g) para GLVQ-F
g
r

(f)



(g)



(h)

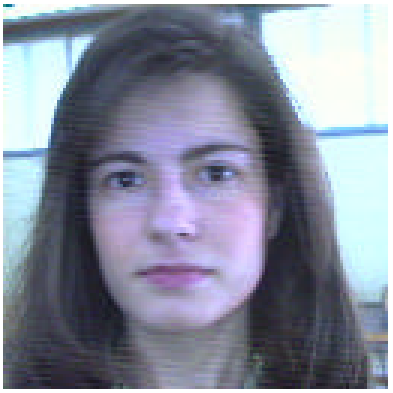
(i)

(j)

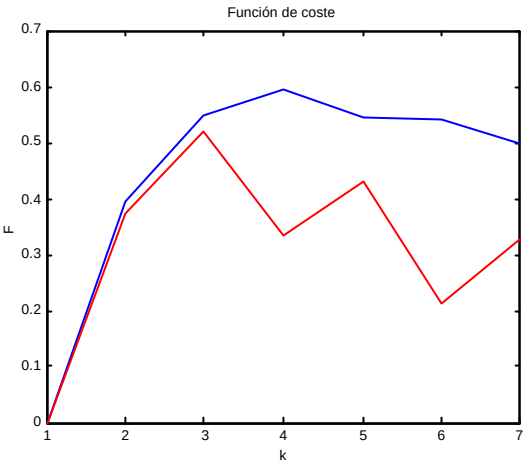
Figura 4.38. Comparación de los métodos UASGM y GLVQ-F para la figura 4.13
(a) Imagen original, (b) Función de coste, (c)(d) "Clustering" en (x,y) para UASGM y GLVQ-F, (e)(f) "Clustering" en (r,g) para UASGM y GLVQ-F, (g)(h) Pixels de piel con modelo para UASGM y GLVQ-F, (i)(j) Segmentación para UASGM y GLVQ-F

En este caso los dos métodos obtuvieron cuatro clases óptimas. La variación para la del GLVQ-F es mayor que para el UASGM, existiendo para la primera varios máximos locales, mientras que para la segunda únicamente hay un máximo. Al aplicar el "clustering" en el método UASGM, la clase piel (en azul en (e)) contiene parte de pixels de las ventanas del fondo, lo que desplaza ligeramente el modelo hacia arriba (rosa en (g)) provocando que se segmenten parte de las ventanas y de las lámparas fluorescentes como piel. Este ruido, salvo en el caso de una ventana, será filtrado al no existir conectividad con el objeto piel (ver figura 4.38(i)). El "clustering" en el GLVQ-F es mejor que el anterior y, por ello, introduce menos pixels en el objeto piel. Al aplicar el modelo sobre él se consigue una buena segmentación. Lógicamente hay reflejos y sombras que no son considerados como piel ya que su color difiere mucho del resto.

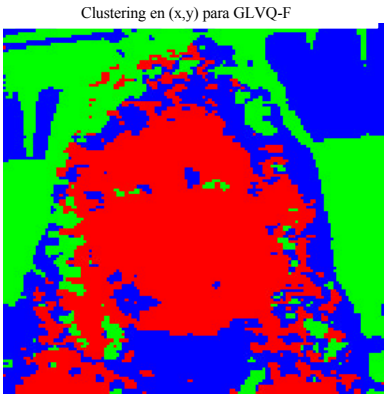
Se puede concluir diciendo que, para este caso, se consigue una mejor segmentación con el método GLVQ-F, aunque con un tiempo de computación el doble que con el primero.



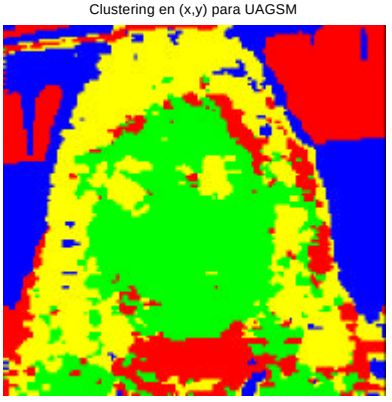
(a)



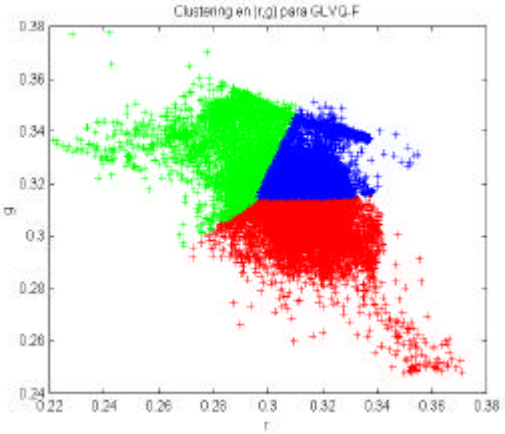
(b)



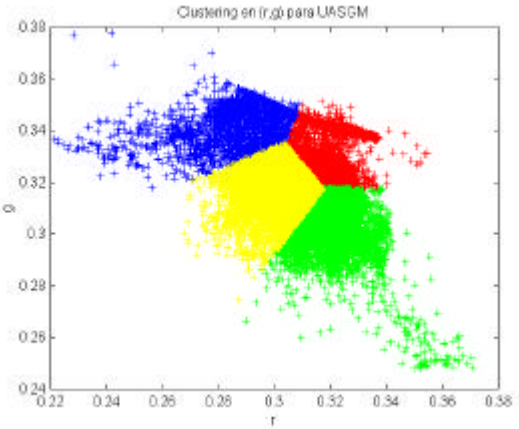
(c)



(d)



(e)



(f)

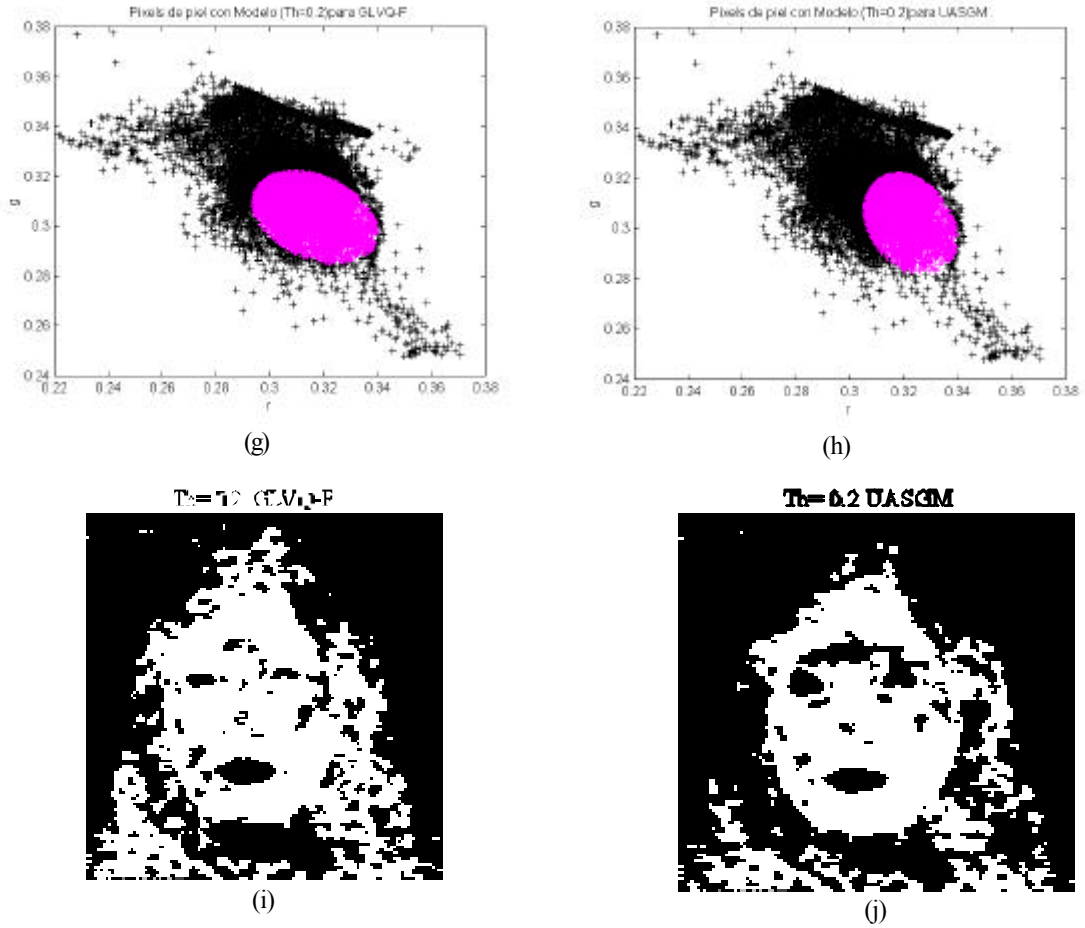
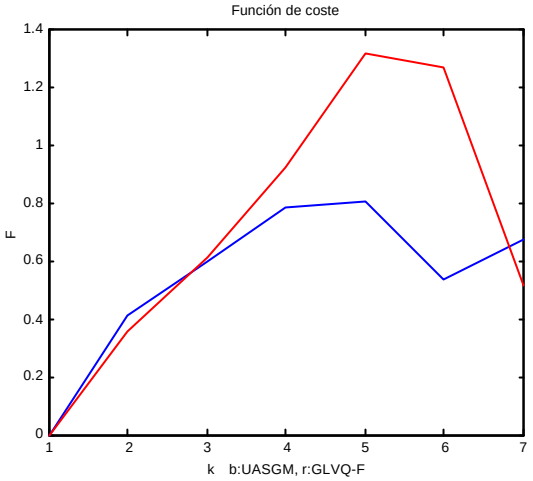


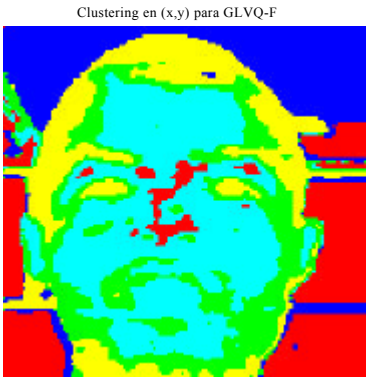
Figura 4.38. Comparación de los métodos UASGM y GLVQ-F para la figura 4.13(b)
(a) Imagen original, (b) Función de coste, (c)(d) "Clustering" en (x,y) para GLVQ-F yUASGM , (e)(f) "Clustering" en (r,g) para GLVQ-F yUASGM , (g)(h) Pixels de piel con modelo para GLVQ-F y UASGM, (i)(j) Segmentación para GLVQ-F y UASGM

En este caso, el método UASGM ha detectado cuatro clases óptimas y el GLVQ-F, tres. Al igual que antes, en esta última función aparecen máximos locales. En este caso el "clustering" es más ajustado para el método UASGM (clase verde en (f)) que para el GLVQ-F (en rojo en (e)) al tener una mayor cantidad de pixels de pelo que la primera. Al aplicar el modelo a este "cluster" se obtiene una mejor segmentación para el método UASGM y con menor carga computacional. Hay que destacar que esta imagen es especialmente difícil de segmentar debido a que el color del pelo del usuario (castaño claro) tiene una crominancia muy parecida a la de la piel. Los objetos de pelo que no tienen conectividad son eliminados, logrando un error de segmentación de aproximadamente un 20%.

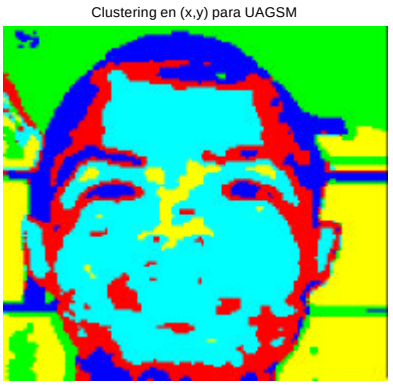
(a)


Función de coste
F
k b:UASGM, r:GLVQ-F

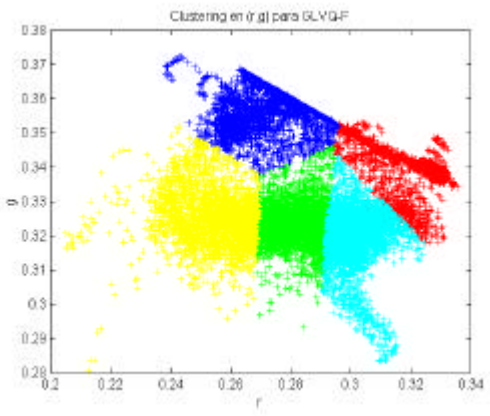
(b)


Clustering en (x,y) para GLVQ-F

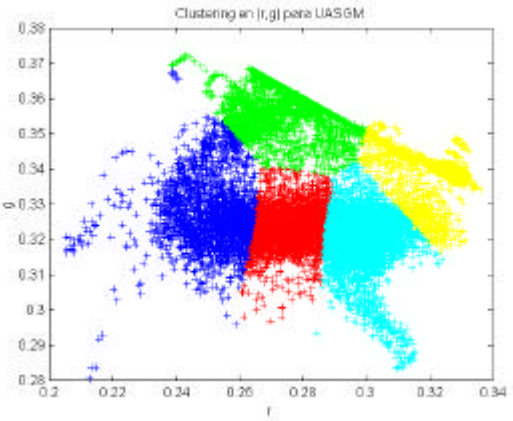
(b)


Clustering en (x,y) para UAGSM

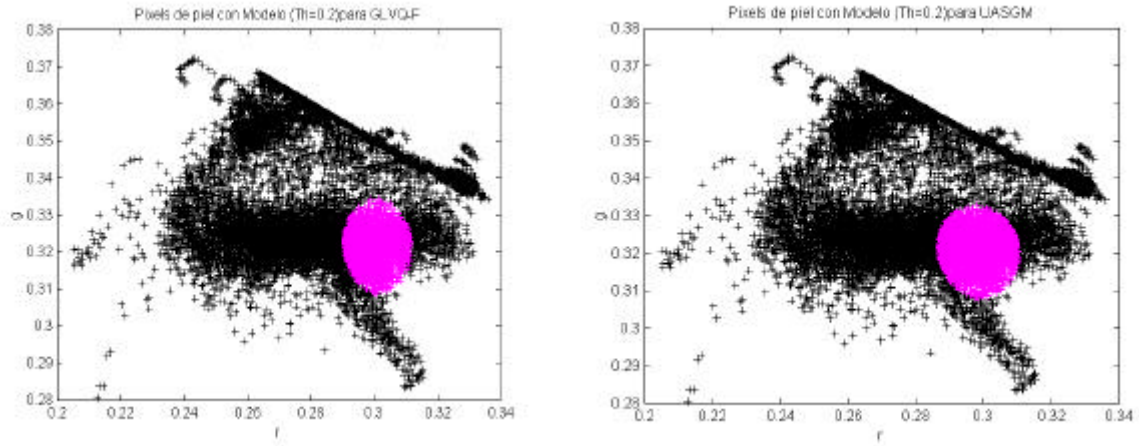
(c)


Clustering en (r,g) para GLVQ-F
g
r

(d)


Clustering en (r,g) para UASGM
g
r

(e)



(g) (h)

(i) (j)

Figura 4.40. Comparación de los métodos UASGM y GLVQ-F para la figura 4.13(c)
(a) Imagen original, (b) Función de coste, (c)(d) "Clustering" en (x,y) para GLVQ-F yUASGM , (e)(f) "Clustering" en (r,g) para GLVQ-F yUASGM , (g)(h) Pixels de piel con modelo para GLVQ-F y UASGM, (i)(j) Segmentación para GLVQ-F y UASGM

Con ambos métodos se obtienen cinco clases óptimas. El "clustering" para ambos métodos (figuras (d) y (e)) es muy similar. Al aplicar el modelo, la clase piel para el método GLVQ-F tiene una dispersión en "r" un poco menor que para el UASGM (ver figuras (g) y (h)). Las segmentaciones para los dos métodos son muy parecidas aunque ligeramente mejores para el UASGM, siendo el tiempo de cómputo de este último menor que el anterior.

Con este experimento queda demostrado el buen funcionamiento del método propuesto en esta tesis para imágenes especialmente complicadas, consiguiendo unos errores de segmentación por debajo del 20% en todos los casos. Comparándolo con el GLVQ-F se obtienen mejores resultados de segmentación que en el primero (exceptuando el caso de la figura 4.38) con un tiempo de cómputo muy inferior.

Experimento 3

En este experimento se va a evaluar la robustez de los métodos ante imágenes con distinta apertura de iris. Para ello se han tomado tres imágenes de prueba: una con el iris en una posición intermedia, otra con el iris muy cerrado y la tercera con el iris muy abierto. Los resultados se pueden ver en las figuras 4.41, 4.42 y 4.43.

La figura 4.41 se corresponde con una imagen en la que el iris está a mitad. En esta imagen los dos métodos obtienen un máximo en la función de calidad para un número de clases igual a cuatro. El "clustering" es muy similar en los dos métodos, al igual que la segmentación. En la tabla 4.15 se presentan los errores de segmentación y los tiempos de computación para los dos algoritmos.

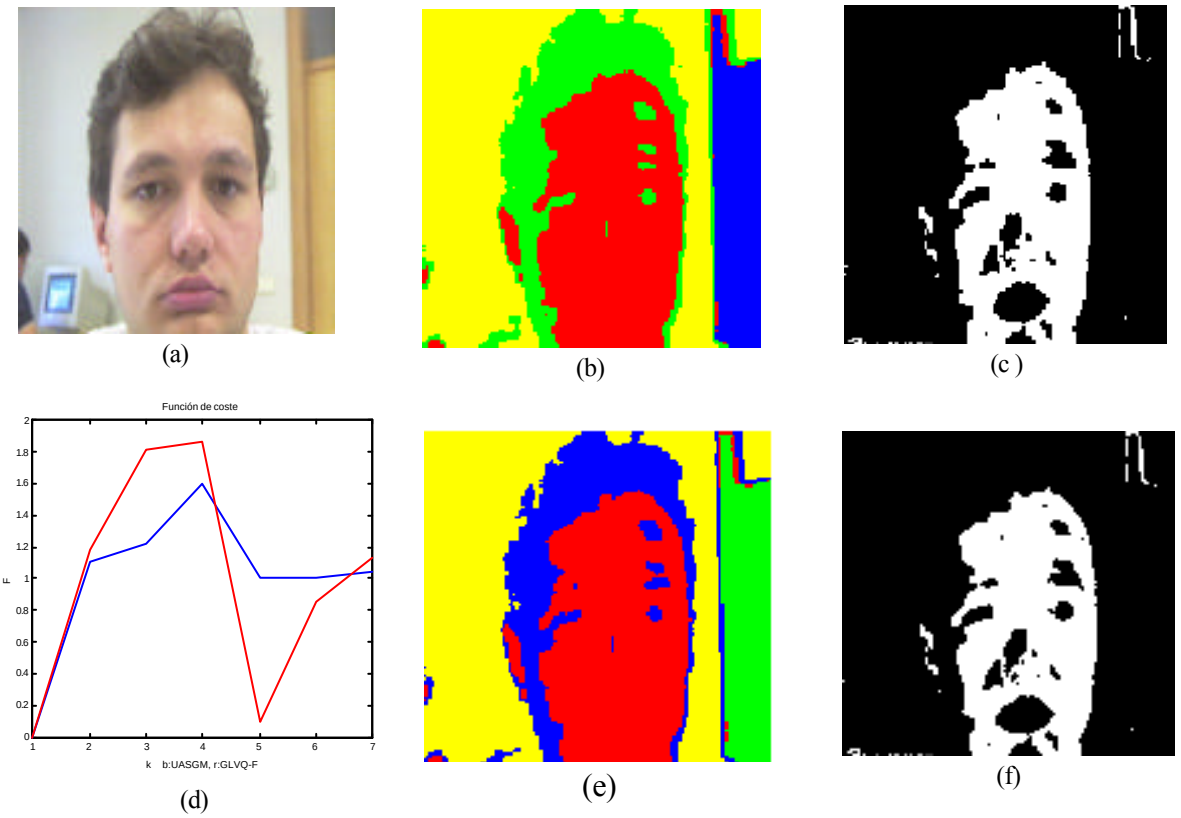


Figura 4.41. Comparación de los métodos UASGM y GLVQ-F para un iris medio
(a) Imagen original, (b) "Clustering" en (x,y) para UASGM, (c)Segmentación para UASGM, (d) Función de coste, (e) "Clustering" para GLVQ-F, (f) Segmentación para GLVQ-F

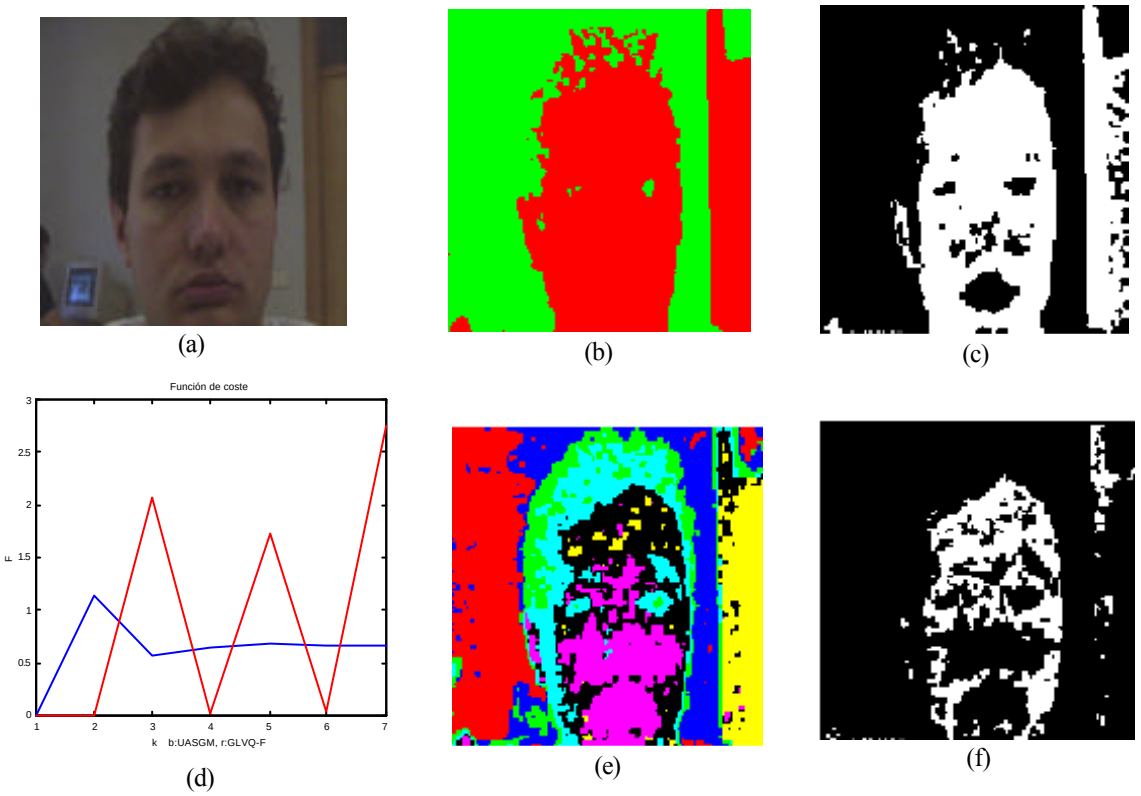


Figura 4.42. Comparación de los métodos GLVQ-F y UASGM para el iris cerrado (a) Imagen original, (b) "Clustering" en (x,y) para UASGM, (c)Segmentación para UASGM, (d) Función de coste, (e) "Clustering" para GLVQ-F, (f) Segmentación para GLVQ-F

Métodos	Error de Segmentación (% de pixel)			Tiempo de computación (T/T_{UASGM})		
Métodos	fig 4.41	fig 4.42	fig 4.43	fig 4.41	fig 4.42	fig 4.43
UASGM	4	5	9	1	1	1
GLVQ-F	4	35	11	1,47	1,58	1,85

Tabla 4.14. Comparación de métodos para el experimento 3

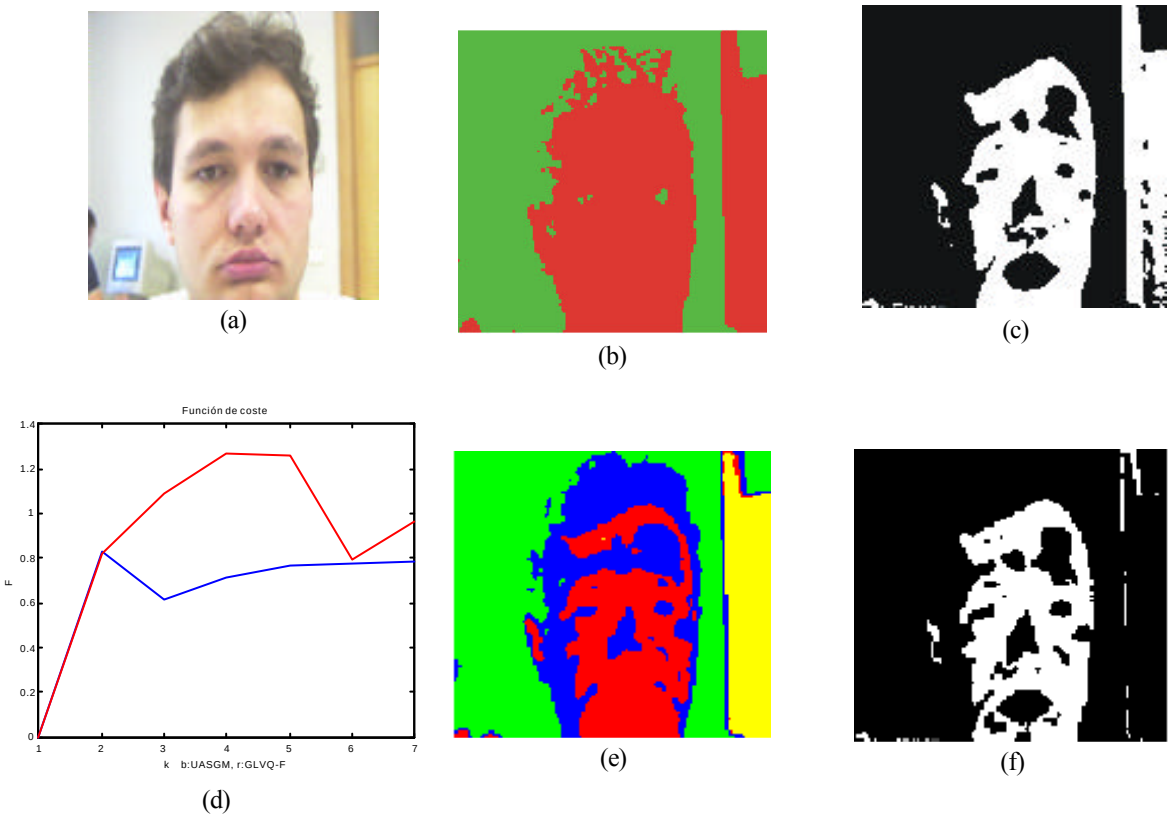


Figura 4.43. Comparación de los métodos GLVQ-F y UASGM para el iris cerrado
(a) Imagen original, (b) "Clustering" en (x,y) para UASGM, (c)Segmentación para UASGM, (d) Función de coste, (e) "Clustering" para GLVQ-F, (f) Segmentación para GLVQ-F

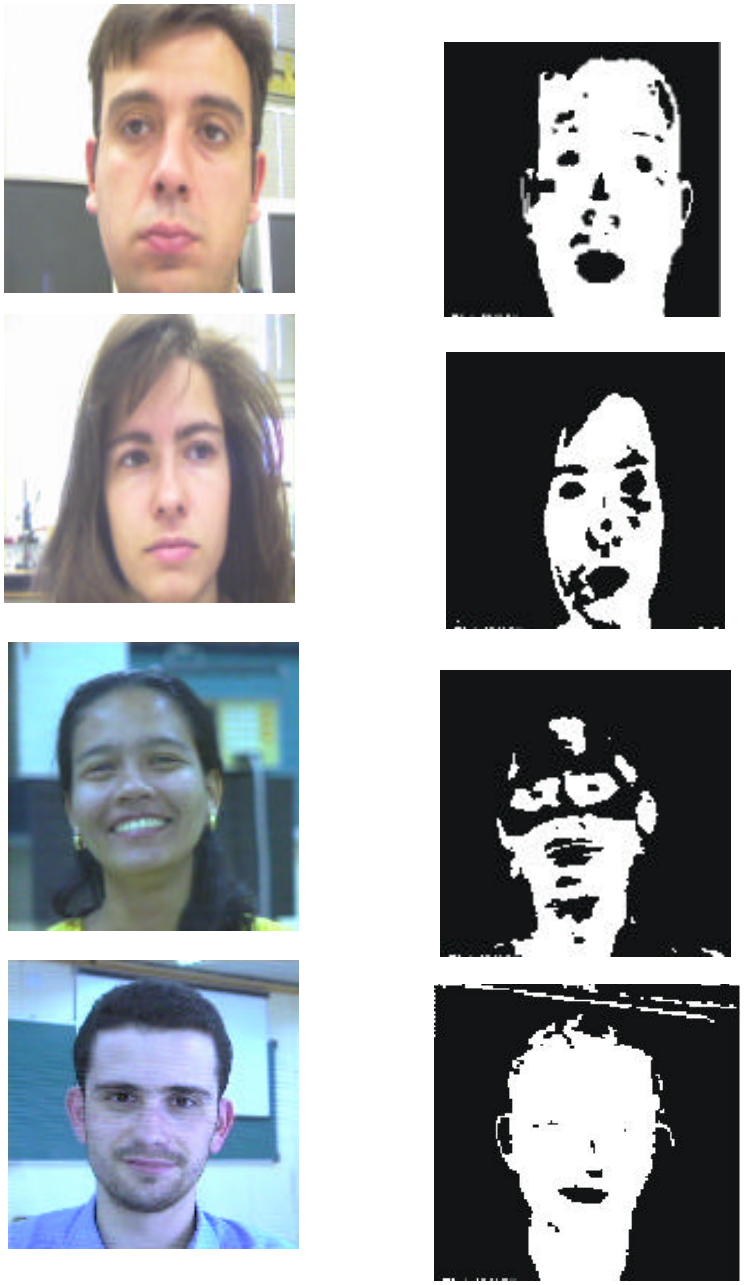
En la figura 4.42, para un iris cerrado, el método UASGM obtiene dos clases mientras que el GLVQ-F obtiene siete, siendo su función oscilante con varios máximos locales. El "clustering" para el primer método detecta como clase piel la piel y además el pelo y la boca. Sin embargo, esta estimación es mejor que la obtenida mediante el método GLVQ-F, en la que la clase piel toma los pixels de la frente y del contorno de la cara, al haberse dividido en un mayor número de clases. La segmentación para el primer método obtiene un error entorno al 5% de los pixels, subiendo al 35% en el segundo método. Además el tiempo de cómputo de éste es mayor que el de aquel.

En la figura 4.43, correspondiente al iris abierto, el método UASGM da dos clases y el GLVQ-F obtiene cuatro. En el primero, se incluye como clase piel parte del marco de la puerta (dado que su crominancia es similar); sin embargo, en la segmentación, el objeto marco no tiene conectividad con el de piel, por lo que se puede eliminar. El segundo método sí que lo detecta, pero en la segmentación obtiene un error un poco por encima del primero.

Por lo tanto, se puede concluir que los resultados con el método UASGM son iguales o mejores que los obtenidos con el GLVQ-F y en un tiempo menor.

Experimento 4

En este caso se presentan algunos otros resultados de segmentación para el método UASGM, donde se puede destacar la robustez ante distintos usuarios, incluso de distintas razas, ante diferentes condiciones de luz y distintos fondos.



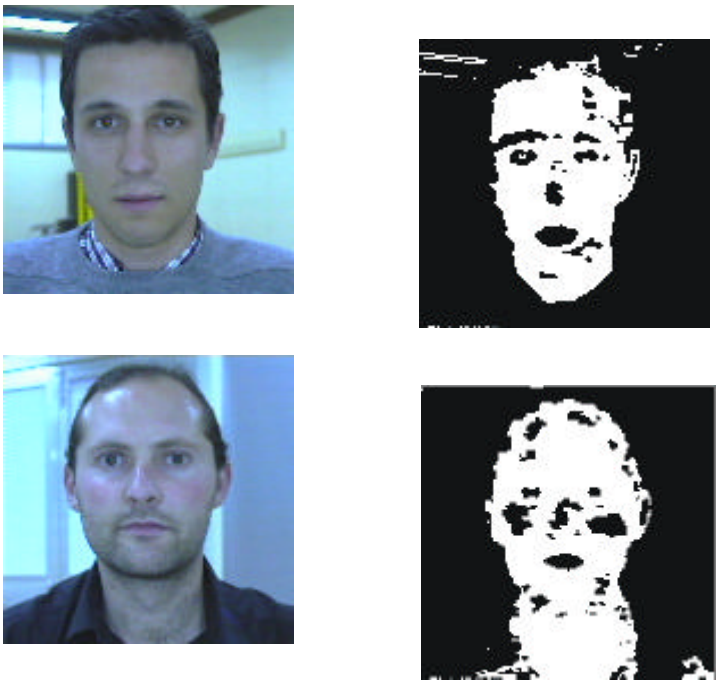


Figura 4.43. Diferentes ejemplos de segmentación

4.7. CONCLUSIONES

En este capítulo se ha realizado un estudio de la segmentación de piel en distintos espacios de color, concluyendo que el espacio "rg" normalizado era el óptimo. Se ha analizado la distribución del color de la piel en el espacio elegido y se ha demostrado que podía modelarse mediante una función gaussiana bidimensional. Se han estudiado las invarianzas del modelo ante distintos usuarios, traslaciones, giros y cambios de iluminación. Para localizar la piel de una persona en una imagen, se ha planteado el estudio del modelado estocástico no supervisado de un histograma, de forma general, y se han introducido una serie de simplificaciones para adaptarlo al problema planteado. Como consecuencia, se ha creado un segmentador de piel llamado UASGM, que es capaz de localizar la piel de una persona, incluso de diferente raza, de forma no supervisada y adaptativa. Para ello, el modelo estocástico se inicializa por un proceso de "clustering", mediante aprendizaje competitivo, basado en el algoritmo VQ , donde el número de clases óptimo se calcula aplicando una modificación del ratio

generalizado de Fisher. Las conclusiones que se pueden sacar respecto al funcionamiento del segmentador son:

1. En cuanto al cálculo del número de clases, mejora o iguala a los métodos: FHV, Evidence, MDL, MML y GMM; invirtiendo un menor tiempo de cálculo.

2. En la mayoría de los casos, los errores de segmentación son menores o iguales para el método UASGM que para el GLVQ-F.

3. El método UASGM tiene un tiempo de ejecución bastante menor que el GLVQ-G

4. La función de coste para el método UASGM es mucho más monótona que para el GLVQ-F, donde la función tiene oscilaciones que dan lugar a la aparición de máximos locales.

5. Se ha demostrado que el sistema obtiene una muy buena segmentación cuando la iluminación de la cara es uniforme y no presenta reflejos o sombras, incluso independientemente de la posición del iris. Por lo tanto, será deseable que la iluminación de la escena sea lo más uniforme posible para que los resultados sean satisfactorios.

6. Por último, decir que, una vez que se segmenta la primera imagen de la secuencia, el modelo se va adaptando al color de la piel del usuario en función del tiempo, de forma que el error de segmentación irá disminuyendo hasta que se estabiliza.

5. GUIADO MEDIANTE SEGUIMIENTO FACIAL

5.1.- INTRODUCIÓN

Una vez segmentado el objeto piel en una secuencia de imágenes, se utiliza esta información para generar una serie de comandos que permitan el guiado de la silla de ruedas. Con el objetivo de hacer un sistema fácil de utilizar y robusto se codificará el mínimo número de comandos que permita el guiado óptimo de la silla. Estos serán: Derecha, Izquierda, Acelerar, Decelerar, Adelante, Atrás, Activo e Inactivo. Para ello se utiliza un modelo geométrico 2D de la cabeza, ojos y boca obtenido de la proyección del objeto real 3D sobre el plano imagen. El sistema se basa en que las características geométricas del modelo cabeza siguen una evolución temporal diferente en función de los distintos movimientos que realice el usuario. Por lo tanto, si inicialmente el sistema aprende el comportamiento del modelo del usuario para cada comando, posteriormente será capaz de determinar los comandos generados por el usuario mediante el análisis de la secuencia de imágenes de su cabeza.

Para llevar a cabo estos objetivos se ha utilizado el seguimiento de un modelo 2D de la cabeza estimando los parámetros del modelo mediante un filtro de Kalman. Posteriormente, se introduce el vector de estado estimado del modelo a una máquina de estados, previamente ajustada para el usuario, que genera los comandos a alto nivel de la silla. Estos comandos se envían a otra máquina de estados

que generará las consignas de velocidad lineal y angular de la silla. Aplicando el modelo cinemático de la misma, estas velocidades se transforman en velocidades angulares para cada rueda y se envían a un módulo de control a bajo nivel, a través de una red de comunicaciones Echelon, donde hay implementados dos controladores PI. En consecuencia, cuando el usuario hace un movimiento de cabeza, el sistema codifica un comando que hace que la silla se mueva. En función del movimiento de la silla detectado por el usuario, es él mismo quien cierra el lazo de realimentación del sistema, generando el mismo comando o bien otro en función de la trayectoria que desea seguir.

5.2.- SEGUIMIENTO DE LA CABEZA

El seguimiento de un objeto segmentado en una secuencia de imágenes es un problema de estimación de un vector de estado variante en el tiempo a partir de observaciones con incertidumbre de dicho vector. Este problema se divide en dos partes igualmente importantes:

1.- Estimación óptima del vector de estado.
2.- Asociación de datos para enlazar el proceso de observación con el de detección.

La estimación óptima del estado implica calcular la mejor estimación posible a partir de una secuencia de medidas. La asociación de datos garantiza la seguridad de que solamente las medidas adecuadas o apropiadas serán consideradas por el estimador.

Supóngase que se desea estimar un vector de estado **x**, y que se dispone de una serie de medidas, **z**, tomadas en tiempos discretos, definidos por el índice n. Las observaciones son una función h(.) del tiempo; así como el parámetro y el ruido asociado a las medidas.

$$z(n) = h[n, x, w(n)] \qquad n = 1,... \tag{5.1}$$

Tomando un conjunto de observaciones K,

$$Z^K = \{z(n-m), m = 1,..., K\} \tag{5.2}$$

el vector de estado, **x** , en un cierto instante n, se puede estimar por:

$$\hat{x}(n) = \hat{x}\left[K, Z^K\right] \tag{5.3}$$

Para solucionar la ecuación anterior se pueden emplear diferentes técnicas:

- Regresiones lineales [Curwen et al.,91]
- Recursiones lineales [Rao, 91]
- Aproximaciones con filtros no lineales [Bar-Shalom&Fortmann, 88]

La técnica de estimación más popular y masivamente empleada en seguimiento visual es el estimador recursivo lineal conocimo como filtro de Kalman. Dicho filtro formula una forma resursiva de estimación lineal por mínimos cuadrados de parámetros variantes con el tiempo, bajo condiciones de ruido aditivo, gaussiano y con media cero. En un sistema de seguimiento por filtro de Kalman se distinguen cuatro fases: observación, validación, estimación y predicción; como se puede ver en la figura 5.1:

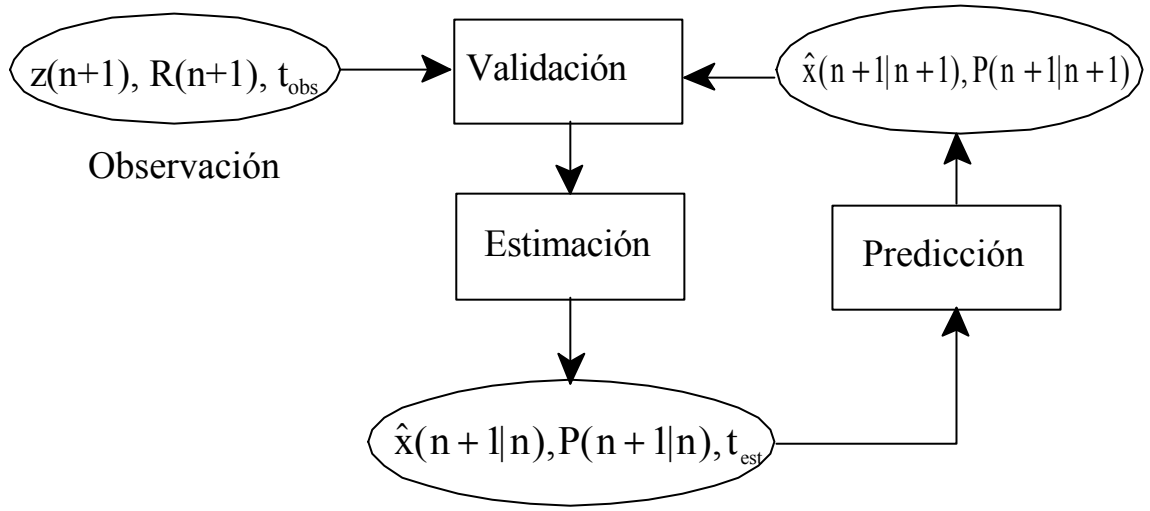


Figura 5.1. Fases del seguidor de cabeza por filtro de kalman

En la figura 5.1, z(n+1), es el vector de muestras observadas de entrada en el instante (n+1), R(n+1) es la matriz de covarianza asociada a la medida, y t_{obs} es una variable temporal que indica el instante de la observación. Asimismo, $\hat{x}(n+1|n)$ es el vector de estado estimado en n para el instante (n+1) en n, P(n+1|n) es su matriz de covarianza asociada y t_{est} indica el momento de cálculo de la estimación. Asumiendo que, de forma general, el vector de estado es variante con el tiempo, existe una diferencia temporal entre el instante en el que se hace la estimación y el momento en el que se introduce en el modelo para validar la siguiente observación. Por lo tanto, hay que introducir una fase de predicción

que estime la variación de la estimación en este intervalo de tiempo. El vector $\hat{x}(n+1|n+1)$ indica la predicción para el instante (n+1) y P(n+1|n+1) será su covarianza asociada.

Las razones de su popularidad son las siguientes:

1.- La estructura recursiva del algoritmo permite que la estimación del estado se ajuste de manera incremental con cada nuevo conjunto de medidas.

2. - La etapa de predicción da una conexión entre la estimación actual y la estimación cuando se hace una nueva observación.

3.- La evaluación de la varianza de la estimación se produce dentro del bucle.

4.- El estimador es óptimo en el sentido Bayesiano (varianza mínima) si el ruido del estado y la observación son gaussianos, en caso contrario se le puede considerar como el estimador lineal óptimo.

Para que el método de estimación sea preciso, es necesario utilizar las medidas relacionadas con el vector de estado objetivo. El proceso de eliminar observaciones falsamente generadas se conoce como *validación*, y el proceso de decidir cuales son las medidas validadas debe usar el estimador se conoce como *asociación de datos*.

A continuación, se van a explicar cada una de las fases del seguidor particularizándolas al problema planteado en esta tesis.

