

**UNIVERSIDAD AUTONOMA DE MADRID**

**ESCUELA POLITECNICA SUPERIOR**



**PROYECTO FIN DE CARRERA**

**CONTRIBUCIONES A TÉCNICAS  
DE SEGMENTACIÓN DE  
IMÁGENES**

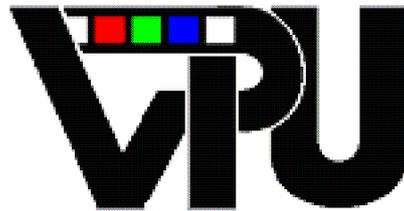
**Elena Ortiz Muñoz**

**SEPTIEMBRE 2009**



# CONTRIBUCIONES A TÉCNICAS DE SEGMENTACIÓN DE IMÁGENES

**AUTOR:** Elena Ortiz Muñoz  
**TUTOR:** José M. Martínez Sánchez



**Video Processing and Understanding Lab**  
**Dpto. de Ingeniería Informática**  
**Escuela Politécnica Superior**  
**Universidad Autónoma de Madrid**  
**Septiembre de 2009**



**Resumen:**

En este proyecto fin de carrera se realiza una amplia documentación del estado del arte de la segmentación de imágenes. Se exponen con detalle y de manera teórica el método KMCC (*K-Means-with-connectivity-constraint*) con características de textura añadidas – basado en características visuales de bajo nivel-; y el M-RSST (Modified-Recursive Shortest Spanning) – basado en características geométricas de los contornos de los objetos-; para exponer a continuación método híbrido que consiste en una fusión de los anteriores. Se presentan también los resultados experimentales que demuestran la viabilidad de desarrollar un algoritmo con múltiples características.

**Palabras clave:**

Segmentación, características sintácticas, características geométricas, características visuales, KMCC, M-RSST, textura, contornos, adyacencia, complejidad.

***Abstract:***

*A wide documentation of the state of art of image segmentation is performed in this master thesis (pfc). It is described in detail the theory method KMCC (K-Means-with-connectivity-constraint) with textural characteristics added - based on low-level visual features; and the M-RSST (Modified-Recursive Shortest Spanning) - based on geometric characteristics of objects' contours; then hybrid approach consisting of a fusion of the above is exposed. We also present experimental results which demonstrate the feasibility of developing an algorithm with multiple features.*

***Keywords:***

*Segmentation, syntactic characteristics, geometric characteristics, visual characteristics, KMCC, M-RSST, texture, contours, adjacency, complexity.*



## **Agradecimientos**

*Tras más de un año de trabajo, son muchas las personas a las que agradecer y difícil no olvidarse de alguien, pero hay que intentarlo.*

*En primer lugar, agradecer a mi tutor, José María Martínez por la confianza depositada en mi. No solo al ofrecirme este proyecto fin de carrera y abrirme las puertas del VPU Lab, sino también a lo largo de la carrera como profesor y como tutor, por partida doble. Gracias por tu paciencia, por la motivación con la que salía tras cada reunión y por tu disponibilidad a pesar de estar siempre hasta arriba.*

*En mi nombre y en el de mi tutor, agradecer sinceramente a Vasilis Mezaris, a Tomasz Adamek y a la Universidad de Dublín (DCU) por las fuentes aportadas, sin las que este proyecto fin de carrera no podría haberse realizado.*

*Dar las gracias en general a todo el profesorado de la EPS que demuestran cada año que no solo saben enseñar, también hacer de nuestra escuela un lugar agradable para estudiar y convivir. En especial, quiero agradecer los consejos, orientación, el tiempo dedicado y por qué no, las risas compartidas a Jesús Bescós y Chema. Espero que con los años mantengáis esas puertas abiertas en vuestros despachos y tener por seguro que más de uno, ¡volveremos de nuevo con “nuestras historias”!*

*¿Y a quién puedo agradecerles más el hecho de haber llegado viva hasta aquí? A los que han compartido estos años conmigo: Gus, gracias por tu humor, tu sensibilidad (aunque me vayas a matar por poner esto!) y por saberme tratar. Chus, porque me encanta que siempre me des tu visión sincera de todo. Pablo, por ser tan diferente y tan igual siempre conmigo. Kiko, porque fuimos la avanzadilla, pero te perdono que me adelantaras! Luqui, porque creo que eras el único que dudaba más que yo de que este día llegara y eso me motivaba ☺ Bárbara, tus palabras siempre alentadoras (venga Ele que no te queda nada... venga Ele que seguro que para el mes que viene ya lo tienes, no?) se agradecen! Sonso, por tus piropos, GUAPA!! Pero sobre todo por hacerme reír cada día. Esther, porque fuiste mi primera compi y por nuestros amaneceres en Dublín que no se olvidan (ánimo, que no queda nada!). Moni, por tu enorme paciencia conmigo y por quererme pase lo que pase (sabes que yo también). Y por último Vero... como no podría resumir ni agradecerte con palabras estos años, solo te diré que estoy segura de que los que vienen serán mejores aun si cabe (FWY)!*

*Fuera del terreno universitario también quería agradecerles a todas las personas que se han interesado y preocupado por la evolución de este proyecto.*

*A las de toda la vida, porque estaban antes, durante y después. La primera, Belen porque con tus esfuerzos creo que eres la persona que más orgullosa se sentirá de mí (después de mis padres!;) ). Gracias por tu ayuda y por ser todo lo tierna y todo lo dura que necesito. A Emma, porque tienes el don de aparecer en el momento que te necesito! Gracias a ti por obligarme a esforzarme durante toda la carrera, incluso cuando las “fuerzas” fallaban, gracias. A Lara, Rocío y Laura porque ellas también se lo merecen y para que vayáis pensando otro apodo para mí!*

*Gracias también a los Pioceros: Noelia, Erika, Iris, Ricar, etc. que se han interesado, me han animado y han sufrido en sus carnes las consecuencias de que “esto nunca acabará”. Lo celebraremos en Bélgica!*

*Por la ayuda técnica: a Henar y Fer (vivan!!!) por darle un poco de color! A Noe, que hubiese sido de mí y mis evaluaciones sin “tu Matlab”! A Peter, porque te mereces estar aquí*

*más que en el párrafo de arriba (yo nunca podría decirte que eres "uno más"!), gracias por tus innumerables ayudas en lo que es pfc y lo que no! Y en especial a Miriam, por tu locura y tu creatividad, porque nunca ha importado la hora ni el lugar, siempre has tenido un "huecco" para ayudarme o simplemente hacerme reír (a carcajadas!!!). Y porque con el mar en el horizonte, todo se hace más fácil... ☺ gracias por ayudarme a darle este enfoque final!*

*Mención especial también para los roameros de Oesia (on behalf of TME!)... por todas esas horas extras en la oficina dedicadas a lo que no era roaming!*

*Por último, a mi familia... a todos, por no cansaros nunca de preguntarme por "mi proyecto"! En especial a ti, Vanessa, gracias! A mi abuela porque sabía que estarías, y sé que estarás en muchas más (yo de mayor quiero ser tan fuerte como tú pero menos quejica! ☺). A mi hermana porque aunque yo no sé si le hará ilusión también se lo merece... y espero que no vuelvas a perderte otra! (boluda!!) Y como no, el agradecimiento más especial a mis padres, porque sin ellos nada de esto sería posible. A mi padre, porque nuestros triunfos son los vuestros... hoy eres un poco más ingeniero que ayer! A mi madre por ayudarme aun cuando yo no te lo pido y porque la energía que tienes nos llena a todos. Gracias a los dos!*

*A todos los que estáis porque lo sois y a los que no están pero también lo son, GRACIAS!*

*Sabía que disfrutaría de estas páginas...*

*Septiembre de 2009,  
Elena Ortiz.*

# Índices





# INDICE

---

<b>1</b>	<b>– INTRODUCCIÓN .....</b>	<b>1</b>
1.1	– Tratamiento Digital de Imágenes .....	1
1.2	– Motivación .....	2
1.3	- Objetivos .....	3
1.4	- Estructura de la Memoria.....	3
<b>2</b>	<b>– ESTADO DEL ARTE EN SEGMENTACIÓN DE IMÁGENES: APLICACIONES Y EVALUACIÓN .....</b>	<b>5</b>
<b>2.1</b>	<b>Introducción .....</b>	<b>5</b>
2.1.1	Segmentación de Imágenes .....	5
2.1.2	Clasificación de Métodos de Segmentación .....	6
2.1.3	Estructura del Capítulo .....	8
<b>2.2</b>	<b>Recuperación de Imágenes Basada en Contenido: Sistemas CBIR.....</b>	<b>9</b>
2.2.1	Sistemas CBIR que utilizan Segmentación .....	10
2.2.2	Anotación Automática de las Regiones en una Imagen .....	13
<b>2.3</b>	<b>Características (Features).....</b>	<b>15</b>
2.3.1	Color .....	15
2.3.2	Textura.....	15
2.3.3	Características Geométricas Generales .....	16
2.3.4	Modelos de Aplicación Específica .....	18
2.3.5	Interacción del usuario.....	18
<b>2.4</b>	<b>Métodos Bottom-Up de Segmentación Automática.....</b>	<b>19</b>
2.4.1	Agrupación .....	19
2.4.2	Morfología matemática.....	20
2.4.3	Algoritmos de Teoría de Grafos .....	21
<b>2.5</b>	<b>Métodos Top-Down de Segmentación.....</b>	<b>23</b>
2.5.1	Segmentación Semiautomática.....	23
2.5.2	Segmentación basada en las formas .....	26
<b>2.6</b>	<b>Evaluación .....</b>	<b>29</b>
2.6.1.	Estrategias de evaluación .....	29
2.6.2	Métodos de Evaluación Relativa .....	31
<b>2.7</b>	<b>Conclusión .....</b>	<b>33</b>
<b>3</b>	<b>– ANÁLISIS Y EVALUACIÓN DEL KMCC-TEXTURIZADO .....</b>	<b>35</b>
<b>3.1</b>	<b>– Introducción .....</b>	<b>35</b>
3.1.1	– Colección de imágenes utilizada .....	35
3.1.2	– Estructura del capítulo .....	36
<b>3.2</b>	<b>– Antecedentes del KMCC Texturizado.....</b>	<b>37</b>

3.2.1 – Algoritmo de Segmentación K-Means .....	37
3.2.2 – Algoritmo de Segmentación KMCC .....	38
<b>3.3 – Una Visión Global del KMCC-Texturizado.....</b>	<b>40</b>
<b>3.4 - Homogeneidad de Textura .....</b>	<b>41</b>
3.4.1 - ¿Qué es la textura? .....	41
3.4.2 – Técnicas de Análisis de texturas.....	42
3.4.3 – Discrete Wavelet Frames.....	43
<b>3.5 – Diseño e Integración del Método.....</b>	<b>45</b>
3.5.1 – Características de Color y Textura .....	45
3.5.2 – Agrupamiento Inicial.....	46
3.5.3 – Filtrado Condicional.....	48
3.5.4 – KMCC .....	50
3.5.5 – Imágenes de Grandes Formatos.....	51
<b>3.6 – Resultados de la Evaluación .....</b>	<b>53</b>
3.6.1 – Método de Evaluación.....	53
3.6.2 –KMCC-Texturizado.....	55
<b>3.7 – Conclusiones.....</b>	<b>60</b>
<b>4 – ANÁLISIS DEL ALGORITMO M-RSST .....</b>	<b>61</b>
<b>4.1 – Introducción .....</b>	<b>61</b>
4.1.1 – Motivación.....	61
4.1.2 – Estructura del capítulo .....	61
<b>4.2 – Algoritmo de Segmentación RSST .....</b>	<b>62</b>
<b>4.3 – Una Visión Global del M-RSST .....</b>	<b>64</b>
<b>4.4 - Homogeneidad de Color .....</b>	<b>65</b>
4.4.1 - Espacios de color.....	65
4.4.2 - Criterio de color optimizado .....	65
4.4.3 - Representación extendida de color.....	66
<b>4.5 - Características Geométricas.....</b>	<b>69</b>
4.5.1 - Adyacencia .....	70
4.5.2 – Regularidad o Ausencia de Complejidad .....	71
<b>4.6 – Diseño e Integración de Múltiples Fuentes .....</b>	<b>74</b>
4.6.1 - Teoría de la Evidencia.....	74
4.6.2 – Introducción a la teoría de Dempster-Shafer.....	75
4.6.3 – Aplicación a la unión entre regiones .....	76
4.6.4 – Diseño de las BBAs para cada fuente.....	78
<b>4.7 – Criterio de Parada .....</b>	<b>83</b>
4.7.1 – Criterio de Parada Basado en el PSNR.....	83
4.7.2 –Criterio de Parada Propuesto .....	83

<b>4.8 – Conclusiones.....</b>	<b>85</b>
<b>5 – ANÁLISIS Y EVALUACIÓN DEL MÉTODO HÍBRIDO.....</b>	<b>87</b>
<b>5.1 – Introducción.....</b>	<b>87</b>
5.1.1 – Objetivos.....	87
5.1.2 – Herramientas utilizadas .....	88
5.1.3 – Estructura del capítulo.....	89
<b>5.2 – Bases del Algoritmo Híbrido .....</b>	<b>90</b>
<b>5.3 – Integración de los Métodos .....</b>	<b>92</b>
5.3.1 – Estructura del KMCC-T .....	92
5.3.2 – Estructura del M-RSST .....	93
5.3.3 – Estructura del Método Híbrido .....	95
<b>5.4 – Resultados de la Evaluación .....</b>	<b>98</b>
5.4.1 – Método de Evaluación .....	99
5.4.2 – Método Híbrido .....	99
<b>5.5 – Conclusiones.....</b>	<b>104</b>
<b>6 – CONCLUSIONES Y TRABAJO FUTURO.....</b>	<b>105</b>
6.1 – Discusión.....	105
6.2 – Trabajo Futuro.....	106
<b>BIBLIOGRAFÍA .....</b>	<b>109</b>



# INDICE DE FIGURAS

---

Figura 2.1 – Un ejemplo de segmentación .....	5
Figura 2.2 – Posible clasificación de métodos de segmentación.....	7
Figura 2.3 – Palabras clave asociadas a regiones vs. Palabras asociadas a imágenes. ....	10
Figura 2.4 – Resultado obtenido tras la aplicación del detector de bordes Canny [REF1].....	11
Figura 2.5 – Resultados obtenidos en la asignación de palabras a regiones del método [9].....	14
Figura 2.6 – Comparación de los espacios CIE LUV (a) vs. CIE LAB (b).....	15
Figura 2.7 – Ejemplo de foto filtrada para análisis de texturas. ....	16
Figura 2.8 – Frames seleccionadas de un video en las que se muestra cómo se va deformando y adaptando el contorno activo o <i>Snake</i> a los labios del locutor. ....	17
Figura 2.9 – Ejemplo de un Binary Partition Tree (BPT) presentado en [BPT], sobre la imagen de una presentadora .....	21
Figura 2.10 – Ejemplo de imágenes con sus respectivas máscaras debajo [TOMASZ].....	28
Figura 2.11 – Conjunto de imágenes sintéticas, utilizado en [127] para la evaluación numérica de los experimentos. La imagen sintética original y su máscara de referencia se muestran en (a) y (b), respectivamente. Las siguientes ilustran distintos tipos de errores típicos en segmentación: la detección incorrecta de bordes en (c) y (d), sub-segmentación en (e) - (h), y sobre-segmentación (i) - (l). ....	33
Figura 3.1 – Ejemplo de segmentación en una imagen real (a) para comparar el K-means (b), y su versión mejorada KMCC (c). [ref21_mezaris] .....	39
Figura 3.2 – a) Las camisas son objetos texturizados si tratamos de identificar personas. b) La imagen podría ser una parte de una camisa hawaiana pero por la escala de la misma la textura será el fondo marrón y liso, mientras que las flores y hojas serán los objetos identificados. ....	41
Figura 3.3 – Texturas en las que se pueden observar sus diferentes características. [tesisUPV].....	42
Figura 3.4 – Fases del KMCC-Texturizado.....	46
Figura 3.5 – Descomposición Discrete Wavelet Frames 1-D de $V$ niveles. ....	47
Figura 3.6 – Esquema detallado del KMCC-Texturizado .....	49
Figura 3.7 – Ejemplo de foto filtrada condicionalmente. ....	49
Figura 3.8 – Esquema de los distintos niveles del KMCC-Texturizado .....	52
Figura 3.9 – Funciones de ponderación para el error espacial en función de la distancia del píxel con la región a la que pertenece.....	54
Figura 3.10 – Error espacial medio obtenido para los escenarios de la Tabla 3.1 con el KMCC-T .....	55
Figura 3.11 – Evaluación del factor LDC respecto al factor de reducción, $W$ . ....	57
Figura 3.12 – a) Errores medios obtenidos en función de $W$ con la aplicación o no del Teorema de Bayes. b) Mejora obtenida gracias al Teorema de Bayes para cada $W$ . ....	57
Figura 3.13 – Error medio obtenido variando el número máximo de áreas.....	58
Figura 4.1 – Influencia del parámetro $q_{avg}$ sobre el error medio espacial. ....	66
Figura 4.2 – Ejemplo de la aplicación del modelo de <i>Adaptive Distribution of Colour Shades</i> [31].....	67
Figura 4.3 – Influencia del parámetro $q_{ext}$ sobre el error medio espacial.....	68

Figura 4.4 – Ejemplo de un caso de jaggedness perjudicial al tener en cuenta la adyacencia. ....	70
Figura 4.5 – Ejemplo en el que la adyacencia empeora el resultado, negando la existencia de un fondo. .	71
Figura 4.6 - Esquema representativo de la dependencia entre las características sintácticas .....	72
Figura 4.7 – Principio de Transversalidad. Ejemplo de una secuencia de objetos que se superponen. ....	73
Figura 4.8 – Ejemplo de las BBAs sin considerar (a) y considerando (b) la fiabilidad de las fuentes [REF_TOMASZ] .....	77
Figura 4.9 – a) BBA original para el criterio de homogeneidad de color y b) BBA optimizada para no despreciar la mínima diferencia de color. ....	82
Figura 4.10 – Ejemplos de imágenes con sus respectivas funciones de coste acumuladas, parámetros $T_s$ , calculados para $T_{cum}=70$ y resultados de la segmentación utilizando la representación extendida de color y las características geométricas descritas anteriormente. ....	85
Figura 5.1 – Ejemplo de las primeras fases de segmentación en una imagen sintética para: KMCC- Texturizado en a) y M-RSST en b) .....	88
Figura 5.2 – Esquema de una posible fusión de ambos métodos basada en las funciones de decisión. ....	90
Figura 5.3 – Esquema de la fusión llevada a cabo para el método híbrido, marcada en rojo. ....	91
Figura 5.4 – Diagrama de bloques del algoritmo KMCC-T. ....	92
Figura 5.5 – Esquema de la herramienta de segmentación con el M-RSST incluido. ....	93
Figura 5.6 – Esquema para la integración de un método nativo en la herramienta de.....	94
Figura 5.7 – Esquema de la integración del método Híbrido en la herramienta de segmentación. ....	95
Figura 5.8 – Diagrama de bloque con las señales internas del KMCC-T .....	96
Figura 5.9 – Etapa del KMCC-T adaptada a la señal proveniente del M-RSST. ....	97
Figura 5.10 – Ejemplo de ejecución del evaluador junto con el fichero resultante. ....	98
Figura 5.11 – a) Mejora aportada por el método híbrido para cada escenario. b) Comparación de errores medios calculados para el KMCC-T y método híbrido.....	100
Figura 5.12 – Errores medios en función del número de áreas del KMCC-T y del método híbrido. ....	102
Figura 5.13 – Comparativa del error medio obtenido para el M-RSST con los dos anteriores. ....	103

# INDICE DE TABLAS

---

Tabla 3.1 – Distintos escenarios objetos de los experimentos y resultados obtenidos en ambas colecciones.....	56
Tabla 3.2 – Error para los escenarios de la Tabla 3.1 variando el número máximo de áreas. ....	58
Tabla 3.3 – Escenarios óptimos para la aplicación del KMCC-T sobre las imágenes seleccionadas.	59
Tabla 4.1 – Parámetros utilizados para el diseño de las BBA y para medir la credibilidad de las fuentes.....	78
Tabla 4.2 – Valores estimados para cada parámetro de la BBA de los criterios de homogeneidad de color. ....	81
Tabla 4.3 – Valores estimados para el valor de atenuación $\alpha_{adj}$ .....	81
Tabla 4.4 – Valores estimados para el valor de atenuación $\alpha_{cpx}$ .....	81
Tabla 5.1 – Escenarios y resultados obtenidos para cada colección por el método híbrido.....	100
Tabla 5.2 – Tabla con los resultados finales del KMCC-T y el método híbrido.....	101
Tabla 5.3 – Variación del error medio en función del parámetro del número máximo de áreas. ....	102



# 1 - Introducción





# 1 – Introducción

En primer lugar, en este capítulo se sitúa al lector en el contexto del tratamiento digital de imágenes. Posteriormente se describen las principales motivaciones y razones que la autora ha llevado a cabo para realizar esta investigación, cuyos resultados se plasman en este proyecto fin de carrera. Por último, se esbozan brevemente los objetivos principales del proyecto seguido de una visión general de la estructura de la memoria.

## 1.1 – Tratamiento Digital de Imágenes

En los últimos años, la tecnología de la computación ha transformado casi todos los aspectos de nuestra vida y cultura. La popularización del ordenador personal, unido a la continua evolución de las redes digitales y a la aparición de la tecnología multimedia, ha dado lugar a una tendencia mundial hacia una nueva "era digital". Los nuevos modelos de producción de contenidos, la distribución y el consumo se han traducido en un gran crecimiento de la cantidad de material digital disponible. Sin embargo, la gran cantidad de información disponible hoy en día también ha inducido un gran problema de sobrecarga. En otras palabras, la nueva "era digital" nos desafía a desarrollar la capacidad de encontrar información útil en un vasto océano de información. La necesidad de herramientas eficientes para organizar, manipular, buscar, filtrar y navegar a través de las enormes cantidades de información digital se pone de manifiesto en el espectacular éxito de los motores Web de búsqueda, basados en texto como Yahoo o Google [1].

La aparición de la tecnología multimedia y el abaratamiento de las herramientas digitales, como escáneres y cámaras digitales, así como la rápida reducción de coste del almacenamiento ha dado lugar a un crecimiento acelerado de las colecciones de medios audiovisuales, tanto de propiedad como de libre disposición en Internet. Actividades como la fabricación, la medicina, el ocio, la educación, etc. hacen uso de inmensas cantidades de datos audiovisuales y según la cantidad de información en formato digital aumenta, la necesidad del desarrollo de herramientas para una efectiva transformación, almacenamiento, gestión, búsqueda y recuperación de la misma se hace evidente.

Una gran cantidad de material visual está disponible en forma de imágenes fijas, ya sea como fotografía de una escena real o como un gráfico que contiene una imagen no real (esbozado por un hombre o sintetizados por ordenador). Las imágenes pueden almacenarse en bases de datos locales o bases de datos distribuidas (como por ejemplo, la World Wide Web) y por otro lado, pueden estar incorporadas en documentos o disponible como objetos por sí mismas.

Algunas de las áreas de aplicación que figuran con más frecuencia en la literatura, donde el uso y recuperación de imágenes hoy en día desempeña un papel fundamental, son: la ley y prevención del delito, la medicina, la moda y el diseño gráfico, editoriales, comercio electrónico, arquitectura e ingeniería de diseño, investigación histórica, etc. Un estudio detallado de los usuarios de imágenes y los posibles usos de éstas ha sido realizado por Eakins y Graham en [2].

Como conclusión, debido a todo lo mencionado existe una clara necesidad de motores de búsqueda de imágenes, ya sea para particulares o uso profesional.

## **1.2 – Motivación**

La segmentación es el primer paso a realizar en cualquier aplicación de análisis de imágenes. Es de gran importancia que en esta primera toma de contacto con la imagen se obtengan resultados fiables ya que serán los cimientos sobre los que después se construirá todo el edificio, es decir, la aplicación. En la actualidad, existe una gran diversidad de aplicaciones basadas en segmentación, como por ejemplo: la indexación de imágenes y vídeos o el reconocimiento de objetos. En éstas y muchas más, es imprescindible conocer de cuántos elementos está compuesta la imagen, su posición, etc. En definitiva, durante el proceso de segmentación no será una preocupación saber qué es cada objeto, sino más bien dónde hay un objeto.

Para cualquiera de nosotros identificar y reconocer un objeto en una imagen puede resultar, a simple vista, una tarea sencilla. Sin embargo, automatizar esta acción tan cotidiana para el ojo humano, puede convertirse en una ardua tarea para un algoritmo de tratamiento de imágenes. Prueba de esto es el estado actual del arte que presenta un amplio abanico de posibilidades con resultados “no siempre satisfactorios”.

La motivación de este proyecto fin de carrera es contribuir al avance de las técnicas de segmentación. Resolver el problema de la división de una imagen en un conjunto de entidades, considerado como paso fundamental para conseguir la comprensión de la estructura de la escena e identificar los objetos relevantes, es la motivación principal de cualquier trabajo sobre segmentación. Para ello realizamos como primer paso un cuidadoso estudio del estado del arte, analizando cuidadosamente varios de los métodos más destacados para afianzar conocimientos relacionados con el campo. En concreto, se pretende defender la idea de que dada una cierta aplicación, es propicio investigar todas las posibles fuentes de evidencias existentes en una imagen. El buen funcionamiento de la segmentación automática junto con la idea de que ésta (aunque imperfecta) es un paso esencial en la comprensión de una escena, abogan por la aparición de nuevos trabajos como este, en contra de la opinión desesperanzada de muchos otros investigadores que lo consideran como un reto inalcanzable.

### 1.3 - Objetivos

Debido a la gran cantidad de documentación existente, el primer objetivo será un estudio del estado del arte con la intención de seleccionar aquellos algoritmos más destacados, basándonos en las últimas aportaciones publicadas, y teniendo en cuenta la intención del trabajo a llevar a cabo.

El segundo objetivo será seleccionar aquellos algoritmos que bajo nuestro punto de vista puedan soportar mejor alguna mejora en su funcionalidad. Este proyecto final de carrera se centra primordialmente en el estudio de dos métodos de segmentación. El primero de ellos se centra en el estudio de dos características de bajo nivel como son el color y la textura, y el segundo se basa en el análisis de las formas de las regiones y objetos.

Seguidamente, se llevará a cabo un exhaustivo análisis de estos algoritmos sobre los que se trabajará con el fin de poder buscar algún tipo de avance para realizar así las aportaciones oportunas al estado del arte.

Posteriormente, se realiza un amplio análisis experimental con el objetivo de incentivar la creación de un nuevo método híbrido y abrir así un pequeño sendero a los numerosos y amplios caminos ya explorados en el terreno de la segmentación de imágenes. El avance propuesto será la combinación de ambos algoritmos, mejorando los resultados del primero con las propuestas del segundo.

Gracias a la aportación del software de ambos algoritmos por parte de sus autores, este PFC se ha centrado en el importante objetivo de integración de múltiples características. Son muchos los trabajos en la literatura que tratan de añadir una o dos características a lo sumo, a la universalmente utilizada: el color. Pero el objetivo principal de este trabajo es tratar de abrir una puerta hacia la integración de múltiples y diversas características dentro de un mismo método. Esta idea se tratará de apoyar mediante los resultados de un amplio estudio experimental, como se verá a lo largo de esta memoria.

El objetivo final es indicar posibles direcciones para futuras investigaciones así como examinar las posibilidades de mejoras o ampliación del este proyecto.

### 1.4 - Estructura de la Memoria

La estructura de la memoria es la siguiente:

- **Capítulo 1**: Introducción, motivación y objetivos del proyecto.
- **Capítulo 2**: Resumen del estado del arte actual en segmentación de imágenes incluyendo señales características, una breve descripción de distintos tipos de métodos y varias estrategias de evaluación.

- **Capítulo 3**: Análisis y evaluación del método KMCC-T. Se entrará en detalles sobre el análisis de texturas para posteriormente mostrar la integración de esta en el método global. Además se llevarán a cabo una serie de evaluaciones del algoritmo en distintos escenarios.
- **Capítulo 4**: Análisis del método M-RSST. Se definen todas las características utilizadas en el algoritmo y se define detalladamente el diseño del mismo con el fin de familiarizarse con él.
- **Capítulo 5**: Análisis y evaluación del método híbrido. Se mostrará tanto teórica como esquemáticamente el diseño y desarrollo de la integración de los dos métodos vistos en los capítulos anteriores. Para concluir, se evaluará y analizará el resultado.
- **Capítulo 6**: Conclusiones y trabajo futuro.

## 2 - Estado del arte en segmentación de imágenes: Aplicaciones y evaluación





## 2 – Estado del Arte en Segmentación de Imágenes: Aplicaciones y Evaluación

En este capítulo se analiza la importancia de la segmentación dentro de un contexto histórico y en especial desde el punto de vista de las aplicaciones más comúnmente utilizadas, esto es, la recuperación de imágenes basándose en el contenido. Se muestra un esbozo de los principales enfoques que se encuentran a día de hoy en la literatura y se revisan detenidamente los métodos más característicos. También se analiza con detalle el tema de la evaluación objetiva de la calidad de la segmentación, que ha recibido relativamente poca atención en la literatura, sobre todo si lo comparamos con el gran número de publicaciones relativas al tema de la segmentación de imágenes propiamente dicho.

### 2.1 Introducción

#### 2.1.1 Segmentación de Imágenes

El término de segmentación (a menudo referido como agrupación o partición), abarca una amplia serie de procesos a través de los cuales se obtiene la división de la imagen en distintas regiones espaciales disjuntas (ver **Figura 2.1**), atendiendo a una cierta homogeneidad de éstas. Dicha homogeneidad se establece según una serie de criterios tales como:

- Características de bajo nivel (por ejemplo, el color o la textura).
- Conocimiento previo sobre la escena o imagen (por ejemplo, suavidad de los bordes).
- Conocimientos de alto nivel (estos son los denominados, modelos semánticos)
- O incluso la interacción de usuarios.



**Figura 2.1** – Un ejemplo de segmentación automática.

La segmentación de imágenes es un paso fundamental para la comprensión de la estructura y la identificación de los objetos en una escena. La diferenciación de las áreas de una imagen, que se corresponderán con zonas homogéneas y/o regiones importantes conocidas como objetos semánticos, es a menudo establecida como el primer paso en muchas aplicaciones basadas en objetos, como por ejemplo, la indexación y recuperación de imágenes (por ejemplo, mediante descriptores del estándar MPEG-7 [3]) o la codificación de la región con mayor o menor calidad en función de su interés (por ejemplo, mediante las funcionalidades propuestas en el estándar JPEG2000 [4]).

### ***2.1.2 Clasificación de Métodos de Segmentación***

Ante la gran variedad de enfoques y métodos de segmentación propuestos a lo largo de la literatura (por ejemplo [6, 7, 8]), así como exhaustivos estudios en [9, 10], es inevitable que surja la intención de clasificarlos de acuerdo a distintos criterios. A pesar de los múltiples intentos de clasificación, elegir el mejor de los criterios seguidos es una ardua tarea; casi tan complicada como elegir el mejor método de segmentación. Sin embargo, el pensamiento generalizado acerca de la inherente jerarquía existente en la división de una imagen, hace prever que una estrategia basada en estructuras de árboles puede acercarse más a la realidad, por lo que algunos autores las dividen en los siguientes grupos:

- **Basados en regiones**, que dependen de la homogeneidad de las características localizadas espacialmente (color, textura, etc.).
- **Basados en contornos**, que utilizan habitualmente información del gradiente para localizar los bordes de las distintas áreas.

Otros grupos de clasificación, según el método matemático empleado son:

- Variacionales.
- Estadísticos.
- Basados en Grafos.
- Morfológicos.

Un criterio interesante, tomado en cuenta en algunos trabajos [11], es el nivel de información utilizada para la agrupación – ver **Figura 2.2**. Teniendo éste en cuenta podríamos hacer una amplia división [12]:

- **BOTTOM-UP**: Se intentan deducir entidades significativas (idealmente objetos) únicamente a partir del análisis de las características visuales de la imagen, por ejemplo, la homogeneidad del color, la textura o las características geométricas. Su principal desventaja es la complejidad de los algoritmos.

- *TOP-DOWN*: Se segmenta la imagen en aquellos objetos cuyas características coincidan más fielmente con las de los modelos utilizados. En este caso se utilizan también las características geométricas para analizar los objetos buscados. La principal desventaja son las limitaciones dependientes de la aplicación.

Cabe destacar el hecho de que el término *bottom-up* también se puede encontrar a menudo refiriéndose a métodos que unen iterativamente píxeles, formando entidades más complejas; mientras que *top-down* se referirá a técnicas de división de regiones o, a métodos donde la segmentación de la imagen submuestreada se utiliza para obtener una mayor resolución de dicha segmentación a posteriori. Las dos parejas de conceptos van bastante ligadas aunque no se debe caer en el error de considerarlos siempre iguales. A lo largo de esta memoria ambos se usan principalmente para referirse al nivel de información utilizado y no necesariamente para indicar en qué orden se utilizan las diferentes escalas de las imágenes segmentadas.

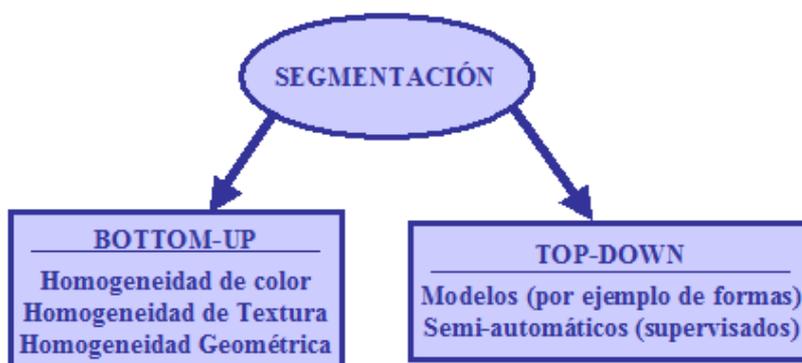


Figura 2.2 – Posible clasificación de métodos de segmentación

Alternativamente, también podemos clasificar los métodos en función de la salida que produzcan en:

- **Basados en objetos**: aquellos de los que se obtienen objetos semánticos como resultado.
- **Basados en regiones**: cuyos resultados son zonas de color y/o textura homogéneas.

Salta a la vista la estrecha relación que existe entre los dos últimos criterios, puesto que normalmente los objetos semánticos significativos sólo pueden ser identificados mediante la utilización de modelos, de ciertos conocimientos previos acerca de una aplicación particular o bien mediante alguna orientación por parte del usuario.

Por otro lado, los enfoques ligados a características visuales normalmente se basan en un criterio de homogeneidad, como el color o la textura. Por esta razón, son atractivos en

muchas aplicaciones que no requieren o no disponen de un modelo para cada uno de los objetos ni de la posibilidad de interacciones de los usuarios. Sin embargo, estos métodos no siempre son útiles si hablamos de extraer objetos semánticos completos debido a que la segmentación por características visuales está sometida a restricciones en muchas aplicaciones. Normalmente, en el caso de contenido general, estos enfoques sólo consiguen dividir la imagen de entrada en regiones homogéneas, aunque en aplicaciones específicas si podrían llegar a detectar objetos.

En contraste con esto, el uso de modelos, a pesar de requerir un conocimiento previo sobre los objetos a extraer, produce una detección bastante precisa. Estos enfoques son usados principalmente para la extracción de objetos semánticos importantes y en aplicaciones que buscan la separación de éstos del resto de la parte irrelevante de la escena, lo que generalmente quiere decir, del fondo.

A pesar de todo, ambos enfoques (características visuales y modelos) no son necesariamente excluyentes y de hecho es realmente interesante su combinación tanto para segmentación como para identificación de objetos, como intentaremos comprobar a lo largo de este trabajo.

Podemos encontrar un buen ejemplo en [13], donde se utiliza una segmentación *bottom-up* para obtener una partición simple de la imagen y poder representar ésta de manera jerárquica mediante un *Binary Partition Tree* (BPT). Este BPT se utiliza después como guía en la búsqueda de la relación óptima entre contornos de referencia y de las zonas homogéneas para la segmentación y detección de objetos.

### **2.1.3 Estructura del Capítulo**

El resto de este capítulo está organizado de la siguiente manera: en la siguiente sección discutiremos la importancia de la segmentación de imágenes en el contexto de aplicaciones CBIR (Content-Based Image Retrieval), para ello nos centraremos principalmente en la revisión de los sistemas CBIR seleccionados que hacen uso de la segmentación y en la descripción de algunos métodos de recuperación de imágenes mediante segmentación automática. La sección 2.3 describe brevemente las características (*features*) que pueden ser utilizadas para la segmentación de imágenes. Una selección de métodos *bottom-up* y *top-down* se examinan en las secciones 2.4 y 2.5, respectivamente. Varios métodos de evaluación son examinados brevemente en la sección 2.6. En el punto 2.7 se desarrolla una breve discusión de las posibilidades de la segmentación de imágenes en el contexto de CBIR y por último se formularán unas breves conclusiones respecto al estado del arte.

## 2.2 Recuperación de Imágenes Basada en Contenido: Sistemas CBIR.

El uso de la segmentación en sistemas CBIR (*Content-Based Image Retrieval*) permite la representación de imágenes divididas en regiones homogéneas. Estas regiones se corresponderán en el caso ideal con objetos semánticos (o en su defecto con partes significativas de éstos) y permitirán la obtención de las características de la escena. Generalmente el objetivo será la obtención de los denominados descriptores de dominio específico [14], para finalmente indexar y recuperar las imágenes de una manera más cercana a la percepción humana, es decir, lograr la comprensión de la escena.

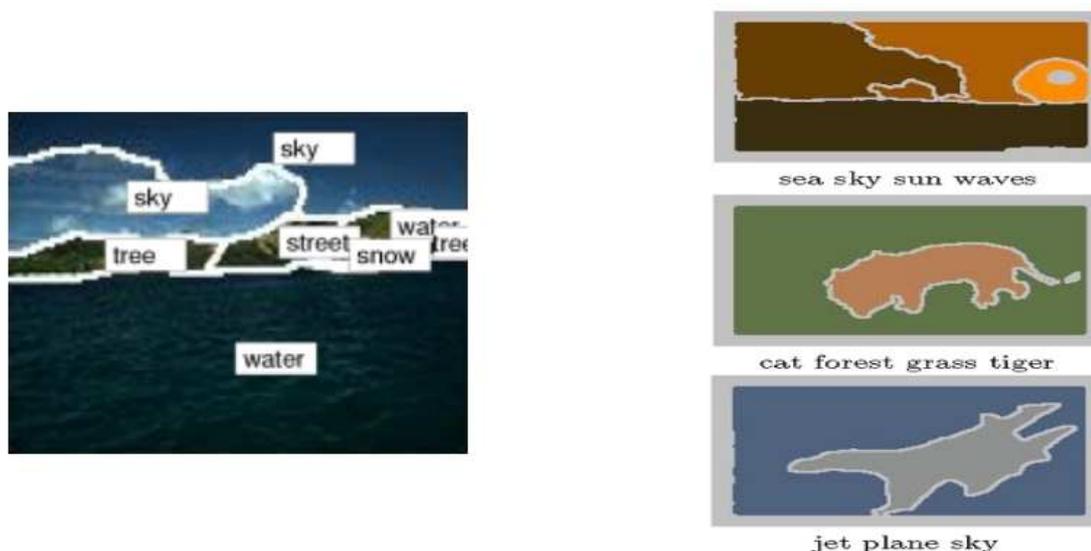
Dentro del estado del arte encontramos los sistemas CBIR como una de las principales aplicaciones que hacen uso de la segmentación. Por esta razón existen múltiples estudios guiados y/o métodos diseñados en torno a estas aplicaciones [6, 15, 16].

Según [17], podemos dividir la utilización de los sistemas de segmentación basados en regiones en dos categorías, en función de la estrategia utilizada para hallar la correspondencia entre las regiones de la imagen consultada y la imagen objeto. Aunque el estudio se realiza sobre la base de la comparación entre *frames* de vídeos y hace incisos en otros temas como cuantificaciones, nos fijamos únicamente en los diferentes métodos utilizados para la comparación trasladándolos a imágenes fijas:

- Asociación por una región individual (*Individual Region Matching*): donde una región seleccionada por el usuario sobre la imagen de consulta (*query*) se compara con todas las regiones de la colección, esto es, con todas las regiones de todas las imágenes. Se realiza una medida de las comparaciones. Al final, estas medidas se ordenan individualmente por regiones, indicando la relevancia de cada imagen. Otra modalidad dentro de ésta, es la posibilidad de fusionar alternativamente distintas regiones (teniendo en cuenta sus medidas de comparación) obtenidas de varias consultas (*queries*).
- Asociación por todas las regiones del *frame* (*Frame Region Matching*): se diferencia ligeramente de la anterior en que en este caso se utiliza la información de todas las regiones que componen cada imagen.

