UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

PROYECTO FIN DE CARRERA

Extracción y gestión de regiones de interés en contenido audiovisual

Helena González Casero

Julio 2008

Extracción y gestión de regiones de interés en contenido audiovisual

AUTOR: Helena González Casero
TUTOR: Víctor Valdés López

Grupo de Tratamiento de Imágenes
Dpto. de Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Julio de 2008

PROYECTO FIN DE CARRERA

Título: *Extracción y gestión de regiones de interés en contenido audiovisual*

Autor: Dª. Helena González Casero

Tutor: D. Víctor Valdés López

Tribunal:

Presidente: Jesús Bescós Cano

Vocal: Simone Santini

Vocal secretario: José M. Martínez Sánchez

Fecha de lectura:

Calificación:

Palabras Clave

Segmentación de imágenes, regiones de interés, descriptores visuales, recuperación de imágenes basada en contenido, seguimiento de objetos.

Resumen

El principal objetivo de este proyecto es explorar las posibilidades que ofrece la caracterización de regiones de una imagen mediante descriptores sencillos (en términos de coste computacional). Para ello se llevará a cabo el diseño e implementación de un sistema que permita, a partir de una segmentación previa y de forma automática, la extracción y almacenamiento de las regiones que componen una imagen así como sus descriptores asociados. Posteriormente, se estudiarán e implementarán dos aplicaciones concretas, basadas en la aplicación del sistema de gestión de regiones y de extracción de descriptores desarrollada anteriormente, como ejemplo y demostración de las posibilidades que ofrece el sistema implementado: recuperación de imágenes basada en consulta y seguimiento de objetos en secuencias de vídeo. Se llevará a cabo una evaluación del rendimiento de ambas aplicaciones comparando los resultados obtenidos con sistemas de referencia. Finalmente, se expondrán las conclusiones y se propondrán las líneas de trabajo futuras.

Abstract

The main objective of this PFC is the exploration of the possibilities that image region characterization based on simple descriptors (in terms of computational performance) can offer. For this purpose, starting from an initial region segmentation, a system for automatic region extraction and characterization as well as associated descriptors calculation will be designed and implemented. Subsequently two possible applications based on the developed system will be studied and implemented as an example of the possibilities that region characterization offers: query-based image retrieval and object tracking in video sequences. A performance evaluation of the proposed applications as well as result comparison with existing reference systems will be carried out. Finally some conclusions together with some future directions will be exposed.

Agradecimientos

Quisiera agradecer a las siguientes personas su implicación, directa o indirecta, en este Proyecto de Fin de Carrera:

En primer lugar, quisiera agradecer a José María Martínez por la confianza depositada en mí al ofrecerme este proyecto y junto a Jesús Bescós por mantener siempre la puerta de su despacho abierta a cualquier duda que nos haya podido surgir durante la carrera.

Este proyecto no habría sido posible sin la valiosa ayuda y orientación de mi tutor, Víctor Valdés, al que estoy muy agradecida.

Asimismo, quisiera agradecer a todos los miembros del GTI por el buen ambiente dentro de éste y por encontrarse siempre receptivos ante cualquier problema que ha podido surgir durante este tiempo.

En el plano personal, me gustaría comenzar dando las gracias a mis padres, Laura y Juan, y a mi hermano Alejandro por su apoyo incondicional en todo lo que hago y en especial por haber puesto todo su esfuerzo para que pudiera estudiar en Madrid la carrera de telecomunicaciones.

Gracias a Javier, por hacer que cada mañana me levante con ilusión, por su interés y ayuda en este proyecto y por conseguir arrancarme una sonrisa tras las largas horas de trabajo.

A mis amigos por contagiarme su optimismo y animarme a seguir adelante.

Por último, pero no por ello menos importante, quiero agradecer la ayuda a mis compañeros de carrera, porque todo fue más fácil y divertido gracias a ellos.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

En este primer capítulo vamos a introducir muy brevemente un resumen de los aspectos principales de este proyecto. En primer lugar, mostraremos las motivaciones que nos han llevado a realizar este trabajo; en segundo lugar, explicaremos cuáles son los objetivos que se han establecido sobre él; por último, comentaremos las secciones en las que está dividida esta memoria, junto a una descripción somera de su contenido.

1.1 Motivación

La cantidad de información audiovisual disponible en formato digital está alcanzando cifras verdaderamente elevadas debidas en gran parte al uso masivo de Internet. Como dato orientativo, *Flickr*, uno de los numerosos sitios de la red que permiten subir y compartir fotos, contaba en Noviembre de 2007 con 2 billones de fotos y una media entre 2 y 3 millones de fotos cargadas por sus usuarios diariamente[1]. El uso de imágenes y vídeo está creciendo también en aplicaciones de seguridad, comercio electrónico o medicina. Este creciente interés precisa diseñar sistemas que nos permitan describir los diferentes tipos de información multimedia para posibilitar su búsqueda y clasificación puesto que la gran cantidad de contenido existente hace que la búsqueda de información se convierta en una tarea cada vez más difícil.

Los descriptores visuales se definen como conjuntos de atributos que es posible extraer o calcular a partir de una imagen realizando una serie de operaciones sobre la misma y que nos permiten conocer el contenido de imágenes y vídeos. Mediante la extracción y gestión de estos descriptores podemos tener un amplio conocimiento de los objetos y eventos presentes en un vídeo o imagen lo que nos facilitará una búsqueda rápida y eficiente del contenido audiovisual de forma automática.

Como veremos, existen numerosos tipos de descriptores por lo que la elección de los adecuados para la caracterización del contenido multimedia no es una tarea superficial y resulta esencial para la utilización eficaz de este tipo de archivos.

[1] Fuente: http://www.techcrunch.com/

1.2 Objetivos

El principal objetivo de este PFC consiste en realizar una caracterización del contenido multimedia visual (imágenes y vídeos) a través de la implementación de un conjunto de descriptores sencillos extraídos a partir de la división de la imagen en regiones espaciales. Demostraremos la utilidad de dicha caracterización mediante dos aplicaciones básicas: recuperación de imágenes en bases de datos y un sistema de reconocimiento y seguimiento de objetos en una secuencia de vídeo.

Como primer paso realizaremos la extracción de las regiones en las que se divide una imagen o cuadro de vídeo (*frame*) mediante la aplicación de una etapa de segmentación, proceso a través del cual se obtiene la división de la imagen en distintas regiones espaciales atendiendo a un criterio concreto (color, forma,…).

Implementaremos una serie de descriptores visuales sencillos que permitan caracterizar el contenido de imágenes y vídeos. En los últimos años han aparecido numerosos estudios que tratan de describir el contenido multimedia de forma automatizada; estudiaremos los más significativos dentro del marco del estado del arte. Nuestro objetivo será demostrar que con nuestro sistema, basado en un conjunto de descriptores relativamente sencillos, puede obtenerse una caracterización del contenido comparable a otras técnicas comúnmente utilizadas.

Una vez desarrollado el sistema que permite caracterizar las regiones en las que se divide la imagen, se considerarán dos posibles aplicaciones: la recuperación de imágenes en bases de datos y una segunda aplicación del sistema para seguimiento de objetos en una secuencia de vídeo que servirán como ejemplo de las potenciales aplicaciones del sistema de caracterización propuesto.

1.3 Organización de la memoria

La memoria de este proyecto está estructurada en cinco capítulos. Veamos una breve descripción de los temas desarrollados en cada capítulo:

- **Capítulo 1**: Introducción, objetivos y motivación del proyecto.
- **Capítulo 2**: Diseño y desarrollo de un sistema de caracterización del contenido audiovisual. Breve análisis de las técnicas y tecnologías utilizadas en la caracterización del contenido multimedia.
- **Capítulo 3**: Aplicación del sistema de caracterización implementado a la recuperación de imágenes en bases de datos. Breve análisis de las técnicas y tecnologías empleadas en la recuperación de imágenes.
- **Capítulo 4**: Aplicación del sistema de caracterización implementado al seguimiento de objetos en secuencias de vídeo. Breve análisis de las técnicas y tecnologías empleadas en los sistemas de seguimiento de objetos.
- **Capítulo 5**. Conclusiones sobre las aplicaciones desarrolladas. Relación de posibles líneas futuras de desarrollo y mejoras del sistema.

Adicionalmente, se incluyen tres anexos en la memoria:

- **Anexo A**: Guía de las clases implementadas.
- **Anexo B**: Diseño del ground-truth de imágenes.
- **Anexo C**: Pruebas efectuadas para la obtención del umbral β.

2 Caracterización del Contenido Audiovisual

2.1 Introducción

La unidad mínima de representación de imágenes es el píxel; sin embargo, el análisis de los píxeles de forma aislada no nos aporta gran información sobre el contenido de una imagen. Es necesario establecer cierta conexión entre los píxeles de la imagen que permita diferenciar las formas que en ella aparecen y así caracterizar su contenido.

Una forma adecuada de organizar los píxeles que componen una imagen es agruparlos en regiones atendiendo a un criterio concreto como pueda ser su color, textura, forma, etc. De esta forma se imita el comportamiento del sistema visual humano cuya percepción de una imagen tiende a distinguir las distintas formas que aparecen. Esta manera de organizar los píxeles permite, una vez localizadas las regiones, caracterizarlas, clasificarlas, etiquetarlas según su importancia y/o definir Regiones de Interés (ROIs) entendidas como un conjunto de píxeles dentro de una imagen o *frame* de un video que proporcionan una información relevante. Existen muchos trabajos publicados que exploran las posibilidades de definir regiones de interés dentro de una imagen. La motivación de este PFC parte precisamente del trabajo realizado por Víctor Valdés López sobre la adaptación de contenido multimedia basada en regiones de interés [1].

En el presente documento, nos limitaremos a obtener y caracterizar las regiones que componen la imagen, sin realizar ninguna valoración en un principio sobre su importancia dentro de la imagen. Para realizar la extracción de las regiones, contaremos con una segmentación previa tras la cual, una vez se han obtenido y almacenado las regiones que componen la imagen, aplicaremos el conjunto de descriptores desarrollados para caracterizar dichas regiones. Además llevaremos a cabo un análisis de los descriptores propuestos, estudiando cuáles resultan más útiles en la métrica de comparación de regiones.

2.2 Estado del Arte

Como se ha dicho anteriormente, las representaciones basadas en regiones ofrecen una forma de realizar un primer nivel de abstracción que nos permite reducir el número de elementos a procesar y además obtener cierta información ‘semántica’ sobre el contenido de la imagen lo que supone un claro avance respecto a la representación clásica basada en

píxel. El preprocesado de la imagen en regiones está presente en numerosas aplicaciones. Un ejemplo de su uso se puede observar en las versiones más recientes de los estándares de codificación internacionales JPEG-2000 [2] y MPEG-4 SVC [3] que permiten la codificación selectiva de regiones según su importancia dentro de una imagen o vídeo. Este tratamiento permite, por ejemplo, reducir el tamaño necesario para el almacenamiento o transmisión del contenido reduciendo la calidad en regiones consideradas como de menor importancia. La naturaleza de lo que se considera como una región o regiones de interés (ROIs) dependerá de la aplicación; por ejemplo, pueden ser regiones de interés las caras humanas o los símbolos grafológicos. En cualquier caso, son necesarias herramientas que nos permitan caracterizar y clasificar las diferentes regiones definidas en una imagen, tarea para la cual se hace uso de los descriptores visuales.

Los descriptores visuales son el primer paso para poder encontrar la conexión entre los píxeles contenidos en una imagen digital y aquello que los humanos recordamos después de haber observado durante unos minutos una imagen o un conjunto de las mismas. Es posible clasificarlos en dos grandes grupos:

1. *Descriptores de información general*: contienen descriptores de bajo nivel, proporcionando una descripción acerca del color, formas y regiones, texturas y movimiento.
2. *Descriptores de información de dominio específico*: proporcionan información acerca de los objetos y eventos que van apareciendo en la escena. Un ejemplo muy concreto seria el de reconocimiento facial.

Existe un enorme interés en la actualidad por desarrollar descriptores audiovisuales que permitan caracterizar las el contenido de forma automatizada. El estándar MPEG-7 [4] desarrollado por MPEG (Motion Picture Expert Group) reúne una colección de descriptores visuales aplicables para su implementación en sistemas de recuperación de contenido multimedia.

2.2.1 Descriptores visuales MPEG-7

El interfaz de descripción de contenido multimedia, MPEG-7 define un conjunto de descriptores visuales empleados con frecuencia en los motores de búsqueda en bases de datos. Para realizar la descripción de un contenido visual nos podemos fijar en distintas propiedades; en la Figura 2-1 se muestran las principales características que es posible describir:

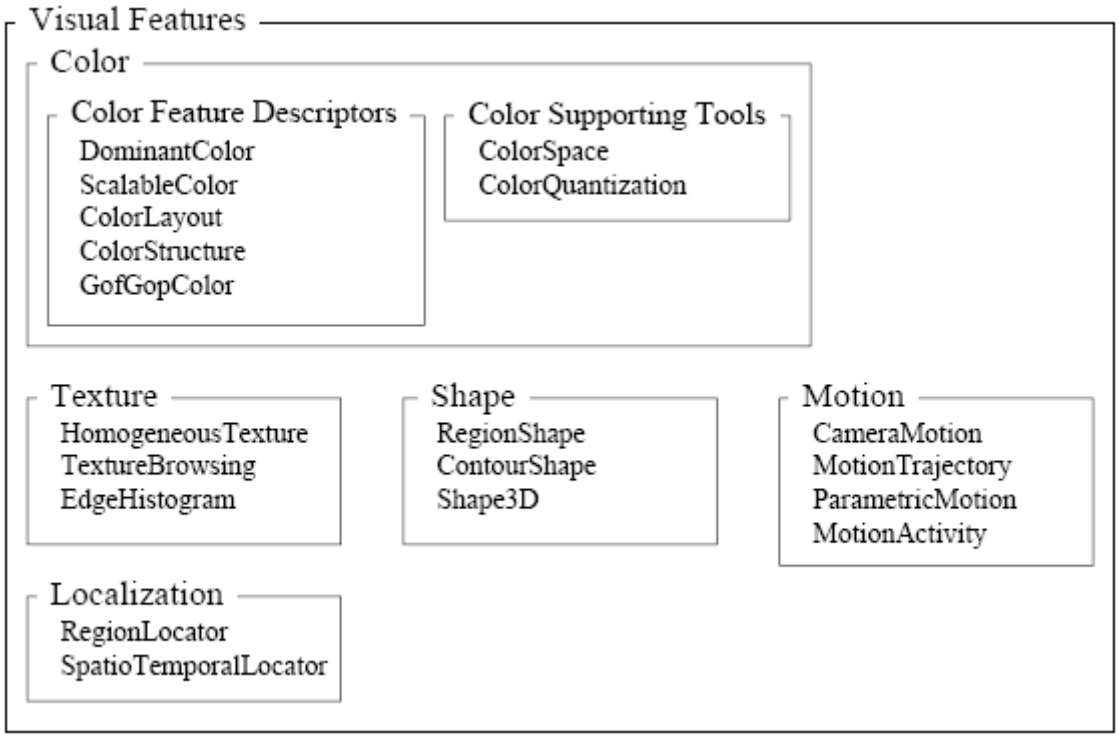

Figura 2-1: Descriptores para representar las características visuales (ISO/IEC 15938-3)

Dentro del marco del estado del arte de este proyecto centraremos nuestra atención en exponer brevemente las herramientas de descripción existentes para cada una de las cinco categorías (color, textura, forma movimiento y localización).

Herramientas para la descripción del Color (Color):

- *DominantColor DS*. Este descriptor permite la caracterización de los colores dominantes (más representativos) de una determinada región.
- *ScalableColor DS*. Este descriptor permite caracterizar la distribución de los distintos colores de una determinada región.
- *ColorLayout DS. Con* este descriptor se especifica la distribución espacial de los colores de una determinada región.
- *ColorStructure DS.* Este descriptor permite caracterizar la distribución espacial local de una determinada región.

Herramientas para la descripción de la textura (Texture):

Los descriptores de textura facilitan la navegación y la recuperación usando dicha característica en bases de datos de imagen y vídeo. Para ello disponemos de las siguientes características: *HomogeneusTexture DS, TextureBrowsing DS y EdgeHistogram DS.*

Herramientas para la descripción de la forma (Shape):

- *RegionShape DS*. Este descriptor especifica la forma de una región de un objeto.
- *ContourShape DS*. Este descriptor especifica un contorno cerrado de un objeto 2D o una región en una imagen o una secuencia de vídeo.
- *Shape3D*. Para la descripción de contornos en 3D.

Herramientas para la descripción del movimiento (Motion):

- *Camera Motion DS*. En este descriptor se especifican un conjunto de operaciones de movimiento básicas con una cámara (*pan y tilt*)
- *MotionTrayectory DS.* El movimiento de un determinado punto, de un objeto o region, puede ser caracterizado mediante este de descriptor.
- *ParametricMotion DS*. Este descriptor permite caracterizar la evolución de una región arbitraria mediante una transformación geométrica 2D.
- *MotionActivity DS*. Con este descriptor se especifica el ritmo del movimiento en una secuencia.

Herramientas para la descripción de la localización (Localization):

Se utilizan para describir elementos en el dominio espacial o temporal dentro de la secuencia de vídeo. Los dos descriptores que lo permiten son *Region Locator DS y Spatio Temporal Locator D*S.

Para obtener una información más detallada sobre los descriptores visuales MPEG7 se puede consultar la parte 3 del estándar ISO/IEC 15938-3 [5].

2.2.2 Descriptores de formas

Las personas pueden reconocer objetos solamente por la forma –está probado que la forma a menudo contiene información semántica–. Esto distingue a los descriptores de forma de otros descriptores visuales elementales como el color, el movimiento o la textura

que, aunque resultan igual de importantes, generalmente carecen de la capacidad de revelar la identidad del objeto por sí solos.

Este hecho ha propiciado un mayor auge de los sistemas de caracterización basados en descriptores de forma [6] (ver Figura 2-2). Las aplicaciones que utilizan este tipo de descriptores pueden dividirse a su vez en dos categorías: los sistemas basados en regiones (*Region-Based*) y los sistemas basados en contorno (*Boundary-Based* o *Contour-Based*). Los primeros incluyen descriptores de momentos [7][8] y los atributos geométricos. Al segundo grupo pertenecen los descriptores de Fourier [9], de tipo CSS [10] y los algoritmos basados en wavelets [11].

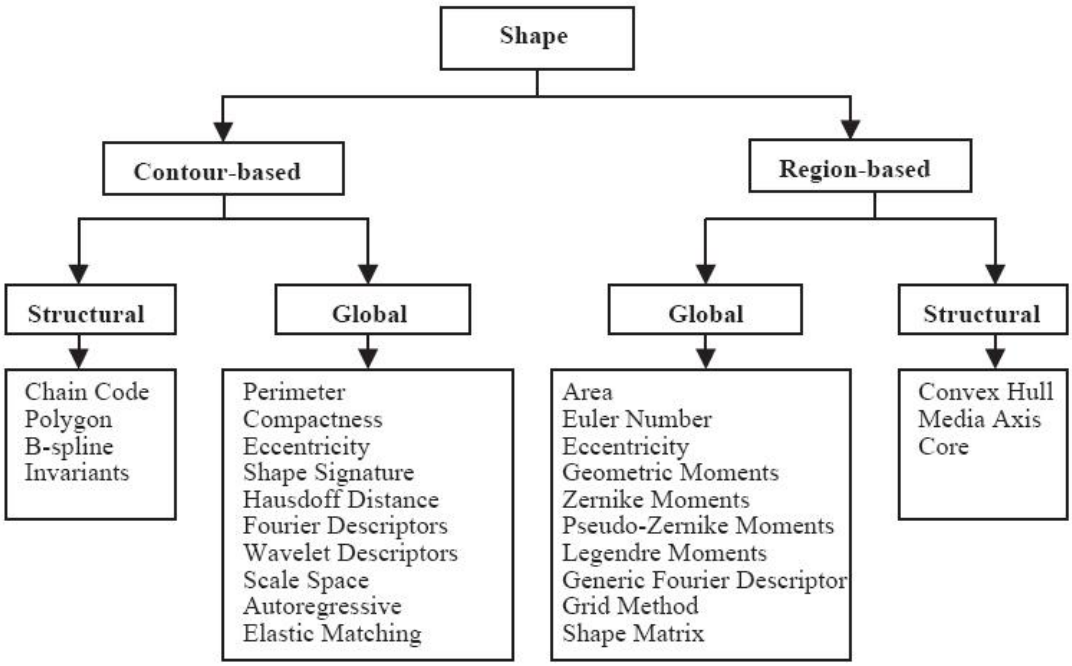

Figura 2-2: Clasificación de los descriptores de formas (Imagen tomada de [6])

Las **Descripciones basadas en Regiones** se utilizan cuando el interés principal es describir las propiedades de la región (el reconocimiento posterior se va a basar en el color, la textura, etc. de los objetos detectados)

La representación de las regiones a través de momentos [7] [8] supone la interpretación de las funciones imagen como densidades de probabilidad de variables aleatorias 2D. Las propiedades de estas variables aleatorias se pueden describir usando características estadísticas como son los momentos.

Los momentos de orden p + q de una imagen I(x, y) son:

$$M_{p,q} = \sum_{x=1}^{N} \sum_{y=1}^{M} x^p y^q I(x,y)$$

Por ejemplo, el momento de orden cero para una imagen binaria es el área del objeto.

$$M_{0,0} = \sum_{x=1}^{N} \sum_{y=1}^{M} I(x,y)$$

Un momento de orden (p+q) es dependiente de la escala, de translaciones, de rotaciones y de cualquier transformación realizada sobre los niveles de gris.

Las **Descripciones basadas en Contornos** se utilizan cuando el interés principal es describir la forma de los contornos (el reconocimiento posterior se va a basar en como es la forma de los objetos detectados). El principal conjunto de descriptores dentro de este grupo son los descriptores de Fourier. Los descriptores de Fourier (FDs) se aplican al contorno de la forma y son invariantes al escalado, la rotación y la traslación. En cambio, resultan muy sensibles al ruido por lo que son más eficientes en la descripción de formas de contornos cerrados. Su definición es la siguiente:

Dado un contorno definido por una sucesión de N puntos en un sentido determinado: $\{(x_0, y_0), (x_1, y_1),\ldots(x_{N-1}, y_{N-1})\}$, donde cada par coordenado es un complejo: $s(k) = x(k) + jy(k)$. La DFT de s(k) son los coef. complejos a(u) y se denominan los *descriptores de Fourier(FD) del contorno*:

$$a(u) = \frac{1}{N} \sum_{k=0}^{N-1} s(k) \exp(-j2\pi uk / N) \qquad s(k) = \sum_{u=0}^{N-1} a(u) \exp(j2\pi uk / N)$$

Los FD's son *características regeneradoras de formas*. El número de descriptores necesarios para la regeneración de la forma dependen de la forma en sí y la precisión deseada.

2.3 Diseño

Como hemos comentado anteriormente, el objetivo principal de este proyecto es la implementación de un sistema que describa las regiones que componen una imagen o frame de vídeo. Abordaremos el problema de la caracterización de las regiones planteando un

conjunto de descriptores relativamente sencillos. Dicha caracterización nos servirá para implementar dos ejemplos de aplicación.

La Figura 2-3 muestra un diagrama básico de las etapas que constituyen el sistema. Tras el módulo de segmentación en el que se determinan las regiones que forman la imagen y el módulo de extracción de descriptores mediante el que obtenemos las características de dichas regiones, se pueden considerar diversas aplicaciones. En nuestro caso, centraremos la atención en la recuperación de imágenes (*Image Retrieval*) y el seguimiento de objetos (*Object Tracking*). En los siguientes subapartados, analizaremos los módulos de segmentación y extracción de descriptores, dejando las aplicaciones para ser estudiadas en el tercero y cuarto capítulo del presente trabajo.

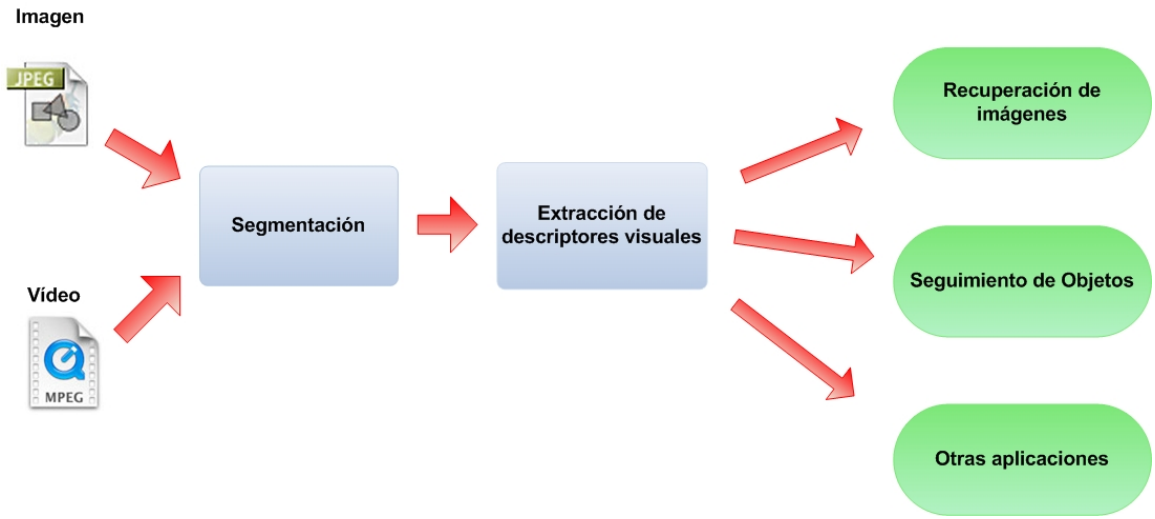

Figura 2-3: Diagrama básico de las etapas del sistema

2.3.1 Segmentación

La segmentación se define como el proceso mediante el cual se divide la imagen en masas de píxeles conectados formando regiones homogéneas que comparten alguna característica común.

Las aplicaciones de las técnicas de segmentación sobre imágenes son diversas, desde el análisis de imágenes médicas a la teledetección o la gestión del contenido multimedia. Nuestro sistema requiere un paso previo de segmentación que separe las regiones de las que se compone la imagen para posteriormente ser tratadas.

Dentro de la gran variedad de técnicas de segmentación aplicadas a imágenes en color, podemos encontrar distintas aproximaciones, como las técnicas basadas en píxeles, en regiones, en bordes o en modelos de color. Un resumen de estas técnicas se puede encontrar en [12].

En nuestro caso, hemos utilizado un segmentador desarrollado por la DCU (Dublín City University) [13]. Este segmentador se basa en el criterio de homogeneidad de color para dividir la imagen de entrada en regiones conexas. Hace uso del algoritmo *Recursive Shortest Spanning Tree (RSST)* [14]. Consiste en un método rápido de crecimiento de regiones que, comenzando a nivel de píxel une las regiones iterativamente de acuerdo con la distancia calculada a partir del color y el tamaño de la región. El segmentador empleado (ver Figura 2-4) permite además definir el número de regiones de la segmentación, no superando un máximo de 255 regiones.

