Escola Tècnica Superior d'Enginyeria de Telecomunicació de Barcelona
UNIVERSITAT POLITÈCNICA DE CATALUNYA

Departamento de Teoría de la Señal y Comunicaciones

Proyecto Final de Carrera

Detección de Gestos en Procesadores de Plano Focal

Autor: Mario García Martín
Director: Josep Ramon Morros i Rubió

Barcelona, Febrero de 2010

Departament de Teoria del Senyal i Comunicacions

A todas las personas que me han apoyado
durante la realización de este proyecto.

"If I find 10,000 ways something won't work, I haven't failed.
I am not discouraged, because every wrong attempt
discarded is another step forward"

–Thomas A. Edison–

Resumen

En este trabajo se estudia el problema de la detección de un conjunto de gestos de la mano. Los gestos son captados por medio de un sistema de visión constituido por una cámara que incorpora un procesador de plano focal (FPP). Este tipo de cámaras incorpora circuitería auxiliar en la capa de sensado permitiendo trabajar en paralelo en cada uno de los píxeles sin necesidad de descargar las imágenes.

El problema de la detección de gestos de la mano es abordado en el contexto de la interacción entre humano y computadora (HCI). En un mundo cada vez más informatizado, el avance hacia una interacción de alto nivel entre el ser humano y las computadoras es prácticamente un imperativo. El objetivo es simplificar el uso de los dispositivos y mejorar la usabilidad de la tecnología para hacerla accesible a un mayor número de personas.

El sistema está configurado de forma que el problema se afronta siguiendo principalmente tres etapas. La primera de ellas tiene como propósito localizar la región que corresponde a la mano y el brazo. La segunda realiza la extracción de una serie de características que constituyen el modelo de la mano considerado. En la tercera y última etapa se realiza el análisis de los patrones que cumplen estas características con el objetivo final de realizar la clasificación del gesto realizado.

Uno de los requisitos principales de las aplicaciones basadas en HCI es que deben dar respuesta a los impulsos en tiempo real. Existe, sin embargo, un compromiso entre el tiempo de respuesta del sistema y la complejidad del modelo de la mano utilizado.

Índice

Capítulo 1

Introducción

1.1. Motivación

El concepto de *Human-Computer Interaction* (HCI) surge de forma intrínseca con el nacimiento de las computadoras. La razón, de hecho, se fundamenta en una idea bastante clara: para que una máquina capaz de llevar a cabo tareas complejas sea útil tiene que estar capacitada para ser manipulada y gestionada por el ser humano.

La interacción entre seres humanos y computadoras es, por ello, una disciplina de gran importancia en multitud de aplicaciones, incluyendo la inteligencia artificial, la robótica, el reconocimiento facial y la interpretación de gestos de la mano entre otros.

Sin embargo, aunque las computadoras han evolucionado tremendamente, en la mayoría de las aplicaciones que se pueden encontrar en la actualidad, la interacción entre humano y computadora se realiza a través de equipos mecánicos simples (como por ejemplo el ratón, el teclado o el joystick).

Estos transductores mecánicos, aunque ofrecen una solución precisa, presentan una serie de inconvenientes que impiden una interacción entre humano y computadora plena y natural. Probablemente, el inconveniente más importante es que comportan la dependencia con un elemento hardware, que puede llegar a ser molesta e intrusiva para el usuario. Existen en los dispositivos mecánicos, además, ciertas restricciones impuestas a los movimientos de los usuarios. Estas restricciones pueden ser causadas por el peso o la incomodidad de éstos. Los dispositivos mecánicos requieren, asimismo, una curva de aprendizaje por parte del usuario para adaptarse a su utilización. Por todo ello, se produce una reducción drástica en la efectividad y naturalidad de la interacción.

La interacción óptima o ideal viene marcada por la interacción natural que se da entre humano-humano. Es un hecho que, para conseguir una interacción satisfactoria, el computador debe ser capaz de interactuar de manera natural con el usuario, de forma similar a como se produce la interacción entre seres humanos. En caso contrario, el intento de comunicación puede llegar a ser insatisfactorio e incluso frustrante.

La interacción entre humanos es, por tanto, la más completa y está basada principalmente en el habla. No obstante, también se encuentran presentes otros tipos de interacción que complementan al habla y sirven para enfatizar ciertos aspectos de la conversación y expresar emociones. Entre éstos se encuentran elementos tan complejos como la mirada, los gestos del

cuerpo (y en concreto de las manos) o la predisposición de las personas que intervienen en la comunicación.

Desde los primeros días de las computadoras se ha invertido un gran esfuerzo en intentar hacerlas entender el habla humana. No ha sido hasta los últimos años que no ha aparecido un interés creciente en intentar introducir los otros aspectos de la interacción entre los seres humanos.

En la actualidad se puede observar la –cada vez más frecuente– aparición de aplicaciones comerciales que dejan atrás el antiguo concepto de interacción basado en elementos mecánicos. La mayoría de las introducciones se han visto en el mercado del ocio: mandos de videoconsolas novedosos que son capaces de capturar el movimiento, móviles *touch sensitive*, etc.

En un contexto cada vez más orientado hacia el mundo audiovisual, es fácil comprobar que la tendencia de las interfaces va dirigida, como norma general, hacia los elementos gráficos. Esto no hace más que facilitar e incentivar el cambio hacia un tipo de interacción más avanzada con los dispositivos.

Debido a la gran expresividad de los gestos manuales, existe un alto aliciente en el estudio de la viabilidad de sistemas que sean capaces de interpretarlos. Los gestos de la mano son manifestaciones de interacción no verbal entre la gente que se componen desde simples acciones como por ejemplo apuntar objetos para llamar la atención sobre ellos, hasta los más complejos que son capaces de expresar sentimientos o permiten comunicarse con los demás.

La interpretación visual de los gestos de la mano o del brazo mediante un sistema de visión (*Computer Vision*) conlleva una ventaja tremenda sobre otras técnicas que requieren el uso de transductores mecánicos: no es de tipo intrusiva. Sin embargo, también comporta un aumento de la complejidad en la implementación.

Los sistemas de visión que incorporan un Procesador de Plano Focal (FPP) son capaces de realizar operaciones sobre las imágenes captadas, directamente en la misma capa física utilizada para el sensado. Este tipo de cámaras ofrecen, por tanto, una solución embebida con una capacidad de procesado inicial muy elevada.

A diferencia de los sistemas de visión autónomos más comunes, las cámaras que incluyen un procesador de plano focal permiten realizar operaciones de procesado de forma paralela en tiempo real y a resolución completa.

1.2. Objetivos

En el presente proyecto se intenta abordar el reconocimiento de una serie de gestos realizados con la mano por medio de un sistema de visión basado en una cámara que incorpora un procesador de plano focal. Se remite al lector al apartado 2.1 para un conocimiento más profundo de este tipo de cámaras.

El objeto es ofrecer al usuario final un sistema capaz de entender e interpretar una serie de gestos preestablecidos. El conjunto de movimientos a interpretar está compuesto por un juego de gestos tanto estáticos como dinámicos.

El sistema está formado por una única cámara que funciona de manera autónoma capturando, analizando y reconociendo los estímulos directamente en el hardware de que está compuesta. Por ello, se trata de una solución compacta, versátil y de bajo precio. Debido a las características de la cámara, es necesaria la implementación de algoritmos alternativos a soluciones ya tratadas con otras cámaras.

El cometido de este trabajo se enmarca dentro del proyecto CENIT VISION [1], que intenta combinar la información derivada de los diferentes caminos posibles de interacción para integrarla y lograr una comunicación capaz de transmitir una verdadera sensación de realidad y presencia.

La idea principal es que el simple hecho de transmitir imágenes y sonido en un sistema de videoconferencia no es suficiente para conseguir una sensación de presencia real en las comunicaciones. Es necesaria la existencia de otras capacidades como son la sensación de contacto visual, la visión de imágenes desde cualquier punto de vista y con sensación de profundidad, la interactividad real por medio de interfaces naturales y sin que se perciba ningún tipo de retardo en la respuesta visual y auditiva, etc.

El objetivo del proyecto VISION es la consecución de un salto cualitativo en las comunicaciones digitales audiovisuales para que las personas separadas por grandes distancias perciban la sensación de estar físicamente reunidas en un mismo lugar.

1.3. Organización del proyecto

Este trabajo sigue una estructura organizada en cuatro bloques principales.

El primer bloque es una introducción al estado del arte de las diferentes disciplinas que se entrecruzan en este proyecto. La sección 2.1 se centra en el análisis de la arquitectura de los procesadores de plano focal haciendo especial hincapié en el sistema de visión Eye-RIS v1.2 [2] que es el dispositivo utilizado como *front end* de la aplicación desarrollada. La sección 2.2 describe una forma particular de descomposición de imágenes en nivel de gris a imágenes binarias que resulta muy práctica en el contexto de operación con procesadores de plano focal. Se introduce el trabajo realizado en la implementación de algunos operadores morfológicos básicos adaptados a esta forma de trabajar con las imágenes. Finalmente, la sección 2.3 desarrolla el estado del arte en que se encuentran los sistemas de interacción entre humanos y computadoras (HCI).

El segundo bloque trata de dar una visión de la importancia del usuario en los sistemas HCI. A lo largo del Capítulo 3 se explica el procedimiento que se ha seguido en la elección de los gestos a interpretar. Para desarrollar con éxito una aplicación HCI no sólo hay que tener en mente el

apartado de implementación o las limitaciones tecnológicas, también juega un papel fundamental el concepto de usabilidad.

El tercer bloque está dedicado al estudio de la solución desarrollada. El Capítulo 4 trata acerca del funcionamiento general del sistema. En la sección 4.3 se refiere el procedimiento que permite determinar la posición de la mano así como el momento en que se produce el inicio y el fin del gesto. En la sección 4.4 se describe la forma en que se sigue la evolución de la mano durante el movimiento. Finalmente, en la sección 4.5 se explica el modo de determinar en tiempo real si el movimiento es de tipo estático o dinámico. Esto permite definir una manera de proceder diferente en función del tipo de movimiento. En el Capítulo 5 se discuten los métodos de análisis y clasificación de los gestos estáticos. En las secciones 5.2.1 a 5.2.4 se presentan cuatro métodos de conteo de dedos y en la sección 5.2.5 se realiza un análisis acerca de las ventajas e inconvenientes de cada uno de ellos con el objetivo final de llegar a una decisión con respecto a cuál es el más conveniente para la implementación en el sistema de visión Eye-RIS. El Capítulo 6 muestra los criterios de clasificación empleados en el caso de los movimientos dinámicos.

El cuarto bloque cierra el proyecto. En el Capítulo 7 se muestran los resultados de detección obtenidos en pruebas empíricas del sistema. Por último, el Capítulo 8 proporciona algunas conclusiones y propone posibles líneas de trabajo futuro.

Capítulo 2

Estado del arte

2.1. El Procesador de Plano Focal

En determinadas aplicaciones donde el espacio, el coste, el consumo o la velocidad de operación son elementos determinantes, puede resultar interesante disponer de una cámara que permita realizar ciertas operaciones directamente en la misma capa que es utilizada para captar la imagen.

La incorporación de una circuitería analógica y/o digital auxiliar en el mismo sistema de captación proporciona la posibilidad de una adquisición y un procesado inicial realizado simultáneamente en todos los píxeles de la imagen sensada. De esta forma, las imágenes no necesitan ser descargadas del sensor para las primeras etapas del procesado.

Esta idea de sensado-procesado conjunto extiende el concepto de *Image Sensor* (IS) al de *Smart Image Sensor* (SIS) [3]. Este tipo de sensores son también llamados *Focal-Plane Processors* (FPP) debido a que procesan imágenes en la misma capa física dónde son sensadas.

Los sistemas que incluyen un procesador de plano focal suelen emplear una arquitectura en la que el procesado de imágenes se logra siguiendo un enfoque jerárquico con dos niveles principales:

- **Procesado inicial**. Este nivel viene justo después de la adquisición. Las entradas son imágenes a resolución completa. Las tareas básicas en este nivel están dirigidas a extraer información útil del flujo de imágenes de entrada. La salida son reducidos conjuntos de datos de características de las imágenes.

- **Post-procesado**. En este punto las entradas son generalmente entidades abstractas, y las tareas van encaminadas a extraer decisiones complejas y al soporte de la toma de acciones. Estas tareas pueden involucrar algoritmos complejos con largos y complicados flujos computacionales y puede requerir mayor precisión que el procesado inicial.

En conjunto, el sistema funciona de forma similar al sistema ocular humano. Es por ello que se contemplan este tipo de soluciones como sistemas bio-inspirados.

De manera semejante a como el ojo humano capta las imágenes a través de la retina y extrae cierta información de la señal captada –como las relaciones de profundidad, las formas de los objetos o las posiciones relativas entre éstos–, el procesador de plano focal capta y realiza las primeras operaciones sobre las imágenes para extraer la información de interés.

En el sistema ocular humano, la información extraída por la retina es enviada al cerebro donde tiene lugar la interpretación de los datos. En los sistemas que introducen un procesador de plano focal, los datos suelen ser enviados a un procesador externo donde se produce el post-procesado (Figura 2.1).

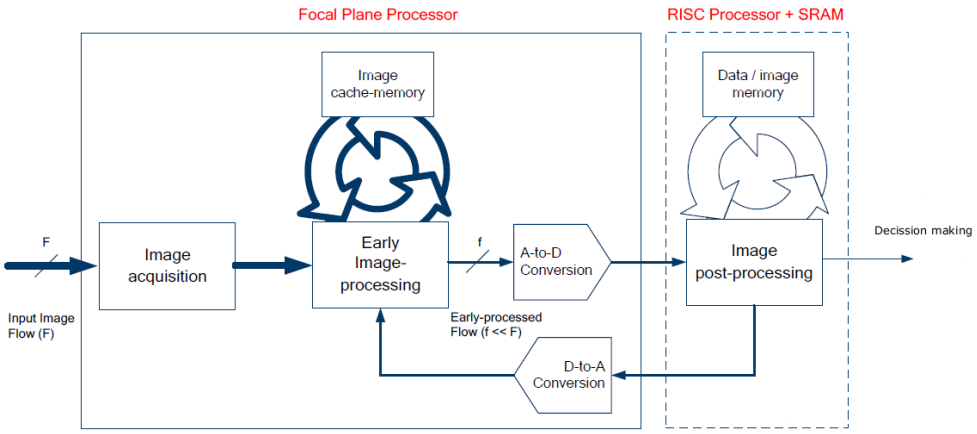


Figura 2.1. Diagrama funcional de un sistema que incluye un procesador de plano focal.

La circuitería de señal mixta que permite llevar a cabo el procesado inicial comprende sensores ópticos, memorias analógicas y digitales para los píxeles, y procesadores analógicos y binarios de tipo lineal y no lineal. Gracias a la inclusión de esta circuitería auxiliar en la capa de sensado, este tipo de sistemas son capaces de realizar operaciones de forma ultra-rápida y trabajando en paralelo.

Por supuesto, no todo son ventajas. Las imágenes captadas por los sensores del procesador de plano focal están compuestas por niveles de gris. No se dispone, en consecuencia, de la información de croma. Asimismo, las imágenes almacenadas en las memorias analógicas están sujetas a degradación con el tiempo debido a su propia naturaleza. Por ello, aunque que los procesadores de plano focal resultan muy adecuados para la realización de operaciones sobre imágenes de tipo binario, hay que tener en consideración la degradación si se realiza un uso intensivo y de forma continuada de las memorias analógicas.

La incorporación de toda la circuitería asociada a cada pixel provoca que los sensores ocupen un espacio mayor. Existe, por tanto, un límite práctico en la resolución que se puede conseguir con este tipo de cámaras. Cada pixel incorpora un sensor. Esto significa que aumentar la resolución del sistema implica un aumento del número de sensores (con su circuitería auxiliar) lo que provoca que el coste se incremente de forma exponencial. Es por ello que los sistemas que incorporan un procesador de plano focal suelen trabajar con una resolución baja cuando la tecnología de sensado se halla en un momento en que se pueden encontrar soluciones que ofrecen alrededor de la decena de megapíxel y que la tendencia es ir en aumento en este aspecto.

2.1.1. El sistema CMOS Eye-RIS v1.2

El sistema de visión AnaFocus Eye-RIS v1.2 es un sistema de visión compacto y modular que incluye todos los elementos necesarios para capturar (sensar) imágenes, mejorar las operaciones de sensado, procesar el flujo de imágenes en tiempo real, interpretar la información contenida en dicho flujo de imágenes, y soportar la toma de decisiones basándose en los datos de salida de la interpretación [4].

Figura 2.2. Sistema de visión AnaFocus Eye-RIS v1.2.

La mayoría de las tareas de procesado inicial en el procesador de plano focal del sistema Eye-RIS v1.2 son llevadas a cabo directamente en el dominio analógico operando con imágenes en niveles de gris. Esta característica tiene ciertas ventajas si se compara con el procesado puramente digital, pero también impone algunas restricciones.

El sistema Eye-RIS v1.2 consiste en los siguientes componentes:

- **Q-Eye™ Smart Image Sensor (SIS)**. El Q-Eye es el procesador de plano focal. Es capaz de realizar adquisiciones ópticas y procesar imágenes de manera altamente eficiente, en términos de velocidad y consumo de potencia. El SIS Q-Eye se programa en un código propietario desarrollado por el grupo AnaFocus que recibe el nombre de CFPP. Este lenguaje ha sido diseñado para ser muy similar al lenguaje C. Se incorporan, además, un conjunto de librerías de funciones para el procesado de imágenes.
- **Procesador RISC ALTERA NIOS® II**. Es el procesador digital que controla el flujo de ejecución y el post-procesado de las imágenes que provienen del Q-Eye. Este procesador puede ser programado en C, C++ y en código ensamblador.
- **Puertos E/S**. El Eye-RIS incluye una variedad de puertos de entrada y salida digitales, como por ejemplo SPI, UART, puertos PWM, GPIOs y USB 2.0.

La siguiente sección se centra en el Q-Eye *Smart Image Sensor* (SIS). Para más información acerca del sistema se remite al lector a [2][4].

Q-Eye Smart Image Sensor

El Q-Eye es un Sensor de Imágenes Inteligente con una resolución consistente en una matriz de 176x144 celdas. La Figura 2.3 ilustra la arquitectura interna del Q-Eye *Smart Image Sensor*.

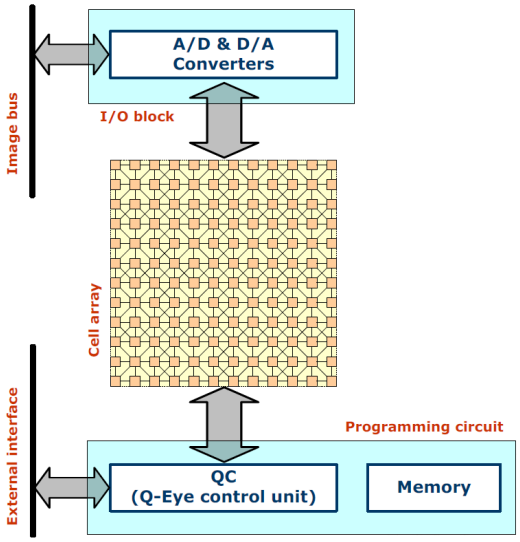


Figura 2.3. Arquitectura conceptual del chip Q-Eye.

La unidad de control del Q-Eye (QC) permite el control sobre las operaciones que se realizan en las celdas mientras que los conversores A/D y D/A son la manera natural de interactuar entre el procesador de plano focal y el procesador digital.

Cada una de las celdas comprende la circuitería que permite llevar a cabo el procesado inicial.

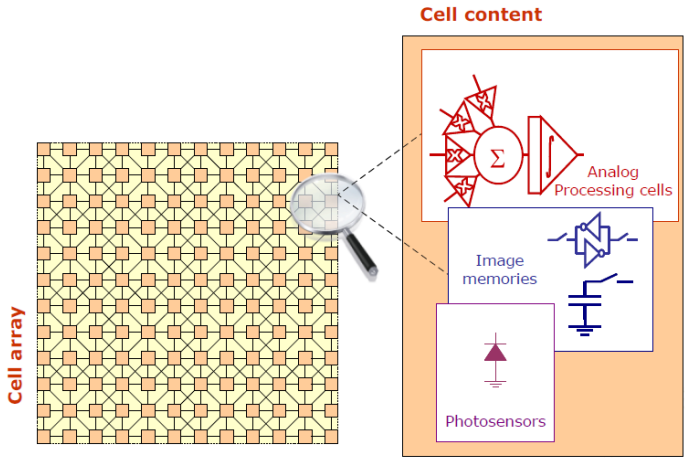


Figura 2.4. Matriz de celdas. Cada celda está acompañada de circuitería que permite realizar operaciones y almacenar los resultados directamente en la capa de sensado.

Cada celda está interconectada de diversas maneras con sus ocho celdas vecinas, permitiendo una adquisición de imágenes y operaciones de procesado espacial en tiempo real de forma altamente flexible. Cada pixel puede sensar la muestra espacial correspondiente de la imagen y procesar estos datos en una cercana interacción y cooperación con otros píxeles.

El control de los bloques contenidos en el interior de cada celda es global. Esto significa que en todas las celdas se realiza la misma operación en el mismo instante de tiempo. Existe, sin embargo, la posibilidad de desactivar de forma selectiva ciertas celdas para que la operación sólo tenga lugar en determinadas regiones de la imagen.

Cada celda incorpora memorias locales analógicas y digitales. Las Memorias Locales Analógicas (LAMs) son utilizadas para almacenar de forma temporal imágenes en niveles de gris con una resolución equivalente de 8 bits.

Las LAMs tienen un tiempo de retención limitado debido a fugas. Están concebidas para retener una imagen durante el tiempo de varias *frames* (el tiempo entre dos *frames* consecutivas) a razón de 25 fps. La Figura 2.5 muestra la variación del valor medio de una imagen almacenada en una LAM –expresada en LSBs (*Least Significant Bit*)– en el tiempo de almacenamiento –expresado en milisegundos–. El valor medio de una imagen decrece alrededor de 0.8 LSBs por cada 40 milisegundos.

Las Memorias Digitales Locales (LDMs) están pensadas para almacenar imágenes binarias. A diferencia de las LAMs, las LDMs son de tipo no volátil y pueden almacenar imágenes binarias tanto tiempo como el chip esté conectado sin degradación alguna.

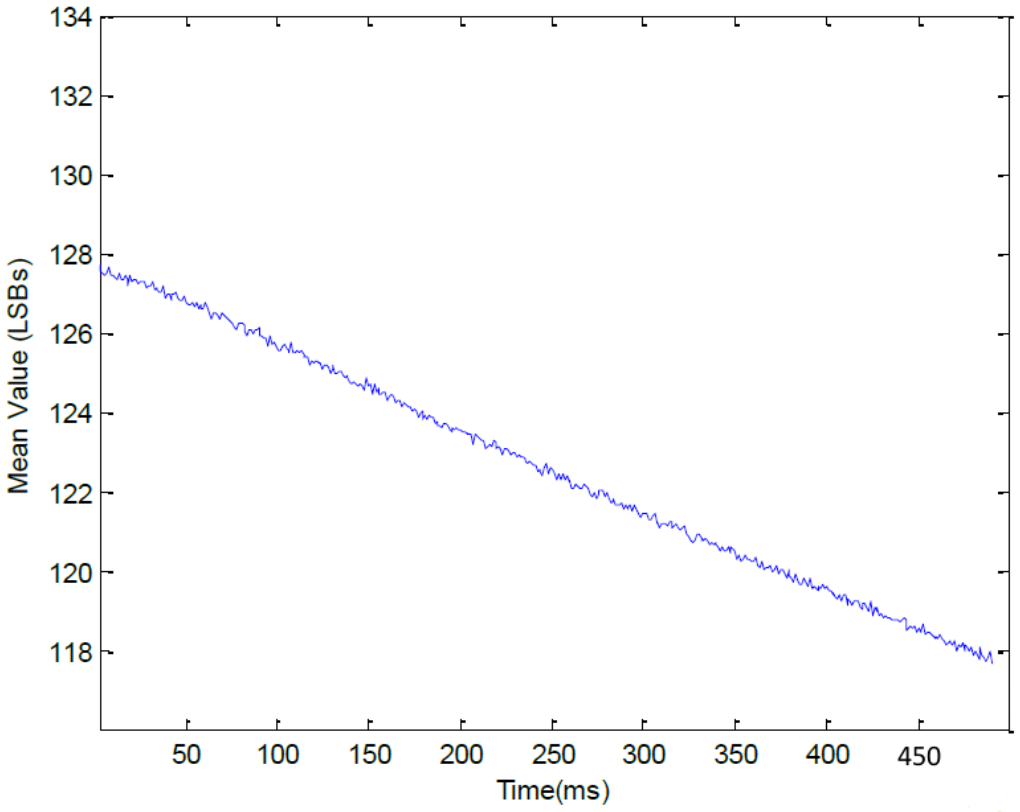


Figura 2.5. Degradación del contenido de una LAM.