Posteriormente, se demostró cómo las palabras clave pueden asociarse con las regiones de las imágenes automáticamente mediante el algoritmo de máxima esperanza y un proceso previo de aprendizaje e indexación [18]. Gracias a esto surge otra posible categoría de sistemas que utilizan la segmentación. Esas palabras pueden ser entonces utilizadas para la indexación y, posteriormente, para consultas mediante texto – ver **Figura 2.3**.

El resto de esta sección tiene como objetivo destacar la importancia de la segmentación en contextos CBIR mediante la revisión de algunos enfoques seleccionados que utilizan algún tipo de partición de imagen.



**Figura 2.3** – Palabras clave asociadas a regiones vs. palabras asociadas a imágenes.

### 2.2.1 Sistemas CBIR que utilizan Segmentación

En los últimos años se ha producido una explosión de sistemas de recuperación de imágenes, desarrollados como plataformas para facilitar la investigación o incluso como productos comerciales. Podemos encontrar amplias revisiones de éstos en [19, 20]. Esta sección ofrece una breve revisión de algunos de los sistemas más conocidos en la literatura, que utilizan la segmentación de imágenes.

Uno de los primeros sistemas CBIR, y también uno de los primeros en proporcionar funcionalidades basadas en detección de objetos fue el conocido como **Query By Image Content** (QBIC) que fue desarrollado por parte del IBM Almaden Research Center [21, 22]. En este sistema se extraen características simples de bajo nivel (color, textura y unos descriptores de forma sencillos) de imágenes u objetos segmentados semi-automáticamente. La segmentación se basa en la técnica de contornos activos [23] que realiza un alineamiento del esbozo aproximado del objeto (proporcionado por el usuario) y el borde más parecido encontrado de la imagen. Cada objeto extraído puede ser anotado mediante descripción textual. Todas las imágenes son representadas por un mapa binario reducido de los bordes para permitir la recuperación basada en los bocetos. El sistema permite la combinación de consultas de palabras clave, es decir, basadas en texto con varias consultas visuales como por ejemplo

queries-by-examples, que segmenta la imagen a partir de un boceto aproximado y la selección de los patrones de color y textura.

Uno de los primeros sistemas de recuperación imagen que utiliza segmentación automática es el **Picasso**, desarrollado por el laboratorio *Visual Information Processing*, de la Universidad de Florencia [24]. El núcleo del sistema es una segmentación piramidal en función del color de las imágenes (es decir, en el nivel más bajo, cada región consta de un solo píxel, mientras que la parte superior de la pirámide se corresponde con la imagen completa) y mediante mezclado iterativo de las regiones adyacentes se irán formando distintos niveles. Éstos se traducirán posteriormente en la pirámide como niveles de resolución de la segmentación. La representación superior se almacena en forma de un grafo en el que múltiples nodos que representan a las regiones se caracterizan por el color (un vector de color, binario y de 187 dimensiones), la posición de su centroide, el tamaño y la forma (alargamiento y la orientación del eje principal). Además, en el conjunto completo de la base de datos, los objetos de interés en cada imagen están delimitados con su mínimo rectángulo y se usa el detector de bordes *Canny* ([25] – véase un ejemplo de su aplicación en la **Figura 2.4**) para extraerlos en cada una de estas áreas rectangulares. El sistema permite consultar por regiones de color (dibujando una región con un color específico o mediante el bosquejo de los contornos de algunos objetos relevantes en una imagen ejemplo), por la textura, o por la forma (croquis dibujados).



**Figura 2.4** – Resultado obtenido tras la aplicación del detector de bordes Canny [25]

**Netra**, desarrollado por el departamento de *Electrical and Computer Engineering*, de la Universidad de California, Santa Bárbara, es otro sistema CBIR que utiliza segmentación de imagen [26]. En este sistema las imágenes son automáticamente segmentadas en regiones homogéneas utilizando una técnica de flujo de bordes estudiada y desarrollada en la misma universidad. Así, cada región se caracteriza por el color (representado por un código de 256 colores), textura (representada por un vector que contiene la media normalizada y la desviación estándar de una serie de transformadas wavelet de filtros de Gabor), la forma (representada por tres vectores de diferentes características: curvatura en cada punto del contorno, función distancia al centroide, y descriptores de Fourier) y la ubicación espacial. La consulta se formula

mediante la selección de una de las regiones de la imagen de consulta o, alternativamente, si la imagen de ejemplo no está disponible, directamente por la especificación del color y la ubicación espacial. Se utilizan distancias euclídeas ponderadas con las que se presentan los resultados de forma lineal en función de la similitud.

**IPure** es un sistema CBIR desarrollado por el IBM India Research Lab, de Nueva Delhi. En este sistema las imágenes están segmentadas en regiones de color homogéneo usando el algoritmo Mean-Shift [27]. Cada región está representada por el color (color medio en el espacio CIE LUV), la textura (coeficientes de una descomposición de la imagen), y la forma (tamaño, orientación de los ejes, y descriptores de Fourier). Además, la disposición espacial es caracterizada por el centroide, el mínimo *bounding-box* (rectángulo capaz de contener la región), y la contigüidad. La imagen consultada ha de ser segmentada siempre en un primer paso y después comenzará la búsqueda con la posibilidad de feedback por parte del usuario.

**Istorama** es un sistema de recuperación de imagen desarrollado por el Instituto de Informática y Telemática, y el Centro de Investigación y Tecnología Hellas, en Grecia [28]. En este sistema todas las imágenes están automáticamente segmentadas por una variante del algoritmo K-Means llamado KMCC [7]. Cada región se caracteriza por su color, tamaño y ubicación. Al igual que en el sistema Blobword, el usuario lanza una consulta seleccionando una única región de la imagen consultada. Además, el usuario puede acentuar o disminuir la importancia de una característica específica, mediante la adaptación de los pesos asociados a cada una de ellas.

**Blobworld** es un sistema CBIR desarrollado en la división de Ciencias de la Computación (Computer Science Division), Universidad de California, Berkeley [6]. En este método se segmentan regiones de imágenes con color y textura uniforme (los también llamados blobs) mediante una variante del algoritmo de la esperanza máxima (Expectation Maximization - EM). Cada región se caracteriza por su color (representada por un histograma de las coordenadas del color en el espacio CIE LAB), su textura (caracterizada por el contraste y la anisotropía de la región), y unos descriptores simples de la forma (tamaño, excentricidad y orientación).

La **query-by-example** (consulta mediante ejemplo), nombrada anteriormente, se formula mediante la selección de una o más regiones desde una de las imágenes consultadas seguida por la especificación de la importancia del blob en sí y también su color, textura, ubicación y forma.

Una metodología interesante para la recuperación de imágenes se presentó en [16]. En este enfoque las imágenes están segmentadas automáticamente por una variante del algoritmo

K-Means llamado KMCC [7] y las consiguientes regiones están representadas por características de bajo nivel, tales como el color, posición, tamaño y forma. La principal innovación de esta metodología es la asociación automática de estos descriptores con unos de un nivel intermedio, pero superior, que forman un vocabulario simple denominado en castellano objeto ontológico (*object ontology*). Dicho vocabulario permite la definición de nuevos conceptos de alto nivel y su consecuente relación con las tendencias en humanos. Por tanto, facilita la consulta usando conceptos principalmente semánticos. Está garantizada su viabilidad en colecciones genéricas sin requerir una definición previa de las correspondencias entre regiones e identificadores, por la simplicidad del “vocabulario” empleado. La utilización del objeto ontológico permite limitar la búsqueda a un conjunto de imágenes potencialmente relevantes. Por último, se utiliza un mecanismo de feedback, basado en *Support Vector Machines* (SVM) junto con descriptores de bajo nivel, para el perfeccionamiento en la recuperación de resultados.

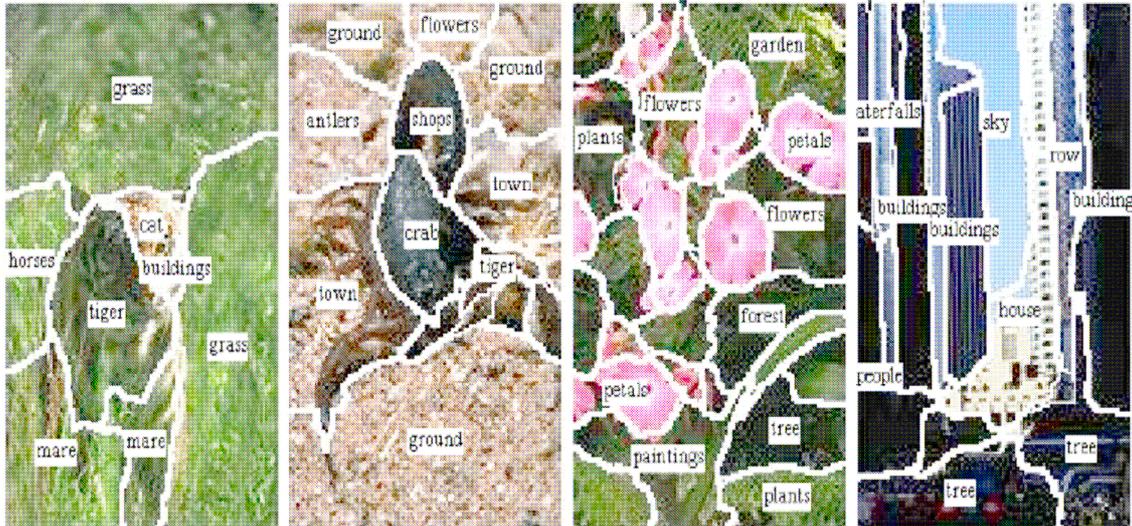
### **2.2.2 Anotación Automática de las Regiones en una Imagen**

Algunos estudios de usuarios sostienen que en la práctica, las consultas basadas en características globales de bajo nivel (histogramas de color, textura, etc.) son sorprendentemente escasas, y al mismo tiempo, demuestran la gran utilidad de un texto relacionado con la imagen [29].

Curiosamente, una serie de trabajos [18] muestran cómo se pueden asociar palabras clave a regiones de una imagen automáticamente mediante un proceso previo de entrenamiento y/o indexación. Estas etiquetas pueden ser usadas después para indexar la imagen y consecuentemente realizar consultas mediante texto.

Uno de los estudios más prometedores hoy en día en este campo se describe en [30]. En él se muestra cómo mediante un modelo estadístico se pueden vincular las palabras con los datos de la imagen, sin una codificación explícita de la correspondencia entre palabras y regiones. Este método combina modelos de aspecto con un método suave de agrupación. Las imágenes y las palabras concurrentes generadas forman nodos que se disponen en una estructura con forma de árbol. La probabilidad de asociación entre palabras y regiones se modela mediante la colección de dichos nodos, que contienen la distribución de probabilidad entre ellas. En el caso de las regiones las distribuciones son gaussianas, mientras que la probabilidad de las palabras es proporcionada por tablas de frecuencia. Las características de una región indicarán por tanto, la probabilidad de encontrarse en uno u otro nodo. Estas probabilidades se utilizan para calcular el peso de cada nodo y así lograr finalmente la emisión de la palabra. Se utiliza el algoritmo de esperanza máxima para estimar los parámetros de la vinculación entre palabras y regiones a partir de la concurrencia de los datos palabra-región usados. A pesar de haberse

demostrado que sólo ciertas regiones son etiquetadas razonablemente bien (por ejemplo: cielo, agua, nieve, gente, peces, aviones), este método representa sin duda un interesante enfoque para el reconocimiento de objetos en imágenes – véase **Figura 2.5**.



**Figura 2.5** – Resultados obtenidos en la asignación de palabras a regiones del método [30].

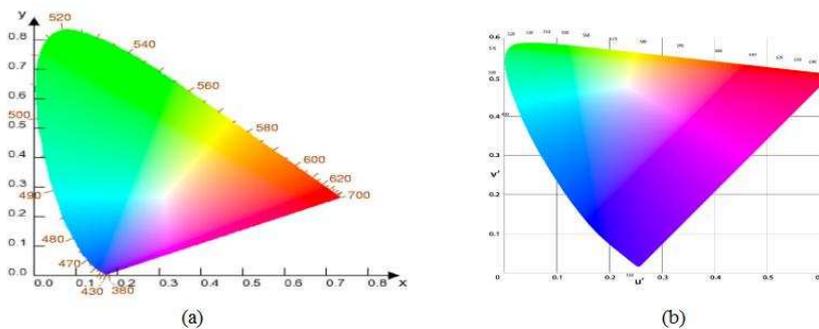
En [31] fueron evaluadas tres categorías de algoritmos de segmentación de imagen (el criterio de máxima esperanza utilizado en *Blobworld* [6], *Normalized Cuts* [32] y *Mean-Shift* [27]) en el contexto de la anotación descrita anteriormente. Por ejemplo, en términos de rendimiento de predicción de la palabra, se constató que el enfoque *Normalized Cuts* ofrece el mejor resultado de los tres métodos de predicción de palabras seguido de cerca por el segmentador *Mean-Shift*.

## 2.3 Características (*Features*)

En esta sección se examinan las posibles características que son prácticas habitualmente en métodos de segmentación de imágenes.

### 2.3.1 Color

El color es la característica principal de bajo nivel en prácticamente todas las técnicas de segmentación de imágenes que se pueden encontrar en la literatura. Una de las principales cuestiones relacionadas con esta característica es la elección de un espacio de color adecuado. Por lo general, es ventajoso que éste garantice una baja correlación entre los componentes y que sea uniforme según la percepción visual, es decir, que la distancia numérica sea proporcional a la percepción visual de la diferencia de color. Debido a los requisitos anteriores se encuentran muchas publicaciones [6, 7, 32] que defienden la utilización de CIE LUV o en su defecto la versión mejorada de éste, CIE LAB (ver **Figura 2.6**), puesto que para colores similares son aproximadamente uniformes.



**Figura 2.6** – Comparación de los espacios CIE LUV (a) vs. CIE LAB (b)

### 2.3.2 Textura

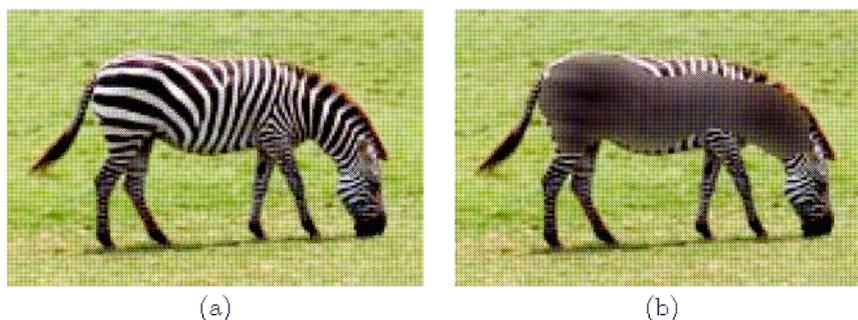
La textura es posiblemente la segunda característica más importante de bajo nivel después del color. Varios estudios sobre la caracterización de la textura [6, 7, 33] han demostrado su gran utilidad en la segmentación, incluyendo los bancos de filtros multiorientados [33] y la matriz del segundo momento [6]. Sin embargo, la utilización de la textura en la segmentación de imágenes no debe considerarse simplemente como una extensión irreflexiva de la característica de color.

Aunque se hará un análisis más exhaustivo sobre esta característica más avanzado el trabajo, hay varias cuestiones a tener en cuenta al hacer uso de la información de textura si se pretende alcanzar resultados satisfactorios para imágenes naturales que con frecuencia entremezclan objetos texturizados con no texturizados.

En primer lugar, podemos destacar un interesante debate sobre la utilización de la textura en la segmentación en [6, 33], donde los autores discuten temas como la selección

adaptativa de la escala, es decir, el tamaño de la ventana utilizada para calcular el descriptor de textura, y el hecho (rara vez discutido en la literatura) de cómo los contornos constituyen un grave problema para muchos métodos de análisis de textura. Esto es, porque los filtros sintonizados a distintas frecuencias espaciales (utilizados a menudo) producen grandes respuestas a lo largo de la dirección de dichos bordes que pueden enmascarar la información relevante.

Por otro lado, cabe señalar cómo el uso del color y la textura conjuntamente resultan a menudo conflictivos (por ejemplo, podríamos considerar el caso de una cebra) y las decisiones y/o resoluciones sobre ello deben abordarse en la etapa final del proceso. Actualmente, el método más común para solventar este tipo de conflictos es un filtrado condicional de las componentes de color, es decir, se suavizan las áreas con textura con el fin de garantizar la homogeneidad de su color (por ejemplo, la cebra se convierte en un caballo gris (característica de color suavizada) con rayas (textura característica)- véase **figura 2.7**) [7].



**Figura 2.7** – Ejemplo de foto filtrada para análisis de texturas [7].

### 2.3.3 Características Geométricas Generales

Muchos han sido los investigadores que han tratado de incorporar el uso del conocimiento previo sobre formas geométricas de los objetos [34, 35] en el proceso de segmentación. La idea principal es utilizar algunas señales geométricas genéricas, que permitan la segmentación de imágenes en sus entidades más significativas, y evitar así la necesidad de modelos más específicos (y complejos). Un reflejo de esto son los métodos basados en el “paradigma del modelo deformable de contornos activos” que incorporan un conocimiento previo sobre la suavidad de los bordes y la resistencia de éstos a las deformaciones [23] - véase **Figura 2.8**.

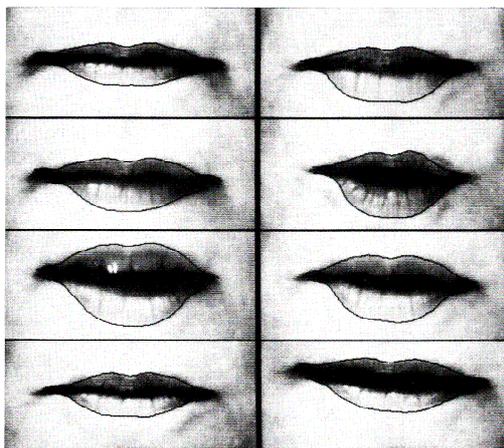
Existen otros métodos que tratan de realizar una agrupación visual, definida normalmente como el proceso que organiza las entidades de la imagen (básicamente bordes y/o regiones) en una estructura de niveles, según la base de algunas medidas genéricas de percepción geométrica.

La mayoría de estos estudios vienen directamente motivados por teorías procedentes del campo de la psicología cognitiva que se relacionan con la percepción humana y el

reconocimiento de objetos. Como ejemplo podríamos nombrar los *Principles of Gestalt Theory* analizados en detalle en [36] y el *Principle of Common-Cause* [37], basado en la idea de que existe una muy pequeña probabilidad de encontrar múltiples relaciones en la información visual.

Las medidas utilizadas habitualmente para la agrupación visual (y potencialmente para la segmentación) son:

- Proximidad
- Continuidad
- Cierre
- Compactibilidad
- Paralelismo de los bordes
- Co-linealidad (Collinearity)
- Concentricidad (Cocircularity)



**Figura 2.8** – Frames seleccionadas de un video en las que se muestra cómo se va deformando y adaptando el contorno activo o *Snake* a los labios del locutor.

Otro ejemplo de un conjunto de medidas útiles para la segmentación son el conjunto de las denominadas “*Características Sintácticas Visuales*”, y en particular el *Criterio de Cuasi-Inclusión*, por el cual abogaron originariamente Salembier y Ferrán, y éste último junto con Casas en [12], para la creación de los Árboles de Partición Binarios (BPTs). Éstas características incluyen: homogeneidad (por ejemplo de forma), compactibilidad, regularidad, inclusión y simetría.

Aunque la utilización de estas propiedades geométricas así como de las relaciones estructurales para la segmentación es algo intuitivo, la integración de toda esa información con propiedades como el color o la textura está lejos de ser una tarea trivial. En los métodos propuestos hasta hoy, esta integración se realiza en modo ad-hoc. A esto se suma que a pesar de

que la generalidad de muchas de las señales geométricas es discutible, se ha trabajado muy poco en la demostración cuantitativa de su utilidad en grandes colecciones (normalmente se muestra un puñado de ejemplos de la segmentación).

### ***2.3.4 Modelos de Aplicación Específica***

En algunas aplicaciones es posible utilizar un modelo previo, descriptivo de lo que se espera en la imagen hasta reducir las ambigüedades y concluir con técnicas totalmente automáticas de segmentación, detección y reconocimiento de objetos. En otras palabras, la división consiste en un proceso guiado por modelos de los objetos específicos buscados. Obviamente, estos modelos deberán permitir medir de algún modo las deformaciones comunes en el mundo real y las variaciones en las distintas clases de objetos.

Los modelos utilizados normalmente para segmentación y reconocimiento de objetos son modelos de formas [38, 39, 40, 41, 42] y de variación de color (o intensidad), diseñados para sintetizar la imagen del objeto modelado (por ejemplo, *Active Appearance Models* (AAMS) [43] o *Eigenfaces* [44]).

### ***2.3.5 Interacción del usuario***

Los métodos *semiautomáticos* o *supervisados* permiten que el usuario, mediante sus interacciones, defina qué objetos han de ser segmentados dentro de una imagen. Como resultado de dichas interacciones se obtienen pistas a cerca del tipo de exigencias de la búsqueda que luego son utilizadas en un proceso automático para segmentar y detectar el objeto. El usuario debe tener la posibilidad de obtener exactamente la segmentación que desee en términos de contenido y exactitud. La cantidad de interacciones necesarias no debe ser excesiva y éstas deben ser realizadas con facilidad y rapidez. Asimismo, es importante que el sistema responda a las interacciones de los usuarios en tiempo real (idealmente de forma instantánea).

Esta condición se asume más crucial según se trate de casos en los que las interacciones se deben aplicar repetidamente con el fin de perfeccionar el resultado de la segmentación sin suponer un gran tiempo de espera para el usuario. Un caso que requeriría de esta repetitividad sería para objetos complejos o con las partes adyacentes homogéneas. Las estrategias para la interacción del usuario se pueden agrupar en tres categorías [45] en las que ahondaremos en la sección 2.5.1:

- Basadas en características.
- Basadas en contornos.
- Basadas en regiones.

## 2.4 Métodos Bottom-Up de Segmentación Automática.

En esta sección se describe una selección de métodos bottom-up de segmentación automática de imágenes fijas y heterogéneas, por lo general a color, en los que no se utiliza a priori ningún conocimiento acerca de la escena representada. Como ya se ha comentado, la literatura en el ámbito de la segmentación es enorme por lo que echando la mirada atrás podemos encontrar muchas metodologías utilizadas para hacer frente a dicho problema y también en este campo concreto.

Entre todas ellas, citaremos algunos ejemplos agrupándolos en: *histogram thresholding* [46]; agrupación (por ejemplo *k-means* [47] o *Mean-shift* [27]); morfología matemática [48]; algoritmos de teoría de grafos [49, 50]; métodos estadísticos [51, 52]; basados en los bordes [53]; redes neuronales [54] y muchos otros.

Para concretar en algunos de ellos en esta sección examinaremos las tres categorías sin duda más importantes en el contexto de la segmentación de imágenes basada en la recuperación de contenido: agrupación, morfología matemática y algoritmos de teoría de grafos

### 2.4.1 Agrupación

Los métodos de agrupación fueron de los primeros enfoques abordados en la segmentación de imágenes. Normalmente estos métodos ignoran la información espacial, centrándose únicamente en características como el color o la textura. Estos datos son computados para cada píxel, asignándole a éste un vector multidimensional, con el objetivo de ir agrupándolos en grupos compactos bien diferenciados, midiendo y comparando la distancia entre ellos. Una vez completada la agrupación se mapea la imagen con etiquetas para concluir así con las áreas finales. El método más utilizado de esta categoría es el Algoritmo *K-Means* [47] y sus equivalentes, algo más difusos (*Fuzzy C-Means*).

Algunos de los principales inconvenientes de estos métodos se listan a continuación:

- Requieren el conocimiento a priori del número de grupos.
- Producen regiones inconexas.
- Asumen agrupaciones aproximadamente esféricas (para los cálculos de las distancias).

Como solución a las dos primeras desventajas surge el algoritmo *K-Means-with-Connectivity-Constrain (KMCC)* [7], que se describe con todo detalle en el **Capítulo 3**.

Otros enfoques de esta categoría se basan en modelos paramétricos de la función de la distribución de las diferentes características. El enfoque más conocido de esta categoría es el ya comentado *Blobworld* [6]. En este enfoque, la distribución conjunta del color, textura, y la

posición se modela con una mezcla de Gaussianas. Los parámetros de este modelo se calculan utilizando el algoritmo de Máxima Esperanza (EM) y la segmentación final se define sobre la base de los píxeles que forman el área, es decir, también se asigna una etiqueta a cada píxel correspondiente al área con la que exista una mayor similitud.

Debido a la complejidad real de las características espaciales, no todas las áreas podrán tener una misma forma. La utilización de varios modelos implica conocer de antemano el número de objetos a identificar, lo que últimamente ha hecho ganar popularidad a los métodos basados en la estimación de la densidad sin parámetros.

En todos los métodos de esta categoría, los centros de cada zona se identifican automáticamente, según la hipótesis de que las regiones más densas se corresponden con los máximos de la función de densidad de probabilidad (llamados modos). Cada grupo por tanto quedará asociado a uno de estos modos (máximos).

Una de las técnicas más poderosas para la detección de los modos de densidad es el método Mean-Shift redescubierto en [27], tras haber sido olvidado durante más de veinte años. En este método, se inicializa en cada píxel unos núcleos (o ventanas) que avanzan consecutivamente en la dirección de máximo incremento de la densidad (es decir, hacia la región donde residen la mayoría de los puntos característicos) hasta alcanzar los máximos locales (picos) de la densidad que conducen a la identificación de los modos (el grupo de centros). Se debe tener muy en cuenta cómo las *mesetas de densidad* pueden hacer que se detenga el proceso, lo que provocaría una acumulación de los modos cuya separación sea menor que el ancho de la ventana del núcleo.

Posteriormente cada píxel se asocia con un modo, que será representativo de la densidad y estará localizado en su región propia. Aunque la técnica fue originalmente diseñada para la segmentación de imágenes sólo mediante el análisis del color, más recientemente, también se ha aplicado con éxito basada en otras características espaciales [27]. Una propiedad interesante de éste enfoque es que es capaz de crear, y con gran acierto, amplios gradientes de regiones que en cambio suelen ser sobre-segmentadas (over-segmented) por otros métodos de la misma categoría.

La capacidad y el rendimiento de esta técnica han alentado a muchos investigadores a utilizarla en diversas aplicaciones tales como la recuperación de imágenes [15], la codificación de vídeo MPEG-4 basada en objetos [55] y la detección y reconocimiento de formas [56].

#### **2.4.2 Morfología matemática**

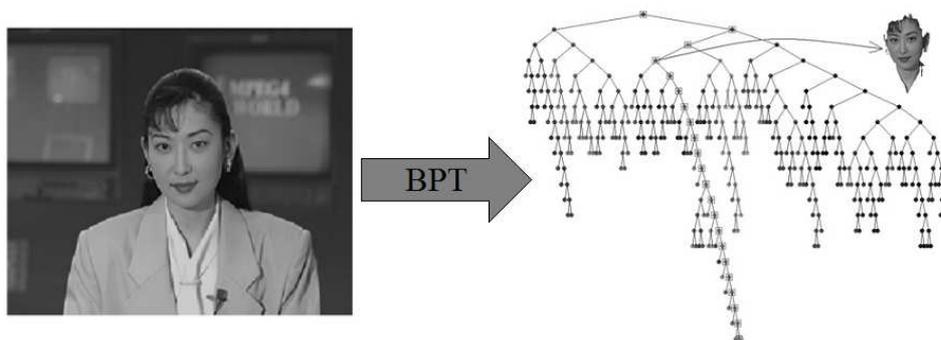
Uno de los métodos más destacados de esta categoría es la *watershed transform* [48], que asocia cada región con un mínimo del gradiente de la imagen. Su aplicación implica típicamente estas tres etapas:

- (i) **Pre-procesamiento** - la imagen es filtrada con el fin de reducir la presencia de ruido.
- (ii) **Extracción de marcadores** - se extraen unos puntos de partida para el proceso de inundación, por ejemplo, se tomarán pequeñas regiones homogéneas.
- (iii) **Watershed transform** - la superficie del gradiente de una imagen es inundada por las fuentes ubicadas en los marcadores de la etapa anterior.

Según aumentan los marcadores, las "presas" se levantan para separar los "lagos" producidos por las diferentes "fuentes". Al final, la segmentación queda definida por el conjunto de "lagos" producidos durante el proceso de inundación. Para reducir la sobre-segmentación, debida a muchos mínimos locales causados por el ruido, otros estudios (por ejemplo [57]) proponen operadores de gradientes multi-escalas.

### 2.4.3 Algoritmos de Teoría de Grafos

En los enfoques de teoría de grafos, las imágenes se representan como grafos ponderados, donde los nodos se corresponden con píxeles o regiones (**ver Figura 2.9**) y en función de la ponderación de los bordes se codifica la información de la segmentación, algunos ejemplos son: *pairwise homogeneity* o *edge strength* [32, 58]. La segmentación se obtiene al dividir (cortar) el grafo por el nivel adecuado, por lo que la elección del criterio óptimo será un punto importante en el diseño de estos métodos.



**Figura 2.9** – Ejemplo de un Binary Partition Tree (BPT) presentado en [59], sobre la imagen de una presentadora.

Varios enfoques de esta categoría hacen uso de los grafos de proximidad entre regiones, más conocidos por sus siglas en inglés RAG (*Region Adjacency Graphs*), que representan mediante nodos cada región homogénea, mientras que el valor de las uniones muestra el grado de similitud entre dichas regiones, es decir, su probabilidad de fusión. Estas medidas de similitud se calculan mediante las distancias entre los colores de las regiones. Una característica

interesante en los RAGs es que pueden simplificarse mediante sucesivas fusiones de regiones vecinas. Sin embargo, estas operaciones tienen que ser guiadas por información global a fin de evitar errores.

Una representación muy útil que surgió a raíz de los RAG fue el *Recursive Shortest Spanning Tree* (RSST) propuesto por Morris et al. en [49] y que ha sido aplicado a la segmentación de imágenes y detección de bordes. El enfoque se amplió posteriormente a las imágenes a color y se mejoró con una versión más rápida propuesta en [60] y un algoritmo equivalente a éste se estudia ampliamente en el trabajo de Salembier [50], que se utiliza para la creación de *Binary Partition Trees* (una estructura jerárquica que ha demostrado ser útil en imágenes para el filtrado, extracción, segmentación y codificación).

A pesar de que todos estos métodos suelen ser computacionalmente eficientes, sus criterios de fusión se basan en propiedades locales del grafo lo que puede llevar a soluciones subóptimas en lo que se refiere a extraer una impresión global de la escena.

Para resolver este problema, se propone en [32] un nuevo criterio global para medir la conveniencia de una partición de imagen, llamado *Normalized Cut*, del que ya hemos comentado ciertos detalles. En él, los píxeles se representan en un grafo no dirigido donde todos los nodos están conectados y la medida en cada uno de ellos es una función de la similitud entre las distintas regiones, basada en características tanto de intensidad como espaciales. El grafo se divide en dos grupos disjuntos, hallando previamente el valor mínimo del *normalized cut*, definido como el coste de la partición (peso total de los bordes eliminados) o como una fracción del total de las posibles conexiones entre todos los nodos en el grafo. Esta optimización del parámetro se puede realizar de manera eficiente resolviéndolo como un problema de autovalores. Los auto-vectores se utilizan para dividir la imagen en dos regiones disjuntas y si se desea, el proceso se continúa recursivamente. En los últimos años esta técnica ha atraído una atención considerable en la comunidad CBIR debido a su alto rendimiento.

## 2.5 Métodos Top-Down de Segmentación.

Partiremos del conocimiento generalizado de que la segmentación de imágenes completamente automática y las técnicas basadas únicamente en el procesamiento a bajo nivel fallan a menudo, no sólo debido a la complejidad del mundo real - ténganse en cuenta ruidos, iluminaciones, reflejos, sombras y variabilidad del color dentro de los objetos, etc. -, sino también debido a que la interpretación de la escena es excesivamente dependiente del contexto en el que ésta se sitúe.

Estas limitaciones de los métodos *Top-Down* puede aliviarse introduciendo en el proceso un conocimiento previo del mundo o la interacción de los usuarios que facilitarán el problema de la segmentación. Como ya se ha explicado anteriormente, el conocimiento previo puede ser de carácter general (por ejemplo, suavidad de los bordes) o específico de la aplicación (por ejemplo, modelos de los objetos buscados).

En esta sección se examinan brevemente varios enfoques seleccionados de segmentación no automática y de detección de objetos basada en modelos.

### 2.5.1 Segmentación Semiautomática

En el pasado se han propuesto múltiples estrategias que cuentan con la interacción del usuario para la extracción de objetos en imágenes. Éstas pueden agruparse en tres categorías [45]:

- i) Basada en características.
- ii) Basada en los contornos/bordes.
- iii) Basada en las regiones.

Cada una de estas categorías requerirá un método distinto de segmentación automática como comprobaremos a continuación.

#### 2.5.1.1 Segmentación Semiautomática Basada en Características.

Estos métodos basados en características de la imagen permiten al usuario dibujar sobre la escena. Generalmente, esto se hace mediante la colocación de unos marcadores que pueden consistir en uno o varios puntos (a veces incluso un “garabato” marcado con el ratón) que se etiquetan según el objeto al que pertenezcan. Varios marcadores (o garabatos) diferentes pueden tener la misma etiqueta, lo que indicará diferentes partes del mismo objeto.

Probablemente el ejemplo más conocido de un enfoque de este tipo es la técnica de *Seeded Region Growing* (SRG) [61]. Este método es controlado por un pequeño número de píxeles, que se conocen como semillas, y que son seleccionados por el usuario de  $K$  conjuntos.

El proceso se desarrolla intuitivamente etiquetando todos los píxeles a partir de estas semillas. En cada paso se etiqueta únicamente un píxel y será aquel que tenga el color más parecido al color medio de sus conjuntos adyacentes, es decir, se agregará al más similar de ellos. Este proceso se repite hasta que todos los píxeles se asignan a alguna región. El método funciona bien en imágenes ruidosas pero es muy sensible a cómo se inicialicen las semillas (el tamaño y la posición de las semillas pueden afectar al color medio inicial y con ello a las iteraciones consecutivas) y en algunos casos da como resultado unos bordes muy irregulares.

Por otro lado, el enfoque ha demostrado una escasa eficacia para los casos en los que la región de interés se compone de más de una subregión homogénea. En la propia publicación, se afirma que el método requiere de un pre-procesado de la imagen en los casos en los que exista variación de la luz y puede incluso considerarse no aplicable para imágenes altamente texturizadas.

Otro enfoque que se apoya en el dibujo de *garabatos* sobre la imagen fue propuesto por Chalom y Bove en [62] y mejorado posteriormente por O'Connor [63]. Ambos enfoques modelan la función de distribución de probabilidad de cada característica (color, textura y localización) parametrizándolas como una suma de distribuciones Gaussianas. La segmentación se realiza mediante el algoritmo de Máxima Esperanza (EM).

Los principales inconvenientes de todos los métodos anteriores son la necesidad adicional de estimar un número de modos para cada objeto y el número relativamente alto de interacciones del usuario necesarias. Una elegante y eficaz solución se describe en [10], donde la imagen de entrada se pre-segmenta automáticamente y se representa en un *Binary Partition Tree* (BPT) para permitir de este modo una rápida respuesta a las acciones de los usuarios. Mediante el BPT se codifican las similitudes entre las regiones obtenidas en la pre-segmentación automática. Esta estructura se utiliza para propagar muy rápidamente las etiquetas de los marcadores de los que hemos hablado anteriormente, que son proporcionados por el usuario generalmente en las grandes zonas (regiones) de la imagen.

Los marcadores establecidos por el usuario suponen fuertes imposiciones en la segmentación mientras que existen otras limitaciones mucho más suaves que además incorporan información tanto de bordes como de regiones. La importancia relativa entre las propiedades de los bordes y de las regiones puede ser controlada por el usuario. Sorprendentemente, el método es válido para imágenes n-dimensionales, por ejemplo, imágenes 3-D o secuencias de vídeo.

### **2.5.1.2 Segmentación Semiautomática Basada en Bordes**

Los métodos basados en bordes o fronteras suelen requerir una inicialización del contorno principal cercano al objeto o la señalización de ciertos puntos clave en el interior de la región. Sin embargo en algunas aplicaciones, por ejemplo, de análisis de imágenes médicas [64], la posición aproximada del objeto que se conoce a priori y la inicialización del contorno se

puede lograr automáticamente. Normalmente, la segmentación se basa en el *paradigma del modelo deformable de contornos activos* (a menudo llamados *Serpientes* o *Métodos Variacionales*), propuesto como primicia por Kass et. al. en [23] como ya vimos en el capítulo 2.3.3. En estos enfoques, los modelos incorporan el conocimiento previo acerca de la suavidad de los contornos y la resistencia a las deformaciones. Otros estudios demuestran cómo se puede reducir la sensibilidad de éstos a los parámetros de inicialización añadiendo una fuerza "interna" que imite el contorno.

Estos primeros modelos de contornos activos se llevaron a cabo con modelos paramétricos que utilizaban representaciones de las curvas envolventes. Uno de los principales avances en el área fue la introducción de los *Geodesic Active Contours* [65] que son modelos de parametrización libre. En estos métodos la segmentación implica la resolución de un problema de minimización de energías mediante la computación de las curvas de mínima distancia. El contorno envolvente se considera el nivel cero de una superficie de dimensiones superiores (con valores positivos y negativos).

Aunque más costoso computacionalmente que los métodos paramétricos, la ventaja implícita es que permite manipular automáticamente los cambios de contorno de las formas, es decir, dividir y fusionar, lo que a su vez permite la detección simultánea de varios objetos. *Geodesic Active Contours* es menos sensible que sus homólogos (métodos paramétricos) a la inicialización puesto que requiere sólo la colocación de la primera curva completamente dentro o fuera de los límites del objeto.

### **2.5.1.3 Segmentación Semiautomática Basada en Regiones:**

Recientemente se ha ampliado la formulación de los conocidos contornos activos para posibilitar la utilización de múltiples señales, introduciendo así el término región. Esto ha concluido en los modelos de regiones activas que ofrecen una mayor solidez frente a las inicializaciones y el ruido.

Por ejemplo, existen segmentaciones bajo supervisión, guiadas por una curva de evolución de varios términos:

- (i) Basados en los bordes (geometría).
- (ii) Basados en regiones (estadística).

Cabe señalar, para terminar, que los enfoques variacionales también se pueden utilizar para segmentación automática. Por ejemplo, en el enfoque propuesto en [66], la primera región de semillas se esparce al azar por toda la imagen ayudada posteriormente de la curva de evolución de Bayes o criterios *Minimum Description Length* (MDL). En otros métodos más complejos, se propone una segmentación automática de texturas basada en la segmentación geodésica de contornos activos en la que las regiones y los parámetros de las distribuciones se

actualizan simultáneamente. De este modo, se produce un aprendizaje dinámico de los modelos de cada región.

### **2.5.2 Segmentación basada en las formas**

En este apartado se tratan algunos de los análisis más interesantes basados en formas que se pueden encontrar en la literatura. Esta característica, por su naturaleza y su fuerte vinculación con el mundo real y con la propia naturaleza de los objetos (en la que se basa) abre un nuevo campo de investigación, tan amplio como interesante. Al mismo tiempo implica la definición de unos modelos que como veremos variaran en el grado de dependencia con la aplicación. Las subsecciones serán las siguientes:

- i) Modelos de formas.
- ii) Modelos estadísticos de formas.

#### **2.5.2.1 Modelos de formas**

Si bien los métodos basados en contornos activos incorporan un conocimiento a priori, relativamente general, como hemos visto suelen necesitar de una cuidadosa inicialización de los modelos en los bordes o de la colocación de ciertas semillas en el interior de las regiones.