5.2.1. Observación

En esta fase se define el vector de estado a utilizar por el seguidor. La elección de este vector viene fijada por el modelo a utilizar. El objetivo es mantener estimado un modelo de la cabeza del usuario, y mediante el análisis del mismo definir los movimientos de ésta. En este punto existen dos enfoques diferentes:

a. *Los que buscan gran precisión*. Usan modelos complejos de la cabeza en 3D. La

transformación de 2D a 3D de un gran número de puntos implica un tiempo de proceso que empeora las prestaciones finales del sistema o hace necesario plataformas hardware más complejas.

b. *Los que buscan rapidez con menos precisión*. Usan modelos 2D de la cabeza para detectar los movimientos de la misma. Con ello se consigue menor precisión que en el caso anterior, pero mayor rapidez.

En nuestro caso, no se necesita una gran precisión del modelo, ya que el número de comandos a generar es muy limitado, pero es imprescindible que el tiempo de computación sea reducido.
Una solución que ha dado buenos resultados ha sido considerar el vector de estado constante entre la estimación y la observación y, por lo tanto, no trabajar con la derivada del estado. Por otro lado, tomar como vector de estados la entrada del sistema sin ruido, es decir, considerar la estimación por Kalman como un filtro óptimo que elimina el ruido del sistema. Para ello, hay que asumir que el modelo será igual a la matriz identidad: H(n)=I. Bajo estas hipótesis el filtro de Kalman se transforma en un filtro por mínimos cuadrados.

Por lo tanto, las variables del vector de estado coincidirán con las de la entrada del sistema. Se usan como variables de entrada las características geométricas del mayor objeto segmentado en la imagen (objeto cara) siguientes: centro de gravedad (x_{0S}, y_{0S}); y tamaño horizontal y vertical del mismo (h_S, v_S), como se puede ver en la figura 5.2. Se han utilizado dos vectores de estado independientes, el vector de desplazamiento horizontal (x_h) y vertical (x_v) del objeto cara:

$$x_h = \begin{bmatrix} x_{0S} \\ h_S \end{bmatrix} \qquad x_v = \begin{bmatrix} y_{0S} \\ v_S \end{bmatrix} \tag{5.4}$$

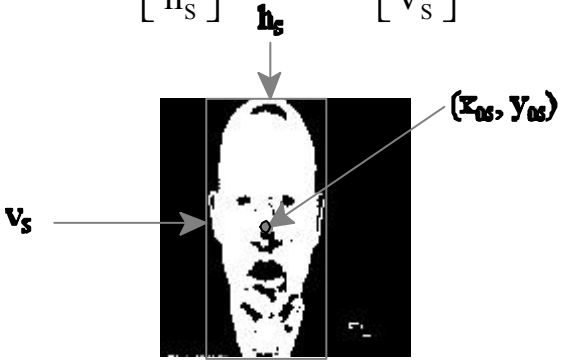


Figura 5.2. Parámetros utilizados en el estimador

El motivo de usar dos vectores de estado en lugar de un único vector de cuatro componentes ha sido hacer más rápido el proceso y evitar calcular matrices inversas de tamaño 4x4. En esta aplicación, los vectores de estado horizontal y vertical son independientes, ya que únicamente están permitidos movimientos en horizontal o en vertical. Sin embargo, se ha demostrado experimentalmente que existe dependencia entre los parámetros x_{OS} y h_S, así como entre y_{OS} y v_S. El tamaño horizontal y vertical de la cara se estima como 2 parámetros independientes ya que la relación de aspecto de la misma cambia con las rotaciones.

La covarianza asociada a la medida (R) se calcula con una ventana de los K últimos datos observados. Así, para una observación se obtienen dos vectores de medidas (z_h ,z_v) compuestos por: los valores del centro de gravedad horizontal y tamaño horizontal de la cara, por un lado; y por otro, el centro de gravedad y el tamaño de la cara verticales. Estas medidas llevarán asociadas dos matrices de covarianza (R_h, R_v) que definirán las variaciones de las variables para las K últimas observaciones.

5.2.2. Estimación y predicción

El filtro de Kalman da una predicción gaussiana del vector de estado. Trabaja con dos tipos de datos: datos medidos con su covarianza asociada (z(n+1), R(n+1)) y datos predichos también con su covarianza asociada $\left(\hat{x}(n+1), P(n+1)\right)$. Dicho filtro estima basándose en una medida de calidad de los datos medidos y predichos en la iteración anterior para la iteración actual. Estas medidas vienen dadas por las matrices de covarianza R(n+1) y P(n+1). Así, si R(n+1) es grande posiblemente habrá datos erroneos en la medida y, por lo tanto, se le dará poco peso a estos datos en la siguiente iteración. Si P(n+1) es pequeña se puede dar gran peso a la predicción, ya que significa que se está prediciendo bien.

Cada vez que se procesa una imagen se produce una observación del vector de estado, z(n-1), junto a una variable temporal, $t_{obs,}$ y una matriz de covarianza, R(n+1). Para actualizar la estimación con la observación se debe primero predecir el valor de la misma al instante de tiempo de la observación. Esta es la tarea que realiza la fase de predicción. En nuestro caso se usa un filtro de Kalman de orden cero, es decir, se considera que las variaciones del vector de estado entre una estimación y la siguiente observación son nulas. Esto es así ya que los movimientos de un sujeto entre observaciones se pueden

dar en cualquier dirección, con lo que no tiene sentido hacer una estimación de derivadas. Por otro lado, el tiempo entre observaciones es muy pequeño (aproximadamente 40 ms) comparado con la secuencia de un movimiento (aproximadamente 1 s) con lo que a la resolución que se está trabajando (diferencias máximas de 50 pixels para los vectores de estado en un movimiento) supone un error máximo de estimación de 2 pixels entre observaciones que serán corregidos en la siguiente estimación.

Por lo tanto, la predicción del vector de estado en t_{obs} será la última estimación en el tiempo t_{est}, que expresado en tiempos discretos queda:

$$\hat{x}(n+1|n+1) = \hat{x}(n+1|n) \tag{5.5}$$

En cuanto a la covarianza de la predicción, depende del intervalo de tiempo transcurrido desde la estimación a la observación:

$$P(n+1|n+1) = P(n+1|n) + \Delta t^2 W \qquad \Delta t = t_{obs} - t_{est} \tag{5.6}$$

donde la incertidumbre en la posición del sujeto crece en función del cuadrado de Δt [Crowley,97]. La matriz W da la pérdida en precisión de cada componente desde la estimación hasta la predicción, y se calcula de forma experimental. Dicho tiempo, en nuestro caso, es del orden de 0,5 ms, lo que hace que el efecto de este término sea despreciable frente a la covarianza estimada y, por consiguiente, no se tiene en cuenta. Así pues, la covarianza predicha coincidirá con la última estimada.

Por lo tanto, dadas las ecuaciones generales de un filtro de Kalman [Brown, 95] en su versión discreta:

$$x(n+1) = F(n)x(n) + G(n)u(n) + v(n) \tag{5.7}$$

$$z(n) = H(n)x(n) + w(n) \tag{5.8}$$

donde (5.7) representa la ecuación de la planta, x(n) es el estado en el tiempo n, u(n) es la entrada o señal de control conocida, v(n) es una señal de ruido gaussiano con media cero y covarianza Q(n). Por otro lado, (5.8) es la ecuación de medidas, donde w(n) es ruido gaussiano de media cero y covarianza R(n). Se asume que inicialmente x(0) es una gaussiana con media $\hat{x}(0|0)$ y covarianza P(0|0) y que los dos procesos de ruido son independientes. Aplicando las simplificaciones introducidas se obtiene un filtro por mínimos cuadrados, según la siguiente ecuación:

$$z(n) = x(n) + w(n) \tag{5.9}$$

de donde se obtienen las siguientes conclusiones:

1.- Se considerará el estado constante en cada iteración y, por lo tanto, no se trabaja con la derivada del estado.

2.- La matriz, H(n)=I, por lo que las variables de medida y de la observación son las mismas.

Por todo ello, a partir de la solución general del filtro de Kalman, dada en la figura 5.3 y aplicando las particularizaciones correspondientes, se obtienen las ecuaciones a aplicar que se muestran en (5.10), (5.11),(5.12),(5.13) y (5.14).

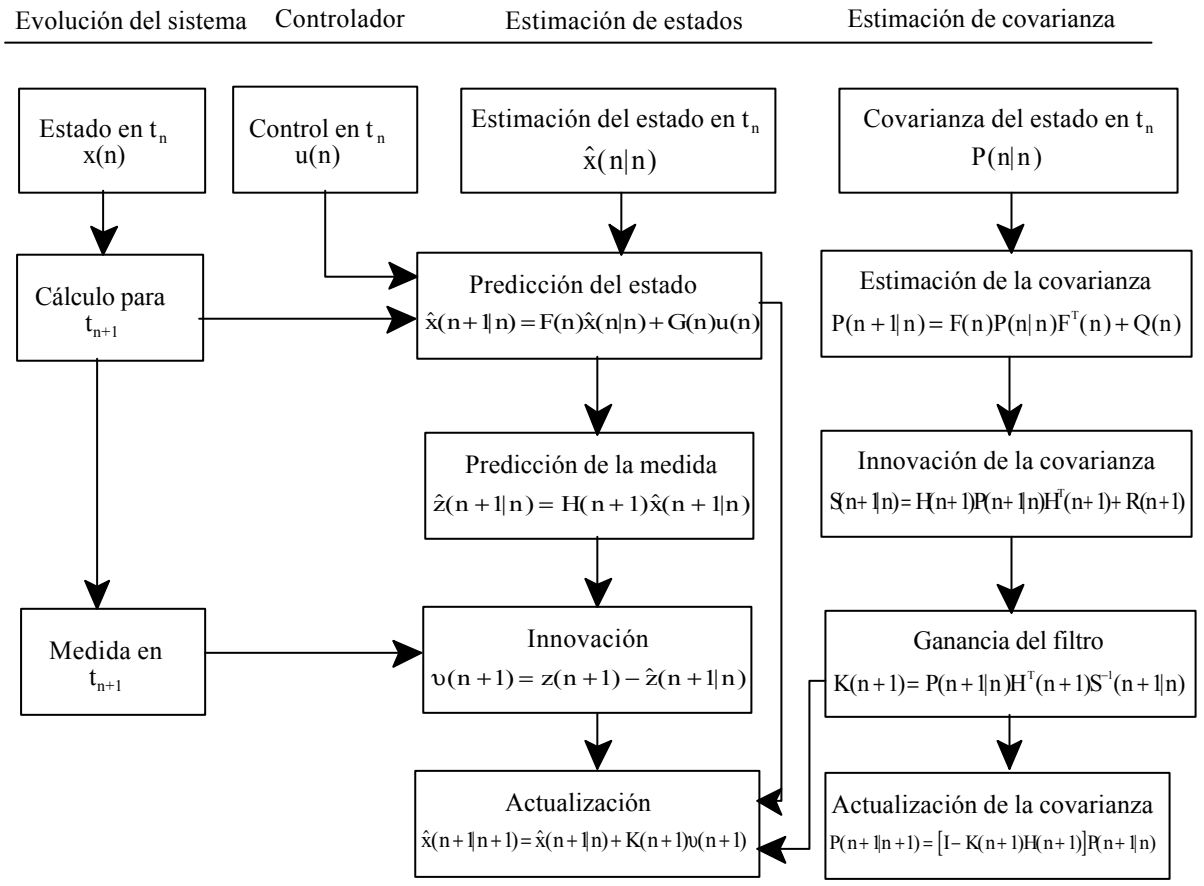


Figura 5.3. Cálculo general de un filtro de Kalman

$$\hat{z}(n+1|n) = \hat{x}(n|n) \tag{5.10}$$

$$K(n+1) = P(n|n)[P(n|n) + R(n+1)]^{-1} \tag{5.11}$$

$$\hat{x}(n+1|n+1) = \hat{x}(n|n) + K(n+1)[z(n+1) - \hat{z}(n+1|n)] \tag{5.12}$$

$$P(n+1|n+1) = P(n|n) - K(n+1)P(n|n) \tag{5.13}$$

$$z_h(n+1) = \begin{bmatrix} x_{0S} \\ h_S \end{bmatrix} \qquad z_v(n+1) = \begin{bmatrix} y_{0S} \\ v_S \end{bmatrix} \tag{5.14}$$

5.2.3. Validación

Para que una estimación sea precisa es necesario que las medidas observadas estén relacionadas con el objetivo planteado. En el filtro de Kalman la validación puede hacerse comparando una observación en el instante t_{obs} (z(n+1)) con la estimación predicha para este instante $(\hat{z}(n+1))$, aceptando únicamente aquellas que se encuentren dentro de un límite de error predefinido a priori. La prueba de validación viene dado por:

$$d(n+1) = \left(\upsilon(n+1)^T S^{-1}(n+1|n)^{-1}\upsilon(n+1)\right) \leq \delta \tag{5.15}$$

con:

$$\begin{aligned} S(n+1|n) &= P(n+1|n) + R(n+1) \\ \upsilon(n+1) &= z(n+1) - \hat{z}(n+1|n) \end{aligned} \tag{5.16}$$

L(n+1) es la innovación y S(n+1|n) es la covarianza de la innovación. La ecuación (5.15) es un "test estadístico cuadrático" en el sentido de que $z(n+1) = \hat{z}(n+1|n)$, pero a menudo se interpreta como la existencia de una "región válida", centrada en el estado de predicción, $\hat{z}(n+1|n)$,en el espacio de observaciones que se encuentra dentro de la región considerada como válida (ver figura 5.4). El parámetro clave de este proceso es el valor de confianza límite,*, el cual es elegido de forma empírica. En nuestro caso se ha ajustado a diez pixels.

Las observaciones utilizadas para actualizar el filtro se extraen a partir del conjunto de medidas

validadas ($Z_V(n)$) definidas por:

$$Z_V(n+1) = \begin{cases} z(n+1) & \text{si} \quad d(n+1) \le \delta \\ \hat{z}(n+1) & \text{si} \quad d(n+1) > \delta \end{cases} \tag{5.17}$$

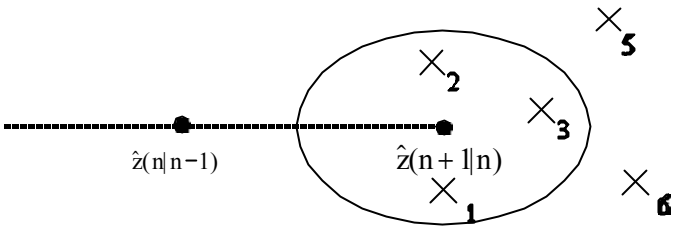


Figura 5.4. Proceso de validación

5.2.4. Resultados

A continuación se presentan diferentes ejemplos de aplicación del estimador diseñado, para los dos vectores de estado considerados y para diferentes movimientos de cabeza. En azul se dan los datos observados y en rojo los estimados.

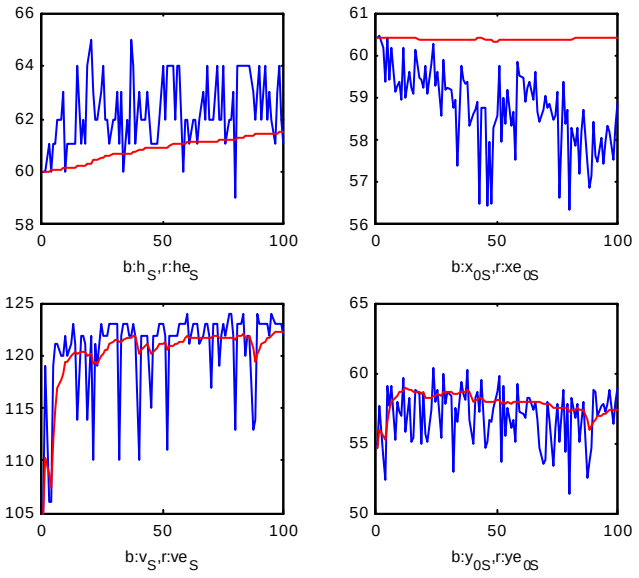


Figura 5.5. Estimación sin realizar ningún movimiento

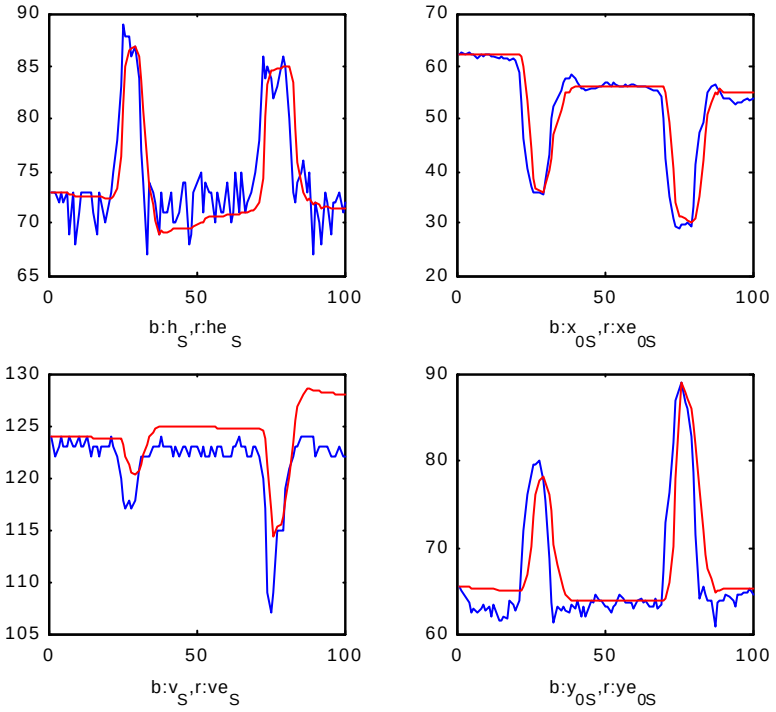


Figura 5.6. Estimación para un movimiento hacia la derecha

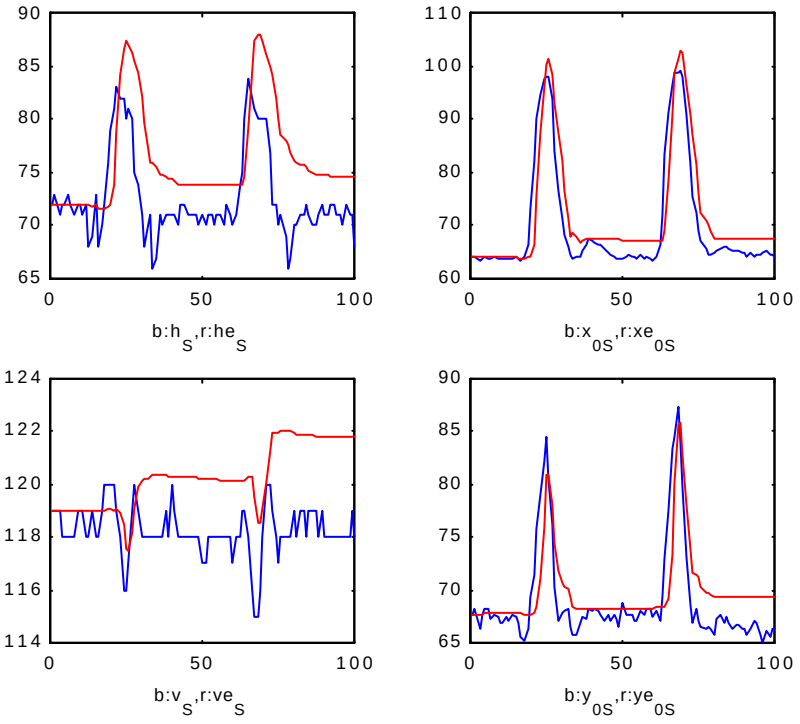


Figura 5.7. Estimación para un movimiento hacia la izquierda

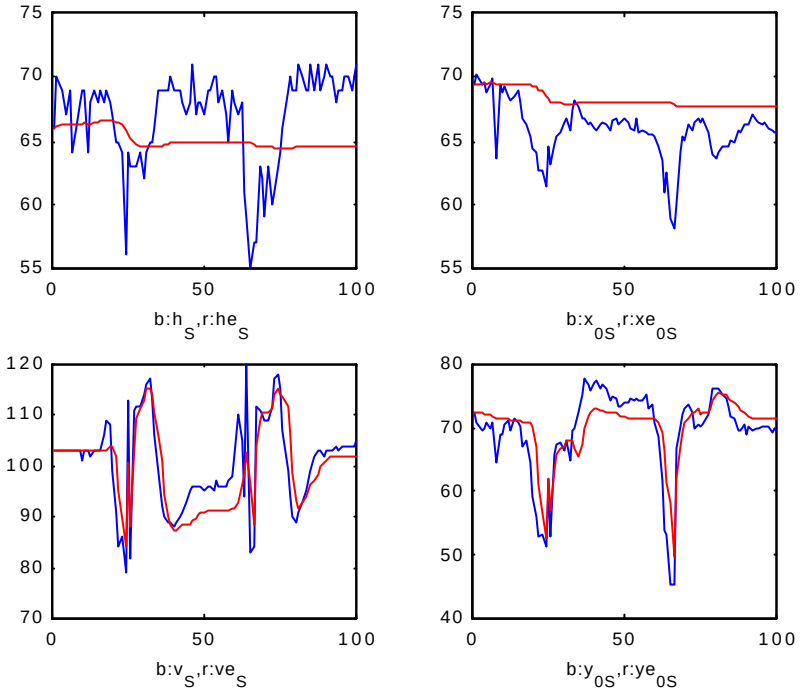


Figura 5.8. Estimación para un movimiento hacia arriba

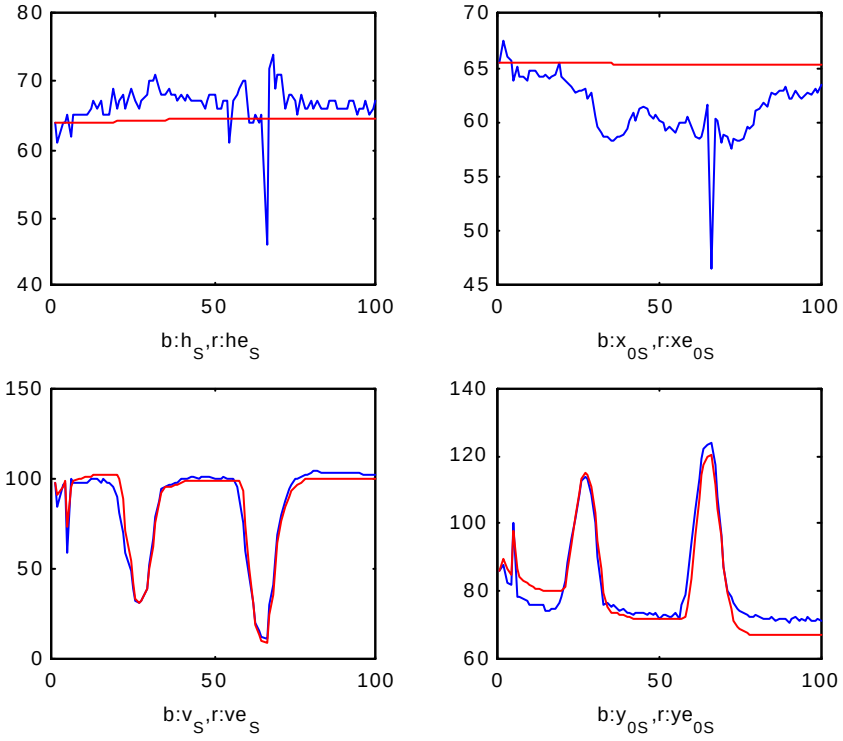


Figura 5.9. Estimación para un movimiento hacia abajo

En la figura 5.5 se da la evolución de las cuatro componentes de los vectores de estado para una secuencia de 100 imágenes y para un usuario que mantiene la cabeza fija delante de la cámara. Como se puede observar, la estimación elimina el ruido del sistema y, después del transitorio inicial del filtro, tanto el vector de estado horizontal como vertical se mantienen muy constantes. El tamaño horizontal sufre una pequeña deriva hacia arriba debida a pequeños movimientos de rotación involuntarios de la cabeza. Asimismo, para este caso, las medidas verticales tienen mayor cantidad de ruido que las horizontales, posiblemente debido a pixels de pelo que se segmentan como piel, al ser su crominancia (marrón claro) similar a la de aquella.

En las figuras 5.6 y 5.7 se muestra la variación de los vectores de estado cuando el usuario realiza dos movimientos con su cabeza hacia la derecha y hacia la izquierda, respectivamente. Observando las gráficas, se concluye que la generación de un movimiento produce una variación en las variables estimadas de más de diez pixels, lo que está por encima del nivel de ruido del sistema. Esta característica será utilizada a posteriori para codificar los diferentes comandos. Como se observa, existe una fuerte dependencia entre las componentes del vector de estado horizontal y vertical. En este caso también existe una cierta dependencia entre el vector de estado horizontal y vertical debido a que cuando un usuario rota su cabeza en horizontal, el cambio de aspecto que experimenta el objeto segmentado piel, también afecta a la componente vertical, al captar con la cámara una vista lateral de la cabeza donde entran en juego el pelo, las orejas y el cuello. Por lo tanto, para este caso no se puede asumir independencia entre los vectores de estado, pero debido a una cuestión de coste computacional, a pesar de ello se han computado como dos vectores independientes.

En las figuras 5.8 y 5.9 se pueden ver los parámetros estimados para un caso en el que el usuario realiza dos movimientos hacia arriba y hacia abajo respectivamente. La estimación filtra las variaciones en horizontal, manteniendo las variables horizontales prácticamente constantes para los movimientos en vertical. La observación vertical de la figura 5.8 aparece muy ruidosa debido a errores del segmentador de piel con el pelo del usuario. Sin embargo, el estimador es capaz de minimizar el ruido grandemente. En este caso, cada movimiento provoca cambios en las variables verticales de más de diez pixels, por encima del nivel de ruido, por lo que también se pueden usar estas acciones para codificar comandos. Hay que destacar que, para estas pruebas, prácticamente no existe dependencia entre el vector de estado horizontal y vertical, aunque sí entre sus componentes.

5.3.- GENERACIÓN DE COMANDOS

El objetivo planteado consiste en realizar una serie de movimientos con la cabeza, ojos o boca, a los que llamaremos *acciones.* A éstas se les asocia unos *comandos* que permitan mover una silla de ruedas. Teniendo en cuenta las personas a las que va dirigido este sistema de guiado, las acciones deben ser fácilmente realizables por una persona con minusvalías y además deben ser robustas, para minimizar las falsas activaciones de comandos. Además se debe procurar que las acciones tengan cierta relación con los comandos a los que dan lugar.

Se eligió un reducido número de acciones que no conllevaran dificultad para el usuario, y que el sistema pudiera diferenciar e identificar sin demasiados problemas. Hay que tener en cuenta que el usuario no va a tener la destreza de repetir exactamente igual una acción varias veces , por lo que es más importante tener pocas acciones que generen comandos generales que muchas acciones que den lugar a comandos muy específicos.

Con estas ideas y analizando los vectores de estado del seguidor se vio que se podían identificar de una forma robusta las siguientes acciones o movimientos de cabeza: *arriba, abajo, derecha, izquierda, adelante y detrás*. Otras acciones intermedias entre éstas, como por ejemplo, arriba-derecha o abajo-izquierda, fueron rechazadas, ya que aumentaban considerablemente la tasa de fallos cuando el sistema estaba en movimiento. Por otro lado, las acciones de adelante y atrás también fueron rechazadas, puesto que la realización de estos movimientos con la cabeza, no era una tarea sencilla para personas con movilidad reducida en el tronco, al tener que incorporarse hacia adelante y hacia atrás.

Con esta filosofía de sencillez y robustez, la realización de un sistema de control para la silla requiere de un mínimo número de comandos que se pueden dividir en tres niveles diferentes, como se muestra en la figura 5.10.

- *Comandos de Activo/Inactivo.* Serán los comandos de mayor nivel y definen si el sistema de guiado se encuentra o no activo. Pueden ser considerados como un conmutador que cambia de un estado a otro y por lo tanto es suficiente con asociarle una única acción que conmute entre el estado de Activo e Inactivo. Además debe ser la acción más simple de realizar y la

más robusta ya que en caso de emergencia se puede utilizar para detener el sistema.

Figura 5.10. Niveles de comandos

• *Comandos de Sentido*. Se encuentran por debajo del nivel de funcionamiento en la jerarquía de comandos. Son necesarios dos comandos que permitan definir el sentido del movimiento, es decir, si el movimiento se realiza hacia delante o hacia atrás. Al igual que en el caso anterior, estos comandos son excluyentes y por lo tanto es suficiente con asociarles una única acción que les haga conmutar. Debe ser una acción fácil de realizar aunque puede ser menos robusta que en el caso anterior, puesto que se impone como condición para la conmutación que la silla se encuentre parada, para que no pueda haber cambios bruscos de dirección, por lo que, en caso de fallo se puede repetir la acción desde parado hasta que se detecte correctamente.

• *Comandos de velocidad*. Serán los que fijen la velocidad lineal y angular del móvil y son los de menor nivel jerárquico. Existen dos posibilidades: velocidades proporcionales a parámetros de los vectores de estado de las acciones, o bien, velocidades fijas por eventos. El primer tipo permite un mejor control del móvil, ya que es el usuario quien define exactamente la velocidad que desea llevar y cómo se producen las aceleraciones. Sin embargo, en un sistema como el planteado tiene el problema de que el usuario no tiene una referencia absoluta de cuál es el movimiento a realizar con la cabeza para provocar un determinado incremento de velocidad y, aunque lo tuviera, en una silla en movimiento sería difícil posicionar la cabeza con exactitud. Es por ello que se ha elegido la segunda opción, en la que se realizan una serie de acciones, que activan ciertos comandos, y a cada uno de ellos se le asocia una velocidad fija. Nótese que los joysticks de las sillas comerciales funcionan según este esquema.

La cuestión ahora es: ¿cuántos comandos utilizar? Parece lógico pensar que se necesita uno para girar a la DERECHA y otro para girar a la IZQUIERDA, haciéndolo en una primera aproximación a velocidad constante. Por otro lado, con los comandos de dirección se tiene resuelto el sentido a seguir, pero hay que definir la velocidad lineal del móvil. Para ello son necesarios otros dos comandos que permitan ACELERAR Y DECELERAR el móvil. No hay que olvidar que también se necesita un comando de "NO OPERACIÓN" que no desencadena ningún movimiento. Por lo que con cinco comandos se puede tener controlada la velocidad lineal y angular de la silla. En cuanto a las acciones asociadas deben estar relacionadas con los comandos a activar y por lo que respecta a su robustez tienen unas restricciones mayores que las de dirección pero menores que las de Activo/Inactivo.

Una vez establecidos los requerimientos de comandos mínimos necesarios, el siguiente paso es asociar a cada uno una acción. Se empezará por el nivel de velocidades:

DERECHA. Parece lógico pensar que la acción que desencadene un giro hacia la derecha sea un movimiento de cabeza hacia la derecha. De esta forma la silla girará en el sentido donde se tiene fijada la mirada hasta que el usuario vuelva a colocar su cabeza en la posición de mirada frontal.

IZQUIERDA. Desencadena un giro de la silla hacia la izquierda y se activa haciendo un movimiento de la cabeza en esta dirección. Al igual que antes, la silla se mueve hasta que el usuario coloque su cabeza en posición frontal.

ACELERAR. Produce un aumento de la velocidad. Se le puede asociar la acción de mirar hacia arriba, ya que de esta forma el usuario levanta su mirada y mira hacia puntos más alejados de la silla dando a entender que quiere ir más rápido. La silla se acelera mientras se esté mirando hacia arriba. En este caso se aplica el concepto de las luces largas y cortas de un coche en conducción nocturna. Si se quiere ir deprisa se ponen las largas para ver más (mirada hacia arriba) y si se quiere ir despacio no es necesario tener tanto campo visual (mirada hacia abajo).

DECELERAR. Produce una disminución de la velocidad y está asociada a la acción de mirar hacia abajo, siguiendo el concepto explicado. La silla se decelera mientras se mantiene la acción hasta que ésta desaparece.

Con esto se cubre el nivel más bajo de comandos, sin embargo se necesitan generar nuevos comandos y ya no quedan acciones disponibles. Para cubrirlas, según los requisitos ya nombrados, se ha acudido a localizar características faciales como la boca y los ojos. Es evidente que en una imagen frontal de una cara es fácil localizar los ojos y la boca de una persona dentro del objeto piel, al ser características muy contrastadas y con un modelo geométrico fijo. El problema radica cuando se tiene imágenes laterales, ya que el modelo cambia y alguna de las características puede desaparecer del campo visual.

Buscando la mayor sencillez del sistema se han añadido las dos siguientes acciones:

• *Detección de labios ocultos en imagen frontal*. Experimentalmente se observó que en el interior del objeto "piel segmentada" aparecía como un objeto bien diferenciado la boca del usuario (ver figura 5.2). Esto es debido a que el color de los labios y de la boca difiere considerablemente del color de la piel en el espacio analizado, por lo que se puede detectar sin dificultad su presencia. La acción consiste en apretar los labios de forma que queden escondidos por la piel que rodea a la boca, así se elimina el objeto boca de la imagen. Así, introduciendo un sistema que sea capaz de hacer un seguimiento de la boca, en el momento que desaparezca el objeto, significará que se quiere activar un comando. Únicamente hay que tener en cuenta que cuando se realiza cualquier acción se corre el riesgo de que parte de la boca desaparezca, con lo que se pueden introducir falsas activaciones. Para evitarlo, se permite la activación de esta acción sólo cuando el usuario se encuentre en posición de no operación, es decir, mirando al frente.

• *Detección de ojo cerrado en imagen frontal*. Otros de los objetos fácilmente identificables por su diferencia de color con la piel dentro de la cara, son los ojos. Éstos también aparecen como dos agujeros en el objeto segmentado piel (ver figura 5.2) y son simétricos respecto al centro de la cara, en imágenes frontales, por lo que pueden ser fácilmente localizables. La acción consiste en guiñar un ojo, con lo que diseñando un sistema de seguimiento de los mismos, en el momento que uno de ellos desaparezca se considerará que el usuario desea ejecutar una acción. La localización de los ojos se complica cuando el usuario rota su cabeza, ya que su modelo geométrico cambia e incluso puede llegar a desaparecer de la escena. Es por ello que tan solo se permite esta acción cuando el usuario esté mirando al frente. En este caso hay otro "handicap" y es el parpadeo. Una persona parpadea aproximadamente una vez cada 5 s., lo que supone que en el sistema de seguimientos de los ojos, hay veces que se detecten los ojos cerrados, lo que se manifiesta como ausencia de agujeros en el objeto segmentado piel, al estar cubiertos por los párpados y por lo tanto considerar estas zonas como de

piel. Teniendo en cuenta que la duración de un parpadeo es corta (< 1s) hay que estimar la posición de los ojos en estos casos y activar la acción cuando el usuario tenga un ojo cerrado un tiempo bastante mayor que el del parpadeo.

Una vez elegidas las dos nuevas acciones hay que asociarles un comando. Debido a que el comando ACTIVO/INACTIVO es más prioritario, se le asocia la acción más robusta, es decir, la detección de labios ocultos en imagen frontal, mientras que al comando de SENTIDO (ADELANTE/ATRÁS) se le asocia la acción de guiñar un ojo, al haberse demostrado que es menos robusta debido a que son objetos de menor número de pixels que la boca y, por lo tanto, son menos inmunes al ruido del sistema y por otra parte debido a las interferencias que suponen los parpadeos.

En la figura 5.11 se muestran las distintas acciones del sistema, junto con la imagen segmentada de piel utilizada para detectarla y los comandos asociados a cada una de ellas. Como se ve, en este caso, el usuario aparece con una gorra. Esto ocurre porque el usuario es calvo y su modelo de piel difiere respecto al de una persona normal. Como se explicará más adelante, de esta forma se consigue uniformizar los modelos de piel de los usuarios.

A continuación se explicará la forma de generar los comandos a partir de los vectores de estado. La idea consiste en evaluar las variaciones de los vectores de estado, de una forma borrosa, respecto a los valores que tienen para la posición de mirada al frente (NO OPERACIÓN). Analizando dichas variables para personas, hombres o mujeres, con el pelo recogido y sin barba se obtienen las conclusiones que se muestran en la tabla 5.1, donde I indica que la variable no cambia, A que aumenta, D que disminuye y X que depende del usuario y de las condiciones de iluminación.

	h_s	x_{0S}	v_S	y_{0S}
No operación	I	I	I	I
Derecha	A	D	D	A
Izquierda	A	A	D	A
Arriba	I	I	A	X
Abajo	I	I	D	A

Tabla 5.1. Variación borrosa de las variables de estado con las acciones

La generación de comandos se produce mediante una máquina de estados que define el estado (comando) en el que se encuentra el sistema en función del tiempo, como se muestra en la figura 5.12.

Figura 5.11. Ejemplo de acciones que generan los distintos comandos

Inicialmente se parte del estado NO OPERACIÓN y se conmuta a uno u otro en función de los valores de las variables de estado (x_h, x_v), de sus derivadas $(\dot{x}_h, \dot{x}_v)$, de una posición de referencia de los vectores de estado (x_{hR}, x_{vR}), correspondiente a sus valores para la posición de NO OPERACIÓN, de unos umbrales calculados para cada usuario a priori y de dos variables de detección de ojos y boca (d_{ojos}, d_{boca}) (ver figura 5.13). Estas últimas controlan la activación de los comandos de SENTIDO y ACTIVO/INACTIVO y serán explicadas en los puntos 5.4 y 5.5. A continuación, únicamente nos centraremos en la generación de los comandos de velocidad.

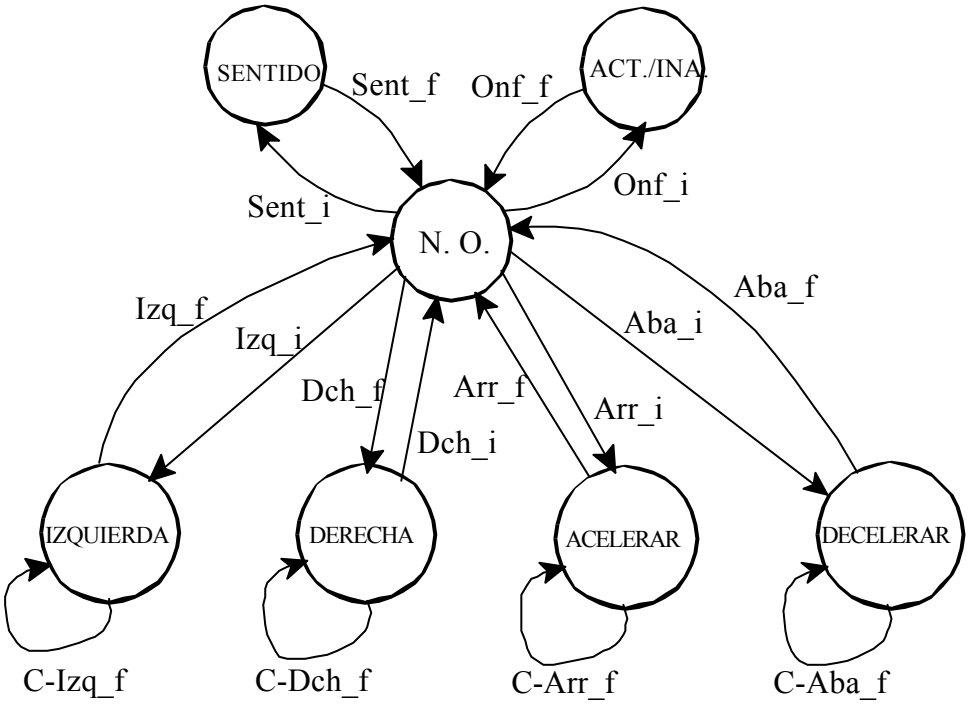


Figura 5.12. Máquina de estados generadora de comandos

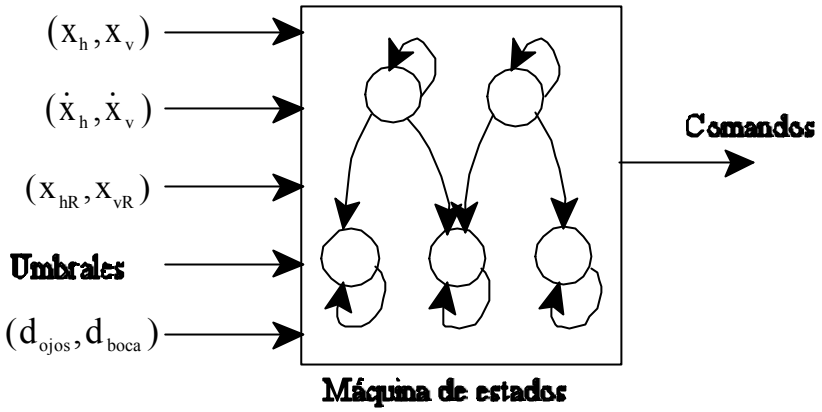


Figura 5.13. Variables de la máquina de estados

La conmutación de estados se realiza por eventos de las variables de estado. Un evento se activa cuando una variable supera un umbral de activación establecido a priori. Como ya se ha comentado, los eventos pueden ser: no modificación, aumentar y disminuir. De esta forma se producen una serie de eventos discretos a partir de variables continuas que se codifican como sigue: no modificación=0, aumento=1, disminución=-1.

Una buena forma de detectar las distintas acciones es trabajar con las derivadas del estado. Estudiando la evolución de estas funciones derivadas se puede calcular el estado del sistema (ver figura 5.14). Para obtener los eventos en las derivadas de las variables de estado de una forma óptima, se calculan unos umbrales de activación a medida del usuario. Dichos umbrales se obtienen en una fase de entrenamiento inicial del sistema. Se calculan dos umbrales para cada variable de la derivada del vector de estados: uno para evaluar su incremento positivo y otra para evaluar su incremento negativo. En esta fase el usuario debe realizar unos movimientos extremos con su cabeza, de aproximadamente 90º hacia la derecha e izquierda, y de 60º hacia arriba y abajo. Se calculan los valores máximos de sus derivadas y se fijan los umbrales de detección de derivadas como un 10% del valor máximo calculado en esta fase. En las figuras 5.14 y 5.15 se muestran dos ejemplos de entrenamiento para dos usuarios distintos.