2.1.2. Futuras versiones

Actualmente, se encuentra en fase de producción un nuevo modelo de dimensiones ligeramente más reducidas que ha recibido el nombre de Eye-RIS v1.3 [5]. Este modelo incorpora nuevas características así como una mejora en las librerías de la anterior versión.

Como novedades más notables se incluye un sistema que permitirá la construcción de redes neuronales, así como un procesador digital auxiliar (Figura 2.6).

Debido a que las tareas de clasificación son bastante frecuentes en una amplia variedad de aplicaciones en el campo de la *Computer Vision* (CV), se ha acondicionado el sistema con una red neuronal artificial de tipo *Multi-Layer Perceptron* (MLP). Se dispone, en consecuencia, de un sistema adaptativo que permite modelar relaciones complejas entre entradas y salidas o encontrar patrones en los datos de entrada. De esta manera se aumentan notablemente las posibilidades en el campo de la decisión y aprendizaje.

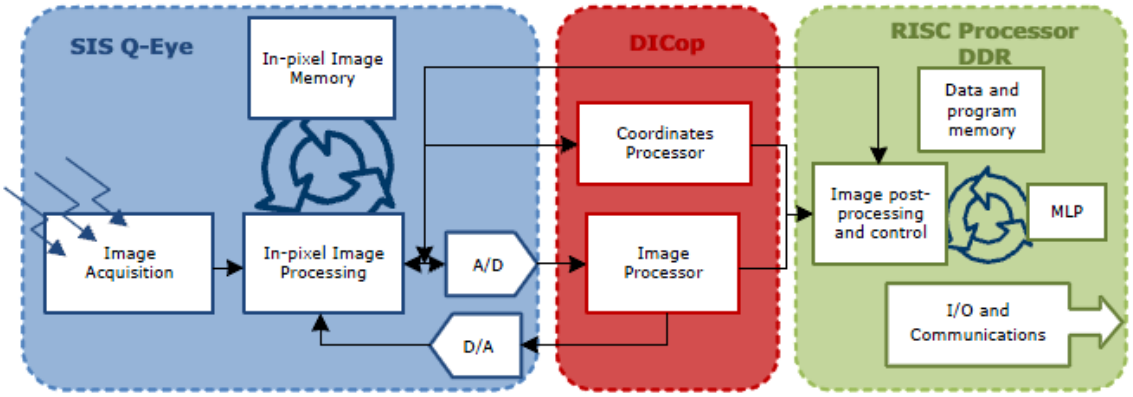


Figura 2.6. Diagrama funcional del sistema de visión Eye-RIS v1.3.

El sistema Eye-RIS v1.3 incorpora, además, un procesador denominado *Digital Image Coprocessor* (DICOP). El DICOP cuenta principalmente con dos objetivos. Uno de ellos es llevar a cabo operaciones de procesado digital de imágenes que no están cubiertas por el Q-Eye. Se añaden, por tanto, nuevas funcionalidades como son el producto entre imágenes y transformaciones geométricas (rotación y escalado de imágenes entre otras). El otro objetivo del DICOP es realizar operaciones iterativas sobre imágenes de nivel de gris que no pueden ser llevadas a cabo por el Q-Eye debido a su naturaleza analógica. Como resultado se consigue con el DICOP una liberación en la carga del procesador digital RISC.

Por último, el nuevo sistema incorpora un puerto Gigabit Ethernet con el objetivo de ofrecer la creación y gestión de redes consistentes en múltiples sistemas de visión autónomos.

2.2. Algoritmos para FPP

Uno de los principales e inherentes problemas de trabajar directamente en el ámbito analógico es la degradación que se produce en las memorias analógicas en el procesador de plano focal.

La solución propuesta en [6] trata de minimizar el uso de las memorias analógicas con el fin de evitar los efectos de este comportamiento. Esta solución parece ser la manera más natural y eficiente de trabajar con este tipo de cámaras. La idea principal consiste en descomponer la imagen en nivel de grises en un formato binario de tal forma que se pueda almacenar en las memorias digitales que no poseen el problema de la degradación.

Este proceso debe ser reversible para poder recuperar la imagen inicial o poder tratar por separado las imágenes resultado de la descomposición y poder recomponer una imagen en escala de grises después del procesado.

En este sentido, resulta práctico el tratamiento que se puede realizar de las imágenes con una descomposición en planos de bit. Esta descomposición ha sido utilizada en [7] [8] y es una buena manera de trabajar con cámaras que incluyen un procesador de plano focal.

2.2.1. Descomposición en planos de bit

La transición de una imagen en escala de grises a un conjunto de imágenes binarias que permitan extender las operaciones morfológicas básicas no resulta sencilla. Esto es debido, en gran parte, a que las operaciones morfológicas no son operaciones lineales.

Sin embargo, el hacerlo puede comportar ciertas ventajas. Por un lado, los operadores no lineales de señales multi-nivel se pueden ver reducidos a análisis más simples de señales binarias. Por otro lado, la descomposición permite que los cálculos puedan ser realizados en paralelo en los diferentes niveles de descomposición.

Sea $I(x,y)$ una imagen en escala de grises con N niveles posibles. Esto es, cada píxel de la imagen está codificado con $b = \lceil \log_2 N \rceil$ bits.

La descomposición en planos de bits de la imagen $I(x,y)$ se expresa matemáticamente como:

$$p_k(x,y) = \left\lfloor \frac{I(x,y)}{2^k} \right\rfloor mod\ 2$$

Donde $k = 0,1,\dots,b-1$ y p_k es el plano de bits de nivel k.

La recomposición se logra mediante:

$$I(x,y) = \sum_{k=0}^{b-1} p_k(x,y) \cdot 2^k$$

En la práctica, la descomposición consiste en separar la imagen en escala de grises en cada uno de sus niveles de bit. Esto es, un plano para los bits de mayor significancia (MSB), otro para el siguiente y así hasta llegar al bit de menor peso (LSB). Por tanto, la imagen en nivel de gris se

descompone en tantos planos como número de bits tenga cada píxel. La Figura 2.7 ilustra una representación gráfica de lo comentado, mientras que la Tabla 2.1 muestra un ejemplo numérico.

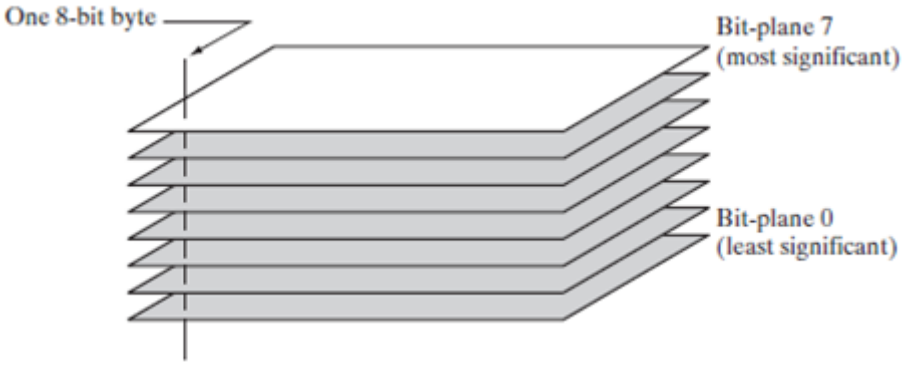


Figura 2.7. Descomposición en planos de bit para $N = 256$ (8 bits).

Valores	125	232	7	85		
	0	1	0	0	Plano de bits 7	MSB
	1	1	0	1	Plano de bits 6	
	1	1	0	0	Plano de bits 5	
	1	0	0	1	Plano de bits 4	
	1	1	0	0	Plano de bits 3	
	1	0	1	1	Plano de bits 2	
	0	0	1	0	Plano de bits 1	
	1	0	1	1	Plano de bits 0	LSB

Tabla 2.1. Descomposición en planos de bit.

Esta descomposición resulta muy práctica por el hecho de que existe una sinergia con la formulación matemática en base 2. Esto comporta una tremenda ventaja en términos de implementación debido a que se trabaja directamente en la base natural de los computadores.

Es posible, de hecho, realizar la descomposición mediante operaciones de desplazamiento de bits. La recomposición se realiza con simples operaciones lógicas de adición de los diferentes planos con los pesos adecuados. La asignación de los pesos se lleva a cabo por desplazamientos de bits.

La Figura 2.8 muestra el resultado de la descomposición en planos de bit de una imagen.

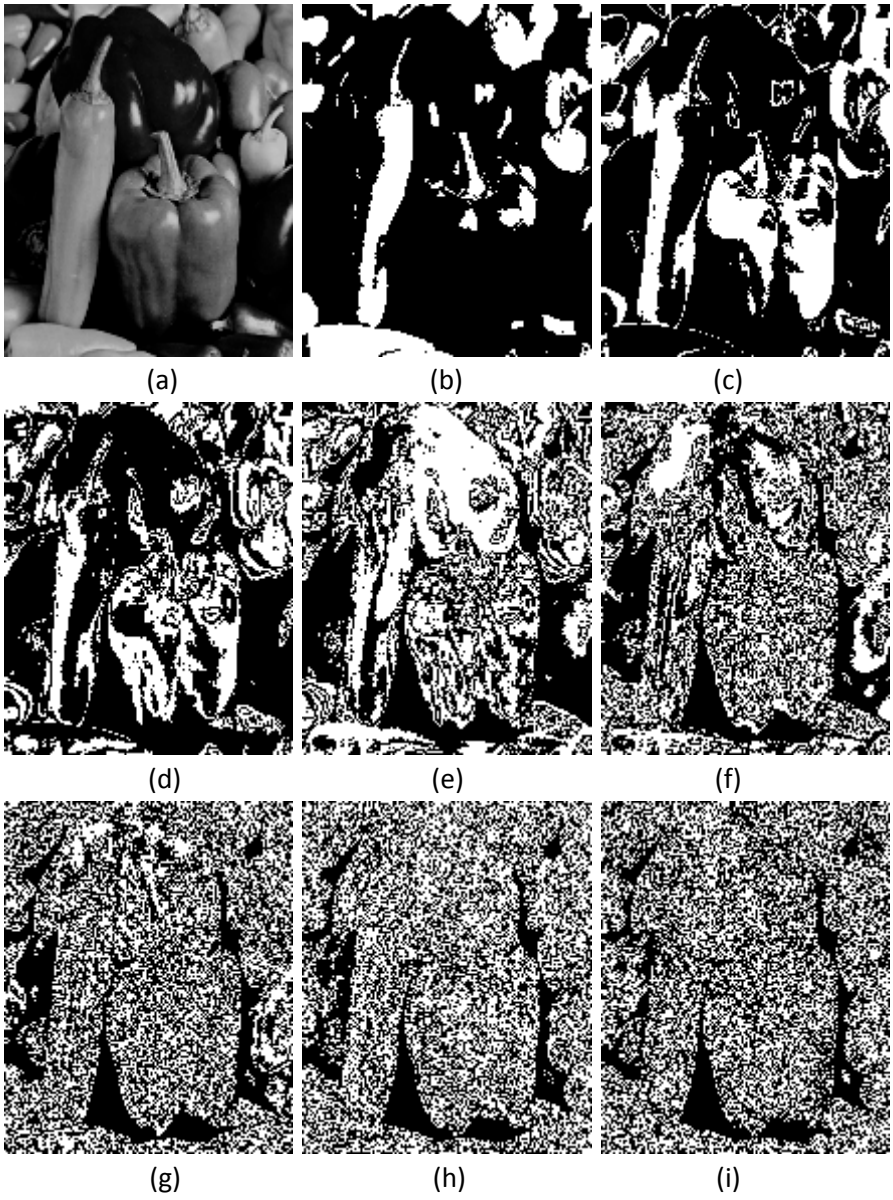


Figura 2.8. Descomposición en planos de bit. (a) Imagen original. (b) *Bit plane* 7. (c) *Bit plane* 6. (d) *Bit plane* 5. (e) *Bit plane* 4. (f) *Bit plane* 3. (g) *Bit plane* 2. (h) *Bit plane* 1. (i) *Bit plane* 0.

Resulta interesante remarcar el hecho que, en este tipo de descomposición, cada plano tiene un peso asociado. En otras palabras, la descomposición está dispuesta de manera jerárquica. En las imágenes binarias resultantes de la descomposición se encuentran en los planos de mayor peso los rasgos más burdos mientras que, según se va bajando de plano, se van encontrando los detalles y el ruido.

A partir de esta descomposición se han diseñado e implementado algoritmos para la adaptación de operaciones morfológicas básicas sobre imágenes en nivel de gris tratadas con procesadores de plano focal, así como para el cálculo del máximo y mínimo de una imagen [7] y el cómputo de su histograma [8].

2.3. *Human Computer Interaction*

El concepto de *Human Computer Interaction* (HCI) es muy amplio y abarca gran cantidad de disciplinas [9]. Explicar concienzudamente el estado del arte de la HCI es por ello una tarea compleja y extensa. Aquí se pretende realizar un esbozo general de los esfuerzos realizados hasta el momento en este campo citando numerosas referencias para que el lector interesado pueda profundizar en la materia a partir de este escrito.

A modo de introducción se van a describir las diferentes taxonomías de la tecnología HCI para posteriormente profundizar sobre cada una de las categorías enfocando el escrito hacia el reconocimiento de gestos de la mano.

Las tecnologías HCI existentes se pueden categorizar básicamente por el sentido relativo para el que el dispositivo está diseñado pero también a través de su configuración. De hecho, cualquier sistema se puede definir generalmente por el número y la diversidad de entradas y salidas que proporciona.

En la terminología empleada en los sistemas HCI, cada uno de los canales de comunicación que permite a los usuarios interactuar con el computador recibe el nombre de modalidad. Así, se habla de sistemas unimodales cuando éstos están basados en una única modalidad mientras que se utiliza el término multimodal para designar a los sistemas que utilizan la combinación de múltiples modalidades.

En los sistemas HCI de Múltiples Modalidades (MMHCI), una de las combinaciones más comúnmente soportadas es la de los gestos y habla. Un ejemplo clásico es el sistema denominado *Put That There* [10]. Este sistema permitía a un usuario mover un objeto a una nueva localización en un mapa situado en pantalla diciendo "*put that there*" mientras se apuntaba con un dedo al objeto y luego apuntando al lugar de destino deseado.

Un sistema MMHCI ideal debería contener una combinación de modalidades simples que interactúan correlativamente. Sin embargo, los límites prácticos y los problemas abiertos en cada modalidad imponen limitaciones a la fusión de los diferentes canales. A pesar del gran progreso realizado en MMHCI, en la mayoría de los sistemas multimodales las modalidades son aún tratadas separadamente y únicamente al final, los resultados se combinan de manera conjunta [11].

Los sistemas basados en una única modalidad pueden ser divididos principalmente en tres categorías basándose en la naturaleza del sentido relativo para el que están diseñados: audición (sistemas *audio-based*), tacto (*sensor-based*), y visión (*visual-based*).

El área concerniente a *audio-based* trata con información adquirida a través de señales de audio. Aunque la naturaleza de estas señales no suele ser tan variable como la de las señales visuales, la información obtenida de las señales de audio puede ser de mayor confianza, útil y, en algunos casos, la única manera de proveerse de información.
Las áreas más usuales de desarrollo en esta sección son el reconocimiento del habla, el reconocimiento del interlocutor, el análisis de las emociones audibles y la detección de signos o ruidos realizados por el ser humano (suspiros, risas, lloro, entre otros).

Las aplicaciones HCI *sensor-based* se basan en la utilización de un sensor físico entre el usuario y la máquina para proporcionar la interacción (los sensores de presión o los sensores hápticos suelen ser los más comunes).

La interacción humano-computadora *visual-based* es probablemente la más extendida en el área de desarrollo HCI. Los principales campos de actuación en esta categoría son el análisis de la expresión facial, el *tracking* del movimiento del cuerpo, el reconocimiento de gestos y la detección de la mirada (*tracking* del movimiento de los ojos).

Aunque el objetivo de cada área difiere según las aplicaciones, se puede desprender un concepto general de cada una. El análisis de la expresión facial generalmente trata con el reconocimiento de emociones de forma visual [12]. La detección de la mirada es usada principalmente para el seguimiento de la atención o focalización del usuario. También existen sistemas de seguimiento ocular para ayudar en las minusvalías en las cuales el seguimiento ocular juega un papel principal en escenarios de comandos y acciones, como por ejemplo el movimiento de puntero y el pestañeo para clicar [13].

El *tracking* del movimiento del cuerpo y el reconocimiento de gestos son principalmente usados para la interacción directa entre humano y computadora en un escenario de comandos y acciones.

El caso particular del reconocimiento de gestos de la mano es un área muy atractiva por la gran expresividad que se puede conseguir con estas extremidades. Por este motivo se han dedicado grandes esfuerzos en este ámbito [14]. Aún así, multitud de problemas siguen abiertos y las soluciones adoptadas son muy variadas.

De forma general, en las aplicaciones de detección de gestos se encuentran involucradas tres fases: la segmentación espacial y/o temporal de la mano que realiza el movimiento, la extracción de características y la clasificación del gesto realizado.

La segmentación espacial en el contexto del reconocimiento de gestos es la tarea de determinar, en una secuencia de video, la localización de la mano que realiza el gesto. La segmentación temporal se encarga de delimitar los instantes de tiempo en que el gesto empieza y termina. En muchos de los métodos existentes de reconocimiento de gestos se asume una segmentación temporal o espacial conocida e incluso ambos casos.

Una segmentación temporal conocida se puede lograr imponiendo un gesto específico como disparador de la detección de la mano [15]. De este modo el gesto empieza o finaliza con una posición conocida a priori. En [16] se presenta un sistema que activa la clasificación de la postura de la mano cuando ésta permanece quieta por un tiempo.

Para facilitar la segmentación espacial, algunos sistemas hacen uso de guantes de un color complementario al color de fondo o evitan el uso de fondos complejos [17].

En [18] se propone la utilización de una *range-camera* o *Z-cam*. Este tipo de cámaras, en lugar de obtener imágenes dónde cada píxel representa el nivel de intensidad lumínica captado, obtienen imágenes dónde cada píxel representa la distancia de los objetos a la cámara. Así, objetos cercanos son representados con niveles altos de intensidad mientras que los objetos lejanos

producen píxeles con un valor inferior. De este modo, se puede aprovechar el hecho de que normalmente la mano que realiza los gestos para la interacción está avanzada con respecto a los demás objetos por lo que tendrá unos valores de intensidad más altos en la imagen producida por la *Z-cam*.

La opción probablemente más extendida en la segmentación espacial de la mano está basada en la combinación de una transformación del espacio de color y un mapa del flujo de movimiento. La idea principal es cambiar del espacio RGB a un espacio de color –típicamente HSV [15] [19] o YCbCr [20]– donde las zonas del color de la piel puedan ser detectadas de forma más fiable. Esta información es combinada con las regiones en que se ha detectado movimiento para determinar la posición de la mano.

Sin embargo, la detección de la mano basada en el color de la piel no soluciona de forma unívoca el problema. En ocasiones puede ser poco fiable por la dificultad de distinguir la mano de otros objetos de color similar a la piel o por la sensibilidad del método a las condiciones de iluminación. Por todo esto, en los sistemas basados en visión es una tarea compleja localizar la mano que realiza el gesto con absoluta precisión.

Una vez obtenida la localización de la región de interés se debe proceder a la extracción de una serie de características que permitan identificar el gesto realizado. Para ello es necesaria la construcción de un modelo de la mano.

Hay que tener en cuenta que en un entorno de interacción entre humano y computadora se desea utilizar la mano para realizar tareas que pueden imitar tanto el uso natural de la mano como manipulador, como su uso en la comunicación del humano con la máquina [21].

La calidad de la interacción está directamente relacionada, por tanto, con el correcto modelado de los gestos de la mano. Para algunas aplicaciones, un modelo tosco y simple puede ser suficiente. Sin embargo, si el propósito es una interacción similar a la natural, un modelo más complejo debe ser tenido en cuenta.

Existe, no obstante, un compromiso entre la complejidad del modelo utilizado y el tiempo de respuesta del sistema.

Los métodos para reconocimiento de gestos de la mano en el contexto de aplicaciones de visión se pueden agrupar en dos categorías principales por lo que al modelado de la mano se refiere: métodos basados en modelos de apariencia de la mano y métodos basados en modelos 3D.

Los modelos 3D pueden describir de forma exacta el movimiento de la mano y su forma, pero la mayoría tienen un coste computacional elevado para un uso en tiempo real. Además, la construcción de modelos 3D de la mano suele requerir el uso de múltiples cámaras para obtener distintos puntos de visión (lo más común es utilizar dos cámaras en lo que se conoce como configuración estéreo).

Algunos métodos de aparición más reciente obtienen modelos 3D con apariencia 2D como el PCA-ICA [19] en el que se considera un modelo de la mano con 15 juntas (ver Figura 2.9).

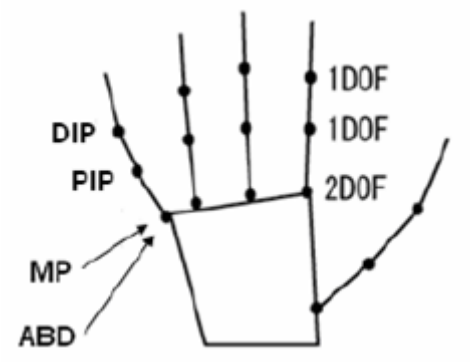


Figura 2.9. Juntas de la mano. Cada dedo tiene cuatro grados de libertad. Dos de ellos son para la junta (MP) y la abducción (ABD) metacarpofalángicas, y las otras dos son para la junta interfalángica proximal (PIP) y la junta interfalángica distal (DIP).

Los modelos basados en la apariencia aportan como principal ventaja un menor coste computacional. Estos modelos tratan de obtener una serie de características que determinan la forma en que está posicionada la mano (el centro de gravedad de la mano, los ejes mayor y menor de la elipse que contiene la mano, la silueta, entre otros).

En [18] se presenta un método de reconocimiento de gestos que utiliza un modelo basado en la apariencia fundamentado en el concepto de distancia geodésica. En [22] y [23] se modela la mano a partir de histogramas de orientación que permiten decidir la posición de la mano.

Gracias a las características extraídas, ya sea con modelos 3D o con modelos basados en apariencia, se puede proceder a la clasificación del gesto. En este campo se encuentran dos aproximaciones muy usadas que han demostrado una gran eficacia. Por un lado se encuentra la clasificación basada en redes neuronales y por el otro la clasificación basada en *Hidden Markov Models* (HMM) [24].

Ambos métodos utilizan sesiones grabadas previamente para generar modelos. Es lo que se denomina aprendizaje del sistema. Para la posterior clasificación de un gesto, se calcula la distancia entre la secuencia captada y los modelos generados y se determina el gesto realizado escogiendo el modelo que ha dado la distancia mínima.

En [25] se hace uso de una *Time-Delay Neural Network* (TDNN) con un algoritmo de aprendizaje basado en *back propagation* para reconocer movimientos dinámicos del ASL (*American Sign Language*).

Los *Hidden Markov Models* son procesos estocásticos que pueden ser utilizados para modelar cualquier serie temporal que pueda ser asumida como un proceso de Markov de primer orden. Por lo tanto, los HMMs suelen emplearse principalmente para el reconocimiento de gestos dinámicos. En la clasificación mediante HMMs cada gesto tiene un modelo oculto de Markov y cada modelo define la secuencia temporal del gesto [20].