En algunas aplicaciones es posible recurrir a modelos mucho más específicos, reduciendo así las ambigüedades y obteniendo técnicas totalmente automáticas de segmentación, detección y reconocimiento de objetos. Esta sección se centra en los modelos de formas y sus distintas aplicaciones. Éstos pueden dividirse en dos categorías:

- (i) Modelos para fines concretos: diseños pensados y elaborados cuidadosamente ("a mano") para un problema muy particular [67].
- (ii) Modelos provenientes de técnicas semiautomáticas entrenados mediante ejemplos de análisis estadísticos [68].

Posiblemente la técnica más conocida que utiliza modelos estadísticos de formas es la conocida como *Point Distribution Model* (PDM) (también llamada *Active Shape Models* (ASMs) o "*Smart Snakes*"), que fue propuesta por Cootes y Taylor [38, 68]. El modelo se obtiene mediante un examen estadístico de las coordenadas de los puntos etiquetados durante el entrenamiento con la transformada de Karhunen-Loeve. Este proceso devuelve la posición media de los puntos, y una descripción de los principales modos de variación existentes. El enfoque se amplió posteriormente a los modelos capaces de sintetizar nuevas imágenes de los objetos de interés, denominados *Active Appearance Models* (AAMs) [69]. Otros enfoques similares incluyen los métodos propuestos por Lades et al. [70] (donde la forma y escala de grises se modela usando filtros de Gabor) o Grenander et al. [71] (cuya aplicación a la segmentación de imágenes parece complicada en principio).

### 2.5.2.2 Utilidad de los Modelos Estadísticos de Formas

Existen una gran variedad de métodos que podrían ser utilizados para modelos estadísticos de formas en el proceso de segmentación [39, 40, 41, 42]. Sin embargo, todos ellos precisan de un paso de optimización para la búsqueda de aquellos parámetros que hagan encajar mejor el modelo, con las características de la imagen (o hipotético grupo de regiones de la imagen).

El más sencillo de todos consiste en adaptar diferentes instancias del objeto esperado a la imagen. Podemos encontrar un ejemplo, en el enfoque propuesto por Cootes et al. [38, 68]. En él, se utilizan distintos conjuntos de parámetros para generar las correspondientes instancias de las diversas formas mediante el modelo *Point Distribution Model* (PDM). Éstas serán después proyectadas sobre la imagen y el conjunto de parámetros que minimice los costes de adaptación es seleccionado como el mejor para interpretar la forma del objeto.

Algunas de las limitaciones de la variabilidad de los contornos son la suavidad y la fidelidad con la imagen segmentada, a parte de la compatibilidad con la forma esperada. Pero sin embargo, la principal dificultad que podemos encontrar en estos enfoques es la necesidad de tener en cuenta las posibles transformaciones entre la colocación del modelo y los objetos presentes en la imagen segmentada.

En [39] encontramos un interesante ejemplo en el que la información de la forma se incorpora durante la evolución del proceso (*Geodesic Active Contours*). Como ya hemos descrito anteriormente, se inserta una primera curva que se establece como el nivel cero de una superficie de dimensiones superiores (en este caso función de la distancia). El entrenamiento consiste en un conjunto de superficies sobre las que se diseña un modelo de una determinada forma. Éstas poseen unas distribuciones similares a las del primer enfoque del que hemos hablado ([38]) para la construcción de PDMs y en cada paso, se estima el máximo a posteriori (MAP) de la posición y la forma del objeto mediante el gradiente de la imagen y el modelo del objeto. Por último, se desarrolla un contorno activo en base al gradiente, la curvatura, la forma y la colocación o postura, según la dirección de máxima estimación a posteriori.

Los modelos estadísticos de formas también pueden ser utilizados como apoyo en los procesos de agrupación de regiones. Uno de ellos, propuesto por Sclaroff y Liu [72], consiste en las siguientes dos etapas:

- (i) Una primera sobre-segmentación, utilizando el algoritmo Mean-Shift para los colores de regiones.
- (ii) A continuación, un agrupamiento de regiones guiado por un modelo de *formas deformables*.

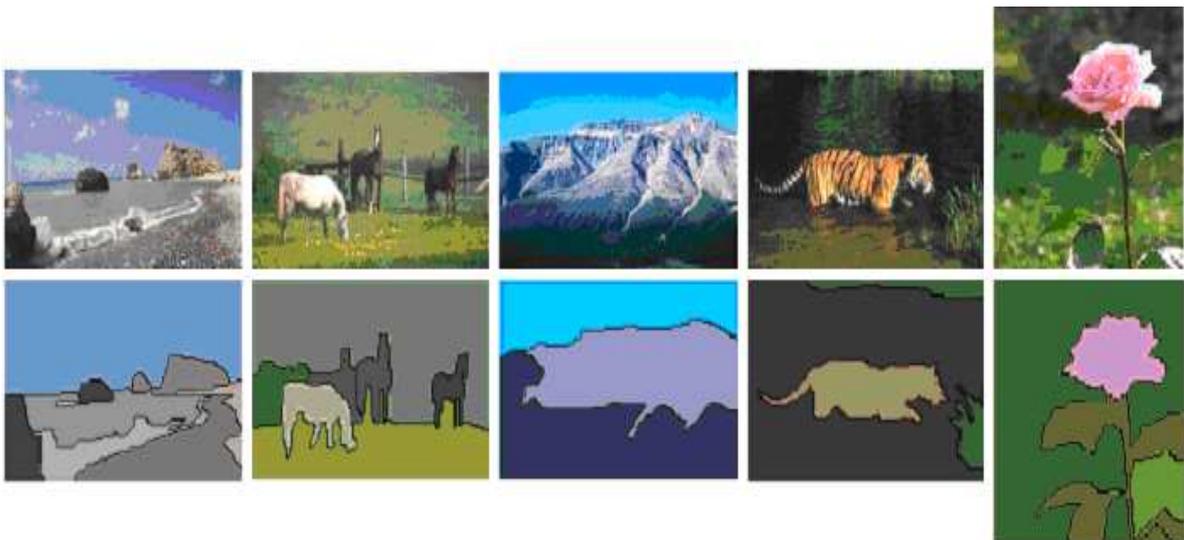
En la segunda etapa, se prueban diversas combinaciones para la agrupación de candidatos adyacentes a cada región. Las regiones candidatas son seleccionadas por sus características de color. El número de hipotéticas fusiones viene limitado por la solidez de los bordes iniciales. Cada posible agrupación se intenta ajustar al modelo, deformando éste y buscando minimizar la función de coste que consta de tres factores:

- (i) Compatibilidad de colores.
- (ii) Área superpuesta entre el modelo y posible agrupación.
- (iii) La deformación.

La división final de la imagen se obtiene mediante la búsqueda de una coherencia global, esto es, una función de coste global con dos términos principales:

- (i) La suma de cada uno de los costes de las distintas hipótesis de agrupación.
- (ii) El número de uniones, minimizado por el uso del algoritmo *Highest Confidence First* (HCF).

El sistema ha demostrado tener una muy buena funcionalidad en imágenes que contienen frutas, hojas, peces, y las células sanguíneas. A pesar de la importante limitación que supone el hecho de que la técnica sólo pueda segmentar objetos de un solo color, la utilización del modelo de forma adecuado permite obtener unos buenos resultados incluso en presencia de sombras, bajo malas condiciones de iluminación y en el caso de objetos superpuestos.



**Figura 2.10** – Ejemplo de imágenes con sus respectivas máscaras debajo [11].

## 2.6 Evaluación

Una buena herramienta de evaluación es imprescindible para el desarrollo de cualquier algoritmo. Sin embargo, en el caso de la segmentación, el tema de su evaluación objetiva se podría decir que ha recibido una muy pequeña atención [73], comparando sobre todo con el amplio número de publicaciones sobre el tema de la segmentación por sí mismo. Esto es bastante sorprendente teniendo en cuenta la necesidad de una herramienta de evaluación objetiva a la hora tanto de implementar nuevos métodos como de comparar los ya existentes.

El objetivo de esta sección es mostrar una pequeña introducción sobre las dificultades que deben ser tenidas en cuenta al comparar distintos resultados y una posterior revisión de los métodos existentes en la literatura. Para evitar una gran extensión nos hemos centrado en los métodos de evaluación de la segmentación de imágenes basada en regiones, aunque muchos de ellos serán válidos igualmente para otros escenarios.

### 2.6.1. Estrategias de evaluación

La práctica actual en el ámbito de la segmentación de regiones justifica nuevos avances proporcionando un conjunto de resultados cualitativos, es decir, presentando ejemplos seleccionados de dichas segmentaciones. Aunque el propio método puede ser suficiente para ilustrar los efectos del procesado de bajo nivel, por ejemplo la detección de bordes, por lo general no proporcionará una demostración satisfactoria en los métodos de segmentación. Esto es especialmente importante en el contexto de los sistemas CBIR que funcionan con contenido general. Otra alternativa es una evaluación subjetiva “*ad-hoc*” con un determinado grupo representativo de espectadores. Idealmente debería reducirse al mínimo la subjetividad frente a las condiciones estrictas de evaluación.

Por ejemplo, [73] sugiere seguir las recomendaciones para la evaluación de la calidad de video desarrollado por la UIT [74]. Sin embargo, este proceso requiere un número significativo de evaluadores y un excesivo consumo de tiempo. Debido a su alto coste, la evaluación rara vez es utilizada.

Los estudios presentados en [75], consistentes en experimentos subjetivos, pueden considerarse la base para una mejor comprensión de la percepción humana de la segmentación. Sin embargo, la segmentación es rara vez el objetivo final en cualquier aplicación sino más bien (por ejemplo, en CBIR) un paso hacia la comprensión de la escena. Por todo esto, la principal dificultad en la evaluación de la segmentación, radica en establecer un criterio adecuado para una aplicación determinada. Debería ser posible realizar una evaluación objetiva del algoritmo de particionamiento, midiendo el rendimiento global del sistema en el que se integra pero desafortunadamente, en el caso de los sistemas de CBIR, este enfoque se vuelve a menudo poco práctico debido a su complejidad. Por lo general, el rendimiento del sistema depende de las

combinaciones de parámetros del sistema y del propio algoritmo de segmentación. Esto hace que sea muy difícil garantizar unas condiciones óptimas de evaluación.

En los últimos años, se ha hecho evidente que el elemento clave de estos sistemas es la interacción entre usuarios, y el pertinente feedback. Pero de nuevo nos encontramos la dificultad de tener en cuenta esos factores tanto en amplios experimentos de evaluación, como en los requeridos para desarrollos de nuevos métodos de segmentación.

Puesto que las tres estrategias anteriores son descartadas para evaluar experimentos con una gran extensión de muestras, a continuación se describen otros métodos de evaluación automática con medidas capaces de evaluar adecuadamente la calidad de la segmentación.

Existen dos clases principales de métodos de evaluación automática:

- Independiente / Autónomo.
- Relativo / Dependiente.

El primero se suele aplicar sólo en los casos en los que la segmentación sin referencias no es posible. Aprovecha los conocimientos disponibles a cerca de las propiedades esperadas en la segmentación deseada para una aplicación en particular. La principal dificultad radica en establecer una medida de evaluación adecuada para el tipo de contenido y la aplicación a la que se dirige. Los resultados de dicha evaluación no se corresponderán necesariamente con la percepción humana de la calidad de segmentación [76], por lo que obviaremos este escenario. No por ello dejaremos de señalar que la evaluación independiente es fundamental para la propia segmentación. Como prueba de ello, hay estudios en los que existe cierta analogía entre las características propuestas para la evaluación independiente [77] y las propiedades geométricas de las regiones establecidas en el esquema de segmentación.

Por otro lado, los métodos relativos de evaluación basados en medidas de similitud, se utilizan para cuantificar las diferencias entre la evaluación y una segmentación de referencia a menudo denominada *ground-truth* (ver **Figura 2.10**). Este tipo de evaluación es mucho más fiable que la anterior y su dificultad radica en conseguir grandes colecciones de datos de prueba con su correspondiente *ground-truth*.

Se debe reconocer también que depender de una sola máscara de referencia para cada imagen puede ser cuestionable, especialmente en el contexto de CBIR. La calidad de la división es dependiente del usuario y de las consultas realizadas, que normalmente no son conocidas con anterioridad y una colección de *ground-truth* con máscaras simples para cada imagen, no tiene en cuenta estas ambigüedades en la segmentación de la imagen.

A pesar de todo esto, los métodos relativos de evaluación, representan la mejor relación entre fiabilidad de la evaluación (valor de la información proporcionada) y la viabilidad de la evaluación de experimentos extensos necesarios, como suele ser el caso. Por ello, la discusión que se presenta en adelante, se limita a los métodos de evaluación relativa.

### 2.6.2 Métodos de Evaluación Relativa

Una de las mayores dificultades en la evaluación objetiva es encontrar una magnitud capaz de medir la calidad de la segmentación con una escala similar al modo de percepción de calidad humano - se pueden encontrar interesantes discusiones sobre esto en las publicaciones nombradas anteriormente [73, 75] y que comentaremos en esta sección.

Ya que ninguno de los métodos propuestos en la literatura parece ser aceptado por un conjunto mayoritario de la comunidad investigadora, en esta sección se desarrolla un breve repaso de los experimentos más importantes. Podemos encontrar una revisión más amplia en [76].

Uno de los primeros métodos aplicables en segmentación de video fue [78]. En él se evalúa la precisión espacial y temporal para escenas compuestas por dos objetos (el objeto a identificar y el fondo). En estudios previos se medía la agudeza espacial en términos de píxeles erróneos, es decir, aquellos que se etiquetan o asignan a una región incorrecta. La ponderación de dichos píxeles desclasificados según su distancia al objeto de referencia se introdujo posteriormente en este método.

En [73] encontramos una metodología un poco más avanzada en la que se identifica un conjunto de características relevantes para ser comparadas y se proponen ciertas medidas objetivas y cuantitativas para ellas. Las categorías de características espaciales propuestas incluyen - **ver Figura 2.11**:

- i) La fidelidad de la forma - calculada de manera similar a la medida de la precisión espacial [78].
- ii) La similitud geométrica - según características geométricas simples como el tamaño, la posición, el alargamiento y la compactación.
- iii) Los bordes - similitudes entre el borde de referencia y el del objeto detectado.
- iv) Los datos estadísticos de similitud - por ejemplo, similitud del brillo y el "enrojecimiento" entre la referencia y el objeto estimado.

Aunque esta lista de características es bastante completa, la utilidad de combinar las distintas medidas en un único criterio de evaluación es cuestionable. La realización de dicha combinación ponderada por los ad-hoc parece poco práctica, puesto que en muchos casos podría ser necesaria la inspección manual de los resultados para determinar qué características contribuyen al error (y de qué manera). A menudo, los estudios se decantan por medidas más simples y más fáciles de entender, como por ejemplo [11].

En [75] encontramos un reciente e interesante estudio sobre el impacto de los cambios de topología, en particular, las regiones y los denominados agujeros (*holes*). En él se llevan a cabo pruebas subjetivas para medir estos factores tanto de forma individual como en combinación. Los experimentos indicaron que, independientemente del contenido, los factores

correspondientes al error de las regiones alcanzan rápidamente el nivel de saturación cuando se tratan tamaños grandes. También se confirma la hipótesis de que la tasa de error en los *holes* o agujeros (grupo de píxeles que no han sido asignados a ninguna región) supone una mayor contribución en comparación con las regiones del mismo tamaño. Otro importante resultado de los experimentos fue la demostración de que diversos tipos de errores topológicos contribuyen de distinta manera al resultado global. Además, las proporciones en las que contribuyen son dependientes del contenido. Por lo tanto, serían necesarios futuros estudios más amplios sobre este tema para poder incorporarlos con cierto fundamento en el cómputo de las cifras de evaluación automática de contenidos generales.

Relativamente simple pero con una gran similitud a la intuición humana, es el método propuesto en [76] y diseñado específicamente para la evaluación de segmentación de imágenes en el contexto de los sistemas de CBIR. Se basa principalmente en el enfoque de [78] para la evaluación de la precisión espacial, pero se extiende a la segmentación de imágenes en las que tanto la máscara de referencia como la segmentada contienen múltiples regiones. Se dedica una especial consideración no sólo a la exactitud de la localización y los bordes, sino también a las sobre- y sub-segmentación.

Para abordar estas cuestiones, la evaluación comienza estableciendo una correspondencia única entre las regiones de referencia y las máscaras evaluadas. Se tienen en cuenta tres tipos de errores:

- i) Errores de exactitud en la asociación entre las regiones de referencia y las evaluadas.
- ii) Errores de sub-segmentación calculada en base a las regiones no asociadas de la máscara de referencia.
- iii) Errores debidos a la sobre-segmentación calculada en base a las regiones no asociadas de la máscara evaluada, es decir, el resultado de la segmentación propiamente dicho.

A pesar de la simplicidad del método, un conjunto de resultados bastante convincentes presentados en [76] dejan ver como se asemeja a la evaluación humana. Una de las ventajas añadidas que ofrece es el reducido número de parámetros que intervienen y la facilidad de interpretación.

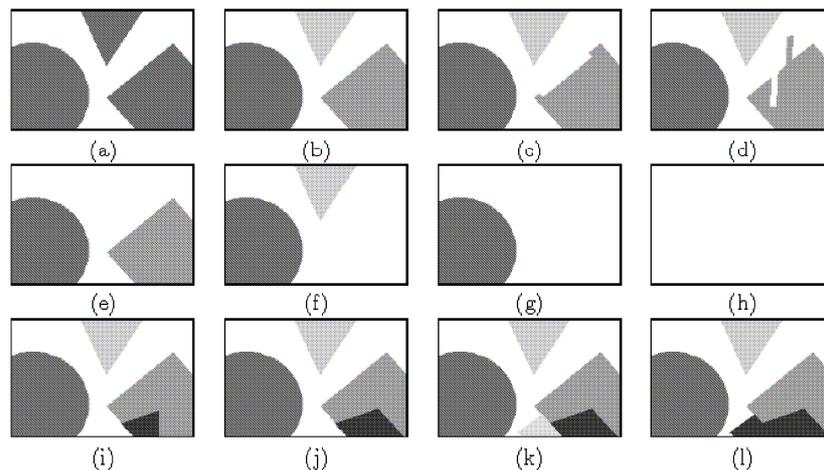
Por último, cabe señalar la alta sensibilidad de los humanos ante errores en los objetos con alto valor semántico. Por ello, en todos estos estudios sobre evaluación se reconoce la importancia de la trascendencia del objeto. Por ejemplo, cierta medida de relevancia indicará las probabilidades de que un objeto en particular sea más manipulable y utilizado [79]. Sin embargo, hoy por hoy, debido a las dificultades para determinar estos niveles de importancia durante la segmentación, rara vez se tiene en cuenta durante la evaluación.

## 2.7 Conclusión

Tras el resumen histórico sobre los distintos métodos, avances y visiones en el campo de la segmentación de imágenes, en este apartado se muestra una reflexión global sobre todo ello.

Desafortunadamente, y a pesar de las décadas dedicadas al estudio e investigación de la segmentación de imágenes, a día de hoy existe un gran número de aplicaciones cuyos resultados muestran como un reto, aún por alcanzar, la resolución satisfactoria de este problema. Aunque, como hemos visto a lo largo de este capítulo, existen escenarios en los que se ha logrado obtener muy buenos resultados, muchos investigadores se muestran escépticos ante el objetivo de lograr un método de segmentación con resultados fiables. Este punto de vista cobra sentido si tenemos en cuenta la idea, generalmente aceptada, de que el contenido relevante de una imagen es dependiente del usuario y de sus necesidades, lo que implicaría una interacción por parte de éste. Consecuentemente con esto, una segmentación completamente automática partiría de una mala definición del “contenido de interés”. A pesar de todo, existen trabajos que aún teniendo en cuenta estos aspectos tratan y analizan el tema desde otras perspectivas, como es el caso de ([11]) centrado en el contexto de los sistemas CBIR.

En este capítulo se han esbozado los aspectos más importantes, y se han descrito por encima los métodos más extendidos. Además se ha hecho hincapié en la definición todos aquellos términos, conocimientos y métodos necesarios para situar al lector en un contexto apto y posibilitar así una fácil comprensión de los siguientes apartados.



**Figura 2.11** – Conjunto de imágenes sintéticas, utilizado en [76] para la evaluación numérica de los experimentos. La imagen sintética original y su máscara de referencia se muestran en (a) y (b), respectivamente. Las siguientes ilustran distintos tipos de errores típicos en segmentación: la detección incorrecta de bordes en (c) y (d), sub-segmentación en (e) - (h), y sobre-segmentación (i) - (l).



### 3 - Análisis y evaluación del KMCC-Texturizado





## 3 – Análisis y Evaluación del KMCC-Texturizado

En este capítulo se expone el método de segmentación denominado KMCC-texturizado [7] y que será objeto de mejora en este proyecto fin de carrera. Éste está basado en el *K-Means-with-connectivity-constraint* (KMCC) [80], que a su vez es una versión del algoritmo original: *K-means* [81].

### 3.1 – Introducción

En esta sección se introduce un algoritmo de segmentación de imágenes a color, cuyo objetivo es obtener los objetos semánticos para su posterior utilización en aplicaciones multimedia. En el método expuesto [7] se combina el análisis de características de color, textura y posición espacial, con el fin de conectar regiones en iteraciones consecutivas y extraer así los objetos finales. Para optimizar el resultado y conseguir que tras múltiples uniones resulten regiones compactas y sin fisuras, se realiza un pre-procesamiento mediante un filtrado. Para la clasificación de los píxeles se hace uso del algoritmo KMCC [80], con una modificación adicional que asegura la conectividad de las áreas. Gracias a un proceso inicial de agrupación automática el algoritmo puede funcionar sin supervisión alguna.

Veremos cómo este método está diseñado y es también útil para segmentar imágenes de un tamaño relativamente grande, aplicándolo sobre versiones reducidas de éstas. Así, se obtiene como resultado una burda segmentación que se refina a posteriori mediante una clasificación basada en el teorema de Bayes, con unos resultados finales muy satisfactorios.

#### 3.1.1 – Colección de imágenes utilizada

La colección de imágenes que se evaluará está compuesta por 100 imágenes procedentes de la *Corel Gallery*<sup>1</sup> y otras 20 procedentes de diversas fuentes, tales como cámaras digitales o móviles. Todas ellas fueron segmentadas a mano previamente, para obtener sus respectivas máscaras de referencia (*ground-truth*) que serán imprescindibles a la hora de la evaluación. Tanto las imágenes como las máscaras de referencia han sido cedidas por la DCU - Dublin City University.

Las imágenes procedentes de la *Corel Gallery* están compuestas de objetos fácilmente diferenciables por su color mientras que el resto, fueron tomadas en condiciones de baja luminosidad y contienen objetos de colores más similares. Dadas estas condiciones, las 100 imágenes de la *Corel Gallery* se prevén fáciles de segmentar, únicamente siguiendo el criterio

---

<sup>1</sup> <http://www.corel.com/>

de homogeneidad de color. Esto las convierte en un reto a la hora de añadir cualquier otro criterio diferente.

Para todos los casos posibles (combinaciones de los criterios de homogeneidad) se lleva a cabo una validación cruzada dividiendo la colección en dos subconjuntos A y B, de 60 imágenes cada uno (50 de la *Corel Gallery* y 10 de las diversas fuentes).

### ***3.1.2 – Estructura del capítulo***

Tras este primer apartado introductorio en el punto 3.2 se presenta el algoritmo predecesor y sus evoluciones que serán la base del KMCC-Texturizado del cual se da una visión global en el siguiente apartado, 3.3. Tras una definición previa de la textura y los distintos métodos utilizados para su análisis en el punto 3.4, se describe detalladamente el funcionamiento del método en la sección 3.5. Por último, en el 3.6 se expondrán resultados experimentales con las consecuentes conclusiones en el punto 3.7.

## 3.2 – Antecedentes del KMCC Texturizado.

En este apartado se describe el algoritmo KMCC [80] con el objetivo de introducir el algoritmo descrito posteriormente en este capítulo y basado en el primero. El KMCC es a su vez una versión mejorada del K-means original [81] por lo que comenzaremos describiéndolo brevemente.

### 3.2.1 – Algoritmo de Segmentación K-Means

El algoritmo K-means [81] es un método de división de información, utilizado para diversas aplicaciones en las que se busca como resultado un número determinado de grupos (objetos semánticos, en el caso de la segmentación de imágenes). Otro ejemplo de utilidad de este algoritmo se puede encontrar en el ámbito de la medicina, por ejemplo [82]. Pero volviendo al marco de la segmentación de imágenes, este algoritmo básicamente busca mediante cálculos de distancias iterativos, la formación de *clusters* (grupos) que serán representados por el valor medio de los píxeles que pertenecen a dicho grupo. A continuación se describe el algoritmo Kmeans tradicional.

De acuerdo a la literatura especializada [83, 84, 85, 86, 87, 88] se pueden identificar cuatro pasos en el algoritmo:

- **Paso 1. Inicialización:** Se define el número de grupos (K) y un centroide por cada grupo. Algunas implementaciones del algoritmo estándar determinan los centroides iniciales de forma aleatoria; mientras que algunos otros procesan los datos y determinan los centroides mediante de cálculos.
- **Paso 2. Clasificación:** Para cada píxel de la imagen, se calcula su distancia a todos los centroides, determinando así el más cercano. El píxel es entonces incorporado al grupo o región asociada a dicho centroide.
- **Paso 3. Cálculo de centroides:** Para cada grupo generado en el paso anterior se vuelve a calcular su centroide como el valor medio de todos los píxeles asignados a ese grupo.
- **Paso 4. Condición de convergencia:** El proceso se detiene al darse una cierta condición preestablecida, entre los centroides del paso 1 y del paso 3. En la literatura se encuentran varias condiciones de convergencia, de las cuales las más comunes son las siguientes: cuando se alcanza un número de iteraciones dado; cuando no existe un intercambio de objetos entre los grupos; o cuando la diferencia entre los centroides de dos iteraciones consecutivas es más pequeña que un umbral dado. Si la condición de convergencia no se satisface, se repiten los pasos dos y tres y se comprueba de nuevo hasta que así sea.

A pesar de su facilidad de programación, su buen rendimiento computacional y su demostrada eficacia, el algoritmo K-means tiene algunos inconvenientes:

- El resultado de las regiones finales está muy vinculado a los centroides iniciales.
- La convergencia en el óptimo global no está garantizada.
- A menudo requiere de un gran número de iteraciones para converger. Lo que afecta en gran medida al coste computacional del algoritmo K-means es el número de iteraciones que necesita realizar, ya que por cada iteración calcula la distancia de todos los píxeles de la imagen a cada uno de los centroides.

### 3.2.2 – Algoritmo de Segmentación KMCC

Aunque el K-Means fue un algoritmo rápidamente extendido para la segmentación de imágenes [89], es un hecho probado que el método tiende a producir regiones inconexas. Esto se debe en parte, a que sólo tiene en cuenta la intensidad o información de color de la imagen, haciendo caso omiso a la información espacial de estas intensidades.

Con el fin de solventar este problema, se propuso una modificación del algoritmo K-Means, conocida como: K-Means con limitación de conectividad o sus siglas en inglés KMCC (*K-Means-with-connectivity-constraint*) [80]. En este algoritmo, se tiene en cuenta la proximidad espacial del píxel a cada región y se añade esta característica al cálculo de los centroides. En el KMCC se añade también el uso del espacio de color CIE-LAB y la capacidad de detectar posibles regiones sobre-segmentadas.

Los pasos a seguir son muy similares al K-means con alguna modificación, debida a las mejoras introducidas arriba:

- **Paso 1. Inicialización:** el proceso de inicialización de los K centroides se basa en el análisis del histograma de color. Se calculan los centros ( $I_k$ ) como en el K-means, por las intensidades de color más dispares y además se añaden los centros espaciales ( $S_k$ ):

$$\bar{I}_k = (\bar{I}_{L,k}, \bar{I}_{L,k}, \bar{I}_{L,k}) \quad (3.1)$$

$$\bar{S}_k = (\bar{S}_{k,x}, \bar{S}_{k,y}) \quad (3.2)$$

- **Paso 2. Clasificación:** se calcula la diferencia de color de todos los píxeles  $p = (x,y)$  con cada uno de los centroides  $I_k$ , así como la distancia espacial a los centros  $S_k$ . Ambas distancias están ponderadas por un factor  $\lambda_1$  y  $\lambda_2$ , respectivamente. El píxel  $p$  será asignado al grupo K que minimice la distancia:

$$D(p, k) = \lambda_1 \|I(p) - \bar{I}_k\| + \lambda_2 \cdot \bar{A} \frac{\|p - \bar{I}_k\|}{A_k} \quad (3.3)$$

con

$$\|I(p) - \bar{I}_k\| = \sqrt{(I_L - \bar{I}_{L,K})^2 + (I_a - \bar{I}_{a,K})^2 + (I_b - \bar{I}_{b,K})^2} \quad (3.4)$$

y

$$\bar{A} = \frac{1}{K} \sum_{k=1}^K A_k \quad \text{siendo } A_k = M_k, \text{ el numero de píxeles asignados a } S_k \quad (3.5)$$

En la **Ecuación 3.3** se normaliza el segundo término referido a las distancias espaciales para equipararlas, independientemente de los tamaños de las regiones. Iterativamente, todos los píxeles serán asociados a un grupo según la condición:

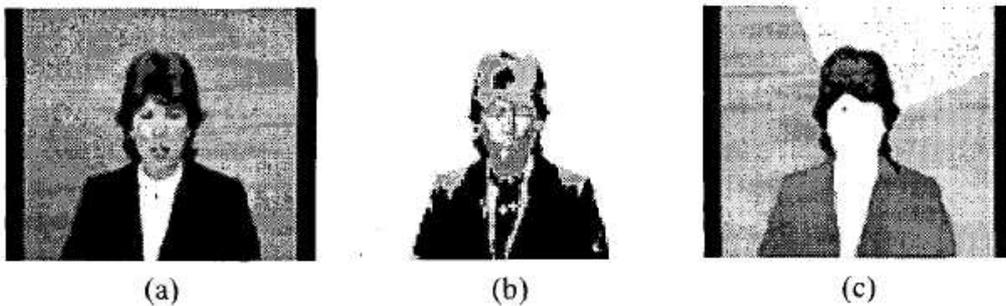
$$\text{Si } |D(p,i)| < |D(p,k)| \quad \forall k \neq i \longrightarrow p = (x, y) \text{ se asigna a la región } R_i \quad (3.6)$$

- **Paso 3. Cálculo de centroides:** Tras el nuevo agrupamiento de píxeles, los centros calculados como las medias de intensidades y de posición variarán. En este paso se recalculan todos los centros de nuevo:

$$\bar{I}_k = \frac{1}{M_k} \sum_{m=1}^{M_k} I(p_m^k) \quad (3.7)$$

$$\bar{S}_{k,x} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,x}^k \quad y \quad \bar{S}_{k,y} = \frac{1}{M_k} \sum_{m=1}^{M_k} p_{m,y}^k \quad (3.8)$$

- **Paso 4. Condición de convergencia:** Si la diferencia entre los centros recién calculados y los anteriores es menor que un cierto umbral se pasa a la última etapa del algoritmo. En caso contrario, se repiten los pasos 2 y 3 hasta que se de la condición de convergencia.
- **Paso 5. Limitación de conectividad.** A pesar de la mejora incorporada mediante los centros espaciales, de nuevo debido a la conectividad limitada, una región uniforme puede quedar segmentada en varias regiones más pequeñas. Mediante algoritmo muy simple de fusión de regiones [80] se extraen las últimas regiones.



**Figura 3.1** – Ejemplo de segmentación en una imagen real (a) para comparar el K-means (b), y su versión mejorada KMCC (c). [80]

Tras ver como en el KMCC se añade el concepto de posición espacial en el método K-means original (ver resultados comparativos en la **Figura 3.1**), en los siguientes apartados veremos el algoritmo KMCC-Texturizado que consiste en una versión del KMCC a la que se le añade el análisis de texturas.

### 3.3 – Una Visión Global del KMCC-Texturizado

En esta variante se presenta el uso de características de textura en combinación con la intensidad y la posición del píxel, que junto con un filtrado de la textura que depende de la intensidad de píxel (filtrado condicional), dota al algoritmo de la capacidad de manejar áreas texturizadas eficazmente y evita dividir los objetos cromáticamente no uniformes en un gran número de regiones. En el KMCC-Texturizado los valores iniciales requeridos se calculan utilizando un procedimiento inicial de agrupación que consiste en dividir la imagen en bloques y asignar un vector de intensidad y textura a cada uno de ellos. Esta agrupación automatizada hace innecesaria la intervención del usuario, lo que facilita el tratamiento de grandes colecciones de imágenes.

Aunque este algoritmo de segmentación es muy rápido cuando se aplica a imágenes de dimensiones relativamente pequeñas, su eficacia disminuye rápidamente cuando las dimensiones de la imagen aumentan. Esto ocurre prácticamente en cualquier algoritmo de segmentación dado que la complejidad computacional es proporcional al número de píxeles de la imagen segmentada. En los últimos tiempos las imágenes de mayores dimensiones se han hecho cada vez más populares, en parte como consecuencia de los avances en el almacenamiento y las tecnologías de la comunicación, lo que hace esencial la existencia de métodos de segmentación eficientes en cuanto al tiempo de ejecución. Por esta razón, junto con este método de análisis de texturas se propone un método para la optimizar el tiempo de segmentación de imágenes relativamente grandes [7], utilizando el Teorema de Bayes.

El algoritmo de segmentación general consta de las siguientes etapas (**ver Figura 3.4**):

- **Etapas 1.** Extracción de *la intensidad de color y la textura* característica de los vectores correspondientes a cada píxel o bloque. Estas se utilizarán junto con las características espaciales en las siguientes etapas.
- **Etapas 2.** Estimación de la cantidad inicial de las regiones y sus centros espaciales, de intensidad y de textura, utilizando el *procedimiento automático de agrupación*. Estos valores serán utilizados por el algoritmo KMCC.
- **Etapas 3.** *Filtrado Condicional*.
- **Etapas 4.** Clasificación final de los píxeles, utilizando el algoritmo *KMCC*.

El resultado de la aplicación del algoritmo de segmentación sobre una imagen en color es una máscara, es decir, una imagen en escala de grises en la que los diferentes valores de gris corresponden a las diferentes regiones formadas por el algoritmo KMCC. De esta forma, podrán compararse y evaluarse con las máscaras de referencia.

### 3.4 - Homogeneidad de Textura

En este apartado se trata la textura desde su definición más abstracta hasta la más técnica (orientada al tratamiento de imágenes), repasando los recursos que han sido utilizados para su análisis y tratamiento. Además, en el último subapartado se hace especial hincapié en el método de transformada de wavelet utilizado por el algoritmo KMCC-texturizado.

#### 3.4.1 - ¿Qué es la textura?

A menudo se puede oír sobre la textura: “sé lo que es pero no podría definirla”. Una definición poco convencional será [90]: la textura es lo que da belleza a la vida. Y es que la textura es lo que hace que podamos diferenciar la música de Mozart y que una obra maestra del arte renacentista sea atractiva para nuestra vista. Es la pequeña variación de los detalles que nos hace sobrellevar el día a día, como por ejemplo la rugosidad de la textura del suelo que nos permite caminar. Si las superficies fueran lisas, los lápices no escribirían, los coches no andarían y los pies no nos mantendrían.

La textura nos rodea y nos envuelve, y existe también en todas las imágenes que creamos. Así como las pequeñas variaciones en nuestras rutinas nos permiten distinguir un día de otro en nuestra vida, la textura nos permite diferenciar todo lo que vemos. Por ello no debe ser ignorada por un sistema automático de visión y debería tomar un papel principal en el procesamiento de imágenes y todos sus avances.



**Figura 3.2** – a) Las camisas son objetos texturizados si tratamos de identificar personas. b) La imagen podría ser una parte de una camisa hawaiana pero por la escala de la misma la textura será el fondo marrón y liso, mientras que las flores y hojas serán los objetos identificados.

Una definición más técnica que podemos encontrar en [90] sobre la textura en el campo del tratamiento de imágenes es la siguiente: la textura es la variación de los datos de una imagen en una superficie menor que la escala de interés. Por ejemplo, en la **Figura 3.2a.**- se muestra la

imagen de varias personas vistiendo camisetas hawaianas. Si estamos interesados en identificar rostros, el patrón de la camiseta será considerado como una textura. Sin embargo, si estamos interesados en identificar un detalle de la camisa (ver **Figura 3.2b**) el patrón se considerará un objeto no texturizado o lo que es lo mismo, cambiará la escala de la imagen.

A parte de la propia definición, cabe nombrar algunos aspectos característicos de la textura como son: rugosidad, suavidad, patrones, escala, bordes, su resolución, etc. Algunas de estas propiedades pueden observarse en los ejemplo de texturas de la **Figura 3.3**.

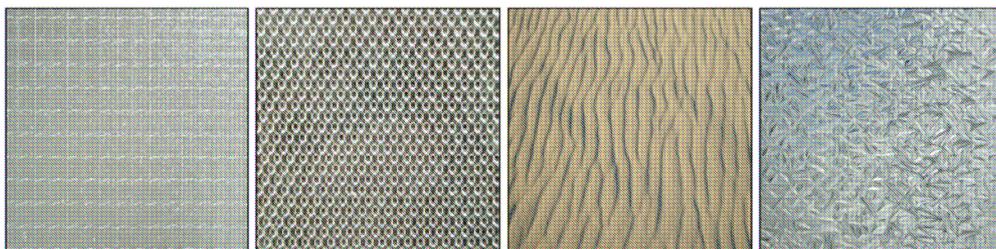
Dos de las principales razones que hacen sumamente importante el tener en cuenta esta característica en el tratamiento de imágenes son las siguientes:

- La textura mal tratada puede convertirse en un inconveniente para un sistema automático de visión. Por ejemplo, si queremos reconocer un objeto por su forma, un mal tratamiento de la textura crearía bordes extras en la imagen del objeto debido a la misma y su forma sería más difícil de reconocer.
- Pero por otro lado, la textura puede ser un dato clave para el reconocimiento de objetos bajo ciertas condiciones puesto que interpreta el material del que éste está hecho. Por ejemplo, en una instantánea aérea podríamos diferenciar fácilmente las zonas agrarias o urbanas en función de la variación de su textura.

### 3.4.2 – Técnicas de Análisis de texturas

El objetivo principal en el análisis de las texturas de una imagen es básicamente extraer las características de las mismas para poder así identificarlas de forma inequívoca y clasificarlas. El objetivo final puede variar en función de la aplicación y algunos ejemplos son: determinar a cuál de un número determinado de clases físicamente definidas pertenece una región de textura homogénea; delimitar en una imagen las zonas con texturas diferentes. En definitiva estará muy ligado a la segmentación de imágenes.

Como se verá más adelante, en la descripción del diseño del método en cuestión, la textura es caracterizada por un conjunto de descriptores extraídos de la misma, que forman un vector. Estos vectores y su consecuente análisis pueden ser de gran utilidad en aplicaciones tales como: teledetección, tratamiento de imágenes médicas, reconocimientos de formas, etc.



**Figura 3.3** – Texturas en las que se pueden observar sus diferentes características. [91]

A pesar de no existir unanimidad respecto al método a utilizar, los autores si coinciden en la agrupación de éstos en 4 tipos principales [91]:

- **Estadísticos.** Basados en el cálculo de descriptores de la distribución de intensidades en la imagen, que son casi constantes para cada textura. Algunos ejemplos desarrollados por Zhang y Tan son: Gráficos polares y polarogramas, métodos óptico-digitales, expansión armónica, etc. [92]
- **Estructurales.** Consisten en entender una textura como una serie de primitivas (microtexturas) y una jerarquía de orden espacial (macro textura) de dichas primitivas. Ejemplos: División en polígonos de Voronoi [93] o histograma invariante y descriptores topológicos [92].
- **Basados en Modelos.** Se apoyan en la construcción de un modelo, cuyos parámetros estimados sobre ciertas imágenes, describen las cualidades de la textura dada. Para el diseño de estos modelos se recurre a métodos de multiresolución autorregresivos [94] y modelos estocásticos [92] entre otros.
- **Basados en Transformadas.** El hecho de tratar una señal desde un dominio distinto del habitual, proporciona ciertas ventajas para su tratamiento ya conocidas en diversos ámbitos. Para el caso del tratamiento de imágenes y en concreto el análisis de texturas se encuentran varios métodos relacionados muy útiles:
  - *Filtros en el dominio espacial:* densidad de bordes, filtros de energía, operadores no lineales, etc.
  - *Transformada de Fourier:* basada en el análisis en el dominio de la frecuencia.
  - *Filtro de Gabor:* partiendo de una transformada de Fourier se aplicará una función gaussiana.
  - *Transformada Wavelet:* nos permite analizar la frecuencia de una imagen en función de la escala.