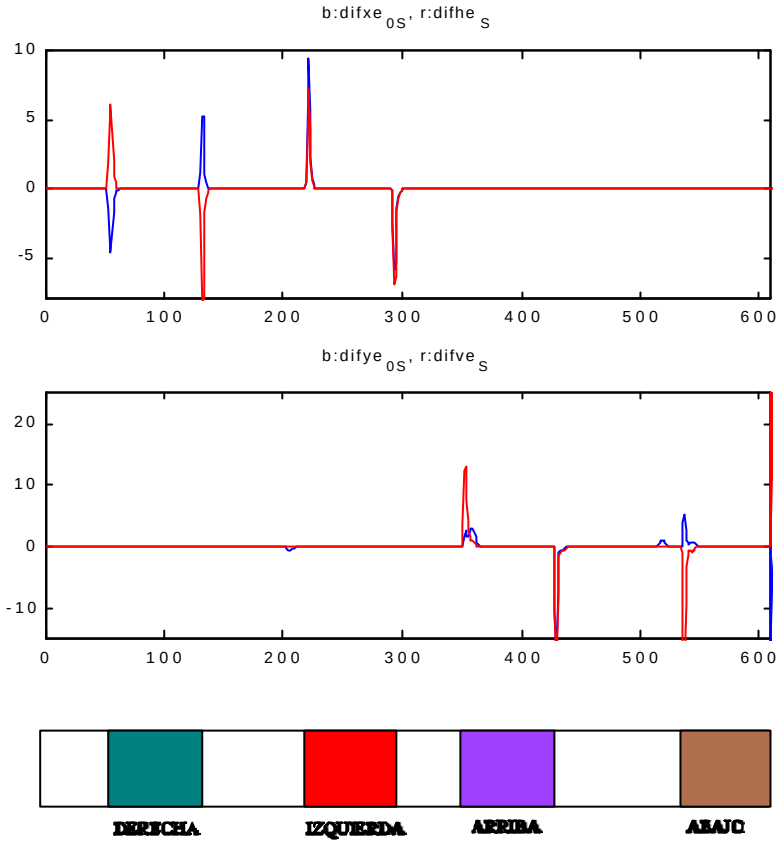


Figura 5.14. Fase de entrenamiento para el usuario 1

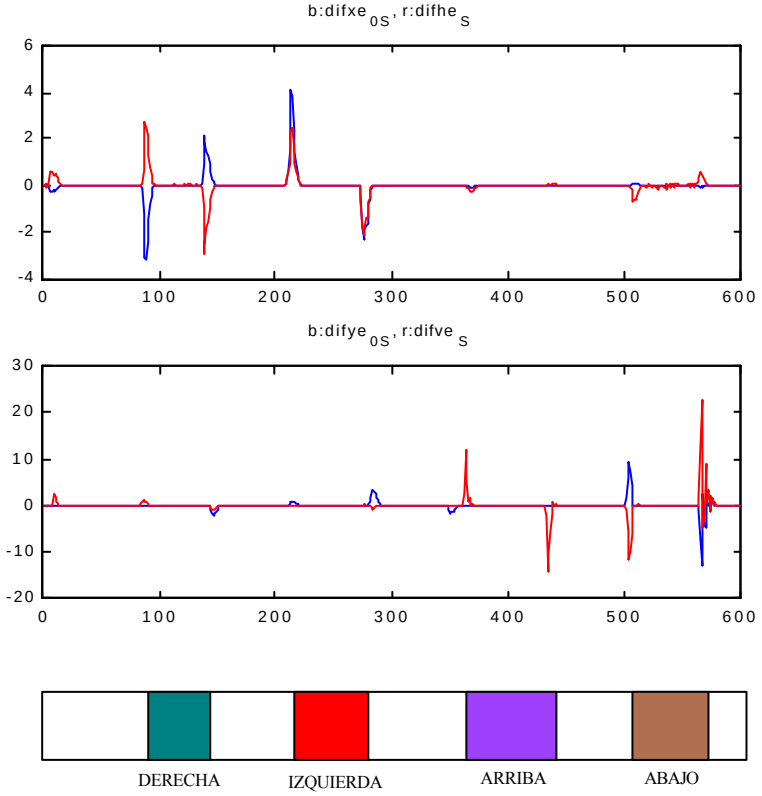


Figura 5.15. Fase de entrenamiento para el usuario 2

Obsérvese que en ambos casos los eventos de inicio y finalización de comandos son los mismos excepto en el comando de ARRIBA, donde la derivada del centro de gravedad "y" para el usuario 1 aumenta en la activación y disminuye en la desactivación y, sin embargo, para el usuario 2 disminuye en la activación y no se modifica en la desactivación. Además hay un pequeño desfasaje en la activación respecto al aumento de la derivada del tamaño vertical. Ello es debido a que al mirar hacia arriba el objeto segmentado piel está formado por la cara y el cuello (ver figura 5.11) viéndose muy influenciado por la iluminación de la escena, ya que crea reflejos sobre la cara que pueden hacer disminuir el centro de gravedad al aparecer áreas de piel con reflejo que no van a ser consideradas piel. Esta situación es función del usuario y de su posición dentro de la escena y, por consiguiente, los valores de esta variable pueden aumentar, disminuir o quedarse como estaban. Estas conclusiones fueron corroboradas mediante la realización de diversos experimentos con distintos usuarios.

Los umbrales se nombran según el siguiente criterio:

TxM= umbral positivo para el centro de gravedad x
Txm= umbral negativo para el centro de gravedad x
TyM= umbral positivo para el centro de gravedad y
Tym= umbral negativo para el centro de gravedad y
ThM= umbral positivo para el ancho
Thm= umbral negativo para el ancho
TvM= umbral positivo para el alto
Txm= umbral negativo para el alto

y la codificación de una variable cualquiera, v, se realizará de la siguiente manera:

$$\begin{aligned} &\text{if}(\Delta v > TvM) \;\rightarrow\; \Delta v = 1 \\ &\text{if}(Tvm < \Delta v < TvM) \;\rightarrow\; \Delta v = 0 \\ &\text{if}(\Delta v < Tvm) \;\rightarrow\; \Delta v = -1 \end{aligned} \qquad (5.18)$$

Como se observa en la figura 5.14, detectando los instantes en los que las derivadas de las variables de estado superan el umbral de disparo, se detectan los inicios y fin de las activaciones de los comandos. Así, si la derivada del centro de gravedad "x" se activa negativamente ( ) x_{OS}=-1) y la horizontal positivamente ( ) h_S=1), se pasa al estado DERECHA y se activa este comando. En este estado, si se activa la derivada del centro de gravedad "x" y positivamente ( ) x_{OS}=1), y la horizontal

negativamente ( ) h_S=-1) se vuelve al estado de NO OPERACIÓN. Si se activa la derivada del centro de gravedad "x" y la horizontal positivamente ( ) x_{OS}=1,) h_S=1), se activa el comando IZQUIERDA hasta que la derivada del centro de gravedad x se active negativamente ( ) x_{OS}=-1) y la horizontal también ( ) h_S=-1), lo que significa que se vuelve al estado de NO OPERACIÓN. Como se puede observar, para la detección de estos comandos únicamente se ha tenido en cuenta el vector de estado horizontal. Esto es debido a que, aunque también se producen variaciones en el vertical, éstas son menores y más dependientes del usuario que las primeras, por lo que se decidió no considerarlas. Las ecuaciones 5.21 y 5.22 indican estos hechos.

$$\begin{aligned} &\text{Der\_i} \rightarrow \text{if}((\Delta x_{os} = -1)\&(\Delta h_s = 1)) \\ &\text{Der\_f} \rightarrow \text{if}((\Delta x_{os} = 1)\&(\Delta h_s = -1)) \end{aligned} \tag{5.19}$$

$$\begin{aligned} &\text{Izq\_i} \rightarrow \text{if}((\Delta x_{os} = 1)\&(\Delta h_s = 1)) \\ &\text{Izq\_f} \rightarrow \text{if}((\Delta x_{os} = -1)\&(\Delta h_s = -1)) \end{aligned} \tag{5.20}$$

Si la derivada del centro de gravedad "y" se activa positivamente ( ) y_{OS}=1) y las variables horizontales no se activan, estando en el estado de NO OPERACIÓN se pasa al estado de ARRIBA. Este estado se mantiene hasta que la derivada del centro de gravedad "y" se activa negativamente ( ) y_{OS}=-1), no habiéndose activado las variables horizontales (ver ecuación (5.21)). En el caso de que la derivada del centro de gravedad "y" se active positivamente ( ) y_{OS}=1) y el tamaño vertical negativamente ( ) v_S=-1), sin que se activen las variables horizontales, se genera el comando ABAJO, hasta que la derivada del centro de gravedad "y" se active negativamente ( ) y_{OS}=-1) y el tamaño vertical positivamente ( ) v_S=1) (ver ecuación (5.24)).

$$\begin{aligned} &\text{Arr\_i} \rightarrow \text{if}((\Delta v_s = 1)\&(\Delta x_{os} = 0)\&(\Delta h_s = 0)) \\ &\text{Arr\_f} \rightarrow \text{if}((\Delta v_s = -1)\&(\Delta x_{os} = 0)\&(\Delta h_s = 0)) \end{aligned} \tag{5.21}$$

$$\begin{aligned} &\text{Aba\_i} \rightarrow \text{if}((\Delta y_{os} = 1)\&(\Delta v_s = -1)\&(\Delta x_{os} = 0)\&(\Delta h_s = 0)) \\ &\text{Aba\_f} \rightarrow \text{if}((\Delta y_{os} = -1)\&(\Delta v_s = 1)\&(\Delta x_{os} = 0)\&(\Delta h_s = 0)) \end{aligned} \tag{5.22}$$

Un modelo de activación de comandos, basado en derivadas y umbrales, implica que hay que realizar los movimientos con una cierta rapidez para superar los umbrales de activación de los comandos. De esta forma, se filtran los movimientos de baja frecuencia involuntarios del usuario, impidiendo que se activen comandos falsos. La rapidez de éstos se elige mediante los niveles de umbral en la fase de

entrenamiento.

Experimentalmente se comprobó que este sistema fallaba si el usuario realizaba una activación correcta del comando pero en la desactivación del mismo hacía un movimiento muy lento que no disparaba la condición de fin de comando. En este caso, el sistema interpreta que se encuentra en un estado de comando activado y espera una condición de fin que no se da, ya que el usuario se encuentra en posición de NO OPERACIÓN. El sistema continuará en este estado hasta que no le llegue una condición de fin válida. Otro caso de fallo se da si no se activa la condición de inicio de comando pero el usuario ha realizado el movimiento, ya que el sistema considera que el usuario está mirando al frente cuando en realidad está mirando a uno de los puntos cardinales que generan activación. Asimismo, un ruido de alta frecuencia del sistema puede provocar una transición errónea de comandos.

Para hacer más robusto el sistema se utilizan, además de las derivadas del estado, las variables de estado del mismo. La idea consiste en añadir una supervisión de baja frecuencia sobre el modelo de derivadas, evaluando las variaciones de las variables de estado respecto a su posición de NO OPERACIÓN. Se utiliza el valor medio de las variables evaluadas sobre una ventana de longitud M, lo cual se indica añadiendo a la variable el subíndice M (v_M). Experimentalmente se ha tomado un valor M=3.

Los valores de referencia de las variables en la posición de NO OPERACIÓN no pueden ser calculados a priori en la fase de entrenamiento, ya que se observó que cuando un usuario realiza un movimiento y luego regresa a la posición de mirada al frente, nunca vuelve exactamente a la misma posición, llegando a haber grandes diferencias con la posición que tenía antes de realizar el movimiento. Es por ello que estos valores deben ser continuamente actualizados en la posición de NO OPERACIÓN. Para ello se utiliza una ventana de los R últimos valores evaluados en el estado de NO OPERACIÓN. Estas variables se indican con el subíndice R y experimentalmente se obtuvo buenos resultados con un valor de R=20.

En cada condición de inicio de transición se comprueba el incremento de las variables de estado respecto a las de referencia. Si este incremento es superior al 90% del calculado en la fase de entrenamiento, se realizará una transición de estado, independientemente de que el modelo de derivadas la haya o no activado. Asimismo, en las condiciones de fin de transición también se hace la misma comprobación y si los incrementos son menores del 10% del calculado en el entrenamiento se

vuelve al estado de NO OPERACIÓN, independientemente de los valores de las derivadas. Por lo tanto, las condiciones de transición de estados quedarán como se indica en la ecuación (5.23).

En la figura 5.15 se muestra un ejemplo en el que en la muestra 550 se activa el comando ARRIBA. Como se observa, el incremento es muy pequeño, pero a pesar de ello el sistema lo detecta. Este comando se desactiva en la muestra 680, sin embargo, las derivadas no alcanzan el umbral y así pues, en el modelo únicamente con derivadas, el sistema busca una condición de terminación para el comando de ARRIBA pero no la encuentra, cometiendo de este modo un error. Las variables de estado (en color verde y rojo) y los valores de referencia (en magenta y amarillo) difieren muy poco, por lo que si se aplica el modelo completo, la condición de incremento mínimo saca al sistema del estado de ARRIBA y lo pasa al estado de NO OPERACIÓN, con lo que la siguiente activación del estado ABAJO será calculada.

$$
\begin{aligned}
&\text{Izq\_i} \rightarrow \text{if}\Big(((\Delta x_{OS}=1)\&(\Delta h_S=1))\#\Big((x_{OSM}-x_{OSR})>0.9\text{Tx}\Big)\&\Big((h_{OSM}-h_{OSR})>0.9\text{Th}\Big)\Big)\Big)\\
&\text{Izq\_f} \rightarrow \text{if}\Big(((\Delta x_{OS}=-1)\&(\Delta h_S=-1))\#\Big(|x_{OSM}-x_{OSR}|<0.1\text{Tx}\Big)\&\Big(|h_{OSM}-h_{OSR}|<0.1\text{Th}\Big)\Big)\Big)\\
&\text{Der\_i} \rightarrow \text{if}\Big(((\Delta x_{OS}=-1)\&(\Delta h_S=1))\#\Big((x_{OSR}-x_{OSM})>0.9\text{Tx}\Big)\&\Big((h_{OSM}-h_{OSR})>0.9\text{Th}\Big)\Big)\Big)\\
&\text{Der\_f} \rightarrow \text{if}\Big(((\Delta x_{OS}=1)\&(\Delta h_S=-1))\#\Big(|x_{OSM}-x_{OSR}|<0.1\text{Tx}\Big)\&\Big(|h_{OSM}-h_{OSR}|<0.1\text{Th}\Big)\Big)\Big)\\
&\text{Arr\_i} \rightarrow \text{if}\Big(((\Delta v_S=1))\#\Big((y_{OSM}-y_{OSR})>0.9\text{Ty}\Big)\&\Big((v_{OSM}-v_{OSR})>0.9\text{Tv}\Big)\Big)\Big)\\
&\text{Arr\_f} \rightarrow \text{if}\Big(((\Delta v_S=-1))\#\Big(|y_{OSM}-y_{OSR}|<0.1\text{Ty}\Big)\&\Big(|v_{OSM}-v_{OSR}|<0.1\text{Tv}\Big)\Big)\Big)\\
&\text{Aba\_i} \rightarrow \text{if}\Big(((\Delta y_{OS}=1)\&(\Delta v_S=-1))\#\Big((y_{OSM}-y_{OSR})>0.9\text{Ty}\Big)\&\Big((v_{OSR}-v_{OSM})>0.9\text{Tv}\Big)\Big)\Big)\\
&\text{Arr\_f} \rightarrow \text{if}\Big(((\Delta y_{OS}=-1)\&(\Delta h_S=1))\#\Big(|y_{OSM}-y_{OSR}|<0.1\text{Ty}\Big)\&\Big(|v_{OSM}-v_{OSR}|<0.1\text{Tv}\Big)\Big)\Big)
\end{aligned}
\qquad (5.23)
$$

5.3.1. Resultados

A continuación se muestran algunos resultados del funcionamiento del generador de comandos para varias secuencias de movimientos de cabeza realizadas por varios usuarios. Éste realiza movimientos de derecha, izquierda, arriba y abajo y para cada uno de ellos se evalúan distintos ángulos de giro. En la figura 5.17 se muestran los resultados para movimientos de derecha. Se han realizado tres secuencias de movimientos con cuatro posiciones de giro cada una (20º, 40º, 60º, 90º). El sistema detecta los giros mayores de 40º en las secuencias primera y tercera. En la secuencia segunda el giro de 40º no

es detectado, ya que la derivada del tamaño horizontal no supera el umbral de activación, al haberse realizado un movimiento suave. Obsérvese que los valores de las variables de referencia se modifican con el tiempo al no tener el usuario una referencia fija para la posición de NO OPERACIÓN. En la figura 5.18 se presenta un ejemplo similar al anterior pero con movimientos de izquierda. En este caso se realizan tres secuencias de tres giros cada una (30º, 60º y 90º). Todas las secuencias son detectadas por el sistema excepto la de 30º de la segunda, dado que la derivada del tamaño horizontal no alcanza el umbral de activación. En la figura 5.19 se dan dos secuencias de cuatro movimientos hacia arriba (20º, 40º, 60º y 90º) y una de uno (90º). Todas ellas son correctamente detectadas. En la figura 5.20 se evalúa el movimiento de abajo mediante tres secuencias de cuatro giros cada una (20º, 40º, 60º y 90º), siendo todas correctamente detectadas.

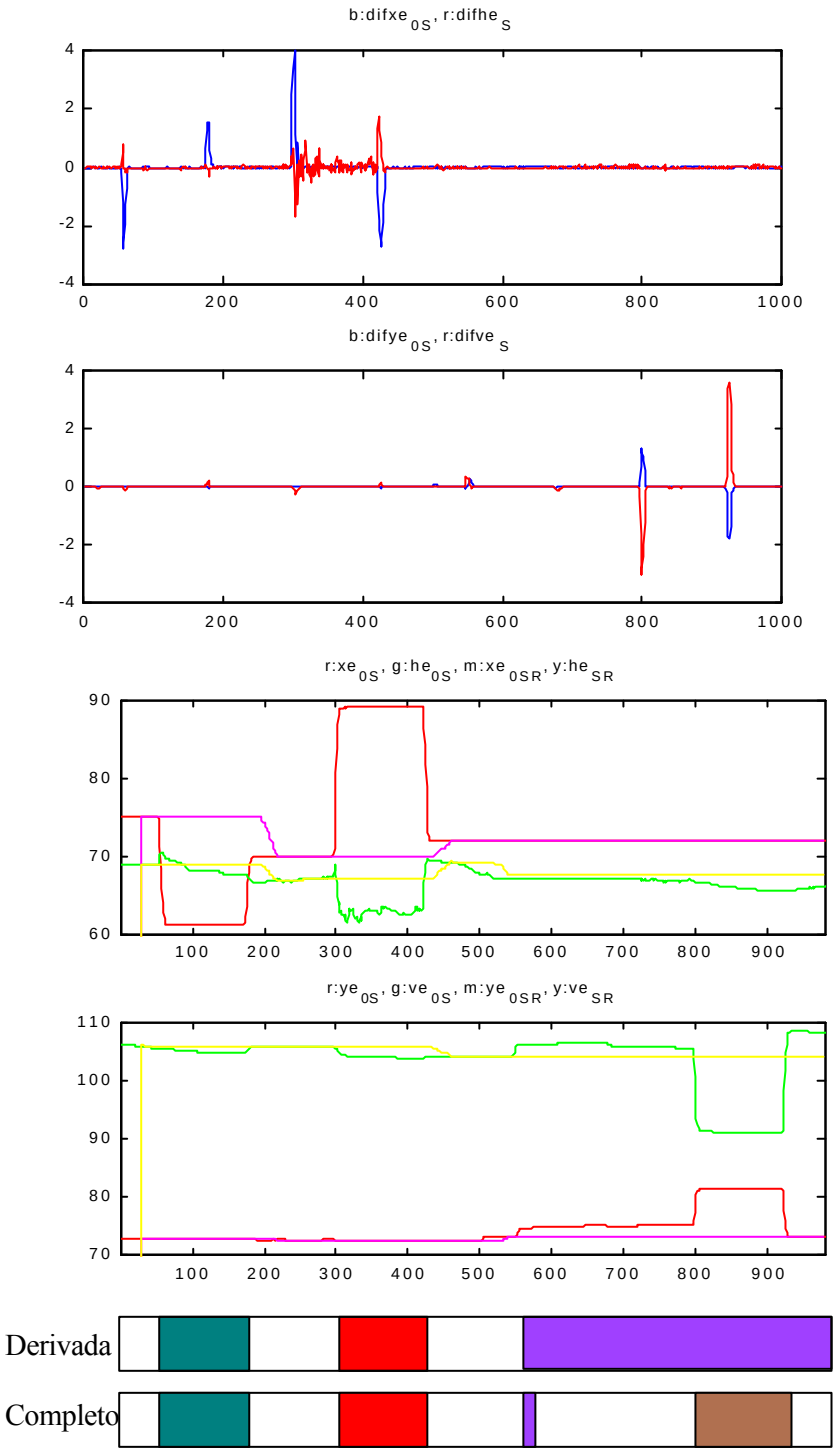


Figura 5.16. Comparación entre el modelo con derivada y modelo completo

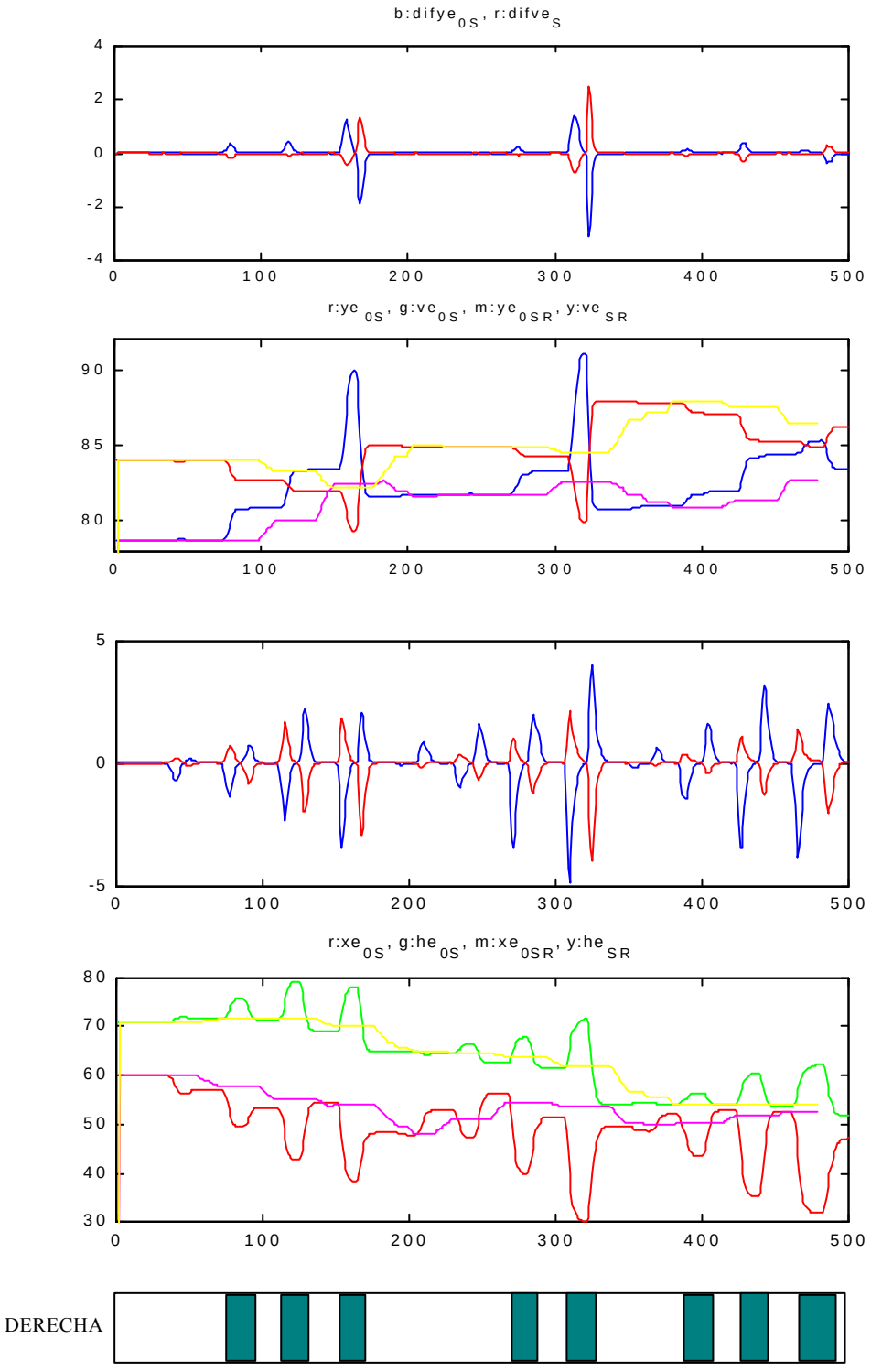


Figura 5.17. Ejemplos de generación de comando DERECHA

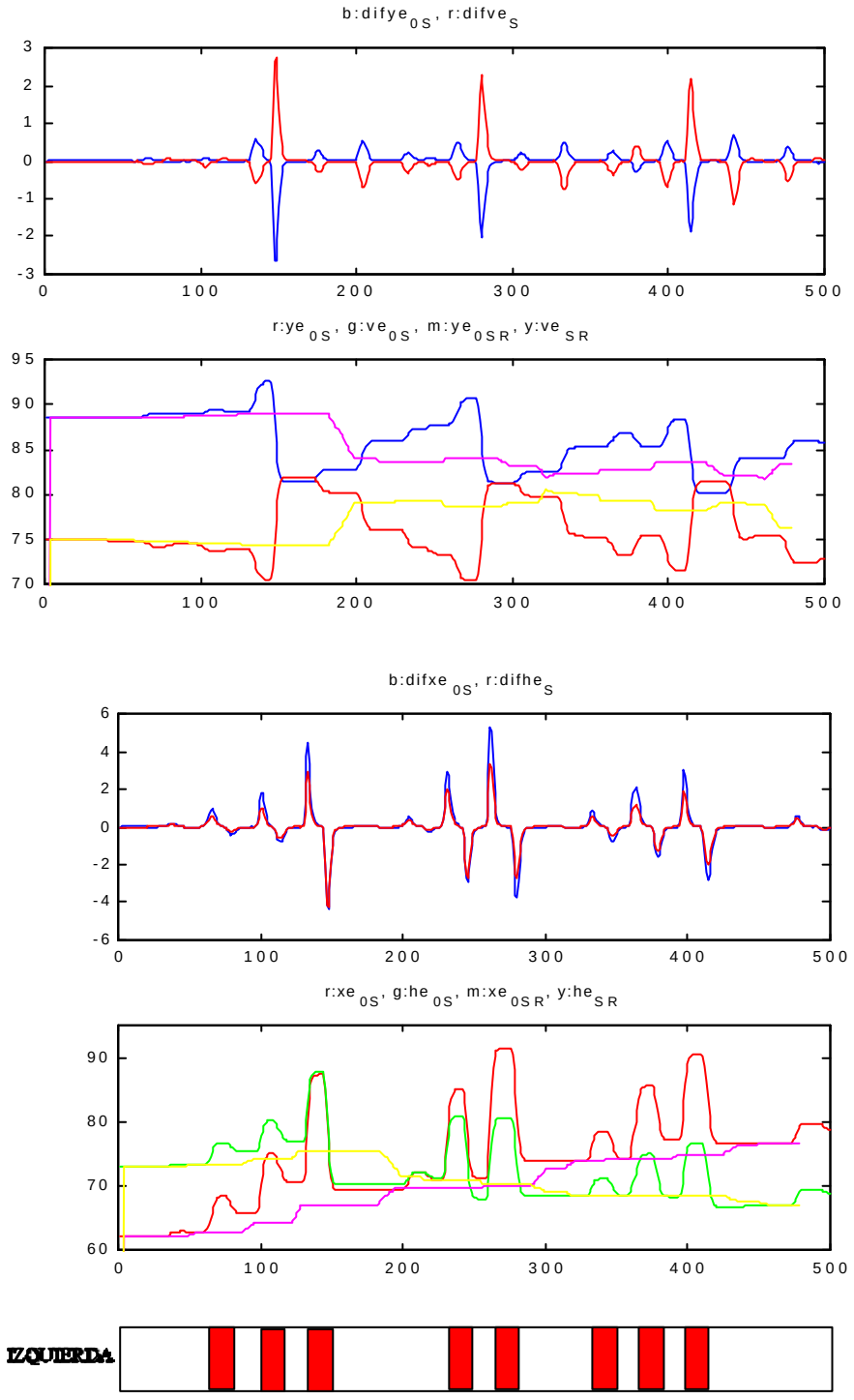


Figura 5.18. Ejemplos de generación del comando IZQUIERDA

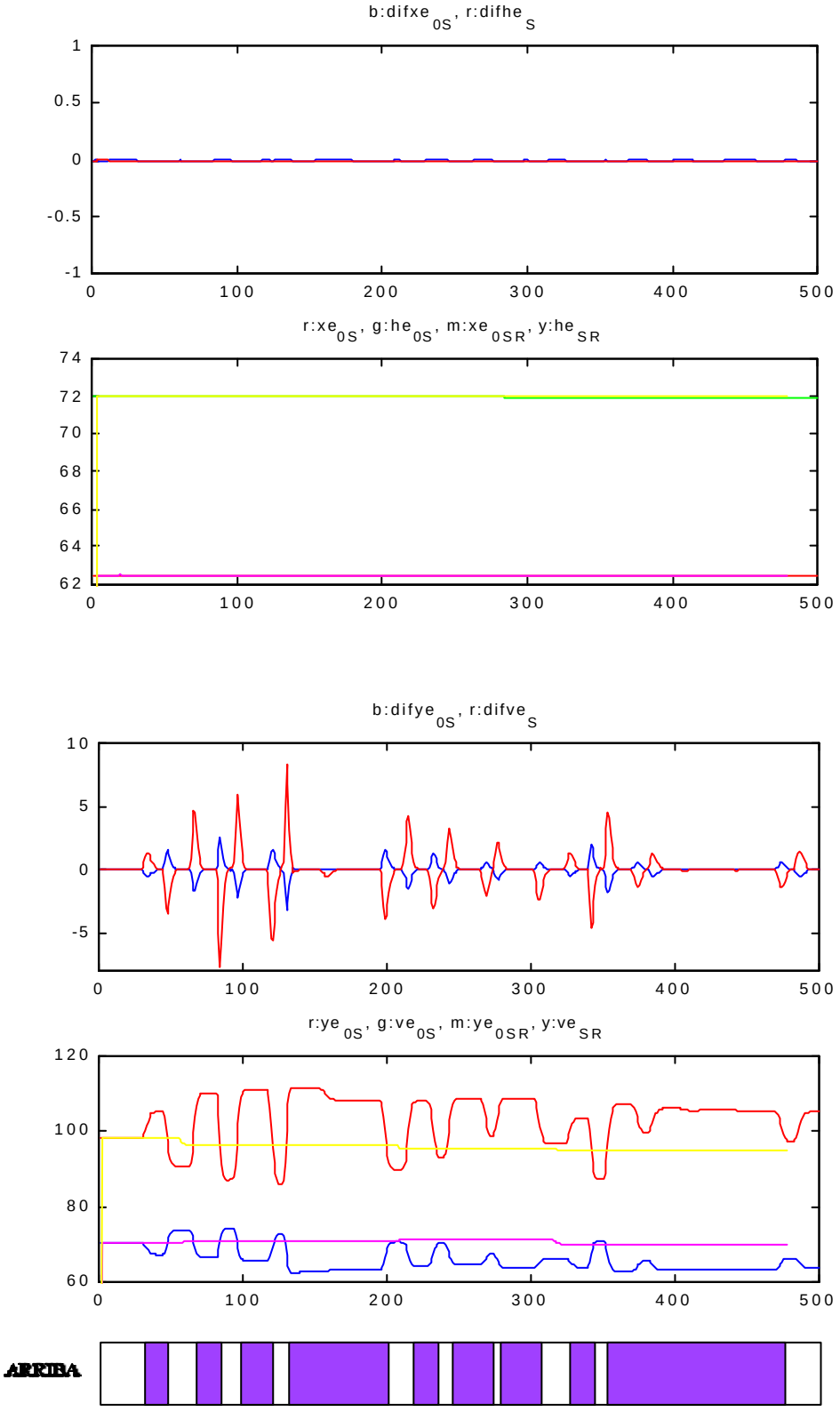


Figura 5.19. Ejemplos de generación de comandos ARRIBA

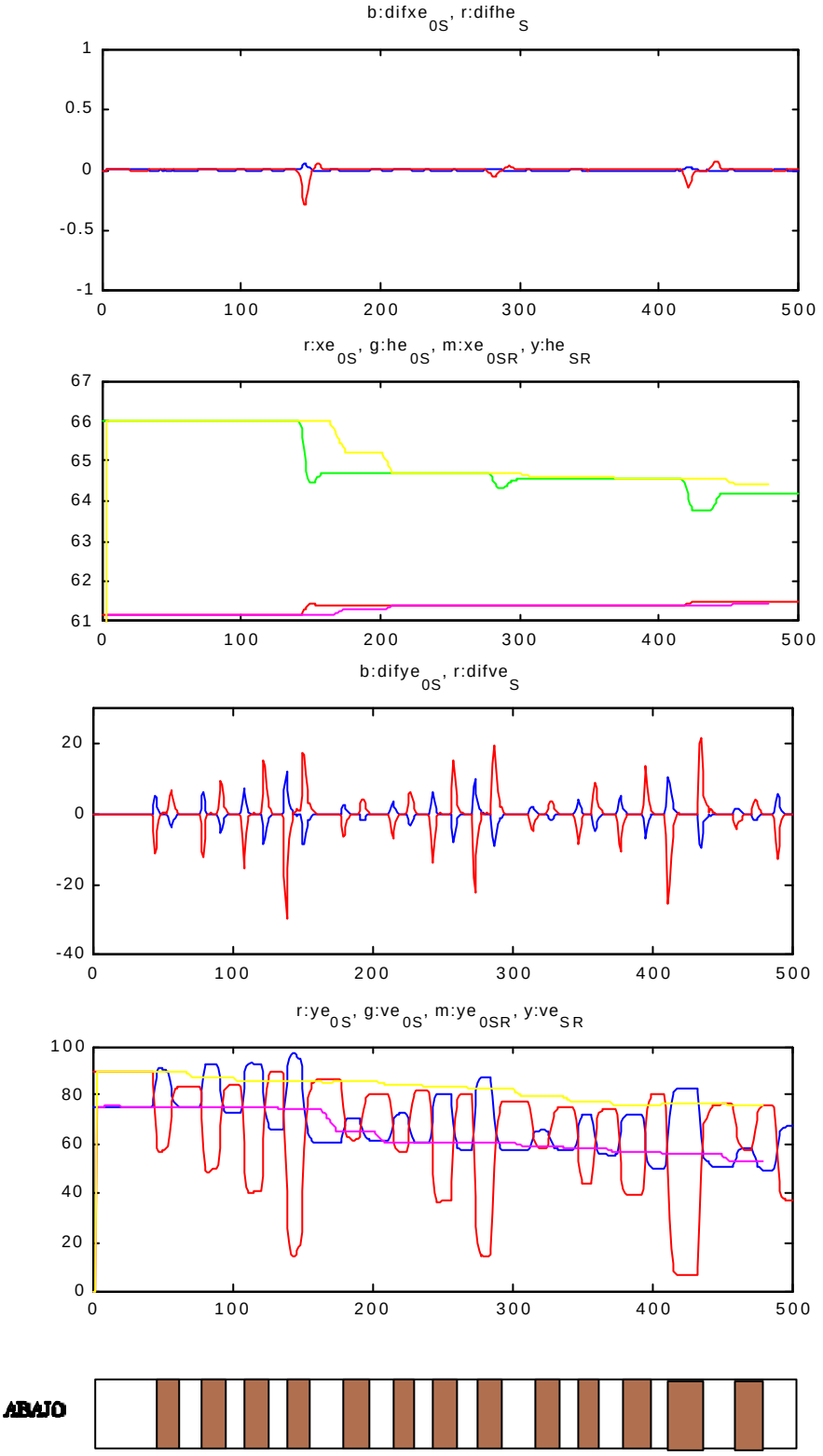


Figura 5.20. Ejemplo de generación de comandos ABAJO

5.4.- LOCALIZACIÓN DE LOS OJOS Y LA BOCA

Como ya se ha comentado en el punto anterior, se han elegido las acciones de guiñar un ojo y de ocultar los labios para activar los comandos de ADELANTE/ATRÁS y ACTIVO/INACTIVO respectivamente. Ello ha sido debido a que son acciones fáciles de ejecutar por usuarios con problemas de movilidad y, por otro lado, a que son fáciles de detectar en imágenes frontales.

Inicialmente se planteó realizar un seguimiento de las características faciales en todo momento. Consultanto la bibliografía existente al respecto, se llegó a la conclusión de que existían trabajos que empleaban diversas técnicas, pero que no trabajaban en tiempo real. Así [Yuille&Hallinan,92] utiliza plantillas deformables, [De Silva et al.,95] emplea proyección de pixels de borde y plantillas deformables, [Yow&Cipolla, 96] aplica filtros de Gabor y una red bayesiana. Los métodos que planteaban sistemas en tiempo real se basaban en: la detección de los pixels más oscuros de la imagen para localizar ojos y boca [Gee&Cipolla, 95][Stiefelhagen et al.,95a], en el empleo de correspondencias [Heinzmann&Zelinsky,97], o de redes neuronales [Baluja&Pomerleau, 94]. Todos ellos aplicaban, una vez detectadas las características, un modelo geométrico de la cara para rechazar falsas detecciones. Los métodos basados en correspondencias y redes neuronales son sensibles a los cambios de iluminación y de aspecto de las características. Por otra parte, en todos ellos su efectividad disminuye considerablemente cuando el usuario gira la cabeza por encima de 30º.

Debido a que el sistema desarrollado en esta tesis requiere giros de cabeza por encima de 30º, nos vimos obligados a replantear la idea inicial del seguimiento en todo momento. Había dos soluciones:

a) *Robustecer la detección para cualquier ángulo de giro de la cabeza*. Esto supone utilizar varios modelos geométricos (vista frontal,vista lateral derecha,vista lateral izquierda) con la consiguiente complicación de la conmutación de modelos.

b) Permitir únicamente la detección en la vista frontal. Con ello se simplifica el sistema al utilizar únicamente un modelo, lo que lo hace más robusto.

En nuestro caso, como los giros de cabeza se realizan para activar comandos, no se puede dar el caso de que el usuario quiera activar dos comandos a la vez, al ser un sistema secuencial. Es cuando está con su cabeza en posición frontal (reposo) cuando activa los comandos.

De esta forma se ha evitado el gran problema de las detecciones laterales de los sistemas en tiempo real con condiciones cambiantes. Por otro lado, para hacer más robusto al sistema, no interesa la posición exacta de las características, sino si éstas se encuentran o no en la cara, ya que cuando el usuario guiñe un ojo u oculte los labios, estas acciones se manifestarán por la desaparición del objeto sobre la imagen.

Ahora la cuestión es, ¿qué método utilizar? El de correspondencias queda eliminado al necesitar de gran cantidad de plantillas y requerir, para que funcione en tiempo real, un hardware específico del que no se dispone. Por lo tanto, la opción que queda es trabajar con los niveles de luminancia de la imagen.

Se parte de una imagen en color en la que se ha segmentado la piel del usuario, así que las características faciales deben ser buscadas en el interior del objeto piel. Para ello se definen dos ventanas (una para los ojos y otra para la boca) que se posicionan a partir de las variables (h_S, v_S) del objeto piel, como se puede ver en la figura 5.21. El área de búsqueda de los ojos se restringe a la mitad superior del objeto piel, con ciertos márgenes en los bordes, y la de la boca a la mitad inferior también imponiendo ciertos márgenes. Se han elegido ventanas amplias, a pesar de que puedan aparecer más objetos de ruido, con el fin de que el posicionamiento valga para cualquier usuario.

Figura 5.21. Ventanas de búsqueda de ojos y boca

Dentro de la ventana de estudio se buscan los huecos existentes en el objeto piel, ya que éstos se corresponden con partes de la cara que tienen un color diferente al de la piel y, por lo tanto, pueden ser candidatos a ser las características buscadas. Obsérvese en la figura 5.20, como aparecen huecos de ruido que deberán ser eliminados.

5.4.1. Localización de los ojos

El problema consiste en determinar qué pareja de huecos de los detectados en la ventana de ojos se corresponde con los ojos. Para ello, se deben tener en cuenta las siguientes características de los mismos: las pupilas son partes oscuras dentro de la imagen, un ojo está formado por la pupila y la esclerótica (blanco del ojo), con colores muy contrastados; satisfacen ciertas restricciones geométricas como posición dentro de la cara, simetría con respecto al eje de simetría facial y anchura mínima y máxima entre ellos.

Inicialmente se decidió detectar las partes más oscuras de la imagen empleando un umbral. El problema que aparece es que éste cambia en función de las personas y de las condiciones de luz, por lo que hay que hacerlo adaptativo. Se programó la estrategia propuesta en [Stiefelhagen et al.,95a] consistente en umbralizar iterativamente la imagen, comenzando con un umbral muy bajo, hasta que se encontraran un par de huecos que satisfacieran las condiciones geométricas impuestas. Este método fallaba con personas con cejas pobladas y oscuras, al detectar éstas como las zonas más oscuras que cumplían las restricciones geométricas. Asimismo, a veces, también detectaba huecos de ruido debidos a sombras.

Se obtuvieron mejores resultados analizando la varianza de la luminancia de los huecos, al cumplirse que los huecos de los ojos son los que poseen una mayor varianza, mayor que la de las cejas. El organigrama del algoritmo empleado se muestra en la figura 5.22.

Una vez posicionada la ventana de búsqueda, se localizan los huecos que caen completamente dentro de ella. Aquellos que tienen conectividad con el borde de la ventana son eliminados. A continuación se hace una exclusión a groso modo por área y por forma. Sólo se analizan aquellos objetos con un área mayor de cuatro pixels, de esta manera se eliminan los objetos debidos a ruido, y con área menor a 100 pixels, con lo que se eliminan objetos grandes que a veces aparecen en los bordes de la ventana debidos al pelo o al fondo de la imagen. También serán eliminados aquellos objetos que tengan un tamaño en vertical mayor que el horizontal, ya que, los ojos son alargados horizontalmente.

Seguidamente se calcula la varianza de la luminancia de los pixels de cada objeto (hueco), que haya pasado la primera restricción, y se ordenan éstos de mayor a menor varianza. Se toman los cuatro objetos de mayor varianza y se calculan las combinaciones de los cuatro elementos tomadas de dos

en dos. Por lo tanto, se forman seis posibles parejas de ojos candidatos. El hecho de tomar cuatro objetos fue debido a que experimentalmente se comprobó que los ojos siempre estaban entre los cuatro objetos con mayor varianza a analizar, con lo que de esta forma se reducía el tiempo de

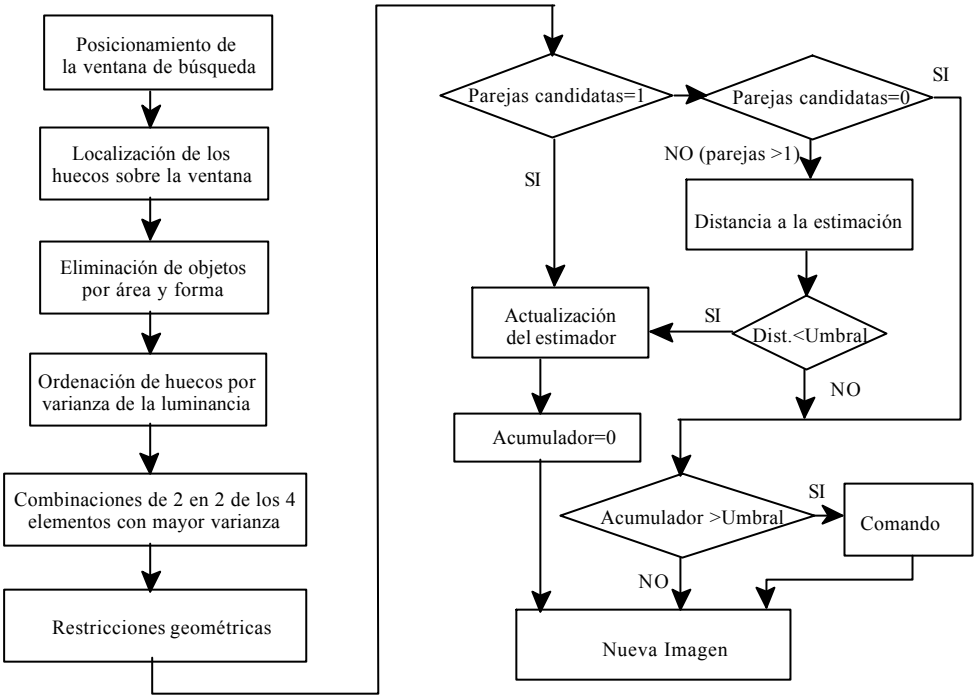


Figura 5.22. Organigrama de la localización de ojos

cómputo.