Los HMMs han sido ampliamente utilizados en el reconocimiento de lenguajes de signos debido a que han probado ser una buena solución para aplicaciones en que el vocabulario a reconocer es extenso. En los lenguajes de signos existen generalmente dos niveles de expresión: el nivel de

palabra (donde cada signo representa una palabra) y el nivel dactilológico (que es una representación manual del abecedario y es utilizado cuando no existe un signo para el elemento que se desea expresar). Con los HMMs el reconocimiento dactilológico de letras se puede extrapolar al reconocimiento de palabras de forma natural utilizando teoría de grafos. Se concatenan los HMMs mediante una red gramática en que la detección de varias letras permite el reconocimiento de las palabras [20].

En [26] se presenta un método que hace uso de los modelos ocultos de Markov para la clasificación de gestos dactilológicos. La innovación de este método es la introducción de un sistema que combina la clasificación y la segmentación utilizando las estimaciones de probabilidad del algoritmo de Viterbi.

Un método de clasificación de aparición más reciente es el presentado en [27] y [28]. Este método también sigue la línea de realizar de forma conjunta y colaborativa la segmentación y la clasificación del gesto. En lugar de asumir que se tiene una detección precisa y correcta de la mano, se realiza la hipótesis de que se tienen una serie de candidatos de la posición de la mano para cada *frame*. El núcleo del método es un algoritmo denominado *Dynamic Space-Time Warping* (DSTW) que permite alinear un par de secuencias (una a probar y la otra un modelo) tanto en espacio como en tiempo.

El algoritmo DSTW está basado en *Dynamic Time Warping* (DTW) que alinea en tiempo dos secuencias y calcula una puntuación de coincidencia, que es usada para clasificar la secuencia de entrada. No obstante, en DTW no se considera la posibilidad de múltiples candidatos, por lo que se asume que puede ser extraído de forma correcta un vector de características de cada *frame*.

Los resultados experimentales demuestran que DSTW consigue una mejora situada entre un 11% y un 21% en la precisión de la clasificación del gesto con respecto al método DTW [27].

Las *Finite State Machines* (FSM) constituyen otra forma de representar la evolución temporal de los gestos [17] [29]. Cada FSM representa un sistema secuencial en que el movimiento relativo de la mano determina el paso al siguiente estado. De esta forma, con un conjunto de FSM se pueden identificar una serie de movimientos.

En [30] se presenta una clasificación basada en un algoritmo denominado *k-mediod clustering* con una distancia basada en el contexto de la forma. La base del algoritmo es la construcción de una estructura en árbol. La parte alta del árbol proporciona una detección de la mano de forma general mientras que las ramas individuales del árbol clasifican las formas válidas a uno de los *clusters* predeterminados. Se consigue un ratio de 99.8% en la detección de la mano y un 97.4% en la clasificación.

La mayoría de los métodos aquí descritos son difícilmente adaptables a la cámara de trabajo debido a las limitaciones para operar en tiempo real cuando se ven involucrados algoritmos complejos que deben ser llevados a cabo en el procesador digital.

La segmentación espacial por color de la piel resulta imposible ya que la cámara sólo capta imágenes en nivel de gris por lo que habrá que lidiar con este problema de una forma alternativa.

Capítulo 3

Estudio de gestos a detectar

En HCI, las soluciones deben ser implementadas como sistemas de información *Human Centered*. Uno de los problemas más importantes es cómo lograr un sinergismo entre el hombre y la máquina. El término *Human-Centered* se usa para enfatizar el hecho de que todo el sistema está diseñado con el usuario humano en mente.

Es por ello que esta doctrina es multidisciplinar y por tanto deben ser incluidos multitud de aspectos psicológicos. Antes de comenzar a desarrollar una serie de algoritmos que permitan identificar los gestos, es importante disponer de un estudio de usabilidad del sistema en términos generales.

3.1. Estudio de usabilidad

En [9] se distinguen tres niveles diferentes en la actividad del usuario para el ámbito de la interacción humano-computadora: físico, cognitivo y afectivo. El aspecto físico determina las mecánicas de la interacción mientras que los aspectos cognitivos tienen que ver con la manera en que los usuarios pueden entender el sistema e interactuar con él. El aspecto afectivo es un aspecto mucho más reciente e intenta no sólo hacer de la interacción una experiencia agradable para el usuario, sino también afectarlo en una manera que haga que siga utilizando la máquina por el cambio de actitudes y emociones hacia el usuario.

El primer y el segundo nivel tienden a determinar qué gestos deberían ser implementados mientras que el tercero se centra en la satisfacción que obtiene el usuario debido a los cambios de comportamiento de la máquina.

Es importante cumplir, en la medida de lo posible, con estos tres niveles para lograr una interacción satisfactoria. Obviamente, una implementación robusta es una condición para el correcto funcionamiento del sistema. Pero también es determinante que el usuario vea la interacción con el sistema de una forma natural. Es por ello que se ha llevado a cabo un experimento de usabilidad realizado por la Universitat Jaume I en colaboración con el Centro Clínico de Psicología Previ de Valencia para determinar qué gestos deben ser implementados [31].

En el estudio se elabora una lista de los movimientos más habituales realizados por un conjunto de personas a las que se les da una instrucción de gesto a realizar (por ejemplo coger un elemento). Posteriormente, los movimientos deben ser relacionados con las instrucciones por evaluadores independientes.

"Si el gesto elaborado por un sujeto experimental es fácilmente reconocible por los evaluadores independientes, también será fácilmente memorizable, pues el hecho de que se produzca la familiaridad con el gesto y su reconocimiento ya pone en marcha los mecanismos de memoria y aprendizaje" [32].

3.2. Conjunto de gestos

De los movimientos derivados del estudio de usabilidad se han elegido un conjunto de gestos conformados por movimientos tanto estáticos como dinámicos para tratar con su integración en un sistema HCI.

En las figuras Figura 3.1 hasta Figura 3.10 se ilustran los gestos finalmente implementados así como una breve descripción del movimiento.

Figura 3.1. Enumerar 0. Puño cerrado.

Figura 3.2. Enumerar 1. Dedo índice extendido.

Figura 3.3. Enumerar 2. Dedos índice y corazón extendidos.

Figura 3.4. Enumerar 3. Los tres dedos centrales extendidos.

Figura 3.5. Enumerar 4. Retirando el dedo pulgar.

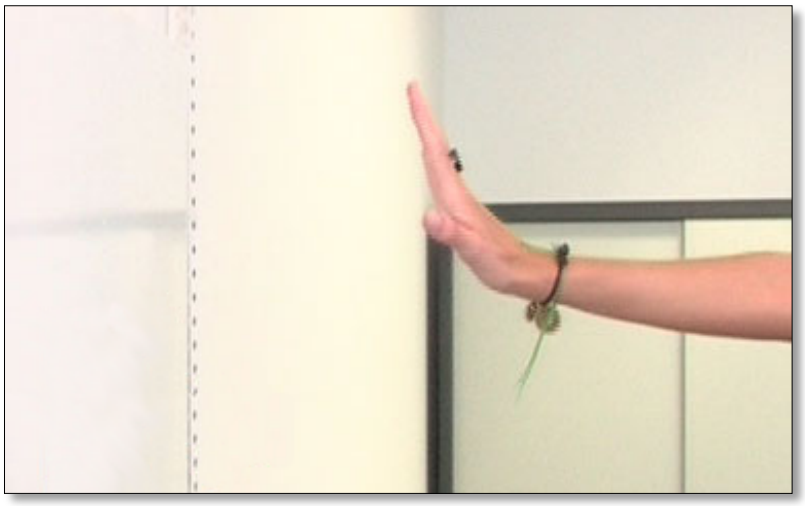


Figura 3.6. Enumerar 5. Todos los dedos extendidos.

Figura 3.7. Abrir menú. Movimiento dinámico. Aparición de la mano por la parte inferior para ir ascendiendo hasta su salida por la parte superior.

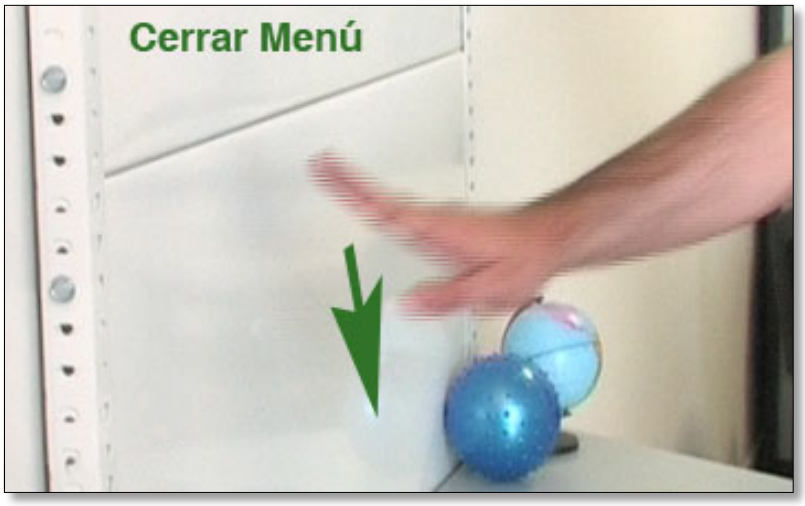


Figura 3.8. Cerrar menú. Movimiento dinámico. Aparición de la mano por la parte superior para descender hasta desaparecer por la parte inferior.

Figura 3.9. Movimiento hacia la derecha. Movimiento dinámico. Entrada de la mano por la parte izquierda para salir por la derecha.

Figura 3.10. Movimiento hacia la izquierda. Movimiento dinámico. Aparición de la mano por la derecha para desaparecer por la parte izquierda.

Capítulo 4

Descripción del sistema

4.1. Funcionamiento global

El funcionamiento general del sistema viene descrito por el esquema de la Figura 4.1.

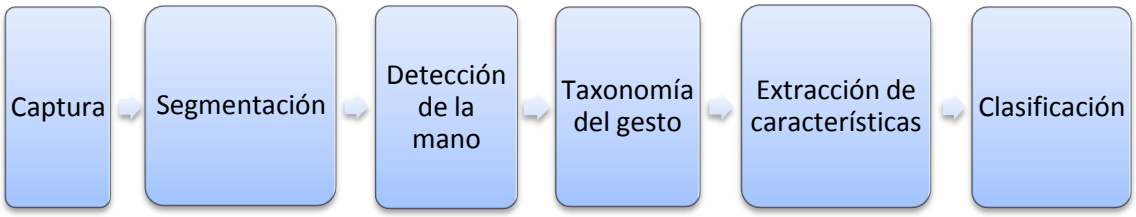


Figura 4.1. Esquema global de funcionamiento del sistema.

En términos generales, se pueden definir tres fases. La primera fase comprende los tres primeros bloques del esquema y tiene como objetivo la localización de la mano. Para ello es preciso determinar la presencia de ésta y obtener una imagen que delimite la zona donde se encuentra. Esta primera fase está desarrollada de forma íntegra en el procesador de plano focal.

La segunda fase comprende los bloques tercero y cuarto del esquema. Con la posición de la mano delimitada, se procede a determinar la taxonomía del movimiento. Este proceso tiene como objetivo la distinción entre movimiento estático y dinámico. Para agilizar el funcionamiento del sistema, la extracción de características se realiza de forma diferente según esta clasificación. Así, para movimientos dinámicos el algoritmo se centrará en la evolución que sigue la posición de la mano mientras que para movimientos estáticos interesa realizar, además, el conteo de los dedos. La taxonomía del gesto involucra el cálculo de momentos de primer y segundo orden por lo que este proceso está desarrollado enteramente en el procesador digital. La extracción de características, por su parte, requiere un tratamiento inicial de las imágenes –que es llevado a cabo en el procesador de plano focal– y un análisis de la información extraída –que debe ser realizado en el procesador digital–.

Finalmente, en la tercera fase se produce la clasificación del gesto realizado. El resultado, en este punto, es uno de los gestos del conjunto definido en la sección 3.2. Esta última fase implica una decisión del movimiento que es realizada en el procesador digital.

4.2. Captura e inicialización

La captura de las imágenes constituye la primera etapa en cualquier aplicación basada en *Computer Vision* (CV). El flujo de imágenes captadas conforma la entrada del sistema.

En esta primera etapa, además, se establece el valor de algunos parámetros que van a determinar el funcionamiento posterior del sistema. La inicialización y calibrado del sistema constituye las etapas:

- Detección de movimiento.
- Ajuste del tiempo de exposición.
- Inicialización de la imagen de fondo.
- Construcción de una máscara para delimitar la región de interés (ROI).

En un primer momento se detecta la presencia de movimiento en una zona específica para determinar si hay algún usuario que desee utilizar el sistema. La información del movimiento se extrae a partir de la diferencia en valor absoluto entre la imagen actual y una imagen capturada un cierto número de *frames* antes.

Sobre la imagen diferencia se aplica un umbral para eliminar la aparición de ruido. A la imagen resultante se le aplica una operación de cierre morfológico y una erosión con la finalidad de eliminar ruidos. Finalmente, se calcula el *BoundingBox* de la imagen resultado de estos procesos para determinar la zona en que se ha producido movimiento.

En términos matemáticos, con $\Re$ denotando la zona de detección de movimiento, se considera que hay un usuario que desea utilizar el sistema si:

$$BoundingBox(\varepsilon_b\left\{\varphi_b\left\{Umbral\left(abs\left(g_{numberImage}-g_{lastImage}\right)\right)\right\}\right\}) \subseteq \Re$$

Dónde la operación de $BoundingBox(\cdot)$ da como resultado el rectángulo que circunscribe los puntos activos de la entrada, ε_b y φ_b son las operaciones de erosión y cierre con elemento estructurante b respectivamente, $abs(\cdot)$ denota la operación de valor absoluto y $g_{numberImage}$ y $g_{lastImage}$ son la imagen actual y una imagen capturada un cierto número de *frames* antes.

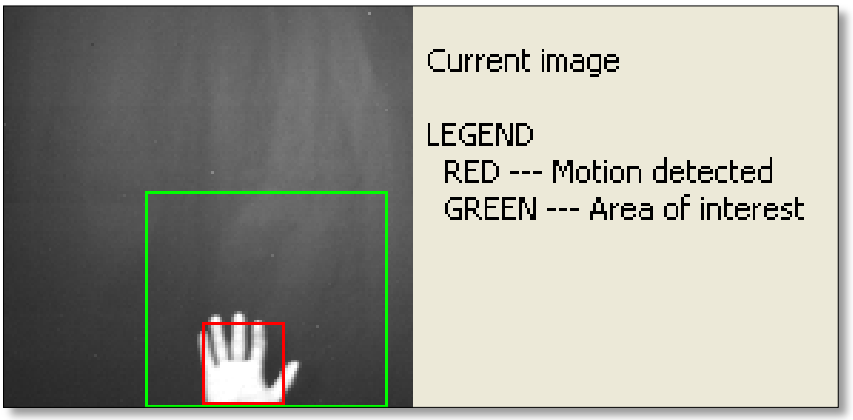


Figura 4.2. *Bounding Box* del movimiento detectado (en rojo) y zona de detección de movimiento (en verde).

Dado que se busca una estimación general del movimiento sin necesidad de un alto detalle, el elemento estructurante elegido para estas operaciones es cuadrado y de tamaño 5x5.

Se inicializa, así mismo, el valor de la imagen de fondo que se utilizará en etapas posteriores. Esta inicialización tiene lugar cuando el movimiento detectado es nulo o cuando se produce en prácticamente la totalidad de la zona de captación de la cámara.

Se ajusta el tiempo de exposición de la cámara. Este parámetro es importante ya que determina qué cantidad de luz se deja pasar y es dependiente de la iluminación en el momento de uso de la aplicación.

Así mismo, con el objetivo de delimitar la zona de interés (ROI), se construye una máscara a partir de la zona donde se ha detectado el movimiento. A partir de este punto, todas las operaciones tienen en consideración únicamente la ROI calculada obviando el resto y agilizando así los cálculos al tratarse de una zona más pequeña.

Todas estas operaciones tienen lugar en el procesador de plano focal. Se trata, por tanto, de una solución efectiva en términos de consumo de tiempo.

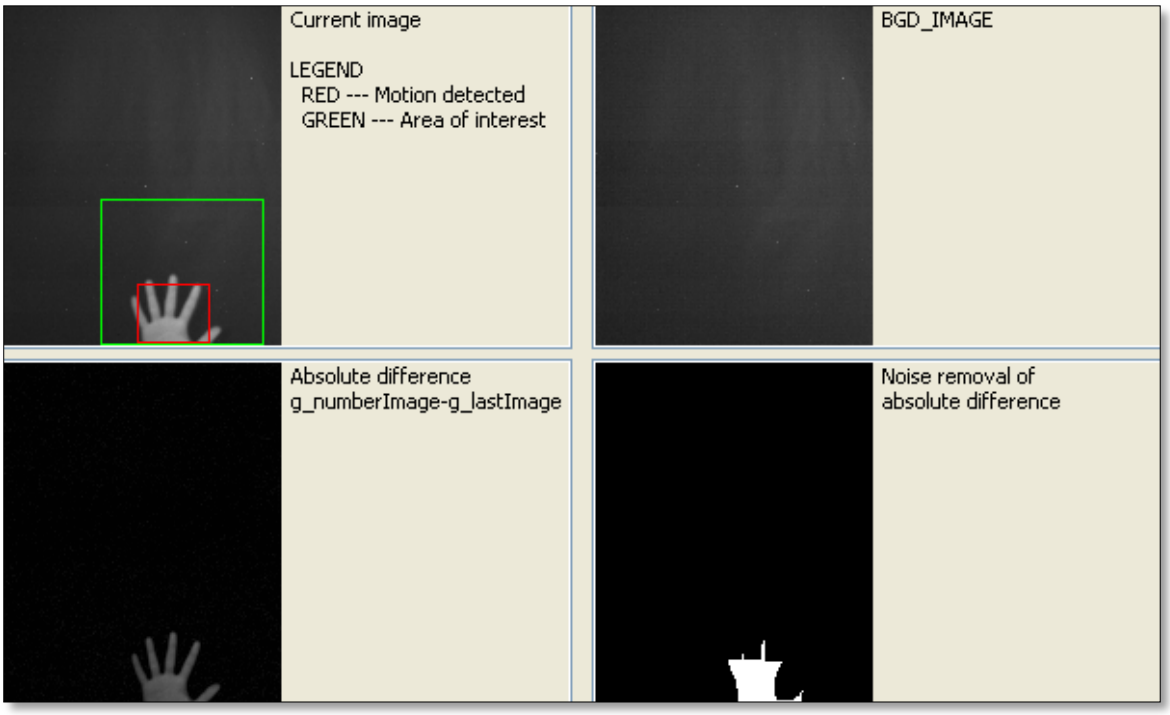


Figura 4.3. Proceso de inicialización de la imagen de fondo así como de la detección de movimiento.

4.3. Segmentación

4.3.1. Segmentación espacial

El objetivo de la segmentación espacial es extraer de la imagen de nivel de gris capturada por el sensor la región correspondiente a la mano. El resultado de la segmentación consiste en una

imagen binaria donde uno de los dos niveles representa la región de interés, es decir, la mano, y el otro nivel representa el fondo. Esto permitirá determinar qué parte de la imagen capturada se analiza.

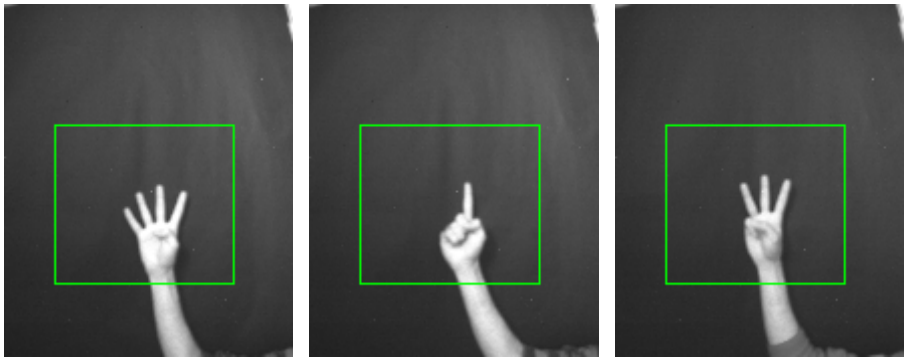
En un escenario real, la segmentación espacial es un proceso complejo influido por una gran cantidad de factores (fondos complejos, cambios de iluminación, aparición de sombras, etc.).
Los métodos existentes que tratan con estos factores en imágenes en nivel de gris suelen ser algoritmos complejos y costosos computacionalmente por lo que no resultan idóneos para ser implementados en procesadores de plano focal [33].

Para desacoplar en lo posible el problema de la segmentación –que no es el objetivo principal de este trabajo– del de reconocimiento de gestos, se utilizan normalmente una serie de restricciones: en el fondo (*background*), en el usuario o en las imágenes.

Las restricciones en el *background* son las más usuales: un fondo uniforme y oscuro simplifica la segmentación enormemente. Restricciones adicionales en el usuario simplifican el problema de la localización de la mano (requerimientos de llevar prendas oscuras de manga larga, por ejemplo). También simplifican el problema de la localización las restricciones sobre las imágenes (usando cámaras focalizadas en la mano por ejemplo). Aunque desde el punto de vista computacional estos métodos son más fáciles de implementar tienden a reducir la naturalidad de la interacción.

Se ha optado por obviar en lo posible los problemas presentados por la segmentación espacial. Por ello, los gestos se han capturado sobre un fondo oscuro, con la cámara focalizada en el brazo y con una iluminación a lo largo del tiempo aproximadamente constante. Este escenario es conocido con el nombre de entorno de laboratorio. Este entorno permite realizar la segmentación mediante operaciones de binarización.

En la Figura 4.4 se pueden observar algunos ejemplos del aspecto de las imágenes capturadas.



(a) (b) (c)

Figura 4.4. Ejemplos de imágenes capturadas. En verde se denota la ROI.

La imagen capturada es sometida a una operación de sustracción del fondo. Esta operación consiste en restar a la imagen de análisis una imagen almacenada en memoria e identificada como fondo. Las zonas dónde no ha habido actividad resultan en zonas oscuras en la imagen

diferencia (en las zonas dónde no ha habido actividad no se han producido cambios, por lo que los píxeles de la imagen captada en estas zonas tienen un valor muy similar al de la imagen de fondo).

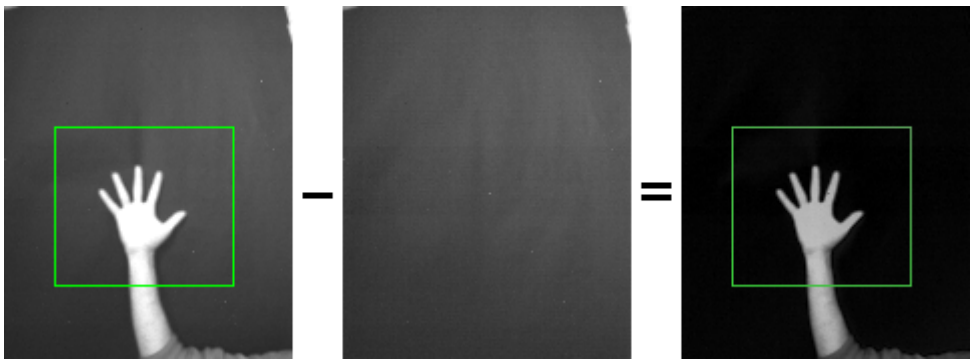


Figura 4.5. Sustracción del fondo.

Debido a que las imágenes utilizadas como muestra siguen un patrón con una zona de interés (la figura de la mano) con niveles de intensidad elevados y un fondo con valores de intensidad muy bajos (fondo oscuro), el hecho de restar el fondo no reporta grandes beneficios con respecto a no restarlo pero permitirá una generalización del método y una mejor segmentación en casos donde pueda haber cierta ambigüedad (como por ejemplo la presencia de algún artefacto en el fondo de iluminación elevada siempre y cuando éste no se solape con la mano).

La extracción de la mano se realiza mediante un procedimiento basado en una operación de binarización. La binarización consiste en dividir la imagen de entrada en dos zonas a través de un umbral: los píxeles que tienen un nivel de intensidad mayor que el umbral se establecen a un valor lógico alto mientras que los píxeles que están por debajo se establecen a un valor nulo.

$$binary(x,y) = \begin{cases} 1, & image(x,y) \geq umbral \\ 0, & de\ otra\ manera \end{cases}$$

El valor del umbral puede ser global para toda la imagen o se puede escoger un valor distinto según la región (umbrales locales).