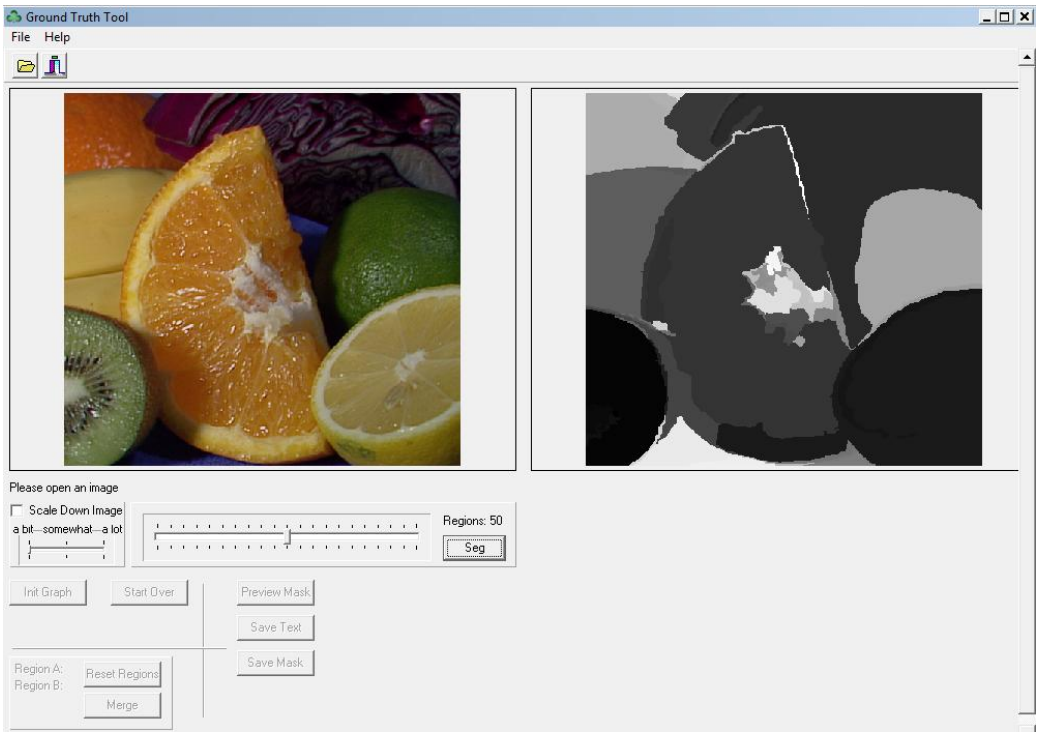

Figura 2-4: Captura de la herramienta de segmentación.

Sin embargo, a pesar de fijar un número determinado de regiones, el comportamiento del segmentador es impredecible en la mayoría de casos. La Figura 2-5 ilustra dos ejemplos de una segmentación no adecuada. En el primer ejemplo (Figura 2-5 (a)) se ha especificado un número demasiado bajo de regiones; el resultado es que regiones como las caras se funden con otras. En el segundo ejemplo (Figura 2-5 (b)) ocurre lo contrario; la cara, que debería constituir una única región, se encuentra sobre-segmentada.

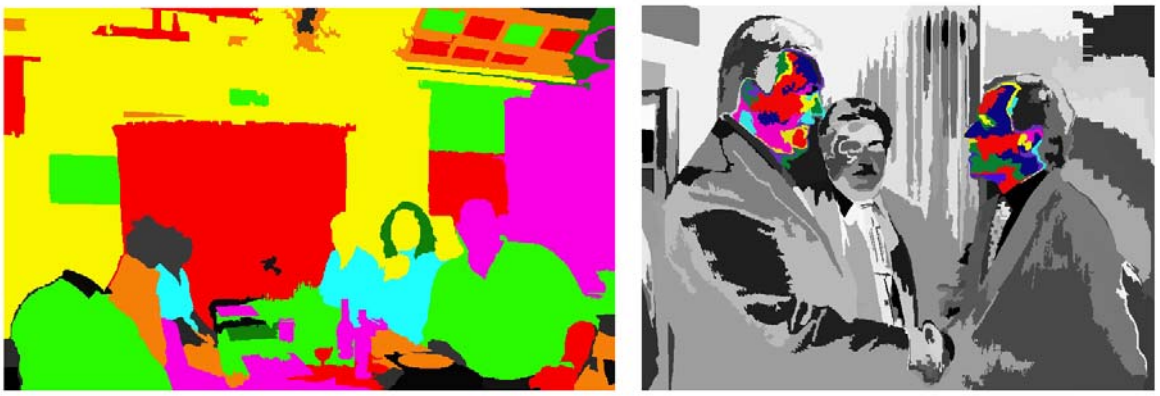

Figura 2-5: Ejemplos de una segmentación inadecuada. (a) Baja segmentación, (b) Sobre-segmentación.

2.3.2 Extracción de los descriptores visuales

2.3.2.1 Gestión de las regiones

Tras la segmentación, se obtiene una imagen fragmentada en regiones según un criterio de semejanza. En la práctica, consideraremos cada región como un conjunto de píxeles *P*, cada uno con coordenadas $p_i = (x_i, y_i)$ dentro de la imagen (con origen $O = (1,1)$ en la esquina superior izquierda) y su identificador de región *id* (valor entre 0 y 255 asociado tras la segmentación).

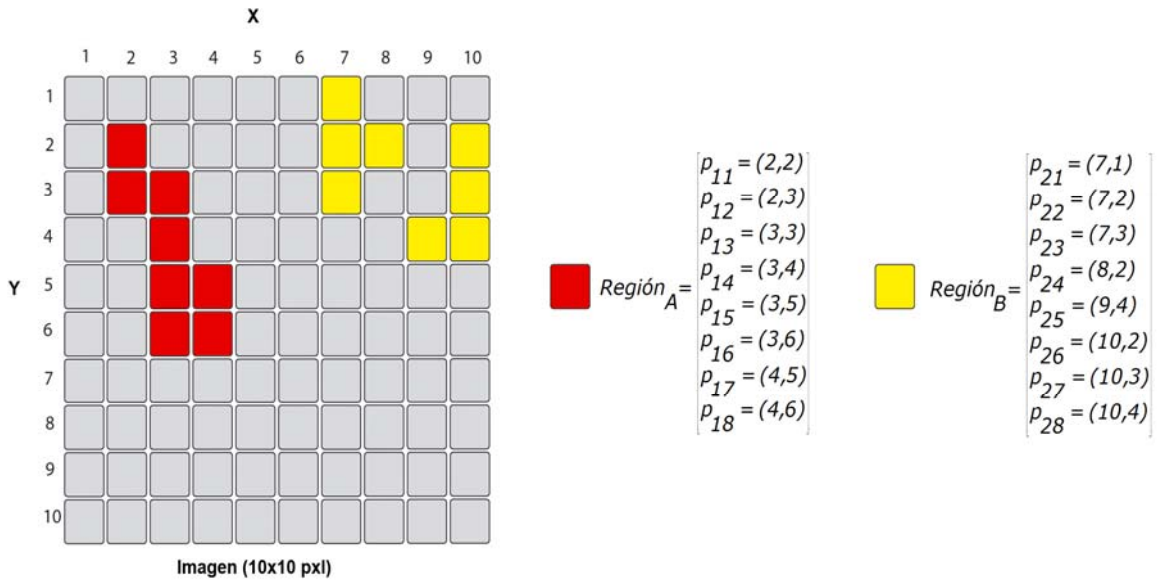

Figura 2-6: Definición de una región como conjunto de coordenadas de píxeles.

2.3.2.2 Definición de los descriptores implementados

2.3.2.2.1 Descriptores Intra-región

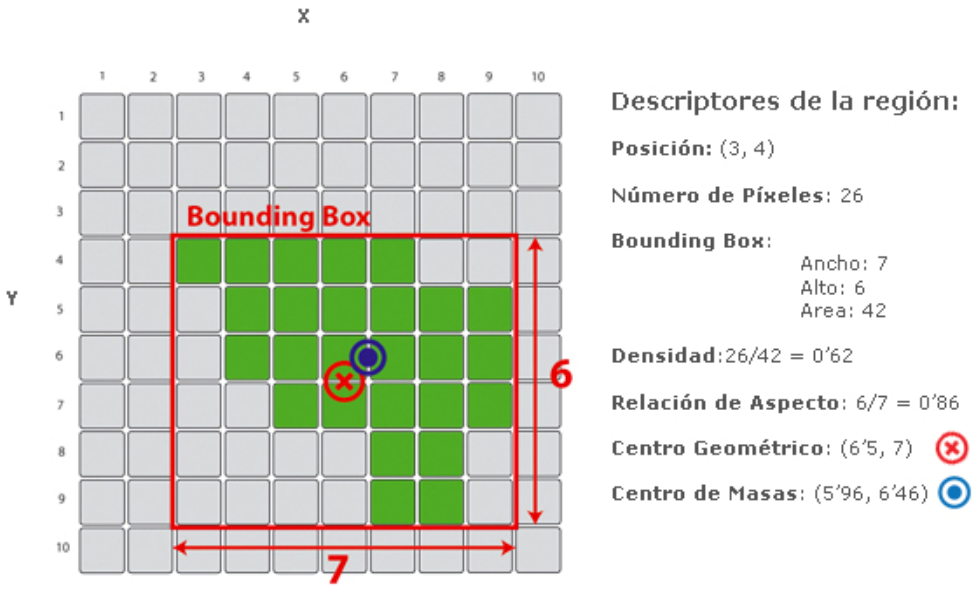

Figura 2-7: Descriptores intra-región

NumeroDePixeles (N). Es el número de puntos que forman la región. Se calcula como el tamaño del vector de píxeles que compone la región.

$$N = cardinal\{P\}$$

BoundingBox (BBox). Es el rectángulo que contiene por completo a la región (ver Figura 2-7).

Ancho: Es el ancho de la BBox. Se calcula hallando previamente la posición máxima y mínima en el eje x de los píxeles que forman la región, $p_i = (x_i, y_i)$.

$$Ancho = x_{max} - x_{min} + 1$$

Alto: Es el alto de la BBox. Se calcula hallando previamente la posición máxima y mínima en el eje y de los píxeles que forman la región, $p_i = (x_i, y_i)$.

$$Alto = y_{max} - y_{min} + 1$$

Area: Es el área de la BBox. Se halla mediante el alto y el ancho.

$$Área = Ancho \times Alto$$

RelacionAspecto: Relación entre el ancho y el alto de la Bounding Box.

$$RelacionAspecto = \frac{Ancho}{Alto}$$

Este descriptor nos permite diferenciar las regiones extendidas a lo ancho de las extendidas más a lo largo dentro de una imagen, como muestran las regiones (a) y (b) de la Figura 2-8. En cambio, no es un buen descriptor para la caracterización de la forma de la región ya que como se puede ver en las regiones (c) y (d) existe la posibilidad de que dos regiones con el mismo tamaño de Bounding Box, y, por consecuencia, la misma relación de aspecto, tengan formas muy diferentes.

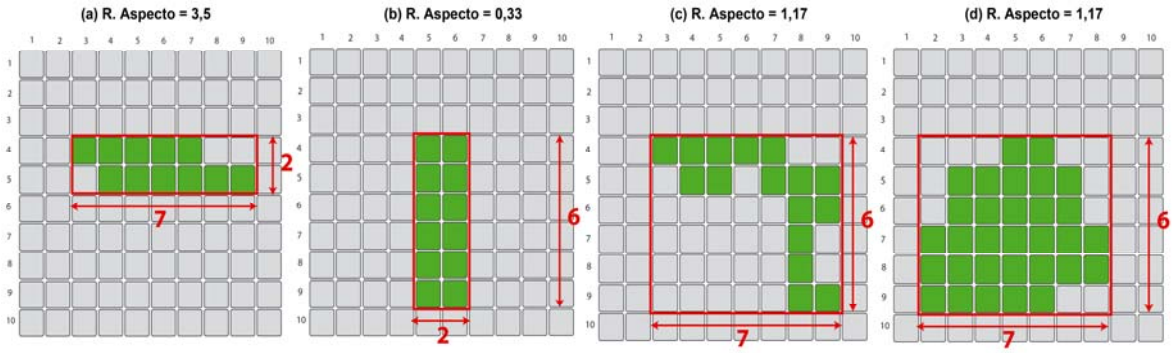

Figura 2-8: Relación de Aspecto para distintas regiones

Densidad: Relación entre el número de puntos de la region, N, y el área de la BBox. Este descriptor nos permite discriminar entre regiones más o menos compactas. Las regiones (a) y (b) de la Figura 2-9 son una buen ejemplo del tipo de regiones caracterizables con este descriptor. No obstante, existen casos como muestran las regiones (c) y (d) de la Figura 2-9 donde el descriptor de densidad no es significativo a la hora de caracterizar la forma de las regiones. Además ambas regiones se encuentran contenidas en dos BBox de la misma dimensión, y por tanto, el descriptor RelacionAspecto combinado con el descriptor de densidad tampoco proporcionaría la información necesaria para discriminar entre estas formas.

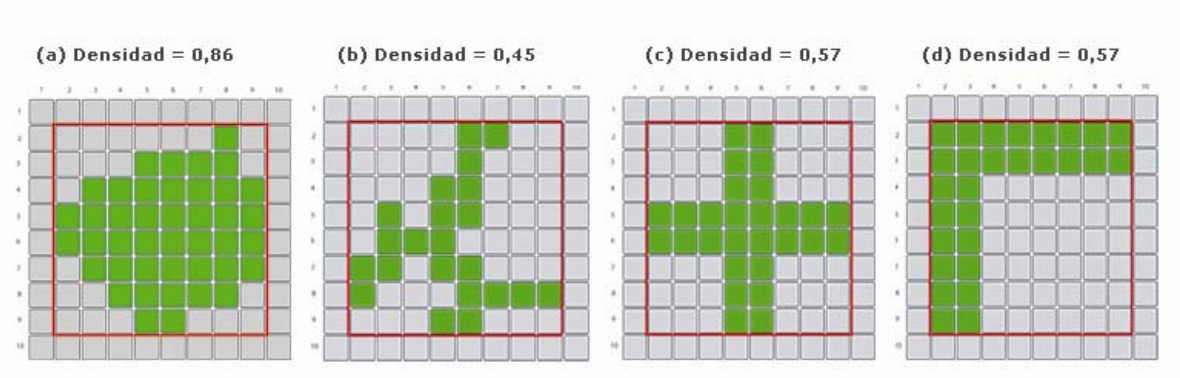

Figura 2-9: Valores de densidad para diversas formas de regiones.

Perímetro: Obtiene una aproximación medida de la longitud del perímetro de la región. Dado que la sencillez es uno de los objetivos de este trabajo, hemos aplicado una aproximación con el fin de evitar técnicas de detección de bordes más complejas y ganar así en velocidad. El primer paso consiste en calcular el número de vecinos *V* de cada píxel, $p_i = (x_i, y_i)$, que compone la región. El número de vecinos *V* de un píxel p_i se calcula como el número de píxeles que rodean a p_i y que forman parte de la misma región. La Figura 2-10(a) muestra un ejemplo del cálculo de *V* para una determinada región. Consideraremos el conjunto de píxeles de borde de una determinada región *PB* como el subconjunto de píxeles con $V(p_i) \leq 6$.

$$PB = \{\, p_i \mid V(p_i) \leq 6 \,\}$$

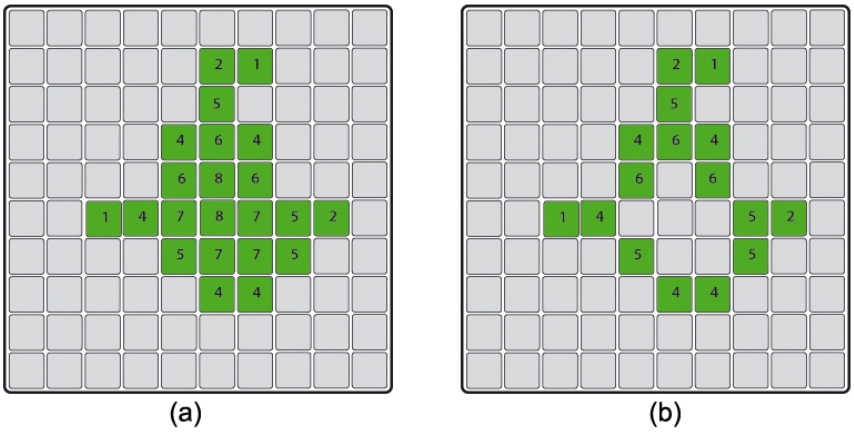

Figura 2-10: (a) número de vecinos de cada píxel de la región,
(b) Píxeles que pertenecen al perímetro de la región.

Para calcular la longitud de del perímetro de la región se consideran los píxeles incluidos en el subconjunto *PB*. Para cada píxel y dependiendo de la distribución de los vecinos se considera un valor aproximado del perímetro. La Figura 2-11 muestra las distribuciones más comunes y el valor del perímetro asignado en cada caso según la orientación del borde

de la forma. Hemos definido la función *CP* (Contribución al perímetro), que devuelve los valores que muestra la Figura 2-11 para las distribuciones típicas o 1 en el resto de casos con una menor incidencia.

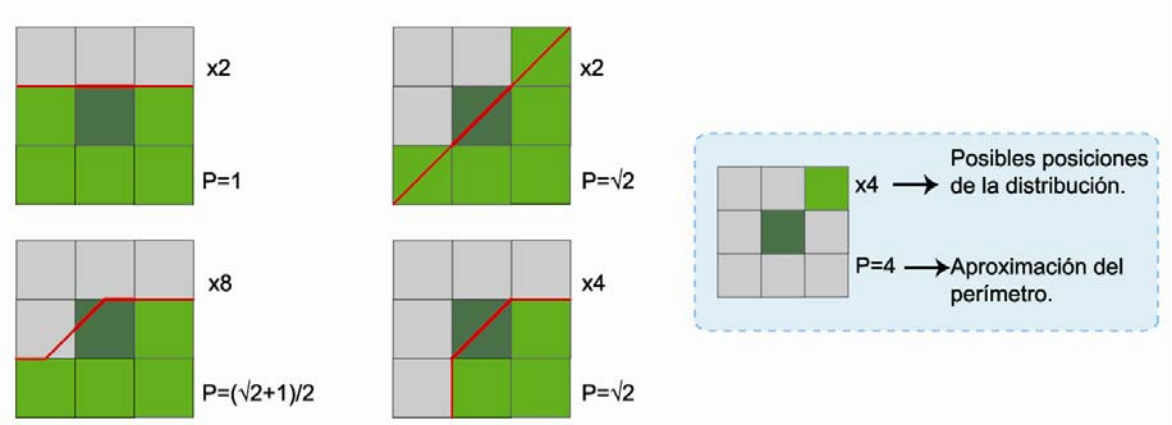

Figura 2-11: Distribuciones de los vecinos típicas para los píxeles de borde y su contribución al perímetro.

El valor aproximado de la longitud del perímetro se calcula aplicando la función de contribución al perímetro *CP* a cada uno de los píxeles de *PB* (conjunto de píxeles del borde).

$$Perímetro = \sum CP(p_i), \quad p_i \in PB$$

Compacidad: El valor intra-región de la compacidad proporciona una medida sobre cómo es de compacta una determinada región, o dicho de otro modo, cuánto se parece esta región al círculo, cuya compacidad ideal es 1. El cuadrado tiene una compacidad ideal de 0.87.

$$Compacidad = \frac{PerimetroIdeal}{Perímetro} \qquad PerimetroIdeal = 2 \times \pi \sqrt{\frac{N}{\pi}}$$

Se trata de un descriptor muy estable, invariante a las variaciones de tamaño de la región y a la rotación. La Figura 2-12 muestra algunos ejemplos de formas, la extracción de los píxeles que forman el perímetro según nuestra aproximación y los valores de compacidad obtenidos. Se observa que, para los ejemplos del cuadrado y del círculo, los valores obtenidos con el descriptor se aproximan a los valores teóricos.

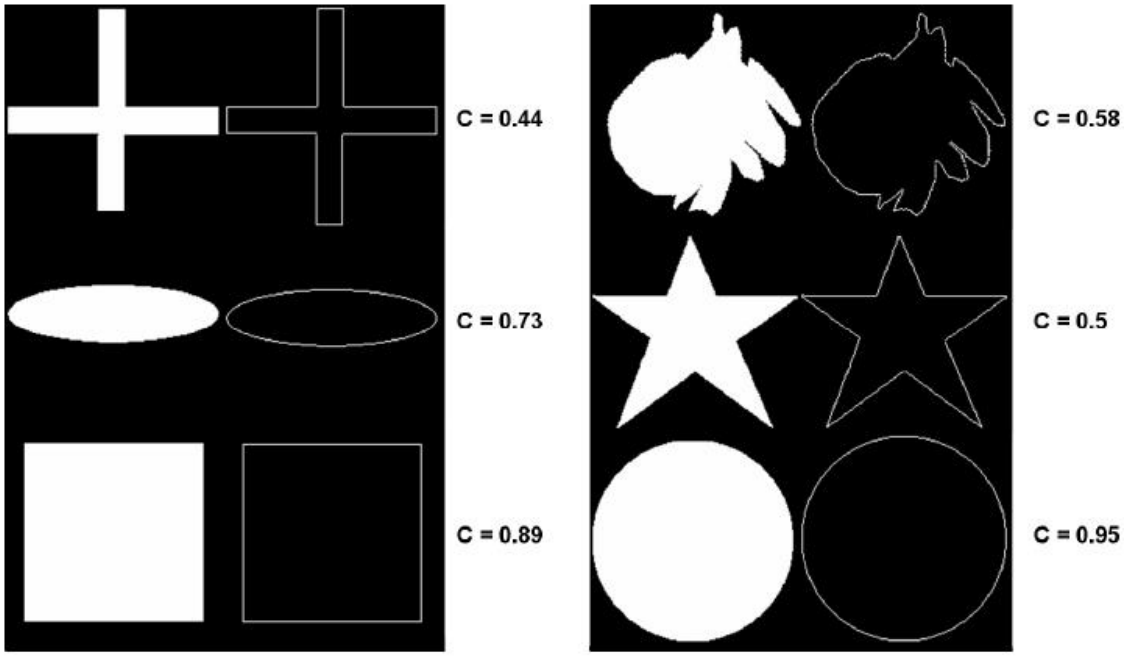

Figura 2-12: Extracción del perímetro de algunas formas y sus valores de compacidad

Centro Geométrico (CG): Posición del centro geométrico de la BBox de la región.

$$CG = (x_g, y_g) = \left(\frac{x_{min} + x_{max}}{2}, \frac{y_{min} + y_{max}}{2} \right)$$

Centro de Masas (CM): Posición del centro de masas de los píxeles que componen la región, considerando el peso de cada píxel como 1.

$$CM = (x_m, y_m) = \frac{\sum_{1}^{N} p_i}{N}$$

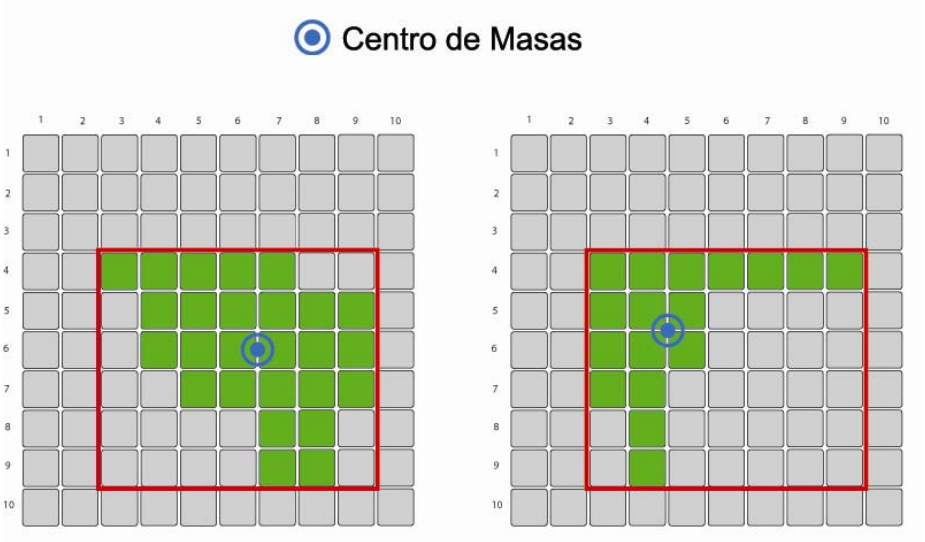

Figura 2-13: Ejemplos de posición de centros de masas.

Este descriptor es adecuado para diferenciar entre las regiones que están distribuidas de forma homogénea dentro de las BBox de las que no lo están. No obstante, su utilización de forma aislada no aporta la suficiente información para discriminar entre diferentes formas de región por lo que suele aplicarse combinado con otros descriptores (por ejemplo, el Centro Geométrico, ver el descriptor Relación de Centros).

Relación de Centros (RC): Con este descriptor obtenemos una medida de la distancia entre el centro geométrico y el centro de masa en relación al tamaño de la BBox. Nos proporciona un vector que apunta hacia la zona de la región donde se encuentre la mayor concentración de píxeles.

$$RC = (x_r, y_r) = \left(\frac{x_m - x_g}{Ancho}, \frac{y_m - y_g}{Alto} \right)$$

La Figura 2-14 ilustra un ejemplo de cómo este descriptor ofrece un indicador de la dirección en la que se encuentra el mayor número de píxeles de la región.

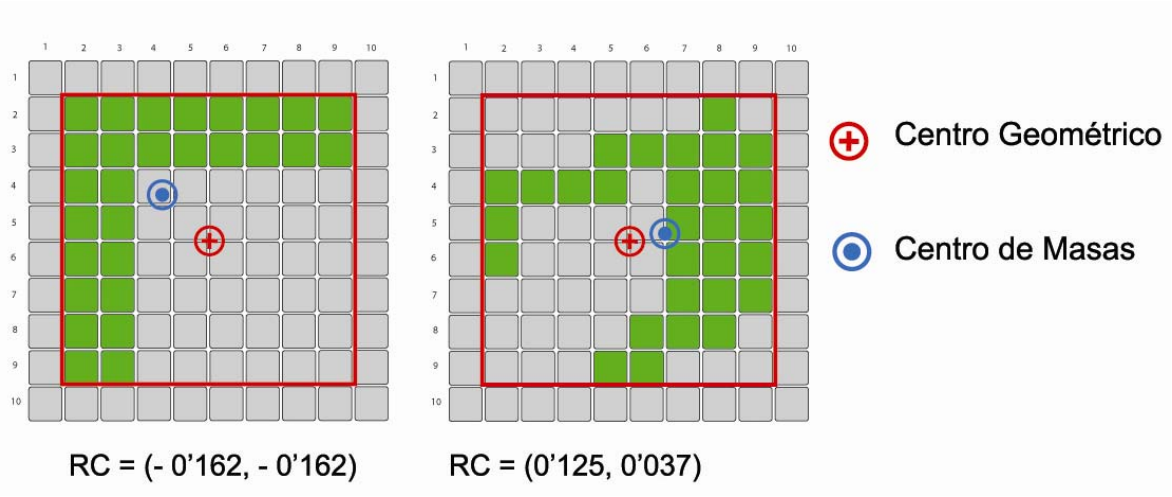

Figura 2-14: Ejemplos de valores de RC

Media RGB: Calcula una media del color de todos los píxeles de la región para cada plano de color RGB.

$$\overline{R} = \frac{\sum_{i}^{N} R(p_i)}{N} \qquad \overline{G} = \frac{\sum_{i}^{N} G(p_i)}{N} \qquad \overline{B} = \frac{\sum_{i}^{N} B(p_i)}{N}$$

Varianza RGB: Calcula la varianza de color de todos los píxeles de la región para cada plano de color RGB.

$$\sigma_R^2 = \frac{\sum_{i}^{N} \left(R(p_i) - \overline{R}\right)^2}{N} \qquad \sigma_G^2 = \frac{\sum_{i}^{N} \left(G(p_i) - \overline{G}\right)^2}{N} \qquad \sigma_B^2 = \frac{\sum_{i}^{N} \left(B(p_i) - \overline{B}\right)^2}{N}$$

2.3.2.2.2 Descriptores Inter-región

Los descriptores hasta ahora vistos, realizan la caracterización de las regiones de forma aislada. Para poder caracterizar las diversas relaciones que se establecen entre las regiones de la imagen, hemos implementado el conjunto de descriptores inter-región.

DistanciaGeométrica: La distancia geométrica se define como la distancia euclídea entre los centros geométricos (x_{gi}, y_{gi}), (x_{gj}, y_{gj}) de cada región i, j.

$$DistGeom(i, j) = \sqrt{(x_{gi} - x_{mj})^2 + (y_{gi} - y_{gj})^2}$$

Distancia de masas: Definición análoga a la anterior utilizando los centros de masa (x_{mi}, y_{mi}), (x_{mj}, y_{mj}) de las dos regiones i, j.

$$DistMasas(i, j) = \sqrt{(x_{mi} - x_{mj})^2 + (y_{mi} - y_{mj})^2}$$

Ambos descriptores nos aportan información sobre la distancia entre las regiones de las imágenes.

Relación de Tamaño: Cociente del número de píxeles de una región entre los de otra. Este descriptor resulta muy útil para comparar regiones por su tamaño. Una relación de tamaño próxima a la unidad indicaría que se trata de regiones de tamaño similar.

$$RelacionTam(i, j) = \frac{N_i}{N_j}$$

Relación de compacidad: Indica si existe similitud de compacidad entre dos regiones. Una relación de compacidad próxima a la unidad indicaría que se trata de regiones de compacidad similar.

$$RelacionCompac\ (i, j) = \frac{Compacidad_i}{Compacidad_j}$$

Se trata de un descriptor muy potente para medir similitud de forma entre dos o más regiones.