A cada una de las parejas de ojos candidatos se les aplica las siguientes restricciones geométricas secuencialmente:

• *Distancia en horizontal*: experimentalmente se han calculado unos límites máximos y mínimos por exceso.

• *Simetría*: las parejas candidatas tienen que ser simétricas respecto al eje facial de la cara. Para cada una de ellas (entre el objeto i y j) se calcula la siguiente medida de simetría, S(i,j):

$$S(i,j) = |cand_i[x] - (h_{Vojos} - cand_j[x])| \qquad (5.24)$$

donde $cand_i[x]$ y $cand_j[x]$ son las posiciones horizontales de los centros de gravedad de los

objetos candidatos, y h_{Vojos} es la anchura de la ventana de búsqueda de los ojos. La variable S será cero si los candidatos se encuentran a la misma distancia del borde de la ventana y además son perfectamente simétricos. A medida que las diferencias de distancias con el borde aumenten, S(i,j) se incrementará linealmente. Se fija un valor máximo de distancia simétrica (S_{max}) de forma que si se supera la pareja candidata será rechazada.

• *Distancia en vertical*: se establece un límite máximo de distancia vertical que es bastante restrictivo, dado que si no se podrían dar como válidas parejas de ojos y cejas. Sin embargo, esto impone que la cabeza no debe estar inclinada puesto que de lo contrario la distancia vertical entre ojos superará el límite y no serán localizados.

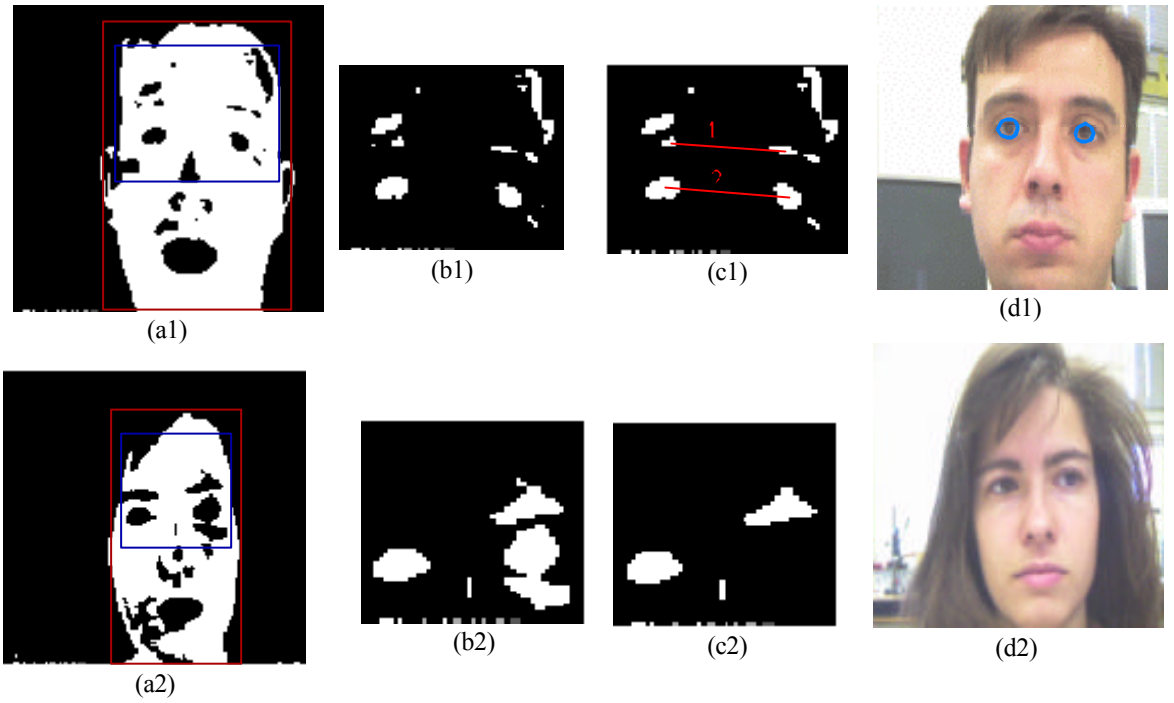
Si más de una pareja satisface todas las restricciones, se elegirá aquella que tenga una componente vertical mayor, para la primera vez que se ejecute el algoritmo, y la que se encuentre más cerca de la estimación hecha para ella, mediante la media de las cinco componentes anteriores, para el resto. Suele ser habitual que pasen las restricciones tanto los ojos como las cejas. De esta forma nos aseguramos que el sistema se centra en los ojos y no en las cejas.

Todos los parámetros utilizados en las restricciones geométricas mencionadas han sido elegidos respecto al tamaño de la cara, por lo que el método es independiente del tamaño de la misma.

En el caso de que ninguna pareja pase las restricciones impuestas puede ser debido a que se haya detectado un parpadeo, que el usuario haya guiñado un ojo, o que el usuario tenga la cabeza girada por movimientos involuntarios. Para que el sistema funcione hay que asumir que inicialmente el usuario se encuentra con los ojos abiertos, para que el detector se enganche con los mismos.

Como el objetivo es detectar la acción de los guiños voluntarios del usuario, una manera de filtrar los parpadeos será mantener la acción un tiempo mayor que el de parpadeo. Los parpadeos de una persona tienen una duración inferior a un segundo, por lo que se fijó el umbral de la acción de guiño en dos segundos. Por lo tanto, cuando no haya parejas candidatas, se incrementa un contador de eventos que va acumulando el número de veces consecutivas que se ha dado esta acción, de manera que si se supera el umbral, se valida la acción de guiño que desencadena la activación del comando ADELANTE/ATRÁS. Se ha fijado un valor de umbral igual a 18, lo que supone detectar un ojo cerrado durante casi dos segundos (se está trabajando a 10 imágenes por segundo).

Figura 5.23. Ejemplos de detección de ojos (a)Ventana de búsqueda (b)Huecos (c) Huecos después de las restricciones y parejas candidatas (d) Ojos detectados



En la figura 5.22 se muestran dos ejemplos de detección para dos usuarios distintos. En el caso 1 (a1, b1, c1, d1), se tienen 2 parejas candidatas y se asigna a la más baja como la de los ojos. En el caso 2 (a2, b2, c2, d2), el objeto ojo está unido a una sombra y es eliminado en la restricción por forma (es más alto que ancho), las posibles parejas con el resto de los objetos no cumplen las restricciones geométricas y por lo tanto no se detectan los ojos.

En la figura 5.24 aparece una secuencia de 500 muestras donde se han realizado varias activaciones del comando ADELANTE/ATRÁS. Se muestran la posición (x,y) de cada uno de los ojos y el acumulador de ojo cerrado. Se producen siete activaciones, al haber detectado un ojo cerrado un número de veces consecutivas igual al umbral. Obsérvese como aparecen acumulaciones involuntarias que en ningún caso superan el umbral, por lo que son filtradas. Éstas son debidas a parpadeos, a falsas detecciones esporádicas de huecos de sombra que son confundidos con ojos o a uniones del hueco

de los ojos con otros de sombra que modifican su posición (centro de gravedad del objeto resultante). Se puede apreciar que cuando no se detecta un ojo se mantiene la estimación de su posición, a partir de las últimas muestras existentes y por lo tanto, se mantiene esta estimación para cuando se abra el ojo nuevamente.

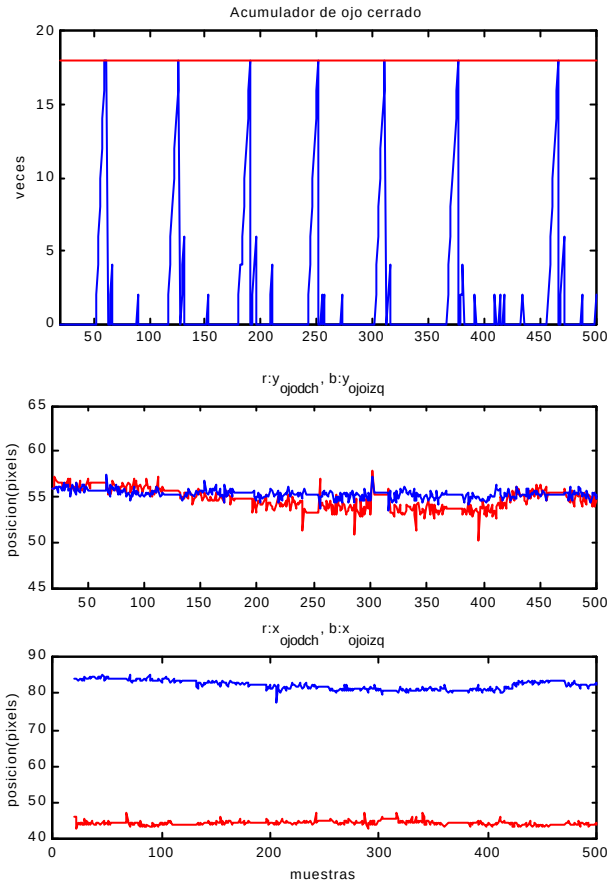


Figura 5.24. Activación del comando ADELANTE/ATRÁS

La localización de ojos únicamente se ejecuta en el estado de NO OPERACIÓN. Cuando el usuario activa algún comando, el estimador se pone a cero y cuando regresa al estado de NO OPERACIÓN se vuelve a inicializar. Ello es debido a que se demostró que cuando se vuelve a la posición de reposo, después de haber realizado una acción, nunca se hace en la posición de reposo inicial, por lo que cualquier estimación que se haga con los datos de la anterior posición de reposo no será válida.

5.4.2. Localización de la boca

El problema de la localización de la boca se basa en determinar cual de los huecos detectados en la ventana de la boca se corresponde con ella. Para ello, se han tenido en cuenta las siguientes características de la misma: es la mayor característica facial interior a la cara, tiene un color rojizo y se encuentra situada en el eje de simetría facial (para imágenes frontales).

Su localización es más sencilla que la de los ojos, al tratarse de un objeto de mayor tamaño y estar situado en un entorno en el que no hay objetos ruidosos. Al lado de los ojos se encuentran las cejas, que crean grandes problemas en la localización de los mismos, ya que hay que evitar que el sistema se centre en ellas en lugar de en los ojos. En este caso el único problema está en personas que tengan bigote de un color muy contrastado con la piel, ya que puede aparecer otro objeto de tamaño similar a la boca. De todas formas se hicieron algunas pruebas comprobando que el objeto bigote tenía conectividad con la boca y, por consiguiente, aparecía como un único objeto. Se comprobó que si éste no era muy grande, al ocultar los labios, el objeto que quedaba era de tamaño inferior al de la boca y entonces se podía filtrar. El único caso en el que fallaba era con personas con un gran bigote y de color muy contrastado con la piel (negro). El sistema planteado en esta tesis no funciona con personas con barba poblada al tener un modelo de piel de la cara totalmente diferente al de una persona sin ella.

El organigrama del algoritmo aplicado en este caso se muestra en la figura 5.25.
Los tres primeros pasos son iguales que en la localización de los ojos. Se posiciona la ventana, se extraen los huecos interiores de la misma y se introducen una serie de restricciones geométricas de tamaño y de forma. En este caso se rechazan todos los objetos con un área inferior a 50 pixels y mayor de 500 pixels. Téngase en cuenta que el tamaño medio suele estar en 200 pixels. Asimismo, también se eliminan aquellos objetos que tengan una elongación vertical mayor que la horizontal, al tratarse de un objeto claramente extendido horizontalmente.

A continuación se toma el objeto con mayor área y se le impone que se encuentre en el centro de la cara, con un cierto margen de error, definido experimentalmente.

$$|\mathrm{cand}_{\mathrm{boca}}[x] - x| < \mathrm{Th}_{\mathrm{boca}} \qquad (5.25)$$

donde $\mathrm{cand}_{\mathrm{boca}}[x]$ es la posición horizontal del centro de gravedad del objeto con mayor área, x es el

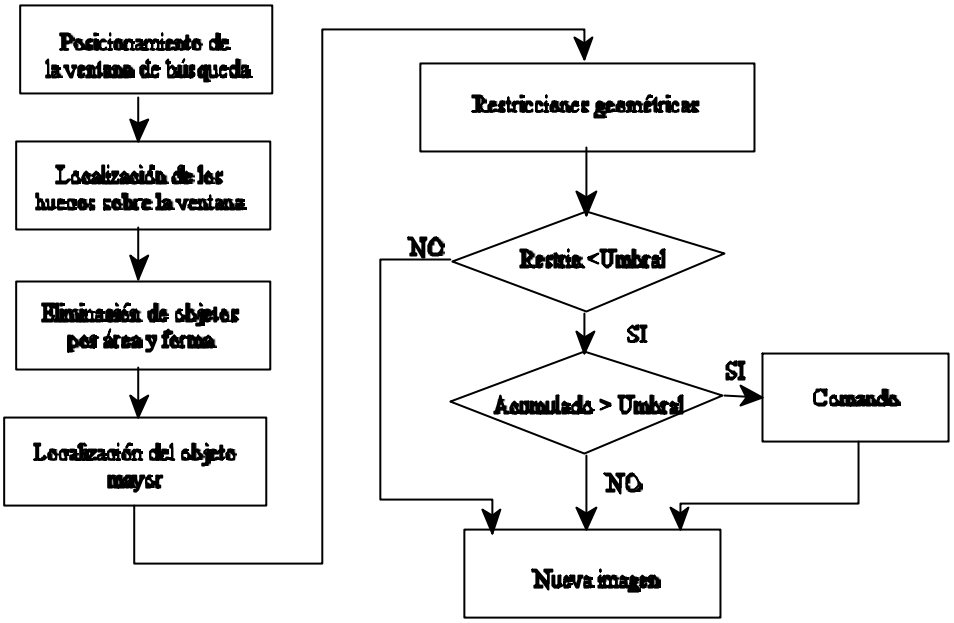


Figura 5.25. Organigrama de la localización de la boca

centro de gravedad del objeto piel y Th_{boca} es el umbral que se ha fijado en 10 pixels.

En el caso de que el candidato pase la restricción es porque el sistema ha detectado la boca. Cuando no se detecte ningún candidato será debido a que el usuario ha realizado la acción de ocultar lo labios, que ha girado la cara de forma involuntaria y, por lo tanto, el objeto no cumple la restricción geométrica, o bien, que debido a luces o sombras el objeto boca se ha partido y de ahí que ya no sea el mayor. Para robustecer la detección ante estos ruidos se utiliza un acumulador de boca oculta que irá acumulando el número de veces consecutivas que se da la acción, de forma que si se supera un umbral, se validará la acción desencadenando la activación del comando ACTIVO/INACTIVO. Se ha fijado el umbral a 19 lo que supone que se debe detectar el ocultamiento de labios durante casi dos segundos consecutivos para validar la acción.

En la figura 5.25 se muestran dos ejemplos de detección para dos usuarios. Como se puede ver en el primero de ellos (a1, b1, c1, d1) se detectan cinco huecos en la ventana que después de aplicarles las restriciones de tamaño y forma se quedan en dos. De éstos el sistema detecta como boca el mayor y más centrado de ellos. En el segundo caso (a2, b2, c2, d2), hay una sombra que tiene conectividad con

la boca y con el borde por lo que es eliminada. Sólo queda un objeto que es eliminado por área, luego no se detecta la boca.

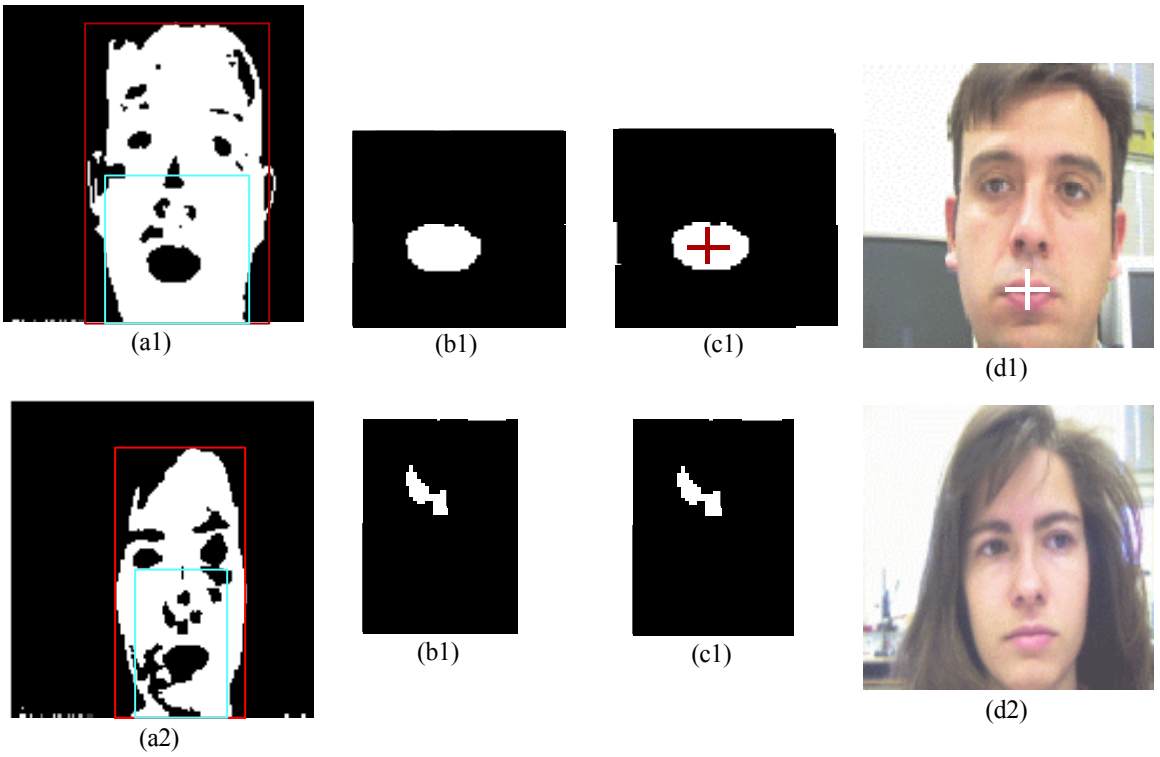


Figura 5.26. Ejemplos de detección de boca (a)Ventana de búsqueda (b)Huecos (c) Huecos después de las restricciones y parejas candidatas (d) Ojos detectados

En la figura 5.27 se dan los valores del centro de gravedad de la boca (x_{boca}, y_{boca}) y el acumulador de boca para una secuencia de 500 muestras. Se producen cuatro activaciones al superar el umbral. También aparecen ciertas acumulaciones no voluntarias debidas a momentáneos giros de la cabeza o falsas detecciones. Sin embargo, éstas no se detectan seguidas y provocan acumulaciones máximas de cinco, por lo que al no llegar al umbral no producen fallo. Cuando no se detecta objeto la posición de éste se toma como cero. Es por ello que las gráficas de posición tienen una forma como la que se muestra, en las que únicamente se representa una ventana de posiciones para tener una mayor resolución. Obsérvese cómo las variaciones en "x" e "y", cuando se detecta objeto, no superan los cinco pixels.

5.5.- CONTROL DE LA SILLA

Una vez expuesta la forma de generación de comandos por parte del usuario, a continuación se describe el sistema de control planteado para la silla en esta tesis. El objetivo consiste en mover la silla, a voluntad del usuario, mediante la generación de una serie de comandos discretos obtenidos con movimientos de cabeza.

Para comprender mejor el funcionamiento del sistema, es conveniente tener una visión general del lazo de control completo mostrado en la figura 5.28.

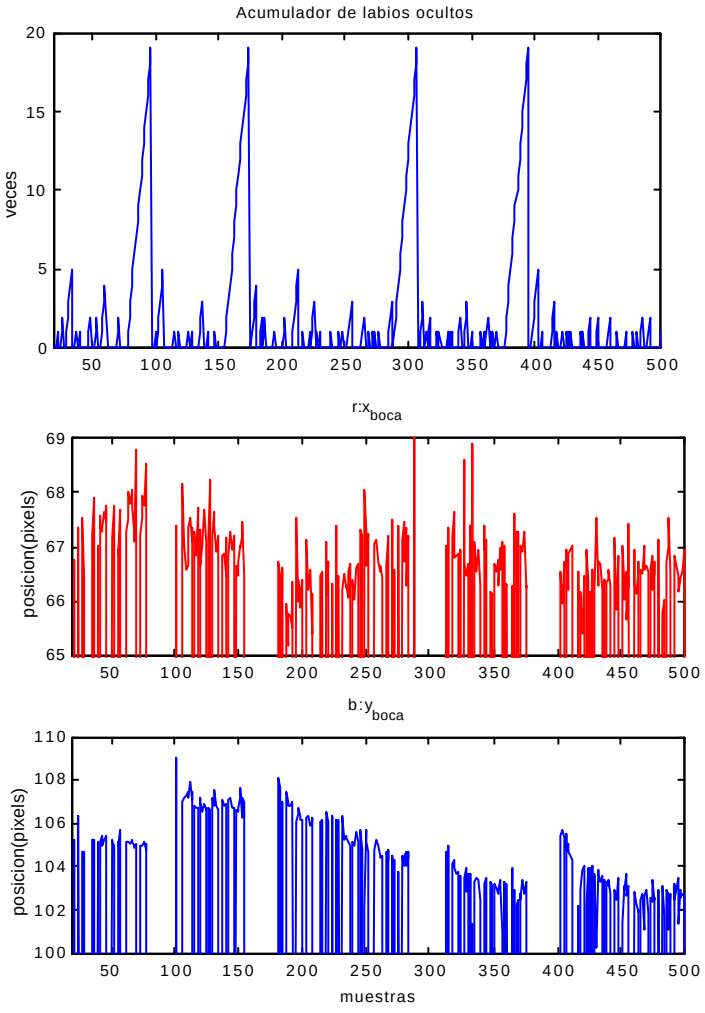


Figura 5.27. Activación del comando ACTIVO/INACTIVO

Como se puede observar, se trata de un sistema de control en lazo cerrado donde el usuario controla la velocidad de la silla. Es el usuario quien se plantea un destino a alcanzar y, conociendo la ubicación actual de la silla, genera una serie de comandos con su cabeza, que hacen que la silla se mueva hasta el destino. Por lo tanto, es el usuario quien establece la trayectoria a seguir y la forma de llegar al destino mediante un conocimiento previo de cómo actúan cada uno de los comandos sobre la velocidad lineal y angular de la silla. La realización de un movimiento con la cabeza desencadena un movimiento de la silla y en función de éste y la trayectoria a seguir deseada es el propio usuario el que realimenta al sistema manteniendo el mismo comando o generando otro.

La misión del control a alto nivel es determinar la velocidad lineal (V) y angular (Ω) que debe llevar la silla a partir de una serie de comandos discretos generados por el usuario y de las limitaciones de velocidad impuestas por la planta. Puesto que el control a bajo nivel de los motores requiere como consignas las velocidades angulares de las dos ruedas, es necesario aplicar las ecuaciones de la cinemática de la silla para obtener estas velocidades a partir de V y Ω. El control a bajo nivel tiene como misión minimizar la diferencia entre las consignas de velocidad de ambas ruedas (ω_d, ω_i) y su velocidad real, obtenida a través de los "encoders" de la silla.

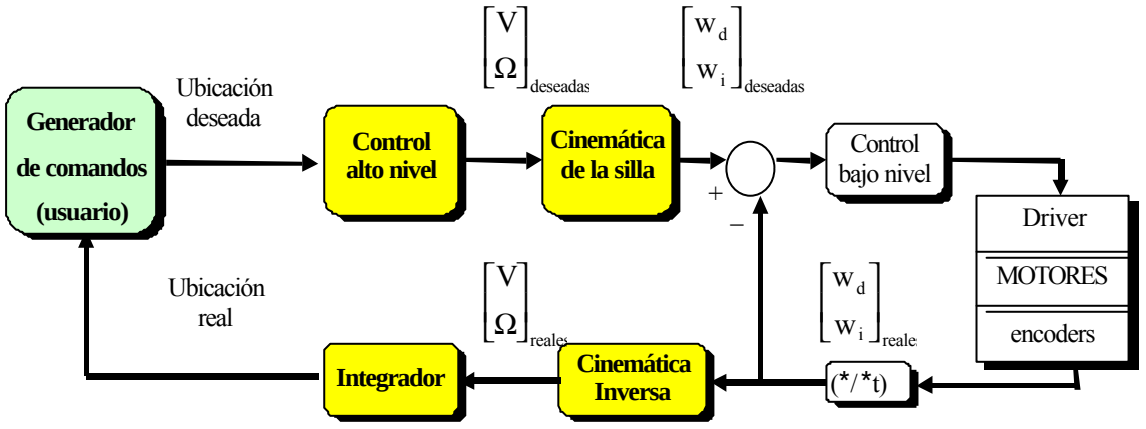


Figura 5.28. Diagrama de bloques del lazo de control

5.5.1.- Control a alto nivel

Una forma básica de control de un móvil por eventos consiste en asignar una velocidad lineal y angular constante a cada uno de ellos de forma independiente. Este es el sistema usado en la mayoría de los controles con joysticks de las sillas motorizadas. Los problemas que acarrea son que en un tiempo muy pequeño se obliga al móvil a cambiar bruscamente su velocidad y como la silla se comporta como un filtro paso bajo con retardo se pueden producir inestabilidades en los transitorios. Por otro lado, las velocidades de los movimientos son fijas[1], lo que reduce la controlabilidad de la silla.

Con el objeto de aumentar el grado de control de la silla, por parte del usuario, se ha utilizado como control a alto nivel una máquina de estados como la mostrada en la figura 5.29. Con ella se obtiene la velocidad lineal y angular de la silla a partir de los comandos generados por el usuario y de la variable temporal.

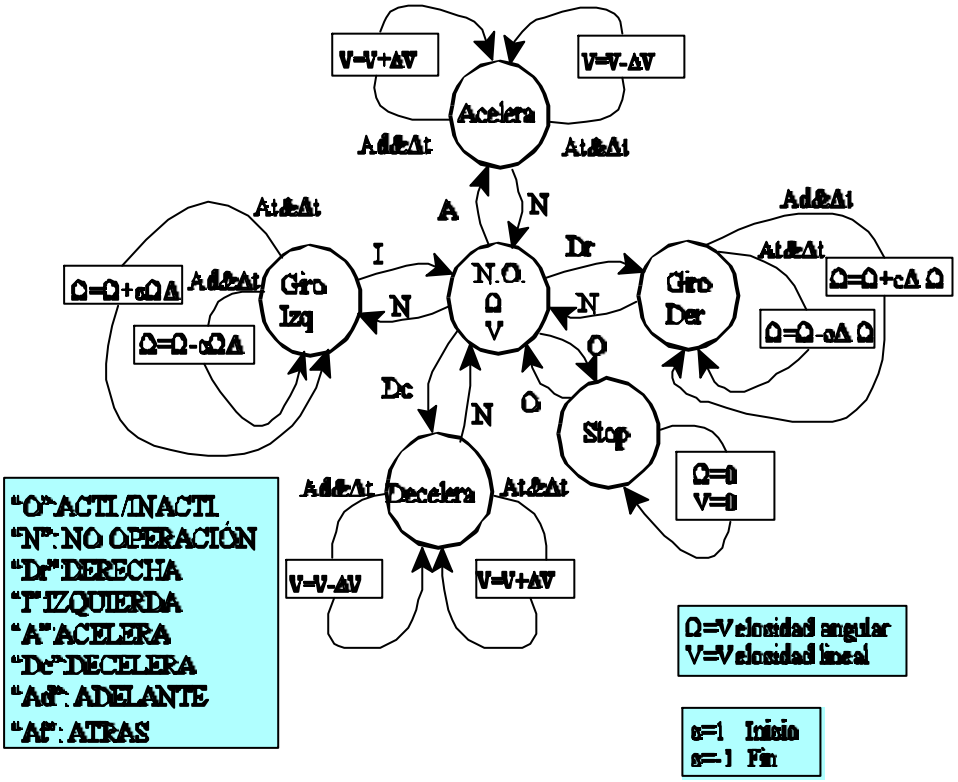


Figura 5.29. Máquina de estados del control a alto nivel

[1]En algunas sillas de ruedas existe un conmutador de velocidad que permite al usuario elegir entre varias velocidades previamente establecidas.

La máquina tiene un estado de NO OPERACIÓN (N.O.) que mantiene las velocidades lineales y angulares sin modificar. Inicialmente el sistema comienza en el estado STOP, donde tanto la velocidad lineal como la angular son igual a cero. Si el usuario activa el comando ACTIVO/INACTIVO, se produce la conmutación al estado de NO OPERACIÓN, desde donde se posibilita el paso a otros estados. Por otro lado, si el sistema se encuentra en el estado de NO OPERACIÓN y recibe un comando ACTIVO/INACTIVO, pasa al estado de STOP, poniendo las velocidades a cero. En este caso el usuario puede realizar movimientos sin que se mueva la silla, ya que el único evento que le puede sacar de este estado es el comando ACTIVO/INACTIVO que lleva nuevamente al sistema al estado de NO OPERACIÓN.

Cada vez que se recibe un comando de DIRECCIÓN se produce una conmutación del sentido de la marcha, así, inicialmente se selecciona el sentido hacia ADELANTE, de forma que un comando de DIRECCIÓN cambiará el sentido hacia ATRÁS y así sucesivamente. Según el estado de dirección del sistema, la salida del controlador cambiará. De modo que si se encuentra en el estado de ADELANTE, un comando de DERECHA provoca un giro hacia la derecha, lo que equivale a aumentar la velocidad angular. Un comando de IZQUIERDA provoca un giro a la izquierda, lo que equivale a disminuir su velocidad angular. Un comando de ACELERA aumenta la velocidad lineal y uno de DECELERA la disminuye. Sin embargo, si el estado del sentido es ATRÁS la respuesta del sistema es la contraria a la explicada. Si llega un comando de DERECHA se produce una disminución de la velocidad angular, lo que hace girar a la silla a la derecha. Si llega un comando IZQUIERDA, se produce un aumento de la velocidad angular, lo que hará girar a la silla a la izquierda. Si el comando es ACELERA disminuirá la velocidad lineal, al ser negativa hacia atrás. Por último, si el comando recibido es DECELERA aumentará la velocidad lineal para que su valor absoluto disminuya.

En cuanto a los giros, para hacer al sistema más estable y teniendo en cuenta las limitaciones físicas de la silla, se introduce una consigna de velocidad angular, que varía linealmente en función del tiempo desde un valor mínimo hasta alcanzar un valor máximo, como se muestra en la ecuación (5.28). La pendiente de la función (m) y el valor máximo de la velocidad angular ($S_{máx}$) son proporcionales a la velocidad lineal del móvil. La velocidad angular mínima ($S_{mín}$) viene fijada por el giro mínimo que efectua la silla, que es de ±100 mrad/s. Así, si el móvil lleva una velocidad lineal baja, gira más despacio que si lleva una velocidad alta, asegurándose que al móvil no se le aplican cambios bruscos

de consignas de velocidad al comenzar siempre el giro con una velocidad angular igual a cero. Cuando finaliza la activación del comando también se aplica una variación progresiva de la velocidad pero en sentido contrario al del inicio. Este efecto es controlado mediante la variable "c" en la ecuación.

$$\begin{array}{l} \text{Giro derecha} \rightarrow \Omega(n) = \Omega(n-1) + cS\Delta\Omega \\ \text{Giro izquierda} \rightarrow \Omega(n) = \Omega(n-1) - cS\Delta\Omega \end{array} \quad \begin{array}{l} \left\{ \begin{array}{l} \text{if } \Omega(n) > \Omega_{max} \rightarrow \Omega(n) = \Omega_{max} \\ \text{if } \Omega(n) < \Omega_{min} \rightarrow \Omega(n) = 0 \end{array} \right\} \\ \left\{ \begin{array}{l} \text{if } \Omega(n) < -\Omega_{max} \rightarrow \Omega(n) = -\Omega_{max} \\ \text{if } \Omega(n) > -\Omega_{min} \rightarrow \Omega(n) = 0 \end{array} \right\} \end{array} \quad \begin{array}{l} S = 1 \rightarrow \text{Adelante} \\ S = -1 \rightarrow \text{Atras} \\ c = 1 \rightarrow \text{Inicio} \\ c = -1 \rightarrow \text{Fin} \\ \Delta\Omega = F(V) \end{array} \quad (5.26)$$

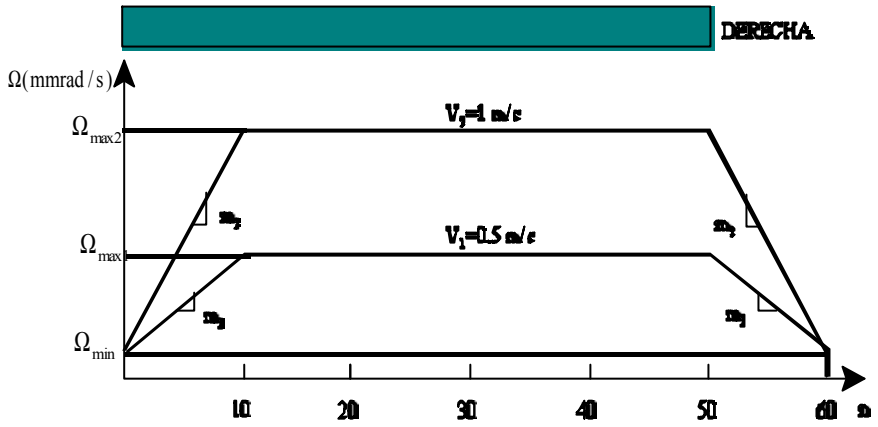


Figura 5.30. Generación de consignas de velocidad angular

En la figura 5.30 se muestra un ejemplo de dos consignas de velocidad angular para dos valores de velocidad lineal diferentes, obtenidos para una activación del comando DERECHA.

El sistema manda 10 muestras de consigna por segundo, lo que supone que cuando se activa un comando, en el primer segundo se produce una aceleración, desde la velocidad angular mínima hasta llegar a la máxima, manteniendo ésta el tiempo que se encuentre activo el comando. Cuando el comando se desactiva se produce una deceleración hasta que la velocidad angular llega a $S_{mín}$.

En cuanto a la velocidad lineal de la silla, es modificada mediante los estados de aceleración y deceleración del control a alto nivel. Cuando se activa el comando ACELERA, se produce un aumento de la velocidad lineal del móvil con incrementos ( ) V) cada 0.1 s, si el sentido es hacia adelante, o bien una disminución de la misma, según el mismo criterio, si el sentido es hacia atrás (ver ecuación (5.29)).

El sistema estará acelerando mientras el comando ACELERA se encuentre activo. De la misma forma, si se activa el comando DECELERA, la velocidad lineal se decrementa, cada 0.1 s, una cantidad discreta ΔV, si el sentido es hacia adelante, o aumenta si el sentido es hacia atrás, hasta que se desactive el comando.

$$\left.\begin{array}{l} \text{Aceleracion} \rightarrow V(n) = V(n-1) + S\Delta V \\ \text{Deceleracion} \rightarrow V(n) = V(n-1) - S\Delta V \end{array}\right\} \begin{array}{l} S = 1 \rightarrow \text{Adelante} \\ S = -1 \rightarrow \text{Atras} \end{array} \qquad (5.27)$$

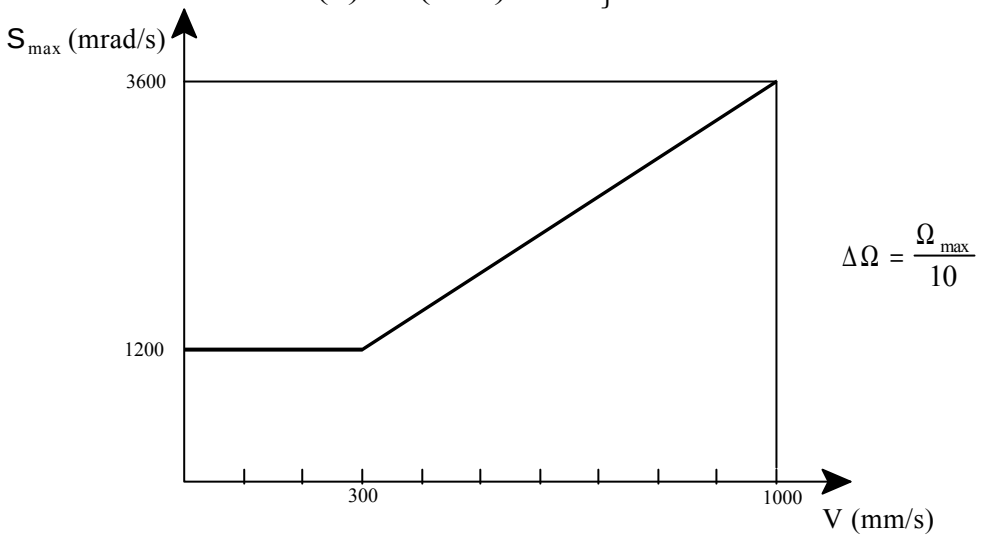


Figura 5.31. Relación entre la velocidad lineal y la velocidad angular máxima

Inicialmente se parte de un valor mínimo de velocidad lineal de ±50 mm/s, y se sigue una progresión lineal con un incremento de velocidad fijo calculado experimentalmente igual a ΔV=100 mm/s, saturando la velocidad máxima del sistema a 1 m/s.

Para conseguir una variación óptima de la velocidad angular del móvil que permita una fácil controlabilidad se ha implementado una función de transferencia que relaciona la velocidad lineal y la velocidad angular máxima del mismo, como la que se muestra en la figura 5.29. Para velocidades lineales por debajo de 300 mm/s, la velocidad angular máxima será de 1200 mrad/s siguiendo una función lineal entre ellas para valores superiores. El incremento de velocidad angular, ΔS, será igual a la velocidad angular máxima entre 10. Esto supone que para alcanzar dicha velocidad es preciso recibir 10 consignas, ya que se manda una cada 0,1 s,lo que equivale a decir que se tarda un segundo en lograr el valor final.

5.5.2.- Cinemática de la silla

El sistema de control realizado en este proyecto genera la velocidad lineal y angular de la silla (V, Ω). Sin embargo, los controladores de los motores de la silla requieren como consignas las velocidades angulares de las ruedas derecha (ω_d) e izquierda (ω_i). La relación entre todas estas variables se deduce fácilmente de la figura 5.30.

La velocidad lineal de cualquier punto sobre el eje que une las dos ruedas de la silla es el producto vectorial de la velocidad angular de la misma por la distancia de dicho punto al centro instantaneo de giro. Puesto que los vectores anteriores se mantienen siempre perpendiculares, se cumplen las siguientes relaciones escalares:

$$\begin{aligned} V &= \Omega \cdot r \\ v_i &= \Omega \cdot \left(r - D/2\right) \\ v_d &= \Omega \cdot \left(r + D/2\right) \end{aligned} \qquad (5.28)$$

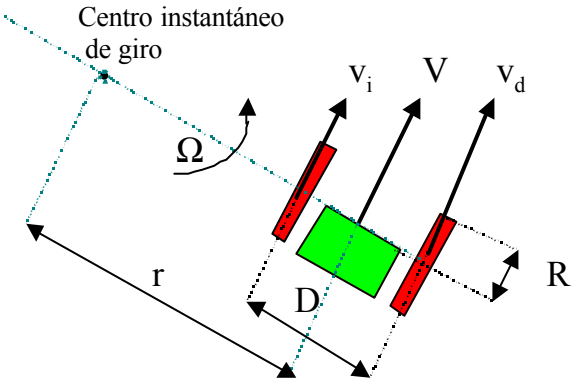


Figura 5.32. Velocidades de las ruedas de la silla

Despejando r de la primera ecuación y sustituyendo en las otras dos, se llega a las siguientes ecuaciones, que obtienen v_i y v_d a partir de V y Ω :

$$\begin{aligned} v_i &= V - \Omega \cdot \frac{D}{2} \\ v_d &= V + \Omega \cdot \frac{D}{2} \end{aligned} \qquad (5.29)$$

Por último, las velocidades angulares de las ruedas pueden obtenerse dividiendo las velocidades lineales anteriores por el radio, R, de las mismas.

$$w_i = \frac{1}{R} \cdot \left(V - \Omega \cdot \frac{D}{2} \right)$$
$$w_d = \frac{1}{R} \cdot \left(V + \Omega \cdot \frac{D}{2} \right) \qquad (5.30)$$

Otras relaciones de interés son las que permiten deducir la velocidad lineal y angular de la silla en cada instante de tiempo a partir de las velocidades ω_i y ω_d de cada rueda, obtenidas a través de los "encoders" de la silla (bloque "Cinemática Inversa" de la figura 5.27). Estas relaciones resultan de las anteriores:

$$V = \frac{(w_i + w_d) \cdot R}{2}$$
$$\Omega = \frac{(w_d - w_i) \cdot R}{D} \qquad (5.33)$$

5.5.3.- Control a bajo nivel

El control a bajo nivel es un control en lazo cerrado para cada una de las ruedas mediante un controlador PI. La velocidad angular de los motores se obtiene mediante dos encoders ópticos colocados en los ejes de los mismos. Estas velocidades se realimentan a dos circuitos integrados, llamados "Neuron-Chips", en los que se implementa el controlador PI. Estos circuitos generan una señal PWM a partir de un código digital de entrada de 8 bits, de forma que la señal PWM tiene un ciclo de trabajo del 50% cuando el dato de entrada es igual a cero. El "Neuron-Chip" recibe las consignas de velocidad angular de las ruedas, generadas por el control a alto nivel, a través de una red LonWorks de ECHELON, e implementa el siguiente controlador PI:

$$\omega(n) = K_P e(n) + K_I \sum_{N=0}^{n} e(N) \qquad (5.32)$$

donde $\mathsf{T}(n)$ es la señal de control en el instante n, e(n) es la señal de error, K_P y K_I son las constantes proporcional e integral del controlador. El período de muestreo del controlador se obtiene de la señal de reloj de la tarjeta, siendo de 10 ms. Las constantes del PI han sido obtenidas de forma experimental

para obtener el mayor margen de estabilidad sin variaciones. De este modo se obtiene la velocidad deseada incluso para grandes cambios de carga.

Las señales PWM atacan a dos drivers SP601 ("Half brigde") que controlan la activación de cuatro transistores MOSFET de un puente en H. De esta forma se obtiene un control de velocidad de cada una de las ruedas.