A partir de la imagen diferencia, se aplica una binarización con un nivel de umbral que permite distinguir entre los píxeles que conforman la mano y el resto. La figura resultante es, por tanto, la imagen binaria de la mano.

En este punto se realiza una operación *AND* lógica con la máscara de la mano (ROI) construida en la fase inicial eliminando así la zona que no es de interés. Finalmente, se procede a realizar un filtrado morfológico con el objeto de mejorar la imagen binarizada inicial eliminando ruido y rellenando huecos. En primer lugar, se rellenan los posibles huecos que puedan quedar contenidos en el contorno de cada uno de los *blobs* (componentes conexos con forma indistinta) resultantes de la binarización. Estos huecos pueden aparecer debido a sombras y/o elementos no controlados. A continuación se eliminan los píxeles aislados (puntos blancos rodeados de píxeles negros) que pudieran aparecer como resultado de la binarización.

Como ejemplo se muestra la Figura 4.6.

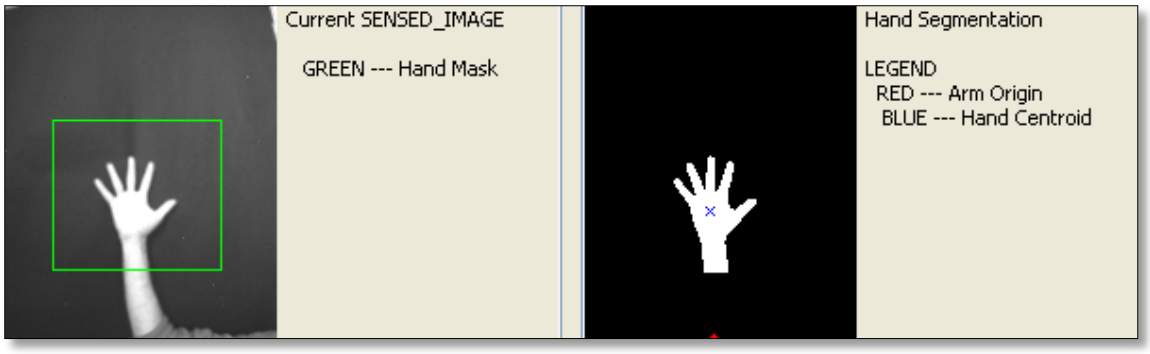


Figura 4.6. En la izquierda, imagen capturada. A la derecha la imagen resultante de la binarización. La imagen resultante está compuesta por un único *blob*.

4.3.2. Segmentación temporal

La segmentación temporal trata de encontrar los momentos en que se inicializa y se finaliza el gesto. Este es un problema ampliamente estudiado cuyo resultado es un conjunto de soluciones muy variadas. En este trabajo se ha optado por considerar que el inicio del movimiento sucede cuando la mano ha aparecido de forma continuada durante un número determinado de imágenes. De igual modo, el movimiento finaliza cuando la mano que realiza el gesto ha desparecido de la zona de captura por un determinado número de imágenes.

Esta forma de trabajar tiene la ventaja de permitir de forma inherente la eliminación del conjunto de imágenes iniciales y finales de la secuencia, que corresponden a las fases de preparación y finalización del gesto respectivamente. Estas fases podrían llevar a malinterpretaciones por parte del sistema del gesto realizado.

A lo largo de la Figura 4.7 se muestra una secuencia de conteo de dos dedos. En la imagen Figura 4.7a la mano no ha sido detectada aún debido a que el enmascarado provoca que no haya suficientes puntos activos como para detectar la presencia de la mano. Este funcionamiento permite eliminar la parte inicial de la secuencia que se corresponde con la fase de preparación del gesto.

En Figura 4.7b se tiene la primera detección de la mano. Sin embargo, esta imagen no es considerada como inicio de la secuencia. En Figura 4.7e se alcanza el umbral de detecciones a partir del cual se considera que la presencia de la mano no es casual o las detecciones han estado provocadas por algún artefacto. Se entiende, por tanto, que en este punto ha empezado un movimiento y la figura es la imagen de inicio de secuencia.

En Figura 4.7l la mano tiene poca presencia en la ROI y no es detectada. Al llegar a Figura 4.7o sin ninguna detección, se da por terminado el movimiento. La imagen Figura 4.7o es, por tanto, la imagen de final de movimiento. Hay que notar que la extracción de características de la mano sólo se realiza mientras ésta es detectada. Esto significa que durante la fase de finalización de la secuencia no se recogen datos.

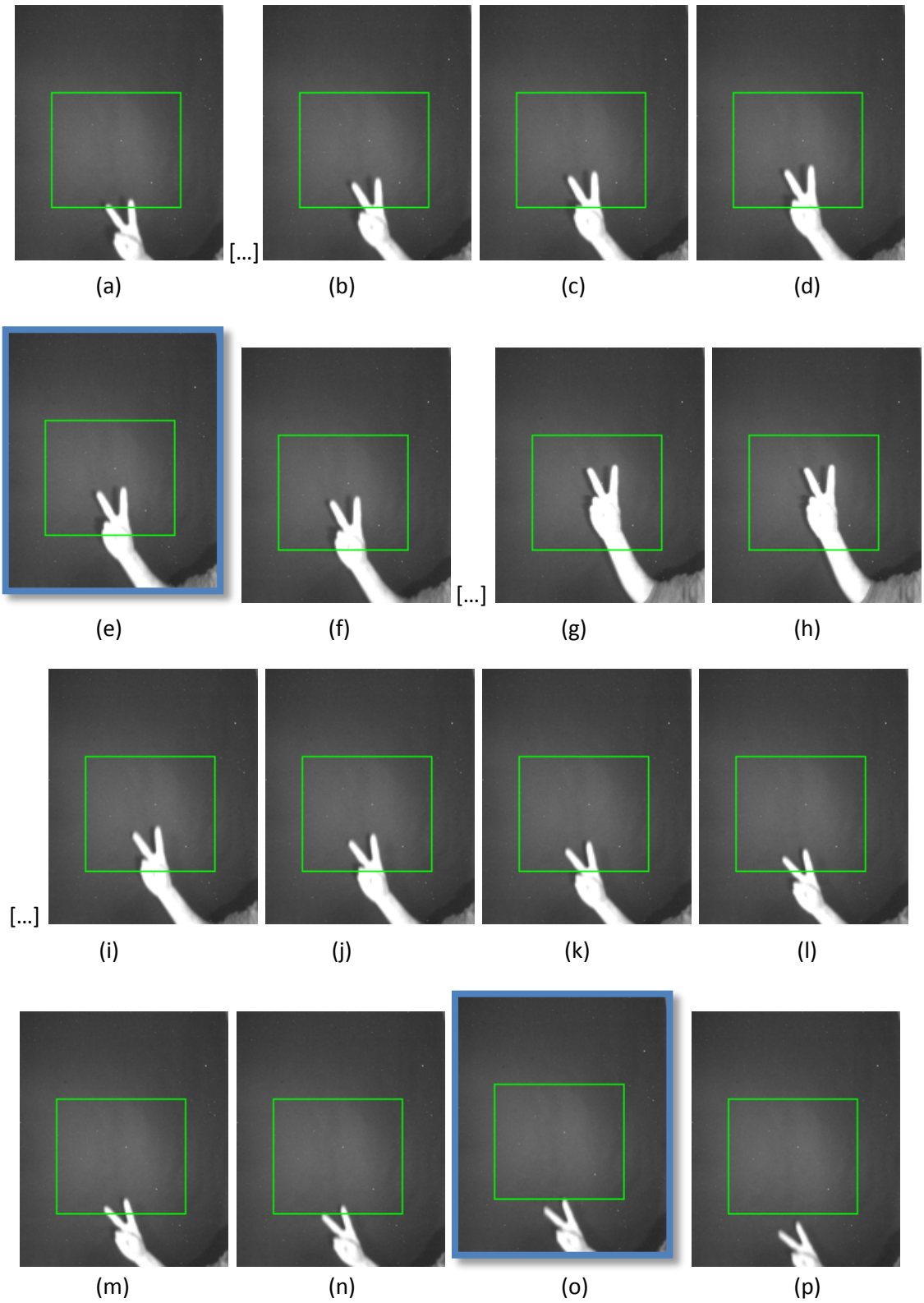


Figura 4.7. Movimiento de conteo de dos dedos.

4.4. Cálculo del centroide

Uno de los parámetros más utilizados para seguir la evolución general de un objeto conexo es el seguimiento del centroide. En definitiva, la localización de un objeto se puede tomar por la evolución de su baricentro.

Para una figura X en el espacio $\mathbb{R}^2$, las coordenadas del baricentro se pueden expresar matemáticamente como:

$$C_x = \frac{\int x\, S_y(x)\, dx}{A}\ ; \qquad C_y = \frac{\int y\, S_x(y)\, dy}{A}$$

Dónde A es el área de la figura X, $S_y(x)$ es la longitud de la intersección de X con la línea vertical en la abscisa x y $S_x(y)$ es el análogo para el eje de las ordenadas.

Una forma más práctica de hallar el centroide consiste en ir erosionando un pixel la imagen de forma iterativa (sin llegar a eliminar las figuras conectadas) hasta que no existen cambios entre las iteraciones (Figura 4.8).

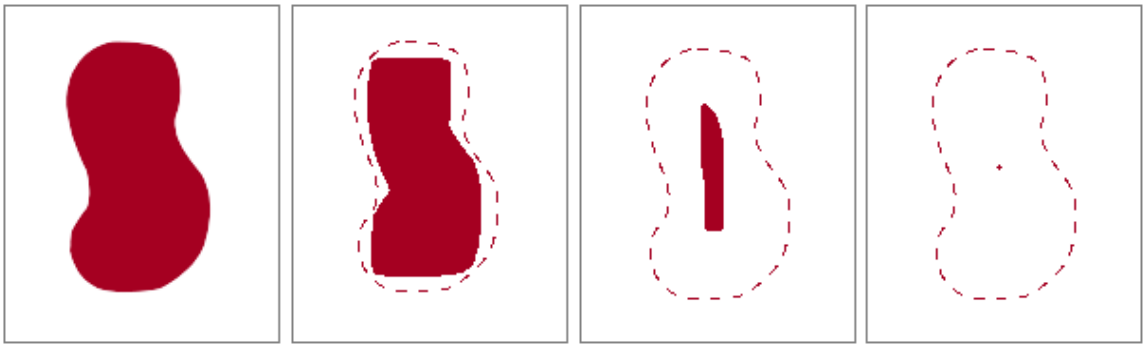


Figura 4.8. Proceso de obtención del centroide de una imagen.

Una situación común cuando se realiza el cálculo del centroide de la mano es que la presencia de parte del brazo puede provocar que el centroide se desplace significativamente en la dirección de éste.

Para subsanar este inconveniente en el cálculo del centroide, se realiza un algoritmo iterativo que permite separar la mano del brazo creando dos *blobs* diferentes. Se calcula inicialmente el centroide de todo el *blob* y éste es tomado como valor inicial del centroide de la mano. Posteriormente, se realizan una serie de erosiones con el objetivo de separar el brazo de la mano. En cada iteración se recalcula el centroide tomando como valor el centroide del *blob* de mayor área que está situado más alejado de la posición por dónde entra el brazo.

Existe, sin embargo, un número mínimo de erosiones para eliminar posible ruido así como los *blobs* que se pueden ocasionar por la presencia de los dedos. Durante esas erosiones no se recalcula el centroide por lo que puede ocurrir que en algunas ocasiones este procedimiento

provoque la desaparición por completo de la mano. Estas ocasiones suelen suceder cuando la mano aún no ha entrado completamente en la región de interés por lo que el valor inicial del centroide es el valor correcto.

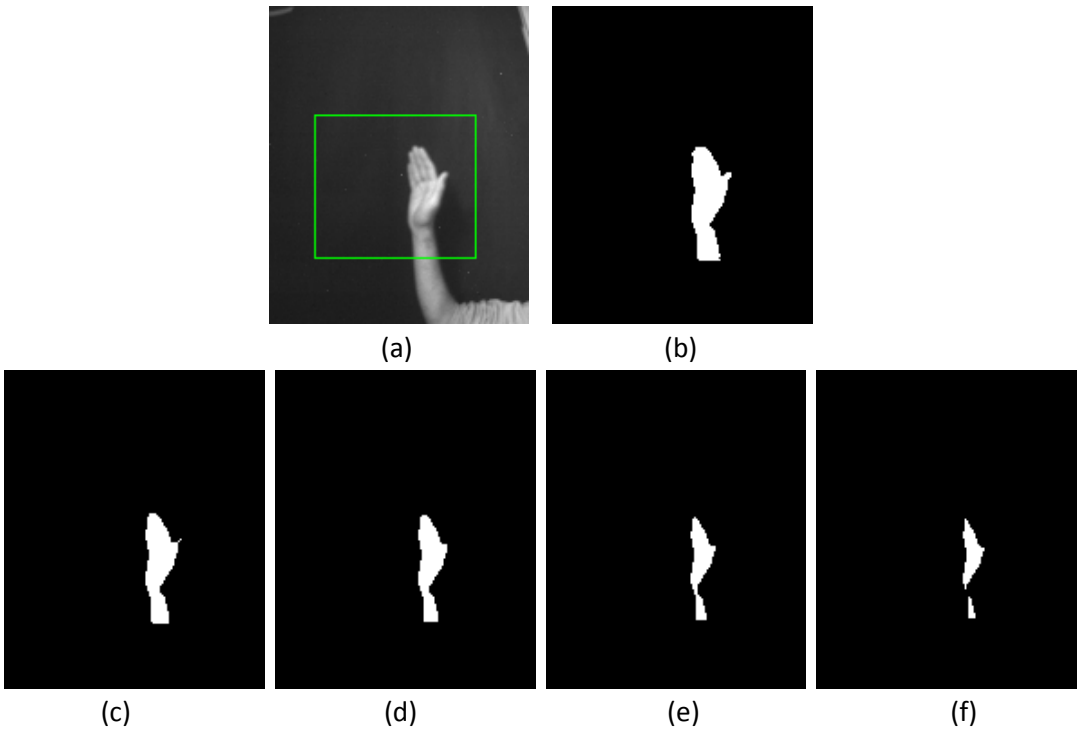


Figura 4.9. Cálculo del centroide. En (f) se separa la mano del brazo dando lugar a dos *blobs*. El centroide del *blob* superior corresponde al centroide de la mano.

4.5. Taxonomía del gesto

La taxonomía del gesto trata de clasificar de forma dinámica el tipo de gesto que se está realizando: estático o dinámico.

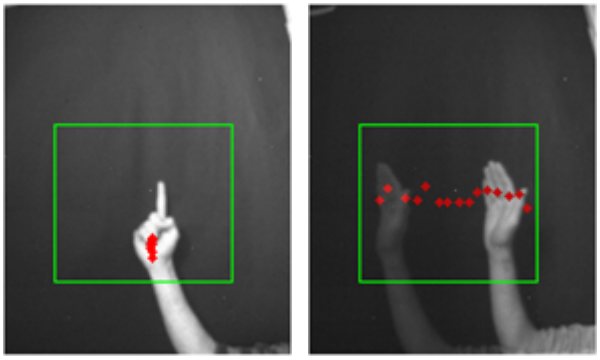
Esta clasificación se realiza en función de la evolución seguida por el centroide. Cada cierto número de *frames* se calcula la varianza de las coordenadas x e y del centroide. El número de coordenadas que se tienen en cuenta para este cálculo está determinado por una ventana temporal.

La varianza del centroide para la coordenada x, C_x y para la coordenada y, C_y en el instante n se expresa, por tanto:

$$var(C_x, n) = \frac{1}{M-1} \sum_{i=n-M+1}^{i=n} (x_i - \bar{x})^2 ; \qquad var(C_y, n) = \frac{1}{M-1} \sum_{i=n-M+1}^{i=n} (y_i - \bar{y})^2$$

Dónde M es el tamaño de la ventana temporal y $\bar{x}$ e $\bar{y}$ denotan la media de las coordenadas x e y respectivamente para la ventana temporal considerada.

Si la varianza calculada de alguna de las coordenadas supera un cierto umbral durante varios cómputos seguidos, el movimiento es clasificado como dinámico. En caso contrario, se trata de un movimiento estático.



(a) (b)

Figura 4.10. Movimiento estático a la izquierda. Movimiento dinámico en la derecha. Se señala con puntos en rojo la evolución del centroide.

La elección de los valores de periodicidad y longitud de la ventana temporal utilizados en el cálculo la varianza del centroide depende del *frame-rate* de trabajo. Para un *frame-rate* de 33 *fps* se ha determinado de forma empírica un valor óptimo de 5 *frames* para el tamaño de la ventana temporal y 2 *frames* de diferencia entre cada cálculo.

C_x	47	55	60	64	70	76	82	88	96	107
C_y	117	115	114	105	102	102	101	109	110	110
						↓				↓
$var(C_x)$		32		54		68		90		148
$var(C_y)$		2		29		41		11		20

Tabla 4.1. Ejemplo de evolución de la varianza del centroide

Este sistema permite hacer la distinción entre ambos tipos de gesto y establecer, por tanto, un tratamiento diferenciado para cada uno de ellos (Figura 4.11).

A lo largo de los capítulos 4 y 5 se discute el tratamiento particular de cada tipo de gesto.

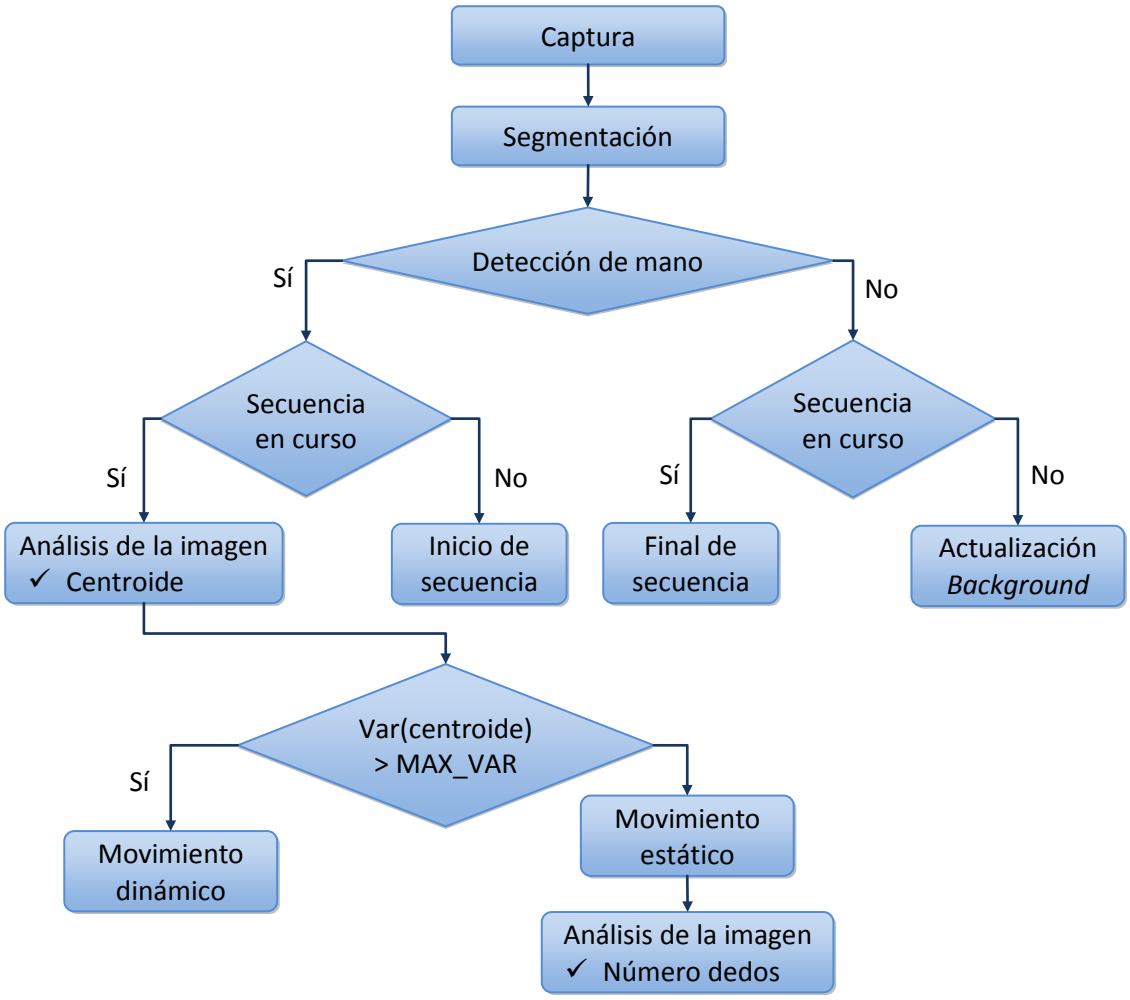


Figura 4.11. Comportamiento del sistema.

Tal como muestra la Figura 4.11, existen cuatro situaciones dentro del comportamiento del sistema:

- La mano está presente y un gesto está siendo procesado. En este caso se realiza el análisis del centroide estableciendo la taxonomía del gesto. Si el movimiento es estático se realiza el conteo de dedos.
- La mano está presente y no se está procesando ningún gesto. Si la mano ha aparecido de forma continua durante un determinado número de imágenes, la imagen que se está analizando es la que determina el inicio del gesto.
- La mano no está presente y un gesto está siendo procesado. Si la mano no ha aparecido de forma continua durante un determinado número de imágenes, se considera que el gesto ha terminado. Se procede, por tanto, a la clasificación del gesto realizado.
- La mano no está presente y no se está procesando ningún gesto. Se entiende en este caso que, al no estar la mano presente y no encontrarse un movimiento en proceso, la imagen capturada corresponde a la imagen de *background*. Es por ello que en esta coyuntura se procede a actualizar esta imagen. Este apartado permite la actualización dinámica del fondo, en caso que se produjera algún cambio.

Capítulo 5

Detección de gestos estáticos

5.1. Nociones básicas

5.1.1. Esqueleto Morfológico

El esqueleto morfológico de una figura puede ser definido como el lugar geométrico de los puntos equidistantes al contorno (Figura 5.1). Generalmente, el esqueleto enfatiza propiedades geométricas y/o topológicas de la figura como su conectividad, topología, longitud, dirección y amplitud. El esqueleto morfológico puede ser empleado como una representación de la forma de la figura si es utilizada conjuntamente con la distancia de sus puntos al contorno de la figura (estos datos conforman toda la información necesaria para reconstruir la figura).

Figura 5.1. Esqueleto morfológico.

El esqueleto de una figura A, $S(A)$, puede ser expresado mediante operaciones de erosión y apertura:

$$S(A) = \bigcup_{k=0}^{K} S_k(A)$$

Con

$$S_k(A) = \varepsilon_b^k\{A\} - \gamma_b\left(\varepsilon_b^k\{A\}\right)$$

Dónde b es un elemento estructurante, $\gamma_b(\cdot)$ denota la operación de apertura morfológica,

$$\gamma_b(X) = (X \Theta b) \oplus b$$

$\varepsilon_b^k\{A\}$ indica k erosiones sucesivas de la figura A,

$$\varepsilon_b^k\{A\} = \overbrace{(\dots((A\Theta b)\Theta b)\Theta\dots)\Theta b}^{k\text{ veces}}$$

Y K es la última iteración antes de que la erosión de A resulte en un conjunto vacío:

$$K = max\{k \mid \varepsilon_b^k\{A\} \neq \emptyset\}$$

De la formulación presentada, se puede demostrar que la figura A puede ser reconstruida a partir de los subconjuntos del esqueleto $S_k(A)$ por medio de la ecuación:

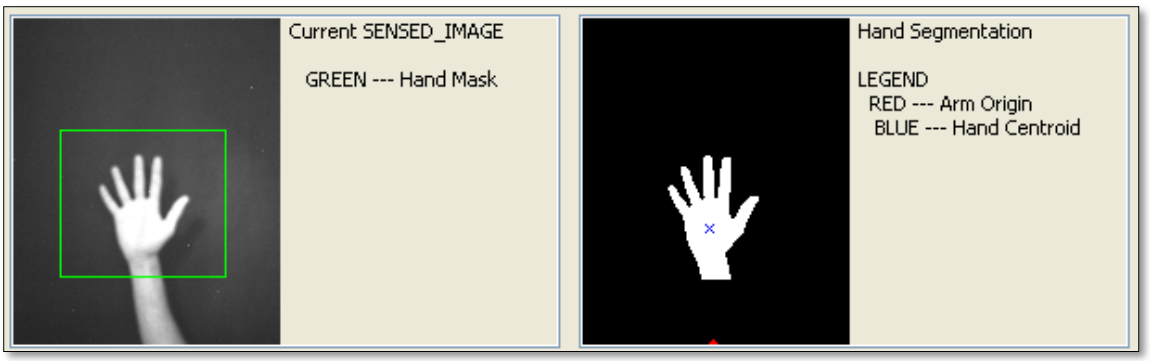
$$A = \bigcup_{k=0}^{K} \delta_b^k\{S_k(A)\}$$

Dónde $\delta_b^k\{S_k(A)\}$ denota k dilataciones sucesivas de $S_k(A)$. Esto es,

$$\delta_b^k\{S_k(A)\} = \overbrace{(\dots((S_k(A)\oplus b)\oplus b)\oplus\ \dots)\oplus b}^{k\text{ veces}}$$

De la expresión morfológica del esqueleto se deduce que su construcción comporta más iteraciones según sea más grande el tamaño de la figura. Sin embargo, esta operación se puede realizar directamente en el Q-Eye por lo que no supone un coste computacional muy elevado. Para una imagen típica como la de la Figura 5.2 el consumo de tiempo ocasionado por el cálculo del esqueleto es de alrededor de los 350 µs

Hay que notar, además, que las imágenes capturadas durante la ejecución de la aplicación se encuentran enmascaradas (Figura 5.2), por lo que la región de interés es menor.