Concentración: Ofrece una medida de el número de regiones que rodean a una dada. Utiliza cómo parámetro la distancia de masas y nos permite distinguir entre zonas con un elevado número de regiones de otras con regiones aisladas (ver Figura 2-15).

$$C(i, j) = \begin{cases} \dfrac{1}{DistMasas(i,j)}, & DistMasas \geq 1 \\ 1, & DistMasas < 1 \end{cases}$$

$$Concentracion_i = \sum_{j=1}^{N} C(i,j)$$

Este descriptor funciona muy bien para detectar una alta concentración de pequeñas regiones debido a que su menor tamaño da lugar a bajos valores de la distancias de masas, y como consecuencia, un alto valor de concentración acumulado.

Gracias a este descriptor se podría considerar dar una mayor relevancia a grupos de regiones pequeñas en lugar de a las regiones aisladas.

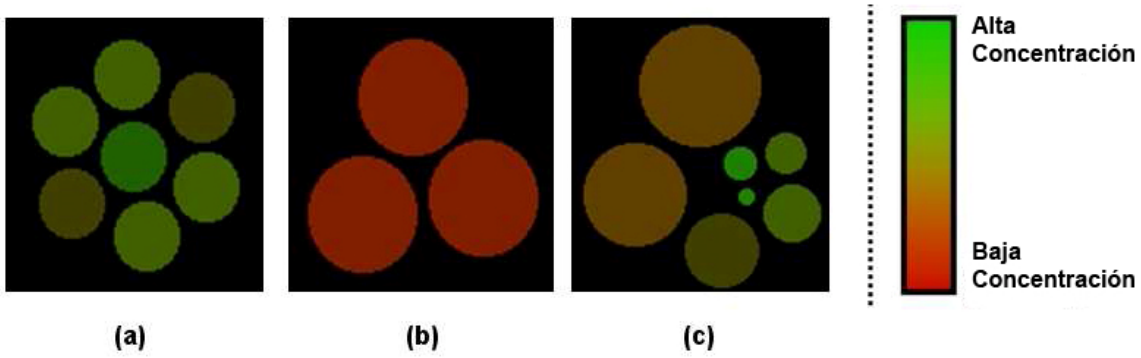

Figura 2-15: Ejemplos de concentración de regiones

ConcentraciónSimilar: Al igual que el descriptor anterior, proporciona una medida de las regiones que rodean a una dada, pero en este caso, se da más peso a regiones de tamaño similar.

$$C(i,j) = \begin{cases} \dfrac{1}{DistMasas(i,j)}, & DistMasas \geq 1 \\ 1, & DistMasas < 1 \end{cases}$$

$$S(i,j) = \sqrt{min(RelacionTam(i,j), RelacionTam(j,i))}$$

$$ConcentracionSimilar_i = \sum_{j=1}^{N} S(i,j) \cdot C(i,j)$$

La Figura 2-16 muestra la comparación entre los descriptores de *Concentración* y *Concentración Similar*. En el ejemplo (a) se obtienen los mismos valores tras aplicar ambos descriptores porque las regiones son todas del mismo tamaño ($S(i, j)= S(j, i)=1$). En el ejemplo (b) la concentración es la misma, en cambio la concentración de regiones similares es menor debido a las diferencias de tamaño apreciables entre las regiones.

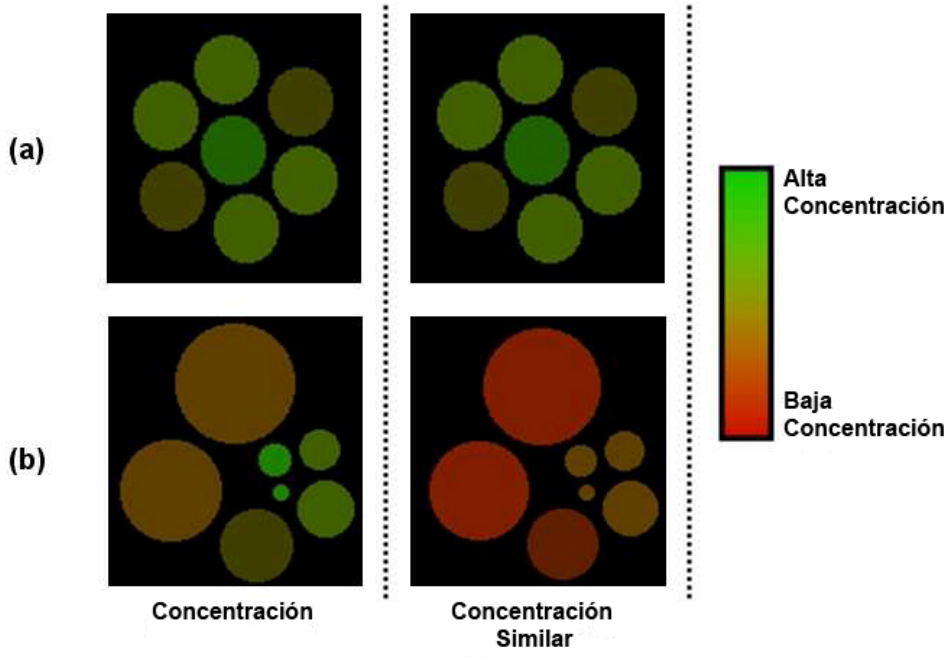

Figura 2-16: Concentración y Concentración Similar.

De la misma forma que el anterior, este descriptor es útil para diferenciar las regiones que se encuentran agrupadas de las que están aisladas dentro de la imagen, pero en este caso se otorga un valor más elevado a las regiones rodeadas de otras similares en tamaño.

Atracción: La fuerza de atracción de una región j sobre otra i es inversamente proporcional a la distancia entre sus centros de masa y directamente proporcional al tamaño de j. La fuerza de atracción entre dos regiones i, j se define como la atracción que ejerce la región j sobre la región i.

$$Atracción(i,\ j)\ =\ \frac{N_j}{DistMasas(i,j)^2}$$

Relación de Contacto: Este descriptor se define como el conjunto de píxeles del perímetro de la región i que están en contacto con el perímetro de la región j entre el número de píxeles del perímetro de la región j. Denominaremos Contacto(i, j) al conjunto de píxeles de la región i que tienen un píxel de la región j en su vecindad (considerando vecinos a los 8 píxeles que rodean a uno dado).

$$RelacionContacto = \frac{Contacto(i,j)}{Perimetro(i)}$$

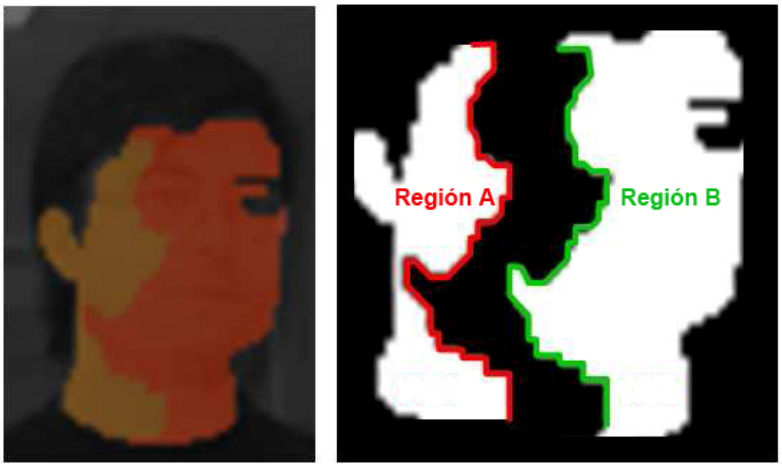

Figura 2-17: Relación de contacto entre dos regiones

La Figura 2-17 ilustra un ejemplo de dos regiones segmentadas que forman parte de la misma cara. Los contornos coloreados en verde y rojo son los puntos en contacto entre ambas regiones. La relación de contacto se obtiene en proporción al número de píxeles que forman el perímetro de cada región.

2.3.2.3 Usos de los descriptores definidos

En este apartado, analizaremos las posibles aplicaciones de los descriptores definidos. La elección de los descriptores que caractericen las regiones dependerá exclusivamente de la aplicación para la que son requeridos. Por ejemplo, si lo que buscamos son imágenes cuyas regiones se encuentran distribuidas espacialmente de una determinada manera elegiremos como descriptor la distancia entre regiones (entre sus centros geométricos o de masa). En otros casos es posible por ejemplo que la posición de una región en la imagen no sea relevante si no simplemente la aparición de una determinada forma y color en una imagen (como es el caso del *tracking* o seguimiento de objetos). A continuación, mostraremos algunos ejemplos de diversas combinaciones de descriptores atendiendo a sus posibilidades de aplicación:

- **Color (Color medio)**

  Existen sistemas de recuperación de imágenes que se basan exclusivamente en este descriptor. Es una opción en los casos donde nos interese clasificar las imágenes

atendiendo únicamente a los colores que predominan en ella. Sin embargo no proporciona información sobre la distribución espacial de los colores en la imagen por lo que existirán situaciones en los que no reflejará el parecido entre imágenes adecuadamente.

- **Forma**

La forma de una región se determina mediante la combinación de dos descriptores: la compacidad y la relación de aspecto. La aplicación únicamente de la forma a la hora de caracterizar una región viene justificada en aplicaciones de clasificación y/o retrieval con imágenes como las que muestra la Figura 2-18, donde el color, la posición o el tamaño no aportan apenas información para diferenciar unos objetos de otros.

Figura 2-18: Ejemplos de imágenes tomadas de *Columbia Object Image Library*[2]

- **Combinación de Color, Posición y Tamaño**

Esta combinación permite comparar unas regiones con otras, teniendo no sólo en cuenta el color de la región, sino también su posición en la imagen y su tamaño. Esta es la combinación que hemos considerado óptima en la posterior aplicación del sistema de recuperación de imágenes similares a una dada.

[2] http://www1.cs.columbia.edu/CAVE/

- **Combinación con descriptores inter-región**

  En usos concretos, puede ser necesario aplicar descriptores menos comunes. Por ejemplo, si queremos detectar un conjunto de regiones parecidas en un área de la imagen emplearemos el descriptor *ConcentraciónSimilar*.

En conclusión, se proporciona un conjunto amplio de descriptores que pueden ser aplicados en diversas situaciones.

Por otro lado, la medida de similitud entre las regiones que hemos utilizado es la distancia. A menor distancia entre regiones, mayor es el parecido entre ellas. Esta distancia será una función que tendrá como argumentos los descriptores que se hayan seleccionado para una aplicación concreta. Por lo tanto, abordaremos esta cuestión en profundidad en los capítulos 3 y 4.

2.4 Desarrollo

Para programar las dos aplicaciones, hemos utilizado el lenguaje C++. Con este lenguaje de programación podremos realizar rutinas que tengan una alta carga computacional de una manera más eficiente que con otros lenguajes de programación (como por ejemplo Java). Además este lenguaje nos permite la fácil integración con las librerías OpenCV.

2.4.1 Tratamiento de la imagen digital con OpenCV

OpenCV es una librería de visión artificial de código abierto desarrollada originalmente por Intel. Se trata de una librería multiplataforma que se puede ejecutar en Mac OS X, Windows y Linux. Está enfocada principalmente al procesado de imágenes en tiempo real.

OpenCV incorpora tipos de datos y funciones que permiten manejar imágenes y vídeo de forma sencilla. En este apartado comentaré los tipos de datos y funciones de esta librería que se han utilizado en el presente trabajo.

CvPoint: Establece las coordenadas de un píxel (punto) de una imagen. El origen es (0,0). Las componentes x e y utilizan enteros y por tanto no pueden expresar valores en punto flotante.

```
typedef struct CvPoint
{
int x;
int y;
}
CvPoint;
```

CvRect: Establece un rectángulo en una imagen. Se determina mediante las coordenadas del píxel situado en la esquina superior izquierda, la anchura y la altura del rectángulo. Se expresa en enteros.

```
typedef struct CvRect
{
int x;
int y;
int width;
int height;
}CvRect;
```

IplImage: Tipo de dato que permite representar imágenes. Este tipo de dato carga las imágenes en la memoria sin comprimirlas, por tanto es fácil calcular el espacio necesario conociendo las propiedades de la imagen (dimensiones y bits por píxel).

```
typedef struct _IplImage
{
int nSize;
int ID;
int nChannels;
int alphaChannel;
int depth;
char colorModel[4];
char channelSeq[4];
int dataOrder;
int origin;
int align;
int width;
int height;
struct _IplROI *roi;
struct _IplImage *maskROI;
void *imageId;
struct _IplTileInfo tileInfo;
int imageSize;
char *imageData;
int widthStep;
```

```
int BorderMode[4];
int BorderConst[4];
char *imageDataOrigin;
}
IplImage;
```

Entre todas las propiedades de este tipo de datos las más importantes son: el tamaño (*nSize*), el ancho (*widht*) y el alto (*height*).

Profundidad de píxel (*depth*): Hace referencia al número de niveles de intensidad que podemos tener por píxel: 8, 16 ó 32 bits. Lo habitual son 8 bits que permiten 256 niveles.

Número de canales (*nChannels*): Número de canales de la imagen. En el caso de imágenes RGB este parámetro vale 3 (canales rojo, verde y azul).

Origen de coordenadas (*origin*): superior-izquierda o inferior-derecha. En el proyecto se utiliza el origen superior-izquierda.

Orden de los canales (*dataOrder*): Determina si los datos de la imagen se cargan entremezclando los píxeles (uno de cada canal, alternativamente) o por el contrario se cargan entremezclando canales (todos los píxeles de cada canal en bloques).

Memoria de datos (**imageData*): Los píxeles se cargan por filas de izquierda a derecha y comenzando por la superior.

Para crear una imagen tan sólo es necesario saber la dimensión, la profundidad de píxel y el número de canales:

```
IplImage* cvCreateImage(CvSize size, int depth, int channels);
```

Cuando la imagen ya no es necesaria la eliminamos liberando la memoria que utiliza mediante:

```
void cvReleaseImage( IplImage** image );
```

2.4.2 Estructuras de datos creadas

2.4.2.1 Clase Imagen

Aunque como acabamos de ver, la librería OpenCV proporciona el tipo de dato IplImage para manejar imágenes, debido a las características de nuestras aplicaciones ha sido necesaria la creación de una clase Imagen que nos permita tratar con la imagen original y su segmentación. Asimismo esta clase almacena otra información necesaria, como por ejemplo el conjunto de regiones que la componen y sus descriptores asociados.

*void CargarImagen (char *original, char *segm);*

Esta función toma como argumentos los nombres de archivo de la imagen original y la imagen segmentada y almacena ambas imágenes en la clase *Imagen*.

void CargaRegiones ();

Mediante la llamada a este método se recorren los píxeles de la imagen segmentada, identificando y almacenando las distintas regiones dentro de *Imagen*.

void ExtraerDescriptores ();

Extrae los descriptores correspondientes de cada región de la imagen, y almacena los valores obtenidos dentro de cada clase *Región*.

void FiltroTamagno (int minimo);

Se trata de un filtro que elimina las regiones más pequeñas, consideradas poco relevantes en procesos posteriores. Es un método parametrizable mediante el valor del mínimo número de píxeles.

void LiberarImagen ();

Este método llama a su vez a la función *cvReleaseImage* de OpenCV para liberar el espacio de memoria ocupado por la clase *Imagen*.

2.4.2.2 Clase Región

Esta clase almacena la información relativa a cada región de la imagen.

Dentro del grupo de métodos de la clase Región, distinguimos los métodos que nos permiten gestionar las regiones (añadir un píxel, ver si un píxel pertenece a una determinada región, obtener el id de una región,…). Estos métodos son: *Region, inicializarVec, RegionExiste, GetIdNumber, SetIdNumber, anyadirPixel, getPixel, estaEnROI.*

Por otro lado, existe otro conjunto de métodos con los que se calculan los descriptores intra-región. Pertenecen a dicho conjunto: *NumberOfPixels, WidthOfBoundingBox, HeightOfBoundingBox, AreaOfBoundingBox, AspectRatio, Density, Perimeter, Compactness, GeometricCentre, MassCentre, CentresRatio, calculaBorde, MeanRGB, RGBVariance*.

En la mayoría de los casos, no vamos a necesitar calcular todos los descriptores que caracterizan la región, sino que sólo requeriremos de un pequeño grupo de ellos. Esta es la razón por la que implementamos el método *calculateRetrievalDescriptors* y *calculateTrackingDescriptors* para calcular sólo los descriptores necesarios de cada una de las aplicaciones.
Por último existen algunos métodos "auxiliares" para el cálculo del perímetro de la región. A este grupo pertenecen: *calculaBorde,calculaVecinos, crearMascaraRegion e imprimirMascaraRegion.*

Para una explicación más detallada de las clases implementadas consultar el Anexo A.

3 Aplicación a la Recuperación de Imágenes

3.1 Introducción

En los últimos años el volumen de imágenes y vídeo digital ha experimentado un notable crecimiento. En este sentido, Internet ha contribuido a este creciente interés por capturar, guardar y transmitir todo tipo de contenido multimedia almacenado en grandes bases de datos tanto públicas como privadas. Como consecuencia, surge la necesidad de extraer o recuperar parte de esta información de forma efectiva.

Esta necesidad da lugar a la idea de los Sistemas de Recuperación de Información Visual también denominados sistemas CBIR (*Content Based Image Retrieval*). Se trata de un área de investigación que ha generado un gran interés en las últimas dos décadas aunque tiene su origen en las técnicas de recuperación de información tradicional. Su objetivo es recuperar imágenes que están relacionadas con una consulta, basándose exclusivamente en el contenido visual de la imagen.

La clave en la recuperación de la información visual se fundamenta en una representación adecuada del contenido visual de las imágenes. Aspectos visuales de bajo nivel como el color, la textura, la forma, las relaciones espaciales o el movimiento junto con otros aspectos de alto nivel como el significado de los objetos y de las escenas se usan como claves para la recuperación de imágenes en la base de datos.

El uso de la información visual en la realización de una consulta surge de las propias limitaciones de los sistemas clásicos de recuperación basados en texto que no son adecuados para modelar la proximidad perceptual entre imágenes, dado que la percepción de las imágenes es completamente subjetiva y depende de factores como el nivel de desarrollo del observador, su edad y experiencias previas, sus intereses y condiciones psicológicas. Por lo tanto, una misma imagen puede ser percibida (y descrita) de forma distinta por dos humanos diferentes. La Figura 3-1 muestra el clásico ejemplo utilizado para describir el fenómeno de la percepción subjetiva. En el dibujo, realizado por W. E. Hill y publicado en 1915, podemos ver según nuestra percepción de la realidad a una joven o una anciana.

Figura 3-1: Ejemplo de doble percepción de la imagen

El uso de los descriptores visuales pretende salvar las limitaciones de los sistemas clásicos basados en texto. En la actualidad es frecuente el uso en los sistemas recuperación de imágenes de información textual (etiquetas que describen la imagen) combinada con la proporcionada por los descriptores visuales. La Figura 3-2 presenta la arquitectura clásica de un sistema de recuperación de imágenes.

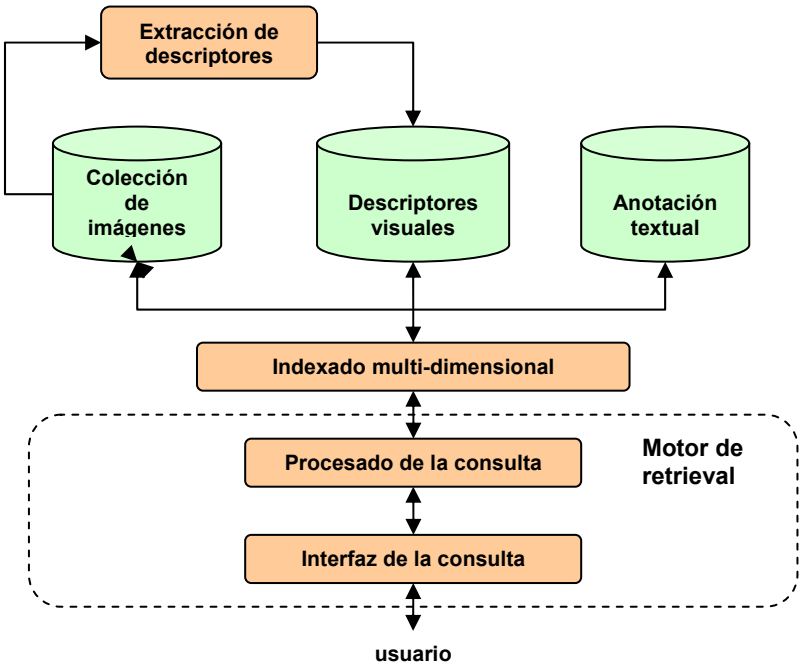

Figura 3-2: Arquitectura de un sistema de recuperación de imágenes

Se ha identificado un amplio rango de posibles usos de la tecnología CBIR. Las áreas potenciales de aplicación incluyen, entre otras: prevención de delitos, uso militar, propiedad intelectual, arquitectura e ingeniería, periodismo y publicidad, diagnóstico médico, información geográfica y sistemas de navegación, archivos históricos y museos, educación, entretenimiento y búsquedas en Internet.

El objetivo de este capítulo será realizar una revisión de los métodos de recuperación de imágenes así como demostrar la aplicabilidad del sistema de descripción de imágenes basado en regiones implementado en el capítulo anterior. Para ello partiremos de los descriptores propuestos, e implementaremos dos algoritmos de comparación entre regiones de la imagen. Además presentaremos los resultados de las pruebas realizadas sobre nuestro sistema contrastados con los obtenidos por una técnica clásica de recuperación de imágenes basada en el cálculo de histogramas de color.

3.2 Estado del Arte

3.2.1 Clasificación de los sistemas de recuperación

Los sistemas de recuperación de imágenes se dividen en dos grandes bloques: basados en texto y basados en contenido. Los sistemas basados en texto utilizan palabras clave o *keywords* para búsqueda de imágenes con un determinado contenido. Aunque son efectivos para determinados fines, presentan numerosas limitaciones. En primer lugar, la necesidad de anotar manualmente las imágenes puede resultar una tarea tediosa para grandes volúmenes de imágenes. En segundo lugar, como se ha visto anteriormente, no existe una única percepción de una imagen lo que dificulta aún más la tarea de anotación y de búsqueda basada únicamente en texto.

Para solventar estos inconvenientes, aparecen las arquitecturas basadas en contenido[15]. Estas nuevas técnicas realizan la anotación de las imágenes de manera automática, lo que evita todo el trabajo de anotación manual. Las técnicas CBIR (*Content-Based Image Retrieval*) utilizan la información visual como la textura, el color y la forma para representar una imagen. La Figura 3-3 ilustra la clasificación básica de los sistemas de recuperación de imágenes.

En la actualidad, prácticamente la totalidad de los sistemas de recuperación visual se basan en la combinación de las dos técnicas anteriormente mencionadas, es decir, recuperación mediante texto y recuperación por contenido.

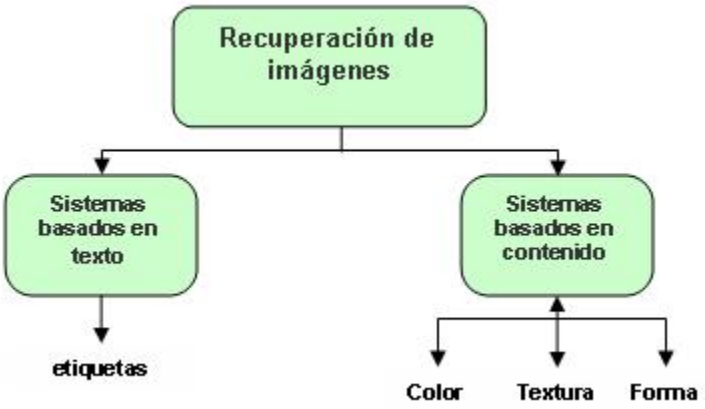

Figura 3-3: Sistemas de recuperación de imágenes

3.2.2 Estudio de algunos sistemas de recuperación reales

Desde principios de la década de los 90, la recuperación de imágenes basada en contenido ha sido objeto de numerosas investigaciones. Fruto de estos trabajos, surgen diversas opciones en lo que se refiere a la forma de realizar una consulta y presentar los resultados:

- Acceso aleatorio.
- Búsqueda por consulta o ejemplo.
- Búsqueda mediante un dibujo.
- Búsqueda por texto (keywords o palabras clave).
- Búsqueda por categorías de imágenes.

Son necesarios estudios que impliquen a usuarios reales en un uso práctico de los sistemas para explorar las compensaciones entre las diferentes opciones de búsqueda mencionadas anteriormente. Este apartado tiene como objetivo el enumerar los sistemas de recuperación de imágenes más representativos destacando sus distintas características tanto en lo que se refiere a los descriptores visuales utilizados como a las técnicas de consulta y presentación de resultados.

La Tabla 3-1 muestra los sistemas estudiados así como los conjuntos de descriptores asociados.

Sistema	*Descriptores*
Blobworld[17]	Color, textura, posición espacial y forma
ImgSeek[18], Retrievr[18]	Color, *wavelets*
QBIC[19]	Color, textura, *keywords*, forma, rectángulos de color orientados, posición espacial.
PhotoBook[20]	Color, textura, forma
SIMPLIcity[21]	Color, textura, forma, localización espacial.
FOCUS[22]	Histogramas de color

Tabla 3-1: Sistemas de recuperación de imágenes y conjuntos de descriptores asociados

En el sistema **Blobworld** [17] el usuario selecciona primero una categoría, que limita el rango de búsqueda. Tal y como se observa en la Figura 3-4, en una imagen inicial se escoge una región (blob). Seguidamente, el usuario debe indicar la importancia del color, de la textura, de la posición y de la forma. Para efectuar la búsqueda, se puede seleccionar más de una región. Los resultados muestran la tasa de parecido de la región de cada una de las imágenes que más se aproxima a las características definidas por la consulta.

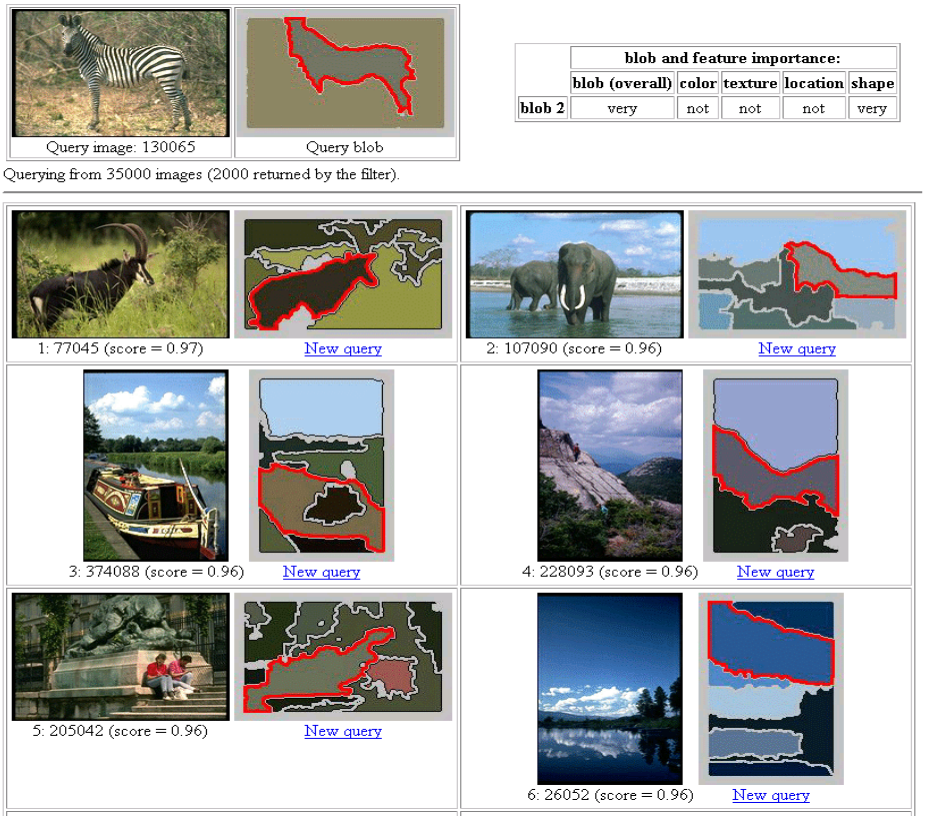

Figura 3-4: Resultados de una consulta con el sistema Blobworld

Dos de las propuestas más atractivas en este sentido, **ImgSeek** y **Retrievr** [18], se basan en un dibujo o en una imagen para realizar la búsqueda en una selección de imágenes proporcionadas por Flickr[3]. Ambos utilizan algoritmos basados en la *transformada wavelet*[4].