5.6. CONCLUSIONES

En este capítulo se ha explicado el método desarrollado para realizar el guiado de la silla de ruedas usando seguimiento facial. Dicho guiado se lleva a cabo mediante comandos que son activados con movimientos de cabeza, ojos y boca del usuario. Los comandos de velocidad son controlados mediante acciones de movimientos de cabeza. Para su detección se ha empleado un seguidor del objeto segmentado piel, sobre la imagen, mediante estimación por filtro de Kalman. Las acciones asociadas a los comandos especiales, ADELANTE/ATRÁS y ACTIVO/INACTIVO, han sido guiñar un ojo y ocultar los labios respectivamente. Su detección se realiza analizando los huecos existentes en el objeto segmentado piel e imponiendo una serie de restricciones geométricas. Una máquina de estados realiza el control a alto nivel de la silla, generando las consignas de velocidad lineal y angular, en función del tiempo, e imponiendo una serie de restricciones en función de la planta. Estas velocidades se envían a un bloque de control a bajo nivel donde se han implementado dos controladores PI, uno para cada rueda.

6. IMPLEMENTACIÓN Y RESULTADOS

6.1.- INTRODUCCIÓN

A la hora de comprobar la eficacia del sistema de guiado propuesto en esta tesis, se han cubierto tres fases de pruebas. La primera ha consistido en la programación de los algoritmos propuestos y la comprobación de su eficacia en comparación con otros métodos propuestos por otros autores. Esta fase debe ir seguida de la realización de pruebas sobre un simulador, que permita ajustar los algoritmos propuestos y comprobar, de una manera segura la viabilidad del sistema de guiado. Finalmente, se han realizado pruebas sobre una plataforma real en entornos interiores y en circunstancias de operación normales.

Como ya se comentó en el capítulo 3, esta tesis es una parte del proyecto SIAMO y por lo tanto debe adaptarse a la arquitectura general concebida para el mismo. Para la realización de las pruebas se ha tomado una de las sillas disponibles, se le ha dotado de las placas de control de motores, de un PC, para la ejecución de los algoritmos, y de una red LonWorks de ECHELON para comunicar el PC con los drivers de los motores. Asimismo, se ha dotado al prototipo básico de una estructura metálica para ubicar una cámara enfrente del usuario y de una plataforma para ubicar el PC.

6.2.- DESCRIPCIÓN DE LA PLATAFORMA DE PRUEBAS

Inicialmente se realizará una descripción física de la plataforma utilizada. Se trata de un modelo comercial de silla de ruedas motorizada GARANT 63E-PRO de la empresa INVACARE. Consta de una plataforma cuadrada (ver figura 6.1) con dos ruedas "locas" en la parte frontal y dos ruedas motrices en la parte trasera movidas por dos motores de forma independiente. En el eje trasero de la silla están colocados los dos motores D.C., que además disponen de una reductora x32 y de "encoders" ópticos que dan 200 pulsos por vuelta.

Los motores proporcionan la potencia necesaria para desplazar la silla y controlar tanto la velocidad como la dirección, aplicando diferentes pares sobre cada una de ellas. Un parámetro importante es la distancia entre las 2 ruedas, la cual depende de la anchura de la plataforma. Es deseable que las ruedas se encuentren lo más separadas posible, ya que de esta forma se mejora la estabilidad estática y dinámica de la silla.

En la silla de ruedas usada en esta tesis (ver figura 6.1), la distancia entre las ruedas es de 540 mm. La distancia entre el eje trasero y el soporte de las ruedas "locas" delanteras es de 500 mm, entre las ruedas "locas" es de 470 mm, el radio de las ruedas motrices es de 160 mm y el de las locas de 80 mm. Sobre esta estructura básica se ha añadido un soporte metálico a una distancia de 1,1 m del suelo y a 80 cm del usuario. En la figura 6.2 se muestra una foto del aspecto real de la silla.

La arquitectura concebida para este prototipo se muestra en la figura 6.3. Como se puede observar, se trata de un sistema de control distribuido basado en la tecnología LonWorks de ECHELON. La idea consiste en descentralizar las tareas del sistema, incrementando la inteligencia de los dispositivos periféricos. De esta forma se aumenta la capacidad del mismo sin incrementar los requerimientos de computación de los procesadores e incluso reduciendo la cantidad de información necesaria en los buses locales del sistema.

El corazón de los nodos de los sistemas construídos con esta tecnología es el "Neuron Chip" [García et al.,97]. Se trata de un chip multiprocesador que realiza las tareas de comunicación con la red a la vez que controla los sensores o actuadores asociados al nodo.

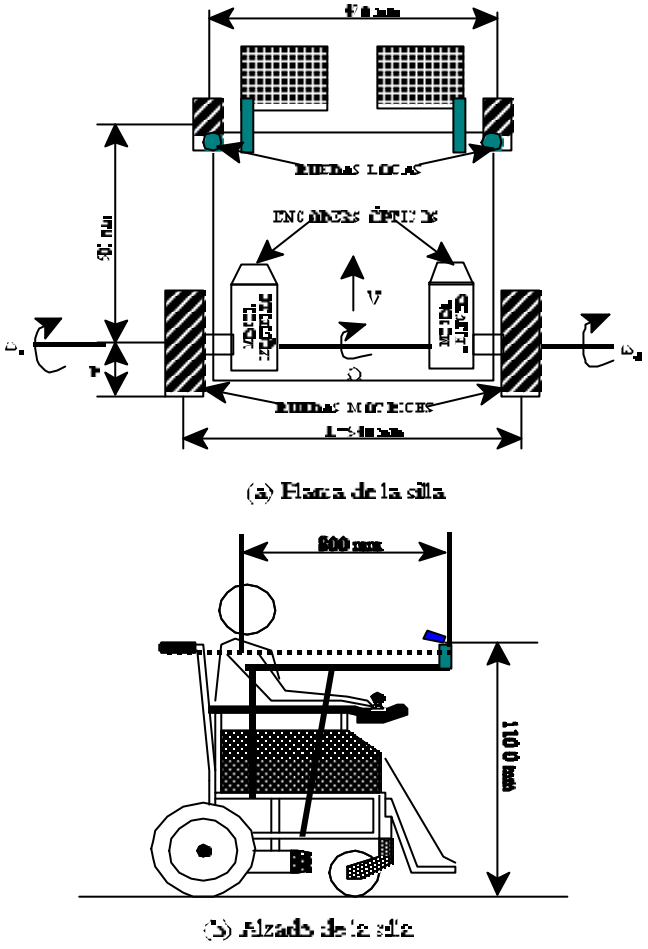


Figura 6.1. Vistas del prototipo usado

Figura 6.2. Aspecto real del prototipo

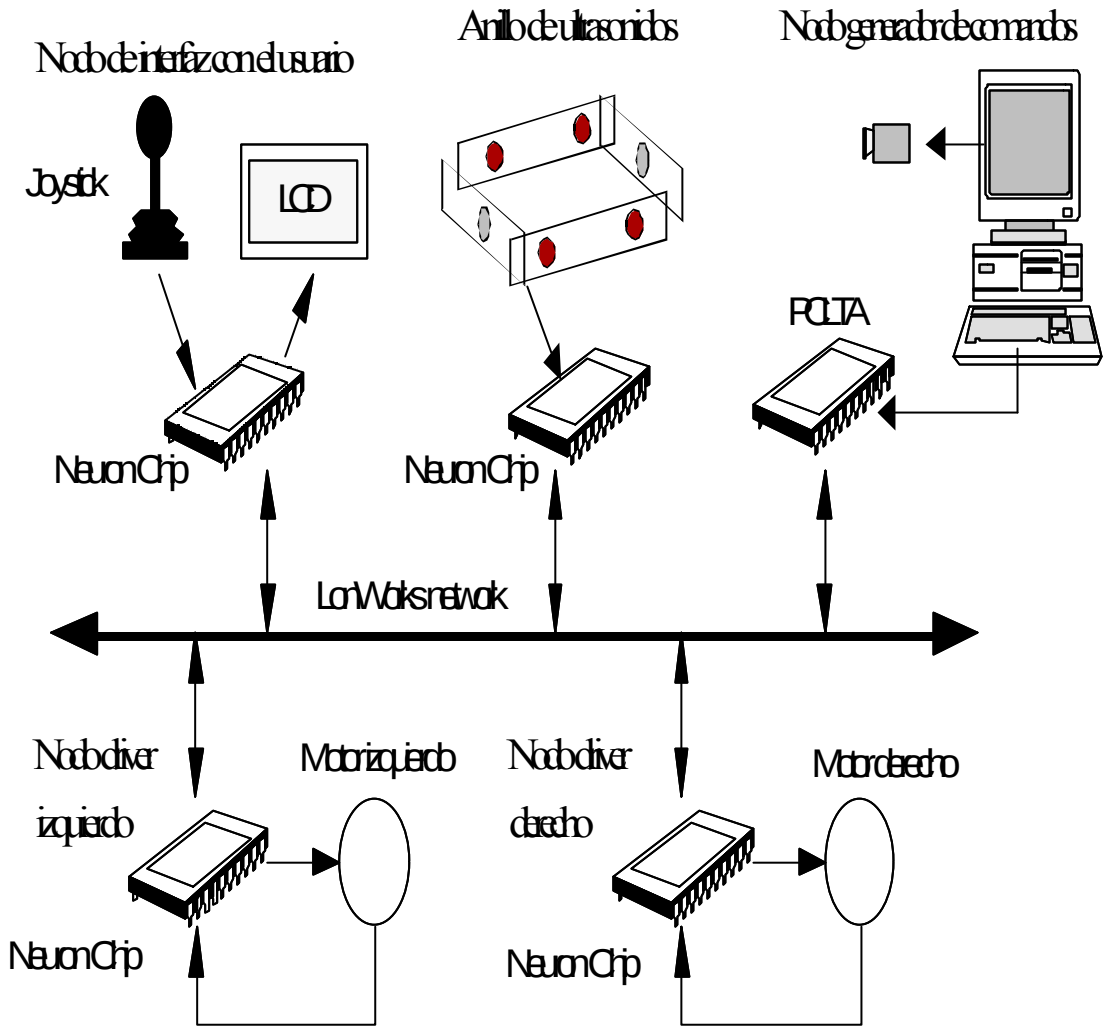


Figura 6.3. Arquitectura del sistema

El "Neuron-Chip" incluye (en firmware y en hardware) las siete capas de comunicaciones especificadas en el protocolo OSI. También es capaz de desarrollar tareas de control por sí mismo: en sus pines de I/O se pueden implementar funciones como generación de señales PWM, interfaces series y paralelos, funciones de contadores/timers, etc. Otras características de interés son las siguientes:

- *Admite diferentes medios de comunicación*: el más común es a través de un bus formado por seis hilos (opción elegida en esta tesis), radio, infrarrojo, fibra óptica.

- *Capacidad de computación*: los dispositivos LonWorks se pueden programar en un

lenguaje de alto nivel (Neuron C)

- *Facilidad de Interfaces de I/O*: dispone de distintos modos de I/O fácilmente programables a través de funciones de alto nivel en firmware, requiriendo un mínimo de componentes externos.

El sistema consta de cinco nodos: generador de comandos, drivers de los motores derecho e izquierdo, interfaz con el usuario y anillo de ultrasonidos.

El nodo generador de comandos se ha implementado sobre un PC. En él se ejecutan los siguientes procesos: visión, máquina de estados generadora de comandos, control a alto nivel y cinemática de la silla. Para establecer la comunicación entre el PC y la red se ha utilizado una placa comercial llamada PCLTA. Dicha placa se pincha en el bus ISA del PC y, mediante un "Neuron Chip" 3120, manda las consignas de las velocidades angulares de cada rueda a los drivers derecho e izquierdo. La comunicación con la PCLTA se establece a través de un enlace DDE (Dynamic Data Exchange) [Echelon, 95].

Los nodos de los drivers derecho e izquierdo realizan el control en lazo cerrado de cada una de las ruedas. Generan las señales PWM dando simplemente el ciclo de trabajo requerido a uno de los objetos de I/O. Estas señales se aplica directamente a unos drivers SP601 que controlan los transistores del puente en H. Asímismo, capturan las señales de los encoders, para obtener la velocidad real de cada rueda, a través de otro objeto I/O. Ejecutan el algoritmo de control PI, escrito en "Neuron C" y cargado en el propio chip. La modificación del software es muy sencilla usando las múltiples funciones firmware incluidas en el "Neuron Chip". Teniendo en cuenta que la constante de tiempo de los motores de tracción es del orden de 100 ms, el algoritmo de control puede ser ejecutado sin problemas. Hay que destacar que los controladores se implementan sobre los propios nodos de comunicaciones y que el software de los mismos puede ser actualizado sobre la EEPROM interna de los "Neuron" a través de la red.

El nodo de interfaz de usuario consta de un joystick, que permite el control de la silla a través de la mano del usuario y de un LCD para visualizar los mensajes de salida y de estado de la misma. Asímismo dispone de una serie de botones que completan la comunicación con el usuario.

El nodo de ultrasonidos está formado por un anillo de sensores utilizados para evitar colisiones con paredes u obstáculos. Este módulo incrementa la seguridad del sistema ante posibles fallos o ante comandos erróneos generados por el usuario. Está concebido como un sistema reactivo, en caso de posible colisión bloquea las órdenes del usuario, se para y devuelve el control al usuario.

Los dos últimos nodos explicados no han sido implementados sobre el prototipo, en esta primera fase. No obstante, se considera interesante su introducción en un futuro.

Para aumentar la controlabilidad de la silla se utilizan señales sonoras para indicar al usuario la activación de los comandos. Se ha usado el criterio de usar el mínimo número de sonidos que ofrecieran la máxima información al usuario. Así, cuando el sistema se encuentra buscando el objeto piel, se emite un tipo de sonido. Cuando se produce una conmutación de los comandos de ACTIVO/INACTIVO y SENTIDO, se emite otro y la activación de cualquier comando de velocidad provoca la activación de otro diferente. Es el propio usuario quien, en función del movimiento que ha realizado, se da cuenta del comando en concreto que ha activado, comprobándolo un pequeño instante después mediante el movimiento de la silla.

6.3.- DESCRIPCIÓN DEL SISTEMA DE VISIÓN

Cuando surge la necesidad de dotar a un móvil con un sistema de visión artificial, se pueden plantear varias alternativas: a) utilizar un hardware específico de altas prestaciones y generalmente alto coste o b) optar por una plataforma hardware estándar, de bajo precio y prestaciones aceptables. Si se considera que el sistema es viable, crematísticamente hablando, la elección es claramente la segunda. Si, por el contrario, el sobreprecio no es inconveniente, la balanza debe decantarse hacia la primera opción.

Como el trabajo desarrollado en esta tesis va encaminado a la realización de un prototipo de silla de ruedas para minusválidos, se ha elegido un sistema de visión artificial de bajo coste y prestaciones aceptables, constituido por una cámara en color CCD, un "frame-grabber" y un PC-Pentium II, cuyas características son:

ORDENADOR PENTIUM II

Microprocesador	Pentium II
Velocidad de reloj	400 Mhz
Memoria RAM	64 MB
Disco Duro	2 GB
Monitor	SVGA

TARJETA DE ADQUISICIÓN DE DATOS ("FRAME-GRABBER")

Modelo	Meteor II de MATROX
Bus	PCI con slot único
Entrada	Vídeo compuesto estándar y no estándar en color y b/n
Velocidad	Transferencia en tiempo real al sistema o a la memoria VGA a más de 130 MB/s.
Resolución	1280x1024x8 bits 1024x768x16 bits 800x600x24 bits

CÁMARA EN COLOR SONY XC-99p

CCD	752(H)x582(V)
Señal de salida	PAL
Soporte para lente	Rosca C
Óptica	12 mm

6.4.- EL SIMULADOR

Con objeto de probar la viabilidad del sistema de guiado propuesto, de una forma real y segura, así como de servir de plataforma de entrenamiento para los posibles usuarios del sistema, se ha realizado un simulador de la silla en tiempo real, basado en un entorno virtual 3D de la primera planta de la antigua Escuela Politécnica. El objetivo planteado fue la creación de una aplicación que diera soporte a un entorno 3D y a un modelo matemático de un móvil que permitiera probar algoritmos de control. Para cumplir este objetivo se analizaron dos posibilidades de diseño:

1. Creación de un entorno dedicado. Diseñar por medio de programación estructurada tradicional

un entorno dedicado y no modificable, provisto además de las respectivas rutinas de detección de colisiones, animación y representación gráfica tridimensional. Ésta sería la opción de diseño a corto plazo de un entorno determinado, teniendo el defecto de que cualquier cambio en éste o en el móvil diseñado acarrearía grandes y costosas modificaciones.

2.- Creación de un entorno de simulación parametrizable. Diseñar haciendo uso de las características de la orientación a objetos (aunque también hubiera sido posible realizarlo con programación estructurada tradicional) un motor capaz de representar y simular un entorno 3D capturado de un archivo en el que se encontrase el entorno definido con unas determinadas normas, y un móvil definido igualmente en un archivo e inmerso en el entorno capturado. Todo ello también provisto de los algoritmos de detección de colisiones, animación y representación gráfica adecuados. Esta propuesta era más costosa de diseñar y desarrollar, teniendo la ventaja de que su reutilización en cualquier otro tipo de proyecto de simulación, o simplemente el cambio de la parametrización del móvil o del entorno son fácilmente introducibles, eliminando costes y tiempo en posteriores desarrollos

De las dos opciones expuestas se optó por la segunda, ya que con ella se podían simular entornos y móviles diferentes de una forma sencilla, aunque ello supusiera un mayor tiempo de desarrollo. Se han utilizado como herramientas de diseño y desarrollo Microsoft Visual C++ 5.0 y los APIs de representación gráfica OpenGL y GLUT, todo ello bajo Microsoft Windows NT 4.0 WorkStation.

Para realizar la representación en pantalla del entorno, en función del movimiento del móvil, se plantearon dos opciones:

- Crear *rutinas propias* de representación visual (representación tridimensional, ocultación de caras no visibles, texturado, iluminación, etc); labor costosa y no crítica para la consecución de nuestros objetivos, además de ser necesario profundizar en complejos cálculos geométricos y matemáticos.

- Hacer uso de un *API o librería de representación tridimensional*, ya creada, que diera soporte para representar gráficamente el entorno y simular el movimiento. Para el entorno de desarrollo elegido (Microsoft Visual C++ 5.0) existían dos posibles alternativas de este tipo: Microsoft DirectX y Silicon Graphics OpenGL.

Se optó por la segunda opción y en particular por el *API OpenGL versión 1.1,* como herramienta de representación, por tener las siguientes ventajas frente a DirectX:

a. *Mayor facilidad de aprendizaje*: OpenGL está constituido por un conjunto mínimo de primitivas (funciones) que permiten asimilar los conceptos principales referentes a los gráficos tridimensionales y adaptarlos a nuestras necesidades.

b. *Gran Extensibilidad*: Por su diseño, se hace posible, aparte de adaptarlo a nuestras necesidades, expandir su capacidad para crear nuevas funcionalidades.

c. Portabilidad: No está ligado a una plataforma (entorno de ventanas) determinada, sino que es el programador el que lo inserta en la que quiere, (DirectX está estrechamente ligado a las MFCs (Microsoft Foundation Classes), lo que lo hace poco portable), siendo posible migrar el código a otros entorno de ventanas y plataformas (UNIX, Linux, Solaris, Irix, X-Windows, Motif, ..) sin necesidad de cambiar de API. Además se hace uso de la librería auxiliar GLUT (Graphics Library Utility Toolkit) que nos proporciona abstracción con respecto al entorno de ventanas en el que se programe, siendo la librería la que se encarga de la manipulación básica de las ventanas.

d. El hardware gráfico existente en el mercado evoluciona proporcionando paralelamente drivers específicos para aceleración 3D por hardware tanto para OpenGL como para DirectX; teniendo más *solidez* en este aspecto OpenGL.

6.4.1. Estructura del motor de simulación 3D

Se entiende por motor de simulación 3D (en terminología anglosajona "*3D engine*") un ente capaz de procesar información sobre un entorno y uno o varios móviles, realizando el control y la simulación de la interacción de éstos con el entorno, teniendo además la posibilidad de ofrecer una representación gráfica de ello.

Para la codificación del entorno existen numerosas técnicas de descripción espacial: enumeración espacial, BSPs (Binary Space Partitioning trees), geometría planar, etc; de las que se escogieron las características más adecuadas a nuestro proyecto, surgiendo lo que denominamos *descripción por*

enumeración espacial-euclídea. Se basa en subdividir la planta de un escenario en sectores planos de igual nivel (que pueden tener diferentes propiedades), con la característica de que todo es descriptible y representable en coordenadas reales.

El motor de simulación consta de tres partes:

1. Entorno. Contiene la descripción geométrica del escenario virtual del simulador. Da la base del conocimiento necesario para realizar el movimiento y hacer su representación visual. Para definir un entorno 3D mínimo se han identificado las siguientes entidades:

• *VPunto*: Coordenadas (x,z) de un punto en un plano (posteriormente adoptará una altura determinada (y)).

•*VLinea*: Formada por objetos de la entidad VPunto que la definen, las cuales además poseen propiedades de color y textura con que serán representadas.

• *Vtextura*: Entidad que soporta la carga de texturas desde ficheros en formato bmp, así como la aplicación de éstas a los diferentes objetos.

• *VSector*: Formada por un subconjunto de objetos de la entidad VLinea que definen un polígono cerrado al que se le dota de unas alturas(y) mínima(suelo) y máxima(techo), además de los colores y texturas con que serán representados. Esta entidad da soporte a sectores planos de espacio, con unas determinadas características fijas.

• *VPuerta*: Entidad derivada de la anterior a la que se añade la funcionalidad de que el móvil o móviles que se muevan por el entorno final puedan interactuar con ella de forma que varíen su estado de abierto a cerrado y viceversa.

• *VMapa:* Da soporte al entorno a representar, contiene miembros de todas las anteriores y define el entorno con el que interactuarán los móviles. Posee la capacidad de capturar mapas de entornos definidos en ficheros con unas determinadas normas de codificación que se han fijado.

2. Móvil. En cada ciclo de simulación toma los valores de velocidad lineal y angular mandados por el sistema de visión, los introduce en el modelo del móvil, obtiene los cambios de posición ( ) x,) y) así como de orientación ( ) 2) del mismo y actualiza su posición en el espacio de trabajo. Para permitir que el objeto móvil interactúe correctamente con el entorno se diseñaron las siguientes entidades:

• *VPunto*: Ya definida, indica la posición del móvil.

• *VVector*: Análoga a VLinea para indicar la orientación del móvil ( ) 2).

• *VMovil*: Efectúa los cálculos de detección de colisiones, da volumen al móvil, definido mediante propiedades como su anchura y altura, y produce una animación, situándolo en el punto de vista adecuado.

• *VPlayer*: Con el fin de encapsular la entidad VMovil y hacerla más robusta se implementa VPlayer como derivada de ésta, añadiendo pseudocomandos de movimiento y la posibilidad de capturar las características de un fichero de estructura predefinida.

3. Render. Es el encargado de mostrar las vistas 2D y 3D del entorno virtual, en función de los movimientos del móvil. Produce la unión del motor 3D interno hasta ahora definido (llamemos así a la parte computacional no gráfica de éste) con un entorno de ventanas determinado (parte gráfica). En la figura 6.5 se muestra un ejemplo de simulación con vistas 2D y 3D. También dispone de un interfaz de usuario (ver figura 6.5), en el que aparece una pantalla informativa del estado del sistema y una barra de menú en la que se pueden seleccionar las siguientes opciones:

• Representación 2D y 3D en ventana o en pantalla completa.
• Seleccionar el tipo de textura entre: plana y texturada (esta opción es útil en máquinas poco potentes seleccionando la textura plana, ya que disminuye el tiempo de cómputo). •Grabar y reproducir trayectorias simuladas.
•Mostrar el rastro de una trayectoria recorrida en la representación 2D.
•Realizar el control por teclado o mediante la comunicación serie con el proceso generador de comandos.
•Cargar un nuevo modelo o un nuevo entorno virtual.

Como características principales del simulador caben destacar: 1) Su ejecución en tiempo real (20 imágenes/). 2) La posibilidad de visualización gráfica en 2D y 3D. 3) Utilización como escenario de un modelo simplificado de la antigua E.P. de la Universidad de Alcalá, pudiendo introducir nuevos entornos. 4) La utilización de un modelo matemático simplificado de una silla de ruedas, el cual puede ser facilmente modificado. 5) El almacenamiento y reproducción de trayectorias y exportación de datos a Matlab.

Para optimizar el funcionamiento del sistema se ha dividido el proceso en dos partes: proceso generador de comandos y simulador. Cada uno de ellos se ejecuta sobre un PC independiente. El proceso generador de comandos enviará la velocidad lineal y angular al simulador vía RS-232.

En la figura 6.5 se muestra un diagrama de bloques de todo el proceso de simulación explicado con las partes que lo componen.

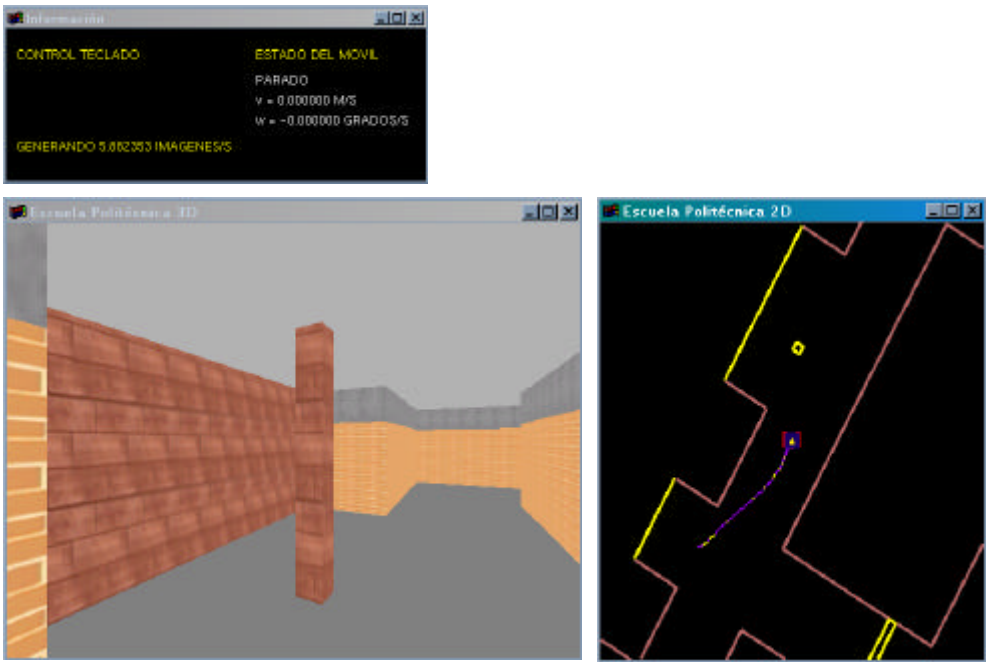


Figura 6.4. Entorno de simulación

6.4.2. Modelo de la silla

Para poder simular el movimiento de la silla de ruedas es necesario disponer de un modelo lo

suficientemente preciso de la planta a mover [Espinosa 98]. Como el objetivo del simulador es probar el sistema de guiado mediante seguimiento facial, se utilizará un modelo simplificado de la misma que asume que las consignas de velocidad que se le dan a los motores son realmente las que tienen, despreciando el retardo de los mismos. No hay que olvidar que en el modelo real existe un lazo de realimentación a bajo nivel que hace que la velocidad de los motores sea la de la consigna y de forma estable.

El modelo de silla desarrollado presenta las siguientes dos entradas de control independientes:

- La **velocidad lineal** (V) del punto central del eje imaginario que une las dos ruedas activas.
- La **velocidad angular** (W), o velocidad de variación del ángulo formado por el eje longitudinal de la silla respecto a otro externo fijo (sistema cartesiano de referencia).

La variable con la que se caracterizará el movimiento de la silla, es decir, la salida del modelo, será la ubicación exacta de la misma respecto a un sistema de coordenadas externo fijo. En la figura 6.7 se muestra un esquema de las variables de entrada y de salida utilizadas en el modelo.

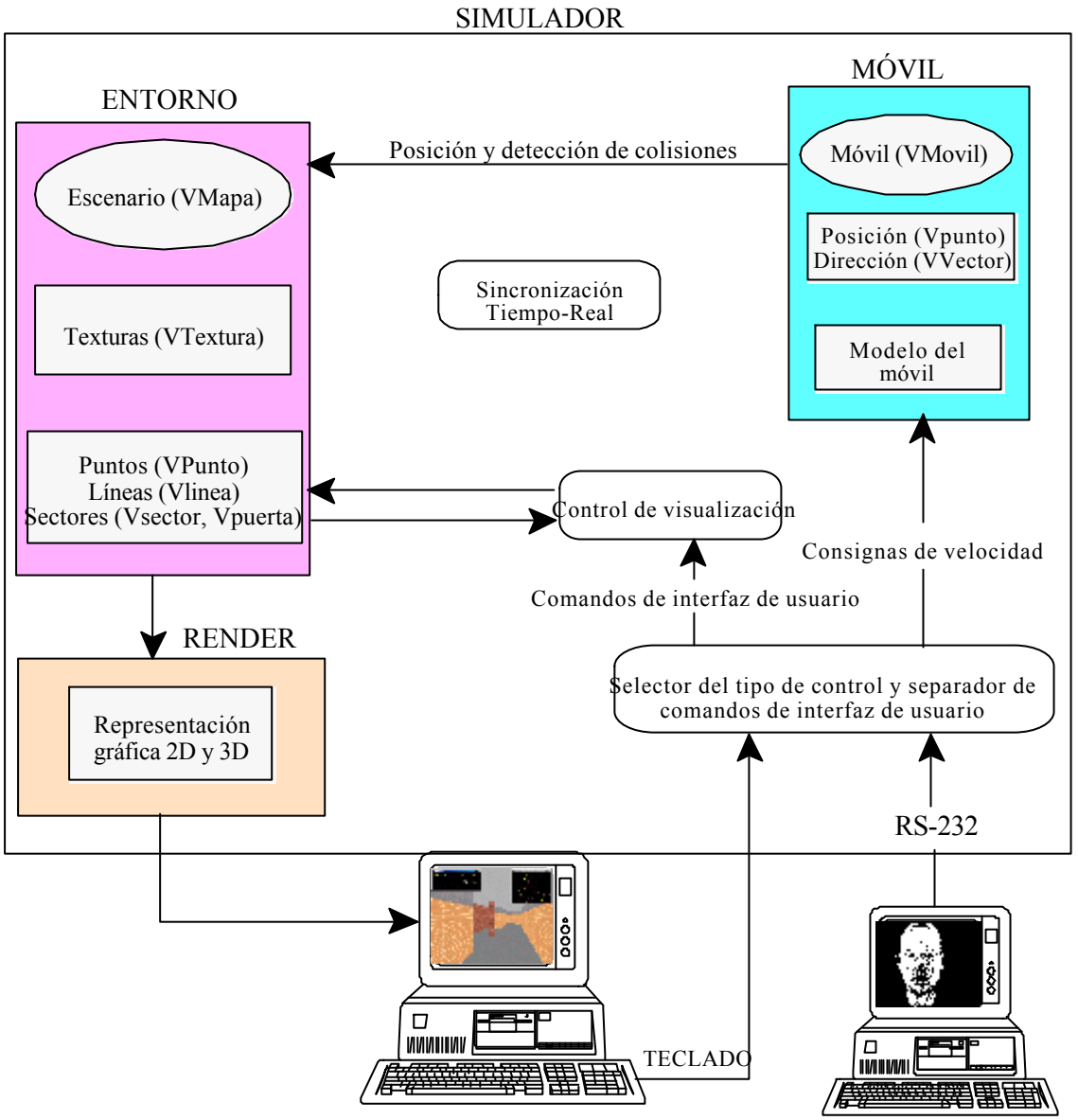


Figura 6.5. Diagrama de bloques del simulador

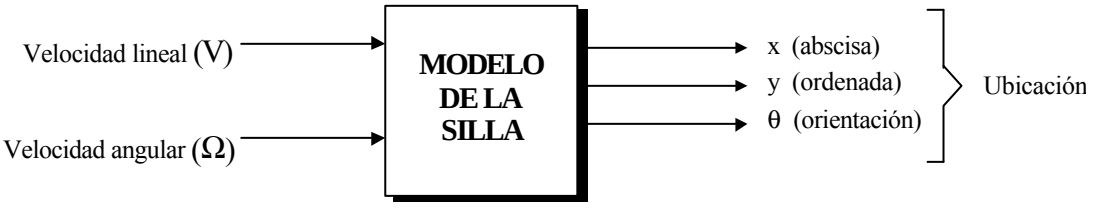


Figura 6.6. Esquema de variables de entrada/salida del modelo de la silla

La respuesta de la silla ante la aplicación de unas determinadas consignas de velocidad angular y lineal se manifiesta como una variación en su ubicación (posición más orientación). Para caracterizar la ubicación de la silla con respecto a un sistema de coordenadas fijo, se requiere el conjunto de las siguientes variables (ver figura 6.7):

- **(x,y)** son las coordenadas cartesianas del punto central del eje transversal de la silla, con respecto a dicho sistema de coordenadas.
- **q** es la **orientación de la silla**, entendida como el ángulo que forman el eje longitudinal de la misma (hacia adelante) con el semieje positivo de abscisas del sistema de referencia fijo.

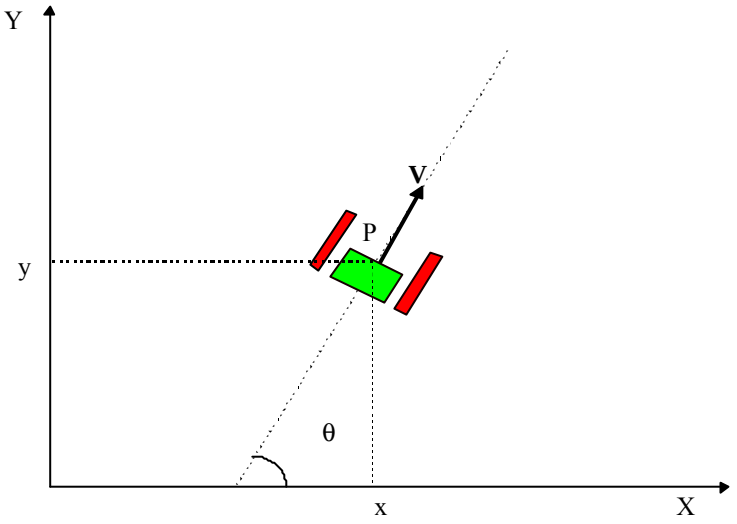


Figura 6.7. Parámetros que definen la ubicación de la silla.

De esta forma, mediante vectores del tipo **(x,y,q)**, queda caracterizada la ubicación exacta de la silla en un sistema de coordenadas fijo. Puesto que las ruedas de la silla tienen dirección fija (la del eje longitudinal de la silla), y el centro instantáneo de giro de cualquier vehículo sobre ruedas se encuentra en la intersección de las perpendiculares de todas sus ruedas, es obvio que la silla siempre girará en torno a algún punto sobre la prolongación del eje imaginario que une las dos ruedas activas. Por este motivo, la velocidad en el punto central de dicho eje imaginario será un vector perpendicular al radio de giro, y, por tanto, con la dirección del eje longitudinal de la silla. En resumen, dicho punto central P se moverá con lo que se ha dado en llamar **velocidad lineal de la silla (V)**, que es una de las variables de entrada al modelo.

Sabiendo además que la **velocidad angular de la silla (W)** se define como la variación de la orientación de la misma (q), es inmediato deducir las ecuaciones del modelo de la silla:

$$\begin{aligned} \dot{\theta} &= \Omega \\ \dot{x} &= V \cdot \cos\theta \\ \dot{y} &= V \cdot \operatorname{sen}\theta \end{aligned} \qquad (6.1)$$

Conocidas las velocidades lineal y angular actuales de la silla V(n) y W(n) y la ubicación actual de la misma [x(n), y(n), q(n)], puede predecirse la ubicación de la silla en el siguiente instante de muestreo mediante la discretización (por el método de diferencias en adelanto) de las ecuaciones del modelo. La predicción será tanto más precisa cuanto más pequeño sea el período de muestreo Ts.

$$\begin{aligned} x(n+1) &= x(n) + Ts \cdot V(n) \cdot \cos(q(n)) \\ y(n+1) &= y(n) + Ts \cdot V(n) \cdot \operatorname{sen}(q(n)) \\ q(n+1) &= q(n) + Ts \cdot \Omega(n) \end{aligned} \qquad (6.2)$$

Según [López, 99] para asegurar una correcta aproximación del modelo discreto de un móvil al continuo, trabajando con velocidades que no excedan de 1m/, es conveniente utilizar periodos de muestreo de 100 m o inferiores. En nuestro caso la velocidad máxima permitida a la silla será de 1 m/ y el período de muestreo es menor de 100 m, por lo que se cumplen las condiciones expuestas.

6.4.3. Resultados del simulador

El simulador se ejecuta sobre un AMD K6-2 a 350 Mhz con 128 Mbytes de RAM y dotado de una tarjeta gráfica con el chip acelerador Nvidia Riva TNT y genera 20 imágenes por segundo, trabajando con óptima calidad visual. Por lo tanto, se han obtenido óptimos resultados con el simulador sin necesidad de utilizar equipos de aplicación específica (como podría ser una estación Silicon de precio superior al millón de pesetas) y con un equipo de un coste aproximado de 200.000 ptas.

El sistema ha sido testeado con personas de diferentes razas obteniendo buenos resultados en el guiado. En la figura 6.8 se muestra un ejemplo de guiado donde se puede ver la trayectoria seguida por la silla en una vista 2D del entorno de simulación. En la figura 6.9 se dan la evolución de las variables de velocidad lineal y angular en función del tiempo para la trayectoria seguida en la figura 6.8.

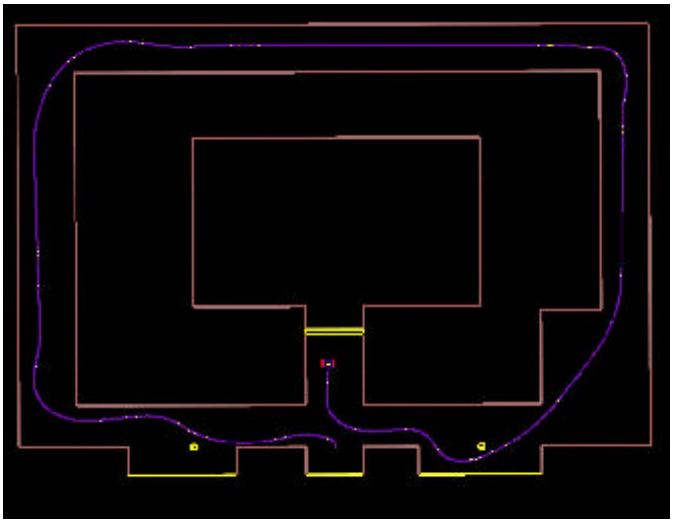


Figura 6.8. Trayectoria simulada de la silla

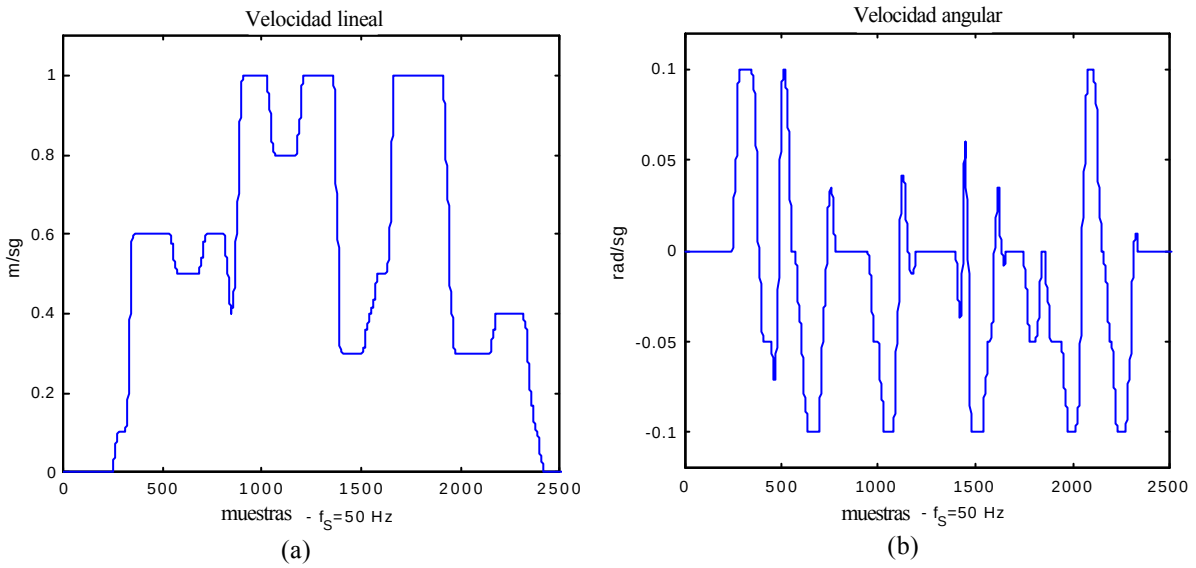


Figura 6.9. (a) Velocidad lineal (b) Velocidad angular

6.5.- PRUEBAS DE GUIADO

Las pruebas realizadas para la fase primera, programación de los algoritmos y comparación con otros métodos, han sido descritas en los correspondientes capítulos de esta tesis. Las pruebas de la fase segunda, simulación, han sido descritas en el punto anterior. Por lo tanto, únicamente quedan por cubrir las pruebas de la tercera fase consistentes en medir las prestaciones del sistema de guiado sobre la silla

real y en condiciones de operación normales.

En la tabla 6.1 se muestran los tiempos de computación de cada uno de los procesos de que consta el sistema, para imágenes con una resolución de 128x128 pixels y con un procesador Pentium II a 400 Mhz. Hay que tener en cuenta que los tiempos de los procesos de visión son orientativos ya que dependen de la imagen capturada, la cual es función del usuario y del fondo que haya en cada momento. Por otro lado, el proceso de "clustering" de la imagen solamente se ejecuta al comienzo y cuando se produce una pérdida del sistema, con lo que no hay que considerarlo en el funcionamiento normal.

Proceso	$T_{proceso}$ (ms)
Clustering de la imagen	730
Seguimiento facial	72
Detección de ojos	6
Detección de boca	5
Generación de comandos	11

Tabla 6.1. Tiempos de computación de los procesos con un Pentium II a 400 Mhz.

Así pues, la primera vez que se ejecuta el proceso y cuando la segmentación es errónea, se ejecutan todas las etapas y, en consecuencia, el tiempo de proceso es de 824 ms. Sin embargo, en funcionamiento continuo el tiempo se reduce a 94 ms, lo que supone tratar algo más de 10 imágenes al segundo y actualizar consignas en el bajo nivel con una cadencia de 10 por segundo.

Debido a la no existencia de sistemas de guiado similares, no se pueden establecer comparaciones con ellos. Por lo tanto, para medir las prestaciones del sistema, se dió a probar el mismo a diferentes usuarios y se les pidió que lo evalúen a través de un formulario donde estaban contemplados aspectos como controlabilidad, facilidad de manejo, etc. Las pruebas se realizaron en el Laboratio de Investigación y en los pasillos del Departamento de Electrónica.