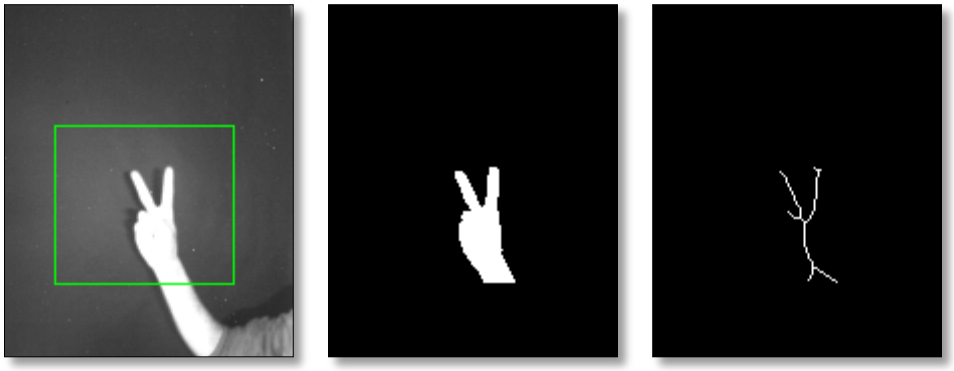


(a) (b)

Figura 5.2. Imagen capturada en una secuencia en tiempo real. (a) Imagen capturada. En verde se muestra la ROI. (b) Segmentación de la mano. Se presenta el centroide y la posición del origen del brazo.

5.1.2. *End points*

Los *end points* son aquellos puntos del esqueleto morfológico donde finaliza alguna de sus ramas.



(a) (b) (c)

Figura 5.3. (a) Imagen capturada. (b) Imagen segmentada y enmascarada. (c) Esqueleto morfológico.

En la Figura 5.4 se puede observar un zoom del esqueleto morfológico de la imagen Figura 5.3c dónde se han marcado en verde los *end points*.

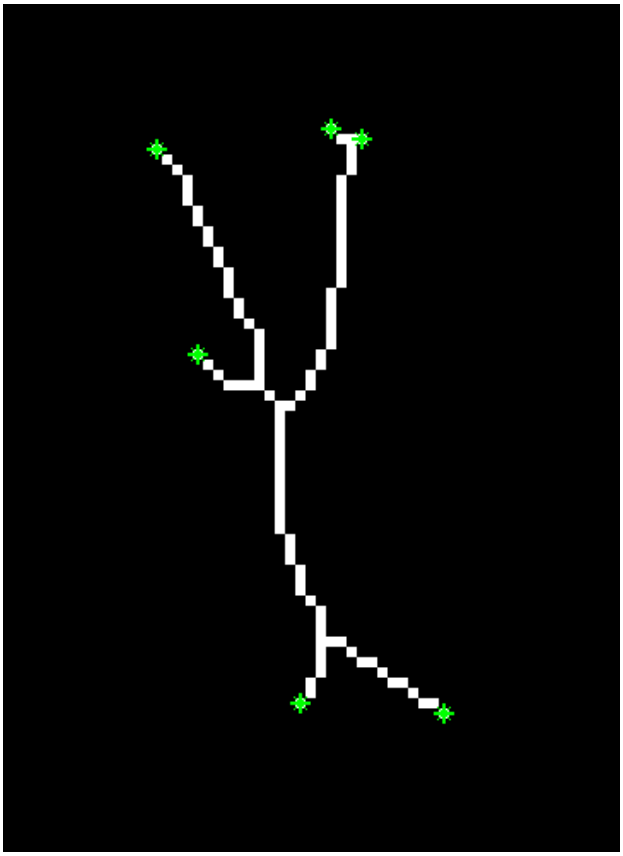
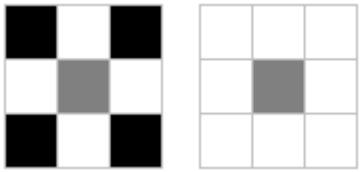


Figura 5.4. Ejemplos de *End points*.

Una definición más formal de *end point* implica la inclusión de conceptos de conectividad.

Se dice que un píxel activo está 4-conectado o tiene conectividad 4 si tiene las mismas propiedades que alguno de sus 4 vecinos más cercanos (Figura 5.5a). Un píxel activo está 8-conectado o tiene conectividad 8 si alguno de sus 8 vecinos más cercanos (Figura 5.5b) está activo.



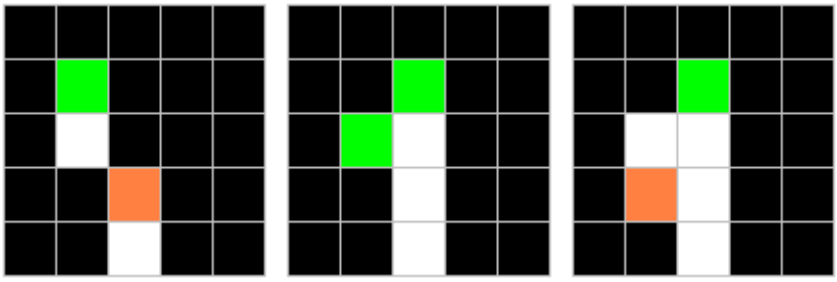
(a) (b)

Figura 5.5. Concepto de vecindad. El pixel central (en gris) es el pixel de estudio, los puntos blancos son puntos activos y los puntos negros son inactivos.
(a) Píxeles de vecindad 4. (b) Píxeles de vecindad 8.

Un punto es tomado como *end point* si:

- Se halla 8-conectado con un único vecino activo.
- Se halla 8-conectado con más de un píxel activo, se halla 4-conectado con un único vecino y los vecinos activos son contiguos.

La Figura 5.6 presenta algunos ejemplos de puntos identificados como *end points* (en color verde) y puntos que no lo son (en color naranja).



(a) (b) (c)

Figura 5.6. Ejemplos de *end points*. (a) El punto en color verde tiene un único vecino de tipo 8-conectivo. El punto naranja tiene dos 8-vecinos activos, está 4-conectado por un único vecino pero no es un *end point* ya que los vecinos activos no son contiguos. (b) Los dos puntos en verde son *end points* ya que tienen más de un píxel 8-conectado activo, un único vecino 4-conectado activo y los vecinos activos son contiguos. (c) El píxel naranja no es un *end point* ya que, aunque todos sus vecinos son contiguos, tiene más de un píxel 4-conectado activo.

Los *end points* pueden ser calculados en el procesador analógico del sistema Eye-RIS. Para una imagen, el tiempo aproximado de cálculo de todos sus *end points* (independientemente del número de ellos) se encuentra alrededor de los 90-95µs.

5.1.3. *Skeleton joints*

Las *skeleton joints* de una imagen esqueleto morfológico de entrada son aquellos puntos en los que confluyen tres o más ramas del esqueleto.

La Figura 5.7 ilustra ejemplos de *skeleton joints*.

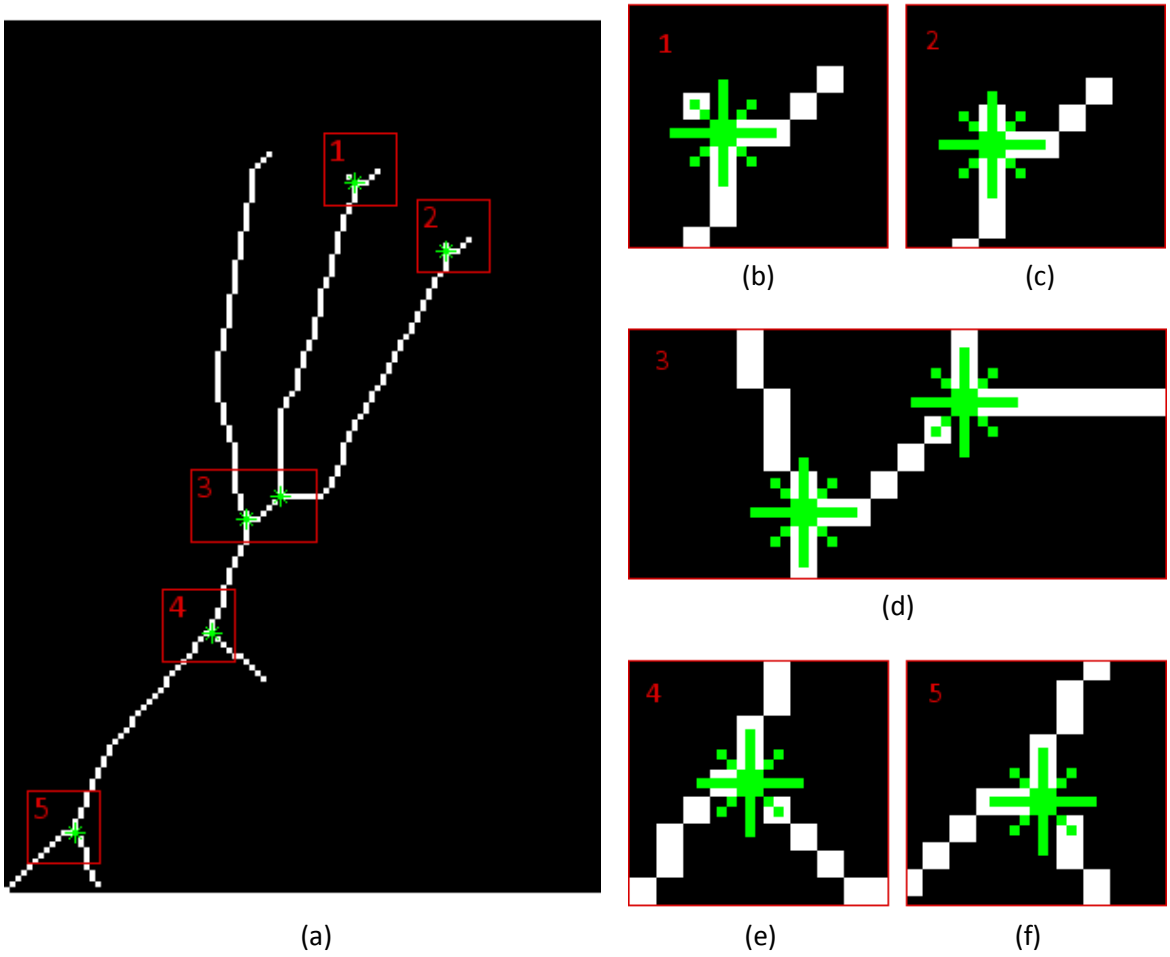


Figura 5.7. S*keleton joints*. (a) Esqueleto morfológico. (b) *Zoom* de *skeleton joint* 1. (c) *Zoom* de *skeleton joint* 2. (d) *Zoom* de *skeleton joints* 3. (e) *Zoom* de *skeleton joint* 4. (f) *Zoom* de *skeleton joint* 5.

Las *skeleton joints* son calculadas por el Q-Eye y la operación toma aproximadamente 103 µs.

5.2. Conteo de dedos

El conteo de dedos es la forma elegida para objetivos como la selección de opciones ordenadas en un menú gráfico. Se han diseñado tres sistemas de conteo de dedos de la mano. El primer y segundo métodos se basan en el concepto de distancia geodésica mientras que el tercer método está basado en la construcción del esqueleto morfológico. Mientras que el primero y segundo resultan métodos más robustos, tienen un coste computacional más elevado.

5.2.1. Distancia geodésica

5.2.1.1. Imagen de distancia geodésica

La imagen de distancia geodésica representa de forma gráfica la distancia geodésica de cada punto de la silueta de la mano al centroide de la misma.

La distancia geodésica desde el centroide de la mano C a un punto x en la silueta S se define matemáticamente como:

$$d_S(C,x) = n \Leftrightarrow x \in \delta_n^S(C) \quad \text{y} \quad x \notin \delta_{n-1}^S(C)$$

Dónde $\delta_n^S(C)$ es la dilatación geodésica n-ésima del punto C en la silueta S y se expresa como:

$$\delta_n^S(C) = \overbrace{\delta(\dots(\delta(\delta(C)\cap S)\cap S)\dots)\cap S}^{n \text{ veces}}$$

Siendo δ la operación morfológica de dilatación.

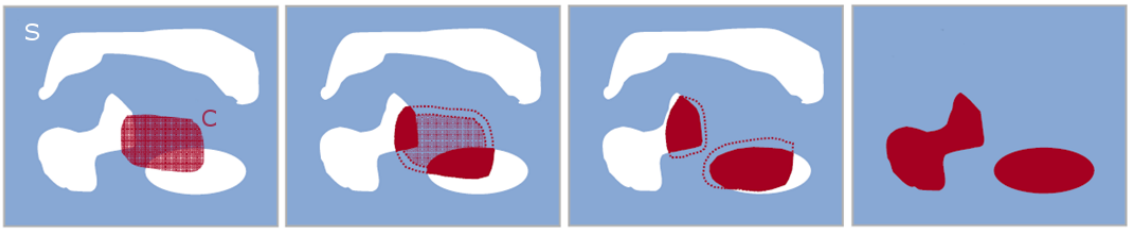


Figura 5.8. Dilatación geodésica del elemento C sobre la silueta S. A la derecha, reconstrucción de los componentes conectados marcados por C.

El proceso para la obtención de la imagen de distancia geodésica consiste, pues, en realizar operaciones de dilatación a partir del centroide. El resultado se interseca con la imagen binaria de la mano y se itera el proceso a partir del resultado. El número de dilataciones que son necesarias para llegar a un píxel cualquiera de la región es el valor de distancia geodésica.

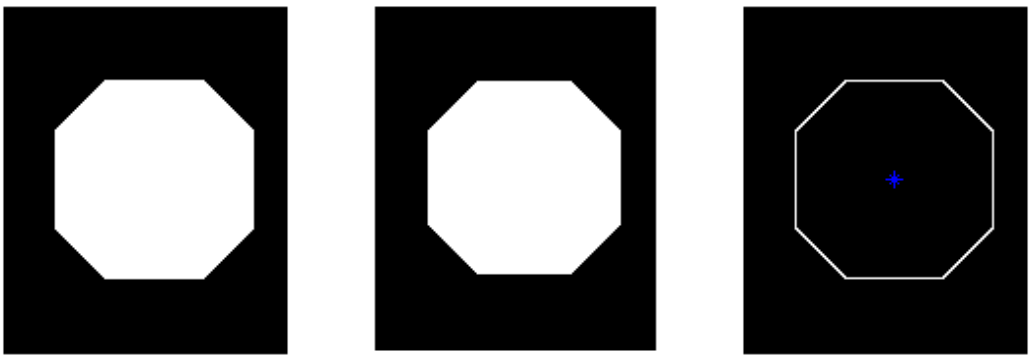
El elemento estructurante ideal para realizar las dilataciones del centroide es el de forma circular. Sin embargo, el tratamiento con elementos estructurantes circulares no resulta sencillo y conlleva un consumo de tiempo considerable. Con el objetivo de conseguir unas dilataciones similares a las que se obtienen con un elemento estructurante circular, se realizan las dilataciones alternando una cruz y un cuadrado de dimensiones 3x3.

En la Figura 5.9c se puede observar como es la evolución general del elemento estructurante final.

El resultado es una imagen multivaluada (o de nivel de gris), donde el valor de cada píxel representa la distancia geodésica respecto al centroide. Las imágenes en nivel de gris están sujetas a degradación en procesadores de plano focal por lo que el procedimiento para la construcción de la imagen de distancia geodésica ha sido adaptado a una forma más adecuada para este tipo de procesadores. Por ello, el resultado se obtiene directamente en la descomposición en planos de bits descrita en la sección 2.2 trabajando así sobre imágenes binarias.

La idea principal consiste en que, al realizar la diferencia entre la dilatación n-ésima del centroide y la dilatación $(n-1)$-ésima se obtiene el lugar geométrico que representa los puntos que se hallan a una distancia geodésica del centroide constante y de valor igual a n (Figura 5.9).

Al intersecar con la imagen binaria de la mano se obtiene el conjunto de puntos activos a distancia n del centroide. Los resultados parciales de cada intersección se van acumulando en los planos de bit del orden al que pertenece la dilatación. Esto es, si se está realizando la operación de dilatación n-ésima (valores a distancia geodésica n del centroide), dónde n se expresa en base binaria como $n_{10} = \left(b_{n_b-1}, b_{n_b-2}, \dots, b_0\right)_2$, los planos de bit en que el resultado debe ser almacenado son aquellos que corresponden a los bits b_i activos.



(a) (b) (c)

Figura 5.9. Círculo de distancia geodésica constante. (a) Dilatación $(n-1)$ –ésima. (b) Dilatación n-ésima. (c) Puntos a distancia geodésica n del centroide (en azul).

La imagen final en nivel de gris se puede obtener por recomposición de las 8 imágenes binarias resultantes. En la Figura 5.10 se muestra un ejemplo de construcción de los planos de bits de la distancia geodésica y su posterior recomposición a niveles de gris.

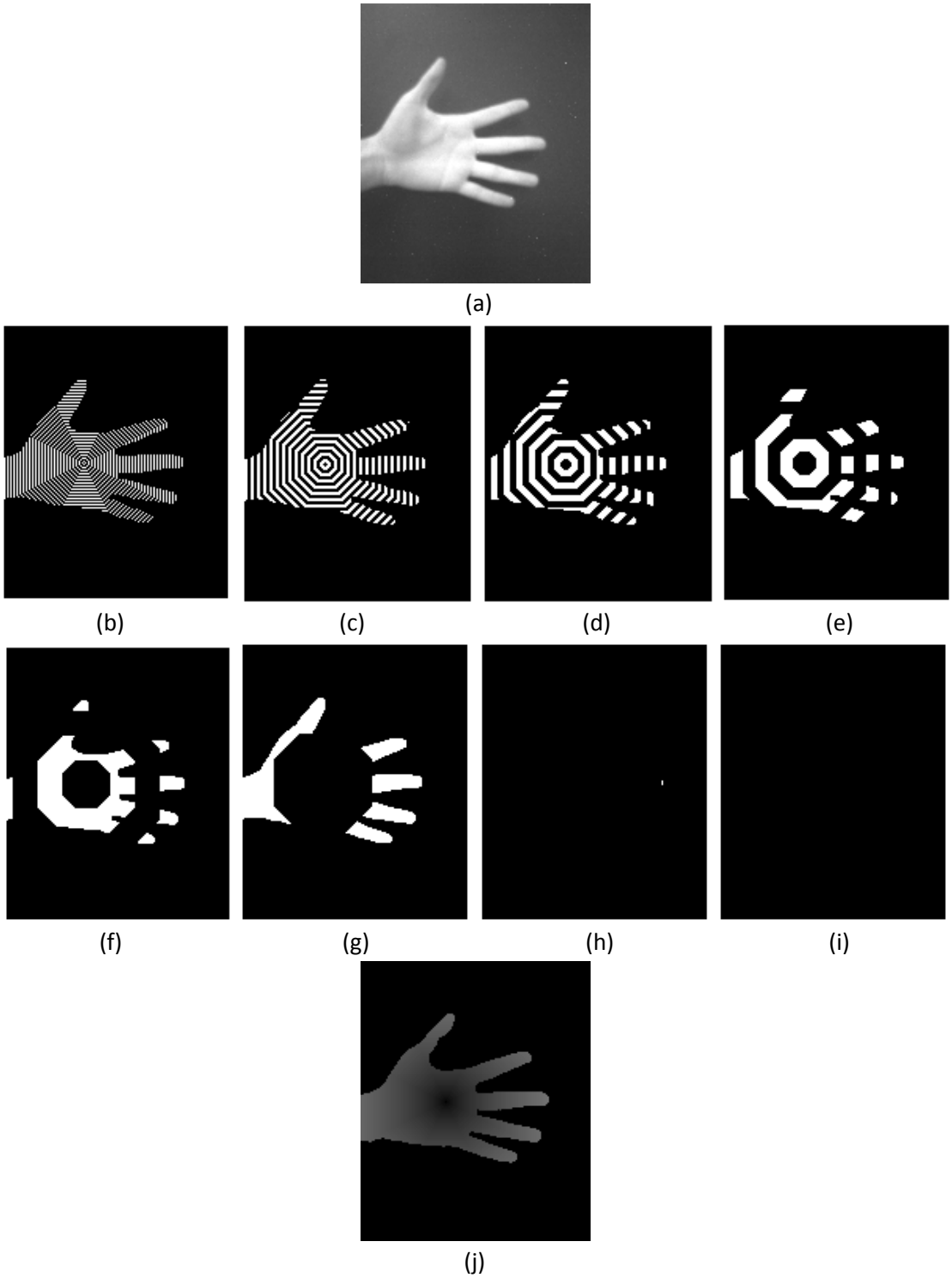


Figura 5.10. Distancia geodésica.
(a) Imagen capturada. (b) a (i) Planos de bit resultantes. (j) Imagen distancia geodésica.

5.2.1.2. Seguimiento de valores de la distancia geodésica

Se pretende extraer en un vector lineal los valores de intensidad de los píxeles que constituyen el borde de la imagen de distancia geodésica así como sus coordenadas en el orden en que se han leído. De este modo se puede establecer una aplicación de correspondencia entre los

valores y su posición en la imagen. El vector resultante se utilizará en la fase de clasificación del gesto. Este proceso se realiza en el procesador digital del sistema Eye-RIS.

El trazado del contorno de una figura conexa es un procedimiento muy frecuente en el procesado de imágenes (es muy utilizado para el reconocimiento de caracteres escritos, por ejemplo). Existen, por lo tanto, diversos algoritmos de recorrido que tratan de resolver el problema.

Aquí se ha optado, no obstante, por adoptar una solución consistente en la construcción del contorno de la imagen y el recorrido de los píxeles activos de ésta. Esta aproximación resuelve, para el caso particular de la aplicación desarrollada, alguno de los problemas que presentan los métodos de trazado del contorno.

Para obtener el contorno de la imagen de distancia geodésica se sigue el procedimiento marcado por el esquema de la Figura 5.11.

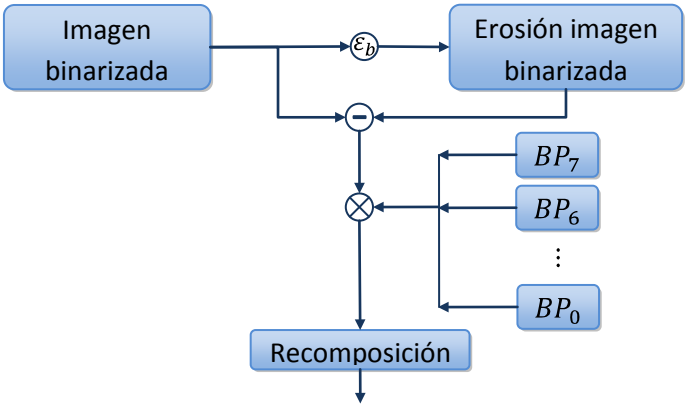
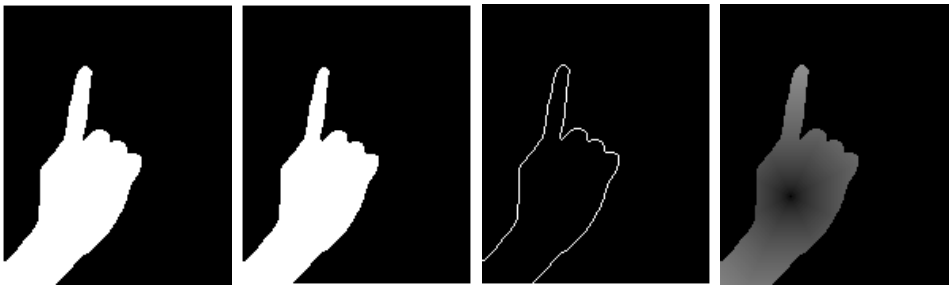


Figura 5.11. Obtención de la imagen de contorno.