Las Figura 3-5 y Figura 3-6 muestran dos ejemplos de los resultados obtenidos con los sistemas Retrievr e ImgSeek respectivamente. La interfaz de ambos es similar: en la parte izquierda se escoge un color y un diámetro de pincel determinado y se realiza un dibujo en una pequeña área. Tras un par de segundos, obtenemos los resultados de las imágenes similares al dibujo realizado.

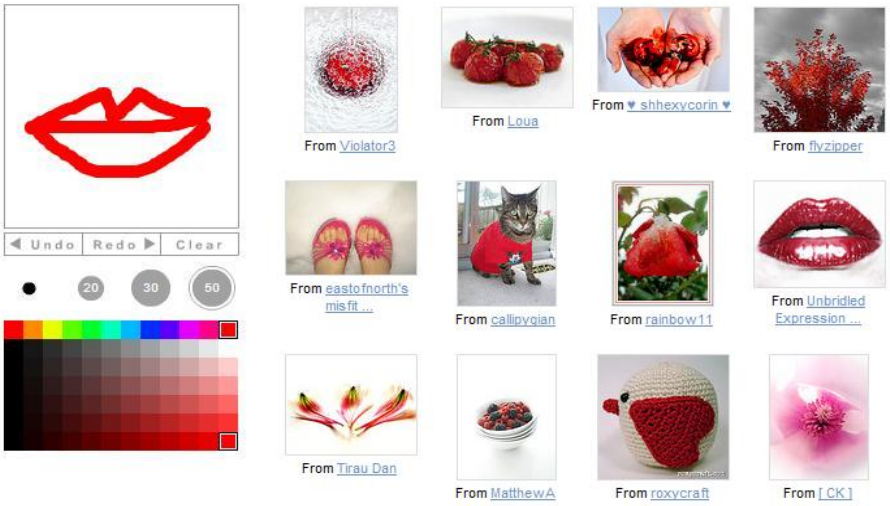

Figura 3-5: Dibujo y resultados de la consulta con Retrievr

[3] http://www.flickr.com/

[4] La *transformada wavelet* parte de una representación espacial en puntos de la imagen para obtener una serie de componentes que se van refinando para crear una representación multiescala de la imagen. A partir de los coeficientes en que se ha descompuesto la imagen es posible calcular una representación numérica de características que se manifiesten en la imagen de manera que sea posible aplicar un esquema de ordenación en base a un criterio de similitud con la consulta inicial. Las *wavelets* han demostrado ser una efectiva herramienta para análisis de imágenes debido al hecho de ser capaces de capturar la información espacial junto a las características visuales [23] .

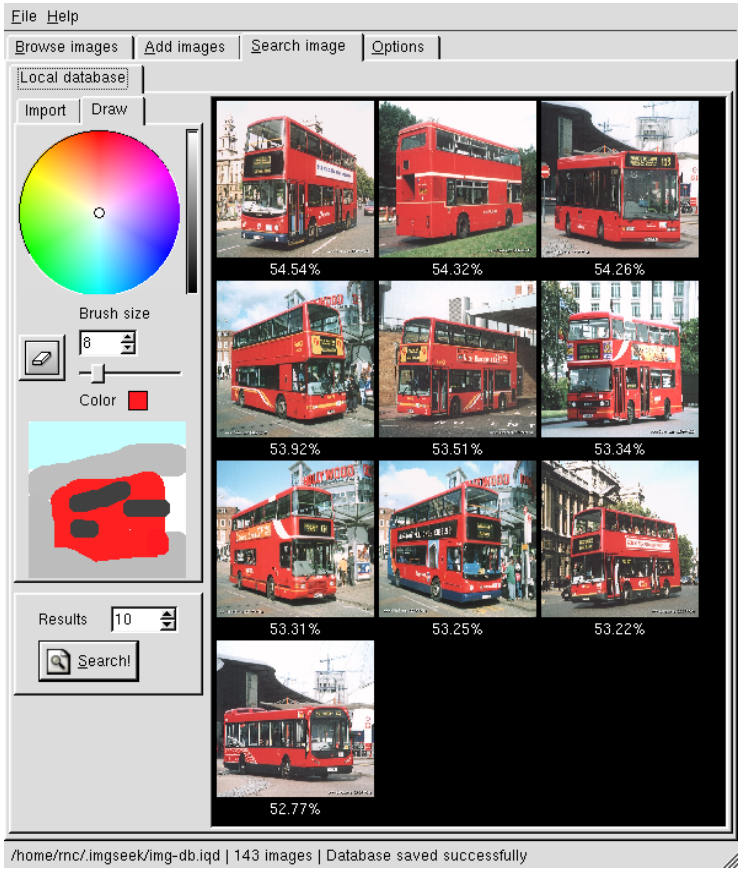

Figura 3-6: Interfaz gráfica y resultados de una consulta con Imgseek.

Por su parte IBM desarrolló en 1995 **QBIC** (Query by Image Content) [19] uno de los primeros sistemas de recuperación de imágenes, y en el que se han basado las arquitecturas desarrolladas posteriormente. Nos presenta la posibilidad de buscar imágenes estableciendo una proporción de los colores que aparecen. Las últimas versiones de QBIC realizan la búsqueda combinando palabras clave y contenido de las imágenes. Una aplicación de este sistema la encontramos en el sitio web del museo del Hermitage[5].

Otra posibilidad más sofisticada nos ofrece ***Photobook*** [18] que aplica una técnica de aprendizaje supervisado conocido como *relevance feedback* (requiere la presencia de un observador humano que refine las consultas realizadas en esta fase) para ajustar las etapas de segmentación y elección de parámetros para la clasificación.

Un gran número de sistemas se basan en la segmentación de regiones por su color y textura. Por ejemplo, **SIMPLIcity** [21] es capaz de clasificar las imágenes automáticamente y crear una base de datos según su contenido semántico y **FOCUS** [22] realiza búsquedas de regiones dentro de las imágenes. Su método consta de dos fases: una

[5] www.hermitagemuseum.org

primera fase para la detección de picos en el histograma de la imagen que permite recuperar las más semejantes y una segunda fase para detección de regiones y codificación del color en cada región que elimina los falsos resultados positivos de la recuperación de la primera fase.

3.3 Diseño

3.3.1 Arquitectura del Sistema

En general, un sistema eficiente de recuperación de información visual debe ser capaz de:

- Definir elementos de recuperación que sean significativos en el contexto de la aplicación, es lo que se denomina modelado de las imágenes. La importancia de esos elementos, descriptores, radica en su capacidad para caracterizar las imágenes y permitir su procesado. Los métodos de búsqueda basados en descriptores son extremadamente eficientes. Es por este motivo que es la forma más utilizada de enfocar las operaciones de búsqueda en el ámbito de las imágenes. Pero hay dos problemas principales en la aproximación basada en descriptores. Por un lado el determinar qué descriptores hay que utilizar según lo que se desea buscar (este punto ha sido abordado en detalle en el apartado 2.3.2.3) y por otro lado la representación de una determinada base de datos en forma de descriptores.
- Proporcionar un método de consulta que permita al usuario especificar de forma natural características selectivas así como información imprecisa: se denominan técnicas de búsqueda. En nuestro caso, emplearemos la técnica de "consulta mediante ejemplo" (*Query By Example*) que consiste en generar una búsqueda a partir de una imagen de referencia proporcionada por el usuario.
- Definir métricas de igualdad o importancia que sean satisfactorias para la percepción del usuario, denominadas medidas de similitud.

La Figura 3-7 muestra un esquema de la arquitectura del sistema de recuperación propuesto. Como se ha visto en el capítulo anterior, se requiere un paso previo de segmentación que diferencie las regiones de las que se compone la imagen. Una vez segmentadas, las imágenes se filtran con el fin de eliminar las regiones de menor interés

(en este caso se considerará como no significativas las regiones formadas por un número de píxeles inferior a un mínimo).

El siguiente paso es la extracción de los descriptores de cada región. Recordemos que cada imagen lleva asociada una serie de regiones que la componen y a su vez cada región de la imagen se caracteriza mediante su conjunto de descriptores. Esta estructura es la que nos permite comparar fácilmente las regiones de una imagen con las de la imagen de referencia. El resultado es un valor de similitud que permite ordenar las imágenes por su mayor semejanza con la imagen de referencia.

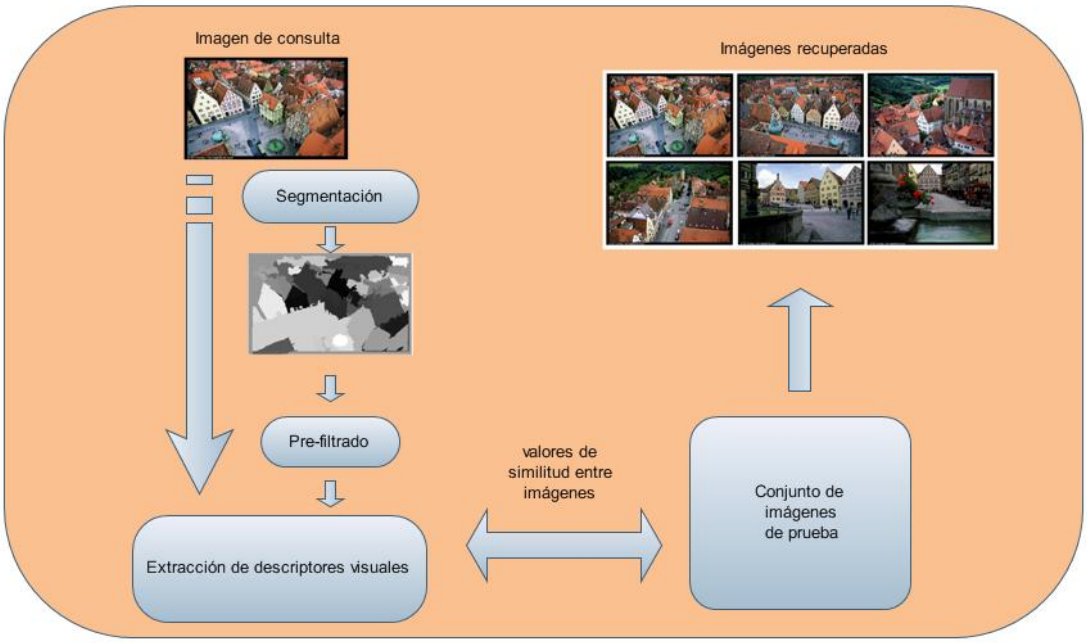

Figura 3-7: Arquitectura del sistema de recuperación

Partiendo de esta arquitectura básica, se estudiarán dos propuestas que tratan los dos algoritmos desarrollados para la comparación de regiones entre imágenes. Antes, es preciso aclarar algunos de los términos que aparecen en la descripción de ambas propuestas:

- ***Imagen de Consulta***: También denominada *query*, es la imagen que tomamos de referencia para la comparación con las imágenes de la base de datos.
- ***Imagen de Búsqueda****:* Cada una de las imágenes de la colección que se comparan con la imagen de consulta.
- ***Descriptor/es preferente/s***: Se define como el descriptor (o descriptores) considerado de mayor relevancia en la comparación.

3.3.1.1 Comparación Basada en Descriptor Preferente (Algoritmo DP)

La comparación se basa en cuatro de los descriptores desarrollados (ver sección 2.3.2.2): Color Medio, Distancia entre Centros de Masas, Relación de Tamaño y Relación de Compacidad.

El algoritmo de comparación entre regiones funciona de la siguiente manera: Dentro del grupo de descriptores $\mu_1, \mu_2, \ldots, \mu_N$ a utilizar se seleccionará un *descriptor preferente* μ^* que dependerá del tipo de búsqueda que efectuemos (ver apartado 2.3.2.3). En nuestro caso, tomaremos como *descriptor preferente* la combinación de color-posición, es decir, Color Medio y Distancia entre Centros de Masas. El primero nos permite ver las regiones que más se aproximan en base al color y el segundo nos permite descartar las regiones similares en color que se encuentran distanciadas. Una vez realizado este primer "filtrado" los otros dos descriptores (la Relación de Tamaño y la Compacidad) nos proporcionarán la información necesaria adicional sobre el tamaño y forma de la región para determinar la similitud entre las regiones.

Con el fin de hacer el algoritmo más flexible a otro tipo de consultas, se asigna un peso $p_1, p_2, \ldots p_N$ variable entre 0 y 1 a cada uno de los descriptores (incluido el *descriptor preferente*). Supongamos que tenemos dos regiones: la región *i* que pertenece a la imagen de consulta A, y la región *j* que pertenece a la imagen de búsqueda B.

Denominamos $d_1, d_2, \ldots, d_*, \ldots, d_N$ a las distancias entre los descriptores $\mu_1, \mu_2, \ldots, \mu^*, \ldots, \mu_N$ de las regiones i, j. Dichos valores se obtienen de la siguiente forma:

- La distancia entre los centros de masa de las regiones i, j es directamente el descriptor inter-región Distancia entre Centros de Masa (ver sección 2.3.2.2.2) dividido entre la diagonal de la imagen para que su valor quede comprendido entre 0 y 1.

$$d_{ij\,posición} = \frac{Distancia\ de\ masas\ (i,\ j)}{diagonal}$$

- La distancia entre los valores de Color Medio se calcula en dos pasos. Primero se halla la distancia entre los valores medios de cada plano de color RGB, divididos entre 255 para que dé un valor entre 0 y 1. De esta forma evitamos que exista una compensación entre ellos. El segundo paso es hacer la media aritmética de las tres distancias obtenidas.

$$d_R = \frac{\left|\overline{R}_i - \overline{R}_j\right|}{255} \quad d_G = \frac{\left|\overline{G}_i - \overline{G}_j\right|}{255} \quad d_B = \frac{\left|\overline{B}_i - \overline{B}_j\right|}{255}$$

$$d_{ijcolor} = \overline{(d_R, d_G, d_B)}$$

- El *descriptor preferente* escogido para la comparación es la combinación de los descriptores de color y posición. Por tanto, la distancia de este descriptor será la suma de las dos distancias respectivas:

$$d_{ij}^{*} = d_{ijcolor} + d_{ijposición}$$

- La distancia entre los valores de tamaño se obtiene directamente del descriptor inter-región *Relación de Tamaño*, mediante la siguiente fórmula:

$$d_{ij_{tamaño}} = 1 - RelacionTamaño(i, j)$$

  Conviene recordar que el descriptor Relación de Tamaño devuelve el menor cociente de los dos posibles, por tanto, la distancia será un valor comprendido entre 0 (si las dos regiones tienen el mismo tamaño) y 1.

- La distancia entre los valores de compacidad se halla de forma análoga a la distancia por tamaño.

$$d_{ijcompacidad} = 1 - RelacionCompacidad(i, j)$$

Para cada región de la imagen de consulta A se realiza la búsqueda de la region j que más se parezca dentro de cada imagen de búsqueda B tomando como criterio el *descriptor preferente* μ^*; dicho de otra forma, se halla la región que minimiza la distancia d^*. Una vez hallada, se calcula la distancia con los demás descriptores. Estas distancias se calculan para todas las regiones de la imagen de consulta y se guardan por separado en d*, $d_{tamaño}$ y $d_{compacidad}$.

$$d^* = \frac{\sum_{i=1}^{\#regionesA} d_{ij}^{*}}{\#regionesA} \qquad d_{tamaño} = \frac{\sum_{i=1}^{\#regionesA} d_{ijtamaño}}{\#regionesA}$$

$$d_{compacidad} = \frac{\sum_{i=1}^{\#regionesA} d_{ijcompacidad}}{\#regionesA}$$

El parecido global entre las dos imágenes A y B se obtiene a partir de la suma de la distancia de cada descriptor multiplicado por el peso asignado a ese descriptor.

$$D_{AB} = \sum_{n=1}^{N} d_n p_n$$

siendo N el número total de descriptores evaluados y D_{AB} la distancia global entre las dos imágenes A y B.

La Figura 3-8 muestra un ejemplo de funcionamiento del algoritmo. La imagen A corresponde a la imagen de consulta mientras que B es la imagen sobre la que evaluamos el parecido. Una vez segmentadas ambas imágenes y extraídas las regiones con sus respectivos descriptores, se procede a la búsqueda de la región de B que más parecido tenga con la región de A según el criterio del descriptor preferente elegido (en este caso la combinación de color medio y posición).

Una vez encontrada la región, se evalúan las distancias con los demás descriptores. Este proceso se repite con todas las regiones de A obteniéndose un valor de parecido entre las dos imágenes.

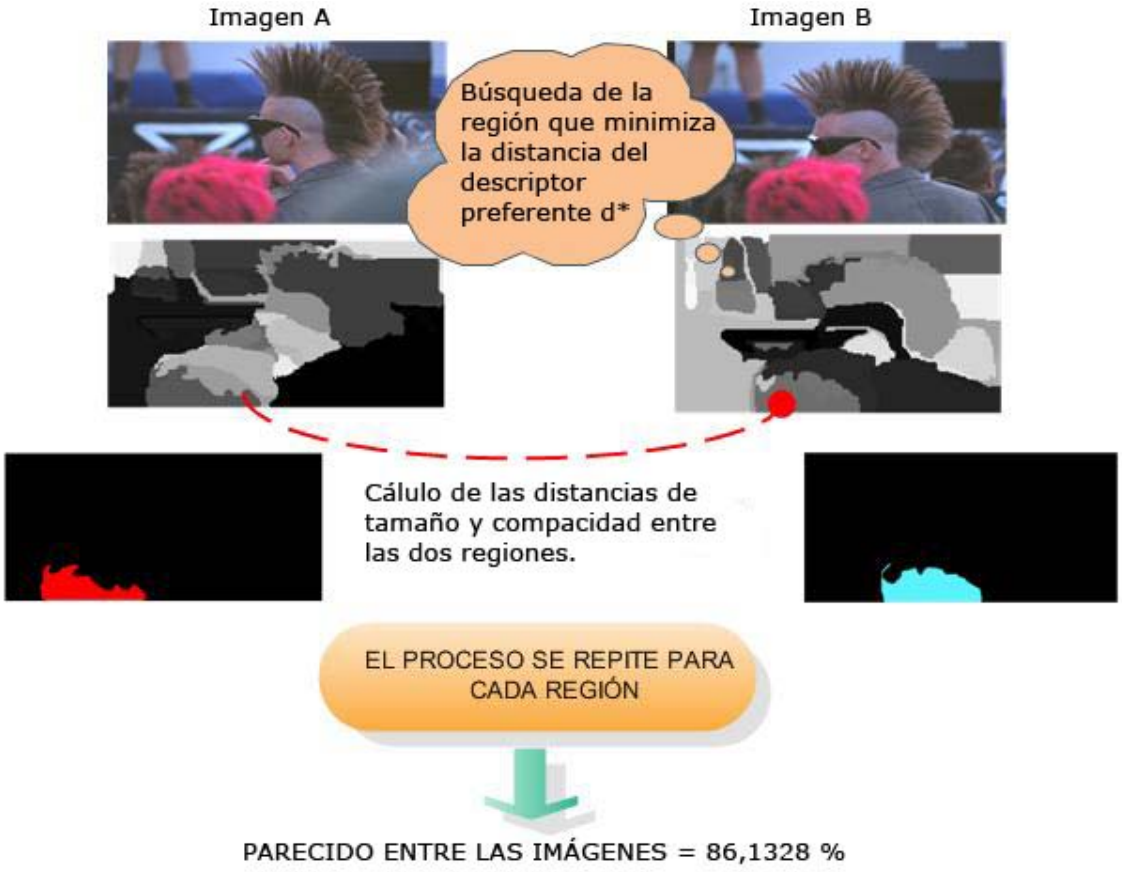

Figura 3-8: Esquema de funcionamiento del algoritmo basado en la elección de un descriptor preferente.

3.3.1.2 Comparación basada en Matriz de Distancias (Algoritmo MD)

Tras analizar los resultados obtenidos con el algoritmo propuesto inicialmente, observamos que en ocasiones no se comportaba de la forma deseada. Al centrarse en la búsqueda de una sola región, la más parecida, podía proporcionarnos como resultado una imagen en la que una sola de las regiones se pareciera a todo el conjunto de regiones de la imagen de consulta. Por lo tanto, era necesario implementar una adaptación del algoritmo de comparación que incluyera este hecho, es decir, que tuviese en cuenta el número de regiones que se encuentran parecidas a una dada y el tamaño de éstas.

La métrica de comparativa se basa, en los descriptores color medio, distancia entre centros de masa y relación de tamaño de las regiones. Sin embargo, hemos tenido en cuenta algunas modificaciones con respecto al algoritmo anterior:

- Se ha añadido pequeña corrección a la distancia para que tenga en cuenta que la separación entre dos regiones con un alto valor de compacidad puede ser grande

aunque dichas regiones estén próximas espacialmente. La corrección consiste en dividir el valor de distancia obtenido entre el Perímetro Ideal (ver definición del descriptor Compacidad en el apartado 2.3.2.2.1) de ambas regiones.

$$PerimetroIdeal = 2 \times \pi \sqrt{\frac{N}{\pi}}$$

Para entender esta corrección, en la Figura 3-9 aparecen dos regiones espacialmente muy próximas (de hecho, se solapan). Sin embargo, el valor del descriptor Distancia entre Centros de Masa es demasiado alto debido al tamaño de las regiones y su alta compacidad. Tras la corrección, el valor obtenido disminuye y se ajusta más a la distancia real entre las regiones.

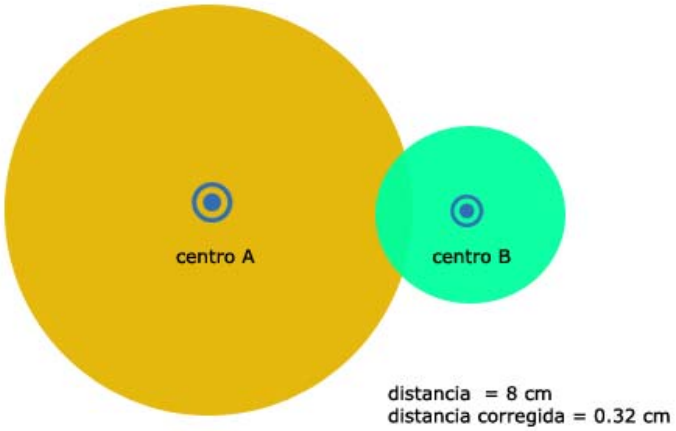

Figura 3-9: Ejemplo de corrección de la distancia

Para relacionar los descriptores color y posición se ha optado, tras varios experimentos entre varias distancias (euclídea, raíz de la suma, raíz del producto), por la raíz cuadrada del producto de los valores de distancia de ambos descriptores. Así, la distancia entre dos regiones i y j sería:

$$d_{ij} = \sqrt{d_{ij}color \times d_{ij}posición}$$

En la primera versión del algoritmo comparábamos cada región de la imagen A con todas las regiones de la imagen B, quedándonos con la región j que obtuviera la menor distancia del descriptor preferente.

La primera mejora del nuevo algoritmo consiste en guardar todos los resultados de las comparaciones en una *matriz de distancias*. Cada distancia d_{ij} es un valor de la matriz

donde i representa a una región de la imagen de consulta y j a una región de la imagen de búsqueda.

$$Matriz\ de\ distancias = \begin{pmatrix} d_{11}\ d_{11}\ \ldots\ d_{1j}\ \ldots\ d_{1R_A} \\ d_{21} \\ \ldots \\ d_{i1} \qquad d_{ij} \\ \ldots \\ d_{R_B 1} \qquad\qquad d_{R_B R_A} \end{pmatrix}$$

R_A = *número de regiones de la imagen A*

R_B = *número de regiones de la imagen B*

En base a un conjunto de pruebas realizadas sobre el algoritmo (ver anexo C), se escoge un *umbral de decisión* β para la distancia por debajo del cual una región se considera parecida a otra. El siguiente paso es el filtrado de los valores guardados en la matriz que estén por debajo de β (los que corresponden a las regiones que se parecen).

Con el propósito de incluir el tamaño de las regiones en la comparación, definimos lo que hemos denominado *tasa de cobertura*. Para cada región i de la imagen A sumamos el número de píxeles de todas las regiones que se parecen de la imagen B. La tasa de una región i contenida en la imagen A sería esta suma entre el número de píxeles de la región i. Se trata de un valor entre 0 y 1; si la suma de píxeles excede al de la región, se considerará que la tasa es 1. El proceso se repite para cada región j de la imagen B con respecto a cada región de A.

$$if\ d_{ij} < \beta \qquad tasaA_i = \frac{\sum_{j=1}^{R_B} NumeroPixeles_j\ x\ \left(1 - d_{ij}\right)}{NumeroPixeles_i}$$

$$if\ d_{ji} < \beta \qquad tasaB_j = \frac{\sum_{i=1}^{R_A} NumeroPixeles_i\ x\ \left(1 - d_{ji}\right)}{NumeroPixeles_j}$$

$$tasaTotalB = \sum_{i=1}^{R_A} tasaA_i\ x\ NumeroPixeles_i$$

$$tasaTotalB = \sum_{j=1}^{R_B} tasaB_j\ x\ NumeroPixeles_j$$

El coeficiente de parecido entre las imágenes A y B se obtiene mediante el producto de las dos tasas de cobertura obtenidas.

3.4 Pruebas y resultados

Este apartado tiene dos objetivos principales. Por un lado, demostrar mediante un conjunto de pruebas realizadas la utilidad de los dos algoritmos implementados para la recuperación de imágenes por contenido, destacando las mejoras introducidas por el algoritmo basado en la matriz de distancias. Por otro lado, realizar una comparativa tanto cuantitativa como cualitativa del sistema propuesto con un sistema de recuperación basado en histogramas de color de la imagen [25].

3.4.1 Descripción de la evaluación cuantitativa. Diagramas precision-recall

La calidad del sistema de recuperación de información está determinada por los resultados comparativos entre imágenes recuperadas e imágenes relevantes para una consulta dada. Existen otras medidas de calidad referentes a velocidad de procesamiento y espacio de almacenamiento, pero en nuestro caso hemos obviado estas medidas, centrándonos en la evaluación de la recuperación.

La evaluación de un sistema de recuperación se realiza utilizando una colección de pruebas (*image collection*) perfectamente caracterizada. Esta colección consiste en un conjunto de imágenes y un conjunto de consultas. Para cada una de las consultas se han seleccionado de forma subjetiva las imágenes relevantes de la colección, creando un ground-truth para la búsqueda en esa colección (la descripción de la colección de imágenes se puede ver en el anexo B). La evaluación del sistema de recuperación consiste entonces en comparar, para cada una de las consultas de la colección de pruebas, las imágenes que el sistema ha obtenido, esto es, las imágenes recuperadas, y las imágenes marcadas como relevantes para esa consulta.

El método más habitual para medir la calidad de un sistema es la utilización de diagramas *precision−recall***.** Tomemos una consulta concreta de la colección de pruebas y el conjunto de imágenes relevantes para dicha consulta. Sea $|b|$ el número de imágenes relevantes. Apliquemos esa consulta al sistema que se desea evaluar. Para esa consulta se

ha recuperado un conjunto de imágenes. Sea |*a*| el número de imágenes recuperadas. La Figura 3-10 ilustra estos conjuntos.