### 3.4.3 – Discrete Wavelet Frames

En este punto se entra un poco en detalle sobre la Transformada wavelet utilizada en el KMMC-texturizado. Se trata de una Wavelet Frames Discreta (Discrete Wavelet Frames – DWF) [95] y es una transformada muy similar a la Transformada Wavelet Discreta, que proviene a su vez del muestreo de la Transformada Wavelet Continua. Para una ampliación de conocimientos teóricos sobre estas y otras transformadas se recomienda la lectura de [91].

Se pueden dar dos razones básicas para exponer la buena funcionalidad de las transformadas wavelet:

- Permiten variar la resolución espacial y representar texturas de diversas naturalezas.
- Hay un amplio abanico de funciones base wavelet, de tal manera que cada aplicación práctica podrá analizarse de manera muy específica. Se utilizará aquella función que a priori genere mejor resultado.

El término FRAME (traducido al castellano como “marco”) ha sido introducido en la transformada wavelet para expresar una variedad de ella que se basa en su redundancia. Este término hace referencia a un conjunto de vectores no independientes N-dimensionales definidos en un espacio  $R^N: \{f_k\}$ , con  $1 \leq k \leq K$  y  $K \geq N$ . Un frame viene expresado en forma de matriz de  $N \times N$  en la que las columnas son los vectores del frame  $f_k$  [96]. Al ser un conjunto de vectores no independientes pueden ser redundantes, por lo que están asociados con sobremuestreo o redundancia.

El punto de partida de este modelo se sitúa en la discretización de la Transformada Wavelet Continua. Para lograr una reconstrucción perfecta se debe dar que la denominada familia wavelet  $\{\Psi_{m,n}\}$ ,  $m, n \in Z$  pueda reconstruir un frame.

La puesta en práctica de la DWF se realiza como una descomposición wavelets sin remuestreos, mediante el empleo de bancos de filtros, de tal manera que las imágenes resultantes de la aplicación de éstos no estén submuestreadas. Esto provoca una alta redundancia en los resultados que se emplea para crear unas representaciones invariantes a la traslación. Aunque esto último incrementa su dificultad para su empleo en ciertas aplicaciones, por otro lado genera descriptores de textura también invariantes con respecto a la traslación de la imagen de entrada. Ésta es una propiedad bastante deseable en el contexto de imágenes, puesto que puede proporcionar una mejor estimación de las estadísticas de textura y una caracterización más detalladas en las zonas próximas a los bordes.

Además de una descripción matemática mucho más detallada, en [95] se pueden comprobar varios resultados de la comparación entre la DWT y la DWF que resaltan a la segunda como ligeramente superior. Además para la extracción de características, la DWF tiende a disminuir la variabilidad de las características de textura estimadas lo que mejora la clasificación final realizada.

En general, los algoritmos que aplican la DWF se comportan bastante bien en cuanto a la segmentación de texturas se refiere. Por todo ello, parece adecuada la elección de la DWF como herramienta de análisis de texturas en el KMCC-Texturizado.

### 3.5 – Diseño e Integración del Método

En este capítulo se describe con detalle el método denominado KMCC-Texturizado y cada una de sus etapas, descritas efímeramente en la visión global del mismo (**punto 3.3**) – ver **Figura 3.6**.

#### 3.5.1 – Características de Color y Textura

En primer lugar vamos a definir la manera de caracterizar la intensidad y sobre todo, la textura puesto que es la novedad añadida por el trabajo [7].

- **COLOR**

Para cada píxel  $p = [p_x, p_y]$  con:

$$p_x = 1, \dots, x_{\max} \quad \text{y} \quad p_y = 1, \dots, y_{\max} \quad (3.9)$$

donde  $x_{\max}$  e  $y_{\max}$  son las dimensiones de la imagen en píxeles, se calculan los vectores de características de color y de textura. Para el primero se utilizan las tres coordenadas de intensidad del espacio CIE-LAB. Como se ha visto anteriormente, este espacio de color es de los más adecuados para métodos de segmentación debido a la uniformidad con respecto a la percepción humana, en contraposición con la del extendido RGB. El CIE  $L^*a^*b$  es aproximadamente uniforme, lo que quiere decir que la distancia numérica entre dos colores se corresponde en gran medida con la distancia en la percepción de ellos. El vector correspondiente a la característica de color del píxel  $p$ ,  $I(p)$  se define como:

$$I(p) = \begin{bmatrix} I_L(p) & I_a(p) & I_b(p) \end{bmatrix} \quad (3.10)$$

- **TEXTURA**

Para caracterizar la textura de cada píxel se utiliza una descomposición Discrete Wavelet Frames (DWF) [95]. Como se vio en el **apartado 3.4.3**, es un método similar a las transformadas wavelets discretas (Discrete Wavelet Transform - DWT), que utilizan un banco de filtros para descomponer cada una de las componentes de intensidad en un subconjunto de bandas. La diferencia principal entre los dos métodos es que en el DWF la salida del banco de filtros no está muestreada. Se ha demostrado que el DWF reduce la variabilidad en la estimación de la característica de textura, mejorando de este modo el rendimiento de la clasificación [95].

El banco de filtro utilizado, basado en el filtro paso bajo de Haar, es el siguiente:

$$H(z) = \frac{1}{2}(1 + z^{-1}) \quad (3.11)$$

, que satisface la condición  $H(z)|_{z=1} = 1$ .

Su complementario paso alto  $G(z)$ , se define respecto al primero como:

$$G(z) = z H(-z^{-1}) \quad (3.12)$$

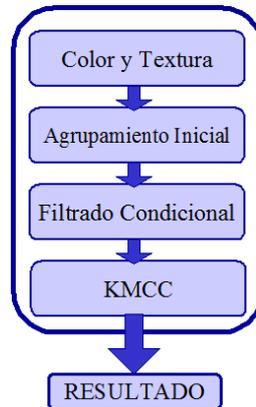
El resto de filtros del banco,  $H_v(z)$  y  $G_i(z)$  con  $i=1, \dots, V$ , se calculan como combinación de  $H(z)$  y  $G(z)$  de acuerdo a las ecuaciones presentadas en [95].

A pesar de su simplicidad, este filtro responde muy bien para la caracterización de la textura que junto con su reducido coste computacional (gracias a la combinación de filtros simples), lo convierten en una buena elección para este método de segmentación.

Finalmente, la textura del píxel  $p$  se caracterizará por las desviaciones típicas de todas las componentes calculadas en una ventana  $\Phi$ , que englobe a  $p$  y sus píxeles contiguos.

Para imágenes relativamente pequeñas (por ejemplo, 150x100 píxeles), será adecuada una descomposición en 2 dimensiones ( $N=2$ ) y con 3 componentes de detalle para cada nivel de descomposición ( $V=3$ ) y para cada componente de intensidad ( $L*a*b$ ) - ver **Figura 3.5**. Por tanto resultaría un vector de 18 componentes ( $3 \times V \times N$ ):

$$T(p) = [\sigma_1(p) \ \sigma_2(p) \ \dots \ \sigma_{9 \times N}(p)] \quad (3.13)$$



**Figura 3.4** – Fases del KMCC-Texturizado

### 3.5.2 – Agrupamiento Inicial

Al igual que otras variantes del K-means, este algoritmo requiere de unos valores iniciales que irán variando a lo largo de la ejecución:

- Una estimación del número inicial de regiones
- Sus respectivos centros espaciales, de color y de textura.

Para calcular estos valores iniciales, la imagen se divide en  $L$  bloques de dimensiones  $f \times f$  píxeles y se asigna a cada bloque  $b_l$ ,  $l=1, \dots, L$ , un vector de color  $I^b(b_l)$  y otro de textura  $T^b(b_l)$ :

$$I^b(b_l) = \frac{1}{f^2} \sum_{p \in b_l} I(p) \quad (3.14)$$

$$T^b(b_l) = \frac{1}{f^2} \sum_{p \in b_l} T(p) \quad (3.15)$$

La distancia entre dos bloques se define como:

$$D^b(b_l, b_n) = \|I^b(b_l) - I^b(b_n)\| + \lambda_1 \|T^b(b_l) - T^b(b_n)\| \quad (3.16)$$

donde  $\| \cdot \|$  son las distancias euclídeas entre los vectores correspondientes.

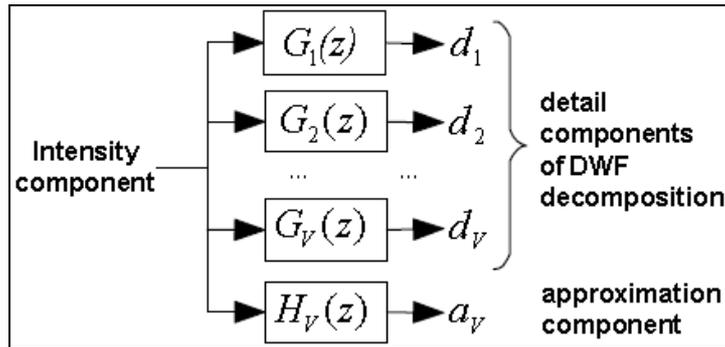


Figura 3.5 – Descomposición Discrete Wavelet Frames 1-D de V niveles.

Para todos los experimentos llevados a cabo se toma  $\lambda_l=1$  puesto que se ha demostrado [7] que es la mejor opción para una colección de imágenes variadas en cuanto a texturas. Un valor  $\lambda_l \ll 1$  da malos resultados en imágenes muy texturizadas, de la misma manera que  $\lambda_l \gg 1$  en imágenes poco texturizadas.

En esta situación, el número de regiones iniciales se estima aplicando sobre los bloques una variante del algoritmo *maximin* que se describe a continuación. El algoritmo consta de los siguientes pasos:

- **Paso 1.** Se escoge el bloque superior izquierdo como el primer centro.
- **Paso 2.** Se calcula la distancia entre ese primer centro y el resto de bloques ( $b_l$ , con  $l=1, \dots, L$ ), y aquel que maximice la distancia será el segundo centro. La distancia  $C$  entre ambos será indicativa del contraste de intensidad y textura de la imagen.
- **Paso 3.** Se calcula la distancia entre todos los bloques y cada uno de los centros (en la primera iteración serán solo dos) y se asigna el bloque al centro más “cercano”. Aquel cuya distancia al centro sea mayor se convierte en un candidato a nuevo centro.
- **Paso 4.** Si la distancia asociada al candidato a centro es mayor que  $\gamma \cdot C$ , donde  $\gamma$  es un parámetro predefinido ( $\gamma \in [0,1]$ ;  $\gamma=0$  para que cada bloque único sea un centro,  $\gamma=1$  para que solo existan 2 centros, los dos primeros), éste se acepta como centro y se vuelve al “Paso 3”. Si por el contrario la distancia es menor, el candidato a centro es rechazado y el algoritmo ha terminado.

Seguidamente se aplica el algoritmo de clasificación K-means (descrito en el apartado 3.2.1). Cuando éste converge, se reconstruye la imagen con las dimensiones iniciales, es decir, cada bloque se divide en  $f \times f$  píxeles a los que se le asigna el mismo valor. Por último, se evalúa la conectividad de las regiones obtenidas hasta el momento. En caso de encontrar regiones inconexas se dividirán en tantas como sea necesario hasta lograr que todas ellas estén conectadas.

Para ello se hace uso de la función “*component labeling*” (etiquetado de los componentes, en castellano) que estudia la conectividad de todas las regiones dividiendo aquellas que lo requieran en  $K'$  regiones menores que si estarán conectadas:  $S_k$  con  $k=1, \dots, K'$ . Se añadirán así  $K'$  nuevos centros definidos al igual que los anteriores y calculados como la media de todos los píxeles:

$$I^s(s_k) = \left[ I_L^s(s_k) \quad I_a^s(s_k) \quad I_b^s(s_k) \right] = \frac{1}{M_k} \sum_{p \in s_k} I(p) \quad (3.17)$$

$$T^s(s_k) = \left[ T_1^s(s_k) \quad \dots \quad T_{9 \times 9}^s(s_k) \right] = \frac{1}{M_k} \sum_{p \in s_k} T(p) \quad (3.18)$$

$$S(s_k) = \left[ S_x(s_k) \quad S_y(s_k) \right] = \frac{1}{M_k} \sum_{p \in s_k} p \quad (3.19)$$

donde  $M_k$  es el numero de píxeles perteneciente a la región  $S_k$ :  $S_k = \{p_1, p_2, \dots, p_{Mk}\}$ .

El número de centros estimado tras este algoritmo, será el número de regiones iniciales de la imagen segmentada.

### 3.5.3 – Filtrado Condicional

Las imágenes pueden contener fluctuaciones muy pronunciadas en su intensidad de color incluso dentro de un mismo objetos (**Figura 3.7a**). Para facilitar el agrupamiento de todos estos píxeles del mismo objeto en una misma región en base a su homogeneidad de textura, se trata de reducir el contraste de su intensidad. Para ello se aplica un filtro de media variable en las zonas adecuadas, alterando así la información de color.

La decisión de si se debe o no aplicar este filtrado a píxel particular se toma evaluando el módulo de su vector de textura  $T(p)$ . Si dicho módulo es mayor que un umbral  $\Gamma$  se aplicará el filtro de intensidad. En caso contrario la intensidad no se verá modificada. Por tanto, se puede expresar la salida del filtro como:

$$J(p) = \begin{cases} I(p) & \text{si } \|T(p)\| < \Gamma \\ \frac{1}{f^2} \sum I(p) & \text{si } \|T(p)\| \geq \Gamma \end{cases} \quad (3.20)$$

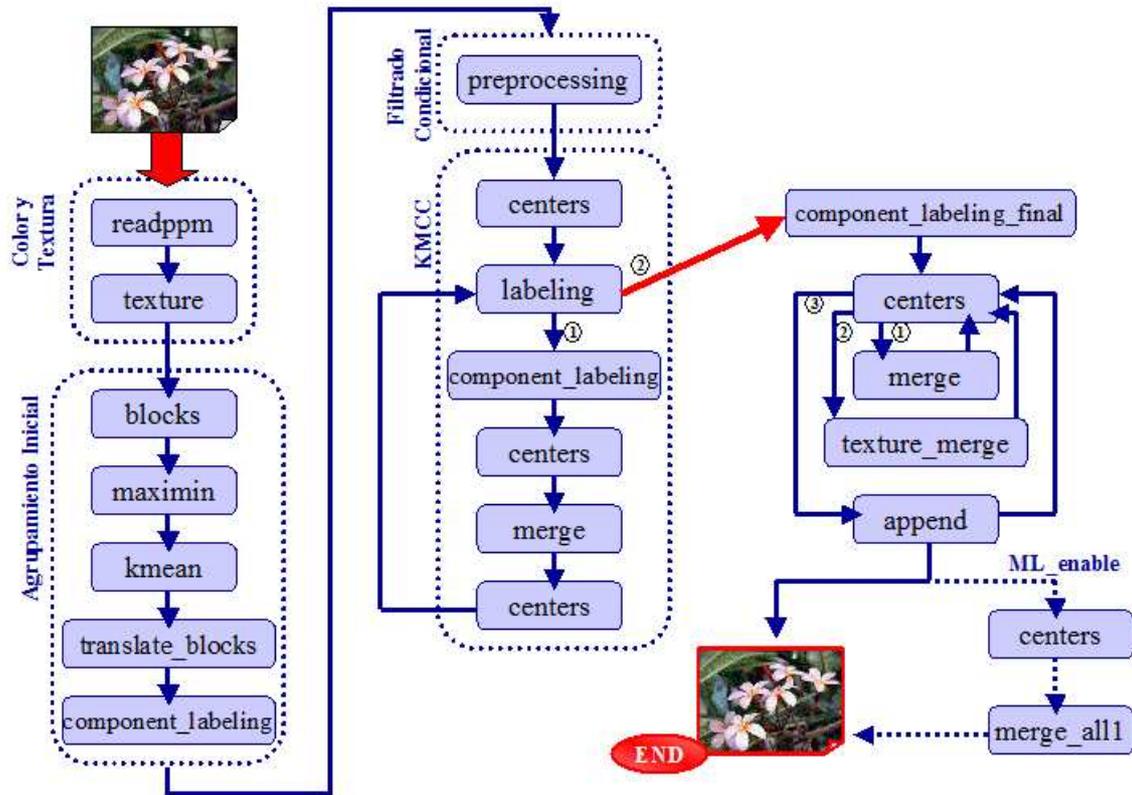


Figura 3.6 – Esquema detallado del KMCC-Texturizado

Un umbral apropiado según los resultados experimentales [7] sería:

$$\Gamma = \max \{0.65 \cdot T_{\max}, 14\} \quad (3.21)$$

donde  $T_{\max}$  es el módulo máximo de los vectores de textura de la imagen:  $\max \|T(p)\|$ . El término  $0.65 \cdot T_{\max}$  se establece para prevenir la aplicación del filtro fuera de los objetos texturizados y no eliminar los bordes. Por otro lado, la constante máxima de 14 se utiliza para evitar un filtrado indeseado en objetos monocromáticos en los que es suficiente estudiar su homogeneidad de color para lograr un agrupamiento adecuado de los píxeles.

Un ejemplo del resultado de este filtro se puede observar en la **Figura 3.7b**.

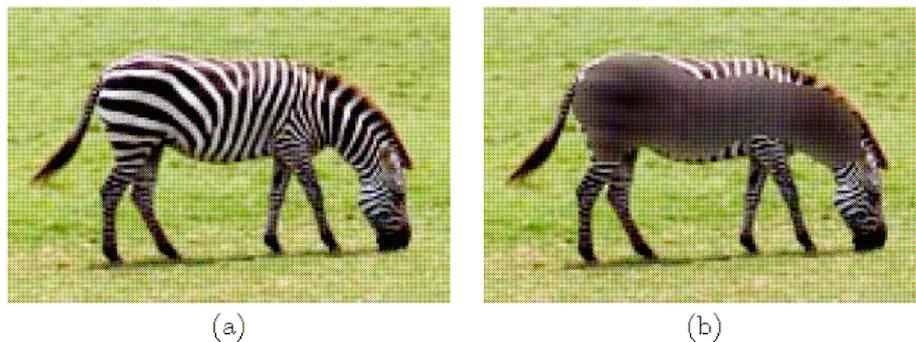


Figura 3.7 – Ejemplo de foto filtrada condicionalmente.

### 3.5.4 – KMCC

Como ya se ha visto en el **apartado 3.2**, el algoritmo K-means original [81] no tiene en cuenta la información espacial de los píxeles. Esta característica se añadía posteriormente en el KMCC [80]. En este método expuesto, se añade también la característica de la textura y por tanto se deberá incorporar un análisis y unos cálculos asociados, en los pasos a seguir. Todo esto, junto con el filtro condicional aplicado resulta en la siguiente secuencia que se puede seguir mediante el diagrama de bloques de la **Figura 3.6**:

- **Paso 1.** Se obtienen los datos iniciales del número de regiones estimadas y sus respectivos centros - **apartado 3.5.2**.
- **Paso 2.** se calcula la distancia de cada píxel a todos los centros. Éste se asigna a la región que minimice dicha distancia. La **ecuación 3.3** es una expresión generalizada para calcular dicha distancia, a la que se le añade un término referente a la textura:

$$D(p, s_k) = \|J(p) - J^s(s_k)\| + \lambda_1 \|T(p) - T^s(s_k)\| + \lambda_2 \frac{\bar{M}}{M_k} \|p - S(s_k)\| \quad (3.22)$$

donde  $\| - \|$  será la distancia euclídea,  $M_k$  es el tamaño en píxeles de la región  $S_k$  y  $\bar{M}$  el tamaño medio de las regiones. El valor de la constante  $\lambda_1=1$  fue ya justificado en el **punto 3.5.2** y el valor de  $\lambda_2$  será:

$$\lambda_2 = 0.4 \frac{C}{\sqrt{x_{\max}^2 + y_{\max}^2}} \quad (3.23)$$

Esta ponderación asegura que un píxel será asignado a una u otra región principalmente en función de su homogeneidad de color y de textura. La normalización del tercer término, por el término que divide el tamaño medio de las regiones entre el tamaño de la región, fue también justificado anteriormente como una manera de evitar la sobresegmentación de las regiones de mayor tamaño.

- **Paso 3.** Se evalúa la conectividad de las regiones formadas y las que no lo están se dividen en el máximo número de subregiones posibles.
- **Paso 4.** Se recalculan los centros (**Ecuaciones 3.17-19**). Las regiones con un tamaño inferior al 0,5% de la imagen serán descartadas como tales, para evitar la creación de pequeñas e insignificantes regiones. Se recalcula el número de regiones K, descontando las que no cumplan esta condición.
- **Paso 5.** Se calcula la distancia de la intensidad y de la textura de las regiones vecinas con el fin de unificar aquellas cuya diferencia no supere un cierto umbral:

$$D^s(s_{k_1}, s_{k_2}) = \|J^s(s_{k_1}) - J^s(s_{k_2})\| + \lambda_1 \|T^s(s_{k_1}) - T^s(s_{k_2})\| \leq \mu \quad (3.24)$$

El umbral utilizado en los experimentos que se van a llevar a cabo será el mismo que el autor del algoritmo utilizó [7]:

$$\mu \begin{cases} 7,5 & \text{si } C < 25 \\ 15 & \text{si } C > 75 \\ 10 & \text{otros} \end{cases} \quad (3.25)$$

donde  $C$  es la medida de contraste de intensidad y textura de la imagen definida en el apartado como la diferencia 3.5.2.

- **Paso 6.** Se recalcula el número  $K$  de regiones y sus respectivos centros.
- **Paso 7.** Si el número de regiones es igual a las que había en el *Paso 6* en la interacción anterior y la distancia entre los centros actuales y los calculados en la interacción anterior es menor del umbral correspondiente se finaliza la ejecución y se puede determinar que el algoritmo converge. En caso contrario, se vuelve al *Paso 2*. La condición de convergencia vendría dada por estas 4 ecuaciones:

$$\left. \begin{array}{l} K = K^{old} \\ \|J^s(s_k) - J^s(s_k^{old})\| \leq c_I \\ \|T^s(s_k) - T^s(s_k^{old})\| \leq c_T \\ \|S(s_k) - S(s_k^{old})\| \leq c_S \end{array} \right\} \text{converge} \longrightarrow STOP \quad (3.26)$$

para  $k=1, \dots, K$ . Sin embargo, no existe la certeza de que el algoritmo vaya a converger para todas las imágenes por lo que se le da un número máximo de iteraciones. Si se exceden las 20 iteraciones la secuencia procede como si las condiciones de convergencia se cumplieran, y finaliza el algoritmo KMCC texturizado.

### 3.5.5 – Imágenes de Grandes Formatos

El método hasta ahora propuesto en esta sección es considerablemente rápido para imágenes de tamaños razonablemente pequeños (100x150 píxeles), pero su efectividad se ve desmejorada al aumentar las dimensiones de la imagen. El tiempo de ejecución aumenta proporcionalmente al número de píxeles al aumentar la complejidad computacional.

Para desarrollar un método válido, bastará con tener en cuenta el hecho ya asumido de que los objetos cuya dimensión sea menor al 0,75% del área total de la imagen serán considerados irrelevantes para las aplicaciones multimedia. En imágenes de grandes dimensiones este tanto por ciento supone una gran cantidad de píxeles que podrían ser unificados en un único bloque y ahorrar costes (computacionales). Esto se traduce en la obtención de una versión reducida de la imagen a segmentar, sin eliminar información de los objetos detectables y agilizando las operaciones a realizar.

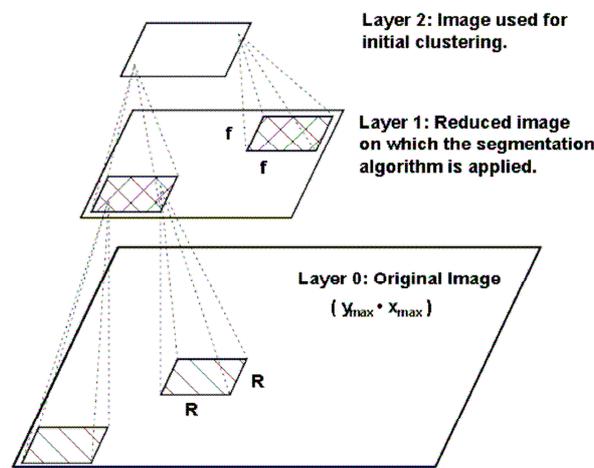
La imagen reducida derivará de la original mediante la asociación de bloques de  $R \times R$  píxeles que se representarán por un único píxel en esa nueva versión (ver **Figura 3.8**) donde  $R$  será el factor de reducción. Para cumplir la condición del tamaño mínimo de los objetos a detectar se debe cumplir siempre que:

$$\frac{\psi}{R^2} \gg 1 \quad (3.27)$$

donde  $\psi$  se define como,

$$\psi = \frac{0.75}{100} \cdot y_{\max} \cdot x_{\max} \quad (3.28)$$

con  $x_{\max}$  e  $y_{\max}$  como dimensiones de la imagen original.



**Figura 3.8** – Esquema de los distintos niveles del KMCC-Texturizado [7].

A pesar de ser el último paso descrito en esta sección, esta reducción de la imagen será realizada como primer paso del algoritmo siempre que sea llevada a cabo. El resto del algoritmo funcionará de la misma manera, tratando cada bloque como si se tratara de un píxel común. Como se observa en la **Figura 3.8**, el algoritmo de agrupamiento inicial realizaría como vimos en 3.5.1 una nueva división para obtener así las primeras regiones con sus respectivos centros de manera automática.

Tras la aplicación normal del método sobre la imagen reducida se procede con la reconstrucción de la original, cuyo punto débil serán los bordes. Para evitar un empeoramiento en la calidad percibida en el resultado se aplicará el teorema de Bayes para la clasificación de aquellos bloques situados en los bordes. Todos los píxeles correspondientes a estos bloques serán reclasificados, para conseguir unos bordes con mayor exactitud y más “agradables” a la vista. La probabilidad de que un cierto píxel pertenezca a una u otra región vendrá condicionada únicamente por su intensidad, ignorando el tamaño de las distintas áreas como se demuestra en [7].

### 3.6 – Resultados de la Evaluación

A continuación se van a exponer los resultados obtenidos tras la aplicación del KMCC-Texturizado, expuesto a lo largo de este capítulo. La selección de imágenes a segmentar (agrupada en dos colecciones A y B, de 60 imágenes cada una) es la que se presentó en detalle en el apartado 3.1.1.

Para la evaluación del método se establecen distintos escenarios en función de los múltiples parámetros de entrada existentes. El objetivo es analizar el comportamiento del mismo y hallar así los casos óptimos para su posterior utilidad (capítulo 5). Dichos parámetros serán los siguientes:

- **W:** indica el factor de reducción de la imagen, es decir, tamaño de los bloques en los que se divide la imagen inicialmente. Variable de 1 a 8. Es el valor  $R$  de la sección 3.5.5.
- **LDC:** establece el número de niveles para la descomposición de la transformada DWF (2, 3 o 4). Es  $N$  en el punto 3.5.1.
- **ML\_enable:** habilita/deshabilita el uso del Teorema de Bayes para la reconstrucción de la imagen, tras aplicar el método, al traducir los bloques creados. Supuestamente deshabilitado para  $W=1$  y habilitado en el resto de los casos.
- **MaxAreas:** fija el número máximo de áreas que se podrán obtener. Podrá venir fijado por la aplicación multimedia.

#### 3.6.1 – Método de Evaluación

Antes de comenzar con la descripción de los resultados obtenidos se describe el método utilizado para dicha evaluación. La herramienta desarrollada por T. Adamek [97], se basa en un método de evaluación que fue desarrollado por V. Mezaris [76] y del cual se han desarrollado múltiples versiones (Ej. [78]).

##### 3.6.1.1 – Método Original de Evaluación [76]

Denotamos  $S = s_1, s_2, \dots, s_K$  como las  $K$  regiones de las imágenes resultado de una segmentación que se van a evaluar mediante la comparación con las  $Q$  regiones de sus correspondientes máscaras de referencia,  $R = r_1, r_2, \dots, r_Q$ . Para ello, en primer lugar se lleva a cabo una asociación de las *regiones resultado* -  $S_K$ , y las *regiones referencia* -  $R_Q$ . Debido a la sobre- o sub-segmentación cabe la posibilidad de que queden regiones sin asignarse (tanto  $S_K$  como  $R_Q$ ).

Una vez establecidas las asociaciones, se computan los errores espaciales. Denominamos el conjunto de parejas de regiones como  $A = (r_q, s_k)$  y por otro lado  $N_R$  y  $N_S$  serán

el conjunto de regiones no asociadas de  $R$  y  $S$ , respectivamente. Con todo ello, el error espacial se evaluará según [78] como:

$$E_q = \sum_{p \in (r_q - r_q \cap s_k)} f_1(p, r_q) + \sum_{p \in (s_k - r_q \cap s_k)} f_2(p, r_q) \quad \forall (r_q, s_k) \in A \quad (3.29)$$

donde  $f_1(\dots)$  y  $f_2(\dots)$  son las funciones de ponderación del error presentadas en [78] para diferenciar los píxeles mal clasificados en función de su distancia a la región a la que pertenecen – véase **Figura 3.9**.

El error  $E_q$  debido a los píxeles pertenecientes a una región  $r_q$  no asociada a ninguna otra (píxeles no asociados) se calcula con la siguiente expresión:

$$E_q = \sum_{p \in r_q} f_1(p, r_q), \quad \forall r_q \in N_R \quad (3.30)$$

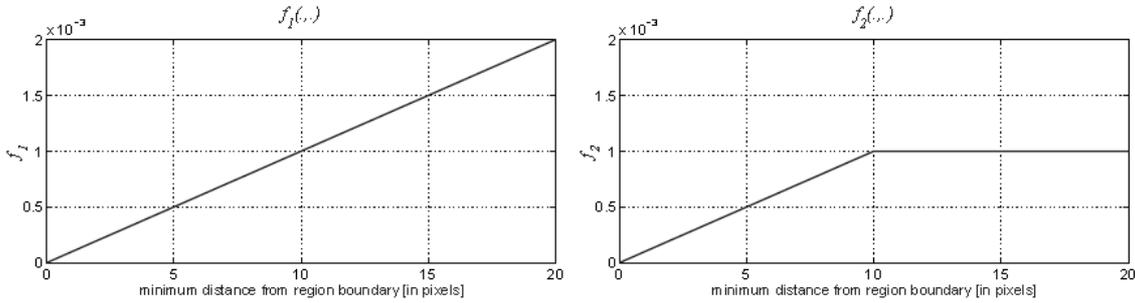
Y análogamente los píxeles no asociados, pertenecientes a una región  $s_k$ , contribuyen al error del siguiente modo:

$$E_k = \alpha \sum_{p \in s_k} f_1(p, r_q), \quad \forall s_k \in N_S \quad (3.31)$$

donde  $\alpha$  es un factor de escala peyorativo, relacionado con la sobre segmentación.

El resultado global del error se computa como la suma de los tres anteriores:

$$E = \sum_{\forall (r_q, s_k) \in A} E_q + \sum_{\forall r_q \in N_R} E_q + \alpha \sum_{\forall s_k \in N_S} E_k \quad (3.32)$$



**Figura 3.9** – Funciones de ponderación para el error espacial en función de la distancia del píxel con la región a la que pertenece.

### 3.6.1.2 – Método Utilizado para la Evaluación

Aunque el uso de las funciones de ponderación,  $f_1$  y  $f_2$ , está más que justificado, es una realidad que aumentan el coste computacional de la evaluación. El cálculo de la contribución que aporta cada píxel erróneo supone una búsqueda de la distancia mínima al borde de la región de referencia a la que pertenece. Esto supone un coste computacional que aumenta exponencialmente con el tamaño de la imagen. Dado que los experimentos realizados consisten en una gran variedad de escenarios de evaluación (todas las combinaciones de los parámetros), se opta por una versión simplificada y más rápida del método anterior desarrollada en [97].

En este enfoque, todos los píxeles erróneos contribuyen en la misma medida al error, independientemente de su distancia a la frontera de su región de referencia. Para poder

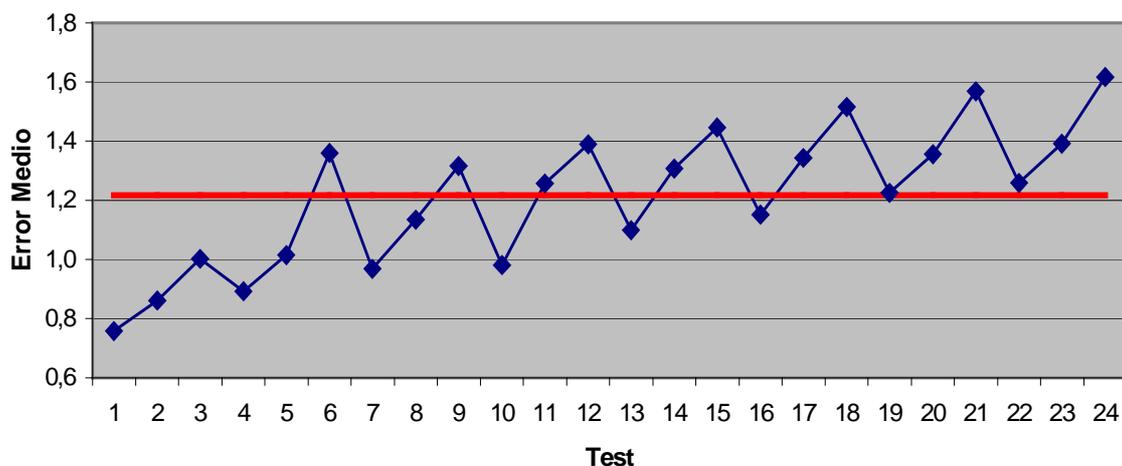
comparar los resultados de imágenes de diferentes tamaños, el error de precisión se normaliza por el tamaño de la imagen:  $E_n = E / N$ , donde  $N$  representa el número total de píxeles de la imagen. Esta versión requiere una única exploración de los resultados evaluados, lo que se traduce en una rápida ejecución sin renunciar a la evaluación significativa de los resultados.

Además, la descripción en [76] da una cierta libertad con respecto a la estrategia a seguir para la asociación de las regiones. De hecho, asumiendo la exclusividad de la asociación y maximizando  $R_Q \cap S_K$ , como único criterio para la vinculación, se pueden crear correspondencias de grandes regiones con solo unos pocos píxeles en común, si la región que le corresponde ya ha sido asociada a otra de la máscara de referencia. Por lo tanto, en esta versión [97] se añade también un requisito de emparejamiento con el fin de evitar estas asociaciones inadecuadas:  $(R_Q \cap S_K) / (R_Q \cup S_K) > 0,5$ .

Dicho con palabras, las regiones de las imágenes evaluadas y las de las máscaras de referencia sólo pueden ser vinculadas si su intersección es por lo menos la mitad del tamaño de su unión. Esta restricción da lugar a un estricto criterio de evaluación, pero que se acerca bastante a la percepción humana de la calidad de la segmentación.

### 3.6.2 –KMCC-Texturizado

Mediante la combinación de los parámetros de entrada ya nombrados (W, LDC, ML\_enable, MaxAreas) se crean 24 escenarios distintos o casos de prueba que se pueden observar en la **Tabla 3.1**. El objetivo será determinar la combinación óptima o aquellas más ventajosas según la respuesta del algoritmo. En la tabla se resumen los resultados numéricos tras la evaluación de las dos colecciones de imágenes (A y B) y su respectiva media para cada escenario.



**Figura 3.10** – Error espacial medio obtenido para los escenarios de la **Tabla 3.1** con el KMCC-T

Test	W	LDC	ML_enable	Max Areas	Col A	Col B	Media
1	1	2	0	20	0,6457	0,8718	0,7587
2	1	3	0	20	0,7790	0,9450	0,8620
3	1	4	0	20	0,8646	1,1409	1,0028
4	2	2	1	20	0,7871	0,9999	0,8935
5	2	3	1	20	0,9625	1,0688	1,0157
6	2	4	1	20	1,3570	1,3631	1,3601
7	3	2	1	20	0,9122	1,0264	0,9693
8	3	3	1	20	1,1040	1,1643	1,1342
9	3	4	1	20	1,3832	1,2486	1,3159
10	4	2	1	20	0,9642	0,9977	0,9810
11	4	3	1	20	1,2881	1,2235	1,2558
12	4	4	1	20	1,4753	1,3048	1,3901
13	5	2	1	20	1,0718	1,1240	1,0979
14	5	3	1	20	1,2996	1,3131	1,3064
15	5	4	1	20	1,5169	1,3754	1,4462
16	6	2	1	20	1,1346	1,1677	1,1512
17	6	3	1	20	1,2894	1,3983	1,3439
18	6	4	1	20	1,5612	1,4721	1,5167
19	7	2	1	20	1,2135	1,2350	1,2243
20	7	3	1	20	1,3976	1,3125	1,3551
21	7	4	1	20	1,6345	1,5010	1,5678
22	8	2	1	20	1,2412	1,2762	1,2587
23	8	3	1	20	1,4800	1,3040	1,3920
24	8	4	1	20	1,7592	1,4731	1,6162

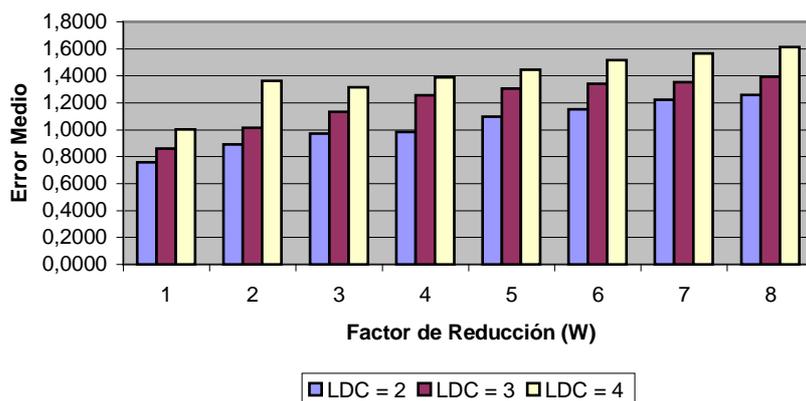
**Tabla 3.1** – Distintos escenarios objetos de los experimentos y resultados obtenidos en ambas colecciones

Además, los resultados se han reflejado también en la **Figura 3.10** mediante un gráfico indicando el valor del error espacial medio de ambas colecciones de imágenes para cada uno de los casos experimentales.

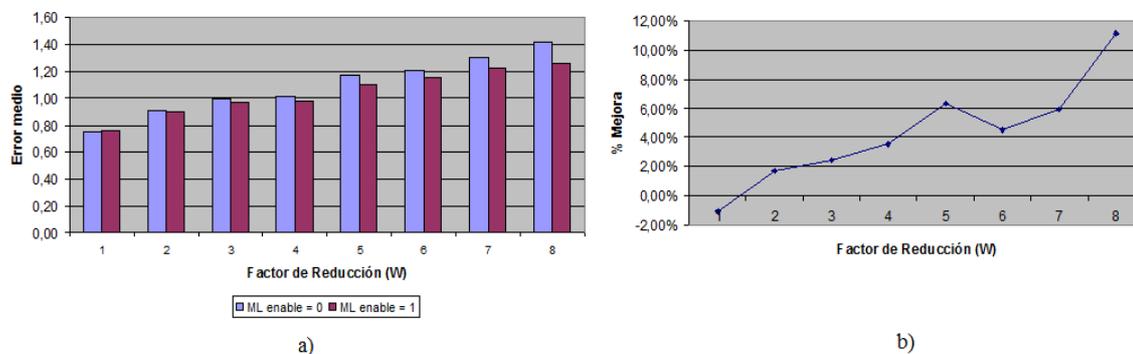
En la **Tabla 3.1** se han resaltado varios escenarios de especial interés (test 1, 4 y 10). Por un lado el error mínimo obtenido para el test 1 en el que factor de reducción es “nulo”, ( $W=1$ ) es decir, la imagen no se divide inicialmente. El resultado de la segmentación es el mejor pero sus tiempos de ejecución serán muy altos. Los resultados de tiempos no se reflejan en esta memoria, a pesar de haber sido tenidos en cuenta durante el desarrollo y experimentación, puesto que no se han considerado de especial interés para este trabajo.