Estas pruebas demostraron que la forma de conmutar el sentido de la marcha, guiñando un ojo, era muy poco robusta y daba gran cantidad de fallos cuando la silla se movía. Ello era debido a que los cambios de fondo y de iluminación introducían gran cantidad de ruido sobre el sistema y afectaban a la detección de los ojos, al ser objetos muy pequeños en la imagen. No hay que olvidar que se está trabajando con imágenes de resolución 128x128 pixels, de forma que un ojo ocupa unos 20 pixels.

Para solucionar el problema, se cambió la acción asociada al comando de SENTIDO por la de ocultar los labios, al ser ésta mucho más robusta por tener un tamaño mayor (el objeto boca ocupa unos 200 pixels en la imagen). Para distinguir entre la acción de conmutación ACTIVO/INACTIVO y cambio de sentido se usó la variable tiempo. De esta forma, si el usuario oculta los labios un tiempo inferior a cinco segundos se activa el comando ON/OFF y si lo hace más de cinco segundos se activa el comando de SENTIDO. Aunque puede parecer que esperar más de cinco segundos para conmutar el sentido es mucho tiempo, no hay que olvidar que el usuario debe estar parado y por lo tanto no hay peligro de que la silla se descontrole.

En la figura 6.10 se muestra la imagen de un usuario mientras evaluaba el sistema. Para probar la controlabilidad del sistema se realizaron una serie de pruebas de guiado en el laboratorio y se obtuvieron las gráficas del recorrido seguido por el usuario y de la evolución de las velocidades lineal y angular, obtenidas a partir de los "encoders" de las ruedas y aplicando un "dead reckoning".

Figura 6.10. Vista de un usuario realizando la prueba

En la figura 6.11 se puede ver la primera las pruebas, consistente en realizar únicamente movimientos de derecha e izquierda. En (a) se muestra la trayectoria seguida por el móvil, para lo cual se representa el movimiento del punto medio del eje que une las ruedas motrices. Como se observa, este punto prácticamente no se mueve al tratarse de giros. Se han realizado tres giros hacia la derecha y dos hacia la izquierda. Se han adquirido cinco muestras por segundo, lo que supone que se han invertido 100 segundos (1,66 minutos) en realizar toda la secuencia de giros. Es de destacar que cuando el valor absoluto de la velocidad angular es grande aparece una cierta componente de velocidad lineal que

compensa las variaciones de la velocidad angular, debida a la posición de las ruedas “locas”.

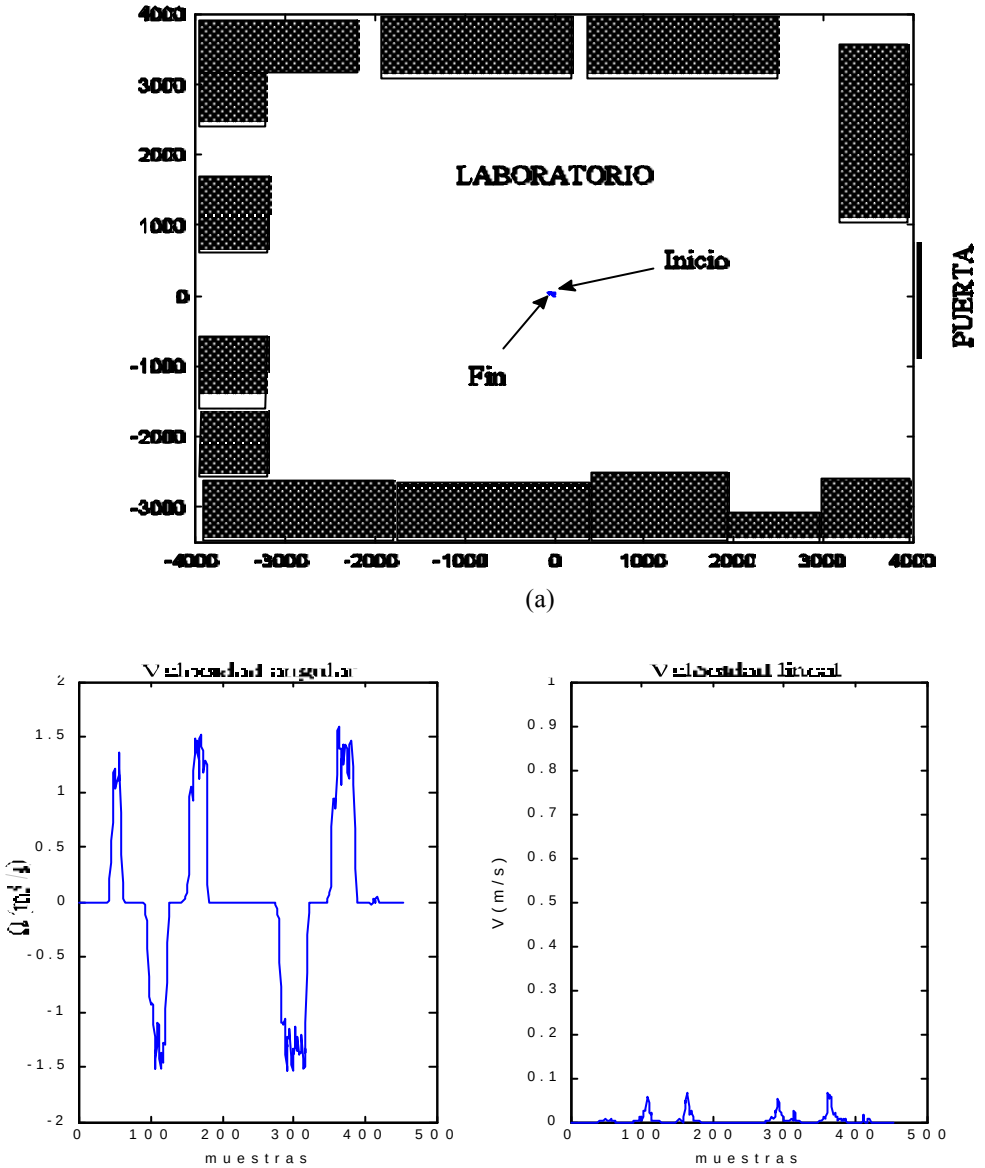


Figura 6.11. Datos obtenidos para un movimiento de derecha/izquierda

En la figura 6.12 se evalúa la aceleración del sistema y los cambios de sentido y conmutación ACTIVO/INACTIVO. Partiendo de la posición inicial se acelera la silla siguiendo una trayectoria recta y se para de repente mediante el comando ACTIVO/INACTIVO.

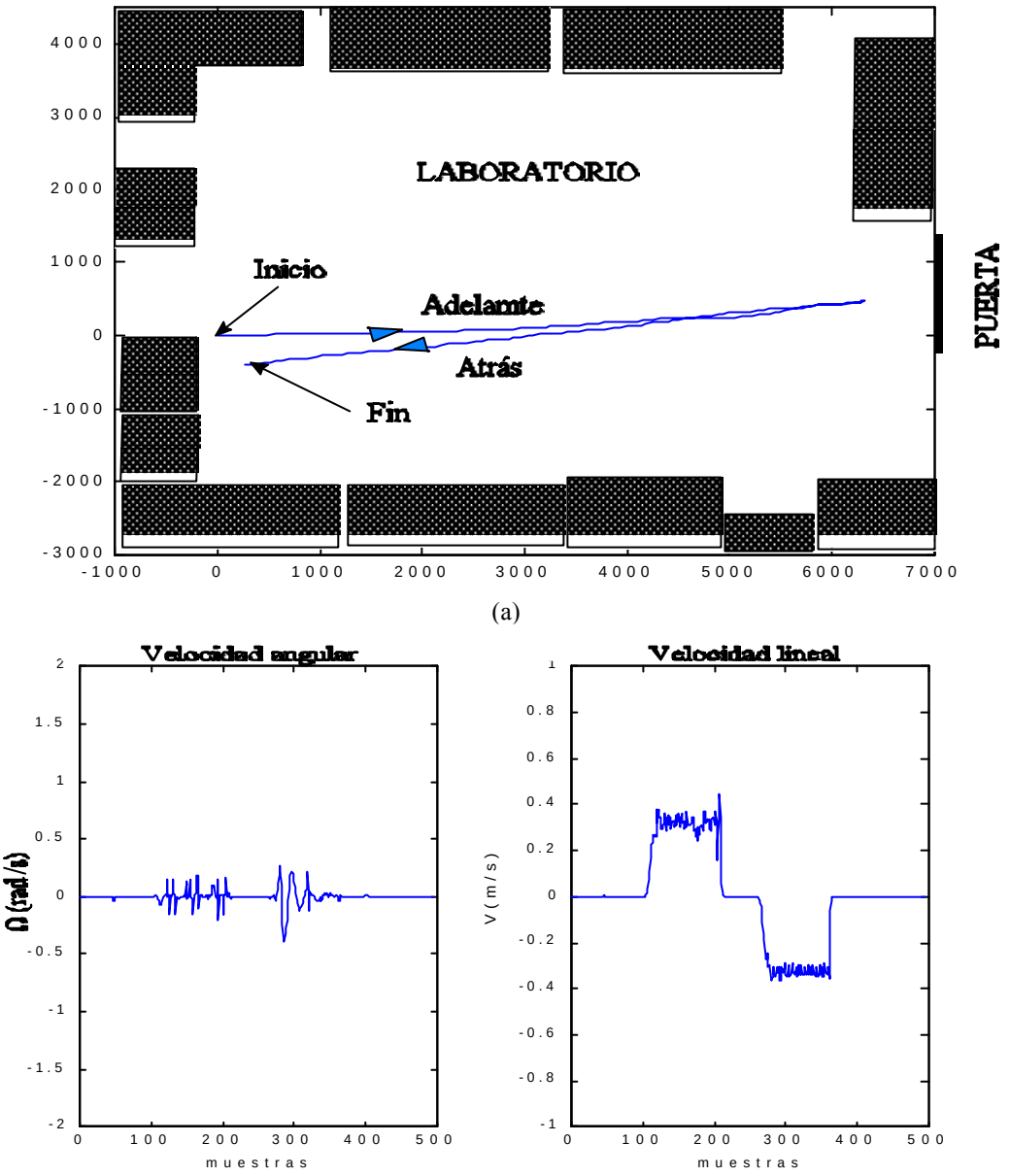
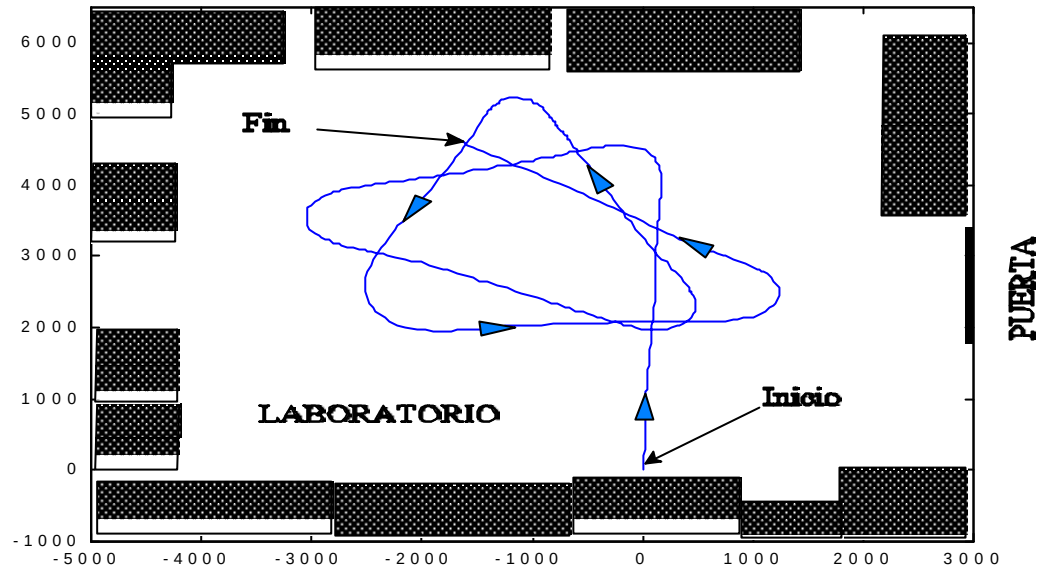


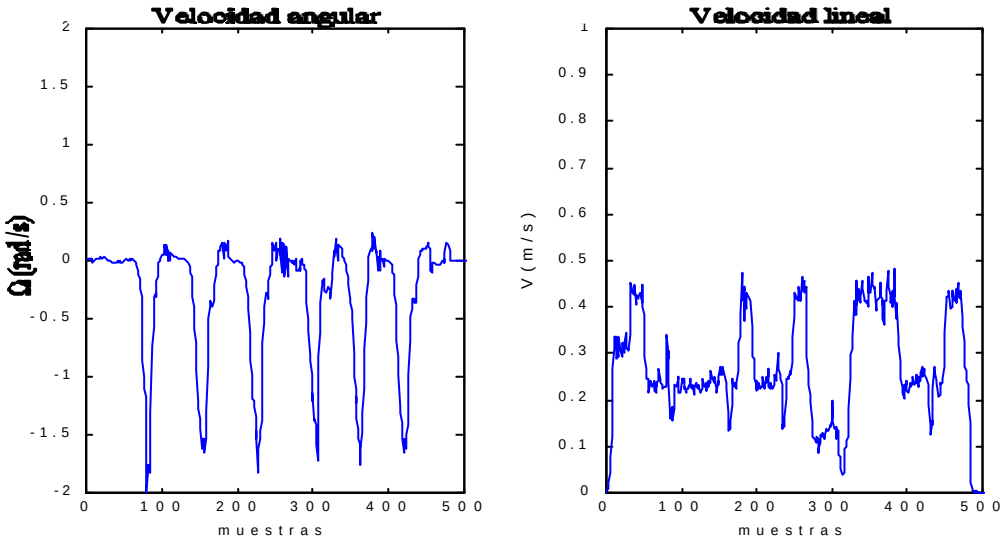
Figura 6.12. Datos obtenidos para un movimiento de adelante/atrás

En (b) se observa el proceso de aceleración y cómo la deceleración se hace de forma brusca. Alcanzada una posición, cercana a la puerta, el usuario se para, cambia el sentido de la marcha y acelera, yendo ahora hacia atrás. Cuando alcanza aproximadamente la posición inicial, la silla para de repente, al activar el comando ACTIVO/INACTIVO. Cabe destacar que pese a que únicamente se generan velocidades lineales, aparece una cierta componente de velocidad angular debido al efecto de la posición de las ruedas "locas".

Por último, en la figura 6.13 se muestra un ejemplo combinado de giros, aceleraciones y deceleraciones.



(a)



(b) (c)

Figura 6.13. Datos obtenidos para giros y movimientos hacia adelante

Inicialmente se acelera la silla siguiendo una trayectoria recta. A continuación se decelera y se gira hacia la izquierda, con lo que se produce un pico negativo en la velocidad angular. Se sigue recto con una

velocidad menor, sigue una nueva curva a la izquierda, se acelera, se decelera un poco y se ejecuta una curva a la izquierda, se acelera y decelera, nueva curva a la izquierda, se hace lo mismo y en la última curva se decelera la silla hasta pararla.

Finalmente, el sistema fué probado por cinco usuarios diferentes, después de un período de entrenamiento en el simulador, a continuación se les pasó una encuesta donde se les planteaba diez preguntas que debían evaluar entre 0 (muy malo) y 10 (muy bueno). Los resultados medios obtenidos fueron:

Pregunta	Puntuación media
Grado de invasividad del método.	9,2
Controlabilidad general del guiado.	7,0
Dificultad del guiado	6,5
Tiempo de adaptación al sistema	6
Evalúe la respuesta del giro	8,1
Evalúe la respuesta de la aceleración/deceleración	7,6
Evalúe la respuesta del cambio de sentido	6
Evalúe la respuesta del control on/off	7
Cree que la realimentación sonora es suficiente para el buen control de la silla	7,8
Considera que puede ser un sistema interesante para ser utilizado por personas discapacitadas con un gran grado de minusvalía	7

Tabla 6.2. Encuesta de evaluación del método

6.6.- CONCLUSIONES

En este capítulo se ha descrito la implementación práctica del sistema de guiado propuesto en la tesis, demostrando la viabilidad del mismo en interiores bien iluminados. Se ha demostrado experimentalmente que la acción de guiñar un ojo era poco robusta para una silla en movimiento y ha sido sustituida por la acción de ocultar los labios. El sistema ha sido evaluado por un reducido número de personas obteniendo las siguientes conclusiones:

- La invasividad del método es baja al ser un sistema totalmente pasivo y no necesitar ponerse

elementos adicionales.

- El sistema tiene cierta dificultad de guiado que se ve decrementada según se va utilizando. Hay que tener en cuenta que, al tener una cámara a 80 cm del usuario, necesita bastante sitio para maniobrar y por lo que necesita de entornos amplios con pocos obstáculos.
- Las respuestas de los giros, aceleración y decelaración son óptimas permitiendo un buen control de los movimientos.

- La respuesta de los comandos ACTIVO/INACTIVO y cambio de SENTIDO es buena, aunque en este último caso hay que esperar demasiado tiempo para que se active. También se debe destacar que cualquier falsa detección, cuando se intenta activar un comando, pone a cero el acumulador, y por lo tanto hay que esperar un tiempo adicional al establecido para que se ejecute el comando.

- La realimentación sonora es suficiente para realizar el guiado del sistema.

- El sistema funciona óptimamente en entornos interiores bien iluminados (habitaciones con todas las fluorescentes encendidas) disminuyendo su respuesta si la iluminación no es uniforme.

- Pese a no estar contemplado incialmente como objetivo de esta tesis, el sistema se probó en entornos exteriores, comprobando que su funcionamiento era muy dependiente de la uniformidad de iluminación existente. Si ésta condición se cumple el sistema funciona correctamente, sin embargo, en estos entornos, existen diferentes fuentes de luz (luz directa del sol, luz reflejada, luz artificial, etc) y aparecen sombras muy pronunciadas, que originan que el color de la cara no sea uniforme, por lo que el objeto segmentado piel no sigue la geometría establecida para la generación de comandos y por lo tanto puede dar comandos erróneos.

7. CONCLUSIONES Y FUTURAS LÍNEAS DE TRABAJO

7.1.- CONCLUSIONES

En esta tesis se ha propuesto un sistema de guiado de una silla de ruedas, pensado para personas discapacitadas, mediante seguimiento facial y a bajo coste. La principal aportación ha sido la aplicación de técnicas de visión artifical para llevar a cabo este sistema de control, no existiendo en la bibliografía consultada ningún otro prototipo similar.

Dentro de las técnicas de visión artificial aplicadas es aportación de la tesis la creación de un algoritmo específico para segmentar piel humana, independientemente del usuario y del fondo, adaptativo a los cambios de iluminación, funcionando incluso con personas de distinta raza. Dicho algoritmo, denominado UASGM, aúna un método de clasificación no supervisada mediante "clustering" por distancia euclídea, con un modelo estocástico gaussiano. Aunque el empleo de modelos gaussianos en segmentación no es una aportación, sí lo es el estudio del espacio de color óptimo y de la metodología elegida para segmentar la piel de personas, concluyendo con la utilización de este tipo de modelo. Por otro lado, también es una aportación la utilización de un método de "clustering" por aprendizaje competitivo mediante el algoritmo VQ, para inicializar el modelo gaussiano, logrando de esta forma una adaptación del mismo al usuario.

Se ha demostrado que el método de "clustering" utilizado es una simplificación del de mezcla de múltiples gaussianas en el modelado de histogramas y de su resolución mediante el algoritmo EM (Expectation-Maximization). Dicha simplificación transforma este método, basado en funciones gaussianas, en el método VQ, basado en distancia euclídea, que utiliza aprendizaje local no supervisado y que es inicializado mediante un histograma aproximado. Mediante este método de "clustering" se obtienen las principales cromaticidades de la imagen, entre ellas se encuentra la correspondiente con el color de la piel y con estos datos se inicializa el modelo. Además, éste se adapta mediante una combinación lineal de los parámetros ya conocidos del mismo, siguiendo el criterio de máxima probabilidad. La obtención del número óptimo de clases en el "clustering" (problema de "validación de cluster") se resuelve aplicando una función de coste a las agrupaciones obtenidas para un número de clases entre dos y un valor máximo, fijando éste para el número de clases que den un máximo en la función de coste. La función de coste elegida es una modificación del ratio generalizado de Fisher.

Experimentalmente se ha demostrado que los resultados obtenidos, respecto al número de clases óptimo del proceso de "clustering", mejoran a los obtenidos mediante los métodos: FHV (Fuzzy Hypervolume), Evidence density y MDL (Minimum Description Length), e igualan los de: MML (Minimum Message Length) y GMM (Gaussian Mixture Modeling), reduciendo el tiempo de computación de todos ellos. Todos los métodos de la comparación están basados en modelos de múltiples gaussianas empleando el algoritmo EM. Asimismo, con el método propuesto se han obtenido mejores resultados de segmentación que aplicando el GLVQ-F (Fuzzy Generalized Learning Vector Quantization), que utiliza el conocido algoritmo de "clustering" FCM (Fuzzy C-Means).

Una vez segmentada la piel, se han utilizado las variaciones, en el tamaño y centro de gravedad, del objeto segmentado para detectar los distintos movimientos de cabeza del usuario. Se ha empleado un sistema de guiado a base de comandos discretos para mover la silla aplicando un criterio de robustez en la detección y de facilidad de manejo. Un sistema generador de comandos analiza las variaciones del objeto segmentado y determina el comando generado por el usuario. También se ha diseñado un módulo detector de guiños de ojos y de ocultamiento de labios para generar los comandos especiales ACTIVO/INACTIVO y ADELANTE/ATRÁS. A partir de los comandos y utilizando las restricciones mecánicas impuestas por la silla, un módulo de control a alto nivel genera las consignas de velocidades lineal y angular a mandar a la misma.

Para ayudar al usuario a adaptarse al sistema de guiado propuesto, se ha diseñado un entorno de

simulación en 3D, donde se reproduce la primera planta de la antigua Escuela Politécnica de la Universidad de Alcalá. Con este sistema se pueden simular los movimientos de la silla sin peligro alguno para el usuario.

Para demostrar las prestaciones del sistema propuesto, se han realizado pruebas sobre una de las sillas prototipo del proyecto SIAMO (que debió de ser convenientemente adaptada para permitir este tipo de guiado) con distintos usuarios y en entornos interiores. En esta fase se detectó que la acción de guiñar un ojo era poco robusta cuando la silla estaba en movimiento, por lo que fue sustituida por la acción de ocultar los labios. La diferenciación entre los comandos de SENTIDO y ACTIVACIÓN se realizó en función del tiempo. Finalmente se presentan los resultados de una encuesta realizada a los usuarios sobre la controlabilidad y rendimiento del sistema.

Como colofón del trabajo cabe también destacar que el algoritmo UASGM puede ser aplicado en otro tipo de proyectos que requieran segmentar un color en una imagen de una forma robusta y adaptativa. Así, fue utilizado por el autor en la detección de defectos sobre piezas de plástico, en un sistema de inspección industrial, obteniendo detecciones correctas en el 98% de los casos. Con esta técnica y trabajando en un espacio de color 2D se obtuvieron mejores resultados que aplicando el método de perfiles de color usando un espacio de color 3D [Duffy&Lacey,98].

7.2.- FUTURAS LÍNEAS DE TRABAJO

En esta tesis se ha puesto de manifiesto la viabilidad de un sistema de guiado mediante movimiento de cabeza, utilizando como sensor una cámara de vídeo y aplicando técnicas de visión artificial. Las líneas de futuro se dirigen en dos direcciones: por un lado, la consecución de un prototipo de silla más sofisticado, que pudiera llegar a ser comercializable, y por otro, la mejora de las técnicas de segmentación desarrolladas.

En cuanto a las mejoras o ampliación de estudios a realizar sobre el prototipo de silla diseñado se pueden considerar, entre otras, las siguientes propuestas:

- Introducción de un modelo 3D de la cabeza que permita calcular, de forma precisa, el ángulo formado por la misma. Mediante un modelo geométrico tridimensional de la forma de la cabeza de una persona y empleando una estimación del mismo, mediante un filtro de Kalman, se puede

robustecer la detección del ángulo de giro de la misma. Por lo que se podría aumentar el número de comandos generados en función de los movimientos de cabeza.

- Robustecer la máquina de estados generadora de comandos aplicando los modelos ocultos de Markov (HMM). Asimismo, se podría introducir un sistema borroso que permitiera elegir entre varias velocidades angulares finales para los giros de la silla, en función de los giros de cabeza, y entre varios valores de aceleración lineal en función del movimiento vertical de cabeza realizado.

- Diseño de un sistema de control en tiempo real proporcional a la dirección de la mirada. Para ello habría que aumentar la resolución de las imágenes, robustecer la detección de giros de cabeza, aplicando un modelo en 3D, y localizar la posición de los ojos. Componiendo las posiciones de la cabeza y de los ojos con respecto a ésta se calcularía la dirección real de la mirada. Utilizando este dato, se aplicaría un modelo fuzzy que proporcionara la velocidad lineal y angular de la silla. De esta forma, se tendría un mayor control de los movimientos de la silla aunque habría que comprobar si este sistema lograba mejorar al propuesto al aparecer otros problemas como la respuesta de la silla a movimientos involuntarios del usuario y a que cuando una persona mira hacia una dirección no sabe exactamente el ángulo concreto de su mirada.

- Aplicación del sistema en exteriores. En esta tesis ha quedado de manifiesto que para un funcionamiento óptimo del sistema se necesita una iluminación uniforme. Dicha limitación es consecuencia del empleo de técnicas de color. En entornos exteriores no se puede garantizar esta premisa, por lo que el sistema presentado puede fallar. Para robustecerlo se propone utilizar un vector de características a analizar más complejo, que aparte del color, considere otros aspectos como: análisis de bordes, de luminancia, de flujo óptico, etc.

- Implementación en "hardware" específico (DSPs) del algoritmo diseñado y empleo de un sistema operativo en tiempo real. De esta forma se conseguirá una reducción del tiempo de proceso de los algoritmos y una programación más estructurada de los mismos mediante eventos.

En cuanto a las mejoras del método de segmentación, se hacen las siguientes propuestas:

- Generalización del método que permita segmentar los N colores de una imagen, mediante N

funciones gaussianas de forma adaptativa y en tiempo real. De esta forma se podría segmentar cualquier color de la imagen de forma óptima y reduciendo la información del histograma al trabajar con modelos matemáticos.

- Implementación del UAGSM empleando un espacio de color tridimensional

Por otro lado, la estructura montada en la silla coincide con la utilizada en las teleconferencias (una cámara enfrente del usuario), con lo que todos los algoritmos aplicados pueden ser directamente trasladables a aplicaciones de este tipo, tan en boga en la actualidad, pudiendo mover un objeto a distancia, reducir la cantidad de información a mandar por la red, al localizar la cara del usuario que es el objeto de interés, o bien actuar con la dirección de la mirada del usuario.

A.1 INVARIANZAS DEL MODELO DE LA PIEL

En este anexo se hace un estudio de la invarianza del modelo de la piel, empleando diferentes pruebas, y ante los siguientes parámetros: diferentes usuarios, traslaciones y giros, zoom y cambios de iluminación.

Distintos usuarios

Se han tomado 20 imágenes de prueba de raza blanca y 10 de raza negra y amarilla, se han calculado los estadísticos de las funciones normales que mejor aproximan las distribuciones de colores de la piel y se han obtenido las diferencias entre los valores de prueba y los patrones correspondientes para cada raza. Los resultados se muestran en la figuras A.1.1, A.1.2 y A.1.3, y de ellos se extraen las siguientes conclusiones: 1) las diferencias respecto a los patrones son iguales para las tres razas. 2) las medias de r y g son muy constantes para todas las imágenes, con unas diferencias máximas respecto a m_r de ±0,02 y la mitad respecto a m_g; las varianzas tienen una menor variación, con un máximo de $\pm 0{,}02 * 10^{-2}$, lo que supone un rango de variación 100 veces menor que las medias.

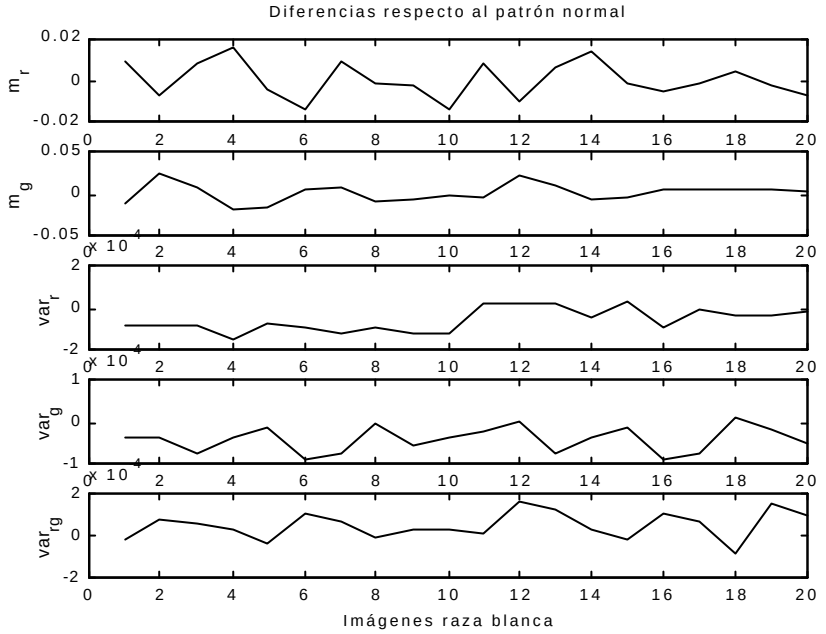


Figura A.1.1. Diferencias de estadísticos para la raza blanca

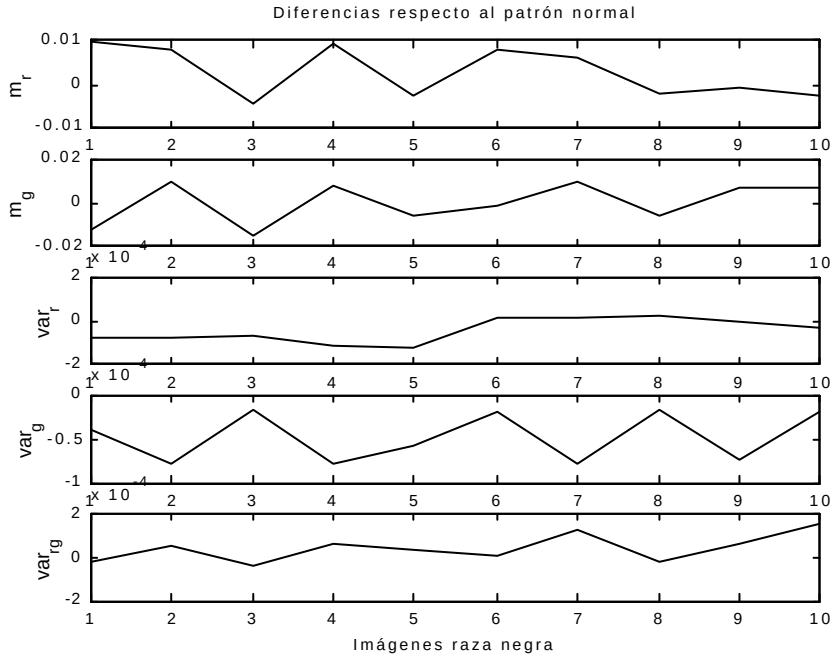


Figura A.1.2. Diferencias de estadísticos para la raza negra

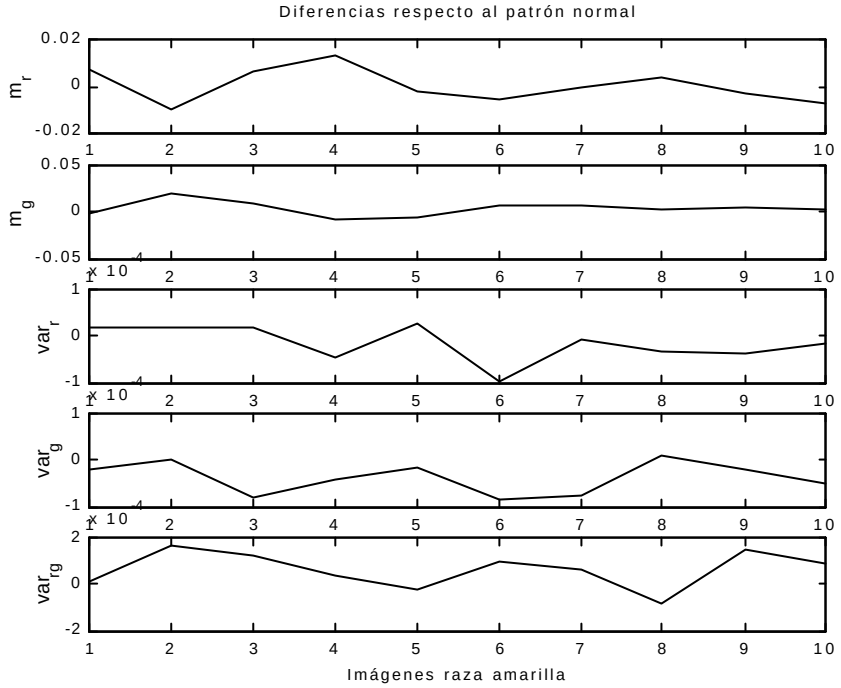


Figura A.1.3. Diferencias de estadísticos para la raza amarilla

Demostrado que las diferencias dentro de una misma raza están muy acotadas, se va a analizar la diferencia entre los modelos patrones de las distintas razas. Para poder apreciarlas mejor se van a proyectar las funciones bidimensionales sobre los ejes r y g, según se muestra en la figura A.1.4. En la Tabla A.1.1. se muestran las diferencias entre los estadísticos de forma numérica.

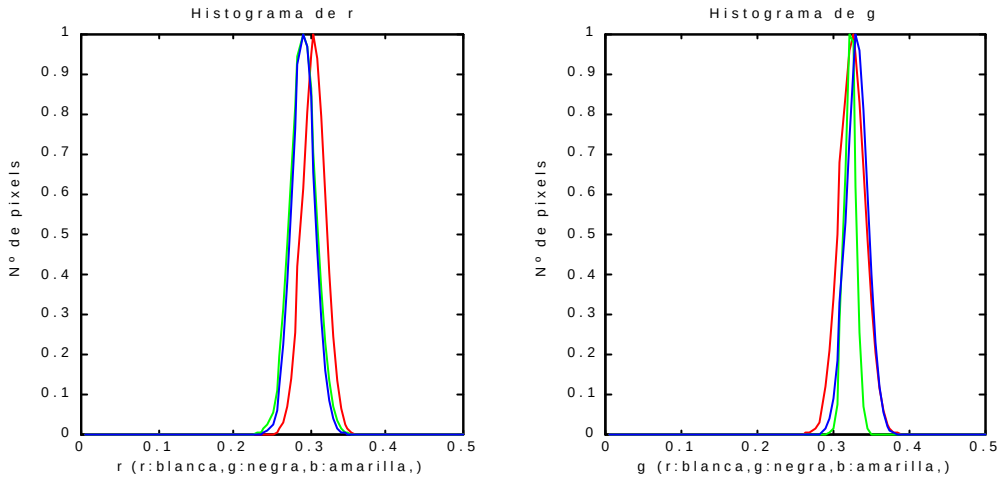


Figura A.1.4. Diferencias entre los modelos de distintas razas

La conclusión que se extrae es que las diferencias entre las distintas razas en este espacio de color son muy reducidas.

Diferencias	m_r	m_g	F_r^2	F_g^2	$F_{rg}^2 = F_{gr}^2$
blanca-negra	0,140	0,025	$-0{,}0644 * 10^{-3}$	$0{,}2280 * 10^{-3}$	$-0{,}0449 * 10^{-3}$
blanca-amarilla	0,137	-0,061	$-0{,}0043 * 10^{-3}$	$0{,}0872 * 10^{-3}$	$-0{,}1010 * 10^{-3}$
negra-amarilla	-0,003	-0,0086	$0{,}0601 * 10^{-3}$	$-0{,}1408 * 10^{-3}$	$-0{,}0561 * 10^{-3}$

Tabla A.1.1. Diferencias entre los estadísticos de las distintas razas

Giros y traslaciones

Se han tomado las cuatro imágenes de test que representan situaciones extremas, en cuanto a giros y traslaciones en nuestro sistema (ver figura A.1.5.), y se han evaluado las diferencias entre los estadísticos de cada una de ellas respecto al patrón de raza blanca, mostrando los resultados en la figura A.1.6.

Imagen 1 Imagen 2 Imagen 3 Imagen 4

Figura A.1.5. Imágenes de test para evaluar la invarianza al giro y traslaciones

En la figura A.1.7 se muestra una comparación entre el modelo patrón y las distribuciones de color de las imágenes de test para los ejes r y g.

La conclusión que se extrae es que los estadísticos varían muy poco respecto al patrón y entre ellos cuando se realizan giros y traslaciones, manteniendo las distribuciones una forma gaussiana.

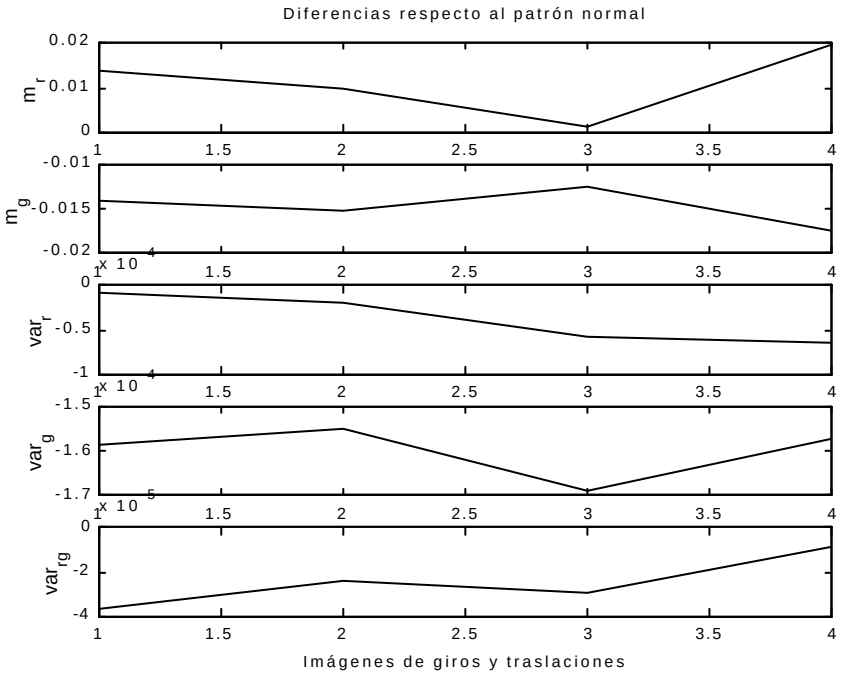


Figura A.1.6. Diferencias de los estadísticos respecto al patrón (giros y traslaciones)

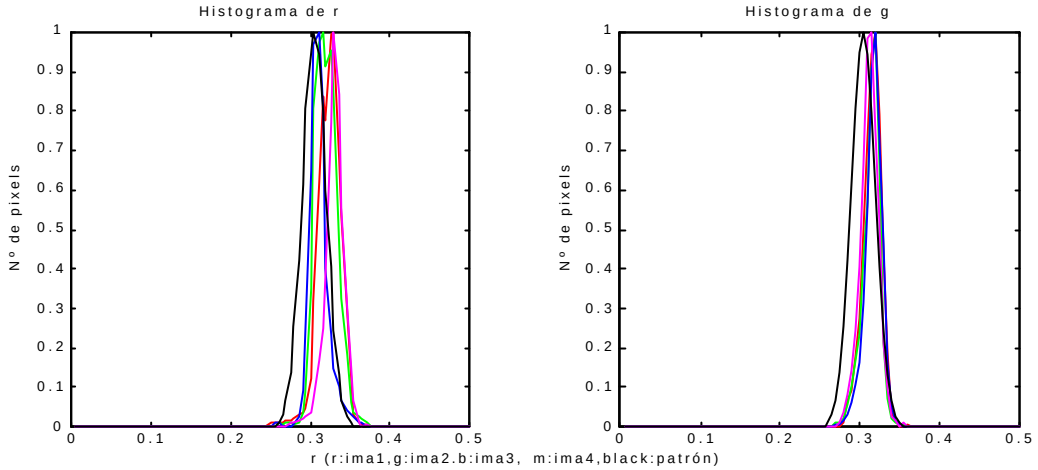


Figura A.1.7. Diferencias entre el modelo patrón y las distribuciones de giros y traslaciones

Zoom

Realizando el mismo estudio que en el caso anterior pero evaluando ahora el margen máximo de zoom permitido en nuestro sistema, según se observa en las imágenes de la figura A.1.8, se obtienen los resultados que se muestran en las figuras A.1.9 y A.1.10.

Figura A.1.8. Imágenes de test para evaluar la invarianza al zoom

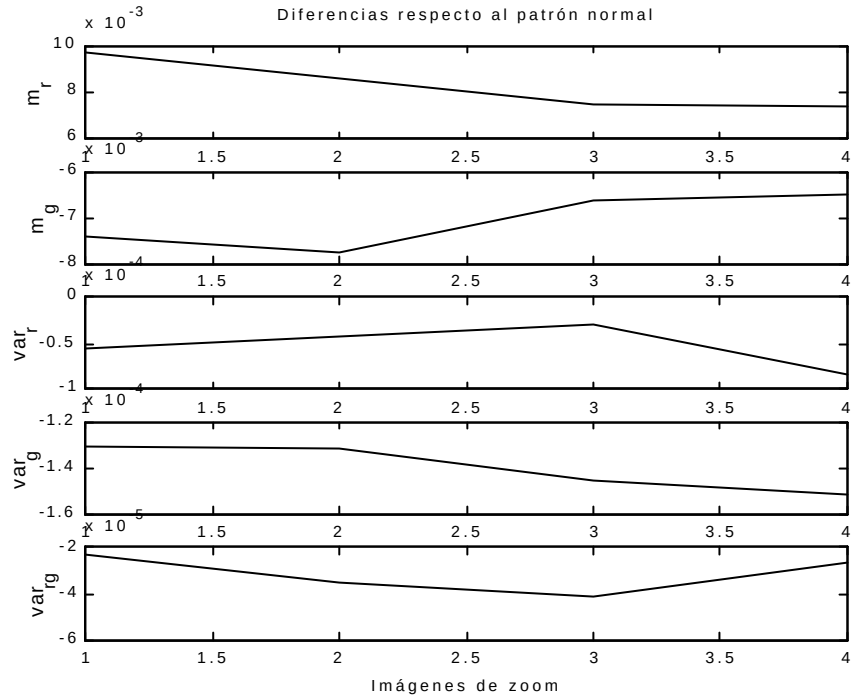


Figura A.1.9. Diferencias de los estadísticos respecto al patrón (zoom)

En este caso, se concluye que las diferencias de estadísticos con el patrón son muy pequeñas, menores que para distintos usuarios y giros y traslaciones, además los valores para las distintas imágenes son prácticamente iguales, debido a que en este caso la imagen es estática y por lo tanto las condiciones de luz no cambian entre ellas.