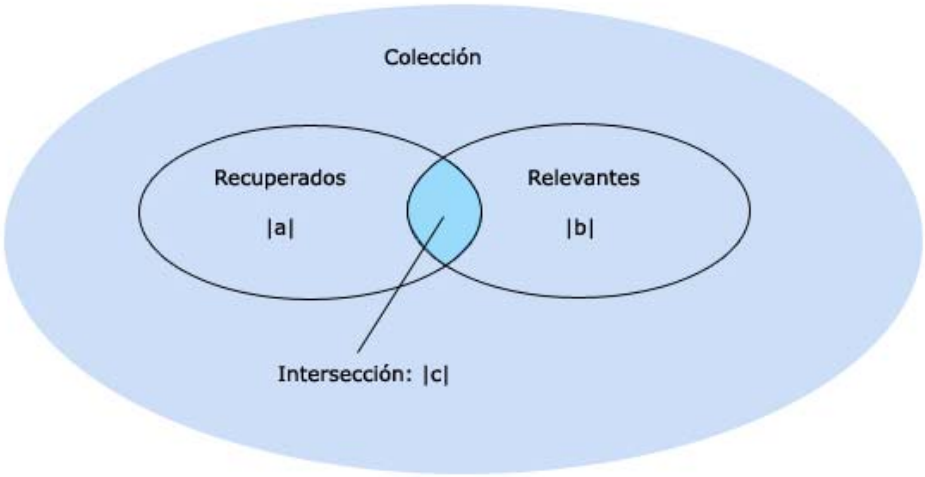

Figura 3-10: Conjuntos de elementos recuperados y relevantes

Las términos *precision* y *recall* se definen como:

Precision: determina cuántos elementos recuperados son relevantes.

$$Precision = \frac{|\,c\,|}{|\,a\,|}$$

Recall: determina cuántos elementos relevantes se han recuperado.

$$Recall = \frac{|\,c\,|}{|\,b\,|}$$

Las medidas *precision* y *recall* suponen que se han examinado todas las imágenes recuperadas. Sin embargo, al usuario normalmente no se le presentan todas las imágenes al mismo tiempo, sino que el sistema las presenta ordenadas utilizando algún criterio interno. En este proceso las medidas *precision* y *recall* van variando con cada una de las imágenes evaluadas. El método de evaluación expuesto a continuación se basa en [24] y consiste en obtener un diagrama según se van revisando las imágenes recuperadas por el sistema que son relevantes, es decir que pertenecen a la misma categoría de la consulta.

Para la explicación nos ayudaremos de un ejemplo. Supongamos que tenemos un conjunto de imágenes de prueba clasificadas en tres categorías: “Cerezos”, “Barcelona”, y “Montañas”; seleccionamos una consulta perteneciente a la categoría “Cerezos”.

Para esa consulta sabemos que hay, por ejemplo, 16 imágenes relevantes (pertenecientes a la categoría "Cerezos"). Lanzamos ahora al sistema que deseamos evaluar la consulta en cuestión, y se obtienen 20 imágenes, ordenadas de acuerdo al criterio del sistema de recuperación. La Figura 3-11 muestra los resultados obtenidos, donde las imágenes recuperadas relevantes a la consulta aparecen diferenciadas con una marca roja.

Figura 3-11: Ejemplo de conjunto de imágenes recuperadas

Analicemos paso a paso las medidas *precision y recall* para cada imagen recuperada que es relevante. Empecemos por la primera imagen relevante que se ha recuperado. Es la primera imagen, de modo que la precisión será 1 de 1 (1 relevante de 1 recuperada), es decir, del 100%. El *recall* es 1 imagen relevante de un total de 16, es decir, el 6,25%. La siguiente imagen relevante ocupa la tercera posición, es decir, la precisión es del 66,67 % (2 relevantes de 3 recuperadas), y el *recall* del 12,50% (2 relevantes de un total de 16) . La siguiente imagen relevante ocupa la séptima posición, la precisión es del 42,86 (3 relevantes de 7 recuperadas) y el *recall* del 18,75% (3 relevantes de un total de 16).

Para el resto de imágenes relevantes se ha obtenido la Tabla 3-2. Por convenio se toma que la precisión se hace cero cuando ya no quedan elementos relevantes en los recuperados.

Recall (%)	Precision (%)
6,25	100,00
12,50	66,67
18,75	42,86
25,00	50,00
31,25	45,45
37,50	46,15
43,75	50,00
50,00	42,11
56,25	0
62,50	0
68,75	0
75,00	0
81,25	0
87,50	0
93,75	0
100,00	0

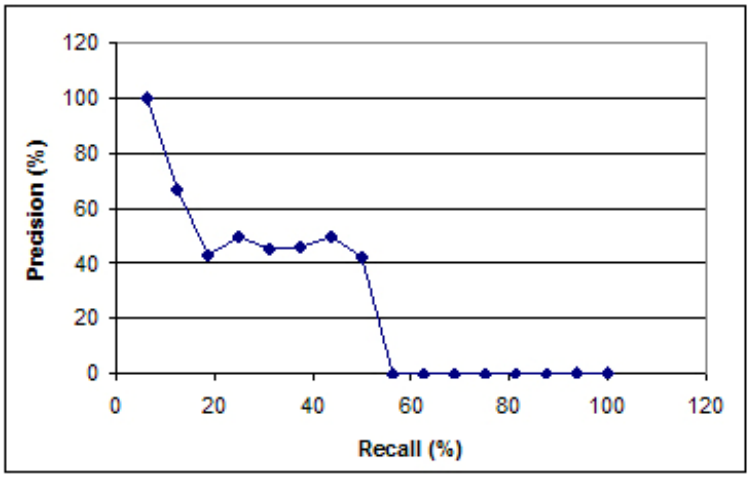

Tabla 3-2: Valores *precision-recall* para el ejemplo.

3.4.2 Descripción del conjunto de imágenes de prueba

Para las pruebas se ha creado un ground-truth propio con una colección de imágenes proporcionadas por el grupo MPEG (ver el anexo B). Se han establecido tres categorías ("gente", "rocas" y "aérea") con 20 imágenes relevantes dentro de cada una. El criterio de selección de las imágenes es subjetivo, es decir, en cada categoría las imágenes se parecen lo bastante entre sí como para distinguirse a simple vista de las otras categorías; sin embargo, se ha permitido un cierto margen de disimilitud (ver Figura 3-12).

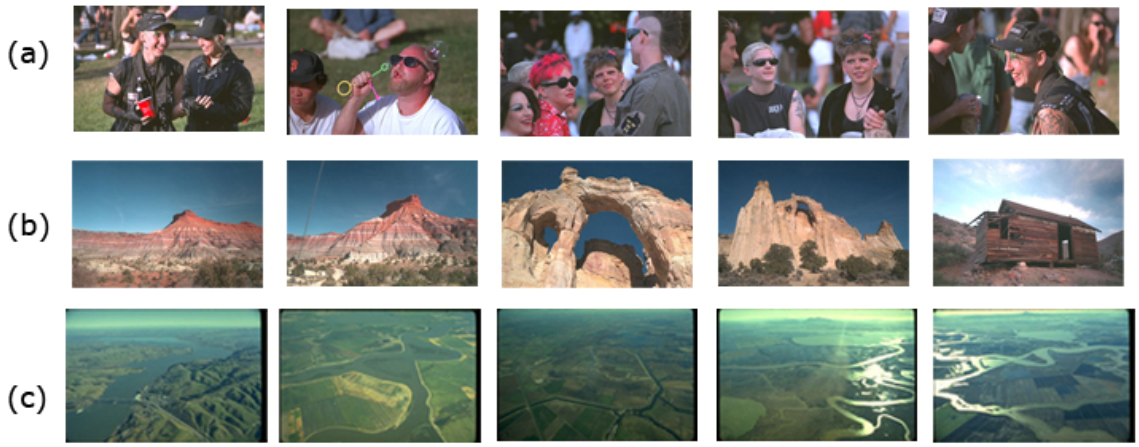

Figura 3-12: Ejemplos de imágenes de cada categoría.(a) “gente” (b) “rocas” (c) “aérea”.

Sobre la colección de imágenes seleccionada se han realizado tres consultas, que se muestran en la Figura 3-13, con el objetivo principal de obtener resultados sobre la calidad de los algoritmos de comparación. A continuación mostraremos los resultados obtenidos con las dos propuestas y con el histograma de color.

Figura 3-13: Imágenes empleadas para las consultas

3.4.3 Histograma de Color

Este apartado tiene como objeto realizar una definición del histograma de color así como la descripción de su implementación, como un paso previo a la comparativa con los sistemas propuestos, basados en descriptores de regiones.

Los descriptores de bajo nivel como son el color, la textura y la forma se emplean con frecuencia en la caracterización de imágenes, por ejemplo para la búsqueda de imágenes por consulta. Dentro de este conjunto de descriptores, el color constituye un potente descriptor visual y uno de los más utilizados en los sistemas de recuperación de

imágenes por contenido. El histograma de color consiste en la representación de la distribución de los colores en la imagen, derivado del cómputo de píxeles en cada rango (ver Figura 3-14). Puede construirse sobre diferentes espacios de color y ser representado en dos o tres dimensiones.

Los histogramas de color se utilizan frecuentemente en los sistemas de recuperación visual debido a que resultan sencillos de implementar y son muy robustos (invariantes a la rotación, a la translación y al escalado del contenido de las imágenes). No obstante, presentan ciertas limitaciones. La primera es su alta sensibilidad a los cambios de iluminación y al ruido presente en las imágenes. Otra desventaja es que el histograma de color no incluye ningún tipo de información sobre la localización de los píxeles en la imagen. En cambio, recientes estudios presentan nuevas aproximaciones del histograma de color que tratan de hacer frente a estos inconvenientes [16].

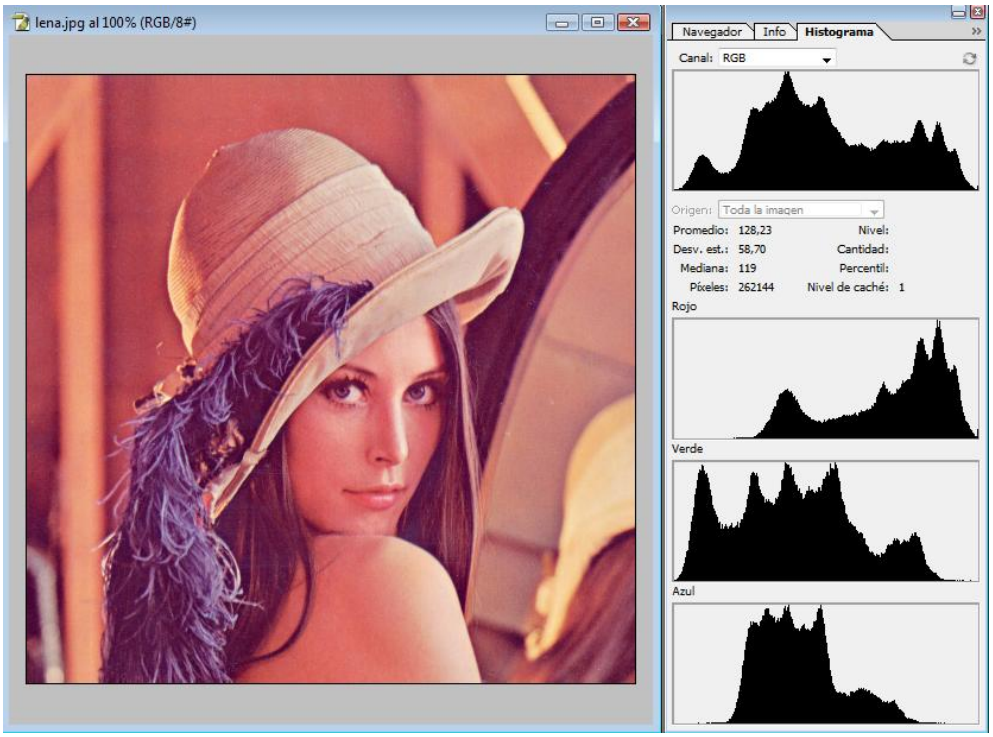

Figura 3-14: Histogramas de color de una imagen obtenidos con el programa Photoshop.

El número de divisiones en el eje del histograma puede elegirse entre 1 y 255. Con 255 divisiones tendríamos un histograma completo que para datos de tipo *double* ocuparía 3canales x 256divisiones x 8bits = 6kB. Sin embargo, suele escogerse un valor de 8 o 10 divisiones por eje porque a partir de estos valores la precisión es prácticamente la misma, en cambio, su tamaño en memoria se reduce a 30bytes.

Por otro lado, para obtener la similitud entre los histogramas de color de las imágenes pueden emplearse distintas métricas. Hemos escogido la distancia euclídea basándonos en el trabajo realizado por [25] donde aparece como una de las más efectivas en la mayoría de casos de recuperación de imágenes. Su fórmula es la siguiente:

$$d^2(h_1, h_2) = \sum_{r=0}^{N-1} \sum_{g=0}^{N-1} \sum_{b=0}^{N-1} \left(h_1(r,g,b) - h_2(r,g,b)\right)^2$$

3.4.4 Resultados y comparativa

3.4.4.1 Análisis cuantitativo

Los resultados desde un punto de vista cuantitativo se obtienen mediante las curvas *precision-recall* para cada una de las consultas realizadas (ver Figura 3-13) El sistema se comportará mejor cuanto mayor sea el área bajo dicha curva.

En primer lugar, presentaremos los resultados obtenidos por los dos algoritmos propuestos y el histograma de color para cada una de las consultas realizadas.

- **Resultados obtenidos por el Algoritmo DP**

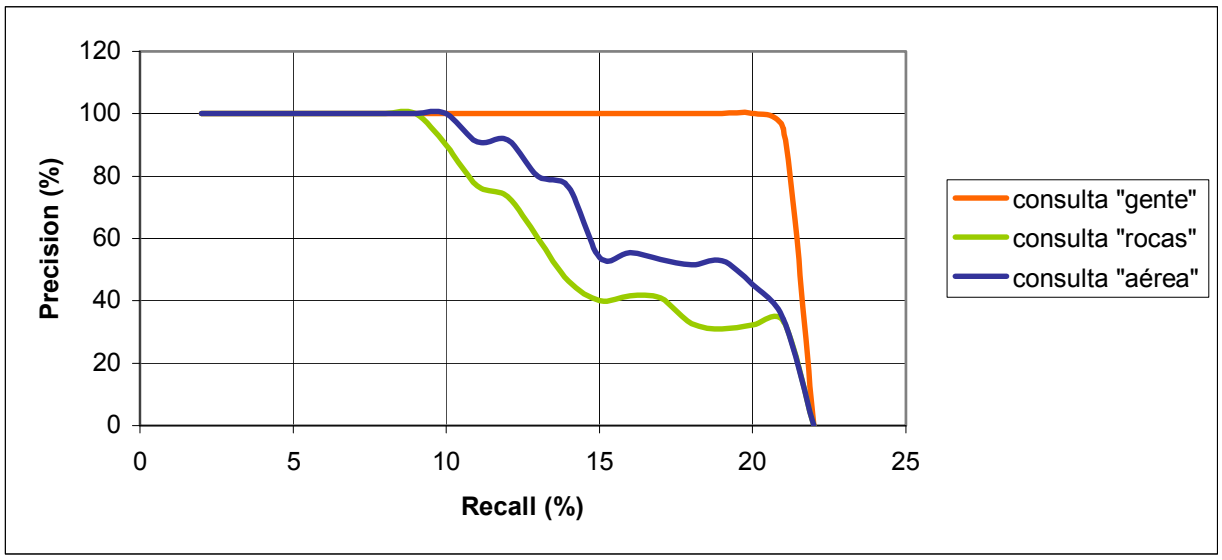

Figura 3-15: Resultados obtenidos por el algoritmo basado en un descriptor preferente.

- **Resultados obtenidos por el Algoritmo MD**

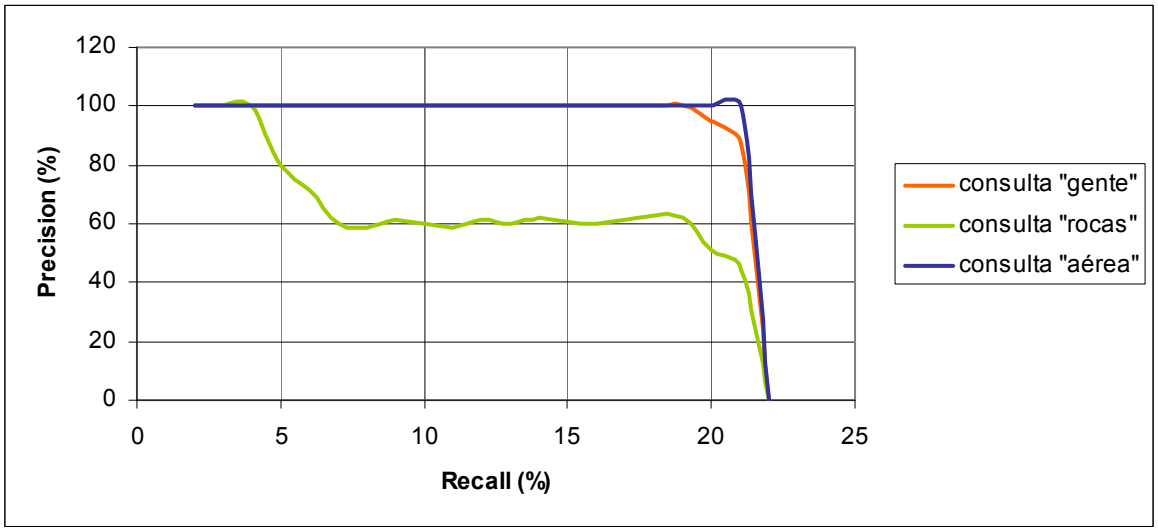

Figura 3-16: Resultados obtenidos por el algoritmo basado en una matriz de distancias.

- **Resultados obtenidos por el Histograma de Color**

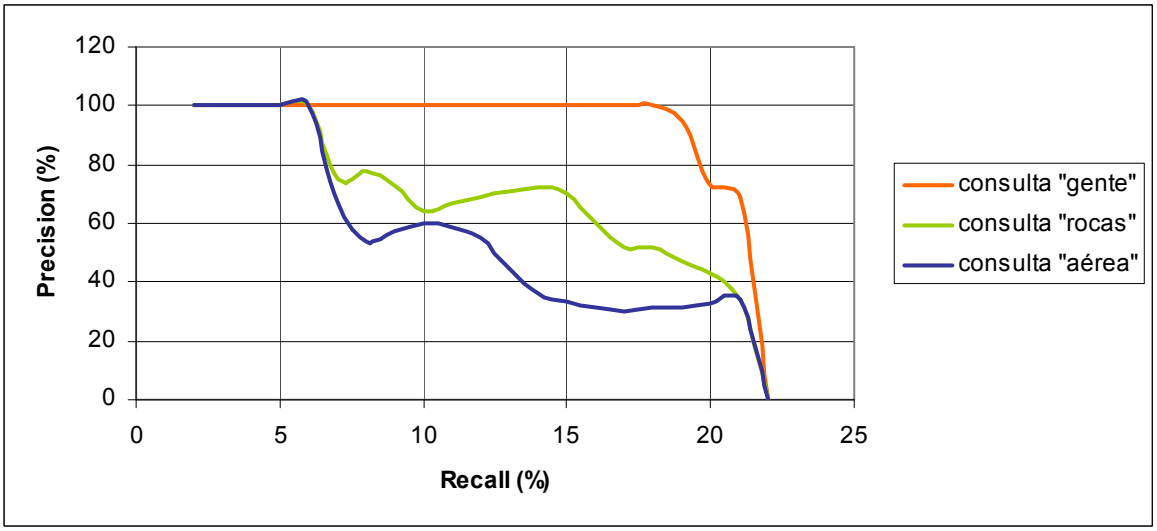

Figura 3-17: Resultados obtenidos por el Histograma de Color

Observamos que para la consulta perteneciente a la categoría “gente” los dos algoritmos propuestos y el de referencia obtienen buenos resultados, siendo más satisfactorios para el algoritmo DP.

En el caso del algoritmo DP, la curva obtenida para la consulta relativa a la categoría “rocas” presenta una alta precisión en las primeras imágenes recuperadas (las que más se

parecen), en cambio, a partir del 40% de imágenes relevantes recuperadas se produce un descenso constante. En el caso del algoritmo basado en la matriz de distancias este descenso se produce antes, alrededor del 18% lo que indica que esta consulta presenta una mayor complejidad en la recuperación de las imágenes que más se parecen. En cambio, la precisión se mantiene, en términos generales, más constante que en el caso del algoritmo basado en descriptor preferente. Podemos afirmar que, para esta consulta, los resultados obtenidos con los dos algoritmos propuestos son similares a los del algoritmo de referencia.

En el caso de la consulta perteneciente a la categoría "aérea" los resultados que se obtienen con el algoritmo basado en la matriz de distancias son visiblemente más satisfactorios.

- **Comparativa de los Algoritmos DP, MD y el Histograma de Color para cada consulta realizada.**

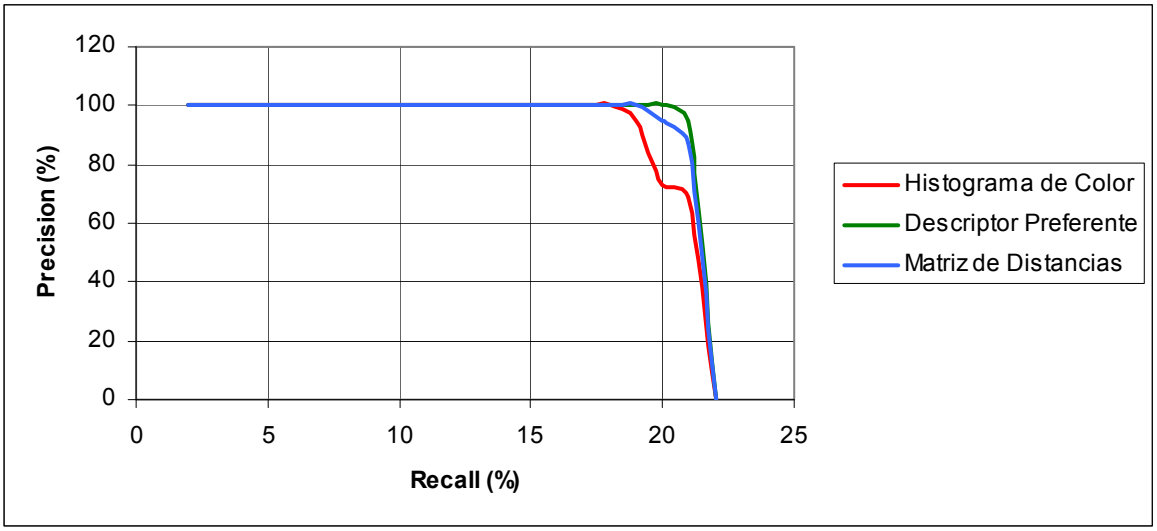

Figura 3-18: Comparativa de resultados obtenidos para la consulta relativa a la categoría "gente"

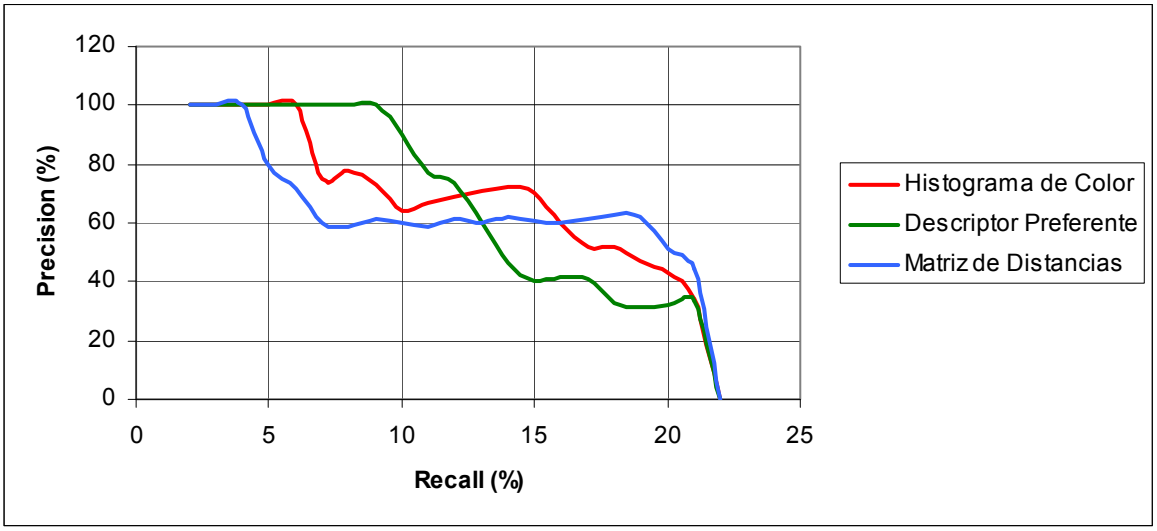

Figura 3-19: Comparativa de resultados obtenidos para la consulta relativa a la categoría "rocas".

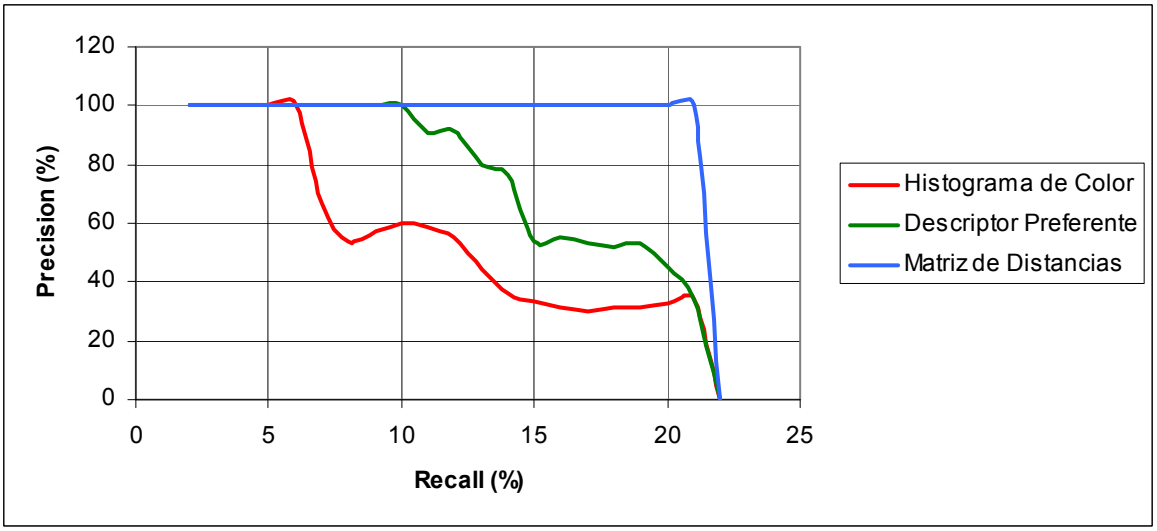

Figura 3-20: Comparativa de resultados obtenidos para la consulta relativa a la categoría "aérea"

3.4.4.2 Análisis cualitativo

El análisis cualitativo se fundamenta en una visión subjetiva de los resultados obtenidos por los dos algoritmos propuestos y el de referencia para cada una de las consultas realizadas. La imagen situada en la parte superior izquierda y bordeada en rojo es la imagen sobre la que se realiza la consulta. A continuación, de izquierda a derecha y de arriba abajo se muestran las imágenes recuperadas por el sistema siguiendo un orden de mayor parecido con la imagen de consulta.

- **Resultados de la búsqueda con el algoritmo DP**

Figura 3-21: Imágenes recuperadas para la consulta relativa a la categoría "gente" con el algoritmo DP

Figura 3-22: Imágenes recuperadas para la consulta relativa a la categoría “rocas” con el algoritmo DP

Figura 3-23: Imágenes recuperadas para la consulta relativa a la categoría “aérea” con el algoritmo DP

- **Resultados de la búsqueda con el algoritmo MD**

Figura 3-24: Imágenes recuperadas para la consulta relativa a la categoría “gente” con el algoritmo MD

Figura 3-25: Imágenes recuperadas para la consulta relativa a la categoría “rocas” con el algoritmo MD

Figura 3-26: Imágenes recuperadas para la consulta relativa a la categoría “aérea” con el algoritmo MD

- **Resultados de la búsqueda con el algoritmo basado en histogramas de color**

Figura 3-27: Imágenes recuperadas para la consulta relativa a la categoría “gente” con el algoritmo basado en histogramas de color.