Por otro lado, son también interesantes los test 4 y 10 en los que para un factor de reducción 2 y 4, respectivamente, se obtiene un error menor a 1 que se considera aceptable. En estos casos queremos destacar el valor del factor  $W$ , que especialmente para el test 10 se traduce en una considerable mejora en tiempo y gasto computacional respecto al test 1. Por tanto, si estamos buscando un equilibrio entre calidad y efectividad, podríamos decir que el test 10 es el “caso óptimo”.

De los datos de la **Tabla 3.1** se puede obtener otra conclusión que se ha reflejado en la **Figura 3.11**. Observando el valor relativamente cíclico del error respecto al LDC para cada factor de reducción ( $W$ ), representado en la **Figura 3.11**, se concluye que el valor adecuado para ambas colecciones es 2. Este parámetro, determinante para el análisis de las texturas, depende del tamaño de la imagen a segmentar. Por tanto, los mínimos en  $LDC = 2$ , corroboran que para imágenes de pequeñas dimensiones (como la mayor parte de las que se están tratando en estos experimentos:  $\sim 120 \times 180$ ), un nivel dos de descomposición será el óptimo para la transformada.



**Figura 3.11** – Evaluación del factor LDC respecto al factor de reducción,  $W$ .



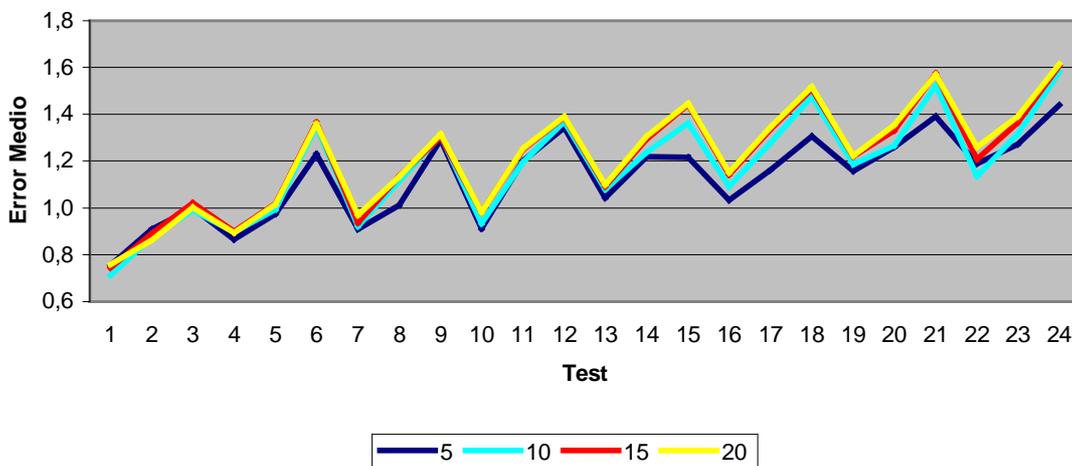
**Figura 3.12** – a) Errores medios obtenidos en función de  $W$  con la aplicación o no del Teorema de Bayes. b) Mejora obtenida gracias al Teorema de Bayes para cada  $W$ .

En general, para cualquier resultado y/o cualquier tipo de representación, podemos ver cómo al aumentar el factor de reducción el error aumenta proporcionalmente. Esto es de esperar puesto que al dividir la imagen original en bloques y asignar un vector de intensidad de color al conjunto de píxeles se está perdiendo información (más información cuanto mayor sean estos bloques). El algoritmo realiza todos los cálculos con dichos bloques para optimizar su rendimiento pero a cambio estará ignorando parte de información. Sin embargo, como se ha comentado en el apartado de diseño, esta pérdida puede ser compensada aplicando el Teorema

de Bayes sobre el resultado final de la segmentación y recalculando una última vez los centros y áreas. Como se observa en la **Figura 3.12a**, en la que se comparan los errores medios obtenidos al aplicar o no (ML enable=1/0) la función correspondiente, el error es menor al aplicar el teorema de Bayes en la reconstrucción de la imagen original. Además, la mejoría será mayor cuanto mayor sea el factor de reducción, esto es, cuanto más información se haya asociado a un mismo bloque – ver **Figura 3.12b**.

W	Test	Max Area = 20	Max Area = 15	Max Area = 10	Max Area = 5
		Media	Media	Media	Media
1	1	0,76	0,74	0,71	0,76
	2	0,86	0,89	0,88	0,91
	3	1,00	1,02	0,99	1,00
2	4	0,89	0,90	0,90	0,86
	5	1,02	1,02	1,00	0,97
	6	1,36	1,36	1,35	1,23
3	7	0,97	0,94	0,92	0,91
	8	1,13	1,14	1,11	1,01
	9	1,32	1,30	1,31	1,29
4	10	0,98	0,98	0,93	0,91
	11	1,26	1,25	1,20	1,20
	12	1,39	1,39	1,37	1,34
5	13	1,10	1,09	1,08	1,04
	14	1,31	1,29	1,24	1,22
	15	1,45	1,44	1,36	1,22
6	16	1,15	1,14	1,09	1,03
	17	1,34	1,34	1,28	1,16
	18	1,52	1,51	1,48	1,31
7	19	1,22	1,22	1,19	1,16
	20	1,36	1,33	1,27	1,26
	21	1,57	1,57	1,52	1,39
8	22	1,26	1,21	1,13	1,19
	23	1,39	1,36	1,31	1,27
	24	1,62	1,61	1,58	1,44
MEDIA		1,22	1,21	1,17	1,13

**Tabla 3.2** – Error para los escenarios de la **Tabla 3.1** variando el número máximo de áreas.



**Figura 3.13** – Error medio obtenido variando el número máximo de áreas.

El parámetro que determina el máximo número de áreas se ha fijado a 20 (un valor relativamente alto), tras haber comprobado que para valores superiores el error medio permanece invariable. Es decir, que el máximo número de regiones de las máscaras de referencia de la colección estudiada es menor que 20. Por tanto así se ha evitado su interferencia, tanto negativa como positiva, en los resultados obtenidos hasta ahora.

Para analizar su influencia de manera individual, a continuación se muestran los resultados obtenidos para los siguientes valores de “Max Areas”: 5, 10, 15 y 20. En la **Tabla 3.2** se reflejan los resultados numéricos del error medio espacial, y se remarca el valor mínimo obtenido para cada valor de W. Como ya hemos justificado anteriormente, para todos los casos el error se minimiza en “LDC = 2” (ver **Tabla 3.1** para asociar número de test y LDC). No obstante, en este experimento nos incumbe para qué valores del máximo número de áreas se obtiene el menor error y como se puede observar en la **Figura 3.13**, la línea correspondiente a “Max Area = 5” destaca levemente por debajo del resto.

El hecho de que este parámetro sea considerablemente bajo puede explicarse de dos maneras:

- 1) Debido a las máscaras de referencia. Los resultados nunca serán completamente objetivos, puesto que dichas máscaras han sido realizadas manualmente. La subjetividad radica en la decisión de qué partes de la imagen serán marcadas como objetos y qué otras se considerarán irrelevantes. Tanto en el sujeto que lleva a cabo la acción, como en los objetivos de la aplicación para la que se diseñan las máscaras marcarán las tendencias.
- 2) El tamaño de la imagen. A pesar de existir cierta subjetividad en el diseño de las máscaras de referencia podemos asumir una dependencia objetiva entre el tamaño de las imágenes y el número de objetos localizados. Es razonable que la importancia de un objeto dependa de espacio que ocupa en proporción a la imagen y requiera unas dimensiones mínimas.

A la vista de estos resultados parciales podemos concluir que habrá varios escenarios óptimos (ver **Tabla 3.3**) para el algoritmo KMCC-Texturizado con unos determinados parámetros que dependerán siempre, de la colección de imágenes que se utilice y mediante los que se debe llegar a un compromiso entre rendimiento y eficacia.

Test	W	LDC	ML_enable	Max Areas	Media
1	1	2	0	10	0,712
				20	0,759
4	2	2	1	5	0,865
				20	0,894
10	4	2	1	5	0,910
				20	0,981

**Tabla 3.3** – Escenarios óptimos para la aplicación del KMCC-T sobre las imágenes seleccionadas.

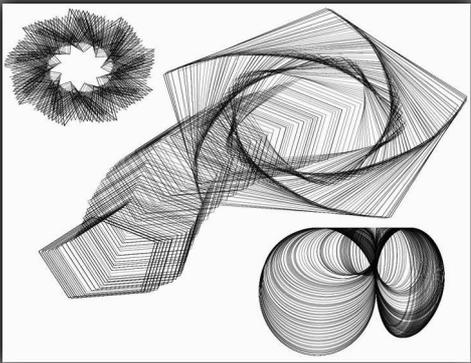
### 3.7 – Conclusiones

A lo largo de este capítulo se ha presentado el método de segmentación KMCC-T (K-Means with Connectivity Constraint-Texturizado) [7] basado en características de intensidad, posición y textura, para la formación de regiones que se corresponden con los objetos contenidos en una imagen. El método incluye una capacidad adicional para lograr una rápida segmentación de imágenes a color de dimensiones mayores. Gracias a ésta, se verá mejorada la eficiencia del tiempo de procesamiento habitual sin suponer un empeoramiento de los resultados. Para ello combina el algoritmo de segmentación con un clasificador de Bayes, aplicable a cualquier otro método de segmentación.

Tras una detallada exposición teórica del mismo se ha llevado a cabo un extenso análisis experimental con el fin de hallar las mejores combinaciones de sus parámetros de entrada y comprobar la influencia de cada uno de ellos.

El algoritmo propuesto es adecuado para su uso como parte de aplicaciones multimedia basadas en objetos, tales como la consulta de imágenes o para definir las regiones de interés en la codificación de imágenes fijas basada en contenido (por ejemplo, en el contexto de la norma JPEG2000).

4 - Análisis del algoritmo  
M-RSST





## 4 – Análisis del Algoritmo M-RSST

En este capítulo se presenta un segundo método objeto de estudio en este proyecto fin de carrera: el algoritmo denominado Modified-Recursive Shortest Spanning Tree (M-RSST) [11].

### 4.1 – Introducción

#### 4.1.1 – Motivación

La motivación para elegir este algoritmo como herramienta de mejora para aplicárselo al KMCC-T [7] presentado en el capítulo anterior, es la diversidad de las fuentes integradas en él. El diseño del M-RSST se apoya en la viabilidad de utilizar la configuración espacial de las regiones y sus propiedades geométricas (conocidas como “características sintácticas” [12]) para mejorar los resultados del algoritmo RSST. Estudiar estas características sintácticas en detalle será el objetivo de este capítulo siempre encaminado hacia la motivación global del trabajo.

Un paso previo para la integración de las nuevas características añadidas en el M-RSST será analizarlas y compararlas con la evidencia dada únicamente por el criterio de homogeneidad de color. También se exponen dos mejoras para dicho criterio de homogeneidad de color. Y finalmente, el método de integración de múltiples fuentes, que permite unificar la información de todas ellas en un solo criterio, diferenciando individualmente su exactitud y fiabilidad. Éste se apoyará en la *Teoría de la evidencia* (o *Belief Theory*) [98], que se verá también a lo largo del capítulo.

#### 4.1.2 – Estructura del capítulo

Tras esta primera sección introductoria, se hace una descripción breve del RSST original, seguida de una primera visión global del M-RSST, que se explicará con más detalle en los puntos siguientes: en el 4.4 se definen dos mejoras para el criterio de homogeneidad de color; seguidamente, en el 4.5 se explican las dos nuevas magnitudes diseñadas para aportar información en el criterio de unión de regiones (las características sintácticas). Una vez descritas todas las fuentes de datos a tener en cuenta, en el apartado 4.6 se describe el método utilizado para la integración de las mismas en una única función de coste de la unión. En el punto 4.7 se describe el criterio de parada utilizado por el M-RSST que fue desarrollado por el mismo autor. Para terminar el capítulo se resumen las conclusiones extraídas del estudio en la sección 4.8.

## 4.2 – Algoritmo de Segmentación RSST

El algoritmo escogido como base del M-RSST, sobre el que se aplicaron las mejoras, es el *Recursive Shortest Spanning Tree* (RSST). Además de la conveniencia de este algoritmo por sus resultados [49] también se tuvo en cuenta su óptima velocidad de ejecución a la hora de la elección. Muy importante teniendo en cuenta que las mejoras supondrán siempre un incremento del coste computacional y que da unos resultados finales muy eficientes en ese aspecto.

El RSST fue propuesto por Morris et al. [49] como posible solución tanto para la segmentación como para la detección de contornos en una imagen. Más tarde se extendió a imágenes de color [99] y se mostró su gran utilidad para codificación de imágenes [100]. También ha sido utilizado para segmentación de video [101] y ha servido en múltiples ocasiones como inspiración o base para el desarrollo de nuevos algoritmos [11].

El método original comienza con un mapeo de la imagen en un grafo ponderado, en el que las regiones (píxeles inicialmente) son representadas por los nodos, y las uniones de éstos representan el coste de la unión de sus correspondientes regiones (o píxeles), calculadas de acuerdo a algún criterio de homogeneidad. La unión de los nodos se realiza iterativamente, conectando en cada una de las iteraciones dos regiones en función del mínimo coste que represente la unión en cuestión. Estas uniones implican unir dos nodos en la representación creada paralelamente al proceso, y a su vez conllevan una actualización de los valores en las uniones (recalculando los costes) y regiones vecinas del grafo. El proceso continúa hasta que se da una cierta condición de parada, como por ejemplo, alcanzar el número de regiones deseadas o que el coste mínimo de unión exceda un umbral predefinido.

El criterio de homogeneidad utilizado para definir la función del coste de las uniones y la condición de parada, juegan un papel clave en la calidad de los resultados. Sobre todo es esencial en el caso de la función de coste (compuesta únicamente por el criterio de homogeneidad de color en el RSST) puesto que en cada iteración solo se unirán aquellas dos regiones que supongan un coste mínimo, lo que significa que la segmentación está controlada por esta función. Dadas dos regiones vecinas  $r_i$  y  $r_j$ , el RSST define la función de coste de la unión de ambas, basada únicamente en un criterio de homogeneidad de color, como:

$$C_{orig}(i, j) = \left\| c_i - c_j \right\|_2^2 \frac{a_i a_j}{a_i + a_j} \quad (4.1)$$

donde  $c_i$  y  $c_j$  son los colores medios de sus correspondientes regiones,  $a_i$  y  $a_j$  los respectivos tamaños y  $\| \cdot \|_2$  representa el módulo.

Para el criterio de parada podemos encontrar una mayor variedad. Algunos de los más destacados en la literatura son el número de regiones deseadas [102], un cierto umbral mínimo aplicado a la función del coste de unión de regiones [49], etc. Sin embargo, ninguno de éstos es capaz de dar como resultado regiones grandes y asegurar al mismo tiempo que el proceso se

pare antes de eliminar los bordes más relevantes de la imagen. Por este motivo el M-RSST presenta, junto con este conjunto de mejoras para el RSST original, un nuevo criterio de parada en [11].

El RSST es capaz de sobre-segmentar rápidamente una imagen (obteniendo unas 100 regiones), siendo de gran utilidad en muchas aplicaciones – véase como ejemplo [102]. Sin embargo, el criterio de color expuesto (**ecuación 4.1**) no es válido en aplicaciones que requieran de una gran simplificación de la imagen ya que no se obtendrán como resultado regiones suficientemente amplias como para lograr una correspondencia semántica aceptable (a menudo los resultados no representarían con exactitud los objetos reales de la escena).

Uno de los puntos débiles de este algoritmo es que utiliza en todas las etapas del proceso de unión de regiones el mismo criterio de homogeneidad, independientemente del número de píxeles que se estén tratando. Esto es precisamente una mejora que se añade a posteriori en el M-RSST.

En resumen, el autor selecciona el algoritmo RSST para la incorporación de las características visuales sintácticas principalmente por las siguientes razones:

- Está basado únicamente en un criterio de homogeneidad de color y cuando se usa para producir segmentaciones con un bajo número de regiones, los resultados suelen tener propiedades geométricas impropias del mundo real. Por ello, la utilización de otras propiedades (las características sintácticas) se intuye como una mejora apropiada.
- Proporciona un fácil acceso a las propiedades geométricas de todas las regiones al ser un método iterativo, con una sola unión por iteración.
- Es capaz de producir una sobre-segmentación con contornos bien definidos, es decir, sin obviar información sustancial.
- Su alta velocidad (especialmente si se compara con algoritmos basados en grafos).
- Su fácil implementación.

### 4.3 – Una Visión Global del M-RSST

Para abordar las limitaciones del RSST original el algoritmo M-RSST propone una función del coste de unión mejorada. Como ya hemos comentado, se integran propiedades geométricas durante el proceso de unión, además de mejorar la de homogeneidad de color y se incluyen todas en una única función de coste.

El proceso de segmentación se divide en dos etapas: una partición inicial, requerida para el análisis de la estructura obtenida mediante el algoritmo RSST original (extrayendo resultados suficientemente buenos y rápidos); una segunda etapa en la que se añade el estudio de los contornos y las formas geométricas, además de un nuevo modelo de color.

La primera etapa asegura el bajo coste computacional del algoritmo global y evita aplicar el análisis de características sintácticas de manera poco eficaz, sobre regiones pequeñas y de contornos poco significativos. Ésta termina forzosamente al alcanzarse un número predefinido de regiones (cien regiones para los experimentos presentados).

En la segunda, se redefine el criterio de homogeneidad basándose en las características sintácticas y de color, y el proceso de unión continúa hasta cumplir el criterio de parada.

A continuación se describen las propuestas, consideradas principalmente para la segunda etapa en la que se parte de regiones con una forma y tamaño supuestamente significativos.

## 4.4 - Homogeneidad de Color

La homogeneidad de color es el criterio básico utilizado en prácticamente todos los métodos de segmentación que se pueden encontrar en la literatura. En esta sección se describen con detalle las modificaciones que presenta M-RSST frente al algoritmo RSST original. El objetivo es encontrar el criterio de homogeneidad de color más apropiado, antes de emplear o añadir otros criterios de características menos influyentes.

### 4.4.1 - Espacios de color

La inmensa mayoría de publicaciones y métodos abogan por la utilización de los denominados espacios perceptualmente uniformes: CIE LUV y su versión mejorada CIE LAB [6, 7, 32]. El espacio utilizado en este algoritmo es el CIE LUV, al considerarlo como el óptimo cuando se trata de buscar un equilibrio entre la uniformidad y el coste de conversión desde los espacios habitualmente utilizados: RGB o YUV.

### 4.4.2 - Criterio de color optimizado

A pesar de su aparente simplicidad, la función de coste definida anteriormente (**ecuación 4.1**) concluye en reglas de unión relativamente complejas. Se da preferencia a las regiones pequeñas incluso cuando el color es significativamente distinto, “castigando” las uniones de regiones mayores. Esto conlleva a la necesidad de que todas las regiones u objetos presentes en la escena sean de tamaños similares, puesto que los resultados tenderán a ello.

Intuitivamente la preferencia por las uniones de regiones pequeñas incluso cuando la diferencia de color sea mayor es una de las principales desventajas del algoritmo RSST. El hecho de que incluso con espacios de color uniformes no se pueda garantizar que las distancias entre colores muy diferentes se correspondan con la percepción humana de uniformidad, es un agravante más.

Para facilitar la comprensión de la mejora, se parte de un ejemplo específico: existen dos posibles uniones de regiones; la primera de ellas es entre dos grandes regiones con color muy similar y la segunda dos regiones pequeñas pero con colores dispares; pero ambas producen el mismo coste de acuerdo a la **ecuación 4.1**.

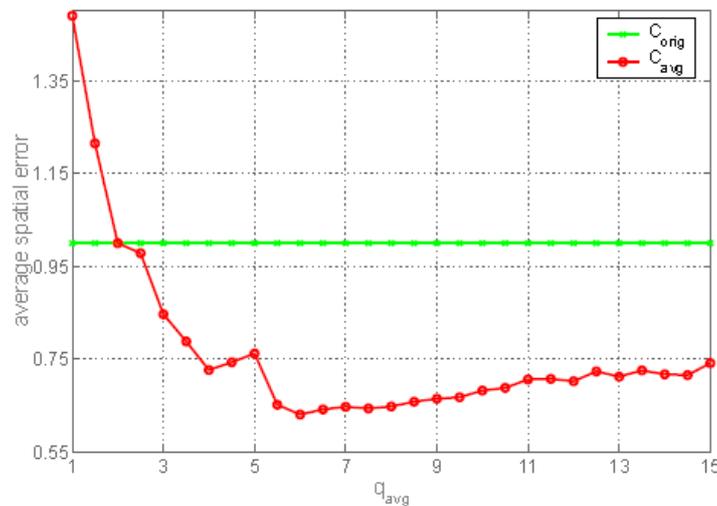
La idea es introducir una modificación sobre dicha ecuación, de manera que la diferencia de color suponga una contribución mayor que los respectivos tamaños para el coste final. Por tanto, dará como resultado un valor de coste inferior para la primera unión del ejemplo propuesto.

La propuesta es una fórmula más adecuada para equilibrar la diferencia de color y la dependencia de tamaño [11], definiendo para ello el parámetro  $q_{avg}$  que se calcula experimentalmente:

$$C_{avg}(i, j) = \left\| c_i - c_j \right\|_2^{q_{avg}} \frac{a_i a_j}{a_i + a_j} \quad (4.2)$$

Este nuevo parámetro introducido fue evaluado posteriormente sobre la colección de imágenes utilizada y en la **Figura 4.1** se muestran los resultados comparados con los de la fórmula original. Se puede observar que para valores superiores a 4.0 del parámetro  $q_{avg}$  se obtiene una disminución considerable del error espacial medio. Como se observa en la **Figura 4.1**, el mínimo error de segmentación se da para  $q_{avg} = 6$ , sin embargo esto será específico para la colección de imágenes, método de evaluación y criterio de parada utilizados.

Se puede observar que el error se reduce de 1 a 0,64 aproximadamente, lo que es una mejora significativa teniendo en cuenta la trivial modificación.



**Figura 4.1** – Influencia del parámetro  $q_{avg}$  sobre el error medio espacial.

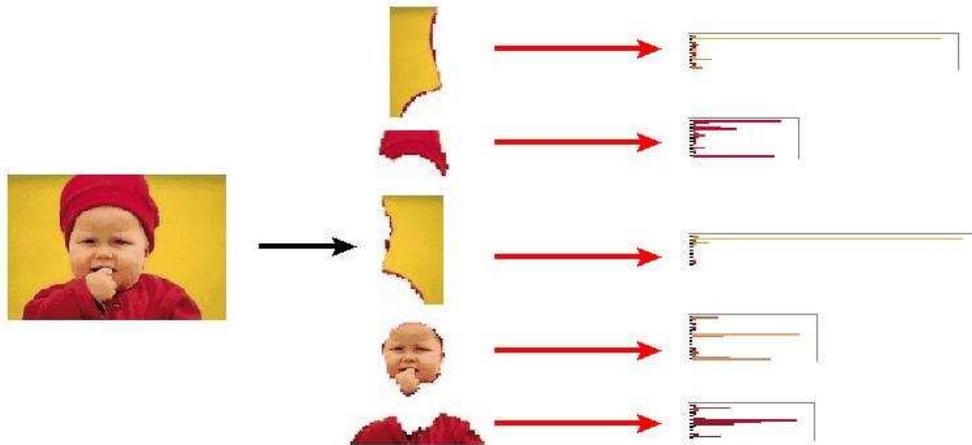
#### 4.4.3 - Representación extendida de color

En este apartado se presenta otra mejora para el criterio de homogeneidad de color, basada en la influencia del uso de los colores medios y la importancia de los tamaños de las regiones al fusionarlas.

En las primeras etapas del proceso de unión, los valores medios de las componentes de color, son suficientes para representar el color de las regiones. Sin embargo, según avanza el proceso y aumenta el tamaño de las regiones, los valores medios dejan de ser adecuados para caracterizar eficazmente el color de dicha región puesto que se pierde información. Para evitar este problema se propone un modelo de color alternativo que el autor denomina de *representación extendida*, consistente en una representación compacta y más precisa del color de las regiones mediante el uso de modelos de *Adaptive Distribution of Colour Shades* (ADCS) [8], o en castellano, distribución adaptativa de los matices de color. En este modelo, cada región contiene una lista de parejas formadas por el color y número de píxeles o población asignado a dicho color. Mediante este listado se representan de manera más precisa las variaciones de color de las regiones complejas – véase **Figura 4.2**.

Se propone como ejemplo el caso extremo de un tablero de ajedrez inicialmente segmentado en cuadrados blancos y negros. Mediante la utilización del método ADCS podríamos representar el color del mismo con total exactitud, utilizando solamente dos parejas color/población (una para el negro y otra para el blanco) sin embargo utilizando el color medio, el resultado sería un objeto completamente gris y habríamos perdido la capacidad de distinguir el contorno de los cuadrados.

El ADCS se ha utilizado en este método de la siguiente manera: después de una segmentación inicial, cada región caracterizada por su color medio, es convertida a la representación ADCS, esta conversión es trivial puesto que en esta etapa cada región contendrá un único par: color medio/tamaño de la región (píxeles). En la siguiente etapa, cuando dos regiones se hayan fusionado, el nuevo conjunto contiene una lista con las parejas de ambas regiones. Al final del proceso de segmentación, el color de la región vendrá definido por una lista de parejas correspondientes a una lista compuesta por las parejas iniciales. Algunas de las ventajas de esta representación extendida de color, son la facilidad de almacenamiento, y la rápida actualización tras una unión. Además no son necesarias suposiciones previas sobre la distribución implícita del color. La **Figura 4.2** muestra un ejemplo de representación extendida de color para una imagen.



**Figura 4.2** – Ejemplo de la aplicación del modelo de *Adaptive Distribution of Colour Shades* [8]

Como se ha demostrado en [8] dos caracterizaciones ADCS pueden ser eficazmente comparadas utilizando la distancia cuadrática [103]. Es decir, la diferencia de color entre las regiones  $r_i$  y  $r_j$  caracterizadas mediante ADCSs se podría calcular como la distancia cuadrática  $d_{quad}^2(I, J)$ , con:

$$d_{quad}^2(I, J) = I^T A^I I + J^T A^J J - 2I^T A^{IJ} J \quad (4.3)$$

donde las matrices  $A^I$ ,  $A^J$  y  $A^{IJ}$  expresan las similitudes de color entre las parejas de I, entre las de J, y entre las de I y J, respectivamente, en base a la distancia euclídea. Por ejemplo:

$$A^{IJ} = \left[ a_{c_i^I, c_w^J} \right] = 1 - \frac{d_{c_i^I, c_w^J}}{d_{\max}} \quad (4.4)$$

donde  $d_{c_v^I, c_w^J}^{I, J}$  será la distancia euclídea entre dos colores  $c_v^I$  y  $c_w^J$  representados en el espacio de color CIE LUV y  $d_{max}$  denota la distancia máxima del espacio.

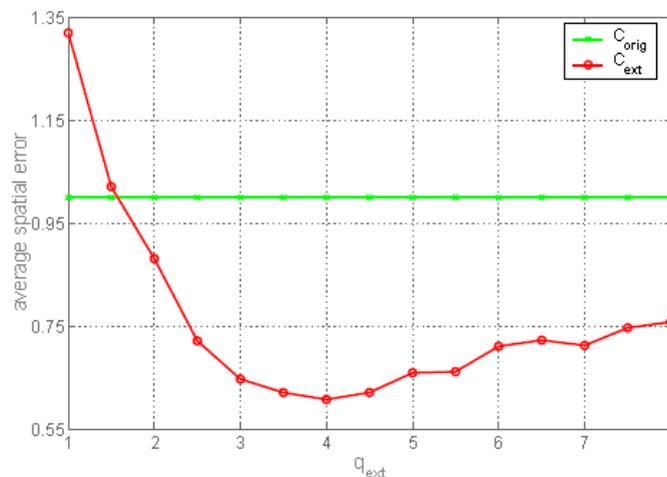
El coste computacional de la distancia cuadrática dependerá del número de parejas listadas para cada región. Sin embargo, esto puede acelerarse mediante una previa computación de los dos primeros términos de la **Ecuación 4.3**, puesto que solamente los dos últimos términos cruzados necesitan ser evaluados para comparar las representaciones. Para este caso específico, el coste computacional para el cálculo de la distancia cuadrática está limitado por el número de regiones obtenidas en la etapa inicial, por lo que es prácticamente insignificante. No obstante, el listado de parejas que caracteriza cada región podría reducirse mediante métodos de agrupación y cuantificación.

Por analogía con la **Ecuación 4.2** la función de costes de unión, basada en la representación extendida de color explicada a lo largo de esta sección, quedaría definida como:

$$C_{ext}(i, j) = \left[ d_{quad}^2(I, J) \right]^{q_{ext}} \frac{a_i a_j}{a_i + a_j} \quad (4.5)$$

La **Figura 4.3** muestra el error medio espacial obtenido en la segmentación para la colección completa tras la utilización de esta fórmula, combinada con distintos valores de  $q_{ext}$ . Como se puede observar el mínimo error medio se obtiene para valores cercanos a 4. Al igual que en el caso anterior la nueva modificación se traduce, en un error bastante inferior al obtenido con la fórmula original.

Hasta este momento sólo se ha tenido en cuenta la homogeneidad dada por el color. En vista de las múltiples variaciones adicionales que parecen ser aceptadas por el RSST original, el autor opta en su trabajo por añadir el análisis de ciertas medidas geométricas. Éstas se integrarán en el conjunto del algoritmo, convirtiéndose así en una nueva fuente de datos a analizar y procesar, es decir, una nueva fuente de información como veremos en el siguiente apartado.



**Figura 4.3** – Influencia del parámetro  $q_{ext}$  sobre el error medio espacial.

## 4.5 - Características Geométricas

Como ya se ha comentado anteriormente, existen muchos métodos a lo largo de la historia de la segmentación automática que han tratado de añadir distintos tipos de propiedades geométricas en distintas fases del proceso. El trabajo que se describe a continuación, desarrollado en [21], se basa principalmente en las conocidas como *Características Sintácticas Visuales* y en particular en el *Criterio de Cuasi-Inclusión* utilizado por Ferrán y Casas para la creación de Árboles de Partición Binarios [12].

Precisamente en este artículo [12] encontramos referencias a las características sintácticas visuales, entre las que se encuentran la homogeneidad (de color o de cualquier otra característica), compactibilidad, regularidad, inclusión y simetría. En esta sección revisaremos un conjunto de éstas y trataremos aspectos relacionados con su medida y su integración en el RSST original.

Se puede justificar fácilmente el uso de estas cualidades de la imagen echando un vistazo a lo que nos rodea: el “mundo real”. Los objetos que vemos son entes *regulares*, por lo general, y observándolos globalmente comprobamos que tienden a formas poco complejas. Además, los objetos también tienden a ser *compactos*, lo que se puede traducir en información muy interesante al estudiar los bordes entre regiones contiguas. Como veremos, analizando los entrantes y salientes existentes en los bordes, de las regiones producidas tras las primeras fases de segmentación, podremos simplificar las formas de los objetos. La existencia de objetos que contienen huecos podría considerarse un problema a la hora de poner en práctica este criterio. Sin embargo, una solución rápida pasa por una simplificación de la escena analizada sin eliminar los bordes más significativos, esto es, se considerará que las pequeñas regiones totalmente incluidas dentro del propio objeto se corresponden con partes insignificantes de éste. Otro aspecto bastante común, aunque quizás algo más discutible que los anteriores, es la *simetría* de los objetos y de las partes de éstos, en la cual también podríamos basarnos para el análisis de las formas.

Por tanto, midiendo la complejidad o la regularidad y la inclusión en los bordes de las formas de los objetos podemos obtener diversos datos que, siempre sujetos a un análisis adecuado, nos proporcionarán una información extra. La idea perseguida en definitiva es encontrar la manera adecuada de extraer cualquier tipo de información implícita en la imagen.

Aunque suelen tratarse por separado, existe una innegable vinculación entre todas estas características, lo que es fácil de comprobar tratando de imaginar un objeto con una alta complejidad y que al mismo tiempo sea compacto y/o de bordes regulares. Esta dependencia dificulta aún más si cabe la integración de varias medidas geométricas en un proceso de segmentación.

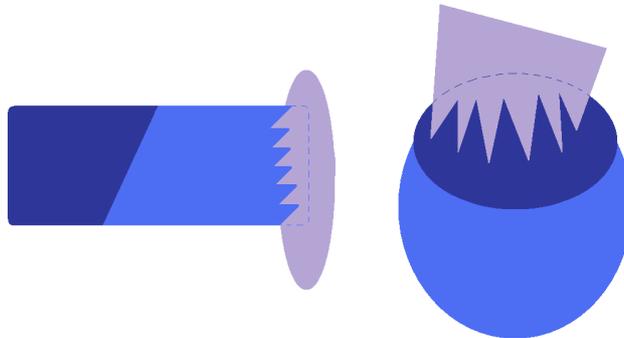
Tras el breve estudio acerca de las posibilidades de las características geométricas de los objetos y estando más familiarizados con las mismas, pasamos a definir a continuación dos medidas para la cuantificación de la contigüidad y la regularidad (o falta de complejidad); dentro del marco de trabajo del RSST original.

#### 4.5.1 - Adyacencia

Como ya hemos dicho anteriormente, consideramos un hecho generalizado que los objetos tienden a ser compactos en el mundo real. Dicho de otro modo, podemos esperar que las regiones sintácticas de una imagen muestren un alto grado de contigüidad. En el caso del algoritmo RSST original, la contigüidad de todas las partes viene impuesta intrínsecamente durante la primera fase de inicialización, en la que las regiones son consideradas adyacentes cuando al menos tienen un píxel colindante en sus respectivos bordes esto es, las regiones son o no son contiguas. Se propone una mejora para esta definición binaria, en la que el término adyacente se define de una manera algo más extendida y se puede medir con algo más de exactitud:

$$C_{adj}(i, j) = 1.0 - \frac{l_{ij}}{\min\{l_i, l_j\}} \quad (4.6)$$

donde  $l_{ij}$  es la longitud del borde que ambas regiones ( $r_i$  y  $r_j$ ) poseen en común y  $l_i$  y  $l_j$  son los perímetros de las mismas.

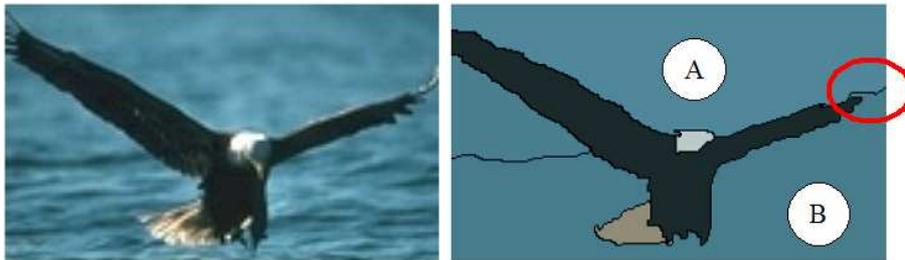


**Figura 4.4** – Ejemplo de un caso de jaggedness perjudicial al tener en cuenta la adyacencia.

Aunque sencilla, esta fórmula proporciona una información muy interesante acerca de la relación geométrica entre las regiones vecinas. Los valores de  $C_{adj}(i, j)$  próximos a cero indican una inclusión casi completa (el área en común de ambas regiones es prácticamente igual al perímetro de la más pequeña de ellas), mientras que valores cercanos a uno muestran una adyacencia mínima que reforzará la idea de no fusión entre las regiones, independientemente de su homogeneidad de color o tamaños.

Existe una característica de los bordes conocida por su término en inglés “jagged”, que se corresponde con una superficie dentada (estilo hoja de sierra) y que puede enmascarar bajos valores de  $C_{adj}$  – véase **Figura 4.4**. Si tomamos como referencia para unir regiones los valores de adyacencia más pequeños se beneficiará la unión de regiones cuyo borde común posea esta característica.

Sin embargo, como cualquier característica que se precie, la adyacencia también tiene excepciones en las que su uso empeora los resultados finales. Como ejemplo, podemos observar el caso particular de la **Figura 4.5** en el que la búsqueda de regiones compactas hace que se niegue la evidente existencia de un fondo homogéneo. En este caso, claramente la fuerte homogeneidad de color debería pesar sobre la característica de adyacencia, en el caso ideal de un método que tuviera distintas fuentes de evidencias y la capacidad de ponderarlas para cada caso. Considerando ahora las regiones A y B, podemos observar que la longitud del borde que ambas comparten es mínima comparada con el perímetro de cualquier de ellas. Esto dará un valor de  $C_{adj}$  cercano a uno, lo que se interpreta como un claro indicio para no llevar a cabo la unión. Obviamente, esto conlleva a un error, fácil de subsanar si se tuviera en cuenta la similitud del color.



**Figura 4.5** – Ejemplo en el que la adyacencia empeora el resultado, negando la existencia de un fondo[11].

#### 4.5.2 – Regularidad o Ausencia de Complejidad

La regularidad (o la falta de complejidad) es otra de las características que viene dada por los habituales rasgos de los objetos en el mundo real. Para estudiar la complejidad de la forma existen dos posibilidades: hacerlo de manera local o global. Para el método presentado se tendrá en cuenta el análisis global, tal y como se justifica más adelante.

El término *complejidad*, aunque usado muy a menudo en el contexto del análisis de formas, no suele estar bien definido. Un trabajo interesante sobre la percepción de la complejidad de las formas es [104]. En él, la medida de la complejidad se basa en medidas de la entropía de las distancias globales, de los ángulos locales, etc. Para no perder la sencillez del conjunto de características definidas hasta ahora diseñadas para su integración en el método RSST original, el autor del algoritmo define una medida más sencilla, sin descartar la implementación de medidas de complejidad más avanzadas como parte del trabajo futuro.

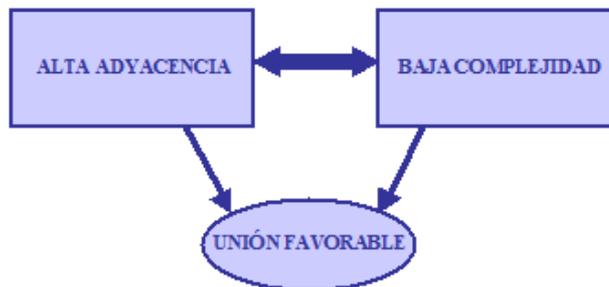
#### 4.5.2.1 - Complejidad de los Contornos Globales

Lógicamente, las formas complejas tendrán un perímetro mayor que las formas más sencillas, en proporción con su área total. Por esta razón podemos definir la complejidad  $x_i$ , de una región  $r_i$  como la razón entre la longitud de su perímetro  $l_i$ , y la raíz cuadrada de su área  $a_i$ . Esta proporción también es utilizada a menudo en la literatura [104] para cuantificar el grado de compactibilidad, con una cierta dependencia del grado de complejidad del contorno local.

Los cambios de la complejidad de la forma global para un par de regiones vecinas  $r_i$  y  $r_j$  provocados por la fusión de ambas se pueden medir con la siguiente fórmula:

$$C_{cpx}(i, j) = \frac{x_{ij}}{\left[ \frac{a_i x_i + a_j x_j}{a_i + a_j} \right]} \quad (4.7)$$

donde  $x_{ij}$  denota la complejidad de la hipotética región formada al unir  $r_i$  y  $r_j$ . El denominador se puede interpretar como la media de las complejidades de cada región por separado  $x_i$  y  $x_j$ , ponderadas en función de sus respectivas áreas.



**Figura 4.6** - Esquema representativo de la dependencia entre las características sintácticas

Valores bajos de  $C_{cpx}$  se interpretan como indicios de que la fusión es favorable respecto a la complejidad de las formas. En concreto, para valores menores de uno, sabremos que la complejidad de la región resultante de la unión, es menor que la media de de las primeras. Y por tanto, la unión será beneficiosa cuando se trata de obtener una región más regular.