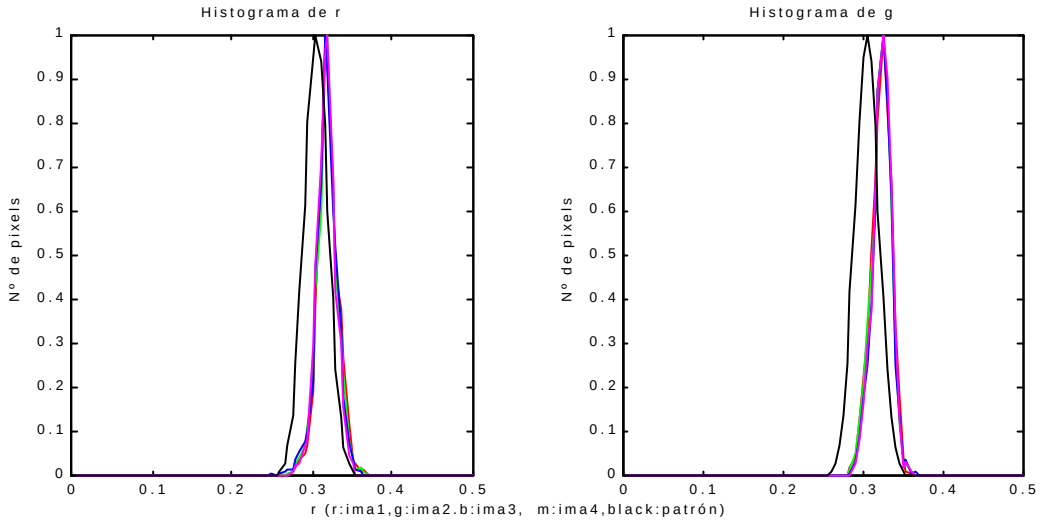


Figura A.1.10. Diferencias entre el modelo patrón y las distribuciones de zoom

Cambios de iluminación

Para evaluar la invarianza a los cambios de iluminación se ha utilizado el mismo método que en los casos anteriores. En la figura A.1.11 se muestran las imágenes de prueba, la imagen 1 se ha tomado con iluminación natural, de la 2 a la 5 se han obtenido mediante iluminación artificial aumentando progresivamente la intensidad luminosa. Los resultados se muestran en las figuras A.1.12 y A.1.13.

Como se puede observar, la forma de las distribuciones se mantiene al variar la iluminación produciéndose un desplazamiento de las mismas. Así a medida que aumenta la luminancia la componente r aumenta y la componente g disminuye. Por otro lado, se dan las mayores diferencias con el patrón de todas las invarianzas analizadas, siendo éstas de un máximo de 0,05 para las medias y 2 $*10^{-4}$ para las varianzas

.

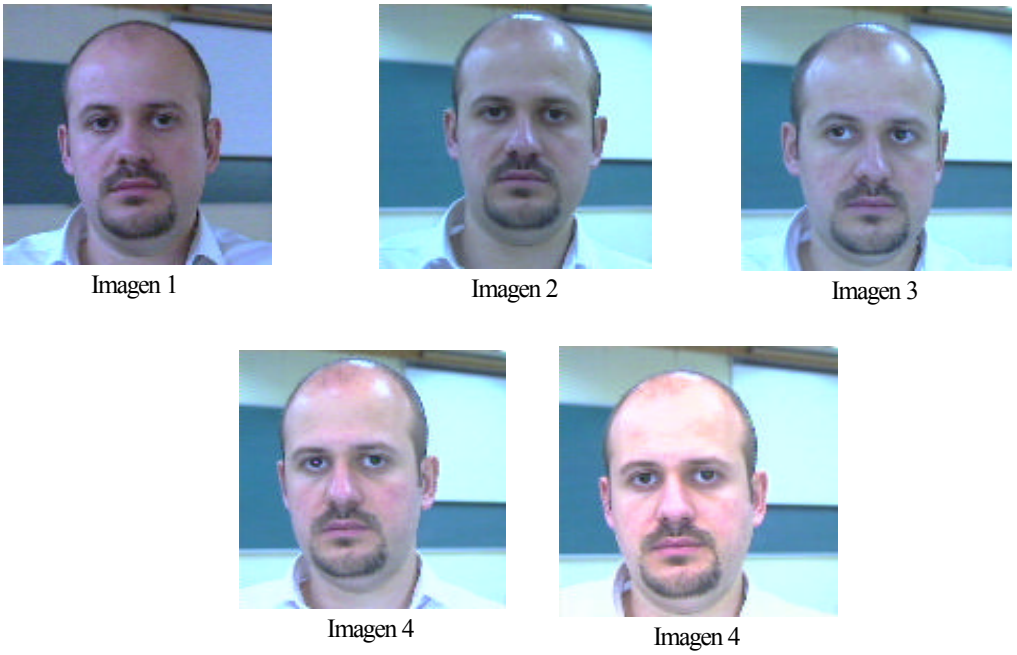


Figura A.1.11. Imágenes de test para evaluar la invarianza a cambios de iluminación

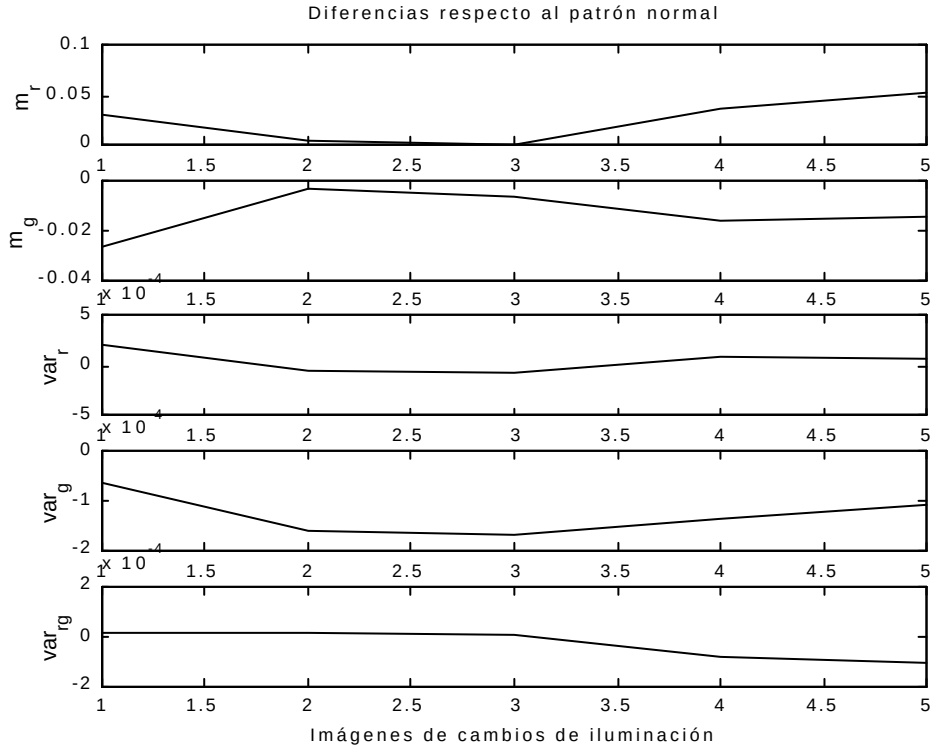


Figura A.1.12. Diferencias de los estadísticos respecto al patrón (iluminación)

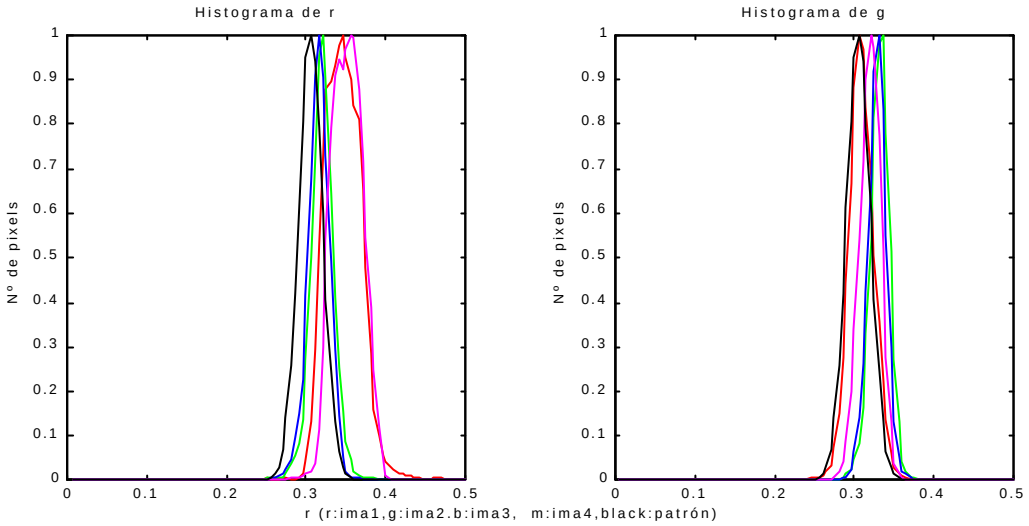


Figura A.1.13. Diferencias entre el modelo patrón y el histograma ante cambios de iluminación

A.2 DEMOSTRACIONES DE LAS ECUACIONES DEL MÉTODO GLVQ-F

En este anexo se demuestran las ecuaciones (4.83), (4.86) y (4.87)

Demostración de la ecuación (4.83):

$$\nabla_{\hat{m}_k} L = \nabla_{\hat{m}_k}\left(\sum_{r=1}^{K} u_r ||x_c - \hat{m}_r||^2\right) = \nabla_{\hat{m}_k}\left(\sum_r \frac{||x_c - \hat{m}_r||^2}{\sum_{j=1}^{K}\left(\frac{||x_c - \hat{m}_k||^{\frac{2}{n-1}}}{||x_c - \hat{m}_j||^{\frac{2}{n-1}}}\right)}\right) = \nabla_{\hat{m}_k}\left(\sum_r \frac{\left(||x_c - \hat{m}_r||^2\right)^{(n-2)/(n-1)}}{\sum_j \left(||x_c - \hat{m}_j||^2\right)^{-1/(n-1)}}\right)$$

$$= \nabla_{\hat{m}_k}\left(\frac{\sum_r \left(||x_c - \hat{m}_r||^2\right)^{(n-2)/(n-1)}}{\sum_j \left(||x_c - \hat{m}_j||^2\right)^{-1/(n-1)}}\right) = \left[\frac{\left(\frac{n-2}{n-1}\right)\left(||x_c - \hat{m}_k||^2\right)^{-1/(m-1)}(-2)(x_c - \hat{m}_k)\sum_j \left(||x_c - \hat{m}_j||^2\right)^{-1/(n-1)}}{\left(\sum_j \left(||x_k - \hat{m}_j||^2\right)^{-1/(n-1)}\right)^2}\right.$$

$$\left. - \frac{\sum_r \left(||x_c - \hat{m}_k||^2\right)^{(m-2)/(m-1)}\left(\frac{-1}{n-1}\right)\left(||x_c - \hat{m}_k||^2\right)^{-m/(m-1)}(-2)(x_c - \hat{m}_k)}{\left(\sum_j \left(||x_k - \hat{m}_j||^2\right)^{-1/(n-1)}\right)^2}\right]$$

$$= \left(\frac{-2}{n-1}\right)(x_c - \hat{m}_k)\left[\frac{(n-2)\left(||x_c - \hat{m}_k||^{-2/(n-1)}\right)}{\sum_j \left(||x_c - \hat{m}_j||^{-2/(n-1)}\right)}\right.$$

$$\left. + \frac{\sum_r \left(||x_c - \hat{m}_r||^{2/(n-1)}\right)^{n-2}\left(||x_c - \hat{m}_r||^{2/(n-1)}\right)^{2-n}\left(||x_c - \hat{m}_r||^{-2/(n-1)}\right)^2}{\left(\sum_j \left(||x_k - \hat{m}_j||^2\right)^{-1/(n-1)}\right)^2}\right] \qquad \text{(A2.1)}$$

$$= \left(\frac{-2}{n-1}\right)(x_k - \hat{m}_k)\left[(n-2)u_k + \left(\sum_{r=1}^{K}\left(\frac{||x_k - \hat{m}_k||^{\frac{2}{n-1}}}{||x_k - \hat{m}_r||^{\frac{2}{n-1}}}\right)^{2-n}\right)u_k^2\right]$$

Demostración de la ecuación (4.86):

Reescribiendo la ecuación (4.84) queda:

$$\nabla_{\hat{m}_k} L = \left(\frac{-2}{n-1}\right)(x_c - \hat{m}_k)\left[(n-2)u_k + \left(\sum_{r=1}^{K}\left(\frac{||x_c - \hat{m}_k||^{\frac{2}{n-1}}}{||x_c - \hat{m}_r||^{\frac{2}{n-1}}}\right)^{2-n}\right)u_k^2\right]$$

$$= (-2)(x - \hat{m}_k)\left[u_k + \frac{\left(\sum_{r=1}^{K}\left(\frac{||x_c - \hat{m}_k||}{||x_c - \hat{m}_r||}\right)^{(4-2n)/(n-1)} - \frac{1}{u_k}\right)u_k^2}{m-1}\right] \qquad \text{(A2.2)}$$

$$= (-2)(x_c - \hat{m}_k)\left[u_k + F(k,m)\right]$$

Evaluar los límites de la ecuación anterior para n64 y n61 es complicado por el hecho de que u_k depende de n. Para simplificar el problema se va a asumir que $||x_c - \hat{m}_r|| > 0$. Se sabe que:

$$\lim_{n\to\infty}\{u_k\} = \frac{1}{K} \qquad \text{(A2.3)}$$

para cualquier k. Por lo que el numerador de F(k,n) puede aproximarse por:

$$\left(\sum_r \left(\frac{||x_c - \hat{m}_k||}{||x_c - \hat{m}_r||}\right)^{-2} - K\right)\left(\frac{1}{K}\right)^2 \qquad \text{(A2.4)}$$

como n64, entonces (n-1)64, por lo que:

$$\lim_{n\to\infty}\{F(k,m)\} = 0 \qquad \text{(A2.5)}$$

combinando este resultado con el de (A2.3), se tiene:

$$\lim_{n\to\infty}\{\hat{m}_j(t)\} = \hat{m}_j(t-1) + \left(\frac{2\alpha(t-1)}{K}\right)(x_c - \hat{m}_j(t-1)) \qquad j = 1,2,...,K \qquad \text{(A2.6)}$$

Demostración de la ecuación (4.87):

A partir de la ecuación A2.2, se hace:

$$\delta_{k,r} = \left(\frac{||x_c - \hat{m}_k||}{||x_c - \hat{m}_r||}\right)^2 \qquad y \qquad \Delta_k = \max\{\delta_{k,1},...,\delta_{k,r}\} \qquad \text{(A2.7)}$$

Obsérvese que) $_k$ $1. Por lo que F(k,m) se puede escribir como:

$$F(k,n) = \frac{\left(\dfrac{\sum_r \delta_{k,r}^{-1}\delta_{k,r}^{1/(n-1)}}{\left(\sum_r \delta_{k,r}^{1/(n-1)}\right)^2}\right) - \left(\dfrac{1}{\sum_r \delta_{k,r}^{1/(n-1)}}\right)}{n-1} = \frac{\left(\dfrac{1}{n-1}\right)\left(\sum_r (\delta_{k,r}^{-1} - 1)\delta_{k,r}^{1/(n-1)}\right)}{\left(\sum_r \delta_{k,r}^{1/(n-1)}\right)\left(\sum_r \delta_{k,r}^{1/(n-1)}\right)} \qquad \text{(A2.8)}$$

Los límites de la ecuación anterior son cero tanto para la neurona ganadora como la no ganadora. Si) $_k$ =1, entonces el denominador de (A2.8) será igual a 1 ya que n61, mientras que la suma del numerador tiene términos distintos de cero solamente para * $_{k,r}$ < 1. Aplicando la regla de L'Hopital en este caso, se deduce que el numerador tiende a cero.

Por otra parte si $)_k > 1$ es facil demostrar que:

$$\frac{\left(\frac{1}{n-1}\right)}{\left(\sum_r \delta_{k,r}^{1/(n-1)}\right)} \to 0$$

$$\frac{\left(\sum_r (\delta_{k,r}^{-1} - 1)\delta_{k,r}^{1/(n-1)}\right)}{\left(\sum_r \delta_{k,r}^{1/(n-1)}\right)} \to \Delta_k - 1 \qquad \text{(A2.9)}$$

En ambos casos se tiene:

$$\lim_{n \to l}\{F(k,m)\} = 0 \qquad \text{(A2.10)}$$

Además es sabido que:

$$\lim_{n \to l}\{u(k)\} = 1 \qquad \text{(A2.11)}$$

para la neurona ganadora y 0 para el resto, con lo que únicamente se actualizarán las neuronas ganadoras, quedando:

$$\lim_{m \to l}\{\hat{m}_j(t)\} = \begin{cases} \hat{m}_j(t-1) + (2\alpha(t-1))(x_c - \hat{m}_j(t-1)) & \text{si} \quad i = \arg\min_r\{||x_c - \hat{m}_r||\} \\ 0 & i \neq j \end{cases} \qquad \text{(A2.12)}$$

A.3 DETECCIÓN DE OJOS MEDIANTE PLANTILLAS DEFORMABLES

En este anexo se presenta un estudio realizado por el autor sobre la aplicación de las plantillas deformables a la detección de ojos. Los resultados obtenidos en cuanto a tiempo de proceso nos obligaron a abandonar esta línea de investigación. El método presenta ciertas modificaciones respecto al original propuesto por Yuille[Yuille et al., 92]. Por un lado, la localización inicial de la plantilla se realiza de forma automática empleando un análisis de características geométricas de los objetos localizados en la imagen de valles umbralizada. Por otra parte, las imágenes transformadas de valles y picos, obtenidas mediante operadores morfológicos, se calculan aplicando un elemento estructurante de tamaño 3x3 que se repite un número n de veces, lo que supone un ahorro de tiempo de proceso. La imagen de bordes es obtenida mediante un filtro espacial y no mediante operadores morfológicos, lo que también logra un ahorro de tiempo de proceso. El cálculo de la energía de picos se calcula evaluando la imagen de picos sobre el área definida por el blanco de los ojos. Este método es más robusto que el propuesto por Yuille, que evalúa la energía sobre las líneas que unen el centro del iris con los centros de la esclerótica. Los valores iniciales de los parámetros de la plantilla no son fijos sino que son extraídos de la localización inicial. Por último, el número de etapas necesarias para el ajuste de los parámetros se ha reducido respecto al método estándar, asimismo se ha independizado el ajuste del iris de la esclerótica (blanco del ojo), lo que mejora considerablemente los tiempos de proceso. Este hecho es fundamental en el sistema ya que la finalidad última del mismo es la de realizar

seguimiento y no identificación.

A3.1. Preprocesamiento.

En esta fase se realizan las transformaciones necesarias sobre la imagen original en niveles de gris (I(x,y)) para destacar ciertas características de interés de la imagen. Éstas son:

- Imagen de picos (N_p (x,y)). Resalta partes claras de la imagen entre partes oscuras.
- Imagen de valles (N_v (x,y)). Resalta partes oscuras de la imagen entre partes claras.
- Imagen de bordes (N_b (x,y)). Resalta los flancos o bordes de la imagen.

Para la obtención de las imágenes transformadas de picos y valles se utilizaron los operadores morfológicos open y close [Serra, 82] con un elemento estructurante cuadrado de tamaño 3x3 fijo. A partir de éste se puede obtener un elemento estructurante de mayor tamaño iterándolo sobre la imagen un número n de veces según la siguiente relación: tamaño=1+2 n. De esta forma se logra una programación más flexible y un ahorro en tiempo de ejecución. En la tabla A3.1 se muestran las operaciones usadas para obtener las imágenes transformadas. El número entre paréntesis que hay al lado de cada operador morfológico indica el número de iteraciones a realizar con el elemento estructurante de 3x3.

La ventana de imagen a analizar tiene un tamaño de 320x240 pixels y se corresponde con el cuarto superior izquierdo de una imagen facial capturada por la cámara. Por lo tanto, habrá que imponer como condición previa que exista un ojo dentro de esta zona.

El método inicialmente hace una localización a "grosso modo", basada en el estudio de los objetos segmentados obtenidos sobre la imagen de valles umbralizada con un nivel de umbral (U_v) calculado automáticamente. Para calcular el umbral U_v se analiza el histograma de la imagen de valles en la ventana de estudio, obteniendo un resultado como el que se aprecia en la figura A3.1.

Los pixels de interés (pixels de valle) serán aquellos que tienen un nivel de gris elevado.Para obtenerlos se fija el umbral U_v en el nivel para el cual el módulo de la pendiente del histograma sea inferior a una constante K_v a la que experimentalmente se le ha dado el valor 5, de forma que la umbralización se hará como se indica en la ecuación (A3.1).

$$
\begin{aligned}
& * \frac{dH(p)}{dp} * < K_v \quad 6 \quad p \,'\, U_v \\
& si \quad N_v(x,y) < U_v \quad 6 \quad N_{vu}(x,y) \,'\, 0 \\
& si \quad N_v(x,y) \$ U_v \quad 6 \quad N_{vu}(x,y) \,'\, 255
\end{aligned}
\tag{A3.1}
$$

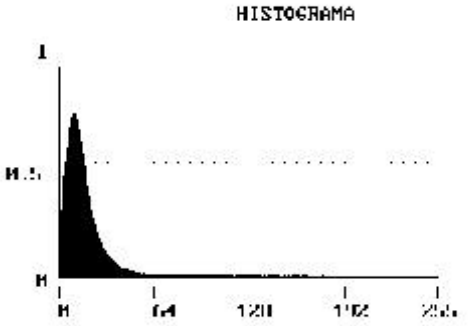


Figura A3.1. Histograma de la imagen de valles

De esta forma los pixels de N_v con valor 255 indicarán pixels de valle, es decir que son pixels oscuros de la imagen original en entornos claros, como se puede apreciar en la figura A3.2(d).

Para el cálculo de la imagen de bordes se aplica directamente un filtro de Sobel, basado en el gradiente. El hecho de usar un filtro espacial en vez de operadores morfológicos supone un ahorro de tiempo de proceso.

Imagen Transformada	**Operación Morfológica**	**Otras operaciones**
N_v (x,y)	close(13)- I(x,y)	
N_p(x,y)	I(x,y)- open(12)	
N_b (x,y)		filtro espacial (3x3)
N_{vu} (x,y)		N_v (x,y) umbralizada

Tabla.A3.1. Operaciones para obtener las imágenes transformadas

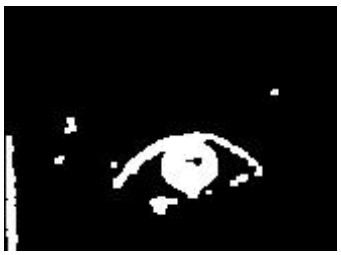
En la figura.A3.2 se pueden apreciar la imagen original, la ventana a analizar así como las imágenes

transformadas obtenidas.

(a) Imagen original

(b) Zona de análisis

(c) Imagen de valles



(d) Valles umbralizada

(e) Imagen de picos

(f) Imagen de bordes

Figura.A3.2. Imágenes transformadas

A3.2. Localización inicial de la plantilla

El objetivo consiste en encontrar el iris dentro de los objetos segmentados en la imagen de valles umbralizada ($N_{vu}(x,y)$) . Para ello se realiza un "open" y un "close" a nivel binario sobre la misma, lo que elimina pequeños objetos y rellena pequeños huecos que existan en los objetos, más tarde se eliminan todos aquellos objetos cuya área sea mayor o menor a unos márgenes previamente establecidos y que fueron calculados experimentalmente en función de la base de datos facial, como se indica en la ecuación (A3.2).

$$\begin{gathered} A_{\min} \# Area\ objeto \# A_{\max} \\ A_{\min} ' 400\ pixels\ y\ A_{\max} ' 10000\ pixels \end{gathered} \qquad \text{(A3.2)}$$

A continuación se calculan los diámetros máximos y mínimos de los objetos segmentados y los ángulos para los que se dan dichos diámetros y se eligen aquellos que se encuentren dentro de los siguientes márgenes:

$$\begin{gathered} 2_{\min} \# (angulo\ diam_{\max} \& angulo\ diam_{\min}) \# 2_{máx} \\ 2_{mín} ' 75º\ y\ 2_{máx} ' 115º \end{gathered} \qquad \text{(A3.3)}$$

Al igual que en el caso anterior los márgenes fueron tomados de forma experimental, analizando la

$$2.15\ diam_{mín} \# diam_{\max} \# 3\ diam_{mín} \tag{A3.4}$$

forma de los objetos IRIS en la base de datos. Dichos objetos tienen una forma similar a la mostrada en la figura A3.3. Como se observa, el iris queda segmentado junto con las pestañas ya que son objetos oscuros que están en contacto (vease figura A3.2(d)), el ángulo que forman los diámetros máximo y mínimo está en torno a los 90º y el diámetro máximo es aproximadamente 2.5 veces mayor que el mínimo.

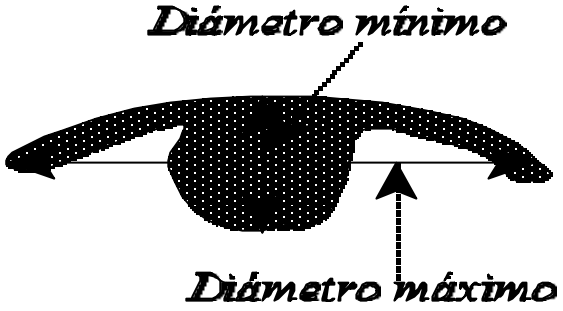


Figura A3.3. Iris segmentado

De entre los objetos que queden se calculan las características geométricas rugosidad (R) y circularidad (C) según las ecuaciones (A3.5) y (A3.6). Se observó que los objetos IRIS se caracterizaban por poseer unos valores de rugosidad y circularidad mínimos, por lo que para segmentarlos se usó una función de coste (f) consistente en la suma de estos dos parámetros, de forma que el iris se correspondía con aquel objeto para el que su función de coste diera un mínimo, como se ve en la ecuación (A3.7).

$$C = \frac{(perímetro\ \ del\ \ objeto)^2}{4\pi\ \ Area\ \ del\ \ objeto} \tag{A3.5}$$

$$R = \frac{perímetro\ \ del\ \ objeto}{perímetro\ \ del\ \ casco\ \ convexo\ \ del\ \ objeto} \tag{A3.6}$$

$$IRIS = objeto_i\ \ tal\ \ que:\ \ f(objeto_i) = mínimo \qquad f(objeto_i) = R_i \% C_i \tag{A3.7}$$

Una vez localizado el objeto IRIS se calcula el diámetro medio de éste tomando como referencia cuatro diámetros obtenidos para los ángulos K 45º respecto a la horizontal, siendo K=0,1,2,3. La plantilla se localizará inicialmente en el centro de gravedad del objeto (x_0 , y_0) y se tomará como radio inicial:

$$Radio\ \ inicial = \frac{Diámetro\ \ medio_{IRIS}}{2} \qquad \text{(A3.8)}$$

A3.3. La plantilla del ojo

La plantilla consta de una colección de curvas parametrizadas que describen la forma del ojo. Está formada por 9 parámetros g=(x_e , x_c, r, a, b, c, 2, p_1, p_2) cada uno de los cuales se puede variar en la etapa de ajuste.

Antes de expresar matemáticamente las curvas se van a definir dos vectores unitarios que darán la orientación del ojo:

$$\overline{e_1} = (\cos 2, sen 2) \qquad \text{(A3.9)}$$

$$\overline{e_2} = (-sen 2, \cos 2) \qquad \text{(A3.10)}$$

Un punto **x** en el espacio vendrá representado por (x_1, x_2) donde:

$$\overline{x} = x_1\overline{e_1} + x_2\overline{e_2} \qquad \text{(A3.11)}$$

La plantilla quedará definida por los siguientes elementos:

1. Una circunferencia de radio r centrada en $\mathbf{x_c}(x_{1c}, x_{2c})$ que se corresponde con el límite entre el iris y el blanco del ojo.

$$(x_1 - x_{1c})^2 + (x_2 - x_{2c})^2 = r^2 \qquad \text{(A3.12)}$$

2. Dos parábolas con centro en $\mathbf{x_e}(x_{1e}, x_{2e})$, de ancho igual a "2b", que se corresponden con los bordes de los párpados superior e inferior. La altura del párpado superior es "a" y la del párpado inferior es "c". El ángulo de orientación es 2.

$$\overline{x} = \overline{x_e} + x_1\overline{e_1} + (a - \frac{a}{b^2}x_1^2)\overline{e_2} \qquad |x_1| \le b \tag{A3.13}$$

$$\overline{x} = \overline{x_e} + x_1\overline{e_1} - (c + \frac{c}{b^2}x_1^2)\overline{e_2} \qquad |x_1| \le b \tag{A3.14}$$

3. Dos parámetros (p_1, p_2) que determinan las posiciones de los extremos de los blancos del ojo

$$\begin{aligned} \overline{x_e} + p_1\overline{e_1} \qquad & p_1 \ge 0 \\ \overline{x_e} + p_2\overline{e_2} \qquad & p_2 \le 0 \end{aligned} \tag{A3.15}$$

4.- La región entre el contorno y el iris se corresponde con el blanco del ojo.

La plantilla es parametrizada como se muestra en la Figura A3.4 y puede asumir cualquier orientación.

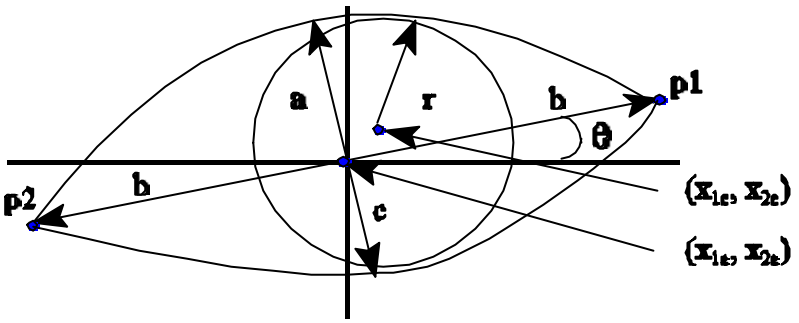


Figura A3.4. Plantilla parametrizada del ojo

Los valores que toman inicialmente los parámetros de la plantilla se toman de la localización inicial según los siguientes criterios:

1. x_e y x_c son el mismo punto.
2.- x_c : centro de gravedad del objeto IRIS (x_0)
3.- r : radio medio del objeto IRIS (r_0)
4.- El ancho 2b del ojo es igual al diámetro máximo del IRIS (b_0)

5.- 2 es el ángulo del diámetro máximo (2_0)

6.- a= r_0 y c=2/3 r_0

7.- p_1 y p_2 extremos del diámetro máximo calculado en la localización inicial (p_{10},p_{20})

A3.4. Función de energía para la plantilla del ojo

El ajuste de la plantilla con la imagen es un proceso dinámico. La plantilla se ajusta en función de las imágenes transformadas tomando como referencia una función de energía, de manera que cuanto mayor sea el ajuste de la plantilla a la geometría del ojo de la imagen a analizar menor será la función de energía.

La energía es función de todos los parámetros de la plantilla y se define en términos de valles, picos, bordes y energía interna:

$$E = E_v + E_p + E_b + E_{inte} \tag{A3.16}$$

La energía de valles viene dada por la integral extendida al área del iris (definida en la plantilla), sobre la imagen de valles.

$$E_v = \frac{c_1}{\text{área iris}} \iint_{\text{área iris}} N_v(\bar{x})\, dA \tag{A3.17}$$

La energía de bordes viene dada por la integral extendida al borde del iris y de la parábola, sobre la imagen de bordes.

$$E_b = \frac{c_2}{\text{borde iris}} \int_{\text{borde iris}} N_e(\bar{x})\, ds + \frac{c_3}{\text{borde parábolas}} \int_{\text{borde parábolas}} N_e(\bar{x})\, ds \tag{A3.18}$$

La energía de picos es la integral extendida al área del blanco de los ojos evaluada sobre la imagen de picos.

$$E_p = \frac{c_4}{\text{área blanco}} \left(\iint_{\text{área blanco}} N_p(\bar{x})\, dA \right) \tag{A3.19}$$

La energía interna será:

$$E_{inte} = \frac{k_1}{2}(\overline{x_e} - \overline{x_c})^2 + \frac{k_2}{2}(b - 2r)^2 + \frac{k_3}{2}((b - 2a)^2 + (a - 1.5c)^2) \qquad \text{(A3.20)}$$

A3.5. Ajuste de la plantilla del ojo

El ajuste de la plantilla se logra minimizando la función de energía. Este ajuste se realiza en cuatro etapas en función de los valores que tomen los coeficientes $\{c_i\}$ y $\{k_i\}$, como se indica en la tabla A3.2. Los valores que toman los coeficientes han sido calculados experimentalmente y dependen de la forma de generación de las imágenes transformadas. Mediante la energía de valles se realiza el ajuste de la posición de la plantilla desde la dada por el método de localización inicial. Mediante la imagen de picos se gira la plantilla hasta que la orientación sea óptima y por último la energía de bordes adapta el borde exterior del iris y los bordes del blanco del ojo. La energía interna prevee al sistema de posibles cambios bruscos en la geometría de la plantilla, permitiendo únicamente variaciones razonables de la misma (este término incorpora un conocimiento a priori de la forma del ojo). Obsérvese como cuando la forma se aleja de la estándar establecida a priori, la energía interna crece, introduciendo valores positivos y grandes a la energía total del sistema, provocando que el ajuste de parámetros se realice en sentido contrario y por lo tanto la forma se vuelve a acercar a la estándar. Del análisis de la base de datos facial se deduce las siguientes relaciones estándar:

$$b = 2r \quad ; \quad b = 2a \quad ; \quad a = 1.5c \qquad \text{(A3.21)}$$

Etapa	c_1	c_2	c_3	c_4	k_1	k_2	k_3
1	1	0	0	0	0	0	0
2	1	1	0	0	0	0	0
3	0	0	0	1	1	5	0
4	0	0	1	0	0	0	1

Tabla A3.2. Etapas para el ajuste de energía

Los componentes que intervienen en la expresión de la energía se pueden poner en función de sus parámetros, de forma que la minimización de la energía habrá que realizarla respecto a los diferentes parámetros que la componen. Para ello se utilizará la técnica del descenso por el gradiente, es decir,

el valor que toma el parámetro en una determinada iteración será igual al valor de la iteración anterior más un incremento, función de la variación temporal del parámetro (f(p)), multiplicado por una constante. Este incremento se realizará en sentido contrario a la derivada de la energía con respecto al parámetro, tal y como se puede ver en las siguientes ecuaciones:

$$p(n) = p(n-1) - K_p f(p) \quad ; \quad 0<K_p<1 \tag{A3.22}$$

$$f(p) = \frac{dp}{dt} = -\frac{\partial E}{\partial p} \tag{A3.23}$$

La convergencia se conseguirá cuando:

$$|p(n)-p(n-1)|<\varepsilon_p \tag{A3.24}$$

El valor de ε_p será diferente dependiendo del parámetro de que se trate, como se indica en la tabla A3.3. Para evitar mínimos locales, se considerará que el sistema ha convergido cuando R muestras consecutivas den un error menor que el margen establecido. Estos valores se han calculado experimentalmente. En la práctica existe también un control del número de iteraciones máximo, de forma que si se superan las 100, se considerará que el sistema ha convergido.

Tolerancia	x_e	r	p_1	p_2	a	c
ε_p	5	1	5	5	1	1
R	5	3	5	5	3	3

Tabla A3.3.Tolerancias del error

A continuación se explica, en detalle, cada una de las etapas de ajuste:

-*Etapa 1*. Sólo se tiene en cuenta la energía de valles (c_1). La plantilla parte de la posición calculada en la localización inicial, con los parámetros tomados de la configuración inicial, y es ajustada evaluando la energía de valles. El parámetro a ajustar será el vector $\mathbf{x_e}(x_{1e}, x_{2e})$ según la ecuación A3.22, teniendo en cuenta que ahora los componentes son vectores. Las variaciones temporales del vector vendrán dadas por:

$$\overline{f}\,(\overline{x_e}) \;=\; \begin{bmatrix} f_1(x_{e1}) \\ f_2(x_{e2}) \end{bmatrix} \;=\; \begin{bmatrix} \dfrac{dx_{e1}}{dt} \\ \dfrac{dx_{e2}}{dt} \end{bmatrix} \;=\; \begin{bmatrix} -\dfrac{\partial E}{\partial x_{e1}} \\ -\dfrac{\partial E}{\partial x_{e2}} \end{bmatrix} \qquad \text{(A3.25)}$$

-*Etapa 2*. La energía estará formada por la suma de la de bordes (c_2=1) y de valles (c_1=1). A partir de ella se realiza una modulación del radio del iris r hasta que se ajuste a su tamaño correcto, a la vez que corrige la posición de x_e calculada en la etapa anterior, ya que son variables dependientes. Por lo tanto, los parámetros a ajustar serán r y $\mathbf{x_e}$. La función de variación temporal del primero de ellos se calculará según la ecuación (A3.26) y la del segundo como se explicó en la etapa anterior, teniendo en cuenta que ahora E representa la energía de bordes más valles.

$$f(r) \;=\; -\left(\frac{\partial E}{\partial r}\right) \qquad \text{(A3.26)}$$

Después de esta etapa, la posición y tamaño del iris se consideran fijos, de forma que su energía ya no interviene en la actualización de los parámetros posteriores, quedando su influencia reflejada en la energía interna. De esta forma se independiza el ajuste del iris de la esclerótica, suposición perfectamente asumible.

-*Etapa 3*. La energía está compuesta por la de picos y las componentes de la energía interna k_1 y k_2. A partir de ella se obtienen los extremos del blanco del ojo, tanto para la parte derecha como izquierda, que vendrán dados por los vectores $\mathbf{p_1}$ y $\mathbf{p_2}$. A partir de los valores de $\mathbf{x_e}$ y r obtenidos en la etapa anterior y tomando el resto de parámetros de las condiciones iniciales se define el área del blanco de los ojos, como se observa en la figura A3.4. Para ello se toman todos los pixels encerrados por las parábolas y se restan los que hay en el interior del iris. Ajustando los parámetros $\mathbf{p_1}(p_{11}, p_{12})$ y $\mathbf{p_2}(p_{21}, p_{22})$ en función de la energía evaluada sobre el área de blancos se localizarán los puntos de menor energía que darán los extremos buscados.

Al igual que en la etapa 1, en este caso hay que ajustar dos vectores. Sus variaciones temporales vendrán dadas por la siguiente ecuación:

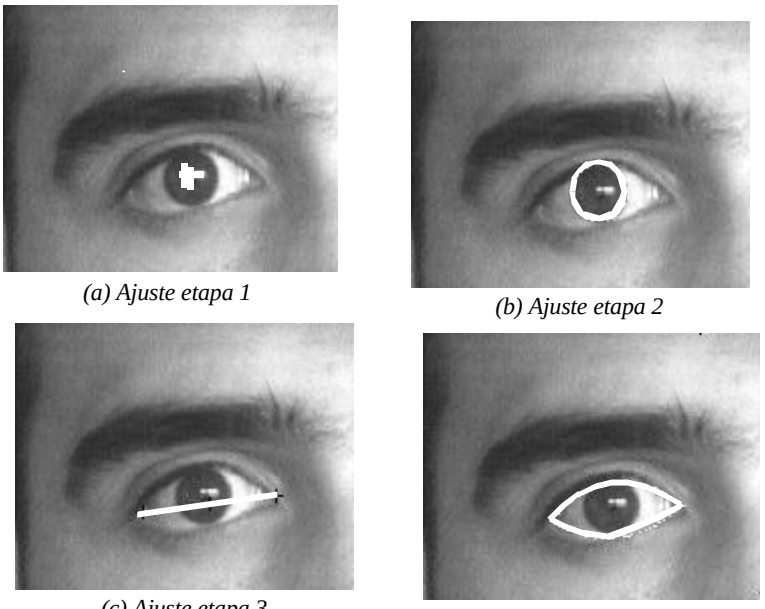
$$\vec{f}\,(\vec{p}_i) \;=\; \begin{bmatrix} f_1(p_{i1}) \\ f_2(p_{i2}) \end{bmatrix} \;=\; \begin{bmatrix} \dfrac{dp_{i1}}{dt} \\ \dfrac{dp_{i2}}{dt} \end{bmatrix} \;=\; \begin{bmatrix} -\dfrac{\partial E}{\partial p_{i1}} \\ -\dfrac{\partial E}{\partial p_{i2}} \end{bmatrix} \quad ; \quad i = 1,2 \qquad \text{(A3.27)}$$

Obsérvese cómo una vez calculados p_1 y p_2 se traza la recta que los une y el punto medio de la misma será: x_e , que no tiene que coincidir con x_c. Por otro lado, el ángulo que forma esta recta respecto a la horizontal dará θ y el parámetro b será igual a la mitad de la distancia entre p_1 y p_2. En la energía de esta etapa intervienen los términos internos k_1 y k_2 que evitan que el sistema se estabilice en unos parámetros muy diferentes de los estándar.

- *Etapa 4.* La energía viene dada por la suma de la de bordes, evaluada sobre las parábolas (c_3), y la energía interna que controla el tamaño de las mismas (k_3). Obsérvese que se ha independizado este ajuste de los anteriores ya que experimentalmente se observó que se obtenían unos resultados similares a los obtenidos sin considerar independencia y el tiempo de proceso era mucho menor. En esta fase se hallarán los valores de los parámetros a y b de las parábolas evaluando la energía en el interior de las mismas sobre la imagen de flancos. Las variaciones temporales de los parámetros vendrán dados por:

$$f(a) = -\left(\frac{\partial E}{\partial a}\right) \quad ; \quad f(c) = -\left(\frac{\partial E}{\partial c}\right) \qquad \text{(A3.28)}$$

En las imágenes de la figura A3.5 se observa el proceso de ajuste en sus diferentes etapas.



(a) Ajuste etapa 1

(b) Ajuste etapa 2

(c) Ajuste etapa 3

(d) Ajuste etapa 4

Figura A3.5. Ejemplo de ajuste de plantilla

A3.6. Resultados experimentales

Se ha probado el método explicado sobre una base de datos faciales de 42 personas (hombres y mujeres) escogidas al azar entre los alumnos de la Escuela Politécnica de la Universidad de Alcalá. Las imágenes fueron tomadas en el laboratorio de investigación del Departamento de Electrónica de dicha Escuela con iluminación ambiente y sin ningún tipo de preparación previa. Las condiciones iniciales asumidas son que todas las caras bajo estudio se situaron delante de un fondo plano y de un color uniforme a una distancia fija, no llevaban ningún objeto contrastado con la cara (gafas, sombrero, etc).

Los resultados obtenidos se pueden ver en la tabla A3.4 donde se evalúan cada una de las etapas de ajuste entre tres conjuntos borrosos: bueno, regular y malo. La clasificación en uno u otro grupo se ha hecho de forma subjetiva siguiendo las siguientes consideraciones: bueno (ajuste perfecto), regular (no se ajusta perfectamente pero el error cometido no es elevado), malo (se comete un gran error).