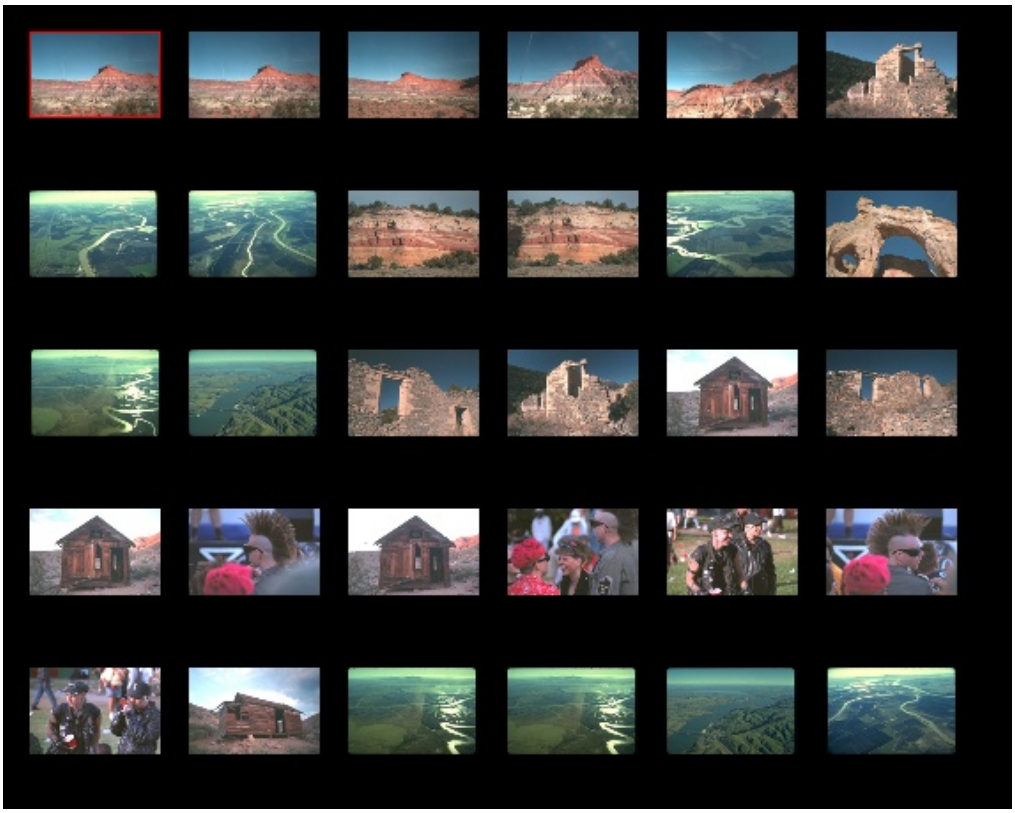

Figura 3-28: Imágenes recuperadas para la consulta relativa a la categoría “rocas” con el algoritmo basado en histogramas de color.

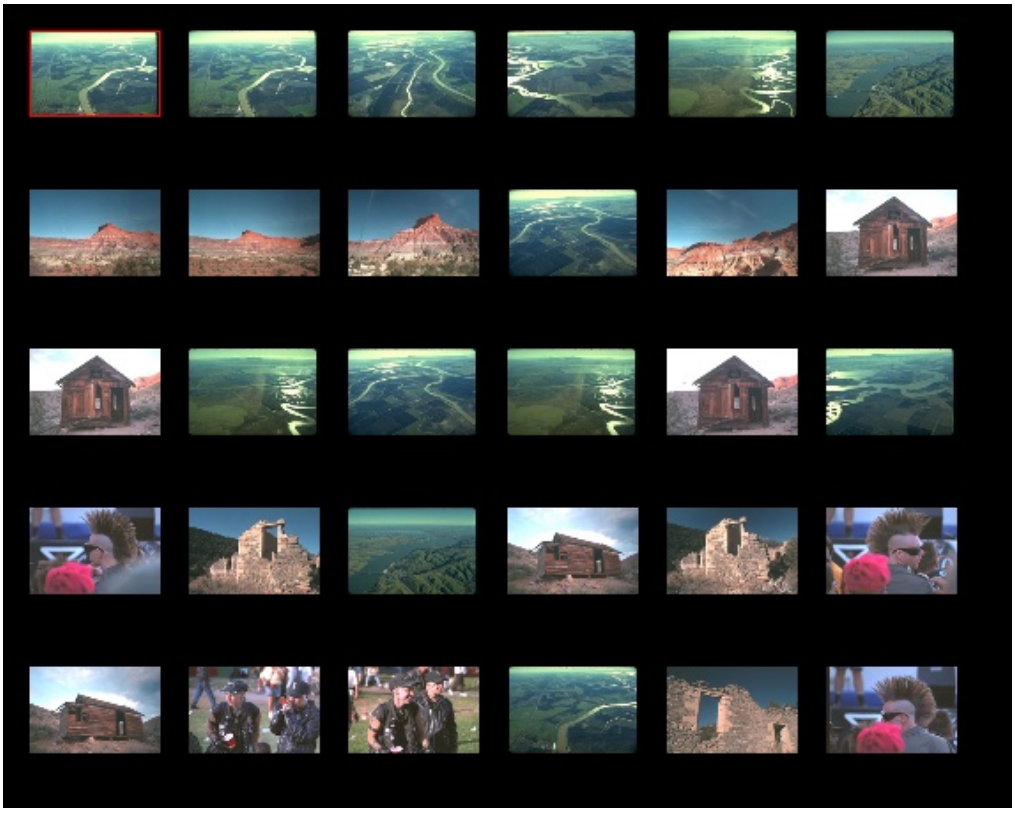

Figura 3-29: Imágenes recuperadas para la consulta relativa a la categoría “aérea” con el algoritmo basado en histogramas de color.

A continuación se presenta una figura que resume los resultados cualitativos obtenidos por los tres sistemas de recuperación.

Figura 3-30: Resumen de los resultados cualitativos obtenidos.

A través del análisis de los resultados obtenidos se pueden deducir las siguientes conclusiones:

- Los dos algoritmos (DP y MD) implementados resultan útiles para la recuperación de imágenes a partir de una *query*. Su comportamiento es similar al del algoritmo basado en histogramas de color en los ejemplos más evidentes, este es el caso de la consulta relativa a la categoría "gente", obteniendo resultados bastante satisfactorios.
- En los casos de imágenes de un parecido no tan evidente, como es el caso de la *query* perteneciente a la conjunto "rocas", tanto el histograma de color como los dos algoritmos implementados presentan dificultades para recuperar las imágenes pertenecientes al mismo grupo de la consulta. En el caso de los algoritmos DP y MD, estas dificultades pueden tener su origen en una segmentación no adecuada de las imágenes.
- En los casos de imágenes con un alto grado de parecido pero afectadas por cambios de iluminación, ruido o sobreexposición, los resultados obtenidos por los algoritmos implementados especialmente por el algoritmo MD son claramente más favorables que los que se obtienen mediante los histogramas de color.

4 Aplicación al Seguimiento de Objetos

4.1 Introducción

En su forma más simple, el seguimiento (*tracking*) puede ser definido como el problema de estimar la trayectoria de un objeto (vehículos, personas,…) en el plano de una imagen cuando éste se desplaza alrededor de una escena. Dicho de otro modo, el seguidor o *tracker* asigna una serie de etiquetas a los objetos móviles en los distintos frames de un vídeo. Adicionalmente, puede proporcionar otro tipo de información sobre el objeto, como su forma, área u orientación.

Durante las dos últimas décadas, y dentro del ámbito de la visión por computador, el seguimiento o *tracking* de objetos en secuencias de vídeo ha merecido una especial atención por parte de la comunidad investigadora. El análisis de vídeo enfocado al seguimiento de objetos consta de tres etapas clave: detección de los objetos de interés, seguimiento de dichos objetos de un frame al siguiente y por último análisis del movimiento que permite reconocer algún patrón de comportamiento. Dentro de las posibles aplicaciones del seguimiento de objetos encontramos, entre otras:

- Detección de movimiento, identificación de personas, detección automática de objetos, etc;
- Seguimiento automático de una escena para identificar actividades delictivas;
- Indexado de vídeos, anotación y recuperación automática de vídeos en bases de datos;
- Interacción persona-computador, reconocimiento de gestos, seguimiento de la trayectoria de la mirada para la entrada de datos a la computadora, etc;
- Monitorización del tráfico automovilístico en tiempo real.

En la práctica, el seguimiento de objetos no es una tarea sencilla ya que habitualmente el objeto a seguir sufre cambios de iluminación, deformaciones, rotaciones u oclusiones a lo largo del vídeo lo que dificulta o imposibilita su seguimiento. Por esta razón, la mayoría de los algoritmos de *tracking* imponen una serie de restricciones en cuanto al movimiento y la aparición de nuevos objetos en la escena. Por ejemplo, suelen asumir un movimiento de los objetos suave sin cambios abruptos y con velocidad y

aceleración constante. También es frecuente un conocimiento previo sobre el tamaño y el número de objetos que aparecen.

En la literatura podemos encontrar numerosas y variadas aproximaciones al problema del seguimiento de objetos. En la sección 4.2.3 de este capítulo se mostrará una posible clasificación de estas técnicas junto con una breve e ilustrativa descripción de los métodos más representativos dentro de cada categoría.

Partiendo de una arquitectura basada en la descripción de regiones de una imagen que ha sido descrita en el capítulo 2, y tras fijar nuestra atención en la aplicación para retrieval de imágenes a lo largo del capítulo 3; en la sección 4.3 de este capítulo se presentará otra posible utilidad del sistema para el seguimiento de objetos en una secuencia de vídeo.

4.2 Estado del Arte

Habitualmente, los sistemas de tracking requieren de tres etapas: la representación del objeto, la detección del objeto y el seguimiento del objeto. Estas etapas están estrechamente vinculadas; por ejemplo, la técnica escogida para la representación del objeto condiciona en gran parte su detección y posterior seguimiento. En los siguientes subapartados se estudiará el estado del arte para cada una de las fases mencionadas.

4.2.1 Representación del objeto

En el contexto del seguimiento de objetos, un objeto puede definirse como un área de la imagen de interés para su posterior análisis o seguimiento. Ejemplos de objetos a seguir podrían ser un barco en el mar, vehículos en una autopista, personas en un recinto o burbujas en el agua en función del dominio en que se esté trabajando. Los objetos identificados en una imagen pueden ser representados por su forma y/o su apariencia.

La Figura 4-1 ilustra una serie de posibles representaciones basadas en la forma de un objeto.

- **Puntos**. El objeto se simboliza a través de un punto, que es el centro de masas del objeto (Figura 4-1 (a)) [27] o por un conjunto de puntos (Figura 4-1 (b)) [28]. En general, esta representación basada en puntos es apropiada para objetos pequeños dentro de una imagen.
- **Formas geométricas primitivas**. La forma del objeto se representa como un rectángulo, elipse (Figura 4-1(c), (d)) [29], etc. Aunque representación más

apropiada para representar objetos rígidos simples, también se emplea en el seguimiento de objetos no rígidos.

- **Silueta del objeto y contorno**. La representación a través del contorno define los límites del objeto (Figura 4-1 (g), (h)). La region interior a ese contorno es lo que se denomina silueta del objeto (Figura 4-1 (i)). Resulta una representación adecuada para el tracking de objetos no rígidos más complejos. [30].
- **Modelos con formas articuladas**. Los objetos articulados se componen de partes del cuerpo unidas entre sí empleando juntas. Por ejemplo, el cuerpo humano está formado por el torso, la cabeza, las piernas, las manos y los pies unidos mediante articulaciones. Para representar un objeto articulado, se pueden modelar las partes que lo constituyen usando cilindros o elipses como muestra la Figura 4-1(e).
- **Modelo basado en Esqueleto**. Es posible extraer el esqueleto de cualquier objeto aplicando la MAT (Medial Axis Transform) a la silueta del objeto [31]. Este modelo se usa comúnmente en el reconocimiento de objetos [32]. La representación basada en esqueleto puede utilizarse para modelar tanto objetos articulados como rígidos (Figura 4-1 (f)).

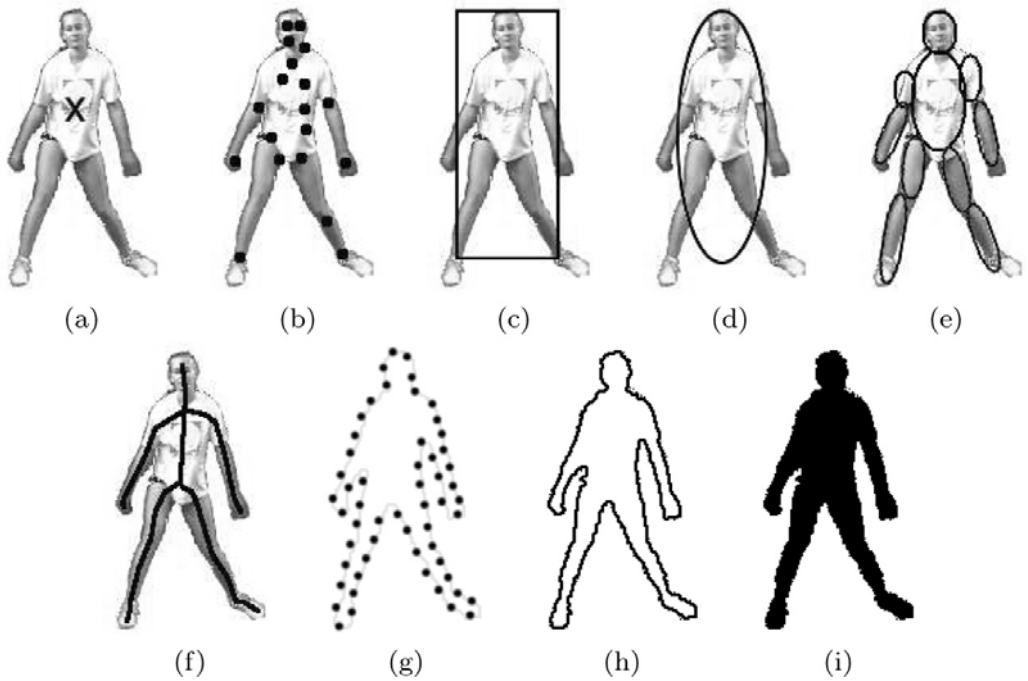

Figura 4-1: Posibles representaciones de los objetos (figura extraída de [26]). (a) Centro de Masas, (b) Múltiples Puntos, (c) Rectángulo, (d) Elipse, (e) Basada en múltiples formas geométricas (f) Esqueleto, (g)Puntos de control sobre el objeto, (h) Contorno del objeto ,(i) Silueta.

Las representaciones de la forma del objeto anteriormente vistas se combinan con información sobre otras características del objeto, como la textura y el color. Existe una gran variedad de algoritmos que procesan tales propiedades, por ejemplo [29] analiza la densidad de probabilidad a través de histogramas de color y textura; otros estudian patrones (*templates*) [33] que permiten obtener información sobre la apariencia y la posición del objeto.

En general, existe una estrecha relación entre las representaciones de un objeto y los algoritmos de *tracking*. De hecho, la representación del objeto elegida depende en gran parte del dominio de aplicación. Por ejemplo, para seguir pequeños objetos a seguir en una imagen, se podría emplear la representación basada en puntos. En cambio, si tenemos objetos de un tamaño medio o grande que se asemejan a un rectángulo o elipse resultaría más apropiada la representación basada en formas geométricas. Para formas más complejas, como figuras humanas tendría sentido utilizar una aplicación basada en la silueta o contorno.

4.2.2 Detección del Objeto

Todo método de *tracking* precisa de un mecanismo de detección del objeto que permita posteriormente realizar un seguimiento del mismo. Una aproximación común a la detección de objetos consiste en utilizar la diferencia entre un cuadro y el cuadro precedente para localizar el objeto móvil. Sin embargo, algunas técnicas emplean la diferencia con un conjunto de *frames* con el propósito de reducir el número de falsos positivos. Dado un grupo de regiones en una imagen, la labor del *tracker* será hallar el objeto que corresponde con el objeto a seguir en cada uno de los *frames*.

En la Tabla 4-1 presentamos algunos de los métodos de detección más habituales.

Categorías	**Trabajo representativo**
Detector de Puntos	*Moravec's detector* [34], *Harris detector* [35], *Scale Invariant Feature Transform (SIFT)* [36], *Affine Invariant Point Detector* [37].
Segmentación	*Mean-shift* [38], *Graph-cut* [39], *Active contours* [40].
Modelado del Fondo	*Mixture of Gaussians* [41], *Eigenbackground* [42], *Wall flower* [43], *Dynamic texture background* [44].

Clasificadores con Supervisión	*Support Vector Machines* [45], *Neural Networks* [46], *Adaptive Boosting* [47].

Tabla 4-1: Clasificación de las técnicas de detección y trabajos asociados

- **Detector de Puntos**

Los detectores de puntos se emplean para encontrar puntos de interés en imágenes, definidos dichos puntos de interés como aquellos píxeles con una textura singular. Estos puntos se asocian al contorno de los objetos, lo que posibilita su seguimiento. Una cualidad deseable de los puntos de interés es su estabilidad ante los cambios de iluminación y ante diferentas puntos de vista de la cámara.

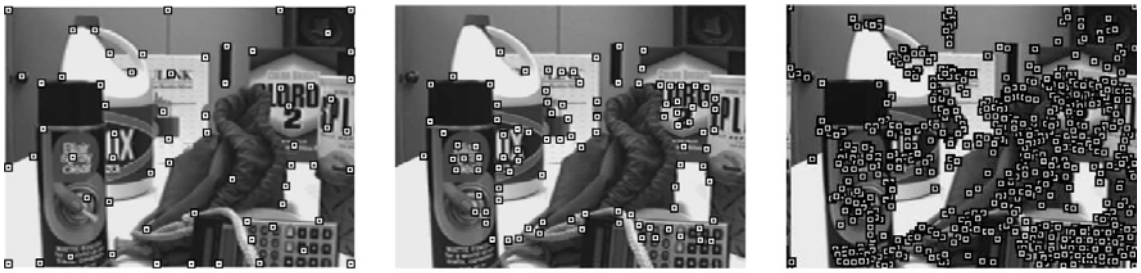

Figura 4-2: Puntos de interés obtenidos al aplicar las técnicas (a) Harris [35], (b) KLT [48], y (c) SIFT [36] (figura extraída de [26]).

- **Segmentación**

Como se vio en la sección 2.3.1 el objetivo de los algoritmos de segmentación consiste en realizar una partición de la imagen en regiones perceptualmente similares. Las dos cuestiones a resolver por todo algoritmo de segmentación son el criterio en base al que se realiza la partición y el método que permite obtener una división en regiones eficiente.

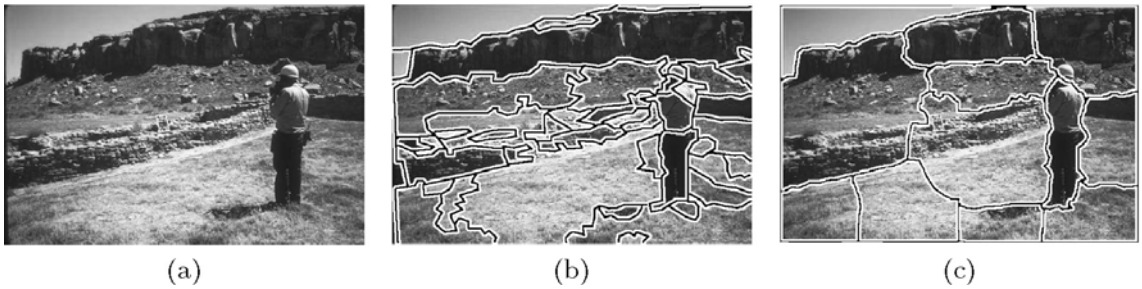

Figura 4-3: Segmentación de la imagen (a), utilizando el algoritmo mean-shift [38] (b) y mediante el algoritmo normalized cuts [39] (c) (figura tomada de [26]).

- **Modelado del Fondo**

La detección de un objeto puede obtenerse construyendo una representación de la escena denominada modelado del fondo y después encontrando variaciones de ese modelo en los *frames* siguientes. Cualquier cambio significativo en una región de la imagen respecto al modelo implicaría la localización de un objeto. Los píxeles que constituyen la región sobre la que se ha detectado el cambio, son marcados para un análisis posterior. Generalmente, se aplica un algoritmo de conexión para unir las regiones que forman el objeto. Este proceso es conocido como sustracción del fondo.

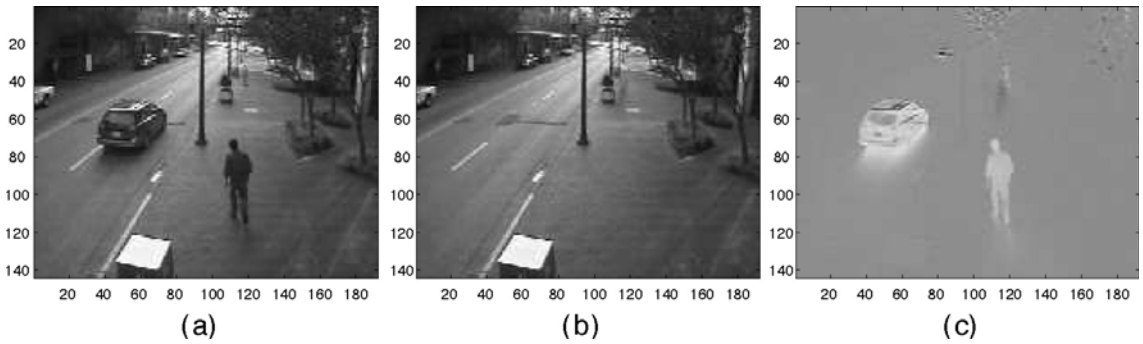

Figura 4-4: Descomposición del espacio basada en la sustracción del fondo de una imagen (figura extraída de [26]): (a) imagen de entrada, (b) Fondo de la imagen reconstruído (c) Imagen diferencia. Se puede ver que los objetos en primer plano están claramente identificados.

- **Clasificadores con Supervisión**

La detección de un objeto puede llevarse a cabo mediante un aprendizaje de las diferentes vistas de un objeto tomadas de forma automática de un conjunto de ejemplos. El aprendizaje de las diferentes vistas de los objetos no requiere un almacenamiento del conjunto completo de posibles patrones. Dado un conjunto de ejemplos de aprendizaje, las técnicas de aprendizaje con supervisión generan una función que relaciona las entradas con las salidas deseadas. Una formulación estándar del aprendizaje con supervisión es el problema de la clasificación donde se aproxima el comportamiento de una función generando una salida bien en forma de un valor continuo, conocida como regresión, o bien una etiqueta de clase lo que se denomina clasificación. En el contexto de la detección de objetos, los ejemplos de aprendizaje se componen de pares de características del objeto y una clase de objeto asociada que es definida manualmente.

La correcta selección de las características o descriptores de los objetos juega un papel importante en el resultado de la clasificación, por tanto, es fundamental elegir un conjunto de descriptores que permitan discriminar entre una clase y otra.

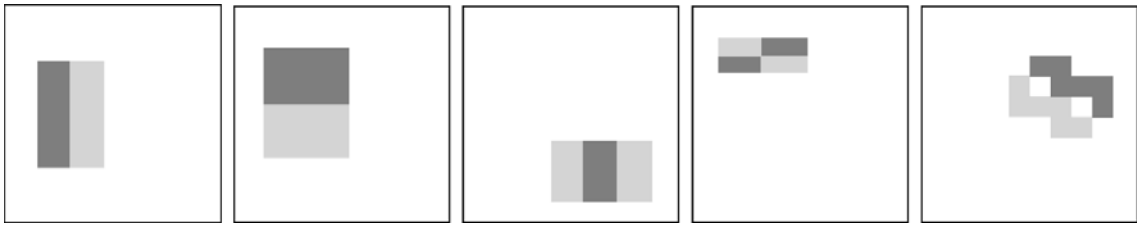

Figura 4-5 (extraída de [26]): Conjunto de filtros rectangulares empleados por [47] para extraer las características necesarias del algoritmo Adaboost. Cada filtro se compone de tres regiones: blanca, gris claro y gris oscuro, con sus pesos asociados 0, -1 y 1 respectivamente. Estos filtros se convolucionan con la imagen para obtener la característica buscada.

4.2.3 Seguimiento del Objeto

La finalidad de un *tracker* de objetos es generar la trayectoria de un objeto a lo largo de un período de tiempo mediante la localización de su posición en cada uno de los *frames* de un vídeo. Las tareas de detectar un objeto y establecer una correspondencia (localizar dicho objeto en otra imagen) a lo largo de una serie de *frames* de vídeo puede realizarse de forma conjunta o separada. En el primer caso, las regiones que componen el objeto en cada *frame* se obtienen a través de un algoritmo de detección de objetos, y la función del *tracker* en este caso sería establecer la relación entre los objetos en los diferentes *frames*. En el segundo caso, la regiones que constituyen el objeto y su correspondencia se estiman de forma conjunta mediante el análisis de los *frames* precedentes. En ambas aproximaciones los objetos se representan por su forma y/o alguno de los siguientes modelos descritos en el apartado 4.2.1. El modelo de representación del objeto escogido limita el tipo de movimiento o deformación que puede experimentar. Por ejemplo, si un objeto se representa mediante un punto sólo se puede emplear un modelo basado en la translación. La Figura 4-6 muestra una taxonomía de los distintos métodos de seguimiento de objetos y la Tabla 4-2 recoge los trabajos más representativos realizados dentro de cada categoría.

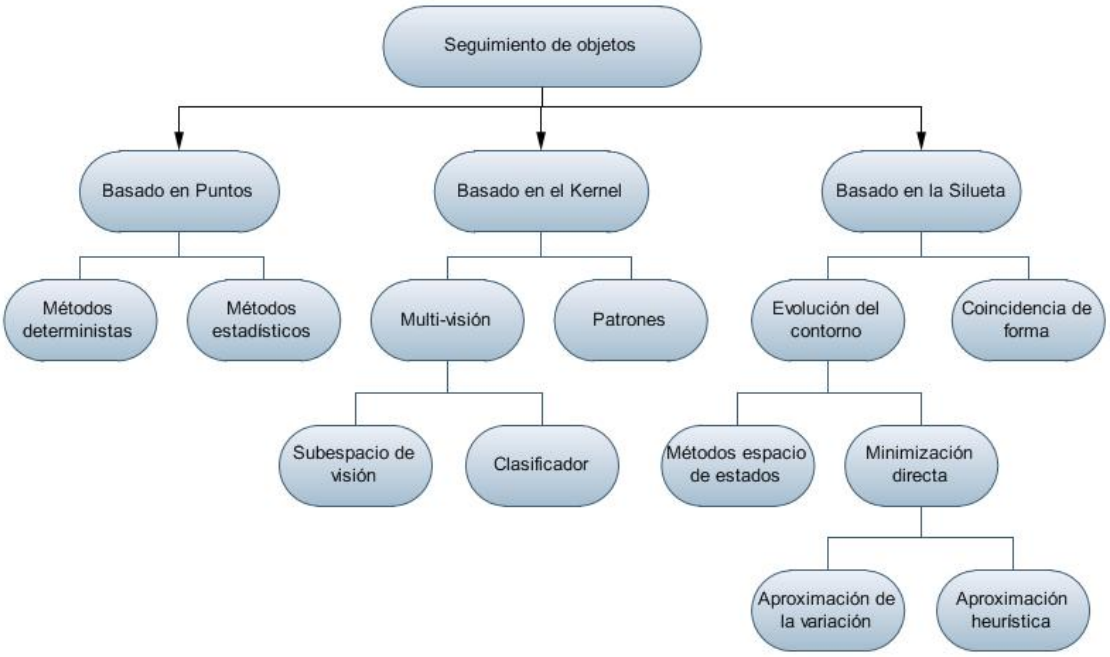

Figura 4-6: Clasificación de las técnicas de *tracking*.

Categoría	Trabajo representativo
Basado en Puntos	
Métodos determinísticos	*MGE tracker* [49], *GOA tracker* [27].
Métodos estadísticos	*Kalman filter* [50], *JPDAF* [51], *PMHT* [52].
Basado en el Kernel	
Modelos de aparición basados en patrones y densidad.	*Mean-shift* [29], *KLT* [48], *Layering* [52].
Modelos multi-visión.	*Eigentracking* [54], *SVM tracker* [55].
Basado en la Silueta	
Evolución del contorno	*State space models* [56], *Variational methods* [57], *Heuristic methods* [58].
Coincidencia de forma	*Hausdorff* [59], *Hough transform* [60], *Histogram* [61].

Tabla 4-2: Trabajos representativos dentro del seguimiento de objetos

- **Seguimiento de objetos basado en puntos**. Los objetos detectados en *frames* consecutivos son representados mediante puntos, y la asociación de los puntos se basa en el estado previo del objeto que puede incluir su posición y su movimiento. Esta aproximación requiere de un mecanismo externo que detecte los objetos en cada *frame*. Un ejemplo de la correspondencia entre objetos se muestra en la Figura 4-7(a).