El procedimiento consiste en realizar una diferencia entre la imagen binarizada de la mano y una erosión de ella misma. El resultado de esta diferencia es una línea que corresponde al contorno de la mano. Para obtener el contorno de la imagen de distancia geodésica se realiza una operación *AND* lógica entre la imagen diferencia y los planos de bit de la imagen distancia geodésica. Finalmente, y para recuperar la imagen en nivel de gris, se realiza la recomposición de los planos de bit. En la Figura 5.12 se muestra un ejemplo del contorno de la imagen de distancia geodésica.



(a) (b) (c) (d)

(e)

Figura 5.12. (a) Imagen binarizada. (b) Erosión de la imagen binarizada. (c) Diferencia. (d) Imagen de distancia geodésica. (e) Contorno de la imagen de distancia geodésica.

El orden de recorrido de los píxeles de la imagen contorno sigue los pasos:

- Elección del punto inicial y punto final.
- Seguimiento de la línea a través de condiciones de conectividad.
- Eliminación de puntos ya recorridos.
- Finalización del algoritmo al llegar al punto final.

Elección del punto inicial y punto final

Para la elección del punto inicial y final del recorrido se define en primer lugar un marco formado por los píxeles que se encuentran a distancia uno del borde de la imagen.

Los candidatos a punto inicial y final se eligen a partir de los puntos del contorno de la imagen de distancia geodésica que están en contacto con este marco. En el caso general, esta intersección da como resultado dos puntos aislados que son tomados como punto inicial y final respectivamente. Sin embargo, existen casos en los que se producen ambigüedades en la elección del punto inicial y final. Para resolverlas se usará la información contenida en el marco formado por los píxeles a distancia dos de los bordes de la imagen.

Según el número de píxeles activos que intersecan el marco a distancia uno se dan los siguientes casos:

- Existe un único candidato: El caso es trivial y se establecen los valores de punto inicial y final como los de este píxel.
- Existen varios candidatos formando un único segmento: Se establece uno de los extremos del segmento como punto inicial y el otro como punto final.
- Existen varios candidatos que no forman un único segmento:
  - Si los candidatos son dos puntos aislados, se toma uno de ellos como punto inicial y el otro como final.
  - Si uno de los candidatos es un punto aislado y el otro es un segmento, el punto aislado se toma como punto inicial. El punto final es el extremo del segmento (en el marco a distancia uno) que sólo tiene un píxel 8-vecino activo, considerando también los píxeles a distancia dos del contorno (Ver Figura 5.14).

- Si los candidatos forman dos segmentos, se toman como puntos inicial y final los extremos de los segmentos (en el marco a distancia uno) que sólo tienen un píxel 8-vecino activo, considerando también los píxeles a distancia dos del contorno.

- Si los candidatos forman más de dos segmentos, se ha capturado en la segmentación algún *blob* no deseado y no se realiza el análisis de la imagen.

Es importante remarcar que el orden en que se recorre el contorno de la imagen de distancia geodésica (sentido horario o anti horario), no tiene ninguna influencia en la forma resultante de los valores extraídos (únicamente implica una inversión).

En las Figura 5.13 y Figura 5.14 se ilustran un par de casos de elección de punto inicial y final.

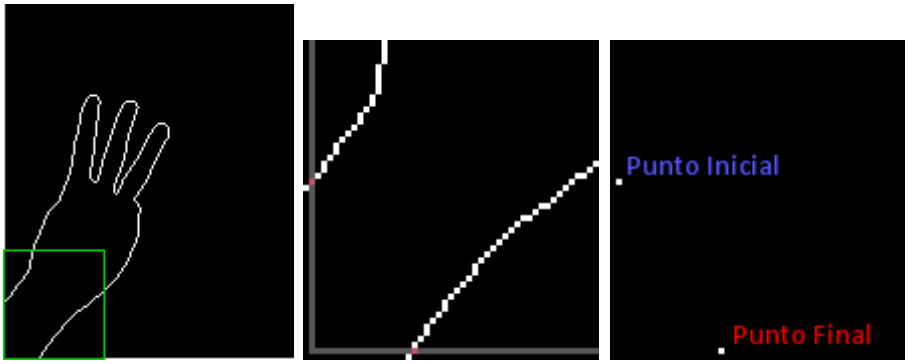
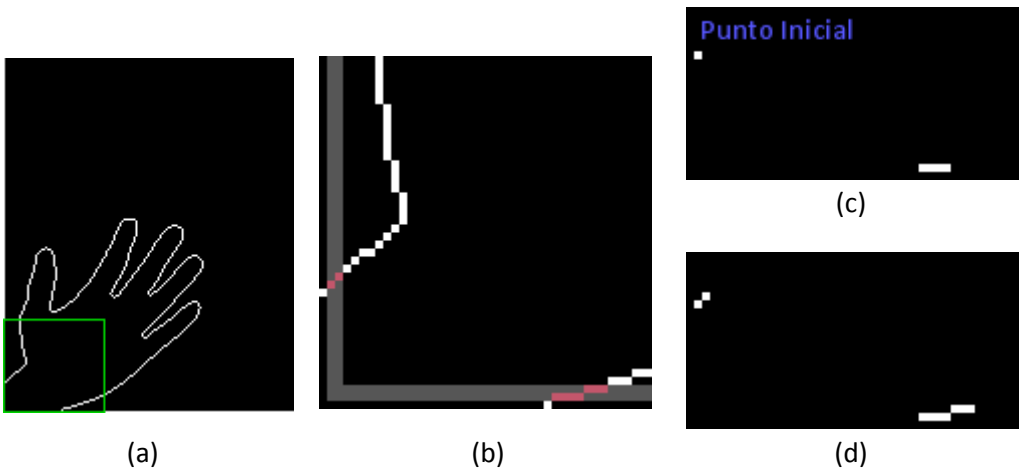
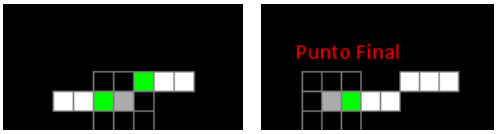


Figura 5.13. Detección del punto inicial y final con dos puntos como candidatos. (a) Contorno de la imagen distancia geodésica. (b) *Zoom* de la zona de interés. (c) Puntos inicial y final.

En el segundo caso (Figura 5.14), para hallar el punto final, se estudia el número de 8-vecinos activos para cada píxel que forma el segmento (Figura 5.14d). Aquel extremo que tenga únicamente un vecino será el punto que se tomará como punto final. Se puede observar en las imágenes Figura 5.14e y Figura 5.14f el número de píxeles vecinos para cada uno de los extremos del segmento.





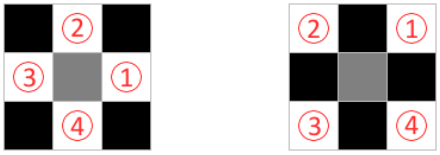
(e) (f)

Figura 5.14. Detección de punto inicial y final con un punto aislado y un segmento como candidatos. (a) Contorno. (b) *Zoom* de la zona de interés. (c) Candidatos. (d) Información de vecindades. (e) 8-vecinos del candidato 1. (f) 8-vecinos del candidato 2.

Seguimiento de la línea a través de condiciones de conectividad

El seguimiento de la línea de contorno de la imagen distancia geodésica se realiza en función de la conectividad del píxel de análisis con sus vecinos más cercanos. En función de la conectividad del píxel se determina la dirección de la exploración. En este sentido, se da prioridad a los píxeles de vecindad 4 y, si ninguno de estos se halla activo, se procede a realizar el análisis con los píxeles 8-vecinos (sin tener en cuenta los píxeles ya analizados).

Se tienen, por tanto, dos matrices que determinan la dirección de exploración a partir del píxel de análisis. Estas matrices y el orden inicial de exploración se muestran en la Figura 5.15.



(a) (b)

Figura 5.15. Dirección inicial de exploración. (a) Píxeles 4-vecinos. (b) Píxeles 8-vecinos.

Hay que notar que existe una correlación en las direcciones de exploración de las dos matrices. La diferencia radica en que el elemento de la Figura 5.15a contempla las direcciones vertical y horizontal mientras que el elemento de la Figura 5.15b tiene en cuenta las direcciones diagonales.

Se analizan los píxeles determinados por la matriz de la Figura 5.15a en el orden correspondiente y, si alguno de ellos se encuentra activo, se toma como siguiente píxel de exploración. Si el análisis de los píxeles 4-vecinos no es concluyente se pasa al análisis de los píxeles determinados por la Figura 5.15b, que pasa a determinar el siguiente píxel de exploración.

Para reducir el número de consultas sobre los píxeles vecinos, las matrices de exploración son actualizadas después de determinar el siguiente píxel de manera que la dirección inicial de la siguiente búsqueda sea la misma que la dirección que se ha seguido en el paso anterior (esto constituye lo que se denomina principio de inercia). El proceso finaliza cuando se llega al punto final.

En Figura 5.16 a Figura 5.18 se ilustra un ejemplo del seguimiento del contorno.

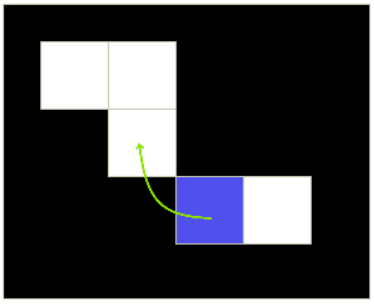


Figura 5.16. Línea de contorno.

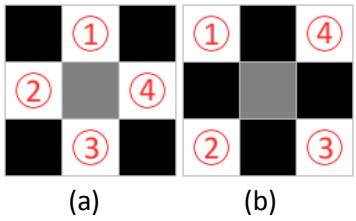


Figura 5.17. Direcciones de exploración. (a) 4-vecinos. (b) 8-vecinos.

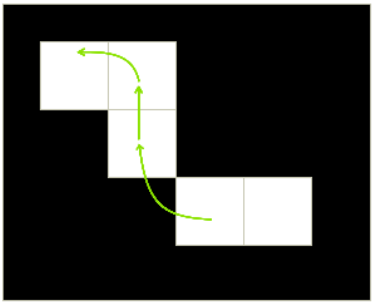


Figura 5.18. Seguimiento de la línea.

En la Figura 5.19 se muestra un ejemplo de la graficación de los valores del contorno de la imagen distancia geodésica –función que recibe el nombre de *Geodesic Distance Function* (GDF)–.

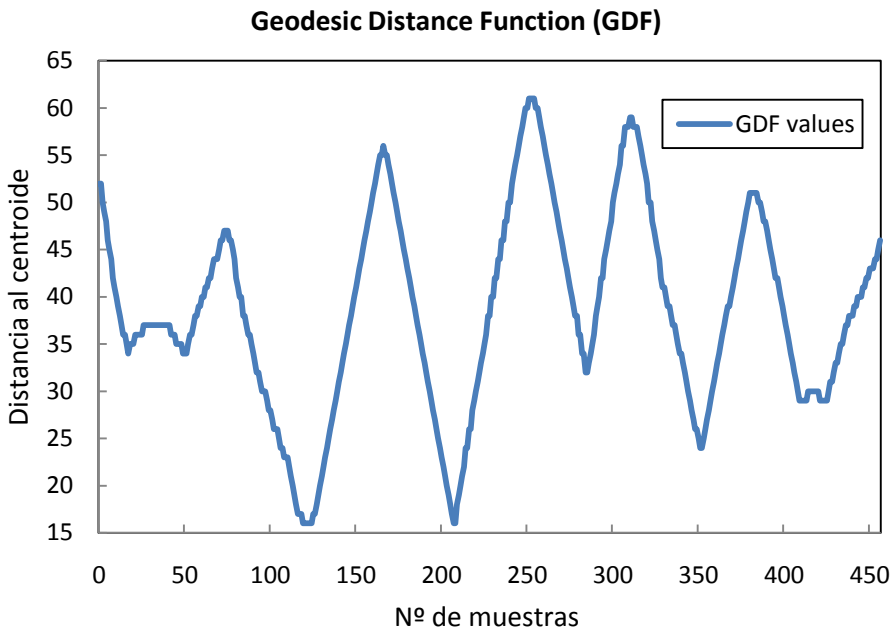


Figura 5.19. Valores de GDF.

5.2.1.3. Detección de máximos y mínimos

Del ejemplo de la Figura 5.19 se desprende que cada dedo resulta en un máximo debido a que éstos se encuentran en una posición más alejada del centroide y, por tanto, presentan un valor de distancia geodésica mayor. Hay que estudiar, por consiguiente, los máximos y mínimos locales de la función GDF.

Sin embargo, la función GDF contiene al inicio y al final máximos que corresponden al recorrido realizado por el brazo. Por ello, la función es recortada eliminando las zonas anteriores al primer mínimo y posteriores al último. En caso de que un máximo o mínimo se mantenga constante a lo largo de una región, se escoge como máximo o mínimo el punto medio de la región.

En la Figura 5.20 se puede observar la función GDF con los máximos y los mínimos detectados.

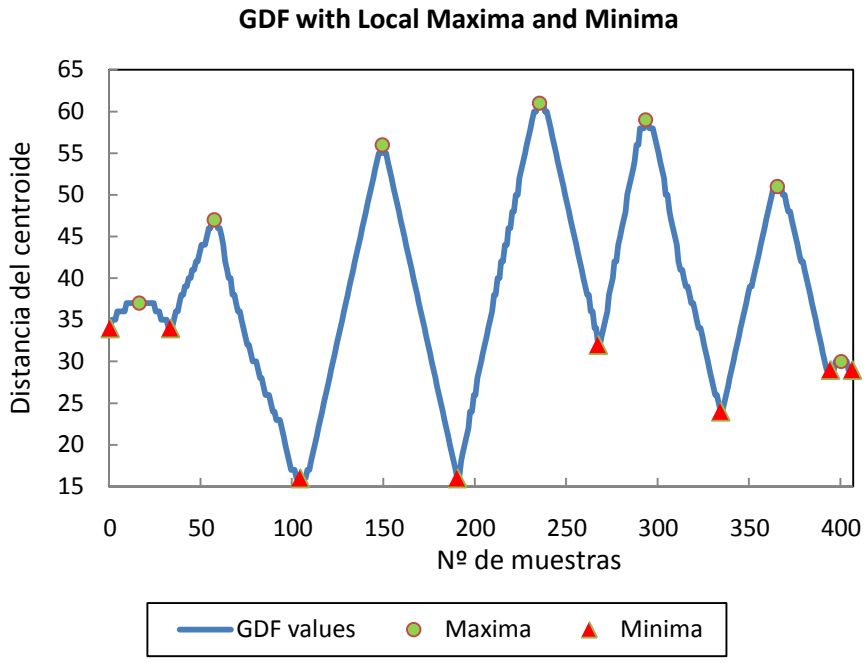
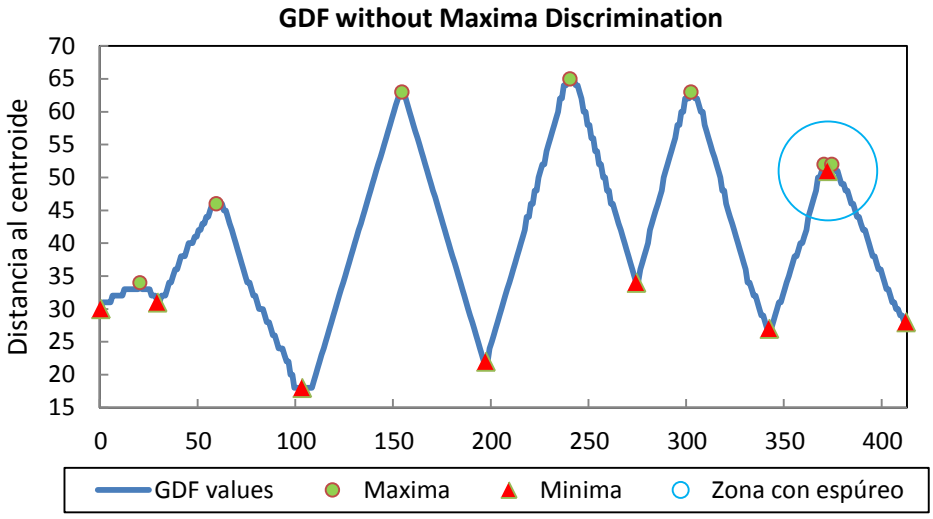


Figura 5.20. Detección de máximos y mínimos locales de la función GDF.

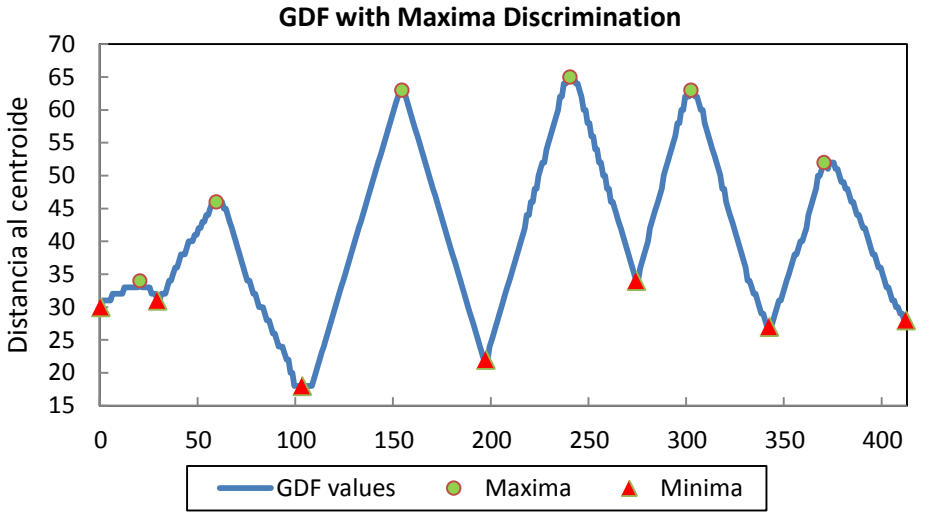
5.2.1.4. Discriminación de máximos

La aparición de espurios en la segmentación o, a veces, el propio contorno de la imagen de distancia geodésica provocan la aparición de máximos locales en posiciones demasiado próximas entre sí. Estos máximos deben ser discriminados así como los mínimos correspondientes. La Figura 5.21 muestra un ejemplo de discriminación de máximos.

(a) (b) (c)



(d)



(e)

Figura 5.21. Discriminación de máximos. (a) Imagen capturada. (b) Imagen binarizada. (c) Contorno. (d) GDF con máximos y mínimos locales. (e) GDF con discriminación de máximos.

El proceso de discriminación localiza las situaciones en que hay dos máximos a una distancia menor que la determinada como mínima y elimina uno de ellos quedándose con el que tiene un nivel de intensidad mayor.

5.2.1.5. Decisión

La decisión del número de dedos se realiza a partir de la estimación de la longitud de los máximos. La longitud de los máximos se entiende como el mínimo entre la diferencia de un máximo y sus dos mínimos colindantes (Figura 5.22).

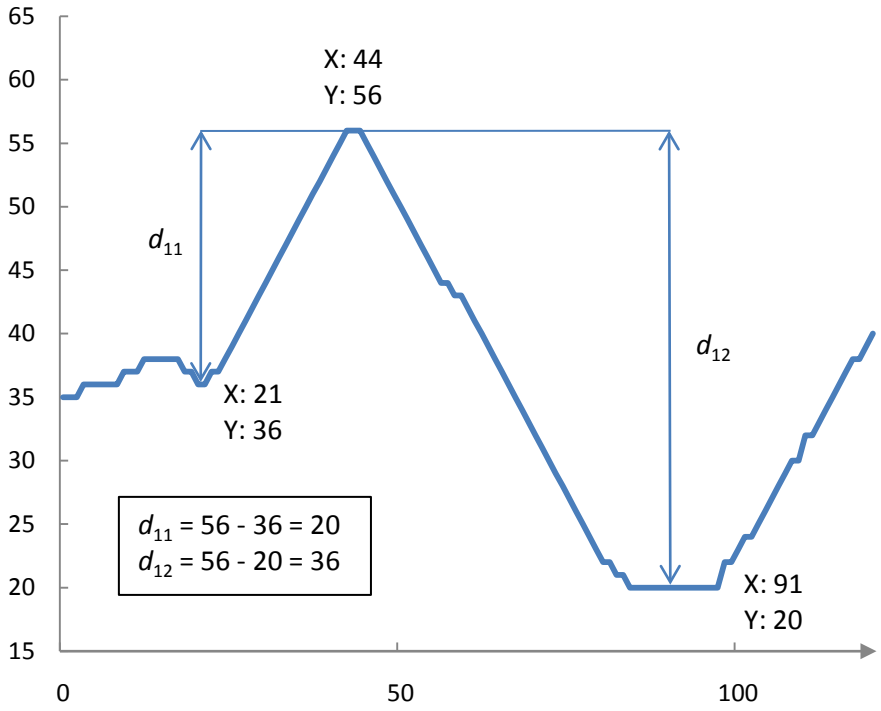


Figura 5.22. Distancia de un máximo. Para el máximo indicado se calculan las distancias d_{11} y d_{12} y se toma como longitud el valor $d_1 = \min(d_{11},d_{12})$.

Las longitudes mayores que el umbral de longitud determinado para un dedo son tomadas como dedos levantados.

Con los resultados del conteo de dedos realizados para cada *frame* se realiza la clasificación del movimiento. El método elegido para este clasificador es el que toma como resultado el conteo de dedos que se ha producido con más probabilidad a lo largo de todo el movimiento. Esto es, el resultado con mayor número de ocurrencias.

5.2.2. Combinación DCT y clasificador kNN

El método de cómputo de dedos desarrollado en el apartado 5.2.1 realiza el conteo basándose en la longitud mínima considerada para un dedo. Este valor ha sido determinado de forma experimental pero está ligado a la distancia a la que el usuario se encuentra al hacer uso del sistema. Por lo tanto, la aplicación resulta dependiente de la distancia y hay que ligar al usuario a una posición aproximadamente constante. Para tratar de mejorar esta limitación se presenta el siguiente método.

Se propone, además, usar un clasificador más robusto, como kNN. En este caso, el primer paso consiste en una transformación de los datos para compactar la información y reducir la dimensionalidad. Entre estas transformaciones, el uso de PCA o DCT suelen ser los más habituales. En este trabajo se ha estudiado el uso de la DCT por su capacidad de discriminación y el menor coste computacional respecto a PCA.

5.2.2.1. Valores de distancia geodésica

Con los valores del contorno de la imagen de distancia geodésica recorridos en el orden apropiado, se realiza la *Dicrete Cosine Transform* (DCT). Los coeficientes de la DCT expresan la secuencia de puntos finitos en términos de suma de funciones coseno oscilando a frecuencias diferentes.