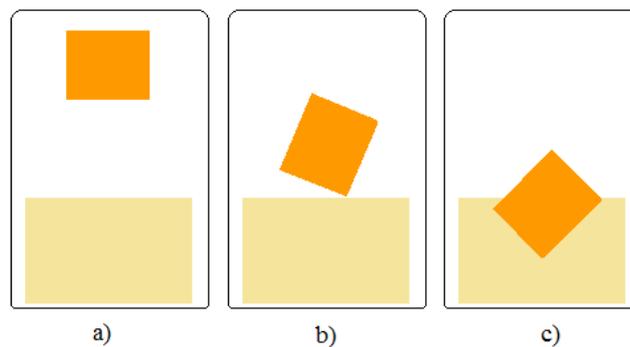
La dependencia de esta característica con la anterior (adyacencia) radica en que según sea mayor el borde común entre las dos primeras regiones, será más probable que la complejidad del resultado de la unión de ambas sea menor. Ambas características trabajaran en la misma dirección, es decir, ante el mismo escenario nos proporcionan las mismas evidencias - véase **Figura 4.6**.

#### 4.5.2.2 - Complejidad de los Contornos Locales

A pesar de no ser algo habitual encontrar objetos con contornos dentados, el algoritmo RSST básico tiende a producirlos entre regiones que no se diferencian con claridad. Por tanto,

esta característica de bordes irregulares podría ser utilizada para detectar trazados artificiales y obtener de ellos indicios de unión entre las regiones que los comparten.

Debe tenerse en cuenta el fenómeno conocido como Principio de Transversalidad [105], que hace referencia al hecho de que también se darán bordes con curvaturas muy pronunciadas, incluso con picos, cuando los objetos se superpongan entre ellos (ver **Figura 4.7**). En [34,100] se llevó a cabo un intento de distinción entre estos salientes o picos y los producidos artificialmente por el propio algoritmo de segmentación (*jaggedness*), integrando esta información como una nueva fuente también en el RSST. La eficiencia del método no está demostrada puesto que sólo se muestran resultados experimentales con una imagen.



**Figura 4.7** – Principio de Transversalidad. Ejemplo de una secuencia de objetos que se superponen.

En un enfoque inicial de este algoritmo [35] fue abordada la detección de los bordes dentados y la utilización explícita de esta información dentro del marco de integración propuesto a lo largo de este capítulo. A pesar de que se obtuvieron resultados satisfactorios para multitud de imágenes, experimentos posteriores revelaron una gran cantidad de fusiones erróneas. De esta experimentación el autor dedujo que previamente al uso de la característica de complejidad de los contornos locales, es necesario abordar algunos problemas como:

- Elección de una escala adecuada para limitar la complejidad local.
- Establecer si la detección de esquinas aisladas con curvaturas muy pronunciadas es más útil que la detección de los bordes dentados para determinar la fusión o la no fusión de dos regiones.
- Encontrar una manera fiable de distinguir las superficies dentadas artificiales de las curvaturas pronunciadas creadas por la superposición de objetos.

Estos problemas, unidos al hecho de que tanto la medida de adyacencia como la de complejidad global pueden detectar las irregularidades en el borde en común de dos regiones, hicieron que el autor se decantara finalmente por no incluir el análisis de la complejidad local en el RSST modificado.

## 4.6 – Diseño e Integración de Múltiples Fuentes

La integración de varias fuentes de información en una única regla para la unión de regiones se presenta como una tarea poco trivial. Aunque las generalidades no son nunca apropiadas, se puede afirmar con cierta rotundidad que la homogeneidad de color es la característica más fiable (intuitivamente, mucho más que cualquier otra característica geométrica). La adyacencia o la regularidad descritas en los apartados anteriores, serán utilizadas como evidencias auxiliares a la hora de calcular el coste de la unión entre dos regiones. Este será un aspecto primordial a tener en cuenta en la integración de diversas fuentes en un mismo algoritmo: la fiabilidad que se le otorgará a cada una de ellas.

Además, como ya vimos anteriormente, existen casos en los que las características geométricas no podrán ser medidas adecuadamente y por tanto en los que deberían ser desestimadas las deducciones obtenidas de éstas. Podemos imaginar el caso en el que al fusionar dos regiones, se obtenga una tercera de menor complejidad, lo que se interpreta como un indicio de que esa fusión es aconsejable. Sin embargo, si la fusión implica al mismo tiempo la supresión de un borde relevante de la imagen, el algoritmo debe anteponer este criterio al de la regularidad de las formas. En el **apartado 4.6.4** se explica detalladamente cómo se lleva a cabo esta distinción en el algoritmo M-RSST.

Por todo lo detallado arriba, el autor optó por el uso de la llamada *Teoría de la evidencia* [106, 107], que aporta un modelo adecuado para manejar información proveniente de fuentes inciertas.

### 4.6.1 - Teoría de la Evidencia

En tiempos pasados era habitual utilizar la teoría de la probabilidad para tratar con datos inciertos, a pesar de su demostrada falta de efectividad en ciertos casos [108]. Posteriormente, se propusieron otros modelos para el manejo de datos imprecisos, como por ejemplo, *Theory of Fuzzy Sets* [109], *Possibility Theory* [110], o para información incierta, *Teoría de la Evidencia* [106, 107]. En este apartado se describe la manera de aplicar la Teoría de la Evidencia (Belief Theory - *BeT* de ahora en adelante) para la integración de varias medidas distintas en el algoritmo RSST mejorado que se describe en este capítulo.

La *BeT* fue introducida por Dempster [106] y Shafer [107]. Posteriormente, Smets [98, 111] propuso una versión revisada denominada Transferable Belief Models (TBM). Sin embargo, el autor de este algoritmo consideró aceptable la primera versión por las siguientes razones:

- Un adecuado tratamiento de las incertidumbres, modelando explícitamente “la duda” o falta de certeza.
- La capacidad de definir distintos grados de fiabilidad de las fuentes.

- Diferenciación entre información desconocida, imprecisa o conflictiva.

En definitiva, la BeT es una herramienta flexible y generalizada que permite representar distintos estados de conocimiento de la información que no podríamos modelar con una teoría común de probabilidad.

#### 4.6.2 – Introducción a la teoría de Dempster-Shafer

Esta ha sido una teoría típicamente utilizada en situaciones en las que se debe tener en cuenta información procedente de distintas fuentes y para tomar una decisión, como por ejemplo en un diagnóstico médico [112], reconocimiento y clasificación [113, 114]. En este caso se utilizará en la computación de las distintas fuentes y medidas vistas hasta ahora para el cálculo del coste de unión de distintas regiones durante el proceso de segmentación.

- **FD – Frame of Discerment**

Definimos  $\Omega$  como un conjunto finito de  $N$  hipótesis  $H_x$ :

$$\Omega = \{H_1, \dots, H_N\} \quad (4.8)$$

- **BBA – Basic Belief Assignmetns**

Por otro lado, se define  $2^\Omega$  como el conjunto de todas las posibles combinaciones de hipótesis,  $2^N$  proposiciones  $A$  combinatorias de  $\Omega$ , como:

$$2^\Omega = \{A_1, A_2, \dots, A_{2^N}\} = \{0, \{H_1\}, \dots, \{H_N\}, \{H_1 \cup H_2\}, \{H_1 \cup H_3\}, \dots, \Omega\} \quad (4.9)$$

Por último, se definen  $U$  fuentes de evidencias (o medidas) denominadas  $S_1, S_2, \dots, S_U$ . La distribución de los datos provenientes de la fuente  $S_U$  se modela con la variable  $m_u$  y se conoce con el término *Basic Belief Assignment* (BBA):

$$m_u : 2^\Omega \longrightarrow [0, 1],$$

$$\text{con } m_u(0) = 0 \text{ y } \sum_{A \subseteq \Omega} m_u(A) = 1.$$

En otra palabras, existirá una distribución  $m_u$  asociada a cada símbolo perteneciente al conjunto  $2^\Omega$  y para cada fuente  $S_U$ .

- **Combinación de Fuentes**

El operador utilizado para la combinación de las fuentes es la suma ortogonal, también conocida como la regla combinatoria de Dempster [111, 115]. El operador  $m^\oplus$ , conmutativo y asociativo, se define como:

$$m^\oplus = m_1 \oplus m_2 \oplus \dots \oplus m_U, \quad (4.10)$$

donde la representación combinada para dos fuentes de información  $S_u$  y  $S_v$  sería determinada por la siguiente ecuación:

$$\forall A \subseteq \Omega \quad m^\oplus(A) = \frac{1}{1-K} \sum_{B \cap C = A} m_u(B) \cdot m_v(C), \quad (4.11)$$

donde el primer término será un factor de normalización y  $K$  representa una medida del conflicto entre las fuentes combinadas ( $m^\oplus(\emptyset)$ ) y se define como:

$$K = \sum_{B \cap C = \emptyset} m_u(B) \cdot m_v(C) \quad (4.12)$$

- **Fiabilidad de las Fuentes**

Cuando una fuente no es considerada cien por cien válida o fiable, ésta podrá atenuarse por un factor  $\alpha_u \in [0,1]$ . La nueva estructura atenuada  $m_u^{\alpha_u}$  se representará como:

$$m_u^{\alpha_u} = \alpha_u m_u(A) \quad \forall A \neq \Omega \quad (4.13)$$

y

$$m_u^{\alpha_u}(\Omega) = 1 - \alpha_u + \alpha_u m_u(\Omega) \quad (4.14)$$

### 4.6.3 – Aplicación a la unión entre regiones

En esta sección se describen los aspectos principales para la aplicación de la BeT en el problema de la integración de múltiples características durante el proceso de decisión de la unión de regiones en la segmentación.

- **FD – Frame of Discerment**

En el caso específico que nos atañe, el espacio  $\Omega$  estará compuesto por dos hipótesis  $A_1$  y  $A_2$ , que se denominarán MERGE y DONTMERGE y serán excluyentes entre sí. Por tanto:

$$\Omega = \{MERGE, DONTMERGE\} \quad (4.15)$$

- **BBA – Basic Belief Assignmetns**

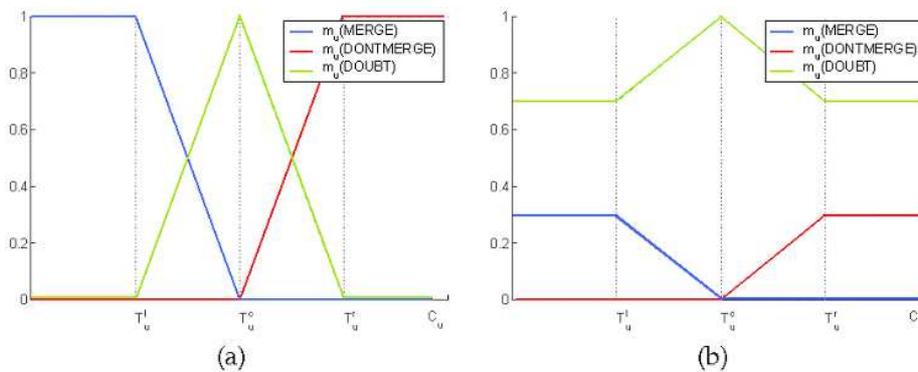
Puesto que  $N = 2$  el conjunto  $2^\Omega$  de  $\Omega$  estará formado por 4 proposiciones:

$$2^\Omega = \{\emptyset, \{MERGE\}, \{DONTMERGE\}, \{MERGE \cup DONTMERGE\}\} \quad (4.16)$$

De aquí en adelante nos referiremos a la proposición  $\{MERGE \cup DONTMERGE\}$  como  $\{DOUBT\}$ .

Siguiendo con las variables expuestas en el apartado anterior, tendremos  $U$  fuentes ( $S_1, S_2, \dots, S_U$ ) de las cuales  $S_1$  siempre será la homogeneidad de color y el resto podrán denotar cualquier otra característica relacionada con la unión de regiones. Una medida  $C_u$  procedente de una fuente  $S_u$  será modelada por la variable  $m_u$ , definida formalmente en el apartado anterior. Dicho de otro modo, para cada medida  $C_u$  (por ejemplo, de homogeneidad de color) el modelo  $m_u$ , interpretará cada uno de sus valores como una proposición  $A$  contenida en  $2^\Omega$ .

Las distribuciones o estructuras denominadas BBA se obtienen en base a conocimientos de la aplicación, a datos estadísticos, o a ambos a la vez, como es el caso de este algoritmo. En la **Figura 4.8a** podemos observar la representación de estas distribuciones para una fuente  $S_u$ . El valor neutral de  $C_u$  denotado como  $T_u^c$ , es el punto en el que no existen evidencias que decanten la decisión hacia ninguna de las dos opciones (unión o no unión). Por tanto, ese punto será el máximo para la BBA correspondiente al evento DOUBT y cero tanto para MERGE como para DONTMERGE. Desde ese punto, según los valores  $C_u$  aumentan (disminuyen) hacia  $T_u^l$  ( $T_u^r$ ) el valor asociado a DOUBT decrece, aumentando el de MERGE (DONTMERGE). Una vez sobrepasado el valor  $T_u^l$  (o por debajo de  $T_u^r$ ) se obtendrán valores indicadores del suceso MERGE (DONTMERGE) y el valor de DOUBT será igual a cero.



**Figura 4.8** – Ejemplo de las BBAs sin considerar (a) y considerando (b) la fiabilidad de las fuentes [11].

- **Fiabilidad de las Fuentes**

En la **Figura 4.8b** se muestra un ejemplo de la BBA resultante cuando se tiene en cuenta una cierta fiabilidad en las fuentes de la **Figura 4.8a**. El factor atenuante, que indica la credibilidad de ambas fuentes, es  $\alpha_u = 0.3$  para este ejemplo. Se puede observar que los valores de los eventos MERGE y DONTMERGE disminuyen y aumentan proporcionalmente al valor del suceso unión, DOUBT. La dificultad reside por tanto en determinar adecuadamente los valores de cada uno de los parámetros – ver **Tabla 4.1**.

- **Combinación de Fuentes**

A pesar de que el resultado de combinar las BBAs procedentes de varias fuentes es bastante intuitivo en este caso, se muestran a continuación unos ejemplos para aplicar las fórmulas de combinación de fuentes de Dempster. Se supone un escenario similar al anterior en el que se combinan las BBAs para dos fuentes de información  $S_u$  y  $S_v$ , del cual se obtendrían las siguientes expresiones:

$$m^{\oplus}(MERGE) = \frac{1}{1-K} (m_u(MERGE) \cdot m_v(MERGE) + m_u(MERGE) \cdot m_v(DOUBT) + m_u(DOUBT) \cdot m_v(MERGE)) \quad (4.17)$$

$$m^{\oplus}(DONTMERGE) = \frac{1}{1-K} (m_u(DONTMERGE) \cdot m_v(DONTMERGE) + m_u(DONTMERGE) \cdot m_v(DOUBT) + m_u(DOUBT) \cdot m_v(DONTMERGE)) \quad (4.18)$$

$$m^{\oplus}(DOUBT) = \frac{1}{1-K} (m_u(DOUBT) \cdot m_v(DOUBT)) \quad (4.10)$$

$$K = m_u(MERGE) \cdot m_v(DONTMERGE) + m_u(DONTMERGE) \cdot m_v(MERGE) \quad (4.20)$$

$T_u^c$	Punto máximo de incertidumbre, asociada al suceso <i>DOUBT</i> .
$T_u^l$	Punto que delimita la inflexión en la curva del suceso <i>MERGE</i> .
$T_u^r$	Punto que delimita la inflexión en la curva del suceso <i>DONTMERGE</i> .
$\alpha_u$	Valor indicativo de la fiabilidad de la fuente $S_u$ .

**Tabla 4.1** – Parámetros utilizados para el diseño de las BBA y para medir la credibilidad de las fuentes.

- **Función Coste de la Unión**

En definitiva la función del coste de la unión entre dos regiones vecinas  $r_i$  y  $r_j$  basada en una o más evidencias se define utilizando la fórmula:

$$C_{total}(i, j) = m_{(i,j)}^{\oplus}(DONTMERGE) - m_{(i,j)}^{\oplus}(MERGE), \quad (4.21)$$

donde  $m_{(i,j)}^{\oplus}(MERGE)$  y  $m_{(i,j)}^{\oplus}(DONTMERGE)$  serán la suma ortogonal de las distintas fuentes asociadas (homogeneidad de color, adyacencia y complejidad de la forma) al evento “unión” y “no unión”, respectivamente. Como se observa en la **Ecuación 4.21**, un valor mayor de la proposición MERGE (DONTMERGE) conllevará a un valor menor (mayor) de  $C_{total}$ . La proposición DOUBT no contribuye explícitamente al valor final pero si se utilizará en el cálculo de las anteriores.

#### 4.6.4 – Diseño de las BBAs para cada fuente

El diseño de las BBAs para cada fuente de información  $S_u$  implica la definición los parámetros  $T^l$ ,  $T^c$ ,  $T^r$  y del factor de atenuación  $\alpha_u$  que fueron calculados experimentalmente. En

esta sección se muestra el planteamiento general seguido por el autor para dicha estimación y se describe en detalle las estructuras de las BBA de cada una de las fuentes utilizadas: homogeneidad de color, adyacencia y complejidad.

#### 4.6.4.1 - Estimación de los Parámetros $T^l$ , $T^c$ y $T^r$

Dada una fuente cualquiera, que aporte medidas  $C_u$  de una cierta propiedad relacionada con la fusión de regiones, los umbrales  $T$  serán calculados analizando y obteniendo estadísticas procedentes de ejemplos con resultados conocidos. Esto es factible utilizando una colección de imágenes y sus respectivas máscaras de referencia a modo de entrenamiento. Se realizará una clasificación automatizada de cada posible unión en dos categorías: “fusión realizable” y “fusión no realizable”.

La colección de uniones etiquetadas se realiza del siguiente modo: cada imagen es segmentada utilizando el RSST original [102], pero añadiendo una condición de fusiones prohibidas (aquellas que no “concurden” con las máscaras de referencia); cuando el número de regiones es menor que un cierto umbral se pasa a la segunda fase del proceso (habitualmente se utilizan 100 regiones), en el cual comienzan a almacenarse todas las propiedades importantes (color, características geométricas y categoría: *fusión realizable* o *fusión no realizable*) de todas las posibles fusiones entre regiones.

Computando todos los datos almacenados se pueden estimar los valores de  $T_u^l, T_u^c$  y  $T_u^r$ : puesto que el parámetro  $T_u^l$  ( $T_u^r$ ) se define idealmente como el punto que delimita la zona del máximo para el evento MERGE (DONTMERGE), podemos determinar el valor de este a partir de la tabla de propiedades anteriormente almacenada como el mínimo (máximo) valor existente de  $C_u$  perteneciente a la categoría *fusión realizable* (*fusión no realizable*). En realidad, para evitar que un valor aislado pueda perjudicar la estimación el valor final de los parámetros se corresponderá con la media de todos los valores obtenidos para cada imagen. Considerando que la colección completa de imágenes  $T$  contiene  $N$  imágenes, los valores  $\hat{T}_u^{l,(I)}$  y  $\hat{T}_u^{r,(I)}$  para una imagen  $I \in T$  vendrán dados por las siguientes expresiones:

$$\hat{T}_u^{l,(I)} = \min_{\forall (i,j) \in \beta^{(I)}} C_u(i, j) \quad (4.22)$$

y

$$\hat{T}_u^{r,(I)} = \max_{\forall (i,j) \in \zeta^{(I)}} C_u(i, j) \quad (4.23)$$

donde  $(i,j)$  se refiere al par de regiones vecinas  $(r_i, r_j)$  y las categorías *fusión realizable* y *fusión no realizable* son indicadas por  $\zeta^{(I)}$  y  $\beta^{(I)}$ , respectivamente. Finalmente, los valores

definitivos de  $\hat{T}_u^l$  y  $\hat{T}_u^r$  se computarán como las medias de los valores parciales obtenidos de cada una de las imágenes de la colección:

$$\hat{T}_u^l = \frac{1}{N} \sum_{\forall I \in \mathcal{T}} \hat{T}_u^{l,(I)} \quad (4.24)$$

y

$$\hat{T}_u^r = \frac{1}{N} \sum_{\forall I \in \mathcal{T}} \hat{T}_u^{r,(I)} \quad (4.25)$$

El parámetro  $T_u^c$  que se corresponde con el valor máximo del evento DOUBT podría calcularse de manera similar pero el autor del algoritmo simplifica calculándolo como el punto intermedio entre los anteriores.

#### 4.6.4.2 - Estimación del Parámetro de Atenuación $\alpha$

En primer lugar y por simplicidad, el factor de atenuación correspondiente a la evidencia dada por la homogeneidad de color,  $\alpha_1$  se fija a uno puesto que se considera la más fiable de todas las fuentes utilizadas en el algoritmo. Para el resto de fuentes se calcula de manera similar a los anteriores, buscando los valores de dichos parámetros que minimicen el error espacial. Este procedimiento intuitivo produce unos resultados bastante buenos aunque como el propio autor reconoce, sería posible una solución alternativa.

En los siguientes apartados se reflejan los valores obtenidos de todos estos parámetros, para cada uno de los criterios que se han descrito anteriormente, explicando las individualidades y características propias.

#### 4.6.4.3 – BBA para la Homogeneidad de Color

En la **sección 4.4** fueron definidas dos mejoras para el criterio de homogeneidad de color presentado originalmente para el algoritmo RSST, cuya eficacia ha sido comprobada por el autor del algoritmo: el criterio de color optimizado ( $C_{avg}$ ) y la representación extendida de color ( $C_{ext}$ ).

Ambas medidas son dependientes del tamaño por lo que el primer paso para su integración como fuente de evidencias será normalizarlas por el tamaño de la imagen,  $A_I$  (sin que esto afecte al proceso de unión):

$$C_{avg}'(i, j) = \frac{C_{avg}(i, j)}{A_I} \quad \text{y} \quad C_{exp}'(i, j) = \frac{C_{exp}(i, j)}{A_I} \quad (4.26) \text{ y } (4.27)$$

Como se puede observar sobre la **Figura 4.9a**, cualquier medida por debajo del valor de  $T^l$  se asociará al evento: MERGE. Para asegurar que ni la mínima diferencia de color deje de tenerse en cuenta, el valor de este parámetro se fijó a cero para todas sus variantes – véase **Figura 4.9b**. Este razonamiento es además apoyado por los resultados obtenidos tras el procedimiento automático explicado anteriormente. El conjunto completo de valores estimados

que definen las estructuras BBA para ambas medidas de homogeneidad de color se muestran en la **Tabla 4.2**.

	$T^l$	$T^c$	$T^r$	$\alpha$
$C_{avg} (q_{avg}=6.0)$	0.0	$2.1 \times 10^8$	$4.2 \times 10^8$	1.0
$C_{ext} (q_{ext}=4.0)$	0.0	$9.4 \times 10^{-5}$	$18.8 \times 10^{-5}$	1.0

**Tabla 4.2** – Valores estimados para cada parámetro de la BBA de los criterios de homogeneidad de color.

	$[C_{avg}]$	$[C_{avg}, C_{cpx}]$	$[C_{ext}]$	$[C_{avg}, C_{cpx}]$
$\alpha_{adj}$	0.4	0.4	1.0	1.0

**Tabla 4.3** – Valores estimados para el valor de atenuación  $\alpha_{adj}$

	$[C_{avg}]$	$[C_{avg}, C_{cpx}]$	$[C_{ext}]$	$[C_{avg}, C_{cpx}]$
$\alpha_{cpx}$	0.9	1.0	1.0	0.5

**Tabla 4.4** – Valores estimados para el valor de atenuación  $\alpha_{cpx}$

Intuitivamente, se puede demostrar que utilizando sólo el criterio de homogeneidad de color en el marco de trabajo descrito (BeT sin fuentes adicionales de evidencias), los resultados obtenidos serán los mismos que al aplicar directamente el criterio. Para ello, recordamos la **Ecuación 4.21**, en la que para el caso de una única fuente, el operador  $m^\oplus$  se reduce a la BBA de dicha fuente y por tanto, el valor  $C_{total}(i,j)$ , aumenta y disminuye proporcionalmente al valor de la medida dada por  $C_{avg}$  o  $C_{ext}$ , en el rango  $[0, T^r]$ . Al añadir una segunda fuente de evidencias, su contribución vendrá determinada por el valor del evento DOUBT de la primera fuente. Es decir, una nueva fuente contribuirá siempre y cuando la información procedente de la homogeneidad de color no sea concluyente (valores cercanos a  $T^r$  en una BBA ideal). Por todo esto, se puede considerar la integración de múltiples fuentes como una extensión o generalización de la función de costes basada únicamente en la homogeneidad de color.

#### 4.6.4.4 – BBA para la Adyacencia

Los valores  $T_{adj}^l, T_{adj}^c$  y  $T_{adj}^r$  calculados automáticamente para esta fuente son 0, 0.5 y 1 respectivamente. Estos valores son aparentemente lógicos teniendo en cuenta la definición dada de adyacencia en el **apartado 4.5.1**: los valores cercanos a cero indican una casi-inclusión mientras que los valores cercanos a uno significan una débil contigüidad.

Los valores estimados para el factor de fiabilidad  $\alpha_{adj}$  en combinación con los dos criterios de homogeneidad de color y con el de regularidad se pueden observar en la **Tabla 4.3**.

#### 4.6.4.5 – BBA para la Complejidad de Contornos Globales

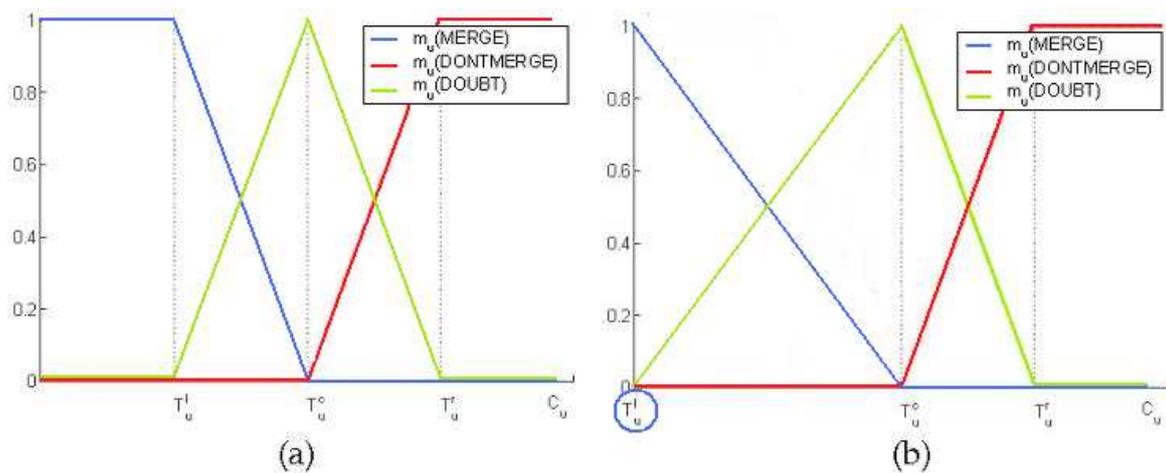
La medida de la regularidad (o ausencia de complejidad) en los contornos globales definida en el apartado 4.5.2.1, implica una estructura algo más flexible para su BBA. Esto se debe al hecho de que su fiabilidad dependa directamente de la longitud del contorno que comparte la hipotética región, resultante de la fusión de otras dos, con el borde de la imagen. Por ejemplo, en el caso de que una región, producto de una fusión de otras dos, comparta una parte significativa de su borde con el contorno de la imagen la información dada por la medida  $C_{cpx}$  no deberá ser tenida en cuenta. Este problema es fácil de abordar mediante el uso de las proposiciones definidas, atenuando el valor para los eventos MERGE y DONTMERGE y favoreciendo proporcionalmente la proposición DOUBT. Esta atenuación viene dada por:

$$\alpha'_{cpx}(i, j) = 1.0 - \frac{l_{ij}}{l_{ij}} \quad (4.28)$$

donde  $l_{ij}$  denota el perímetro del área  $r_{ij}$  (resultado de la unión de  $r_i$  y  $r_j$ ) y  $l_{ij}$  será el borde compartido con la imagen completa.

El resto de parámetros habituales son estimados con el método explicado, pero teniendo en cuenta solo aquellos casos que cumplía la siguiente condición, con el objetivo de ignorar la influencia negativa de los casos adversos:  $\alpha'_{cpx}(i, j) > 0.5$ .

Así, los valores obtenidos para  $T_{cpx}^l, T_{cpx}^c$  y  $T_{cpx}^r$  fueron 0.7, 1.1 y 1.4, respectivamente. Estos valores son igualmente lógicos pues siguen las premisas ya establecidas en la definición de la medida de complejidad de contornos globales. En la **Tabla 4.4** se reflejan los valores estimados para la fiabilidad de esta fuente al combinarla con las de homogeneidad de color.



**Figura 4.9** – a) BBA original para el criterio de homogeneidad de color y b) BBA optimizada para no despreciar la mínima diferencia de color.

## 4.7 – Criterio de Parada

En este punto se desarrolla una discusión a cerca de la posible mejora de un método de segmentación dado (como por ejemplo el RSST-Modificado descrito en esta sección) mediante el diseño adecuado del criterio de parada. Para ello se presentará en primer lugar un criterio basado en la búsqueda del pico de la señal o relación señal a ruido (conocido por sus siglas en inglés PSNR - Peak Signal to Noise Ratio) entre la señal original y la segmentada. Y posteriormente se describe una nueva propuesta por T. Adamek en [116].

### 4.7.1 – Criterio de Parada Basado en el PSNR

En este caso, la partición simple de la imagen se basa en el valor del PSNR (Peak Signal to Noise Ratio) calculado entre la original y el resultado de la segmentación – se puede encontrar un ejemplo en [41]. Todas las uniones llevadas a cabo durante este proceso (una por iteración) se almacenarán en un árbol de partición binario (BPT) junto con los respectivos valores del PSNR. Posteriormente, se obtendrá la partición fácilmente a partir del BPT desactivando aquellos nodos que superen un cierto umbral establecido para el PSNR. Una posibilidad para calcularlo en una imagen de  $M \times N$  píxeles es siguiendo la siguiente expresión:

$$PSNR = 20 \log_{10} \frac{MAX_L}{\sqrt{\frac{1}{M \cdot N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (L(i, j) - K(i, j))^2}} \quad (4.29)$$

donde L denota la luminancia de la imagen original y K la de la imagen segmentada (previamente reconstruida con sus intensidades medias). MAXL será el valor máximo de la luminancia.

El principal inconveniente de este método es que el valor del PSNR no refleja adecuadamente la similitud entre las imágenes a comparar y además sería difícil encontrar un umbral apto al mismo tiempo para imágenes altamente texturizadas como para imágenes con objetos de colores similares.

### 4.7.2 – Criterio de Parada Propuesto

Este criterio publicado por T. Adamek en [116] y añadido en el M-RSST de la manera que se describe a continuación, se presenta como una alternativa al criterio basado en la medida del PSNR.

Una diferencia esencial con el método anterior es que no se comparan los valores obtenidos en cada iteración con un mismo valor de umbral sino que se toma la decisión en

función de la evolución de la función acumulativa del coste de unión. Al igual que en el apartado anterior se hace uso de los BPTs para lograr la imagen resultado.

Definimos la función de coste acumulada,  $C_{cum}$ , como aquella que mide el coste total de todas las uniones acontecidas hasta lograr  $t$  de regiones:

$$C_{cum}(t) = \begin{cases} \sum_{n=t}^{N_I-1} C_{mrg}(n) & \text{if } 1 \leq t < N_I \\ 0.0 & \text{otherwise} \end{cases} \quad (4.30)$$

donde  $C_{mrg}(n)$  denota el coste de unión de dos regiones pasando de  $n+1$  a  $n$  regiones midiéndolo con un criterio dado.  $N_I$  indica el número de regiones en la partición inicial, llevada a cabo en la primera fase del proceso, por lo que solo los costes de las uniones realizadas en la segunda fase se tendrán en cuenta y contribuirán al valor final de  $C_{cum}$ .

La **Figura 4.10** muestra las medidas obtenidas para  $C_{cum}$  en cada iteración  $t$  para tres escenarios diferentes de segmentación. La idea básica de este método se resume en encontrar el número de regiones  $t_s$  que divida la curva en dos regiones de manera que, en el intervalo  $[1, t_s]$  ésta decrezca con una pendiente mucho mayor a la del intervalo  $[t_s, N_I]$ . Se puede desequilibrar para así provocar una sub o sobre-segmentación desplazando el valor de  $t_s$ .

Para encontrar este punto, el autor utiliza el método descrito en [117] de umbrales de dos niveles y diseñado para trabajar con distribuciones unimodales (umbrales basados en histogramas). El principio básico de este algoritmo es la asunción de que el modo de la distribución tendrá en su base un pico detectable que será considerado como umbral. Su utilidad ha quedado demostrada en múltiples aplicaciones: detección de bordes, flujo óptico, diferenciación en la textura de imágenes, etc. y en este caso ha sido utilizado para determinar el criterio de parada en base al número de regiones optimas,  $t_s$ .

El procedimiento consiste en trazar una línea imaginaria del punto  $(1, C_{cum}(1))$  al  $(T_{cum}, C_{cum}(T_{cum}))$  tal y como se muestra en la segunda fila de la **Figura 4.10** para los tres escenarios distintos, y calcular la distancia perpendicular máxima a la curva  $C_{cum}$ . El parámetro  $T_{cum}$  puede ser utilizado para equilibrar la segmentación hacia un resultado sub o sobre-segmentado si se desea. Sin embargo, el factor dominante y más influyente en el resultado final será la forma de la función de coste acumulada  $C_{cum}(t)$ .

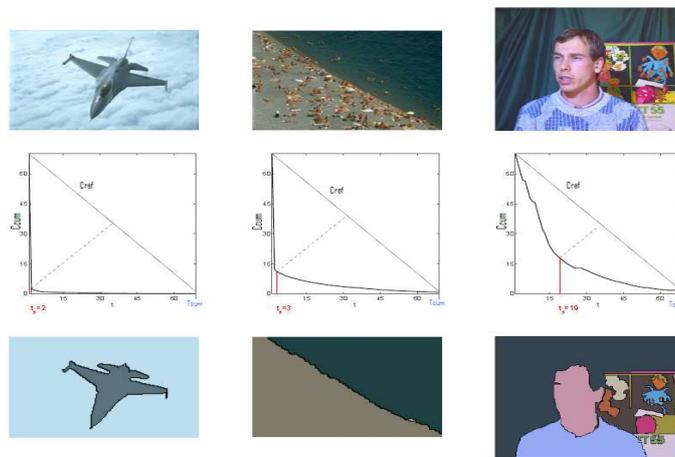
## 4.8 – Conclusiones

A lo largo de este capítulo se han presentado varias extensiones que el M-RSST añade al algoritmo RSST [49] y que permiten la segmentación de imágenes en grandes regiones. Éstas podrán reflejar unos objetos presentes en la escena, muy similares a los que podemos encontrar en la vida real.

Una de estas propuestas consiste en un nuevo entorno para la integración de datos procedentes de múltiples y diversas fuentes de información. Además permitirá asociar diferentes niveles de exactitud y fiabilidad a cada una de las fuentes. Otras extensiones incluyen un nuevo modelo para la medida de homogeneidad de color, soluciones prácticas para el análisis de las propiedades geométricas de las regiones y de su configuración espacial. Éstas últimas de gran interés por la complementariedad que suponen con el KMCC-Texturizado. Además se expone un nuevo criterio de parada sencillo. Todo ello destinado a la producción de regiones que contengan los objetos más destacados presentes en la escena.

Todas las extensiones anteriores pueden verse como un estudio de viabilidad de la utilización de la configuración espacial de las regiones y sus propiedades geométricas (las llamadas características sintácticas [12]) para mejorar la calidad de la segmentación producida por el RSST, aumentando así su utilidad para diversas aplicaciones. La efectividad demostrada por estas características [11], refuerzan la idea de poder aplicar las mismas a otros métodos de segmentación. Del mismo, otra rama de posible investigación sería añadir nuevas fuentes, diseñando y desarrollando sus respectivas BBAs.

Tal y como se ha demostrado experimentalmente en [11], las soluciones propuestas son de carácter genérico y logran una segmentación satisfactoria de imágenes del mundo real. Todo ello de manera automática, lo que será de gran interés para nuestro trabajo posterior.



**Figura 4.10** – Ejemplos de imágenes con sus respectivas funciones de coste acumuladas, parámetros  $T_s$  calculados para  $T_{cum}=70$  y resultados de la segmentación utilizando la representación extendida de color y las características geométricas descritas anteriormente.



## 5 - Análisis y evaluación del método híbrido





## 5 – Análisis y Evaluación del Método Híbrido

En este capítulo se presenta la innovación aportada en este proyecto fin de carrera: un algoritmo que vamos a denominar “Híbrido” de aquí en adelante, dado que se trata de la combinación de los dos métodos de segmentación (vistos en los capítulos 3 y 4).

### 5.1 – Introducción

Hasta ahora hemos visto dos métodos distintos de segmentación, de los cuales se han expuesto sus bases teóricas, incluyendo definiciones e información adicional para mejorar la comprensión de los mismos.

Un primer método denominado KMCC-Texturizado, que mediante el análisis del color, las texturas y la posición de los píxeles de la imagen, calcula las distancias euclídeas entre ellos para dividir la imagen en amplias regiones (cada una con su respectivo centro). En su etapa inicial se puede interpretar que parte, de una imagen con dos zonas diferenciadas con sus respectivos centros. Posteriormente, mediante sucesivas iteraciones va calculando el píxel más alejado de estos centros para obtener nuevos centros y nuevas regiones – ver **Figura 5.1a**.

El segundo método conocido como M-RSST, se basa en una distribución jerárquica que está representada por un BPTs (Binary Partition Trees). En éste, la división de la imagen comienza con tantas regiones como píxeles existan y estarán representadas por los nodos del BPT. Las uniones entre regiones (píxeles inicialmente) se irán formando también de manera iterativa, pero en este caso en función del coste asociado a las mismas. Tras una primera fase, se obtendrá una burda segmentación uniendo aquellos píxeles que minimicen los costes, únicamente en función de su homogeneidad de color – ver **Figura 5.1b**.

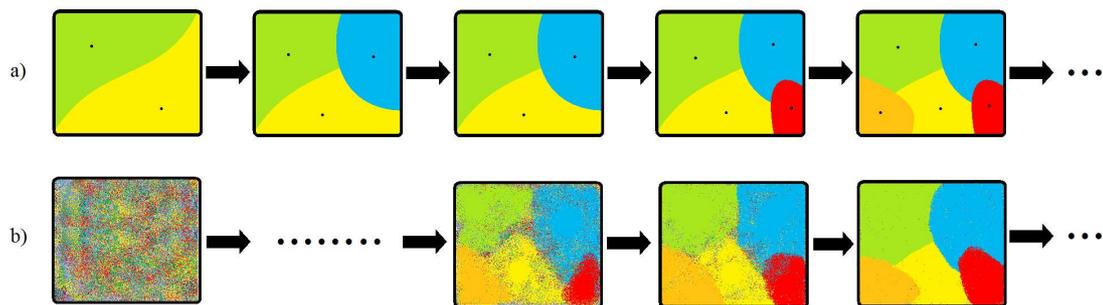
Se puede pensar que ambos algoritmos empiezan en diferentes extremos y avanzan en direcciones opuestas hacia el mismo destino. Por ello, desde este punto de vista, no parece una tarea trivial crear una combinación de ambos. Sin embargo es un hecho que, como se ha comprobado a lo largo su estudio de ambos, tanto la textura como las características sintácticas demuestran ser propiedades inertes a los objetos del mundo que nos rodea. Y esto las convierte en objetos de análisis necesarios para lograr un tratamiento de imágenes adecuado.

#### 5.1.1 – Objetivos

Estudiar la viabilidad de un método híbrido con estas características sintácticas y las de bajo nivel (del KMCC-T) es la motivación principal de este capítulo y uno de los objetivos principales, si no el principal, de este proyecto fin de carrera.

Con este fin se ha diseñado una versión del KMCC-T en el que hemos sustituido uno de sus módulos por el M-RSST adaptando adecuadamente las señales de ambos. Seguidamente, se han repetido para esta versión híbrida todos los experimentos sobre los que se evaluó el primero de los métodos, como se justificará más adelante.

En este capítulo se verá cada uno de los métodos desde un punto de vista más práctico con el objetivo de exponer de manera clara cómo se ha embebido el M-RSST en el KMCC-T. El análisis de estos resultados corroborará la idea inicial, de que la aportación dada por cada característica individualmente puede sumarse al aplicarlas en conjunto.