Etapa	Bueno	Regular	Malo
Inicial	42	0	0
Etapa 1	38	3	1
Etapa 2	41	1	0
Etapa 3	27	10	5
Etapa 4	29	8	5
Total (%)	8476	1047	476

Tabla A3.4. Resultados del método

Aproximadamente un 85% de los ojos fueron localizados y ajustados correctamente, en un 10% el ajuste fue regular y únicamente en un 5% el ajuste fue malo. Si las imágenes están bien contrastadas y sin sombras el grado de acierto sube por encima del 95%, bajando considerablemente si estas condiciones empeoran.

En la figura A3.6 se muestra un ejemplo de ajuste y en la tabla A3.5 los datos obtenidos de la ejecución del algoritmo sobre la imagen de la figura A3.6. En el tiempo de la localización inicial está incluido el tiempo de generación de la función de valles. En la etapa 2 se incluye la generación de la imagen de bordes y se logra el ajuste de r y x_e. En la etapa 3 el tiempo que se indica es la suma del de generación de la imagen de picos y el del ajuste de los puntos p_1 y p_2. Como se puede observar el tiempo de ejecución total obtenido en este ejemplo es del orden de los segundos, muy por debajo de los obtenidos por el método estándar que está entre 5 a 10 minutos. La exactitud que se logra es inferior a la estándar pero como la aplicación está orientada a localización y no a identificación puede ser aceptable.

Figura A3.6. Ejemplo de ajuste

Etapas	**Resultado**	**Tiempo (ms) (Pentium a 100 Mhz + Matrox Image 640)**
Loc. Inicial	x_c=(223,201)	1538
Etapa 1	x_c=(226,203)	989
Etapa 2	r= 24 x_c =(227,204)	7692
Etapa 3	2= 0.60 º	12307
Etapa 4	a=21, b=16	3076
TOTAL		25.6 sg.

Tabla A3.5. Datos obtenidos para la figura A3.6

A.4 SEGUIMIENTO DE LOS OJOS USANDO PROYECCIONES DE PIXELS DE BORDE Y PLANTILLAS DEFORMABLES

En este apéndice se presenta un estudio realizado por el autor sobre el empleo de la técnica de proyección de pixels de borde y la aplicación de plantillas deformables al seguimiento de ojos. Debido a que la robustez del método no era muy elevada y a que el tiempo de cómputo permitía analizar unicamente dos imágenes por segundo, esta línea de investigación fue abandonada.

El método realiza una localización de los ojos de una cara en la imagen inicial de una secuencia de imágenes faciales en movimiento, para posteriormente hacer un seguimiento de los mismos en las siguientes imágenes. Para llevar a cabo estos objetivos, primeramente se obtiene una imagen de bordes de la inicial empleando un filtro Sobel de tamaño 3x3. Seguidamente se divide la imagen de bordes en franjas horizontales y se proyecta cada una de ellas sobre una línea horizontal. De esta forma se logra una reducción de los datos a tratar y una mejor adecuación a la resolución requerida para detectar características faciales.

Analizando las líneas de proyección se observa que la línea que se corresponde con la franja en la que están los ojos es la de mayor densidad de pixels de borde de los tres cuartos superiores de la imagen. Estudiando el nivel de gris de ésta se deduce que los ojos están situados en dos máximos de la línea. En estos máximos se ubicará una plantilla formada por dos círculos que se ajustarán, en posición y radio, a los iris de los ojos, evaluando el nivel medio de gris de los pixels encerrados dentro de la

plantilla.

En las siguientes imágenes se mantienen los parámetros de la plantilla fijos y se hace un seguimiento consistente en la búsqueda del mejor ajuste de la plantilla en el entorno de la localización de la imagen anterior, siguiendo una secuencia de búsqueda en forma de roseta. De esta manera se mejora, en robustez y tiempo de proceso, la versión anterior del método [Bergasa et al., 98a] en la que se usaba una plantilla de ventanas y se realizaba una búsqueda dentro de ellas mediante una umbralización adaptativa y estudiando las características geométricas de los objetos umbralizados.

El sistema desarrollado requiere los siguientes requisitos: (1) El usuario debe estar frente a un fondo plano y uniforme y a una distancia prácticamente constante. (2) El usuario no debe llevar ningún objeto que genere gran contraste con la cara (sombrero, gafas, etc). (3) La imagen facial no debe estar inclinada más de 20º.

A4.1. Método de seguimiento implementado

En este punto se exponen las técnicas utilizadas para detectar los ojos sobre una imagen.

A4.1.1. Obtención de los pixels de borde de la imagen.

Para detectar los bordes de la imagen (I(x,y)) se ha utilizado un filtro Sobel definido por dos máscaras de tamaño 3x3 ($h_1(x,y)$ y $h_2(x,y)$), tal que:

$$G_x(x,y) = I(x,y) * h1(x,y) \quad \text{(A4.1)}$$

$$G_y(x,y) = I(x,y) * h2(x,y) \quad \text{(A4.2)}$$

$$I_B(x,y) = \sqrt{G_x^2(x,y) + G_y^2(x,y)} \quad \text{(A4.3)}$$

Con ello se logran unos resultados similares a los obtenidos por la técnica "Edge focusing" propuesta por Fredrik Bergholm [Bergholm, 89] y aplicada por C. De Silva [De Silva et al., 95] en la extracción de bordes de características faciales, pero disminuyendo el tiempo de proceso de 5 sg. a 100 msg. La imagen de bordes se puede ver en la figura A4.1 (b).

A4.1.2. Técnica de proyección de los pixels de borde

Partiendo de la imagen de bordes ($I_B(x,y)$), se divide ésta en N franjas horizontales de anchura igual a la de la imagen original (ANI) y de altura igual a ALI/24, siendo ALI la altura de la imagen de bordes. Para la obtención de la imagen de bordes proyectada ($I_{BP}(x,y)$) se aplica la siguiente ecuación:

$$I_{PB}(x,k)=\frac{1}{paso}\sum_{y=kpaso}^{(k+1)paso} I_B(x,y) \;\; ; \;\; paso=ALI/N \; ; \; 0\le k\le N-1 \qquad \text{(A4.4)}$$

El número de líneas de proyección que se emplean (N) es de 25. Este parámetro es fijo ya que el usuario está a una distancia fija de la cámara de, aproximadamente, 1m. Teniendo en cuenta el tamaño medio de los ojos y que la cabeza ocupará en torno a tres cuartas partes de la imagen, para una imagen de 640x480 pixels, se obtiene que el tamaño de la franja que contiene a los ojos debe ser de aproximadamente 20 pixels, lo que da un número de líneas de proyección N de 24. En la figura A4.1(c) se muestra el resultado de aplicar el método explicado sobre la imagen de la figura A4.1(b). En la figura A4.1(d) aparece una representación en 3D de las proyecciones de pixels de borde.

A4.1.3. Localización de la línea de los ojos (LO)

Una vez obtenidas las líneas de proyección, el objetivo es localizar la línea que contiene a los ojos. Se comprobó experimentalmente que la línea de los ojos era la que mayor densidad de pixels de borde poseía en los tres cuartos superiores de la imagen, de esta forma se excluyen las líneas de borde correspondientes a los hombros, y que suponen ruido aleatorio para este tipo de localización (usuario con camisa de rayas, etc).

$$Suma\_borde(k)=\frac{1}{ANI}\sum_{x=0}^{ANI-1} I_{PB}(x,k) \qquad \text{(A4.5)}$$

$$LO = máx\{Suma\_borde(k)\} \qquad \text{(A4.6)}$$

A4.2.4. Análisis de la línea de los ojos

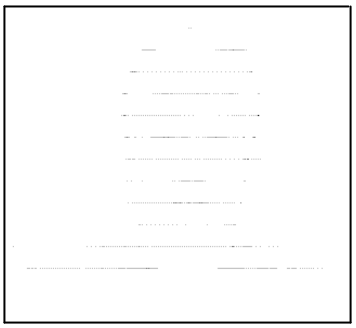
Una vez hallada la línea de los ojos se observa que la posición de los mismos se corresponde con máximos dentro de la línea, como se puede observar en la figura A4.1 (e). Para localizarlos se realiza un filtrado (LOF) y una normalización (LOFN) según las ecuaciones (A4.7) y (A4.8).

$$LOF(x) = \frac{1}{m}\sum_{k=0}^{m-1} LO(x-k) \quad ; \quad m = 9 \tag{A4.7}$$

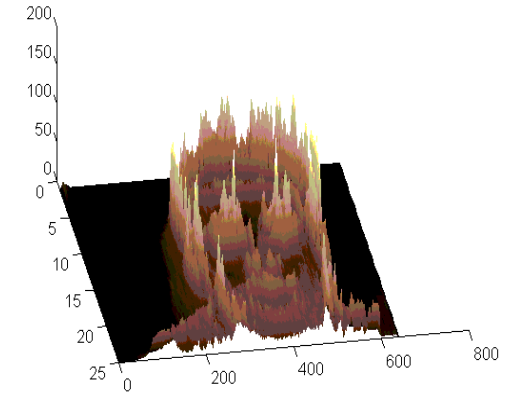
$$LOFN(x) = 255\left(\frac{LOF(x) - LOFMin}{LOFMax - LOFMin}\right) \tag{A4.8}$$

(a) Imagen Original

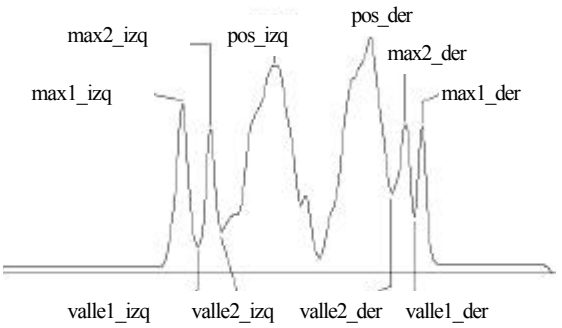
(b)Imagen de bordes



(c)Proyección de pixels de borde



(d) Representación 3D de las proyecciones de pixels de borde



(e)Línea de los ojos

Figura A4.1.Resultado del proceso

A continuación se emplea un método de cálculo de máximos y mínimos sobre la línea de los ojos filtrada, consistente en detectar los cambios de signo de la pendiente de la misma. Inicialmente se obtienen los primeros máximos (max1_der, max1_izq) que se encuentren, partiendo de los correspondientes extremos de la línea. Éstos representan el borde generado entre el fondo uniforme y el pelo de la persona. En segundo lugar, a partir de los primeros máximos se calculan los primeros valles (valle1_der, valle1_izq) que se encuentren hacia el centro de la imagen. En tercer lugar, comenzando en los valles, se calculan de nuevo los siguientes máximos (max2_der, max2_izq). Éstos representan el borde que existe entre el pelo y la cara de la persona. Por último, a partir de los últimos máximos calculados, se obtienen de nuevo los siguientes valles que se encuentren (valle2_der, valle2_izq).

El centro de la cara queda determinado como:

$$d = max2\_der - max2\_izq \quad (ancho\ de\ cara)$$
$$Centro\_cara = d/2 \qquad \text{(A4.9)}$$

Para calcular la posición aproximada de los ojos se buscan los máximos absolutos desde el centro de la cara hasta los segundos valles calculados.

A4.1.5. Plantilla deformable de seguimiento

Para realizar el seguimiento se utiliza la técnica de plantillas deformables. La plantilla utilizada está formada por dos circunferencias de radio variable, como las mostradas en la figura A4.2. Los parámetros a ajustar serán: (x_{oi}, y_{oi}), (x_{od}, y_{od}) y r_o. Inicialmente (x_{oi}, y_{oi}) y (x_{od}, y_{od}) se colocan en los máximos absolutos de la línea de los ojos y r_o=0.061*Distancia de LO a lo alto de la cabeza.

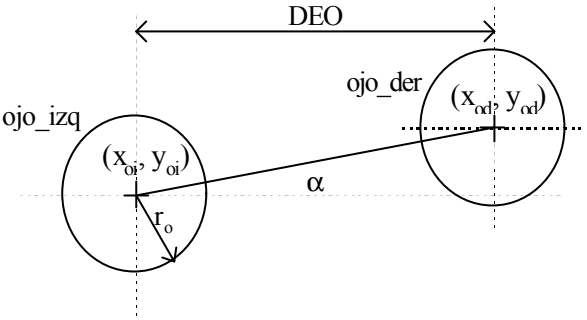


Figura. A4.2.Plantilla de seguimiento

El ajuste de la plantilla a la imagen facial consiste en situar los dos círculos sobre sendos iris de los ojos. Para ello se busca un mínimo en la función de energía E, evaluada independientemente en cada ojo.

$$E = \frac{1}{iris\_plantilla} \iint_{iris\_plantilla} I(x,y)dA \qquad \text{(A4.10)}$$

donde E indica el nivel medio de gris de la imagen sobre el área de la plantilla que se corresponde con el iris, es decir, sobre cada uno de los círculos.

La zona de búsqueda viene definida por el método de la roseta (ver figura A4.3). Está formada por cinco círculos: uno en el centro y otros 4 en las direcciones de 0º,90º,180º y 270º cuyos centros se encuentran separados m pixels del círculo central. Se toma inicialmente la dirección 0º, ya que en principio todas las direcciones tienen la misma probabilidad de ser direcciones de minimización de energía. Se realiza un desplazamiento en esta dirección de m en m pixels hasta obtener un mínimo en la función de energía. Una vez obtenido el mínimo, se mueve la roseta hasta ese punto y se pasa a evaluar la siguiente dirección en el sentido antihorario, realizando el mismo proceso hasta completar las 4 direcciones de la roseta. Al final, el círculo central estará centrado en el iris del ojo. En la siguiente imagen se comienza evaluando la dirección en la que se ha obtenido un máximo desplazamiento en la imagen anterior, de esta forma se realiza una predicción de la dirección del movimiento. Este método

dio mejores resultados que el empleo de la técnica de descenso por el gradiente empleada en [Bergasa et al.,97].

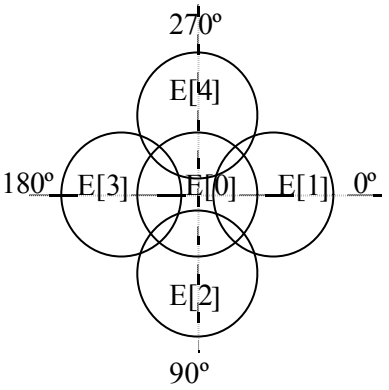


Figura. A4.3.Roseta de búsqueda

El proceso de seguimiento en una secuencia de imágenes se realiza en 3 etapas.

En la *primera etapa* se hace una localización fina de los parámetros (x_{oi}, y_{oi}), (x_{od}, y_{od}), sobre la primera imagen, a partir de sus posiciones iniciales mediante el método de la roseta, tomando como r_o el valor a priori.

En una *segunda etapa* se realiza el ajuste de r_o manteniendo fijos los parámetros de posición calculados en la etapa anterior, también sobre la primera imagen. Se evalúa la energía para un margen de radios comprendidos entre $r_{oinicial}$ "5 pixels y se toma aquel para el que se obtenga un mínimo.

En la *tercera etapa* se lleva a cabo el seguimiento de los ojos, manteniendo constante el radio, y con un valor igual al obtenido en la segunda etapa. El seguimiento se realiza buscando mínimos de energía partiendo de las posiciones de los ojos localizadas en la imagen anterior, como se observa en la figura A4.4. Se impone como requisito que los movimientos del usuario deben ser lentos. No obstante se establecerán unos controles de distancias y ángulos que permiten detectar si la localización ha sido incorrecta, en este caso se vuelve a ejecutar nuevamente el método de detección sobre toda la imagen.

A4.2. Resultados

Se ha probado la robustez del método, en la detección de los ojos en la primera imagen, sobre una base de datos de 50 personas tomadas al azar entre los alumnos de la Escuela Politécnica de la Universidad de Alcalá, logrando una localización con un error menor a radio/2 del iris en el 96 % de los casos. Asimismo se ha calculado el error de seguimiento cometido en 4 secuencias distintas de 100 imágenes, como la mostrada en la figura A4.5.

En la figura A4.5(a) se muestran 3 imágenes de la secuencia 1. En (b) tenemos las coordenadas (x,y) ideales para el ojo derecho calculadas de forma supervisada sobre la secuencia de imágenes (xidealder, yidealder) así como las coordenadas obtenidas por el algoritmo (xrealder, yrealder). En (c) se representan las mismas señales que en (b) pero para el ojo izquierdo y vienen etiquetadas de la siguiente forma: xidealizq, yidealizq, xrealizq, yrealizq.

Como se puede observar, el algoritmo sigue la posición de los dos ojos correctamente. En la Tabla A4.1 aparecen los errores medios de seguimiento obtenidos para las 4 secuencias analizadas. Estos errores, para el peor de los casos, son 9 pixels en horizontal sobre una resolución de 640 y 4 en vertical sobre 480.

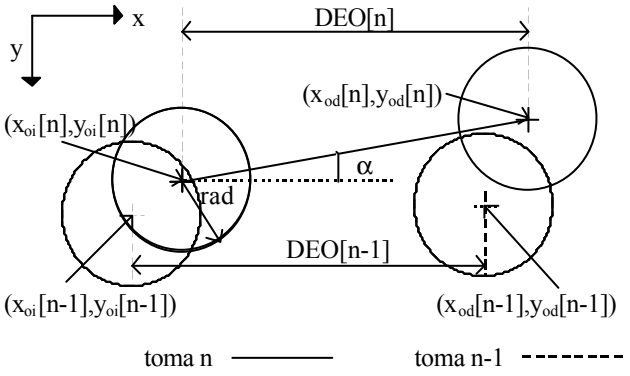


Figura.A4.4.Seguimiento por plantillas

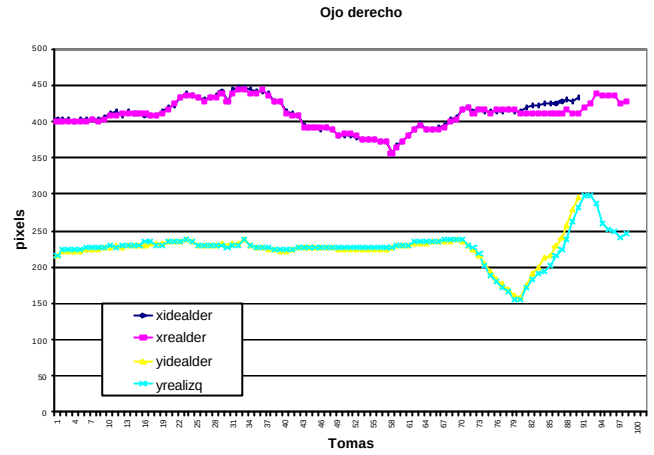
Secuencia	Error medio (pixels)			
	x_izd	y_izd	x_der	y_der
1	2,32	3,04	3,33	3,41
2	8,71	3,32	5,57	2,69
3	3,25	2,89	4,56	3,34
4	3,51	3,15	4,21	2,89

Tabla A4.1. Errores de seguimiento

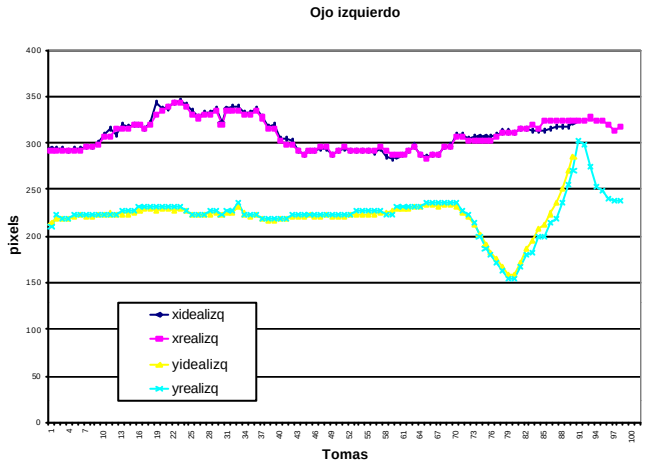
Las imágenes han sido adquiridas a través de una cámara en B/N TC104X de BURLE y digitalizadas con una tarjeta MATROX Image-640 con una resolución de 640x480 pixels instalada en un PC-Pentium a 100Mhz. con 32Mb de RAM. El tiempo de proceso logrado con el hardware indicado es de 2 imágenes por segundo.

La programación se ha realizado en Borland C++ y se ha empleado la librería de funciones MIL de MATROX.

(a) Secuencia de imágenes



(b) Coordenadas del ojo derecho



(b) Coordenadas del ojo izquierdo

Figura. A4.5. Secuencia de seguimiento

BIBLIOGRAFÍA

[Anderson, 73] T.W. Anderson, "Asymptotically efficient estimation of covariance matrices with linear structure". The Annals of Statistics, Vol 1, Nº1, 135-141, 1973.

[Audet et al.,92] J. Audet, Y. Lozac´h, M. Montiglio, D. Mauger, Y. Giasson. "The sliding disk control interface: A successful technology transfer". RESNA 15th Annual Conference, pp.418-420, 1992.

[Ballard&Brown, 82] Ballard D.H. and Brown C.M. (1982). Computer Visión. Prentice-Hall, Englewood Cliffs, New Jersey.

[Baluja&Pomerleau, 94]. S. Baluja and D. Pomerleau. "Non-Intrusive Gaze Tracking Using Artificial Neural Networks". Research Paper CMU-CS-94-102, School of Computer Science, Carnegie Mellon University, Pittsburgh PA, USA. 1994

[Beattie&Bishop,98]P.D. Beattie, J.M. Bishop. "Self-localisation in the "Senario" autonomous wheelchair". Journal of Intelligent and Robotic Systems. Nº22, pp:255-267, Kluwer Academic Publishers, 1998.

[Bell et al.,94]D. A. Bell, J. Borenstein, S.P. Levine, Y. Koren, L. Jaros. "An assistive navigation system for wheelchairs based upon mobile robot obstacle avoidance". IEEE International Conference on Robotics an Automation. pp.2018-2022. San Diego. 1994.

[Bergasa et al.,96]L.M. Bergasa, M. Mazo, C. T. San Juan, J. A. Herradas. "Guiado de un móvil mediante los movimientos oculares". I Jornadas de Inteligencia Artificial, Control y Sistemas Espertos, E.U. Politécnica. Universidad de Alcalá, pp 25-33. 1996.

[Bergasa et al., 97] L.M. Bergasa, M. Mazo, L. Boquete, M.A. Sotelo, R. Asensio, M.A. García. "Identificación de los ojos de una cara usando plantillas deformables". SAAEI'97, pp 590-596,

Septiembre 1997, Valencia.

[Bergasa et al., 98a] L. M. Bergasa, M. Mazo, A. Gardel, R. Mateos, D. Altés, R. Matey. "Detección y seguimiento de características faciales usando proyecciones de pixels de borde y una plantilla de ventanas". Informática'98. Conferencia Internacional sobre el Control de Sistemas Industriales. La Habana, Cuba, Febrero 1998.

[Bergasa et al., 98b] L. M. Bergasa, M. Mazo, L. Boquete, M.A. García, A. Gardel "Seguimiento de los ojos de una cara usando proyecciones de pixels de borde y una plantillas deformables". SAAEI'98, pp 309-312, Septiembre 1998, Pamplona.

[Bergholm, 89] F.Bergholm "On the information in edges and optical flow". PhD Thesis. University of Stockholm.Sweden. May 1989.

[Borgolte et al.,98] U. Borgolte, H. Hoyer, C. Bühler, H. Heck, R. Hoelper. "Architectural concepts of a semi-autonomous wheelchair". Journal of Intelligent and Robotic Systems. Nº22, pp:233-253, Kluwer Academic Publishers, 1998.

[Bourhis&Agostini,98] G. Bourhis, Y. Agostini. "Man-machine cooperation for control of an intelligent powered wheelchair". Journal of Intelligent and Robotic Systems. Nº22, pp:269-287, Kluwer Academic Publishers, 1998.

[Brown, 95] C. Brown, "Tutorial of Filtering, Restoration and State Estimation". Technical Report 534, University of Rochester, Computer Science Departament, 1995.

[Bar-Shalom&Fortmann, 88] Y. Bar-Shalom and T. E. Fortmann. "Tracking and Data Association", Academic Press.

[Campbell et al. 96] N.W. Campbell,B.T. Thomas and T. Troscianko. "Segmentation of Natural Images using Self-Organising Feature Maps". In British Machine Vision Conference, pages 223-232. British Machine Vision Association, September 1996.

[Campbell&Thomas 97] N. W. Campbell and B. T. Thomas. "Automatic Selection of Gabor Filters

for Pixel Classification". In Sixth International Conference on Image Processing and its Applications, pages 761-765. IEE, July 1997.

[Canon, 95] Test af Canon UC-X1Hi, HIFI electronik, 1995

[Chapman, 91] J.E. Chapman. "Use of an eye-operated computer system in locked-in syndrome" CSUN's 6th Annual International Conference: Technology and Persons with Disabilities, LC Technologies, Virginia U.S.A. 10p.

[Cheng&Zelinsky,97] G. Cheng and A. Zelinsky, "Supervised Autonomy: A Paradigm for Teleoperating Mobile Robot", Proceeding of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), September1997

[Civit&Abascal, 98]A. Civit and J. Abascal. "TetraNauta:A Wheelchair Controller for Users with Very Severe Mobility Restrictions". Improving the Quality for the European Citizen. I. Placencia Porrero and E. Ballabio (Eds.)IOS Press, 1998.

[Clarke et al.,98] L. Clarke, P. Harper and R. B. Reilly. "Video Based Gesture Recognition for Augmentative Communication" Improving the Quality for the European Citizen. I. Placencia Porrero and E. Ballabio (Eds.) IOS Press, 1998.

[Craig&Nisbet,93] I. Craig, P. Nisbet. "The smart wheelchair:An augmentative mobility toolkit". ECART Conference. pp. 13-27. Stockholm, Sweden, 1993.

[Crowley&Coutaz,95] J.L. Crowley and J. Coutaz, "Vision for Man Machine Interaction" EHCI'95. Grand Targhee, August 95.

[Crowley et al.,95] J.L. Crowley, F. Bérard and J. Coutaz, "Finger Tracking as an Input Device for Augmented Reality", IWAGFR'95- International Workshop on Gesture and Face Recognition, Zurich, June 1995.

[Curwen et al., 91] R. M. Curwen, A Blake and R. Cipolla, "Parallel implementation of Lagrangian dynamics for real-time snakes. In Mowforth, P., editor, British Machine Vision Conference, 29.35,

Glasgow. Springer-Verlag, London.

[Daugman, 97] J. Daugman, "Face and Gesture Recognition: Overview", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, Nº 7, pp:675-676, July 97.

[De Silva et al., 95a] L.C. De Silva, K. Aizawa, M. Hatori. "Detection and tracking of facial features". In Proc. of SPIE Visual Communications and Image Processing'95 (VCIP'95), pp.2501/1161-2501/1172, Taipei, Taiwan. May 1995.

[De Silva et al., 95b] L.C. De Silva, K. Aizawa, M. Hatori. **"**Detection and tracking of facial features by using edge pixel counting and deformable circular template matching". IEICE TRANS INF&SYST. VOL.E78-D, No 9, pp.1195-1207, September 1995.

[De Silva et al., 95c] L.C. De Silva, M. Tahara, K. Aizawa. " A teleconferencing system capable of multiple person eye contact (MPEC) using half mirrors and cameras placed at common points of extended lines of gaze". IEEE Transactions on circuits and systems for video technology. Vol.5. Nº 4, August 1995.

[Dubuisson et al., 96] Marie-Pierre Dubuisson, Anil K. Jain and Sridhar Lakshmanan. Vehicle Segmentation and Classification using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18 Nº 3, pp 293-307. March 1996.

[Duda&Hart,73] R. O. Duda and P.E. Hart, "Pattern Classification and Scene Analysis", J. Wiley, N.Y., 1973.

[Echelon, 95] LonManager DDE Server user's guide. Echelon Corporation,USA, 1995

[Escudero, 77] L. F. Escudero, "Reconocimiento de patrones", Ed. Paraninfo, Madrid, 1977

[Espinosa, 98] F. Espinosa, "Aportación al modelado y control óptimo adaptativo de tracción y dirección para el guiado de vehículos autónomos" Tesis Doctoral. Escuela Politécnica. Universidad de Alcalá, 1998.

[Essa&Pentland, 97] I. A. Essa and A. P. Pentland, "Coding, Analysis, Interpretation and Recognition of Facial Expressions", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 19, Nº 7, July 1997

[Ferrario&Lodola,92] M. Ferrario, M. Lodola. "A multifunctional nape joystick" RESNA 15th Annual Conference, pp.524-526, 1992.

[Frey et al, 92] L.A. Frey, K.P. White and T.E. Hutchinson "Eye-gaze word processing", IEEE Transactions on Systems, Man and Cybernetics Nº20 (4), pp 944-950, 1992.

[García et al.,97] J.C. García, M. Marrón, J.A. García, M.A. Sotelo, J. Ureña, J.L. Lázaro, F.J. Rodríguez, M. Mazo, M. Escudero. "An autonomous wheelchair with a LonWorks network based distributed control system".In Proc. of II Field and Services Robotics (FSR'97). Canberra. Australia. December 1997.

[Gath et al.,89] I. Gath and B. Geva. "Unsupervised Optimal Fuzzy Clustering". IEEE Trans. Pattern Analysis and Machine Intelligence", Vol.11, Nº7, pp.773-781, July 1989.

[Gee&Cipolla, 94a] A.H. Gee and R. Cipolla. "Determining the gaze of faces in images". Department of Engineering. University of Cambridge. Cambridge CB2 1PZ, England. March 1994.

[Gee&Cipolla, 94b] A.H.Gee and R. Cipolla. "Non Intrusive Gaze Tracking for Human-Computer Interaction". Department of Engineering. University of Cambridge. Cambridge CB2 1PZ, England. 1994.

[Gee&Cipolla, 95] A.H. Gee and R. Cipolla. "Fast visual tracking by temporal consensus".Department of Engineering. University of Cambridge. Cambridge CB2 1PZ, England. February 1995.

[Gee&Cipolla, 96] K.C. Yow and R. Cipolla. "Finding Initial Estimates of Human Face Location." Department of Engineering. University of Cambridge. Cambridge CB2 1PZ, England. 1996

[Glenstrup&Engell-Nielsen,95] A. J. Glenstrup and T. Engell-Nielsen, "Eye Controlled Media: Present and Future State". PhD, DIKU (Institute of Computer Science) University of Copenhagen, Denmark,

1995.

[Gong&Sakauchi,95]Y. Gong and M. Sakauchi, "Detection of Regions Matching Specified Chromatic Features", Computer Vision and Image Understanding, Vol.61, Nº2, pp.263-269, 1995.

[Gonzalez et al., 95] A.I. Gonzalez, M. Graña and A. D'Anjou, "An Analysis of the GLVQ Algorithm", IEEE Transactions on Neural Networks, Vol. 6, Nº 4, pp: 1012-1016, July 1995.

[Hallinan, 92] Hallinan P.W. A robust deformable template for human eyes. Technical report. Robotics Laboratory, Harvard University, 1992.

[Harwin&Rahman,92] W. Harwin, T. Rahman. "Safe software in rehabilitation mechatronic and robotics design".RESNA 15th Annual Conference, pp.100-102, 1992.

[Heinzmann&Zelinsky, 97]Heinzmann and A. Zelinsky, "Robust real-time face tracking and gesture recognition". Proceedings of IJCAI´97, International Joint Conference on Artificial Intelligence, August 1997.

[Hoyer et al.,94] H. Hoyer, R. Hoelper. "Intelligent omnidirectional wheelchair with a flexible configurable functionality". RESNA 17 th. Annual Conference. pp. 353-355. Nashville. 1994.

[Hunke, 94] M. Hunke, "Locating and Tracking of Human Faces with Neural Networks". Technical Repor CMU-CS-94-155, School of Computer Science, CMU, Pittsburgh, U.S.A.,1994.

[Hunke&Waibel,94] M. Hunke and A. Waibel, "Face Locating and Tracking for Human-Computer Interaction",Proc. 28th Asilomar Conference on Signals, Systems&Computers, Monterey, CA, USA, 1994

[Jain et al.,96] Anil K. Jain, Yu Zhong and Sridhar Lakshmanan. "Object Maching using Deformable Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 18 Nº 3, pp 267-278. March 1996.

[Karayiannis et al.,96]N.B. Karayiannis, J.C. Bezdek, N.R. Pal, R.J. Hathaway and P. Pai,"Repairs

to GLVQ: A New Family of Competitive Learning Schemes", IEEE Transactions on Neural Networks, Vol. 7, Nº 5, pp: 1062- 1071, September 1996.

[Katevas et al.,95]N. Katevas, S. Tzafestas et al. "SENARIO- The autonomous mobile robotics technology for locomotion handicap. Operational and technical issues" 2nd TIDE Congress. pp. 371-374. Paris.

[Kohonen, 97] T. Kohonen, "Self-Organizing Maps", Springer-Verlag, 1997

[Lagan et al.,98] D.A. Lagan, J.W. Modestino and J.Zhang. "Cluster Validation for Unsupervised Stochastic Model-Bases Image Segmentation". IEEE Transactions on Image Processing, Vol. 7, Nº2, pp. 180-195, February, 1998

[Lawrence et al., 97] S. Lawrence, C. L. Giles, A. C. Tsoi and A. D. Back "Face Recognition: A convolutional Neural-Network Approach", IEEE Transactions on Neural Networks, VOL 8, NO 1, January 1997.

[Littmann&Ritter, 97]E. Littmann and H. Rilter, "Adaptive Color Segmentation-A comparation of Neural and Statistical Methods", IEEE Transactions on Neural Networks, VOL 8, NO 1, January 1997.

[López, 99] M.E. López, "Aplicación de controladores óptimo y borroso al seguimiento de trayectorias de una silla de ruedas", Proyecto Fin de Carrera, Escuela Politécnica, Universidad de Alcalá, 1999.

[Maravall,93] D. Maravall Gómez-Allende, "Reconocimiento de formas y visión artificial", Ed. RAMA, 1993.

[Matía et al.,98] F. Matía, R. Sanz, E.A. Puente. "Increasing intelligence in autonomous wheelchairs". Journal of Intelligent and Robotic Systems. Nº22, pp:211-232, Kluwer Academic Publishers, 1998.

[Mazo et al.,95b] M. Mazo, F.J. Rodríguez, J.L. Lázaro, J. Ureña, J. C. García, E. Santiso, P. Revenga. "Electronic control of a Wheelchair Guided by Voice Commands". Control Eng. Practice, vol.3, nº5, pp 665-674, 1995.

[Mazo et al.,95a] M. Mazo, F.J. Rodríguez, J.L. Lázaro, J. Ureña, J. C. García, E. Santiso, P. Revenga and J.J. García. "Wheelchair for Physically Disabled People with Voice, Ultrasonic and Infrared Sensor Control". Autonomous Robots, 2, pp:203-224. Kluwer Academic Publishers, 1995.

[Mazo et al. 98] M. Mazo, F. J. Rodríguez, J. Ureña, J. C. García, J. L. Lázaro, R. García."Integral System for Assisted Mobility". 2nd International Workshop on Intelligent Control (IC´98)JCIS'98 Proceedings. Editor: Paul P. Wang. pp: 361-364. Durham (USA), 1998.

[Meier et al., 97] U. Meier, R. Stiefelhagen and J. Yang "A preprocessing of visual speech under real word conditions".Proceedings of European Tutorial & Research Work Shop on Audio-Visual Speach Proccesing. 1997.

[Meier et al. 99] U. Meier, R. Stiefelhagen, J. Yang, A. Waibel "Towards unrestricted lip reading". Second International Conference on Multimodal Interfaces (ICMI99), Hong Kong 1999. (to appear)

[Melén, 98]K. Melén. "Independent Transport for Persons with Severe Multiple Disabilities- The "Slingan" Project". Improving the Quality for the European Citizen. I. Placencia Porrero and E. Ballabio (Eds.)IOS Press, 1998.

[Moreira&Fontoura,96] J. Moreira and L. Da Fontoura Costa. "Neural-based color image segmentation and classification using self-organizing maps". IX SIBGRAPI Proceedings, pp.47-54, October 1996.

[Nelisse,98]. M.W. Nelisse. "Integration Strategies using a modular architecture for mobile robots in the rehabilitation field". Journal of Intelligent and Robotic Systems. Nº22, pp:181-190, Kluwer Academic Publishers, 1998.

[Niemann, 90] H. Niemann, "Pattern Analysis and Understanding", Springer Series in Information Sciences, Ed. Springer-Verlag, 1990.

[Oliver et al.,96] J.J. Oliver, R.A. Baxter, C.S. Wallace. "Unsupervised learning using MML", Proc. 13th Int´l Conf. Machine Learning (ICML'96). Pp.364-372, San Francisco, 1996.

[Poveda et al.,98] R. Poveda, R. La Fuente, J. Sánchez-LaCuesta, E. Viosca, J. Prat, J.M. Belda, C. Soler-Gracia. "Problemática de los usuarios de sillas de ruedas en España". Instituto de Biomecánica de Valencia. ISBN:84-923974-0-3. 1998

[Rao, 91] B. S.Y. Rao, "Data Fusion Methods in Decentralized Sensing Systems", PhD thesis, University of Oxford, Robotics Research Group.

[Rissanen, 78] J. Rissanen. "Modeling by Shortest Data Description", Automatica, Vol.14, pp. 465-471, 1978.

[Roberts et al., 98]S.J. Roberts, Dirk Husmeier, Iead Rezek and William Penny. "Bayesian Approaches to Gaussian Mixture Modeling". IEEE Transactions on Image Processing, Vol. 20, Nº11, pp.1133-1142,November, 1998

[Roberts, 97] S.J. Roberts, "Parametric and Non-Parametric Unsupervised Cluster Analysis", Pattern Recognition, Vol. 30, Nº 2, pp. 261-272. 1997

[Rodríguez, 97] F. J. Rodríguez Sánchez, "Contribución a un sistema de detección de bordes de carreteras, mediante visión artificial, orientado al guiado de un robot móvil", Tesis Doctoral, Dpto Electrónica, Universidad de Alcalá, 1997.

[Rowley et al., 98] H. A. Rowley, S. Baluja and T. Kanade, "Neural Network-Based Face Detection". IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol 20, Nº 1, January 1998.

[S-H Lin et al.,97] S-H Lin, S-Y Kung and L-J lin "Face Recognition by Probabilistic Decision-Based Neural Network", IEEE Transactions on Neural Networks, Vol 8, Nº 1, January 1997.

[Schilling et al.,98] K. Schilling, H. Roth, R. Lieb and H. Stützle. "Sensors to Improve the Safety for Wheelchair Users". Improving the Quality for the European Citizen. I. Placencia Porrero and E. Ballabio (Eds.)IOS Press, 1998.

[Scott&Findlay, 93]D. Scott and J.M. Findlay. "Visual search, eye movements and display units", Human factors report, University of Durham, South Road, Durham, DH1 3LE, UK, 1993.

[Serra, 82] Serra, J., 1982. Image, Analysis and Mathematical Morphology, Academic Press: New York.

[Shackleton& Welsh, 91] M.A. Shackleton and W.J. Welsh. "Classification of Facial Features for Recognition." IEEE CVPR, pp 573-578, 1991.

[Smyth et al, 94] C. Smyth, B.B. Bates, M.C. Lopez and N.R. Warw. "A comparation of eye-gaze to touch panel and head-fixed recicle for helicopter display control and target acquisition during a simulated armed reconnaissance mission". Proceedings of the 2nd Mid-Atlantic Human Factors Conference, pp 49-53, 1994.

[Stiefelhagen et al., 97a] Stiefelhagen, J. Yang and A. Waibel "A model based gaze tracking system". IEEE International Joint Symposia on Intelligence and Systems-Image, Speech&Natural Language Systems. 1997

[Stiefelhagen et al., 97b]R. Stiefelhagen, J. Yang and Alex Waibel "Tracking Eyes and Monitoring Eye Gaze", Proc. of the Workshop on Perceptual User Interfaces, pp. 98-100, Banff, Canada, October 1997.

[Sung&Poggio, 98] K. Sun and T. Poggio. "Example-based learning for view-based human face detection". IEEE Transactions on Pattern analisis and machine intelligence, Vol 220, nº1 , January 1998.

[Swain, 90] M.J. Swain, "Color Indexing" PhD Thesis, Supervised by Dana H. Ballard, Departament of Computer Science, University of Rochester, New York, 1990.

[Umbaugh, 98] S. E. Umbaugh. "Computer Vision and Image Processing". Ed. Prentice Hall, 1998.

[User's Manual of SMI, 95] SensoMotoric Instruments GmbH "View- System options". Teltow, Germany, 1995

[Van Der Loos et al.,92] H.F.M. Van Der Loos, D.S. Lees, L.J. Leifer. "Safety considerations for rehabilitative and human service robot systems". RESNA 15th Annual Conference, pp.322-324, 1992.

[Wakaumi et al.,92]H. Wakaumi, K. Nakamura, T. Matsumura. "Development of an automated wheelchair guided by a magnetic ferrite marker lane". J. Rehabilitation Research and Development. Nº 29, pp.27-34. 1992

[Yanco&Gibs, 97] H.A. Yanco and J. Gibs. "Preliminary Investigation of a Semi-Autonomous Robotic Wheelchair Directed Through ElectrodesProceedings of the Rehabilitation". Engineering and Assistive Technology Society of North America Annual Conference, RESNA Press, 1997, pp. 414-416.

[Yang&Waibel, 97], J. Yang, A. Waibel, "Skin-Color Modelong and Adaptation", Technical Report CMU-CS-97-146. School of Computer Science. Carnegie Mellon University, 1997.

[Yang et al., 98a] J. Yang, R. Stiefelhagen, U. Meier and A. Waibel, "Real-Time Face And Facial Feature Tracking And Applications". Proceedings of Workshop on Audio-Visual Speech Processing, pp. 79-84, Terrigal, South Wales, Australia, 1998.

[Yang et al., 98b] J. Yang, R. Stiefelhagen, U. Meier and A. Waibel, "Visual Tracking for Multimodal Human Computer Interaction" Human Factors in Computing Systems: CHI 98, pp. 140-147, Los Angeles, CA, April 1998

[Yow&Cipolla, 95] K.C. Yow and R. Cipolla. "Towards an automatic human face localization system." Department of Engineering. In Proc. British Machine Vision Conference, Vol 2, pp. 701, Birmingham, October 1995. Springer-Verlag.

[Yow&Cipolla, 96] K.C. Yow and R. Cipolla. "Feature-based human face detection". Department of Engineering. University of Cambridge. Cambridge CB2 1PZ, England. August 1996.

[Yuille et al.,92]Yuille A.L., Hallinan P.W. and Cohen D.S. " Feature extraction from faces using deformable templates." International Journal of Computer Vision, pp. 99-111, 1992.

[Yuille&Hallinan, 92]Yuille A.L. and Hallinan P.W. "Deformable Templates in Active Vision", ed. A. Blake and A.L. Yuille. M.I.T. Press,1992.

[Zelinsky&Heinzmann, 96] A. Zelinsky and J. Heinzmann, "Real-time visual recognition of facial gestures

for human-computer interaction". Proc. Of IEEE International Conference on Automatic Face and Gesture Recognition, pp 351-356, October 1996.

[Zhang et al., 97]J. Zhang, Y. Yan, M. Lades. "Face Recognition: Eigenface, Elastic Matching and Neural Nets". Proceedings of the IEEE, Vol 85, No 9, September 1997.