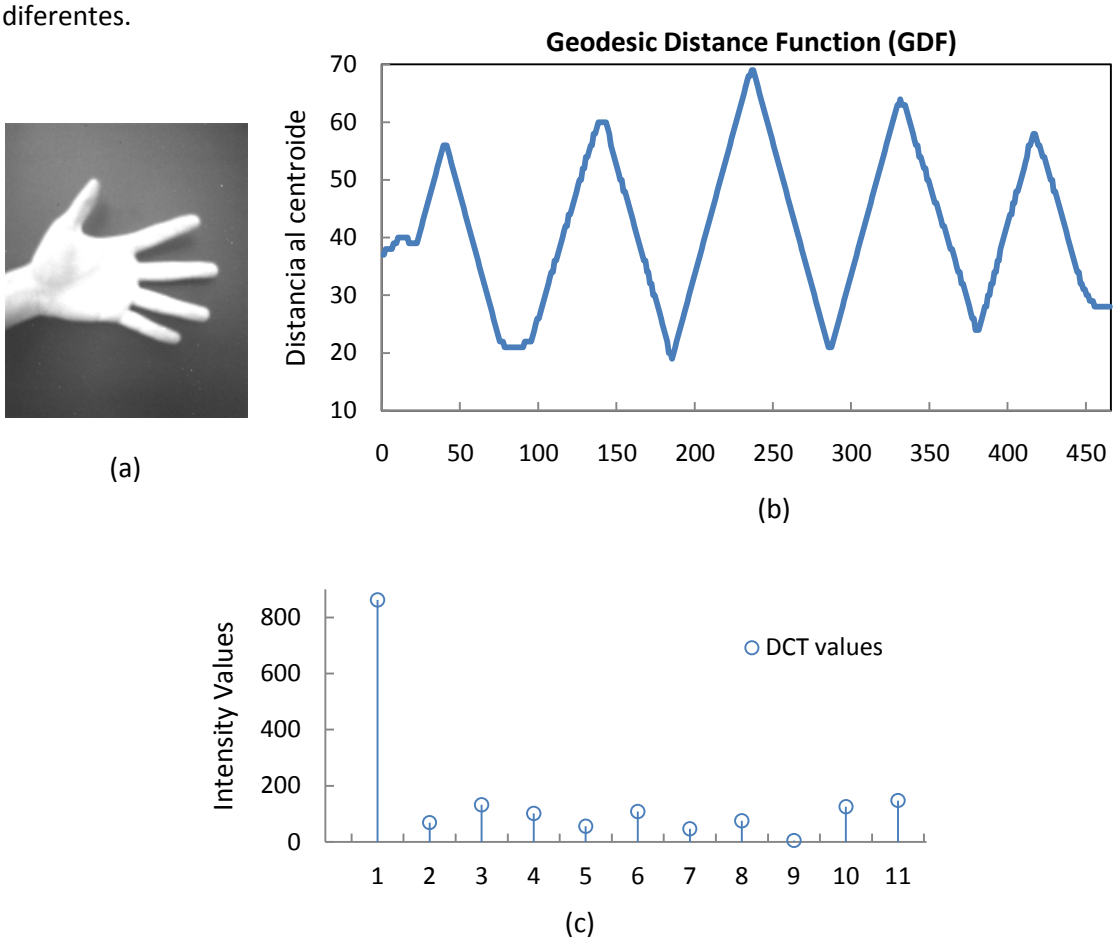


Figura 5.23. Transformada DCT. (a) Imagen capturada. (b) Función de distancia geodésica. (c) DCT de la función de distancia geodésica.

5.2.2.2. Decisión. Clasificador kNN

Para la decisión del número de dedos, es necesario efectuar un entrenamiento del sistema. Este entrenamiento consiste en la grabación previa de los gestos a detectar en diferentes posiciones y distancias a la cámara y el cómputo de los valores de la DCT correspondiente. Experimentalmente, se ha comprobado que truncar la DCT a los primeros diez términos es una práctica adecuada para reducir el número de valores y preservar los coeficientes de mayor potencia que permitan mantener una diferenciación de los distintos gestos.

El espacio muestral de las plantillas está compuesto por las sesiones de entrenamiento y se halla dividido en n clases y m prototipos. Mientras que las clases designan los diferentes gestos a detectar, los prototipos designan el número de grabaciones de cada prototipo. Se tiene, en consecuencia, $n = 6$ (conteo desde cero dedos hasta cinco) y $m = 6$ ya que se han realizado seis grabaciones para cada gesto.

La clasificación se realiza comparando los datos de la secuencia captada con las plantillas almacenadas. La plantilla con una mayor puntuación de coincidencia es la elegida como gesto realizado.

Esta clasificación es muy utilizada y es conocida con el nombre de clasificador k-*Nearest Neighbour* (kNN). El clasificador kNN permite clasificar una secuencia de datos a través de la distancia de ésta a las secuencias establecidas como plantillas. La secuencia de test es clasificada como la clase de la secuencia plantilla con menor distancia entre las k secuencias más cercanas a la secuencia de test. La elección del valor de k depende de los datos. De forma general, un valor alto de k reduce el efecto de posibles ruidos en la clasificación pero provoca que los límites entre las clases sean menos definidos. Se ha establecido experimentalmente para la clasificación un valor de $k = 1$.

La distancia entre la secuencia de test y las plantillas se define como:

$$\min_{n} \sum_{i=1}^{10} \left(S_i - p_{n,m_i}\right)^2$$

Dónde S_i es la secuencia de test y p_{n,m_i} es el prototipo m de la plantilla de clase n.

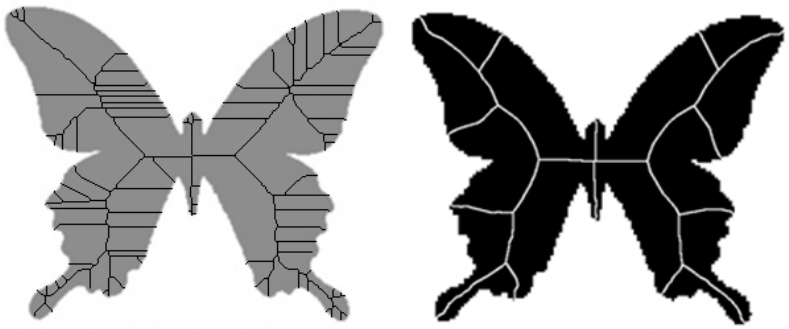
5.2.3. *End points* y *skeleton joints*

Este método de conteo de dedos consiste en localizar los *end points* y *skeleton joints* en la imagen del esqueleto morfológico de la mano. A través de estos puntos hallados por el sistema Q-Eye, se desea deducir el número de dedos levantados en el momento de capturar la imagen.

De las descripciones e imágenes mostradas en los apartados 5.1.2 y 5.1.3 se puede comprobar que los *end points* son una aproximación del extremo final del dedo mientras que las *skeleton joints* son una aproximación del extremo inicial.

5.2.3.1. Discriminación

Debido a la propia construcción del esqueleto, aparecen tanto *end points* como *skeleton joints* que no son válidos para el objetivo que se pretende. Se observa, por ejemplo, que en muchos casos el extremo final de un dedo produce en el esqueleto una bifurcación que provoca la detección de dos *end points* y una *skeleton joint*. La construcción de esqueletos más precisos que evitan este tipo de situaciones ha sido ampliamente estudiada en trabajos como [34] [35]. Este tipo de aproximaciones intentan eliminar ramas innecesarias resultado de la construcción del esqueleto a través de operaciones de *prunning* sobre las ramas que cumplen ciertos criterios. Sin embargo, implican un mayor coste computacional.



(a) (b)

Figura 5.24. Esqueleto morfológico. (a) Configuración habitual. (b) Esqueleto Bai.

Así mismo, al enmascarar la imagen para seleccionar una región de interés más pequeña que la imagen completa, se produce un corte del esqueleto morfológico y esto causa la aparición de *end points* que no son de interés.

La aparición de estos puntos conlleva la necesidad de realizar un proceso de discriminación entre los puntos obtenidos. A los puntos hallados por el sistema Q-Eye se les dará el nombre de *end points* y *skeleton joints* iniciales mientras que, una vez hecha la discriminación, se denominarán como *end points* y *skeleton joints* finales.

Las operaciones de discriminación se realizan en el procesador NIOS II, la parte digital del sistema Eye-RIS.

Discriminación de *End Points*

La discriminación de *end points* iniciales tiene como objetivo eliminar todos aquellos puntos detectados como *end points* por el Q-Eye que no corresponden con el extremo final de algún dedo.

Se eliminan aquellos *end points* que están situados en los límites de la imagen. Hay que notar que se entiende, en este punto, los límites de la imagen como aquellos definidos por la región de interés (ROI).

También son eliminados aquellos *end points* que están situados muy próximos entre sí. Este criterio es empleado para eliminar los *end points* que se producen en las bifurcaciones que se crean en el esqueleto para los extremos finales de los dedos. De todos los *end points* que estén a una distancia menor que una distancia umbral, sólo permanece el que presenta la coordenada y menor (esto es así por el funcionamiento de descarga de los puntos del sistema Q-Eye).

En la etapa final de discriminación son descartados los puntos que están situados por debajo del centroide más un cierto umbral.

En la Figura 5.25 se ilustra el proceso de discriminación de *end points*. Se produce una eliminación por hallarse el *end point* en los límites de la imagen (*end point* 6), una por posición próxima entre puntos (*end point* 2) y una por estar situado el punto por debajo del centroide (*end point* 5).

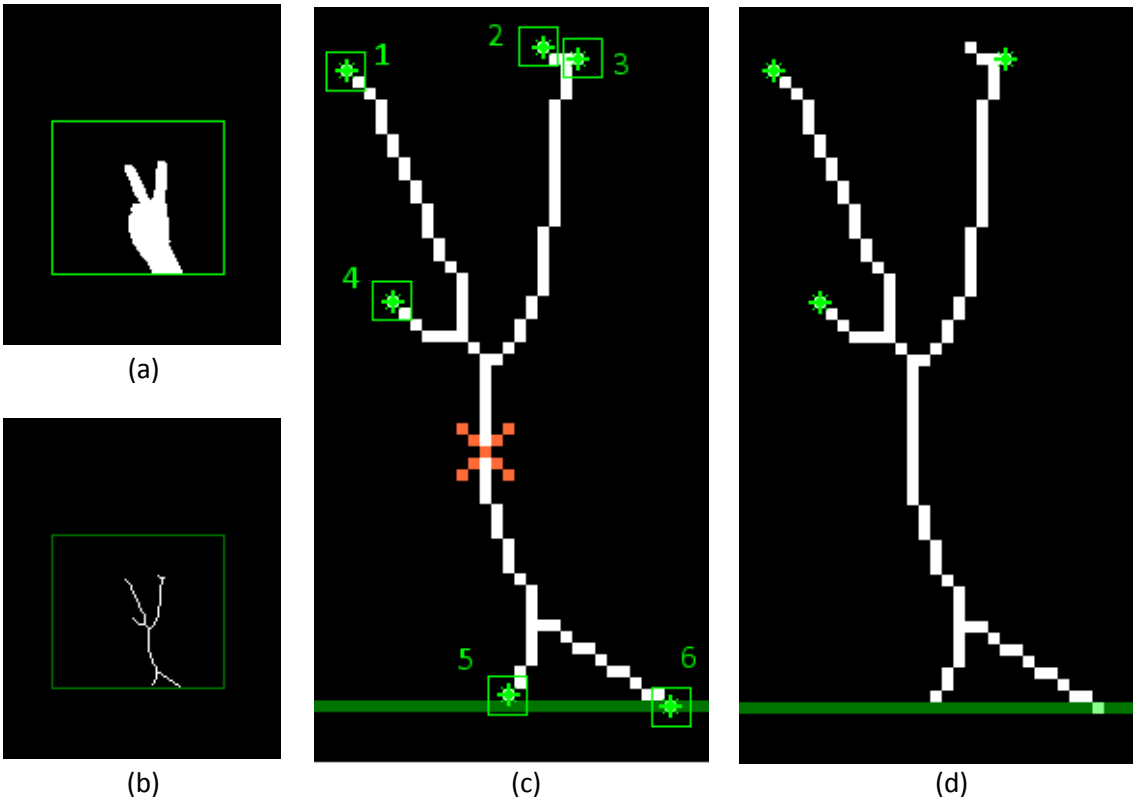


Figura 5.25. Discriminación de *end points*. (a) Imagen segmentada con ROI en verde. (b) Esqueleto morfológico. (c) *End points* iniciales (centroide en rojo). (c) *End points* finales.

Discriminación de *Skeleton Joints*

Se descartan aquellas *skeleton joints* que están situadas por debajo del centroide más un cierto umbral. Este umbral no tiene porque ser el mismo que el umbral definido para eliminar los *end points*. Sin embargo, es lógico pensar y se ha comprobado experimentalmente que el mismo valor obtiene buenos resultados.

Como último paso se eliminan las *skeleton joints* que están muy próximas a algún *end point* inicial. Es muy importante remarcar el hecho de que se mide la distancia de cada *skeleton joint* con los *end points* iniciales, es decir los devueltos en un primer momento y sin haber pasado el proceso de discriminación. Esto es así debido a que, en los casos de bifurcación del esqueleto morfológico, podría ocurrir que el algoritmo se quedara con el *end point* más alejado de la *skeleton joint* y esta última no fuera eliminada. En la Figura 5.26 se muestra un ejemplo de este caso.

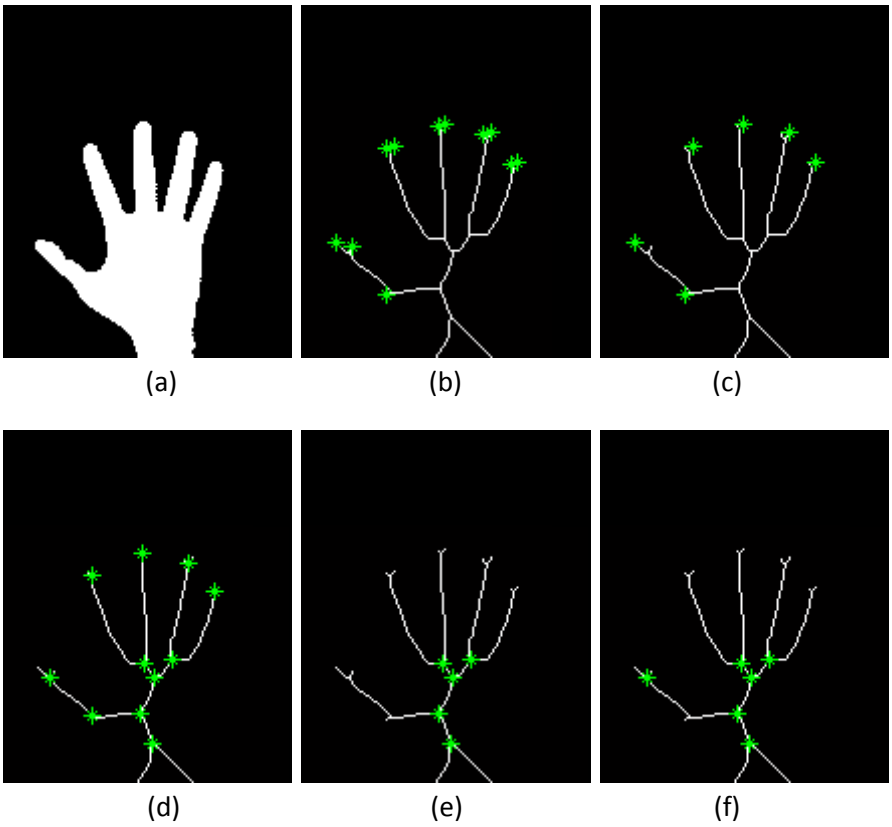
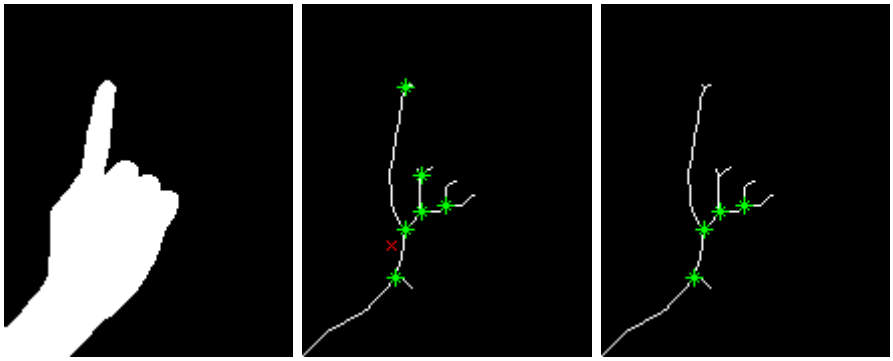


Figura 5.26. Ejemplo de no eliminación de una *skeleton joint* que debería ser eliminada por proximidad de *end points*. (a) Imagen segmentada. (b) *End points* iniciales. (c) *End points* finales. (d) *Skeleton joints* iniciales. (e) *Skeleton joints* finales al tener en cuenta los *end points* iniciales. (f) *Skeleton joints* finales al tener en cuenta los *end points* finales.

En la Figura 5.27 se muestra un ejemplo del proceso de discriminación de *skeleton joints*.



(a) (b) (c)

Figura 5.27. Discriminación de *skeleton joints*. (a) Imagen segmentada. (b) *Skeleton joints* iniciales (en verde) con centroide (en rojo). (c) *Skeleton joints* finales.

5.2.3.2. Decisión

Con los *end points* y las *skeleton joints* finales, se procede al cálculo de la longitud de cada uno de los extremos hallados. Esta longitud se entiende como la distancia mínima entre el *end point* y cada una de las *skeleton joints*.

$$\ell_k = \min_i \{d_M(EP_k, SJ_i)\}$$

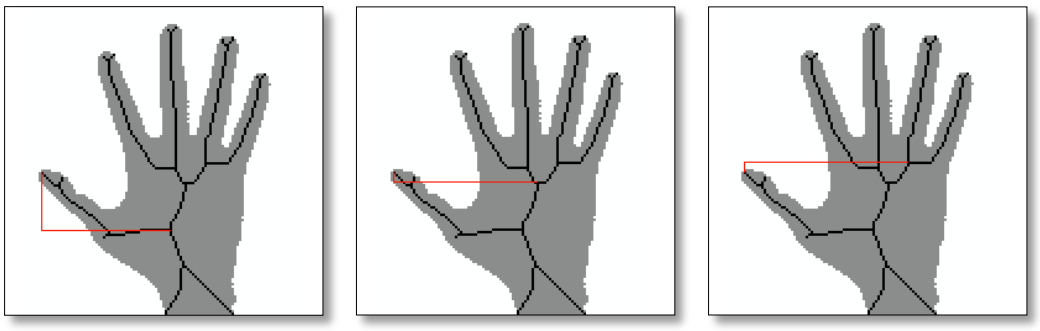
Con ℓ_k la longitud del *end point* k-ésimo EP_k, SJ_i la skeleton join i-ésima y $d_M(\cdot)$ la distancia de Manhattan.

Una distancia más precisa en términos topológicos consiste en, partiendo de un *end point*, ir recorriendo los píxeles del esqueleto hasta llegar a la primera *skeleton joint* y tomar este valor como la longitud del *end point*. Sin embargo, esta distancia necesita un uso intensivo de estudio de conectividades para determinar el recorrido a través de los puntos del esqueleto que resulta computacionalmente costoso en el NIOS II. La distancia de Manhattan ofrece una alternativa más eficiente en términos de cálculo. Es por este motivo que se ha tomado como aproximación a la distancia descrita.

La distancia de Manhattan en el espacio $\mathbb{R}^2$ entre un punto A de coordenadas (x_1, y_1) y un punto B de coordenadas (x_2, y_2) se define como:

$$d_M(A, B) = abs(x_1 - x_2) + abs(y_1 - y_2)$$

Dónde $abs(\cdot)$ denota la operación de valor absoluto.



(a) (b) (c)

Figura 5.28. Distancia del *end point*. Se computa la distancia de Manhattan del *end point* a todas las *skeleton joints* y se toma como distancia el mínimo de todos estos valores.

Las longitudes de los *end points* que superan un cierto umbral son contabilizadas como dedos.

Dado que la construcción del esqueleto morfológico resulta en una figura que depende en gran medida de la distancia a la que se encuentra la mano de la cámara, el sistema final tiene una elevada dependencia con la distancia del usuario.

Para la clasificación del movimiento, el sistema tiene en cuenta todos los conteos realizados para cada *frame* por lo que cabe la posibilidad de utilizar clasificador para obtener resultados más robustos.

5.2.4. *End points*

La dependencia del método de conteo de dedos basada en la localización de los *end points* y las *skeleton joints* con la distancia del usuario a la cámara resulta, en la práctica, muy elevada. Pequeñas variaciones en la distancia de la mano a la cámara implican cambios elevados en el tamaño de las ramas del esqueleto morfológico. Si la mano del usuario se acerca levemente a la cámara, la longitud de las ramas del esqueleto se incrementa. Esto provoca que algunas ramas residuales del esqueleto sean contadas como dedos. Si la mano se aleja de la cámara, la longitud de las ramas del esqueleto se reduce dando lugar a que algunas no sean contadas como dedos.

El siguiente método de conteo de dedos intenta solucionar el problema de la dependencia con la distancia del método anterior. La metodología es similar pero se elimina el concepto de longitud de los *end points* haciendo que la distancia a la cámara sea menos influyente.

La mayoría de errores que se cometen en el conteo de dedos basado en el método anterior son causados por el hecho de que una mínima variación en la posición del usuario respecto a la cámara implica cambios importantes en la distancia de los *end points* a las *skeleton joints*. Dado que el umbral mínimo de longitud de una rama del esqueleto morfológico para considerar que ésta corresponde a un dedo es una constante, el brazo debe permanecer a una distancia constante de la cámara en la cual este valor mínimo es el adecuado para la correcta identificación del gesto.

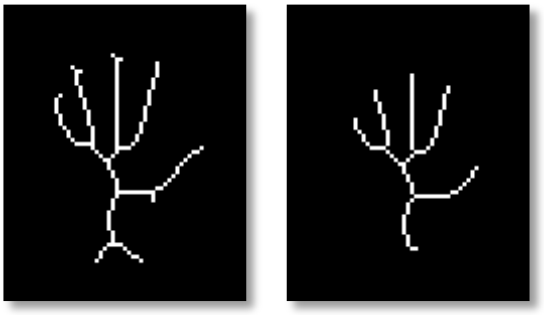
Sin embargo, esta distancia es indispensable para descartar ramas de menor longitud que se crean en la construcción del esqueleto morfológico.

5.2.4.1. Tratamiento del esqueleto morfológico

La solución que se propone consiste en ejecutar un tratamiento previo de la figura del esqueleto morfológico de tal modo que las ramas problemáticas sean eliminadas en la medida de lo posible.

El tratamiento se fundamenta en la operación morfológica de *hit and miss*. Esta operación busca ciertos patrones en el esqueleto dando como resultado una imagen en que los únicos píxeles activos son los que cumplen el patrón definido. Los patrones están definidos por medio de una matriz cuadrada de tamaño 3x3 en la que cada elemento puede tomar los valores 1 para designar un píxel que debe estar activo, 0 para el caso inactivo o *Do Not Care* (DNC) para ignorar el píxel situado en esa posición.

Se pasan una serie de ocho patrones que realizan la búsqueda de píxeles que se hallan en los extremos finales del esqueleto morfológico y los resultados son utilizados para eliminar estos puntos de la imagen original. El resultado global de este tratamiento se puede observar en la Figura 5.29.



(a) (b)

Figura 5.29. Tratamiento del esqueleto morfológico. (a) Esqueleto morfológico. (b) Primera iteración de cálculo del centroide sobre el esqueleto.

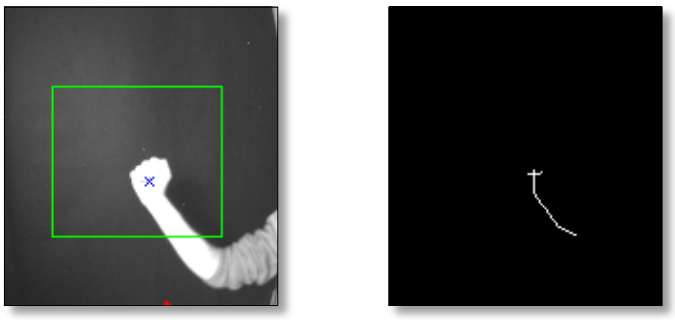
Se obtiene de esta manera una mejora de la topología del esqueleto morfológico en el contexto del conteo de dedos. Si todas las ramas residuales del esqueleto morfológico resultan eliminadas, el número de dedos extendidos corresponde con el número de *end points*. Sin embargo, aún existen algunos inconvenientes que necesitan ser resueltos. El caso del conteo de cero dedos produce una figura que cuenta, generalmente, con un único *end point* (Figura 5.30a).

Este esqueleto, además, tiene una forma muy similar al caso de conteo de un dedo (Figura 5.30b). La diferencia entre ambos radica en que el esqueleto generado por un dedo extendido es de una longitud mayor que el esqueleto que se genera en el caso de no tener ningún dedo extendido.