**Figura 5.1** – Ejemplo de las primeras fases de segmentación en una imagen sintética para: KMCC-Texturizado en a) y M-RSST en b)

### 5.1.2 – Herramientas utilizadas

- **KMCC-T:** El código del algoritmo, proporcionado por su autor V. Mezaris, está desarrollado en lenguaje de programación C. Para su explotación se ha utilizado la herramienta **Microsoft Visual C++ 6.0**.
- **M-RSST:** de este algoritmo no se dispondrá de su código abierto. El material proporcionado por la DCU (Dublin City University) consiste en una aplicación java en la que se encuentra embebido este método de segmentación junto con otros tantos y las librerías (.dll's) para cada uno de ellos. El código de la aplicación ha sido tratado y manejado con **Eclipse SDK 3.4.0** y el compilador **Apache Ant 1.7.1**, para añadir el KMCC-T y posteriormente el método híbrido.
- **Híbrido:** este algoritmo se ha desarrollado sobre la aplicación **JAVA** de la DCU: “Segmentation tool”. Aunque la base del mismo será el KMCC-T, desarrollado en lenguaje C, la librería que contiene el M-RSST será solamente accesible desde la herramienta de segmentación. Por este motivo, será necesario el uso de funciones nativas **JNI (Java Native Interface)** para lograr la ejecución conjunta de ambos.
- **Evaluación:** el código del evaluador esta desarrollado en lenguaje C++ por lo que de nuevo Microsoft Visual C++ 6.0 será la aplicación utilizada para su ejecución. Para

todos los cálculos posteriores de errores medios o análisis de los ficheros resultantes del evaluador se ha hecho uso de **Matlab 7.0**.

- **Imágenes:** La colección de imágenes sobre la que se aplicarán los casos experimentales será la misma que la utilizada para el método KMCC-T con el fin de realizar una comparación objetiva. La descripción de dicha colección puede encontrarse en el **apartado 3.1.1**. Para el manejo de las imágenes y los cambios de formato necesarios que se explicarán más adelante se ha requerido de la aplicación **IrfanView 4.0**.

### ***5.1.3 – Estructura del capítulo***

Tras esta introducción en la que se remarcan los objetivos perseguidos y se indican las herramientas utilizadas para su consecución, se pasa a exponer en la sección 5.2 las bases del método propuesto denominado “Híbrido”. En el punto 5.3 se explicarán en detalle las estructuras por separado de cada uno de los métodos para concluir con la integración de ambos en el método híbrido. Seguidamente, en 5.4 se mostrarán los resultados obtenidos de la aplicación del mismo. Para finalizar se resumirán las conclusiones obtenidas en el apartado 5.5.

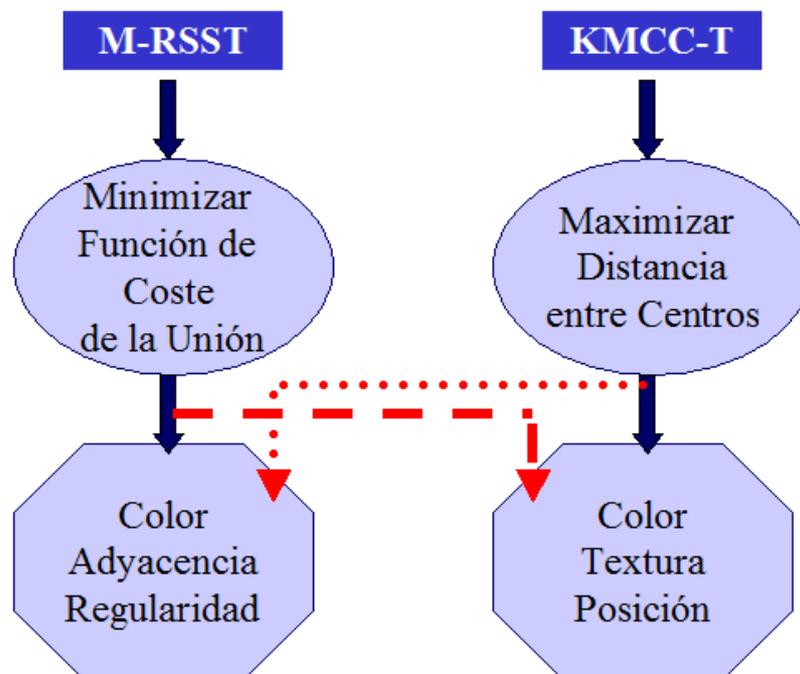
## 5.2 – Bases del Algoritmo Híbrido

La idea esencial del método es la fusión de las características de bajo nivel (textura, posición y color) por un lado, con las denominadas características sintácticas (adyacencia y regularidad de las formas geométricas, además del color).

Aunque desde el punto de vista de las características de los objetos, tenidas en cuenta por cada método estos parecen intuitivamente complementarios, debemos analizar el conjunto de los mismos. Hemos visto como cada uno utiliza un método para gestionar la creación de regiones:

- MRSST: minimizando la función de costes que dependería del color, de la adyacencia y de la regularidad.
- KMCC-T: maximizando la distancia de los centros que se calculará por la homogeneidad de color, textura y posición espacial.

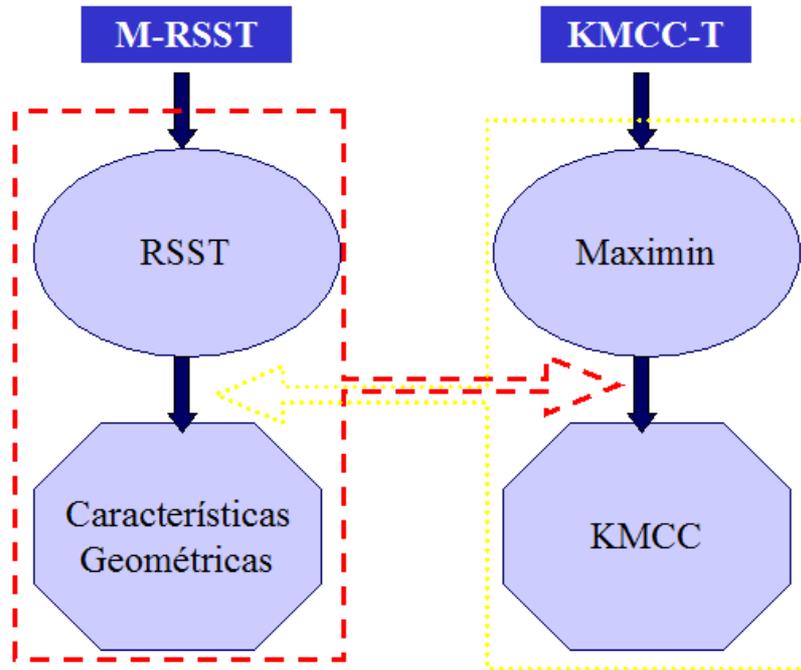
Una posibilidad tenida en cuenta fue la integración de las características de uno en la función de decisión del otro – ver **Figura 5.2**. Es decir, añadir las características sintácticas como medidas en el KMCC-T y del mismo modo en el cálculo de la distancia; o añadir el análisis de texturas al M-RSST e integrarla en la función de coste con su respectiva BBA.



**Figura 5.2** – Esquema con las dos posibilidades de fusión en función de las características medidas y las funciones de decisión desarrolladas en cada método.

Además de las funciones de decisión, hemos diferenciado dos etapas claras en ambos algoritmos en las que se trata la imagen de distinta manera. Una primera etapa de inicialización en la que se llevaba a cabo una rápida e inexacta segmentación. En el caso del M-RSST esto

estaba justificado puesto que el uso de las características geométricas requiere comenzar desde unas regiones con un cierto tamaño de las que se pueda extraer esa información que no contendrá un píxel. Esto se lograba mediante la aplicación del RSST original. Por otro lado, el KMCC-T está provisto de una fase previa a la segmentación KMCC, en la que se realiza un rápido agrupamiento con el único objetivo de inicializar el número de regiones y los centros de éstas de manera automática. En la **Figura 5.3** se muestra esta segunda posibilidad para fusionar ambos métodos.



**Figura 5.3** – Esquema de la fusión llevada a cabo para el método híbrido, marcada en rojo.

Debido a la dificultad, puesta de manifiesto en los capítulos anteriores, de desarrollar de una manera viable la medida de las mismas a lo que se añade la incertidumbre sobre su compatibilidad, se optará en este trabajo por la segunda opción propuesta (ver **Figura 5.3**). En concreto, se optará por adaptar el M-RSST como fase inicial de agrupamiento del KMCC-T, marcado en rojo en el esquema. La explicación y justificación es que debido al material del que se dispone de cada algoritmo solo será posible hacerlo en ese sentido, tal y como se verá más adelante.

En resumen, a continuación se esbozará una fusión sencilla de ambos métodos con la intención a posteriori de obtener datos numéricos que aboguen por una más concisa implementación futura o por el contrario desestimen la idea.

### 5.3 – Integración de los Métodos

Para explicar la integración de los algoritmos estudiados es preciso mostrar una visión global de los mismos. En primer lugar se analizan las señales y estructuras de ambos individualmente. Para continuar se irá mostrando paso a paso cómo se logra embeber uno dentro del otro. El procedimiento consistirá en integrar ambos métodos por separado en la herramienta de segmentación de la DCU para posteriormente y gracias a dicha aplicación poder ejecutar el método híbrido.

#### 5.3.1 – Estructura del KMCC-T

El diagrama de bloques del KMCC-T se puede observar en la **Figura 5.4**. El algoritmo parte de una imagen de dimensiones  $X_{max} \times Y_{max}$  y el primer paso es dividirla en bloques, obteniendo así una nueva representación de dicha imagen que tendrá dimensiones  $M \times N$ . Como vimos en el capítulo 3 (**apartado 3.5.5**) esta primera división es una utilidad para poder segmentar imágenes de grandes formatos. Ésta estará controlada por el factor de reducción que podrá variar en función de las necesidades y objetivos de la segmentación.

El siguiente paso consiste en convertir la imagen a un formato adecuado para su posterior tratamiento, este es, CIE LAB. En ese momento la imagen está acondicionada para comenzar con su segmentación en el bloque denominado “segm”.

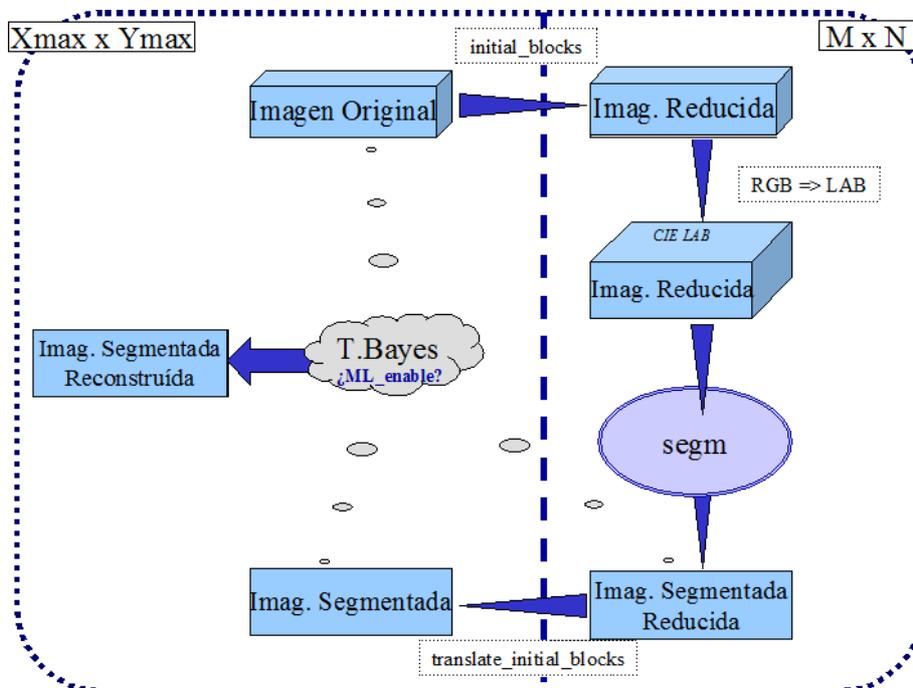
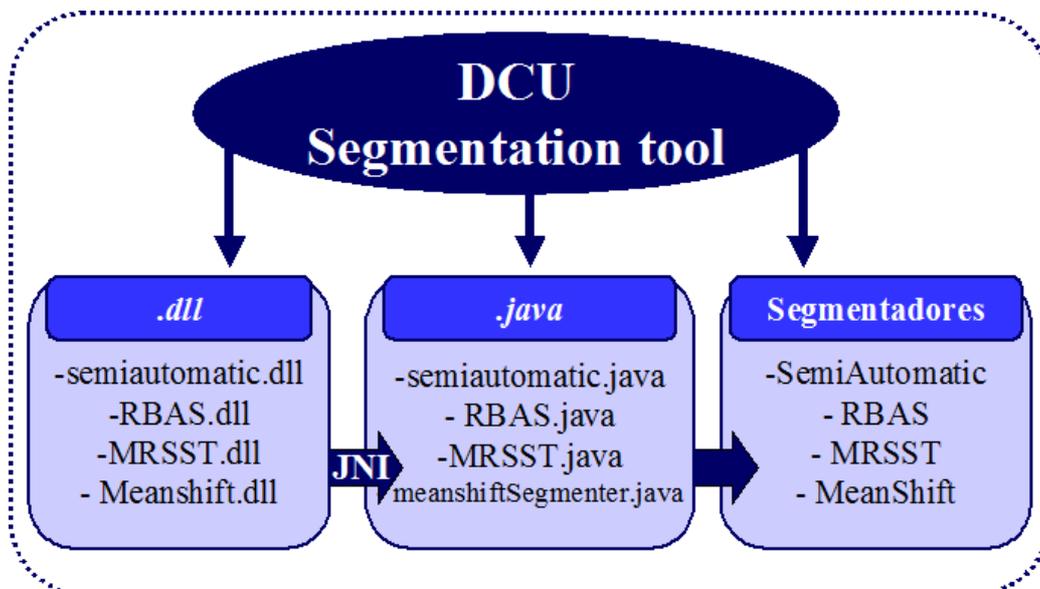


Figura 5.4 – Diagrama de bloques del algoritmo KMCC-T.

Las imágenes tratadas previamente a la segmentación propiamente dicha (bloque “segm”), serán señales multi-dimensionales. Ya sean en el espacio de color RGB o CIE LAB, e independientemente de sus dimensiones, las imágenes tendrán: tres dimensiones correspondientes a su intensidad de color (R,G y B o en su defecto L, a y b); más dos dimensiones espaciales (X e Y). De ahí la forma tridimensional del cubo para todas ellas, frente a los bloques de dos dimensiones, para las señales posteriores a la segmentación.

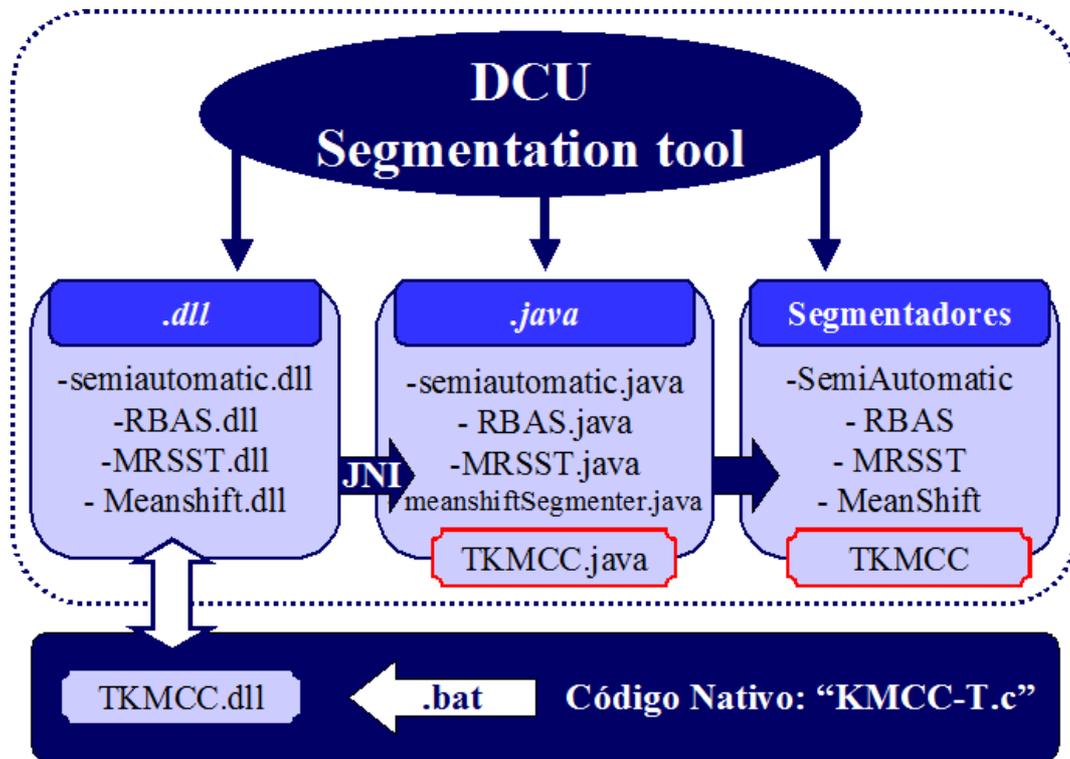
El resultado de “segm” será una mascarará de dimensiones  $M \times N$ , en la que los píxeles de cada región estarán marcados con el número asociado a la misma. En primer lugar se traducen los bloques para recuperar las dimensiones iniciales de la imagen. Para reconstruir por completo la imagen se podrá aplicar el Teorema de Bayes que reclasificará los píxeles situados en los bordes de las regiones. De este modo se refinan los contornos de los objetos segmentados y se mejoran considerablemente los resultados en los casos en los que se haya aplicado algún factor de reducción a la imagen inicial.



**Figura 5.5** – Esquema de la herramienta de segmentación con el M-RSST incluido.

### 5.3.2 – Estructura del M-RSST

En el caso del M-RSST no dispondremos en ningún caso de su código. Por ello no podremos conocer, acceder ni manipular ninguna de sus señales internas. Todo el material disponible relacionado con este método se encuentra encapsulado en su librería dinámica o DLL (Dynamic Linking Library). En la **Figura 5.5** se puede ver de manera esquematizada los elementos principales de los que consta la herramienta de segmentación (*segmentation tool*) proporcionada por la Universidad de Dublín, DCU. En esta herramienta encontramos integrado el M-RSST.



**Figura 5.6** – Esquema para la integración de un método nativo en la herramienta de

La aplicación, desarrollada en un entorno JAVA, dispone de una interfaz en la que el usuario puede explorar y abrir la imagen o video a segmentar, así como seleccionar el método deseado y configurar otros aspectos tales como: la carpeta contenedora de las imágenes resultado, tipo de visualización para las mascararas resultantes, establecer el valor de los parámetros de entrada del método (en caso necesario), etc.

Todas estas características son comunes a todos los métodos de segmentación integrados en la aplicación. En la herramienta existirá un fichero .java para cada algoritmo. Desde esa clase, se cargará la DLL propia del método. En ese momento y mediante el uso de las JNI (Java Native Interface) se puede ejecutar el algoritmo desde la interfaz de la herramienta, independientemente del lenguaje en el que haya sido desarrollado.

Por tanto, para añadir un nuevo algoritmo de segmentación en esta herramienta será necesario (en la **Figura 5.6** se muestra de manera esquemática el proceso):

- En primer lugar, crear la librería dinámica con el código fuente del algoritmo. En el caso mostrado en la **Figura 5.6** se muestra como se ha añadido el KMCC-Texturizado en la herramienta. Se ejecuta un fichero tipo .bat que a partir del código (.c) y sus respectivas cabeceras (.h) crea la librería denominada: **TKMCC.dll**. Esta librería se añadirá en la correspondiente carpeta de librerías de la herramienta.

- Seguidamente, se procede a crear la clase correspondiente al algoritmo: **TKMCC.java**. En el fichero, que será similar a cualquiera de los existentes se añadirán los parámetros característicos del método y algún otro campo más en concordancia con la aplicación: descripción breve del algoritmo, valores por defecto de los parámetros, etc. Desde esta clase se cargará el respectivo .dll, se obtendrán los parámetros definidos por el usuario a través de la interfaz y se tomará la imagen seleccionada para segmentar. Todo ello será ejecutado y tratado como si se tratara del programa original (lenguaje C).
- Por último, se añade la clase a la lista de **segmentadores** mostrados en la interfaz. Aunque no se ha hecho hasta ahora, se debe especificar que no todos los algoritmos integrados en la herramienta aparecerán por obligación en la interfaz. Entre las opciones de la misma, existe una lista editable con las clases que se desean utilizar.

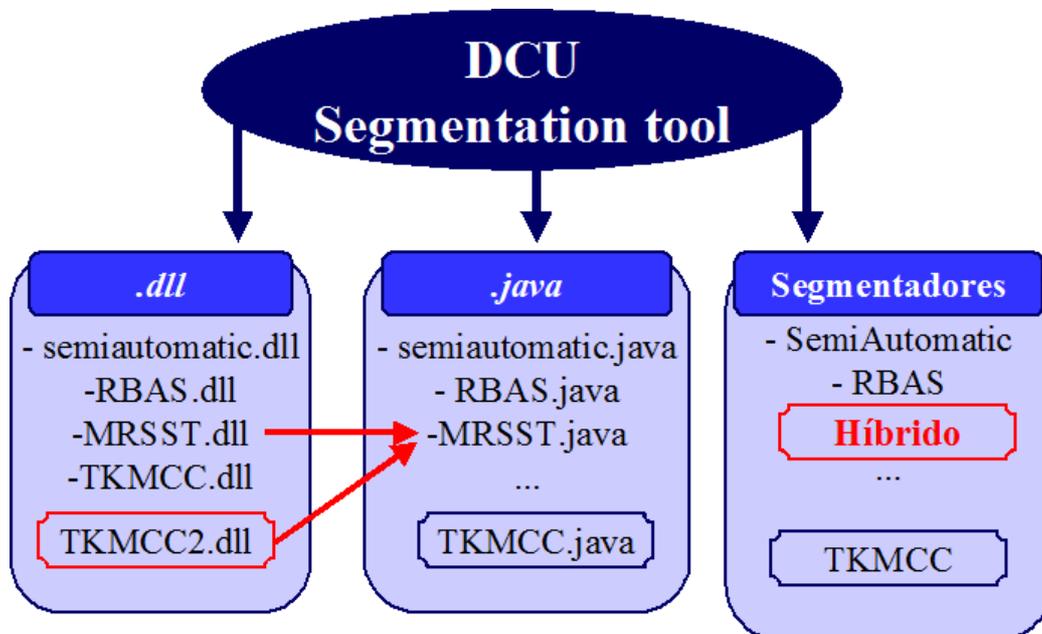


Figura 5.7 – Esquema de la integración del método Híbrido en la herramienta de segmentación.

### 5.3.3 – Estructura del Método Híbrido

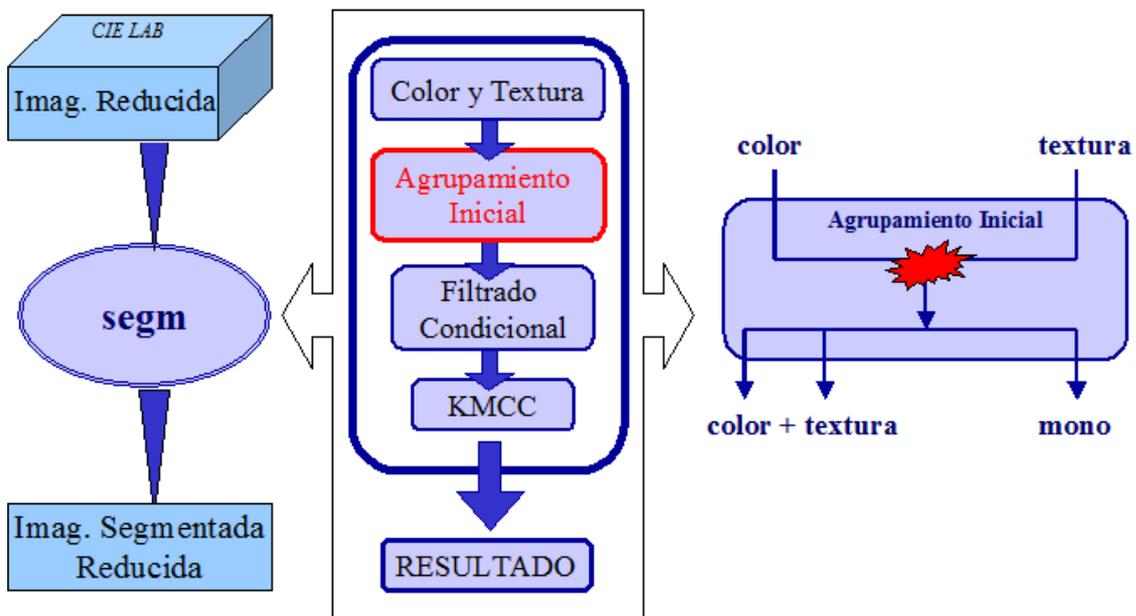
Una vez vistas las estructuras generales del funcionamiento conocido de ambos algoritmos y la manera de integrar un nuevo algoritmo en la herramienta de segmentación, se puede explicar fácilmente la integración del método híbrido. Como veremos, será ligeramente distinto al caso visto anteriormente con el T-KMCC.

Como hemos comentado en el apartado anterior, cada librería está creada a partir de un determinado algoritmo. Sin embargo, previo a la creación de las DLL's estos códigos han debido ser adaptados a la aplicación Java. Esto es, cada librería dinámica está desarrollada para

un método en concreto pero también para ser cargada y ejecutada desde una clase Java en concreto. Por este motivo, la librería “MRSST.dll” está estrechamente vinculada a la clase “MRSST.java” y no podrá ser utilizada desde ninguna otra.

Por tanto, como se observa en la **Figura 5.7**, el único modo viable para poder utilizar ambas librerías en un mismo segmentador será utilizar y modificar “MRSST.java”. Será necesario crear una nueva librería “TKMCC2.dll” adaptada a esta clase. De este modo, podremos ejecutar ambos métodos en la misma clase y añadir el segmentador “Híbrido” en la herramienta.

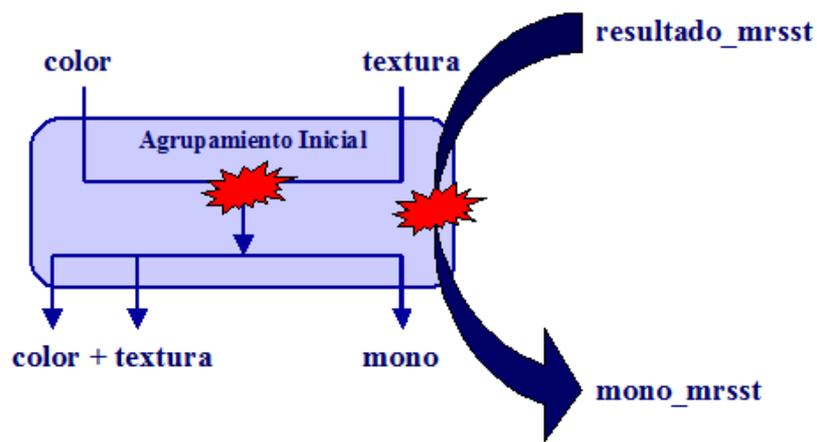
Después de esto, comienza una segunda fase en la integración: adaptar las señales internas para poder sustituir la etapa de “Agrupamiento Inicial” del KMCC-T por los resultados obtenidos en M-RSST. En la **Figura 5.8** podemos observar en detalle algunas de las señales involucradas en la etapa de agrupamiento, además de un recordatorio en forma de esquema de la localización de la misma.



**Figura 5.8** – Diagrama de bloque con las señales internas del KMCC-T

En la etapa de agrupamiento inicial se toman como argumentos de entrada la señal de intensidad de color (versión reducida, según el valor del factor de reducción) y por otro lado la señal con la información de textura, obtenida en la etapa anterior. Para no entrar en detalles innecesarios sobre la secuencia de ejecuciones, se han reseñado las señales más importantes. Por un lado la textura y el color, que serán las herramientas principales a lo largo del algoritmo. Y por otro lado, la máscara segmentada inicializada mediante el método “maximin” descrito en el **apartado 3.5.2**. De esta máscara parte el KMCC (previo filtrado condicional) para llevar a cabo todos sus cálculos y modificar centros y regiones en función de las distancias calculadas.

Sustituyendo la señal mono calculada en el bloque original por el resultado del M-RSST el algoritmo continuaría indiferente a lo acontecido pero partiendo de una máscara mucho más valiosa que la anterior y arrastrando consigo el análisis de las características sintácticas estudiadas. Esto se puede muestra en la Figura 5.9 de manera esquemática, sin que quede de manifiesto la incompatibilidad de las dimensiones de las máscaras. El resultado del M-RSST será de las mismas dimensiones que la imagen original:  $X_{max} \times Y_{max}$ . Por ello habrá que aplicarle antes una función homóloga a “initial\_blocks” (ver **Figura 5.4**) para obtener una versión reducida de la máscara. En este caso no se están calculando colores medios puesto que como ya hemos comentado en las máscaras cada píxel contendrá el valor asociado a su región. Por ello, en este caso, en la versión reducida, cada bloque contendrá el valor de la región que más presente estuviera en el conjunto de píxeles que representa, es decir, la región predominante.



**Figura 5.9** – Etapa del KMCC-T adaptada a la señal proveniente del M-RSST.

Como ya hemos dicho, de aquí en adelante el KMCC-T proseguiría con su ejecución habitual a pesar de que se estén ejecutando ambos métodos conjuntamente desde la herramienta de segmentación. En el siguiente capítulo se mostrarán y analizarán los resultados obtenidos del método híbrido.

## 5.4 – Resultados de la Evaluación

En esta sección se muestran los resultados obtenidos tras la aplicación y posterior evaluación del método Híbrido descrito en esta sección. Los parámetros a combina serán los mismos que para los métodos fusionados, que como recordaremos serán para el caso del KMCC-T:

- **W:** indica el factor de reducción de la imagen, es decir, tamaño de los bloques en los que se divide la imagen inicialmente. Variable de 1 a 8. Es el valor  $R$  de la sección 3.5.5.
- **LDC:** establece el número de niveles para la descomposición de la transformada DWF (2, 3 o 4). Es  $N$  en el punto 3.5.1.
- **ML\_enable:** habilita/deshabilita el uso del Teorema de Bayes para la reconstrucción de la imagen, tras aplicar el método, al traducir los bloques creados. Supuestamente deshabilitado para  $W=1$  y habilitado en el resto de los casos.
- **MaxAreas:** fija el número máximo de áreas que se podrán obtener. Podrá venir fijado por la aplicación multimedia.

En el caso del M-RSST no tendremos ningún tipo de control puesto que como hemos visto, se podría interpretar como una especie de caja negra integrada en la herramienta de segmentación. Por ello se hará un pequeño experimento aunque sus resultados serán poco indicativos. Además se debe tener en cuenta que la integración, debido a las circunstancias impuestas, ha sido efectuada en consonancia con el KMCC-T.

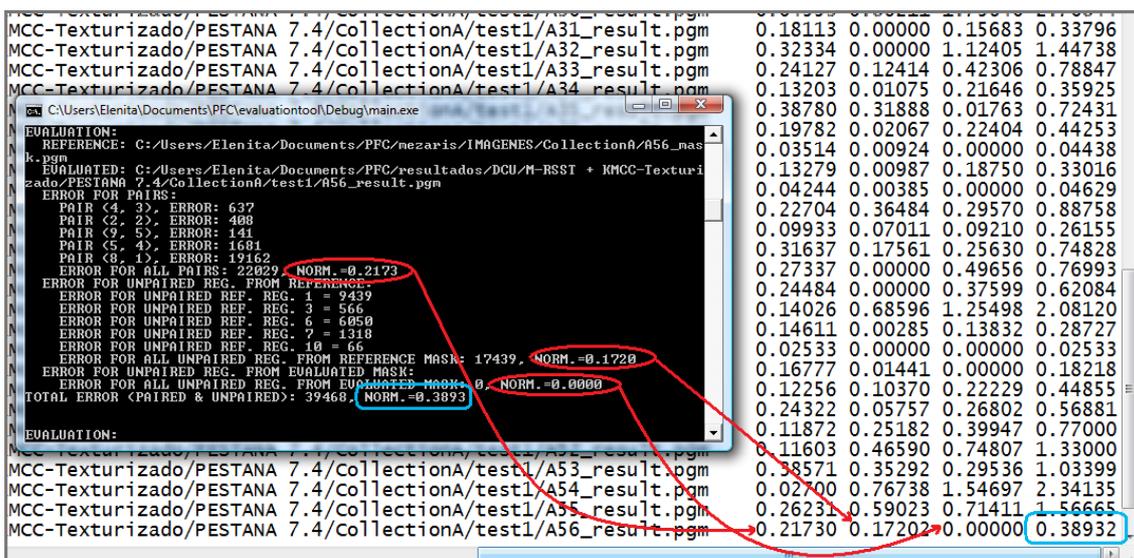


Figura 5.10 – Ejemplo de ejecución del evaluador junto con el fichero resultante.

### 5.4.1 – Método de Evaluación

El método de evaluación será el mismo que utilizado para el KMCC-T puesto que se trata lograr unos resultados objetivos y equitativos. Por ello, para el estudio del método híbrido utilizaremos la herramienta de evaluación propuesta en [97], y descrita en el **apartado 3.6.1**.

En la **Figura 5.10** se muestra un ejemplo del fichero obtenido tras la evaluación para cada escenario. Como se puede observar en el fichero se leen tantas líneas como imágenes en las que se incluye el path de las máscaras que se comparan, seguido de cuatro columnas: la primera será el error asociado a las parejas de regiones formadas; la segunda y tercera será el error asociado a las regiones no asociadas de la máscara de referencia y de la segmentada respectivamente; en última instancia la suma de las tres anteriores. Este fichero es computado por un script en Matlab que calcula el error medio para cada fichero de texto, es decir, para cada escenario.

Un añadido respecto a la evaluación realizada anteriormente (**apartado 3.6**) es la manipulación de las imágenes previa a su evaluación. Los resultados del KMCC-T serán formato ‘.pgm’ que es el aceptado por el evaluador. Sin embargo, la herramienta de segmentación de la DCU proporcionará resultados en formato ‘.png’. Por este motivo será necesaria la aplicación Irfanview que corregirá los formatos.

### 5.4.2 – Método Híbrido

Los escenarios sobre los que se va a aplicar el método híbrido se resumen en la **Tabla 5.1**. Como ya hemos dejado entrever serán los mismos que se probaron para el KMCC-T. En este caso el objetivo será comparar los resultados obtenidos en ambos casos. Para ello, en la **Tabla 5.2** se enfrentan los resultados de ambos algoritmos. Se añade una cuarta columna en la que se refleja la mejora que ha supuesto la modificación realizada en la etapa de inicial de agrupamiento automático.

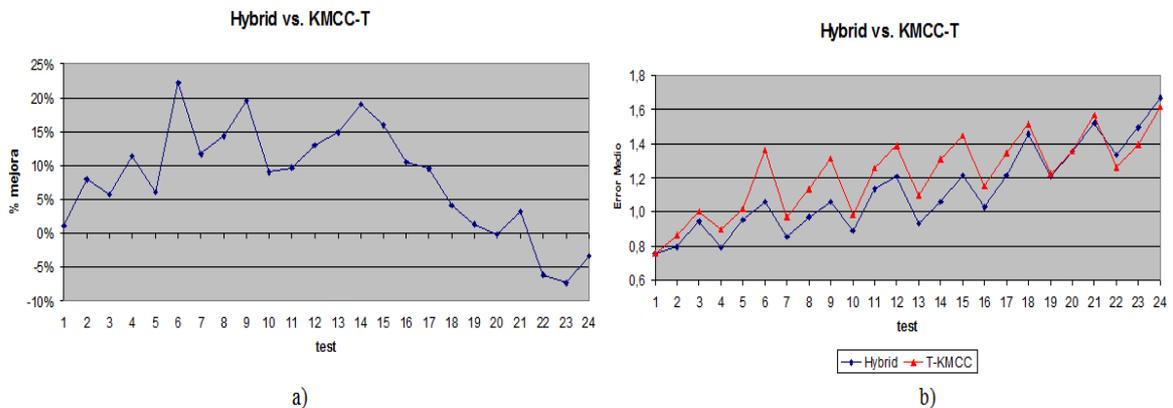
En la **Figura 5.11a** se pueden observar gráficamente la evolución de dicha mejora así como los errores de ambos algoritmos por separado (**Figura 5.11b**). Como era de esperar, también en el método híbrido se observa un comportamiento cíclico respecto al parámetro LDC. En gran parte de los casos se puede observar a simple vista como el error se encuentra bastante por debajo del que obteníamos para el KMCC-T.

El caso más favorable del método evaluado se da de nuevo para el test 1. Sin embargo como ya se justificó en el capítulo 3, se buscará el equilibrio entre un buen resultado en la evaluación y un buen rendimiento computacional. Siguiendo el mismo criterio, según el cual los valores inferiores a uno serían aceptables, encontraremos muchos más escenarios interesantes en este caso.

Test	W	LDC	ML_enable	Max Areas	Col A	Col B	Media
<b>1</b>	<b>1</b>	<b>2</b>	<b>0</b>	<b>20</b>	<b>0,691</b>	<b>0,811</b>	<b>0,751</b>
2	1	3	0	20	0,674	0,914	0,794
3	1	4	0	20	0,874	1,016	0,945
<b>4</b>	<b>2</b>	<b>2</b>	<b>1</b>	<b>20</b>	<b>0,726</b>	<b>0,857</b>	<b>0,791</b>
<b>5</b>	<b>2</b>	<b>3</b>	<b>1</b>	<b>20</b>	<b>0,868</b>	<b>1,038</b>	<b>0,953</b>
6	2	4	1	20	1,028	1,088	1,058
<b>7</b>	<b>3</b>	<b>2</b>	<b>1</b>	<b>20</b>	<b>0,795</b>	<b>0,917</b>	<b>0,856</b>
<b>8</b>	<b>3</b>	<b>3</b>	<b>1</b>	<b>20</b>	<b>0,861</b>	<b>1,082</b>	<b>0,972</b>
9	3	4	1	20	0,966	1,151	1,058
<b>10</b>	<b>4</b>	<b>2</b>	<b>1</b>	<b>20</b>	<b>0,833</b>	<b>0,953</b>	<b>0,893</b>
11	4	3	1	20	1,106	1,163	1,135
12	4	4	1	20	1,222	1,198	1,210
<b>13</b>	<b>5</b>	<b>2</b>	<b>1</b>	<b>20</b>	<b>0,848</b>	<b>1,021</b>	<b>0,935</b>
14	5	3	1	20	0,955	1,162	1,059
15	5	4	1	20	1,193	1,239	1,216
16	6	2	1	20	0,988	1,072	1,030
17	6	3	1	20	1,140	1,289	1,215
18	6	4	1	20	1,411	1,498	1,454
19	7	2	1	20	1,190	1,227	1,208
20	7	3	1	20	1,325	1,390	1,358
21	7	4	1	20	1,557	1,477	1,517
22	8	2	1	20	1,274	1,397	1,336
23	8	3	1	20	1,552	1,436	1,494
24	8	4	1	20	1,691	1,648	1,669

**Tabla 5.1** – Escenarios y resultados obtenidos para cada colección por el método híbrido

Estos casos serán los test 4, 5, 7, 8, 10 y 13 (ver **Tabla 5.1**). Destacaremos especialmente el test número 13 en el que con un factor de reducción alto como es 5, se obtiene un muy buen resultado. Esto significa una mejora de casi un 15% respecto al valor de error obtenido para el KMCC-T.



**Figura 5.11** – a) Mejora aportada por el método híbrido para cada escenario. b) Comparación de errores medios calculados para el KMCC-T y método híbrido.