- **Seguimiento de objetos basado en el kernel**. El *kernel* (núcleo) del objeto se refiere a la forma del objeto y su apariencia. Por ejemplo, el *kernel* puede ser un patrón rectangular o una forma elíptica con un histograma asociado. El seguimiento de un objeto se realiza mediante una evaluación del movimiento del *kernel* en *frames* sucesivos (Figura 4-7 (b)).

- **Seguimiento de objetos basado en la silueta.** El *tracking* se realiza estimando la region del objeto en cada *frame*. Los métodos basados en la silueta utilizan información codificada dentro de la región del objeto. Esta información puede encontrarse en forma de modelos de densidad de aparición o forma que se utilizan para generar mapas de bordes. Dados unos modelos de un objeto, se realiza un seguimiento de las siluetas mediante la búsqueda de coincidencias o bien siguiendo la evolución del contorno (Figura 4-7 (c) y (d)). Ambos métodos pueden considerarse en términos generales como una segmentación del objeto aplicada en el dominio temporal usando la información generada en los *frames* previos.

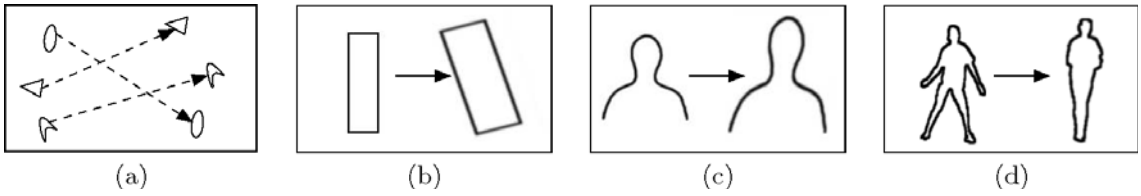

Figura 4-7 (extraída de [26]): Diferentes aproximaciones al seguimiento de objetos. (a) Correspondencia multipunto, (b) Transformación paramétrica de un patrón rectangular, (c, d) Dos ejemplos de evolución del contorno.

4.3 Diseño e implementación

El objetivo será demostrar las posibilidades de implementación de algoritmos de *tracking* de objetos a partir del sistema de caracterización de regiones de interés desarrollado anteriormente (ver capítulo 2). Para ello se ha llevado a cabo el desarrollo de un sistema sencillo que sea capaz de realizar el seguimiento de uno o varios objetos a lo largo de una

secuencia de vídeo. Recordemos brevemente la estructura del sistema del que partimos antes de estudiar el algoritmo.

- Disponemos de una clase Imagen que almacena el *frame* de vídeo original, una imagen procedente de su segmentación y un vector con las regiones que forman dicho *frame*.
- Cada región R_k de la Imagen I reúne un conjunto de valores característicos, como la posición de la región, su color medio, su tamaño, etc. Así como una etiqueta k asociada a la región R_k

Conviene aclarar que dentro de las etapas del seguimiento anteriormente definidas en la sección 4.2, nos centraremos exclusivamente en la fase de seguimiento de trayectorias de los objetos para demostrar una de las posibilidades de la caracterización de objetos basada en regiones.

Los descriptores visuales elegidos para medir la similitud entre regiones en este caso son el color medio, la relación de tamaño y la distancia entre los centros de masa de las regiones. Se establecen una serie de umbrales para cada descriptor que permiten filtrar las regiones "candidatas" al *matching* con la región a seguir. Los valores óptimos de los umbrales se determinan de forma experimental y permanecen invariables.

Los detalles sobre cada uno de los descriptores se han descrito en la sección 0 y la forma de medir las distancias es la misma que se utilizó en la aplicación de recuperación (ver sección 3.3.1.1). No obstante, recordamos la definición de las distancias empleadas para cada uno de los descriptores:

$$d_{ij\,posición} = \frac{Distancia\ de\ masas\ (i,\ j)}{diagonal}$$

$$d_R = \frac{\left|\overline{R}_i - \overline{R}_j\right|}{255} \quad d_G = \frac{\left|\overline{G}_i - \overline{G}_j\right|}{255} \quad d_B = \frac{\left|\overline{B}_i - \overline{B}_j\right|}{255}$$

$$d_{ij\,color} = \overline{(d_R, d_G, d_B)}$$

$$d_{ij\,tamaño} = 1 - RelacionTamaño(i,\ j)$$

La Figura 4-8 muestra un esquema de las etapas del algoritmo de *tracking*. Se parte de la elección manual por parte del usuario en el primer *frame* de la región a seguir R_k y la introducción en el programa de su etiqueta k (ver Figura 4-8(a)). Una vez identificada la región en el *frame* actual n, los pasos que permiten encontrar dicha región en el *frame* siguiente n+1 se describen a continuación:

1) Para cada región del *frame* n+1, se evalúan las distancias de color, tamaño y posición con la región a seguir R_k del *frame* n. Si los valores obtenidos son menores que el umbral α establecido en cada caso, la región R_j del frame n+1 se considera "candidata" al *matching* con la región a seguir, y por tanto pertenece al conjunto C (ver Figura 4-8(b)).

$$\left(D_{tam}(R_k, R_j) \leq \alpha_{tam}\right) \cup \left(D_{pos}(R_k, R_j) \leq \alpha_{pos}\right) \cup \left(D_{color}(R_k, R_j) \leq \alpha_{color}\right) \Leftrightarrow R_j \in C$$

2) Se elige entre todas las regiones del conjunto C a la región que obtiene la mínima distancia de posición con la región a seguir (ver Figura 4-8(c)).

$$min(D_{pos}(R_k, R_j)), \forall R_j \in C \Rightarrow R_j = R_k$$

3) El *frame* n+1 marca y guarda la región obtenida que pasa a compararse con las regiones del *frame* n+2, repitiéndose el proceso anterior.

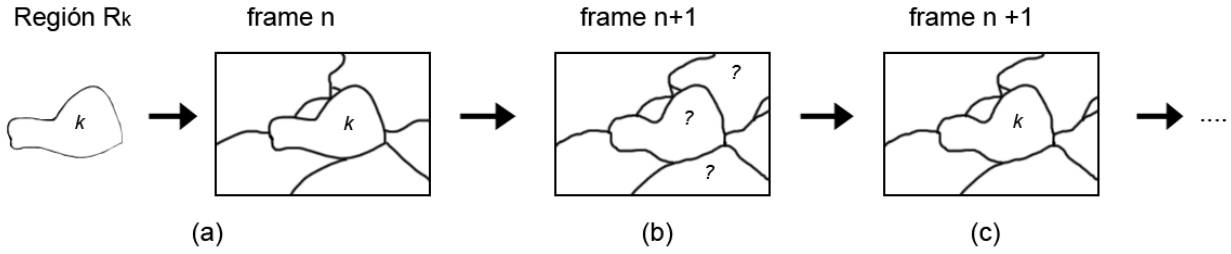

Figura 4-8: Etapas del algoritmo de *tracking*

4.4 Pruebas y resultados obtenidos

En esta sección presentaremos los resultados obtenidos al aplicar el algoritmo de seguimiento de objetos a varias secuencias de vídeo representativas con el fin de demostrar el uso de un sistema basado en descriptores sencillos en el seguimiento de objetos. Asimismo, se mostrarán los problemas que han surgido debido a las limitaciones que presenta esta técnica.

Para la realización de las pruebas, hemos contado con un conjunto de vídeos procedentes del ground-truth "*A Ground Truth for Motion-Based Video-Object Segmentation*" [62] y que resultan representativos ya que introducen distintos niveles de dificultad en el seguimiento de las regiones. Sobre cada uno de los *frames* se ha efectuado una segmentación en 100 regiones.

La primera secuencia, "tenis", muestra una escena de una persona jugando con una pelota. Se han escogido los primeros 100 *frames* de la secuencia sobre los que se trata de seguir dos regiones de interés, la cara y la pierna derecha. Un conjunto de *frames* resultado representativos se pueden ver en la Figura 4-9.

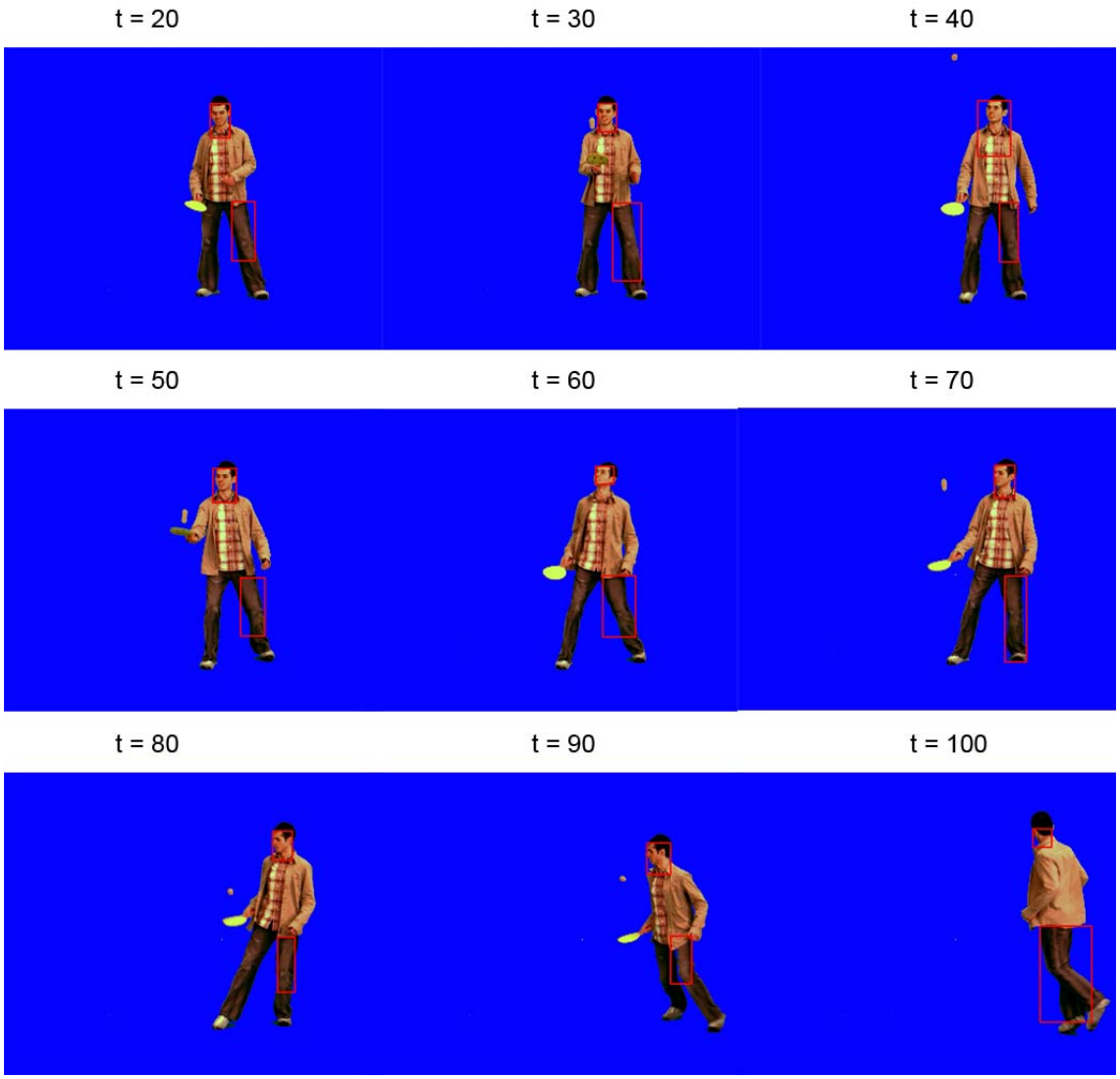

Figura 4-9: *Frames* resultado del seguimiento en la secuencia "tenis".

A pesar de las dificultades que presenta esta secuencia (la chaqueta posee un color muy similar al de la piel, existen oclusiones, giros, …) el sistema consigue detectar las regiones a seguir.

La segunda secuencia denominada “ingravidez” muestra de nuevo la interacción de una persona con un objeto rígido. En este caso se han escogido 200 *frames* desde el inicio de la secuencia, ya que resultaba interesante comprobar el seguimiento de la pelota tanto cuando no hay nada más en la escena (aproximadamente los primeros 80 cuadros) como cuando aparece la persona. La Figura 4-10 muestra una selección de los *frames* obtenidos para este ejemplo sobre un fondo homogéneo.

Por otro lado, se ha considerado probar el sistema sobre el mismo ejemplo pero con un fondo más complejo, es decir, un fondo multimodal con posibles cambios de iluminación, movimiento, etc. Los resultados obtenidos se pueden ver en la Figura 4-11.

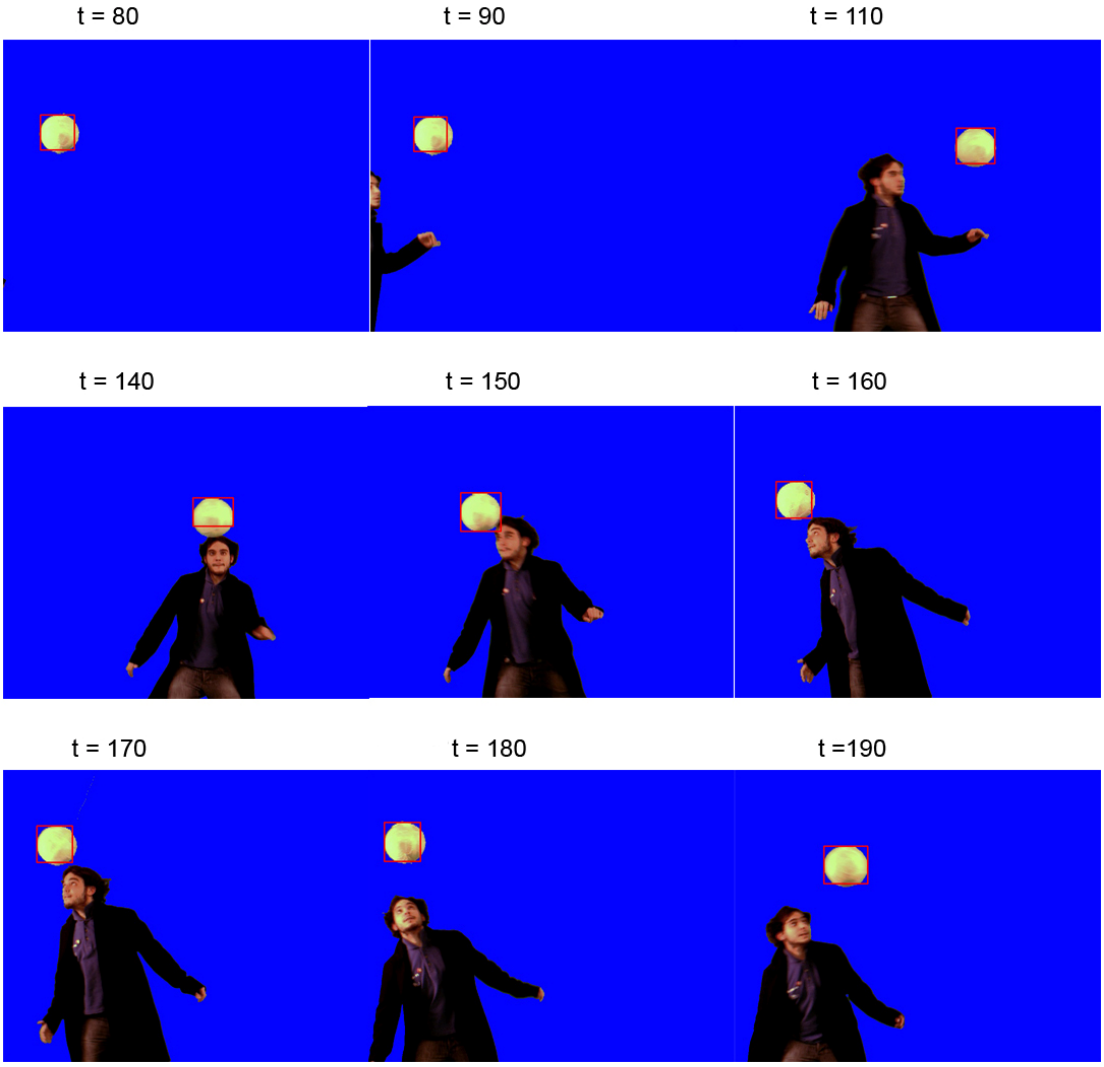

Figura 4-10: Frames resultado del tracking en la secuencia “ingravidez” con fondo homogéneo.

Figura 4-11: Frames resultado del tracking en la secuencia "ingravidez" con fondo multimodal.

- **Problemas encontrados**

- Existe una evidente dependencia de la segmentación, esto hace que en ocasiones si existen variaciones en la segmentación de un determinado objeto en *frames* consecutivos, haya dificultades para encontrar la región, y ésta puede perderse. Un ejemplo de esta circunstancia se da entre los cuadros 102 y 103 de la secuencia "ingravidez". En el cuadro 102 la segmentación distingue la región principal que forma la pelota (ver Figura 4-12(a)), en cambio, en el siguiente cuadro, a pesar de no haber ningún cambio significativo en la escena, dicha región "desaparece", se funde con otra región próxima.

El resultado es que el sistema no encuentra la región y toma como región a seguir la que encuentra más parecida (ver Figura 4-12 (b)). Afortunadamente en este ejemplo, la región vuelve a ser localizada dos frames más tarde.

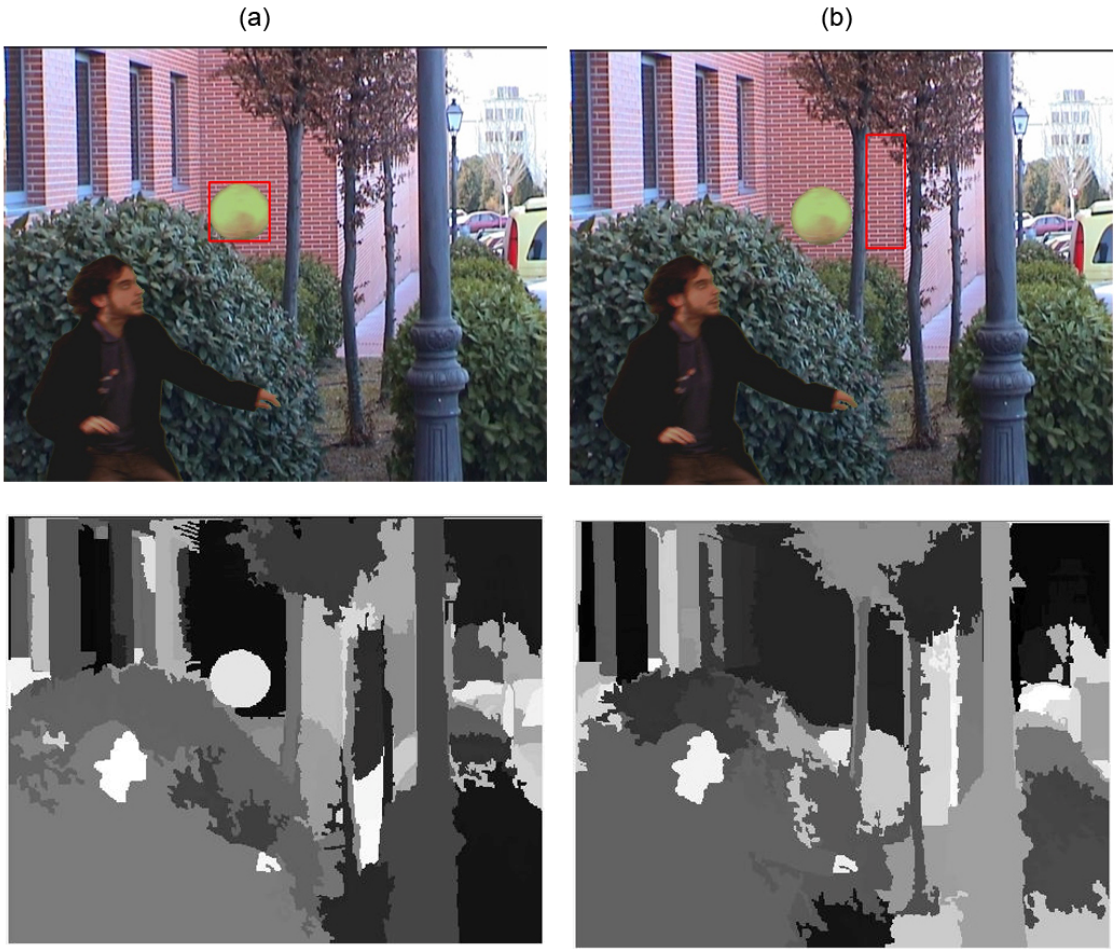

Figura 4-12: Problema derivado de la segmentación. (a) Frame 102 y su segmentación (b) Frame 103 y su segmentación.

- El algoritmo propuesto para el seguimiento de regiones presenta algunas limitaciones, que quizá podrían solucionarse añadiendo un sistema de seguimiento de mayor complejidad (estimación de trayectorias, seguimiento en ventanas temporales, etc.). Uno de los problemas más claros surge del hecho de que se realice la búsqueda a partir de únicamente el *frame* precedente. Si se da el caso en que el sistema no elija la región óptima (como ocurre en el caso anterior), el error se propaga en cuadros sucesivos sin que exista una posibilidad de recuperación (ver Figura 4-13). Para los ejemplos presentados se ha conseguido solucionar este problema ajustando el valor de los umbrales de los descriptores.

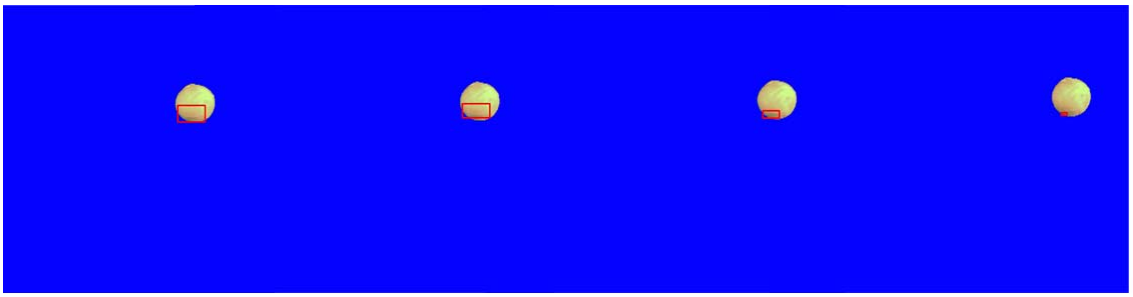

Figura 4-13: Problema derivado del algoritmo de seguimiento. Resultados anteriores al ajuste de umbrales de los descriptores.

5 Conclusiones y Trabajo Futuro

5.1 Conclusiones

En esta memoria se ha detallado el diseño e implementación de un sistema para la extracción y caracterización de forma eficiente de las regiones de una imagen (previamente obtenidas a partir de un algoritmo de segmentación genérico). Para ello se ha desarrollado un conjunto de descriptores sencillos con el fin de obtener una caracterización de las regiones eficiente desde un punto de vista del coste computacional y con aplicación a diversas situaciones. Partiendo de [1] en el que se estudiaba la aplicación de la caracterización de regiones a la adaptación de contenido multimedia se han desarrollado en este proyecto dos aplicaciones adicionales y de interés en el marco de la gestión del contenido multimedia que demuestran la utilidad de este tipo de caracterización: recuperación de imágenes basada en contenido y seguimiento de objetos en una secuencia de vídeo.

Los principales objetivos del proyecto han quedado suficientemente cubiertos tras haber completado las siguientes etapas:

i. Estudio exhaustivo del estado del arte en el área de los descriptores visuales.
ii. Diseño e implementación de un sistema para la extracción y almacenamiento de las regiones y sus descriptores asociados. Estudio de los descriptores que componen el módulo.
iii. Estudio exhaustivo del estado del arte en el área de la recuperación de imágenes por contenido, destacando las propuestas más atractivas.
iv. Diseño e implementación de un algoritmo que permite, una vez integrado en el módulo de extracción y caracterización de regiones, realizar la recuperación de imágenes similares a una consulta dada basándose en el contenido de las mismas.
v. Análisis y verificación de los resultados obtenidos en la aplicación de *retrieval*, mediante una comparativa con un sistema de referencia.
vi. Estudio exhaustivo de las técnicas vigentes para seguimiento de objetos, así como la realización de una síntesis de las mismas que permita entender nítidamente las los retos y dificultades que presenta este área.

vii. Diseño y implementación de un algoritmo que permite, una vez integrado en el módulo de extracción y caracterización de regiones, realizar el seguimiento de la trayectoria de los objetos marcados por el usuario en una secuencia de vídeo.

viii. Análisis y verificación de los resultados obtenidos en la aplicación de *tracking* de objetos, destacando algunas de las limitaciones encontradas.

5.2 Trabajo futuro

En el desarrollo de este proyecto, se han identificado algunos puntos de interés a ser estudiados en un futuro. En esta sección se describen algunas propuestas de mejora y líneas de investigación a seguir ordenadas según su aparición en la memoria.

- Para la segmentación de imágenes se ha utilizado una técnica relativamente sencilla ya que se trataba de obtener resultados independientes de la segmentación. A medida que avanzamos en el desarrollo del proyecto han surgido algunas limitaciones derivadas de una segmentación inadecuada. Una posibilidad de mejora a investigar sería el empleo de técnicas de segmentación más sofisticadas que permitieran obtener un número de regiones variable de cada imagen según el contenido de las mismas.
- En el área de los descriptores visuales, podrían explorarse algunos de los descriptores desarrollados menos utilizados, como la concentración o la relación de contacto, en usos específicos.
- Dentro del ámbito de la recuperación de imágenes, consideramos viable la investigación de una aplicación para detectar cambios de toma en un vídeo. Esta aplicación podría realizarse mediante la observación de cambios bruscos en el parecido de las regiones de un *frame* al siguiente.
- El sistema de seguimiento implementado ha demostrado la aplicación de un sistema de extracción y caracterización de regiones al seguimiento de objetos. Creemos que sería útil continuar esta línea de investigación, añadiendo más complejidad al algoritmo de forma que se puedan obtener resultados satisfactorios en entornos no controlados.
- Asimismo, se ofrecen otras posibilidades de aplicación a vídeo por explorar, como el estudio de la actividad de movimiento en una secuencia a través del seguimiento de varias regiones.

- Por último, sería deseable el empleo de un ground-truth tanto de imágenes como de vídeo más grande y heterogéneo.

Referencias

[1] V. Valdés, "*Region Of Interest Based Content Adaptation*", Trabajo de Iniciación a la Investigación, Escuela Politécnica Superior, Universidad Autónoma de Madrid, 2006.

[2] A. Skrodas, Ch. Christopolous, T. Ebrahimi, *"The JPEG2000 Still Image Compression Standard"*, IEEE Signal Processing Magazine, 18(5):36-58, September 2001.

[3] J.R. Ohm (ed.): *"Introduction to SVC Extension of Advanced Video Coding"* ISO/MPEG doc. N7315, 2005.

[4] José M. Martínez, *"MPEG-7 Overview"*, in ISO/IEC JTC1/SC29/WG11N6828, October 2004. http://www.chiariglione.org/mpeg/standards/mpeg-7/mpeg-7.htm.

[5] ISO/IEC 15938-3:2001, *"Multimedia Content Description Interface - Part 3: Visual"*, Version 1.

[6] Zhang, Lu, *"Review of shape representation and description techniques"*, in Pattern Recognition Journal, 37: 1-19, 2004.

[7] R.C. Veltkamp, *"Shape Matching: Similarity Measures and Algorithms"*, International Conference on Shape Modeling & Applications, pp.188-197, 2001.

[8] T. Lin and Y. Chou, *"A Comparative Study of Zernike Moments"*, IEEE/WIC International Conference on Web Intelligence (WI'03). pp. 516-519, 2003.

[9] D. Zhang and G. Lu, *"A Comparative Study of Fourier Descriptors for Shape Representation and Retrieval"*, in ACCV2002: The 5th Asian Conference on Computer Vision, pp. 23- 25, January 2002.

[10] S. Abbasi, F. Mokhtarian, and J. Kittler, *"Curvature Scale Space Image in Shape Similarity Retrieval"*, Multimedia Systems, vol. 7, pp. 467- 476, 1997.