(a) (b)

Figura 5.30. Esqueleto morfológico tratado. (a) Conteo de cero dedos. (b) Conteo de un dedo.

Sin embargo, tratar de diferenciar estos dos gestos por medio de una distancia mínima entre las longitudes de la rama del esqueleto lleva al mismo problema que plantea el método de conteo anterior: una dependencia elevada con la distancia entre el usuario y la cámara. Además, en la construcción del esqueleto morfológico tratado, en ciertas ocasiones la primera iteración del cálculo del centroide no elimina totalmente las ramas que se pueden crear debido a las bifurcaciones, apareciendo más de un *end point*. Este es el caso que se muestra en la Figura 5.31.



(a) (b)

Figura 5.31. Conteo de cero dedos. (a) Imagen segmentada. (b) Esqueleto morfológico tratado.

5.2.4.2. Tratamiento de la imagen segmentada

Dado que la información proporcionada por el esqueleto modificado no es suficiente para extraer el número de dedos extendidos, resulta necesario combinar el esqueleto modificado con una versión de la imagen segmentada tratada de forma previa.

La idea principal del método se basa en la construcción de un *blob* que, usado de forma conjunta con el esqueleto morfológico, tenga como *end points* únicamente los dedos extendidos. Este *blob* debe contener, por tanto, información sobre el área delimitada por la palma de la mano.

Para conseguir este *blob* se realiza una operación de apertura morfológica (una erosión seguida de una dilatación) con un elemento estructurante cuadrado de dimensiones 7x7 sobre la imagen segmentada. Esta operación tiene como objetivo eliminar los dedos de la imagen segmentada.

A continuación se realiza un proceso iterativo de *pruning* hasta que no se producen cambios en la imagen. El *pruning* elimina los *end points* encontrados en la imagen. De esta forma, se evita que en la imagen tratada existan este tipo de puntos de manera que la única aportación de información en este sentido sea obtenida a través del esqueleto morfológico. Finalmente, se realiza una operación de dilatación para aumentar el área.

(a) (b)

Figura 5.32. Tratamiento de la imagen segmentada. (a) Imagen segmentada. (b) *Blob* de la palma de la mano.

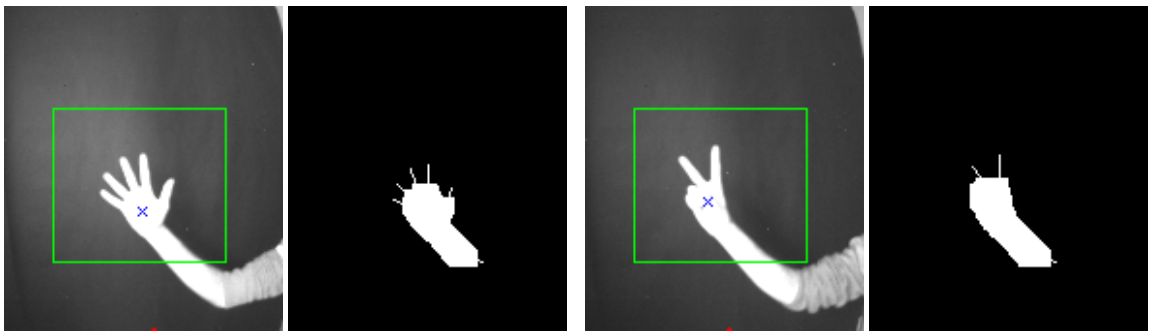
5.2.4.3. Discriminación de *end points*

Al combinar el esqueleto morfológico modificado y la imagen de la palma de la mano, se obtiene una imagen en la que las ramas del esqueleto son, principalmente, las que corresponden a los dedos extendidos (ver Figura 5.33a y Figura 5.33b).

Hay que realizar, sin embargo, un proceso de discriminación de los *end points* en que se eliminan aquellos que están situados en el borde de la imagen y en una posición inferior a la del centroide.

5.2.4.4. Decisión

El número de *end points* resultado de la discriminación se corresponde con el número de dedos levantados. En el caso de conteo de cero dedos, la imagen de la palma de la mano enmascara los *end points* que se pudieran crear en la construcción del esqueleto.



(a) (b)

Figura 5.33. Resultado de la combinación de esqueleto e imagen de la palma de la mano. (a) Conteo de cinco dedos. (b) Conteo de dos dedos.

5.2.5. Comentario final

A lo largo de la sección 5.2 se han mostrado cuatro métodos diferentes de conteo de los dedos de la mano.

El primer método presenta una gran robustez en la clasificación del gesto realizado. Sin embargo, conlleva cierta dependencia en la posición del usuario. Además, debido a que se trata de un procedimiento intensivo en cálculo y en análisis de las imágenes tiene un coste computacional elevado. Su implementación en el procesador de plano focal da como resultado un conteo que no es realizable en tiempo real.

El segundo método resuelve el problema de la restricción en la posición del usuario presente en el primer método manteniendo una robustez similar en la clasificación. Sin embargo, exige la transformación al dominio de la *Discrete Cosine Transform*. Este proceso conlleva a un aumento aún más agresivo del coste computacional por lo que, en términos de uso en tiempo real, resulta más impracticable que el primer método.

El tercer método, aunque es el más simple y presenta una robustez en la clasificación menor que los dos métodos anteriores, es un algoritmo que no comporta una gran carga computacional. Sin embargo tiene una dependencia con la posición mucho mayor que la del primer método. Para mejorar la robustez en la clasificación del movimiento, se clasifica el gesto utilizando un método basado en el resultado con más probabilidad de los valores contados.

El cuarto método incluye las ventajas del método tercero pero además elimina en gran medida la dependencia con la posición del usuario a la cámara. Por este motivo resulta la mejor opción para la ejecución en tiempo real.

Método	Robustez (Probabilidad de detección)	Velocidad (µs)
Distancia geodésica	97,00%	138.170
Conteo DCT	97,50%	(*)
End points y *skeleton joints*	65,29%	927,81
End points	95,91%	739,06

(*) Dato no disponible. Implementación en MATLAB.

Tabla 5.1. Robustez-velocidad de los métodos de conteo.

Capítulo 6

Detección de gestos dinámicos

Los gestos dinámicos reconocidos por el sistema son movimientos que consideran las cuatro direcciones cardinales posibles.

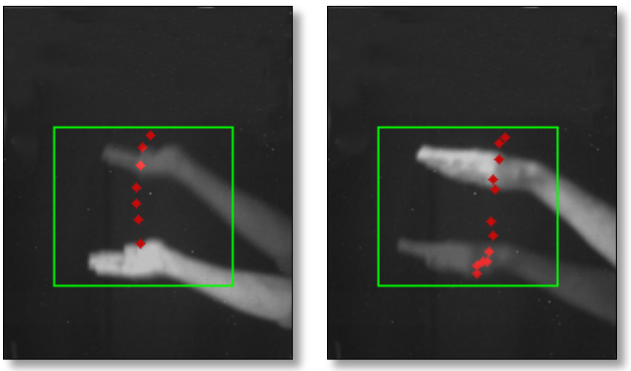
Mientras que los movimientos ascendentes (que aparecen por la parte inferior de la pantalla para desaparecer por la parte superior) o descendentes (aparecen por la parte superior y desaparecen por la inferior) se utilizan para abrir y cerrar menú respectivamente, los movimientos hacia derecha o izquierda pueden ser empleados, por ejemplo, para navegación hacia delante y atrás.

Para la clasificación de los gestos dinámicos se tiene en cuenta únicamente la evolución del centroide por lo que, si el movimiento ha sido clasificado como dinámico, el cálculo del número de dedos no se realiza.

Hay que notar que el cálculo del centroide se realiza siguiendo el mismo procedimiento que el mostrado en la sección 4.4.

6.1. Abrir/Cerrar menú

Si la evolución del centroide ha sido tal que la coordenada y ha decrecido más que un cierto umbral se considera que el movimiento realizado es un movimiento de abrir menú. Se recuerda que en el movimiento de abrir menú la mano aparece por la zona inferior de la pantalla y desparece por la parte superior.



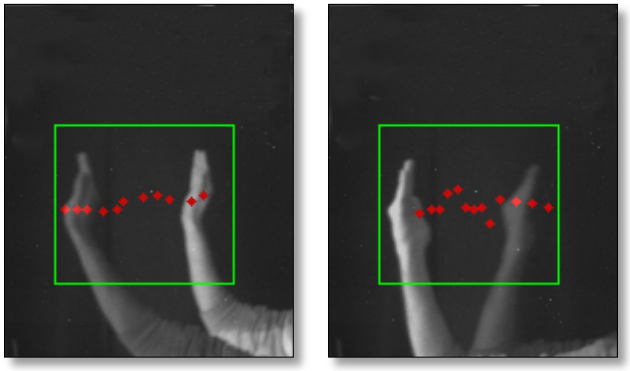
(a) (b)

Figura 6.1. Movimientos dinámicos. En rojo, la evolución del centroide.
(a) Abrir menú. (b) Cerrar menú.

Si la coordenada y ha crecido más que un determinado umbral se determina un movimiento de cerrar menú.

6.2. Desplazamiento hacia derecha/izquierda

Si el centroide ha seguido una evolución en que la coordenada x ha decrecido por encima de un cierto umbral el movimiento es un gesto de desplazamiento hacia la derecha.



(a) (b)

Figura 6.2. Movimientos dinámicos con evolución del centroide.
(a) Movimiento hacia la derecha. (b) Movimiento hacia la izquierda.

Si la coordenada x ha crecido más que un cierto umbral, se establece un movimiento de desplazamiento hacia la izquierda.

Capítulo 7

Resultados

En la decisión final del gesto realizado se tiene en cuenta la totalidad de la evolución del movimiento. Esto significa que la decisión tiene lugar una vez se considerada el movimiento finalizado. En el caso del conteo de dedos, esta forma de proceder aumenta la probabilidad de acierto del número de dedos extendidos ya que se tiene un mayor espacio muestral. La solución implementada ordena los conteos realizados por orden descendiente de número de ocurrencias y toma como gesto el resultado con mayor probabilidad en la secuencia de análisis.

Los resultados que se muestran han sido tomados durante la ejecución en tiempo real de la aplicación y se ha considerado un espacio muestral de 181 movimientos. Para el conteo de dedos, al tener un resultado parcial para cada imagen, se han analizado un total de 2800 muestras.

El resultado individual del conteo de dedos para las *frames* es el que se muestra en la Tabla 7.1.

Gesto	Detección correcta	Detección incorrecta
0 Dedos	100,00%	0,00%
1 Dedo	100,00%	0,00%
2 Dedos	99,75%	0,25%
3 Dedos	96,66%	3,34%
4 Dedos	91,15%	8,85%
5 Dedos	87,91%	12,09%

Tabla 7.1. Conteo de dedos para cada *frame*.

Se observa una tendencia creciente en la probabilidad de detección incorrecta a medida que aumenta el número de dedos extendidos. Esto es debido, en el caso general, a que en el proceso de desaparición de la mano de la región de interés (ROI), resulta inevitable que se realice algún conteo cuando algún *end point* ha desaparecido. En el caso particular del conteo de cinco dedos también se encuentra la situación de que, en ciertas ocasiones, la operación de apertura que se realiza sobre la imagen segmentada no logra eliminar completamente los dedos. Esto afecta sobre todo al conteo del dedo pulgar que, al tener la rama del esqueleto menor, queda oculta por la imagen segmentada del dedo.

En la Tabla 7.2 se muestra el porcentaje de detecciones correctas de los gestos devueltas por el sistema.

Gesto	Detección correcta	Detección incorrecta	Gesto no reconocido	Secuencias analizadas
0 Dedos	85,00%	0,00%	15,00%	20
1 Dedo	100,00%	0,00%	0,00%	16
2 Dedos	100,00%	0,00%	0,00%	16
3 Dedos	100,00%	0,00%	0,00%	15
4 Dedos	93,33%	0,00%	6,67%	15
5 Dedos	100,00%	0,00%	0,00%	15
Derecha	88,46%	7,69%	3,85%	26
Izquierda	100,00%	0,00%	0,00%	23
Arriba	82,36%	0,00%	17,64%	17
Abajo	83,33%	0,00%	16,67%	18

Tabla 7.2. Porcentaje de acierto del sistema.

El porcentaje de gestos no reconocidos en los movimientos estáticos de conteo de dedos es debido a una decisión incorrecta del sistema en la taxonomía del movimiento. Una fase de retorno a la posición de reposo de la mano realizada a gran velocidad (esto es, la mano es retirada de la ROI de forma muy brusca) puede causar que el movimiento sea clasificado como dinámico y, por lo tanto, que no sea reconocido como ninguno de los posibles movimientos.

El caso general de no reconocimiento del gesto de desplazamiento hacia la derecha es debido a que el recorrido que debe realizar la mano es bastante amplio y en ocasiones, cuando la mano está a punto de desaparecer de la ROI entra parte del hombro por el otro lado. Esto causa una variación en el centroide que provoca que el movimiento no sea reconocido correctamente. Las detecciones incorrectas son debidas a confusiones con el movimiento de desplazamiento hacia abajo. Esto es debido a que, al ser el recorrido del gesto bastante amplio, la mano tiende a desplazarse en dirección hacia abajo en la parte final del recorrido.

Tanto para el caso de movimiento hacia arriba como hacia abajo, se tiene un porcentaje de gestos no reconocidos elevado. Esto es causado, en el caso general, porque en el caso de los movimientos dinámicos, el algoritmo de cálculo del centroide de la mano (sección 4.4) no tiene el comportamiento deseado al encontrarse la mano de perfil. Esto provoca que en ocasiones no se separen los *blobs* del brazo y de la mano. Al tomar el valor del centroide del *blob* general, el brazo tiene una mayor influencia y desplaza el centroide a un recorrido con menor movimiento. Esto causa que al comprobar la evolución, la coordenada y no supere el umbral necesario.

Capítulo 8

Conclusiones y perspectivas

8.1. Conclusiones

A lo largo de esta memoria se ha mostrado la construcción de un sistema capaz de discernir entre un número finito de gestos simples en un entorno controlado. Para desacoplar el problema de la segmentación del problema del análisis de las imágenes se ha simplificado la binarización limitando el escenario a espacios con fondo uniforme oscuro y con iluminación constante del brazo.

Se han implementado y estudiado varios métodos que hacen posible el conteo de dedos para la interacción entre un humano y una computadora a partir del análisis de imágenes capturadas por una cámara que incorpora un procesador de plano focal. Se ha realizado, además, una justificación acerca de la conveniencia de uso de cada uno de los métodos.

El sistema también permite la identificación de movimientos dinámicos en las cuatro direcciones principales.

Durante la realización de los métodos se han puesto en relieve las limitaciones y las ventajas del uso de una cámara con procesador de plano focal. Mientras que para las primeras fases de análisis de las imágenes ésta tiene un comportamiento muy eficiente en términos de velocidad, en las fases de tratamiento de datos y decisión el procesador digital presenta un comportamiento más lento cuando se ven involucrados algoritmos complejos para ofrecer respuesta en tiempo real.

Se ha realizado un método que permite realizar en tiempo real la distinción entre movimientos estáticos y dinámicos. Este bloque permite liberar de carga al procesador efectuando un análisis diferente de las imágenes según el tipo de movimiento realizado por el usuario.

8.2. Futuras líneas de proyecto

En esta sección se dan algunas posibles extensiones al trabajo realizado en este proyecto. Las futuras líneas de proyecto se pueden agrupar en tres categorías:

La primera categoría incluye las mejoras en la segmentación espaciotemporal. En lo referente a la segmentación espacial, en la binarización actual existe fijado un valor umbral constante cuyo funcionamiento se ha comprobado de forma correcta experimentalmente para todas las

situaciones estudiadas cuando se realiza la sustracción del fondo. Una solución más elaborada consistiría en elegir este valor de manera independiente para cada imagen capturada. Ello se podría hacer a partir del cálculo del histograma y con algún algoritmo que decida el nivel óptimo del *threshold*. Sin embargo, no hay que perder de vista que, para que la respuesta del sistema se produzca en tiempo real, existe un compromiso entre el nivel de sofisticación (para el caso general, a mayor sofisticación, mayor coste computacional y por tanto, mayor inversión de tiempo) y unos resultados aceptables. Mejorar el proceso de segmentación espacial es un proyecto muy ambicioso que tendría como resultado el funcionamiento correcto del sistema en espacios más reales y/o generales. En cuanto a la segmentación temporal, un sistema capaz de detectar el inicio y final de secuencia sin necesidad de que se produzca la retirada de la mano resulta de una mayor usabilidad.

La segunda categoría es el cálculo de parámetros. El método de cálculo actual del centroide resulta en ocasiones poco efectivo (es el caso de los movimientos dinámicos en que la mano aparece en una posición de perfil). Se podrían incluir, además, nuevos parámetros que permitieran una decisión del movimiento con una mayor confianza así como la implementación de un número mayor de gestos.

La tercera y última categoría versa sobre los puntos de visión. Una idea que cobra cada vez más importancia, es el análisis de la situación a partir de más de un ángulo de visión. La combinación de imágenes captadas desde diferentes posiciones sobre una misma situación permite extraer información que una única cámara no puede conseguir. Si se complementa la cámara situada en la parte superior de la pantalla con otra situada, por ejemplo, en alguno de los lados (a izquierda o derecha del usuario) se tendría, además de la información frontal, información de profundidad. Esto permitiría tener una gran precisión en la detección de nuevos movimientos como por ejemplo el de señalar.

Referencias bibliográficas

[1] http://cenit-vision.org.

[2] A. Rodríguez-Vázquez et al., “The Eye-RIS CMOS Vision System,” *Analog Circuit Design: Sensors, Actuators and Power Drivers*, H. Casier, M. Steyaert, and A. H. M. Van Roermund, Eds., pp. 15-32. Springer Netherlands, 2008.

[3] A. Rodríguez-Vázquez et al., “CMOS Architectures and Circuits for High-Speed Decision-Making from Image-Flows,” *Proceedings of the Society of Photographic Instrumentation Engineers (SPIE)*, Vol. 6940, pp. 69402F-69402F-10, Apr. 2008.

[4] “Eye-RIS v1.2 Hardware Description,” Anafocus, 2008.

[5] “Eye-RIS v1.3 Hardware Description,” Anafocus, 2009.

[6] A. Frías Velázquez, “Design and implementation of real-time image processing algorithms for FP camera,” *Master Thesis*, Technical University of Catalonia, Summer 2008.

[7] A. Frías Velázquez and J. R. Morros, "Gray-Scale Erosion Algorithm Based on Image Bitwise Decomposition: Application to Focal Plane Processors," *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 845-848, Apr. 2009.

[8] A. Frías Velázquez and J. R. Morros, "Histogram Computation Based on Image Bitwise Decomposition," *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2009.

[9] F. Karray, M. Alemzadeh, J. A. Saleh and M.N. Arab, “Human-Computer Interaction: Overview on State of the Art,” *International Journal on Smart Sensing and Intelligent Systems*, Vol. 1, No. 1, Mar. 2008.

[10] R. A. Bolt, “Put-That-There: Voice and Gesture at the Graphics Interface,” *ACM SIGGRAPH Comput. Graph.*, Vol. 14, No. 3, pp. 262–270, 1980.

[11] A. Jaimes and N. Sebe, “Multimodal human-computer interaction: A survey,” *Computer Vision and Image Understanding*, Vol. 108, pp. 116-134, 2007.

[12] H. Gunes and M. Piccardi, “Bi-modal emotion recognition from expressive face and body gestures,” *Journal of Network and Computer Applications*, Vol. 30, pp 1334-1345, 2007.

[13] L.E. Sibert and R.J.K. Jacob, “Evaluation of eye gaze interaction”, *Conference of Human-Factors in Computing Systems*, pp 281-288 (2000).

[14] S. Mitra and T. Acharya, “Gesture Recognition: A Survey”, *IEEE Transactions on Systems, Man, and Cybernetics-Part C: Applications and Reviews*, Vol. 37, No. 3, pp. 311-324, May 2007.

[15] Y. Fang, K. Wang, J. Cheng and H. Lu, “A Real-Time Hand Gesture Recognition Method,” *IEEE International Conference on Multimedia and Expo*, pp. 995-998, Jul. 2007.

[16] J. Triesch and C. Von der Malsburg, “A Gesture Interface for Human-Robot Interaction,” *Proceedings of 4th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 546-551, Apr. 1998.

[17] J. Davis and M. Shah, “Gesture Recognition,” Technical Report CS-TR-93-11, Department of Computer Science, University of Central Florida, 1993.

[18] A. Signoriello, “Hand Feature Analysis for Gestural Interfaces,” Final Project, Technical University of Catalonia, 2009.

[19] M. Kato, Y. Chen and G. Xu, “Articulated Hand Tracking by PCA-ICA Approach,” *Proceedings of International Conference on Automatic Face and Gesture Recognition*, pp. 329-334, Apr. 2006.

[20] P. Goh and E. Holden, “Dynamic Fingerspelling Recognition Using Geometric and Motion Features,” *IEEE International Conference on Image Processing*, pp. 2741–2744, 2006.

[21] F. K. H. Quek, “Eyes in the Interface,” *Image Vision Computing*, Vol. 13, No. 6, pp. 511-525, Jul. 1995.

[22] W. T. Freeman and M. Roth, “Orientation Histograms for Hand Gesture Recognition,” *Proceedings of IEEE International Workshop on Automatic Face and Gesture Recognition*, pp. 296-201, Jun. 1995.

[23] H. Zhou, D. J. Lin and T.S. Huang, “Static Hand Gesture Recognition based on Local Orientation Histogram Feature Distribution Model,” *Proceedings of IEEE Conference on CVPR*, pp. 161-169, 2004.

[24] F. Chen, C. Fu and C. Huang, “Hand Gesture Recognition Using a Real-Time Tracking Method and Hidden Markov Models,” *Image and Video Computing*, Vol. 21, No. 8, pp. 745-758, Aug. 2003.

[25] M. H. Yang and N. Ahuja, “Extraction and Classification of Visual Motion Patterns for Hand Gesture Recognition,” *Proceedings of the IEEE CVPR*, pp. 892-897, 1998.

[26] L. Gui, J. Thiran and N. Paragios, “Finger-Spelling Recognition Within a Collaborative Segmentation/Behavior Inference Framework,” *Proc. of the 16th European Signal Processing*, 2008.

[27] J. Alon, V. Athitsos, Q. Yuan and S. Sclaroff, “Simultaneous Localization and Recognition of Dynamic Hand Gestures,” *Proceedings of IEEE Workshop Motion and Video Computing*, Vol. II, pp. 254-260, 2005.

[28] J. Alon, V. Athitsos, Q. Yuan and S. Sclaroff, “A Unified Framework for Gesture Recognition and Spatiotemporal Gesture Segmentation,” *IEEE Transactions on PAMI*, Vol. 31, No. 9, pp. 1685-1699, Sept. 2009.

[29] M. Yeasin and S. Chaudhuri, “Visual understanding of dynamic hand gestures,” *Pattern Recognition*, Vol. 33, pp. 1805-1817, 2000.

[30] E. Ong and R. Bowden, “A boosted Classifier Tree for Hand Shape Detection,” *Proc. of 6th IEEE International Conference on Automatic Face and Gesture Recognition*, pp. 889-894, May 2004.

[31] D. Castilla López et al., “E.8.2.7. Evaluación de la interfaz de usuario. Usabilidad en el lenguaje gestual: línea de base mediante el test de usuarios,” *Entregable VISION*, Sept. 2008.

[32] D. C. Fernandes, “Uma revisão dos modelos da memória de reconhecimento e seus achados empíricos,” *PSIC – Revista de Psicologia da Vetor Editora*, Vol. 6, No. 2, pp. 23-32, Dez. 2005.

[33] C. Stauffer, W. E. L. Grimson, "Adaptive Background Mixture Models for Real-Time Tracking," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 2, pp. 246-252, 1999.

[34] X. Bai, L. J. Latecki, and W. Y. Liu, “Skeleton Pruning by Contour Partitioning with Discrete Curve Evolution,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 29, No. 3, pp. 449-462, 2007.

[35] X. Bai and L. J. Latecki, “Discrete Skeleton Evolution,” *International Conferences on Energy Minimization Methods in Computer Vision and Pattern Recognition* (EMMCVPR), pp. 362-374, 2007.