Test	Media		% Mejora
	T-KMCC	Híbrido	
1	0,759	0,751	1,04%
2	0,862	0,794	7,94%
3	1,003	0,945	5,77%
4	0,893	0,791	11,47%
5	1,016	0,953	6,16%
<b>6</b>	<b>1,360</b>	<b>1,058</b>	<b>22,21%</b>
7	0,969	0,856	11,67%
8	1,134	0,972	14,33%
<b>9</b>	<b>1,316</b>	<b>1,058</b>	<b>19,58%</b>
10	0,981	0,893	8,98%
11	1,256	1,135	9,66%
12	1,390	1,210	12,99%
13	1,098	0,935	14,87%
<b>14</b>	<b>1,306</b>	<b>1,059</b>	<b>18,97%</b>
<b>15</b>	<b>1,446</b>	<b>1,216</b>	<b>15,93%</b>
16	1,151	1,030	10,53%
17	1,344	1,215	9,62%
18	1,517	1,454	4,11%
19	1,224	1,208	1,31%
20	1,355	1,358	-0,21%
21	1,568	1,517	3,22%
22	1,259	1,336	-6,11%
23	1,392	1,494	-7,33%
24	1,616	1,669	-3,29%

**Tabla 5.2** – Tabla con los resultados finales del KMCC-T y el método híbrido.

Respecto a las mejoras en % obtenidas para cada escenario (**Tabla 5.2**) comentaremos que los mejores resultados se han obtenido en los casos: 6, 9, 14 y 15. Con picos de mejoras superiores al 20% (test 6). Es interesante, el hecho de que los mejores porcentajes se hayan obtenido en escenarios de LDC = 3 o 4, a pesar de que los mejores resultados siguen dándose en LDC=2. Podría plantearse la posible relación del parámetro LDC con el algoritmo M-RSST y sobre todo con el modo en el que se ha integrado en el KMCC-T. Pero para llegar a cualquier conclusión aceptable habría que hacer un estudio más amplio ese parámetro aplicado directamente al M-RSST (lo que no es posible en este caso).

A continuación, y continuando con esta evaluación paralela a la realizada para el KMCC-T se han variado de 5 a 20 el número máximo de regiones detectables. Los resultados se muestran en la **Tabla 5.3** y su comparación con el la versión original del algoritmo KMCC-T se refleja en la **Figura 5.12**. De nuevo, el caso óptimo se da para “Max Área = 5” con un 4% de mejora. Las mejoras de los errores medios varían entre un 8 y un 6% para el resto de casos.

W	Test	Max Area = 20	Max Area = 15	Max Area = 10	Max Area = 5
		Híbrido			
1	1	0,75	0,76	0,74	0,76
	2	0,79	0,80	0,82	0,85
	3	0,94	0,93	0,95	1,03
2	4	0,79	0,79	0,76	0,80
	5	0,95	0,95	0,98	0,96
	6	1,06	1,06	1,07	1,09
3	7	0,86	0,85	0,84	0,80
	8	0,97	0,97	1,00	1,00
	9	1,06	1,06	1,05	1,07
4	10	0,89	0,89	0,89	0,87
	11	1,13	1,13	1,13	1,13
	12	1,21	1,21	1,20	1,18
5	13	0,93	0,91	0,91	0,89
	14	1,06	1,06	1,07	1,08
	15	1,22	1,20	1,19	1,11
6	16	1,03	1,03	1,01	1,02
	17	1,21	1,22	1,20	1,16
	18	1,45	1,45	1,44	1,34
7	19	1,21	1,20	1,16	1,15
	20	1,36	1,35	1,32	1,29
	21	1,52	1,50	1,50	1,43
8	22	1,34	1,32	1,29	1,29
	23	1,49	1,47	1,39	1,31
	24	1,67	1,64	1,58	1,54
MEDIA		1,12	1,11	1,10	1,09

Tabla 5.3 – Variación del error medio en función del parámetro del número máximo de áreas.

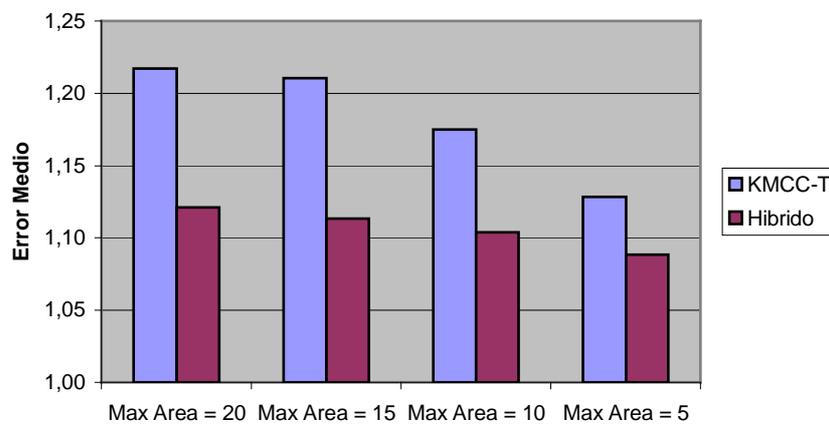
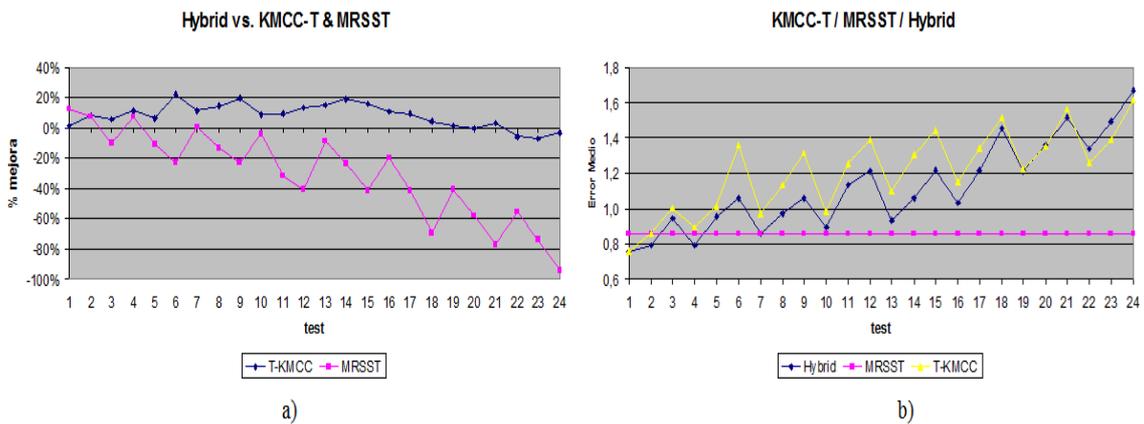


Figura 5.12 – Errores medios en función del número de áreas del KMCC-T y del método híbrido.

Por último, se realizó un experimento corto segmentando con el M-RSST integrado en la herramienta las dos colecciones de imágenes. En la **Figura 5.13** se comparan los resultados con los obtenidos anteriormente. Dado que no existe la posibilidad de manipular el algoritmo ni ejecutarlo bajo distintos escenarios el resultado será un único error medio: 0,8582. Este resultado será inferior para la gran mayoría de casos en los que se aplique el factor de reducción.

De nuevo para proponer alguna conclusión con cierta objetividad se deberían realizar experimentaciones adaptando el M-RSST. Sin embargo podremos aventurarnos a afirmar que se observa una incompatibilidad entre las características geométricas y el tratamiento de las imágenes de grandes formatos del KMCC-T.

Una manera de comprobarlo sería aplicar individualmente el análisis de texturas realizado al M-RSST, diseñando su respectiva BBA. Esto sería una parte del trabajo futuro propuesto tras estos últimos resultados.



**Figura 5.13** – Comparativa del error medio obtenido para el M-RSST con los dos anteriores.

## 5.5 – Conclusiones

En este capítulo hemos visto la manera de lograr una fusión sencilla de los dos métodos de segmentación estudiados en este proyecto con el objetivo de determinar la viabilidad de desarrollo de un método con características de distinta naturaleza.

Se aboga por la integración de múltiples fuentes en un mismo método independientemente del método utilizado posteriormente para gestionar las regiones creadas a partir de la imagen original. Los resultados experimentales han resultado satisfactorios en el sentido que se esperaba, teniendo en cuenta que la adaptación se realizó para el algoritmo KMCC-T. A pesar de esto, también se han obtenido resultados positivos al comparar los resultados con el M-RSST para unos casos específicos.

En conclusión, diremos que las características sintácticas y visuales (o de bajo nivel) no serán excluyentes en ningún caso si no que, siempre que exista la posibilidad, se deberán combinar en beneficio del proceso de segmentación. En el algoritmo descrito en este capítulo, se han focalizado los esfuerzos en un acercamiento hacia la integración de dichas características manteniendo la base de una segmentación automática, robusta y eficaz.

## 6 - Conclusiones y trabajo futuro





## 6 – Conclusiones y Trabajo Futuro

En este último capítulo se recogen las conclusiones extraídas a lo largo de este proyecto fin de carrera. Además se indican algunas de las posibles líneas de trabajo futuro.

### 6.1 – Discusión

A lo largo de este trabajo se ha realizado un primer estudio abarcando un amplio campo en el tratamiento de imágenes: la segmentación de imágenes. Una vez repasado el estado del arte y conocidas doctrinas necesarias para introducirnos en ese terreno se seleccionan dos algoritmos de especial interés por su aparente afinidad.

Tras el estudio teórico de ambos métodos, plasmado en esta memoria, y con la idea en mente de cómo se produciría la fusión de ambos se llevo a cabo la evaluación de uno de ellos. El motivo de evaluar solo el KMCC-T, en un principio, se debe a que dada la manera en que se ha efectuado la fusión, era interesante la comparación con él (aunque finalmente se realiza también una pequeña comparación con el M-RSST). Posteriormente se procedería a embeber en él el método M-RSST, obteniendo el que hemos denominado: “Híbrido”. Sobre éste último se ha llevado a cabo la misma evaluación, con el fin de comparar el comportamiento ante la variación de sus parámetros de entrada y evaluar la mejora de manera objetiva.

Con mejoras de hasta un 20% en los resultados, podemos apostar por la creación de un nuevo algoritmo basado en las características visuales y sintácticas que por separado se han integrado en el KMCC-T y el M-RSST, respectivamente.

Los resultados no serán tan satisfactorios si los comparamos con el M-RSST pero esto era de esperar en tanto y cuanto la adaptación, tanto teórica como práctica, está pensada y diseñada a favor del KMCC-T.

En definitiva, las características sintácticas demuestran ser una importante fuente de información que servirá de enlace entre las características de bajo nivel y la interpretación semántica de la escena. Aunque éstas no aportarán por si solas una evidencia lo suficientemente fuerte como para formar objetos semánticos complejos, a partir de grupos o regiones muy dispares en cuanto a color o textura, si se puede demostrar que mejoran significativamente el comportamiento en una segmentación automática.

Una última conclusión obtenida de esta tarea de investigación y común a la gran mayoría de trabajos sobre segmentación de imágenes es que la universalidad de la segmentación radicará en la universalidad de las aplicaciones multimedia. Así como cada aplicación se diseña con un cometido y una finalidad, no podemos pretender desarrollar una segmentación que sirva

para todas ellas. Las imágenes objeto de segmentación, la funcionalidad de la aplicación y la subjetividad del usuario serán factores clave a la hora de determinar la calidad de un método.

## 6.2 – Trabajo Futuro

Son muchos los caminos que han derivado hasta este trabajo, y muchos son también los que se derivan del mismo. Tanto los aportes individuales que se puedan realizar para cada uno de los algoritmos por separado como las modificaciones aplicables al método híbrido, múltiples y variadas, se numeran en esta sección.

Las posibilidades de trabajo futuro que incluyen la mejora individual de cada uno de los métodos podrán ser:

- **M-RSST**: se podrán investigar y añadir nuevas características geométricas así como estudiar la dependencia entre las ya definidas.
- **KMCC-T**: el campo del análisis de las texturas está lejos de la perfección. Este algoritmo presenta una gran flexibilidad ante cualquier mejora de este tipo. Podrá ser fácilmente evaluable gracias a que la textura se define como una señal independiente en el propio algoritmo.

Tras haber demostrado la viabilidad de un algoritmo que combine características sintácticas y visuales se pueden abrir nuevas líneas de trabajo basándonos en esos resultados. En particular para el método híbrido, y para este proyecto fin de carrera, las vías de investigación posibles podrían incluir:

- Ampliar el banco de imágenes con máscaras de referencia para poder ampliar a una minuciosa labor experimental. Observar el comportamiento ante diferentes tipos de imágenes con distintos tamaños pero sobre todo evaluar un gran número de imágenes es imprescindible en cualquier investigación que se precie.
- Introducir las características geométricas directamente en el núcleo del KMCC-T (bloque KMCC). Esto supondrá una adaptación de la función distancia en la que habría que ponderar estas nuevas medidas. Por otro lado, esto implicaría volver al agrupamiento inicial mediante el algoritmo “maximin”, cuya mejora sería otro reto a tener en cuenta.

- Incluir el análisis de texturas en el M-RSST con su respectiva estructura BBA. En ese caso se podría estudiar en detalle el comportamiento del parámetro LDC y su complementariedad o no, con las características geométricas.
- Sustituir la primera fase del M-RSST por el KMCC-T. Esta opción es intuitivamente muy acertada y fácil de implementar una vez se disponga de ambos códigos o de una versión mas moldeable del MRSST. El análisis de formas y contornos partiría así de una imagen sobre-segmentada y basada en información extra gracias a la textura, añadida a la evidencia dada por la intensidad de color.
- Modificar el método utilizado para el análisis de texturas es una línea de trabajo futuro propuesta para el propio KMCC-T. Pero si se enfoca esta mejora al método híbrido, se desarrollará un método que ya no tenga que analizar las texturas de una imagen pixelada (ardua tarea en el análisis convencional de texturas). Se partirá de regiones con un cierto significado semántico (aportado por el M-RSST) en la etapa de agrupamiento inicial.
- Evaluación del evaluador. A lo largo de esta investigación hemos encontrado trabajos con resultados muy dispares en cuanto a la evaluación de las máscaras segmentadas. Esto ha hecho que solo se hayan tenido en cuenta los resultados obtenidos por el evaluador definido con las imágenes de nuestra colección al comparar los métodos. Evaluar los métodos individualmente, así como el híbrido con distintos evaluadores podría ser una manera de medir objetivamente la objetividad del evaluador utilizado en este trabajo o en cualquier otro.



## Bibliografía





## BIBLIOGRAFÍA

---

- [1] S. Brin and L. Page, "The Anatomy Of A Large-Scale Hypertextual Web Search Engine," In Proc. The 7th Int'l Www Conf. (Www'98). Brisbane, Australia, 1998.
- [2] J. P. Eakins and M. E. Graham, "Content-Based Image Retrieval: A Report To The Jisc Technology Application Programme," Institute For Image Data Research, University Of Northumbria At Newcastle, Uk, [www.unn.ac.uk/lidr/Report.html](http://www.unn.ac.uk/lidr/Report.html), Technical Report, 1999.
- [3] B. S. Manjunath, P. Salembier, and T. Sikora, Introduction To Mpeg-7: Multimedia Content Description Language. New York: John Wiley & Sons Ltd., Isbn: 0-471-48678-7, 2002.
- [4] J. M. Martínez, "Mpeg-7 Overview," Iso/Iec Jtc1/Sc29/Wg11, Tech. Rep. N6828, Mar. 2003.
- [5] D. Santa-Cruz and T. Ebrahimi, "An Analytical Study Of Jpeg 2000 Functionalities," In Proc. Ieee Int'l Conf. On Image Processing, Vol. 2, 2000, Pp. 49–52.
- [6] C. Carson, S. Belongie, H. Greenspan, and J. Malik, "Blobworld: Color- and Texture-Based Image Segmentation Using Em and Its Application To Image Querying and Classification," Ieee Trans. Pattern Anal. and Machine Intell., Vol. 24, No. 8, Pp. 1026–1037, Aug. 2002.
- [7] V. Mezaris, I.Kompatsiaris, and M.G.Strintzis, "Still Image Segmentation Tools For Object-Based Multimedia Applications," Int'l Journal Of Pattern Recognition and Artificial Intell., Vol. 18, No. 4, Pp. 701–725, June 2004.
- [8] J. Fauqueur and N. Boujemaa, "Region-Based Image Retrieval: Fast Coarse Segmentation and Fine Color Description," Journal Of Visual Languages and Computing, Special Issue On Visual Information Systems, Vol. 15, Pp. 69–95, 2004.
- [9] W. Skarbek and A. Koschan, "Colour Image Segmentation - A Survey," Technical University Of Berlin, Department Of Computer Science, Germany, Technical Report 94-32, Oct. 1994.
- [10] P. Salembier and F. Marqués, "Region-Based Representations Of Image and Video," Ieee Trans. Circuits Syst. Video Technol., Vol. 9, No. 8, Pp. 1149–1167, Dec. 1999.
- [11] T. Adamek, Tesis Doctoral "Using contour information and segmentation for object registration, modeling and retrieval", School of Electronic Engineering, Dublin city university, June 2006.
- [12] C. F. Bennstrom and J. R. Casas, "Binary-Partition-Tree Creation Using A Quasiinclusion Criterion," In Proc. 8th Int'l Conf. On Information Visualization (Iv'04), Ieee Computer Society Press, London, Uk, 2004.
- [13] O. Salerno, M. Pardas, V. Vilaplana, and F. Marqués, "Object Recognition Based On Binary Partition Trees," In Proc. Int'l Conf. On Image Processing (Icip '04), Vol. 2, Oct. 2004, Pp. 929–932.
- [14] H.González, "Extracción Y Gestión De Regiones De Interés En Contenido Audiovisual", Escuela Politécnica Superior De La Universidad Autónoma De Madrid, Julio 2008.
- [15] G. Aggarwal, S. Ghosal, and P. Dubey, "Efficient Query Modification For Image Retrieval," In Proc. Ieee Conf. Computer Vision and Pattern Recognition (Cvpr'00), Vol. Ii, June 2000, Pp. 255–261.

- [16] V. Mezaris, I. Kompatsiaris, and M. G. Strintzis, "Region-Based Image Retrieval Using An Object Ontology and Relevance Feedback," *Eurasip Journal On Applied Signal Processing*, No. 6, Pp. 886–901, June 2004.
- [17] F. Souvannavong, B. Merialdo, and B. Huet, "Region-Based Video Content Indexing and Retrieval," In Proc. 4th Int'l Workshop On Content-Based Multimedia Indexing (Cbmi'05), 2005.
- [18] P. Duygulu, K. Barnard, N. De Freitas, and D. Forsyth, "Object Recognition As Machine Translation: Learning A Lexicon For A Fixed Image Vocabulary," In Proc. Ieee Int'l Conf. On Computer Vision, 2002, Pp. 97–112.
- [19] F. A. Cheikh, "Muvis: A System For Content-Based Image Retrieval," Ph.D. Dissertation, Tampere University Of Technology, 2004.
- [20] R. C. Veltkamp and M. Tanase, "Content-Based Image Retrieval Systems: A Survey," Department Of Computing Science, Utrecht University, Technical Report Uu-Cs-2000-34, Tech. Rep., Oct. 2000.
- [21] M. Flickner, H. Sawhney, W. Niblack, J. Ashley, Q. Huang, B. Dom, M. Gorkhani, J. Hafner, D. Lee, D. Petkovic, D. Steele, and P. Yanker, "Query By Image and Video Content: The Qbic System," *Ieee Computer Magazine*, Vol. 28, No. 9, Pp. 23–32, Sept. 1995.
- [22] Ibm, "Qbic Project," [Http://Www.qbic.Almaden.Ibm.Com](http://www.qbic.almaden.ibm.com).
- [23] M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," *Int'l Journal Of Computer Vision*, Vol. 1, Pp. 312–331, 1988.
- [24] P. Pala and S. Santini, "Image Retrieval By Shape and Texture," *Pattern Recognition*, Vol. 32, No. 3, Pp. 517–527, 1999.
- [25] Y. Luo and R. Duraiswami, "Canny Edge Detection On Nvidia Cuda", In *Computer Science & Umiacs*, University Of Maryland, College Park, 2008.
- [26] W.-Y. Ma and B. S. Manjunath, "Netra: A Toolbox For Navigating Large Image Databases," *Multimedia Systems*, Vol. 7, No. 3, Pp. 184–198, 1999.
- [27] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 24, No. 5, 2002.
- [28] I. Kompatsiaris, E. Triantafillou, and M. G. Strintzis, "Region-Based Colour Image Indexing and Retrieval," In Proc. Int'l Conf. On Image Processing (Icip'01), Thessaloniki, Greece, Vol. 1, 2001, Pp. 658–661.
- [29] M. Markkula and E. Sormunen, "End-User Searching Challenges Indexing Practices In The Digital Newspaper Photo Archive," *Information Retrieval*, Vol. 1, Pp. 259–285, 2000.
- [30] K. Barnard, P. Duygulu, N. De Freitas, D. Forsyth, D. Blei, and M. I. Jordan, "Matching Words and Pictures," *Journal Of Machine Learning Research*, Vol. 3, Pp. 1107–1135, 2003.
- [31] K. Barnard, P. Duygulu, R. Guru, P. Gabbur, and D. Forsyth, "The Effects Of Segmentation and Feature Choice In A Translation Model Of Object Recognition," In Proc. Ieee Conf. On Computer Vision and Pattern Recognition (Cvpr'03), 2003.
- [32] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 22, No. 8, Pp. 888–905, Aug. 2000.
- [33] J. Malik, S. Belongie, J. Shi, and T. Leung, "Textons, Contours and Regions: Cue Integration In Image Segmentation," In Proc. Ieee Int'l Conf. On Computer Vision, Corfu, Greece, Sept. 1999.
- [34] Y. Zeng and A. G. Constantinides, "Perceptual Saliency Weighted Segmentation Algorithm," In Proc. 7th Int'l Conf. On An Image Processing and Its Applications, July 1999.

- [35] T. Adamek, N. E. O’connor, and N. Murphy, “Region-Based Segmentation Of Images Using Syntactic Visual Features,” In Proc. 6th Int’lworkshop On Image Analysis For Multimedia Interactive Services (Wiamis’05), Montreux, Switzerland, Apr. 2005.
- [36] T. Adamek, Tesis Doctoral “Using contour information and segmentation for object registration, modeling and retrieval”, APPENDIX A, School of Electronic Engineering, Dublin city university, June 2006.
- [37] E. Engbers and A. Smeulders, “Design Considerations For Generic Grouping In Vision,” *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 25, No. 4, Pp. 445–457, Apr. 2003.
- [38] T. F. Cootes and C. J. Taylor, “Active Shape Models - Smart Snakes,” In Proc. British Machine Vision Conf. (Bmvc’92), Springer Verlag, 1992, P. 266.
- [39] M. E. Leventon, W. L. Grimson, and O. Faugeras, “Statistical Shape Influence In Geodesic Active Contours,” In Proc. *Ieee Conf. On Computer Vision and Pattern Recognition (Cvpr’00)*, Vol. I, 2000, Pp. 316–323.
- [40] A. Tsai, A. Yezzi, W. Wells, C. Tempany, D. Tucker, A. Fan, W. Grimson, and A. Willsky, “Curve Evolution Technique For Image Segmentation,” In Proc. *Ieee Conf. On Computer Vision and Pattern Recognition (Cvpr’01)*, Vol. I, 2001, Pp. 463–468.
- [41] D. Cremers, T. Kohlberger, and C. Schnorr, “Nonlinear Shape Statistics In Mumford-Shah Based Segmentation,” In Proc. *European Conf. On Computer Vision (Eccv’02)*, Vol. Ii, 2002, Pp. 93–108.
- [42] M. Rousson and N. Paragios, “Shape Priors For Level Set Representations,” In Proc. *European Conf. On Computer Vision (Eccv’02, Lncs 2351)*, 2002, Pp. 78–92.
- [43] M. B. Stegmann, “Active Appearance Models: Theory, Extensions and Cases,” Master’s Thesis, Informatics and Mathematical Modelling, Technical University Of Denmark, Lyngby, [Http://Www.Imm.Dtu.Dk/~Aam/](http://Www.Imm.Dtu.Dk/~Aam/), 2000.
- [44] M. Turk and A. Pentland, “Eigenfaces For Recognition,” *Journal Of Cognitive Neuroscience*, Vol. 3, No. 1, Pp. 71–86, 1991.
- [45] B. Marcotegui, P. Correia, F. Marqu´Es, R. Mech, R. Rosa, M. Wollborn, and F. Zanoguera, “A Video Object Generation Tool Allowing Friendly User Interaction,” In Proc. *Ieee Int’l Conf. On Image Processing (Icip’99)*, Kobe, Japan, 1999.
- [46] M. Cheriet, J. N. Said, and C. Y. Suen, “A Recursive Thresholding Technique For Image Segmentation,” *Ieee Trans. On Image Processing*, Vol. 7, No. 6, Pp. 918–920, 1998.
- [47] J. Mcqueen, “Some Methods For Classification and Analysis Of Multivariate Observations,” In Proc. *5th Berkely Symp. On Math. Stat. and Prob.*, Vol. 1, 1967, Pp. 281–296.
- [48] L. Vincent and P. Soille, “Watersheds In Digital Spaces: An Efficient Algorithm Based On Immersion Simulations,” *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 13, No. 6, Pp. 583–598, June 1991.
- [49] O. Morris, M. Lee, and A. Constantinides, “Graph Theory For Image Analysis: An Approach Based On The Shortest Spanning Tree,” In *Iee Proceedings*, Vol. 133, Apr. 1986, Pp. 146–152.
- [50] P. Salembier and L. Garrido, “Binary Partition Tree As An Efficient Representation For Filtering, Segmentation, and Information Retrieval,” In Proc. *Ieee Int’l Conf. On Image Processing (Icip’98)*, Chicago (Il), Usa, Oct. 1998.
- [51] S. Geman and D. Geman, “Stochastic Relaxation, Gibbs Distributions and The Bayesian Restoration Of Images,” *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 6, No. 6, Pp. 721–741, 1984.

- [52] Z. Tu and S. Zhu, "Image Segmentation By Data-Driven Markov Chain Monte Carlo," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 24, No. 5, Pp. 657–673, 2002.
- [53] W. Perkins, "Area Segmentation Of Images Using Edge Points," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 2, No. 1, Pp. 8–15, 1980.
- [54] N. Papamarkos, C. Strouthopoulos, and I. andreadis, "Multithresholding Of Colour and Gray-Level Images Through A Neural Network Technique," *Image and Vision Computing*, Vol. 18, Pp. 213–222, 2000.
- [55] J. Guo, J. Kim, and C. Kuo, "Fast and Accurate Moving Object Extraction Technique For Mpeg-4 Object Based Video Coding," In *Proc. Spie Visual Comm. and Image Processing*, 1999, Pp. 1210–1221.
- [56] L.Liu and S. Sclaroff, "Deformable Shape Detection and Description Via Modelbased Region Grouping," In *Proc. Ieee Conf. Computer Vision and Pattern Recognition (Cvpr'99)*.
- [57] D. Wang, "A Multi-Scale Gradient Algorithm For Image Segmentation Using Watersheds," *Pattern Recognition*, Vol. 30, No. 12, Pp. 2043–2052, 1997.
- [58] R. Nock and F. Nielsen, "Statistical Region Merging," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 26, No. 11, Pp. 1452–1458, Nov. 2004.
- [59] H. Lu, J.C. Woods and M. Ghanbari, "Image Segmentation By Binary Partition Tree", In *Electronics Letters* 17th August 2006 Vol. 42 No. 17
- [60] S. H. Kwok and A. G. Constantinides, "A Fast Recursive Shortest Spanning Tree For Image Segmentation and Edge Detection," *Ieee Trans. Image Processing*, Vol. 6, No. 2, Pp. 328–332, Feb. 1997.
- [61] R. Adams and L. Bischof, "Seeded Region Growing," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 16, No. 6, Pp. 641–647, June 1994.
- [62] E. Chalom and V. Bove, "Segmentation Of An Image Segueunce Using Multidimensional Image Attributes," In *Proc. Ieee Int'l Conf. On Image Processing (Icip'96)*, Lausanne, Switzerland, Vol. 2, Sept. 1996, Pp. 525–528.
- [63] T. Adamek, N.E. O'connor, G. Jones, "An Integrated Approach For Object Shape Registration and Modeling", Centre For Digital Video Processing, Dublin City University
- [64] T. Mcinerney and D. Terzopoulos, "Deformable Models In Medical Image Analysis: A Survey," *Medical Image Analysis*, Vol. 1, No. 2, Pp. 91–108, 1996.
- [65] V. Caselles, R. Kimmel, and G. Sapiro, "Geodesic Active Contours," *Int'l Journal Of Computer Vision (Ijcv)*, Vol. 22, No. 1, Pp. 61–79, 1997.
- [66] S. C. Zhu and A. Yuille, "Region Competition: Unifying Snakes, Region Growing and Bayes/Mdl For Multiband Image Segmentation," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 18, No. 9, Pp. 884–900, 1996.
- [67] A. L. Yuille, D. S. Cohen, and P. Hallinan, "Feature Extraction From Faces Using Deformable Templates," *Int'l Journal Of Computer Vision*, Vol. 8, No. 2, Pp. 99–112, 1992.
- [68] T. F. Cootes, C. J. Taylor, D. Cooper, and J. Graham, "Active Shape Models – Their Training and Application," *Computer Vision and Image Understanding*, Vol. 61, No. 1, Pp. 38–59, Jan. 95.
- [69] T. Cootes and C.J.Taylor, "Statistical Models Of Appearance For Computer Vision," University Of Manchester, [Http://Www.Isbe.Man.Ac.Uk/~Bim](http://Www.Isbe.Man.Ac.Uk/~Bim), Tech. Rep., Mar. 2004.
- [70] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Von Der Malsburt, R. Wurtz, and W. Konen, "Distortion Invariant Object Recognition In The Dynamic Link Architecture," *Ieee Trans. On Computers*, Vol. 42, Pp. 300–311, 1993.

- [71] U. Grenander and M. Miller, "Representations Of Knowledge In Complex Systems," *Journal Of The Royal Statistical Society*, Vol. 56, Pp. 249–603, 1993.
- [72] S. Sclaroff and L. Liu, "Deformable Shape Detection and Description Via Modelbased Region Grouping," *Ieee Trans. Pattern Anal. Machine Intell.*, Vol. 23, No. 5, 2001.
- [73] P. Correia and F. Pereira, "Objective Evaluation Of Relative Segmentation Quality," In *Proc. Int'l Conf. On Image Processing (Icip'00)*, Vancouver, Canada, Sept. 2000, Pp. 308–311.
- [74] Itu-R, "Methodology For The Subjective Assessment Of The Quality Of Television Pictures," *Recommendation Bt.500-7*, 1995.
- [75] E. D. Gelasca, T. Ebrahimi, M. C. Q. Farias, and S. K. Mitra, "Impact Of Topology Changes In Video Segmentation Evaluation," In *Proc. 5th Int'l Workshop On Image Analysis For Multimedia Interactive Services (Wiamis2004)*, Lisboa, Portugal, Apr.2004.
- [76] V. Mezaris, I. Kompatsiaris, and M. Strintzis, "Still Image Objective Segmentation Evaluation Using Ground Truth," In *Proc. 5th Cost 276 Workshop (2003)*, Berlin, 2003, Pp. 9–14.
- [77] P. Correia and F. Pereira, "Standalone Objective Evaluation Of Segmentation Quality," In *Proc. 3rd Int'l Workshop On Image Analysis For Multimedia Interactive Services (Wiamis'01)*, Tampere, Finland, May 2001.
- [78] P. Villegas, X. Marichal, and A. Salcedo, "Objective Evaluation Of Segmentation Masks In Video Sequences," In *Proc. Workshop On Image Analysis For Multimedia Interactive Services (Wiamis'99)*, Berlin, May 1999, Pp. 85–88.
- [79] P. Correia and F. Pereira, "Estimation Of Video Object's Relevance," In *Proc. European Conf. On Signal Processing (Eusipco'2000)*, Tampere, Finland, Sept. 2000.
- [80] I. Kompatsiaris and M.G. Strintzis, "Content-Based Representation Of Colour Image Sequences", *Informatics and Telematics Institute*, 2001.
- [81] J. Macqueen, "Some Methods For Classification and Analysis of Multivariate Observations".
- [82] J. Pérez, M. F. Henriques, R. Pazos, L. Cruz, G. Reyes, J. Salinas, A. Mexicano, "La IO aplicada a la solución de problemas regionales", 2º Taller Latino Iberoamericano de Investigación de Operaciones.
- [83] L. Bottou, Y. Bengio: *Convergence Properties of the K-means Algorithms*. *Advances in Neural Information Processing Systems*. MIT Press, 1995.
- [84] G. Hamerly, C. Elkan: *Alternatives to the K-means Algorithm that Find Better Clusterings*. *Proc. 11th International Conf. On Information and Knowledge Management CIKM'02*. ACM. Virginia, USA (2002).
- [85] T. Kanungo, N.S. Netanyahu, A.Y. Wu: *An Efficient Kmeans Clustering Algorithm: Analysis and Implementation*. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 24, No. 7 (2002).
- [86] T. Kanungo, D.M. Mount, N.S. Netanyahu, C.D. Piatko, R. Silverman, A.Y. Wu: *A Local Search Approximation Algorithm for k-Means Clustering*. *Proc. 18th Annual ACM Symposium on Computational Geometry (SoCG'02)*. Barcelona, Spain (2002) 10-18.
- [87] D. Pelleg, A. Moore: *X-means: Extending K-means with Efficient Estimation of the Number of Clusters*. *Proc. 17th International Conf. on Machine Learning (2000)*.
- [88] J.M. Peña, J.A. Lozano, P. Larrañaga: *An Empirical Comparison of Four Initialization Methods for the K-Means Algorithm*. *Dept. of Computer Science and Artificial Intelligence, University of the Basque Country, San Sebastian, España*.

- [89] I. Kompatsiaris and M. G. Strintzis, "Spatiotemporal Segmentation and Tracking of Objects for Visualization of Videoconference Image Sequences," *IEEE Trans. Circuits and Systems for Video Technology*, to appear 2000.
- [90] M. Petrou, "Image Processing: Dealing with Texture", Ed. John Wiley & Sons, 2006.
- [91] A. Fernandez, Tesis Doctoral "Estudio de Técnicas basadas en la Transformada Wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales", Universidad Politécnica de Valencia, Febrero 2007.
- [92] J. Zhang and T. Tan, "Brief review of invariant texture analysis methods", *Pattern Recognition*. 2002, vol. 35, nº 3, p. 735-747.
- [93] Randen, Tesis Doctoral "T. Filter and filter bank design for image texture recognition", Norwegian University of Science and Technology. Stavanger, 1997.
- [94] J. Mao and A. Jain, "Texture classification and segmentation using multiresolution simultaneous autoregressive models", *Pattern recognition*, 1992, vol. 25, nº2, p. 173-188.
- [95] M. Unser, "Texture classification and segmentation using wavelet frames", *IEEE Trans. on Image Processing*, vol. 4, no.11, pp. 1549-1560, Nov. 1995.
- [96] K. Skretting and J. Husoy, "Texture classification using sparse frame based representations", *EURASIP journal on applied signal processing*, 2006: vol. 2006, artículo 52561, 11p.
- [97] T. Adamek, Tesis Doctoral "Using contour information and segmentation for object registration, modeling and retrieval", School of Electronic Engineering, Dublin city university, June 2006. p. 31-33
- [98] P. Smets, "The Normative Representation Of Quantified Beliefs By Belief Functions," *Artificial Intelligence*, Vol. 92, Pp. 229–242, 1997.
- [99] T. Vlachos and A. G. Constantinides, "A Graph-Theoretic Approach To Color Image Segmentation and Contour Classification," In *Proc. 4th Int'l Conf. Image Processing and Its Applications*, Maastricht, The Netherlands, Apr. 1992.
- [100] Y. Zeng, "Perceptual Segmentation Algorithm and Its Application To Image Coding," In *Proc. Int'l Conf. On Image Processing*, 1999, Pp. 820–824.
- [101] E. Tuncel and L. Onural, "Utilization Of The Recursive Shortest Spanning Tree Algorithm For Video-Object Segmentation By 2-D Affine Motion Modeling," *Ieee Trans. Circuits Syst. Video Technol.*, Vol. 10, No. 5, Pp. 776–781, Aug. 2000.
- [102] A. Alatan, L. Onural, M. Wollborn, R. Mech, E. Tuncel, and T. Sikora, "Image Sequence Analysis For Emerging Interactive Multimedia Services - The European Cost 211 Framework," *Ieee Trans. Circuits Syst. Video Technol.*, Vol. 8, No. 7, Pp. 802–813, Nov. 1998.
- [103] J. Hafner, H. Sawhney, W. Equitz, M. Flickner, and W. Niblack, "Efficient Color Histogram Indexing For Quadratic Form Distance Functions," *Ieee Trans. Pattern Anal. and Machine Intell.*, Vol. 17, No. 7, Pp. 729–736, July 1995.
- [104] Y. Chen and H. Sundaram, "Estimating The Complexity Of 2d Shapes," In *Proc. Int'l Workshop On Multimedia Signal Processing (Mmsp'05)*, May 2005.
- [105] S. Loncaric, "A Survey Of Shape Analysis Techniques," *Pattern Recognition*, Vol. 31, No. 5, Pp. 983–1001, 1998.
- [106] A. Dempster, "Upper and Lower Probabilities Induced By Multivalued Mapping," *Annals Of Mathematical Statistics*, Vol. 38, Pp. 325–339, 1967.
- [107] G. Shafer, "A Mathematical Theory Of Evidence," Princetown Univ. Press, Princetown New Jersey, 1976.
- [108] J. Bezdek, "Fuzziness Vs. Probability - The N-Th Round," *Ieee Trans. On Fuzzy Systems*, Vol. 2, No. 1, Pp. 1–42, 1994.

- [109] L. Zadeh, "Fuzzy Sets As A Basis For A Theory Of Possibility," *Fuzzy Sets and Systems*, Vol. 1, Pp. 3–28, 1978.
- [110] D. Dubois, J. Lang, and H. Prade, "Automated Reasoning Using Possibilistic Logic: Semantics, Belief Revision, and Variable Certainty Weights," *Ieee Trans. On Knowledge and Data Engineering*, Vol. 6, Pp. 64–71, 1994.
- [111] P. Smets and R. Kennes, "The Transferable Belief Model," *Artificial Intelligence*, Vol. 66, No. 2, Pp. 191–234, 1994.
- [112] J.F. Reyes Saldaña y R. García Flores, "Toma de decisiones mediante técnicas de razonamiento incierto", *Ingenierías*, vol. VIII, No.28, julio-septiembre 2005.
- [113] A. Al-Ani and M. Deriche, "A New Technique For Combining Multiple Classifiers Using The Dempster-Shafer Theory Of Evidence," *Journal Of Artificial Intelligence Research*, Vol. 17, Pp. 333–361, 2002.
- [114] V. Girondel, A. Caplier, and L. Bonnaud, "A Belief Theory-Based Static Posture Recognition System For Real-Time Video Surveillance Applications," In *Proc. Ieee Int'l Conf. On Advanced Video and Signal Based Surveillance (Avss'05)*, Como, Italy, Sept. 2005.
- [115] P. Smets, E. Mamdami, D. Dubois, and H. Prade, *Non-Standard Logics For Automated Reasoning*, Ser. Isbn 0126495203. Academic Press, Harcourt Brace Jovanovich Publisher, 1988.
- [116] T.Adamek,N. E. O'Connor, "Stopping Region-Based Image Segmentation at Meaningful Partitions", Centre for Digital Video Processing, Dublin City University, Dublin 9, Ireland. 2007.
- [117] P. Rosin, "Unimodal Thresholding," *Pattern Recognition*, Vol. 34, No. 11, Pp. 2083–2096, Nov. 2001.



## **PRESUPUESTO**

**1) Ejecución Material**

- Compra de ordenador personal (Software incluido)..... 2.000 €
- Alquiler de impresora láser durante 6 meses ..... 50 €
- Material de oficina ..... 150 €
- Total de ejecución material ..... 2.200 €

**2) Gastos generales**

- 16 % sobre Ejecución Material ..... 352 €

**3) Beneficio Industrial**

- 6 % sobre Ejecución Material ..... 132 €

**4) Honorarios Proyecto**

- 640 horas a 15 € / hora..... 9600 €

**5) Material fungible**

- Gastos de impresión..... 60 €
- Encuadernación..... 200 €

**6) Subtotal del presupuesto**

- Subtotal Presupuesto..... 12060 €

**7) I.V.A. aplicable**

- 16% Subtotal Presupuesto ..... 1929.6 €

**8) Total presupuesto**

- Total Presupuesto..... 13989,6 €

Madrid, Junio de 2009

El Ingeniero Jefe de Proyecto

Fdo.: Elena Ortiz Muñoz  
Ingeniera Superior de Telecomunicación



## PLIEGO DE CONDICIONES

Este documento contiene las condiciones legales que guiarán la realización, en este proyecto, de contribuciones a técnicas de segmentación de imágenes. 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.