[11] J. Wang, G. Wiederhold, O. Firschein, S.Wei, *"Wavelet-Based Image Indexing Techniques with Partial Sketch Retrieval Capability"*, in Proceedings of the Fourth Forum on Research and Technology Advances in Digital Libraries, pp. 13-24, 1997.

[12] J. Strckrott, *"A Survey of Image Segmentation Techniques for Content Based Retrieval of Multimedia Data"*, FIU Department of Computer Science, 2001.

[13] S. Cooray, N. O'Connor, S. Marlow, N. Murphy and T. Curran, *"Semi-automatic Video Object Segmentation using Recursive Shortest Spanning Tree and Binary Partition Tree"* in WIAMIS 2001: Workshop on Image Analysis for Multimedia Interactive Services, Finland 2001.

[14] E. Tuncel and L. Onural, *"Utilization of the recursive shortest spanning tree algorithm for video-object segmentation by 2-D affine motion modelling"*, IEEE Transactions on Circuits and Systems for Video Technology, 10(5): 776-781, Aug 2000.

[15] J. P. Eakins and M. E. Graham, *"Content-based image retrieval"*, JISC technology applications programme, report 39, January 1999.

[16] J. Han and M. Kuang, *"Fuzzy Color Histogram and Its Use in Color Image Retrieval"* IEEE Trans. Image Process., vol. 11, Nº. 8, pp. 944-952, August 2002.

[17] C. Carson, S. Belongie, H. Greenspan, J. Malik, *"Blobworld: Image Segmentation Using Expectation-Maximization and Its Application to Image Querying"* IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 24, Nº. 8, pp. 1026-1038, Aug 2002.

[18] C. E. Jacobs, A. Finkelstein, D. H. Salesin., *"Fast Multiresolution Image Querying"*. Proceedings of SIGGAPH 95, *Computer Graphics* Proceedings, Annual Conference Series, pp. 277-286, August 1995.

[19] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, D. Qian Huang, B. Gorkani, M. Hafner, J. Lee, D. Petkovic, D. Steele, D. Yanker, P. IBM Almaden Res. Center, San Jose, CA; *"Query by image and video content: the QBIC system"*, Computer, Vol 28, pp. 23-32, Sep 1995.

[20] A. Pentland, R. W. Picard, and S. Sclaroff, *"Photobook: Tools for Content-Based Manipulation of Image Databases"*, Proc. SPIE, vol. 2185, Storage and Retrieval for Image and Video Databases, pp. 34-47, Feb. 1994.

[21] J. Z. Wang and J. Li, *"SIMPLIcity: Semantics-Sensitive Integrated Matching for Picture Libraries"*, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol 23, Nº 9, pp. 947-963, September 2001.

[22] M. Das, E. M. Riseman, B. A. Draper, *"FOCUS: Searching for multi-colored objects in a diverse image database"*, Computer Vision and Pattern Recognition, Vol , 17-19, pp. 756 – 761, Jun 1997.

[23] N. Suematsu, Y. Ishida, A. Hayashi, T. Kanbara, *"Region-based image retrieval using wavelet transform"*, Proc. 15th Internat. Conf. on Vision Interface, pp. 9-16, Canada May 2002.

[24] Á. F. Zazo, C. G. Figuerola, J. L. Berrocal, R. Gómez, *"Recuperación de información utilizando el modelo vectorial"*. CLEF-2001. Technical Report DPTOIA-IT-2002-006, Departamento de Informática y Automática, Universidad de Salamanca. Available: http://tejo.usal.es/inftec/2002/DPTOIA-IT-2002-006.pdf.

[25] Sangoh Jeong, *"Histogram-Based Color Image Retrieval"*, Psych221/EE362 Project Report, March 2001. http://scien.standford.edu/class/psych221/projects/02/sojeong/.

[26] A. Yilmaz , O. Javed , M. Shah, *"Object tracking: A survey"*, ACM Computing Surveys (CSUR), v.38 n.4, p.13-es, 2006.

[27] C. Veenman, M. Reinders, and E. Backer, *"Resolving motion correspondence for densely moving points"*. IEEE Trans. Patt. Analy. Mach. Intell. vol. 23, nº 1, pp.54-72.

[28] D. Serby, S. Koller-Meier, and L. V. Gool, *"Probabilistic object tracking using multiple features"*, IEEE International Conference of Pattern Recognition (ICPR), pp. 184-187. 2004.

[29] D. Comaniciu, V. Ramesh, and P. Meer, *"Kernel-based object tracking"*. IEEE Trans. Patt. Analy. Mach. Intell. 25, pp. 564-575. 2003.

[30] A. Yilmaz, X. Li and M. Shah, *"Contour based object tracking with occlusion handling in video acquired using mobile cameras"*. IEEE Trans. Patt. Analy. Mach. Intell. 26, 11, 1531–1536. 2004.

[31] D. Ballard, and C. Brown, *"Computer Vision"*, Prentice-Hall. 1982.

[32] A. Ali and J. Aggarwal, *"Segmentation and recognition of continuous human activity"*, IEEE Workshop on Detection and Recognition of Events in Video. 28–35. 2001.

[33] P. Fieguth and D. Terzopoulos, *"Color-based tracking of heads and other mobile objects at video frame rates"*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 21–27. 1997.

[34] H. Moravec, *"Visual mapping by a robot rover"*, Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI). 598–600. 1979.

[35] C. Harris and M. Stephens, *"A combined corner and edge detector"*, 4th Alvey Vision Conference. 147–151. 1988.

[36] D. Lowe, *"Distinctive image features from scale-invariant keypoints"*, Int. J. Comput. Vision 60, 2, 91–110. 2004.

[37] K. Mikolajczyk and C. Schmid, *"An affine invariant interest point detector"*, European Conference on Computer Vision (ECCV). Vol. 1. 128–142. 2002.
[38] D. Comaniciu and P. Meer, *"Mean shift analysis and applications"*, IEEE International Conference on Computer Vision (ICCV). Vol. 2. 1197–1203.
[39] J. Shi and J. Malik, *"Normalized cuts and image segmentation"*, IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 888–905. 2000.
[40] V. Caselles, R. Kimmel and G. Sapiro, *"Geodesic active contours"*, IEEE International Conference on Computer Vision (ICCV). 694–699. 1995.
[41] C. Stauffer and W. Grimson, *"Learning patterns of activity using real time tracking"*. IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 747–767. 2000.
[42] N. Oliver, B. Rosario and A. Pentland, *"A bayesian computer vision system for modeling human interactions"*, IEEE Trans. Patt. Analy. Mach. Intell. 22, 8, 831–843. 2000.
[43] K. Toyama, B. J. Krumm and B. Meyers, *"Wallflower: Principles and practices of background maintenance"*, IEEE International Conference on Computer Vision (ICCV). 255–261. 1999.
[44] A. Monnet, A. Mittal, N. Paragios and V. Ramesh, *"Background modeling and subtraction of dynamic scenes"*, IEEE International Conference on Computer Vision (ICCV). 1305–1312. 2003.
[45] C. Papageorgiou, M. Oren and T. Poggio, *"A general framework for object detection"*, IEEE Internacional Conference on Computer Vision (ICCV). 555–562. 1998.
[46] H. Rowley, S. Baluja and T. Kanade, *"Neural network-based face detection"*, IEEE Trans. Patt. Analy. Mach. Intell. 20, 1, 23–38. 1998.
[47] P. Viola, M. Jones and D. Snow, *"Detecting pedestrians using patterns of motion and appearance"*, IEEE International Conference on Computer Vision (ICCV). 734–741. 2003.
[48] J. Shi and C. Tomasi, *"Good features to track"*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 593–600. 1994.
[49] V. Salari, and I. K. Sethi, *"Feature point correspondence in the presence of occlusion"*. IEEE Trans. Patt. Analy. Mach. Intell. 12, 1, 87–91. 1990.
[50] T. Broida and R. Chellappa, *"Estimation of object motion parameters from noisy images"*. IEEE Trans. Patt. Analy. Mach. Intell. 8, 1, 90–99. 1986.
[51] Y. Bar-Shalom and T. Foreman, "*Tracking and Data Association"*, Academic Press Inc. 1988.
[52] R. L.Streit and T. E. Luginbuhl, *"Maximum likelihood method for probabilistic multi-hypothesis tracking"*, Proceedings of the International Society for Optical Engineering (SPIE.) vol. 2235. 394–405. 1994.
[53] H. Tao, H. Sawhney and R. Kumar, *"Object tracking with bayesian estimation of dynamic layer representations"*. IEEE Trans. Patt. Analy. Mach. Intell. 24, 1, 75–89. 2002.
[54] M. Black and A. Jepson, *"Eigentracking: Robust matching and tracking of articulated objects using a view-based representation"*. Int. J. Comput. Vision 26, 1, 63–84. 1998.
[55] S. Avidan, *"Support vector tracking"*, IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 184–191. 2001.
[56] M. Isard and A. Blake. *"Condensation - conditional density propagation for visual tracking"*. Int. J. Comput. Vision 29, 1, 5–28. 1998.

[57] M. Bertalmio, G. Sapiro and G. Randall, *"Morphing active contours"*, IEEE Trans. Patt. Analy. Mach. Intell. 22, 7, 733–737. 2000.
[58] R. Ronfard, *"Region based strategies for active contour models"*, Int. J. Comput. Vision 13, 2, 229–251. 1994.
[59] F. Hausdorff, *"Set Theory"*, Chelsea Publishing Company. 1962.
[60] K. Sato and J. Aggarwal, *"Temporal spatio-velocity transform and its application to tracking and interaction"*, Comput. Vision Image Understand. 96, 2, 100–128. 2004.
[61] J. Kang, I. Cohen and G. Medioni, *"Object reacquisition using geometric invariant appearance model"*, International Conference on Pattern Recongnition (ICPR). 759–762. 2004.
[62] F. Tiburzi, M. Escudero, J. Bescós, J.M. Martinez, *"A Ground Truth for Motion-Based Video-Object Segmentation"*, ICIP'08 Workshop on Multimedia Information Retrieval, Proc ICIP'2008, accepted for presentation.

Glosario

ROI	Region Of Interest
JPEG	Joint Photographic Experts Group
MPEG	Moving Pictures Expert Group
CSS	Curvature Scale Space
FD	Fourier Descriptor
DFT	Discrete Fourier Transform
RSST	Recursive Shortest Spanning Tree
CBIR	Content-Based Image Retrieval
QBIC	Query By Image Content
IEEE	Institute of Electrical and Electronics Engineers

Anexos

A Estructuras de datos creadas

Con el fin de agrupar la información que se utiliza en el programa de una forma estructurada que facilite la tarea de extracción de los descriptores, se han creado varias estructuras de datos cuyos métodos han sido descritos en el apartado 2.4.2 de esta memoria. A continuación, presentamos la definición de dichas estructuras junto con una breve especificación de sus atributos.

A.1 Clase Imagen

Esta estructura almacena la información referente a cada una de las imágenes o cuadros de vídeo analizados.

```
class IMAGEN
{

        // ATRIBUTOS PÚBLICOS
        public:
                int height,width,step,channels;
                IplImage* img;
                IplImage* img_ori;
                char *nombre_img;
                char *nombre_img_ori;
                Region vRegiones[N];
                Region vRegFiltradas[N];
                int numeroRegiones;
                int numeroRegionesFiltradas;

        //METODOS PÚBLICOS
        public:

                IMAGEN();
                void CargarImagen(char *original, char *segm);
                void CargaRegiones();
                void ExtraerDescriptores();
                void FiltroTamagno(int minimo);
                void LiberarFrame();
};
```

height, width, step, channels son los atributos de la imagen original y coinciden con los de la estructura IplImage de la librería OpenCv (ver apartado 2.4.1).

img referencia a la imagen segmentada en formato PGN.

img_ori direcciona a la imagen original en formato JPEG.

nombre_img y ***nombre _img_ori*** son punteros a las cadenas que contienen los nombres de las imágenes *img* e *img_ori*.

vRegiones[N] es un *array* de dimensión máxima N = 255 que almacena las regiones que componen la imagen.

vRegionesFiltradas[N] es un *array* de dimensión máxima N = 255 que almacena las regiones tras el filtrado.

numeroRegiones y ***numeroRegionesFiltradas*** contienen el número de regiones respectivas de los *arrays* anteriores.

A.2 Clase Región

```
class Region
{
	// ATRIBUTOS PÚBLICOS
	public:

		int id_number;
		std::vector<std::vector<pixel> > vec
		int xmin, xmax, ymin, ymax;
		int x_total, y_total;
		std::vector<pixel> borde;
		unsigned char **mascaraRegion;
		bool flagMascaraRegion;

		//DESCRIPTORES INTRA

		int NumeroDePixeles ;
		double ancho;
		double alto;
		double area;
		double relacionAspecto;
		double perimetro;
		double densidad;
		double compacidad;
		pixel centro_geom;
		pixel centro_masas;
		CvPoint2D32f relacionCentros;
		double mediaR, mediaG, mediaB;
		double varR, varG, varB;

	//METODOS PÚBLICOS
	public:

		Region();
		void inicializarVec();
		bool RegionExiste();
		int GetIdNumber();
		void SetIdNumber(int num);
		void anyadirPixel(pixel p);
		pixel getPixel(int i, int j);
		bool estaEnROI(pixel punto);
		void calculateRetrievalDescriptors();
		void NumberOfPixels();
		void WidthOfBoundingBox();
		void HeightOfBoundingBox();
		void AreaOfBoundingBox();
		void AspectRatio();
		void Density();
		void Perimeter();
		void Compactness();
		void GeometricCentre();
		void MassCentre();
		void CentresRatio();
```

```
                void MeanRGB();
                void RGBVariance();
                void calculaBorde();
                int calculaVecinos(pixel p);
                void crearMascaraRegion();
                void imprimirMascaraRegion();

};
```

id_number es un identificador numérico diferente para cada región que es asignado de forma automática por el segmentador.

vec es una matriz de dos dimensiones que contiene los píxeles de la región.

xmin, xmax, ymin, ymax son las componentes de dos coordenadas opuestas de la *Bounding Box* que contiene la región.

x_total, y_total acumulan el valor de las componentes x e y de los píxeles de la región, lo que permite hallar el centro de masas de la misma.

Borde es el vector que contiene los píxeles que pertenecen al perímetro de la región.

mascaraRegion es una matriz auxiliar para hallar el número de vecinos de cada píxel de la región.

flagMascaraRegion es un *flag* cuyo valor es 'true' si ya se ha creado *mascaraRegión*.

A.3 Estructura Píxel

```
struct pixel
{
        int x, y;
        int pixel_id;
        int red, green, blue;
        bool esBorde;
};
```

x, y son las coordenadas del píxel.

pixel_id determina a qué región pertenece un determinado píxel, asignándole la misma etiqueta de la región.

red, green, blue son los valores de las componentes RGB del píxel.

esBorde tiene valor 'true' si el píxel pertenece al borde de al menos una región.

B Diseño del Ground-Truth de imágenes

En este anexo se describe el ground-truth utilizado en las pruebas del sistema de recuperación de imágenes descrito en el capítulo 3. Las imágenes pertenecen a una colección proporcionada por el grupo MPEG y tienen un formato JPG. Se han seleccionado 60 imágenes estableciéndose tres categorías (con 20 imágenes cada una) de acuerdo al contenido de las mismas. Todas las imágenes se han redimensionado a un valor de 200x133 píxeles con el fin de obtener una mayor velocidad en el procesado de extracción de las regiones. A continuación presentamos las imágenes que componen el ground-truth, clasificadas según la categoría a la que pertenecen; así como una tabla donde se realiza una breve descripción de las mismas.

- **Categoría "gente"**

- **Categoría “rocas”**

- **Categoría “aérea”**

Categoría	*Descripción*
'Gente'	Conjunto de imágenes con la característica común de que aparecen personas en ellas. Se ha escogido este conjunto porque la aparición de caras humanas es menos evidente que en otras imágenes con personas en primer plano frontal. Predominan los tonos de la piel, los grises-azulados y algunos elementos rojos. Al segmentar las imágenes de esta clase se obtiene cierta homogeneidad en el tamaño de la mayoría de las regiones.
'Rocas'	Conjunto de imágenes con la característica común de que aparecen rocas en un espacio abierto y árido. Se ha escogido este conjunto por que a pesar de la uniformidad de color, las imágenes contienen elementos diversos (las ruinas, la casa de madera,…) lo que dificulta la recuperación. Predominan los tonos marrones y rojizos junto con el color azul presente en el cielo. Al segmentar las imágenes de esta clase predominan las regiones grandes o muy pequeñas.
'Aérea'	Conjunto de imágenes con la nexo común de estar tomadas desde el aire hacia un paisaje similar. Se ha escogido este conjunto porque existe una evidente similitud en las imágenes, sin embargo, algunas de las imágenes están más afectadas por el ruido, los cambios de iluminación o la sobreexposición. Regiones de tamaño y forma desigual.

C Pruebas efectuadas para la obtención del umbral β

Este anexo recoge las pruebas más representativas efectuadas sobre el algoritmo basado en la matriz de distancias para decidir el valor del umbral de decisión por debajo del cual una región se considera parecida a otra, lo que determina el parecido global entre imágenes.

Valor de β	Descripción	Imágenes recuperadas
0.05	Umbral demasiado alto. Todas las imágenes en la comparativa obtienen entre un 95 y un 100 por cien de parecido con la imagen de consulta, valores que no reflejan la realidad.	
0.01	Umbral demasiado estricto. Las imágenes obtienen una tasa de parecido comprendida entre un 0 y un 20 por cien, lo cual se aleja del parecido real.	

0.02	Valor de umbral adecuado aunque todavía demasiado estricto. Las imágenes obtienen entre un 0 y un 60 por cien de parecido con la consulta.	
0.03	Umbral escogido para evaluar el parecido entre las imágenes. Las tasas de parecido están entre un 35 y un 100 por cien, lo que supone un amplio rango de valores posibles que reflejan adecuadamente el parecido real.	

PRESUPUESTO

1) Ejecución Material

- Compra de ordenador personal (Software incluido)..... ..... 2.000 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 2.150 €

2) Gastos generales

- 16 % sobre Ejecución Material ..... 344 €

3) Beneficio Industrial

- 6 % sobre Ejecución Material ..... 129 €

4) Honorarios Proyecto

- 900 horas a 15 € / hora ..... 13500 €

5) Material fungible

- Gastos de impresión ..... 60 €
- Encuadernación ..... 40 €

6) Subtotal del presupuesto

- Subtotal Presupuesto ..... 15750€

7) I.V.A. aplicable

- 16% Subtotal Presupuesto ..... 2520 €

8) Total presupuesto

- Total Presupuesto ..... 18270 €

Madrid, Julio de 2008

El Ingeniero Jefe de Proyecto

Fdo.: Helena González Casero
Ingeniero Superior de Telecomunicación

PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de un "Sistema de Extracción y Gestión de Regiones de Interés en Contenido Audiovisual". En lo que sigue, se supondrá que el proyecto ha sido encargado por una empresa cliente a una empresa consultora con la finalidad de realizar dicho sistema. Dicha empresa ha debido desarrollar una línea de investigación con objeto de elaborar el proyecto. Esta línea de investigación, junto con el posterior desarrollo de los programas está amparada por las condiciones particulares del siguiente pliego.

Supuesto que la utilización industrial de los métodos recogidos en el presente proyecto ha sido decidida por parte de la empresa cliente o de otras, la obra a realizar se regulará por las siguientes:

Condiciones generales

1. La modalidad de contratación será el concurso. La adjudicación se hará, por tanto, a la proposición más favorable sin atender exclusivamente al valor económico, dependiendo de las mayores garantías ofrecidas. La empresa que somete el proyecto a concurso se reserva el derecho a declararlo desierto.

2. El montaje y mecanización completa de los equipos que intervengan será realizado totalmente por la empresa licitadora.

3. En la oferta, se hará constar el precio total por el que se compromete a realizar la obra y el tanto por ciento de baja que supone este precio en relación con un importe límite si este se hubiera fijado.

4. La obra se realizará bajo la dirección técnica de un Ingeniero Superior de Telecomunicación, auxiliado por el número de Ingenieros Técnicos y Programadores que se estime preciso para el desarrollo de la misma.

5. Aparte del Ingeniero Director, el contratista tendrá derecho a contratar al resto del personal, pudiendo ceder esta prerrogativa a favor del Ingeniero Director, quien no estará obligado a aceptarla.

6. El contratista tiene derecho a sacar copias a su costa de los planos, pliego de condiciones y presupuestos. El Ingeniero autor del proyecto autorizará con su firma las copias solicitadas por el contratista después de confrontarlas.

7. Se abonará al contratista la obra que realmente ejecute con sujeción al proyecto que sirvió de base para la contratación, a las modificaciones autorizadas por la superioridad o a las órdenes que con arreglo a sus facultades le hayan comunicado por escrito al Ingeniero Director de obras siempre que dicha obra se haya ajustado a los preceptos de los pliegos de condiciones, con arreglo a los cuales, se harán las modificaciones y la valoración de las diversas unidades sin que el importe total pueda exceder de los presupuestos aprobados. Por consiguiente, el número de unidades que se consignan en el proyecto o en el presupuesto, no podrá servirle de fundamento para entablar reclamaciones de ninguna clase, salvo en los casos de rescisión.

8. Tanto en las certificaciones de obras como en la liquidación final, se abonarán los trabajos realizados por el contratista a los precios de ejecución material que figuran en el presupuesto para cada unidad de la obra.

9. Si excepcionalmente se hubiera ejecutado algún trabajo que no se ajustase a las condiciones de la contrata pero que sin embargo es admisible a juicio del Ingeniero Director de obras, se dará conocimiento a la Dirección, proponiendo a la vez la rebaja de precios que el Ingeniero estime justa y si la Dirección resolviera aceptar la obra, quedará el contratista obligado a conformarse con la rebaja acordada.

10. Cuando se juzgue necesario emplear materiales o ejecutar obras que no figuren en el presupuesto de la contrata, se evaluará su importe a los precios asignados a otras obras o materiales análogos si los hubiere y cuando no, se discutirán entre el Ingeniero Director y el contratista, sometiéndolos a la aprobación de la Dirección. Los nuevos precios convenidos por uno u otro procedimiento, se sujetarán siempre al establecido en el punto anterior.

11. Cuando el contratista, con autorización del Ingeniero Director de obras, emplee materiales de calidad más elevada o de mayores dimensiones de lo estipulado en el proyecto, o sustituya una clase de fabricación por otra que tenga asignado mayor precio o ejecute con mayores dimensiones cualquier otra parte de las obras, o en general, introduzca en ellas cualquier modificación que sea beneficiosa a juicio del Ingeniero Director de obras, no tendrá derecho sin embargo, sino a lo que le correspondería si hubiera realizado la obra con estricta sujeción a lo proyectado y contratado.

12. Las cantidades calculadas para obras accesorias, aunque figuren por partida alzada en el presupuesto final (general), no serán abonadas sino a los precios de la contrata, según las condiciones de la misma y los proyectos particulares que para ellas se formen, o en su defecto, por lo que resulte de su medición final.

13. El contratista queda obligado a abonar al Ingeniero autor del proyecto y director de obras así como a los Ingenieros Técnicos, el importe de sus respectivos honorarios facultativos por formación del proyecto, dirección técnica y administración en su caso, con arreglo a las tarifas y honorarios vigentes.

14. Concluida la ejecución de la obra, será reconocida por el Ingeniero Director que a tal efecto designe la empresa.

15. La garantía definitiva será del 4% del presupuesto y la provisional del 2%.

16. La forma de pago será por certificaciones mensuales de la obra ejecutada, de acuerdo con los precios del presupuesto, deducida la baja si la hubiera.

17. La fecha de comienzo de las obras será a partir de los 15 días naturales del replanteo oficial de las mismas y la definitiva, al año de haber ejecutado la provisional, procediéndose si no existe reclamación alguna, a la reclamación de la fianza.

18. Si el contratista al efectuar el replanteo, observase algún error en el proyecto, deberá comunicarlo en el plazo de quince días al Ingeniero Director de obras, pues transcurrido ese plazo será responsable de la exactitud del proyecto.

19. El contratista está obligado a designar una persona responsable que se entenderá con el Ingeniero Director de obras, o con el delegado que éste designe, para todo relacionado con ella. Al ser el Ingeniero Director de obras el que interpreta el proyecto, el contratista deberá consultarle cualquier duda que surja en su realización.

20. Durante la realización de la obra, se girarán visitas de inspección por personal facultativo de la empresa cliente, para hacer las comprobaciones que se crean oportunas. Es obligación del contratista, la conservación de la obra ya ejecutada hasta la recepción de la misma, por lo que el deterioro parcial o total de ella, aunque sea por agentes atmosféricos u otras causas, deberá ser reparado o reconstruido por su cuenta.

21. El contratista, deberá realizar la obra en el plazo mencionado a partir de la fecha del contrato, incurriendo en multa, por retraso de la ejecución siempre que éste no sea debido a causas de fuerza mayor. A la terminación de la obra, se hará una recepción provisional previo reconocimiento y examen por la dirección técnica, el depositario de efectos, el interventor y el jefe de servicio o un representante, estampando su conformidad el contratista.

22. Hecha la recepción provisional, se certificará al contratista el resto de la obra, reservándose la administración el importe de los gastos de conservación de la misma hasta su recepción definitiva y la fianza durante el tiempo señalado como plazo de garantía. La recepción definitiva se hará en las mismas condiciones que la provisional, extendiéndose el acta correspondiente. El Director Técnico propondrá a la Junta Económica la devolución de la fianza al contratista de acuerdo con las condiciones económicas legales establecidas.

23. Las tarifas para la determinación de honorarios, reguladas por orden de la Presidencia del Gobierno el 19 de Octubre de 1961, se aplicarán sobre el denominado en la actualidad "Presupuesto de Ejecución de Contrata" y anteriormente llamado "Presupuesto de Ejecución Material" que hoy designa otro concepto.

Condiciones particulares

La empresa consultora, que ha desarrollado el presente proyecto, lo entregará a la empresa cliente bajo las condiciones generales ya formuladas, debiendo añadirse las siguientes condiciones particulares:

1. La propiedad intelectual de los procesos descritos y analizados en el presente trabajo, pertenece por entero a la empresa consultora representada por el Ingeniero Director del Proyecto.

2. La empresa consultora se reserva el derecho a la utilización total o parcial de los resultados de la investigación realizada para desarrollar el siguiente proyecto, bien para su publicación o bien para su uso en trabajos o proyectos posteriores, para la misma empresa cliente o para otra.

3. Cualquier tipo de reproducción aparte de las reseñadas en las condiciones generales, bien sea para uso particular de la empresa cliente, o para cualquier otra aplicación, contará con autorización expresa y por escrito del Ingeniero Director del Proyecto, que actuará en representación de la empresa consultora.

4. En la autorización se ha de hacer constar la aplicación a que se destinan sus reproducciones así como su cantidad.

5. En todas las reproducciones se indicará su procedencia, explicitando el nombre del proyecto, nombre del Ingeniero Director y de la empresa consultora.

6. Si el proyecto pasa la etapa de desarrollo, cualquier modificación que se realice sobre él, deberá ser notificada al Ingeniero Director del Proyecto y a criterio de éste, la empresa consultora decidirá aceptar o no la modificación propuesta.

7. Si la modificación se acepta, la empresa consultora se hará responsable al mismo nivel que el proyecto inicial del que resulta el añadirla.

8. Si la modificación no es aceptada, por el contrario, la empresa consultora declinará toda responsabilidad que se derive de la aplicación o influencia de la misma.

9. Si la empresa cliente decide desarrollar industrialmente uno o varios productos en los que resulte parcial o totalmente aplicable el estudio de este proyecto, deberá comunicarlo a la empresa consultora.

10. La empresa consultora no se responsabiliza de los efectos laterales que se puedan producir en el momento en que se utilice la herramienta objeto del presente proyecto para la realización de otras aplicaciones.

11. La empresa consultora tendrá prioridad respecto a otras en la elaboración de los proyectos auxiliares que fuese necesario desarrollar para dicha aplicación industrial, siempre que no haga explícita renuncia a este hecho. En este caso, deberá autorizar expresamente los proyectos presentados por otros.

12. El Ingeniero Director del presente proyecto, será el responsable de la dirección de la aplicación industrial siempre que la empresa consultora lo estime oportuno. En caso contrario, la persona designada deberá contar con la autorización del mismo, quien delegará en él las responsabilidades que ostente.