

Proyecto Fin de Grado
Ingeniería de las Tecnologías de
Telecomunicación

Análisis de la aplicación de algoritmos de K-means y
Continuous Max-Flow a la segmentación de
imágenes en color

Autor: Luis Fernando Pinto Sánchez-Matamoros

Tutor: José Antonio Pérez Carrasco

Dep. Teoría de la Señal y Comunicaciones
Escuela Técnica Superior de Ingeniería
Universidad de Sevilla

Sevilla, 2015



Proyecto Fin de Grado
Ingeniería de las Tecnologías de Telecomunicación

Análisis de la aplicación de algoritmos de K-means y Continuous Max-Flow a la segmentación de imágenes en color

Autor:

Luis Fernando Pinto Sánchez-Matamoros

Tutor:

José Antonio Pérez Carrasco

Profesor Ayudante Doctor

Dep. de Teoría de la Señal y Comunicaciones

Escuela Técnica Superior de Ingeniería

Universidad de Sevilla

Sevilla, 2015

Proyecto Fin de Grado: Análisis de la aplicación de algoritmos de K-means y Continuous Max-Flow a la segmentación de imágenes en color

Autor: Luis Fernando Pinto Sánchez-Matamoros

Tutor: José Antonio Pérez Carrasco

El tribunal nombrado para juzgar el Proyecto arriba indicado, compuesto por los siguientes miembros:

Presidente:

Vocales:

Secretario:

Acuerdan otorgarle la calificación de:

Sevilla, 2015

El Secretario del Tribunal

A mis padres

A mis hermanos

Agradecimientos

Me gustaría utilizar estas líneas para agradecer a aquellas personas que me han ayudado a lo largo de mi carrera universitaria.

En primer lugar, expresar mi agradecimiento a D.José Antonio Pérez Carrasco por haberme brindado la oportunidad de trabajar con él, haberme prestado su apoyo a lo largo del desarrollo de este trabajo y su confianza para asignarme este proyecto.

A mis padres, Leonor y Fernando, a los que les debo ser lo que soy. Me han enseñado el valor de una buena educación y apoyado durante todo este episodio de mi vida. A mi hermanos Leonor e Ismael que me han aportado todos los valores que un hermano puede pretender y me han dado fuerzas y consejos como ninguna otra persona podría hacerlo.

También me gustaría agradecer a mis compañeros de carrera Jon Senra, Jesús Rivera y Pablo Pizarro, con los que he compartido grandes momentos y espero seguir compartiéndolos en esta nueva aventura que comenzamos juntos.

Hacer especial mención a mi tío Javier, el cual fue la persona que más me influyó a la hora de elegir la carrera.

A todos ellos, muchas gracias.

Luis Fernando Pinto Sánchez-Matamoros

Sevilla, 2015

Resumen

En el campo del procesamiento de imagen se realizan cada año numerosos avances en relación a los algoritmos y técnicas para el tratamiento y segmentación de imágenes. En este trabajo se propondrá un uso conjunto de dos algoritmos utilizados para la segmentación de imágenes por medio del color. El primero de ellos, un algoritmo basado en clústering, K-means. El segundo, categorizado como algoritmo de minimización de la función de energía, Continuous Max-Flow.

En primer lugar, se introducirá el objetivo principal del trabajo, así como los resultados que se pretenden conseguir mediante la realización del mismo y la metodología aplicada. También se detallará la búsqueda necesaria que se ha llevado a cabo para el desarrollo de la bibliografía.

En segundo lugar, se definirá un marco teórico necesario para el correcto entendimiento del trabajo. Además, se mostrarán las distintas imágenes usadas en el desarrollo del proyecto así como la forma de obtener los resultados numéricos.

En tercer lugar, se mostrarán los algoritmos utilizados y los experimentos realizados para la obtención de los valores numéricos. También se incluirán las tablas con los resultados más relevantes.

Por último, se comentarán las conclusiones en base a los resultados obtenidos. También se propondrán las posibles líneas futuras que ofrece este trabajo.

Abstract

In the image-processing field, there is a big progress on the treatment and segmentation techniques of images, every year.

In this paper, a new joint use of two image segmentation algorithms is proposed. The first one is based on clustering, K-means. The second one is based on energy minimization, Continuous Max-Flow.

Firstly, the main objective of the project will be introduced. Moreover we will explain what we want to get and the methodology. Also, we will detail how we have got all the references.

Secondly, we will define a theoretical framework to allow us to understand this project. Besides, we will show all the different images that we have used in this project, as well as, how we have got all the numerical results.

Thirdly, we will show all the algorithms we have used and all the experiments that have been made to get those numerical results. Tables are also included with the most relevant results

Finally, conclusions will be discussed based on the results. Besides possible future development will be proposed according to this project.

Índice

Agradecimientos	ix
Resumen	xi
Abstract.....	xiii
Índice	xiv
Índice de Tablas.....	xvi
Índice de Figuras	xvii
Notación.....	xix
1 Introducción.....	1
1.1 Justificación.....	1
1.2 Objetivos	2
1.3 Estado del arte.....	2
1.4 Metodología	2
1.4.1 Base de datos fama	3
1.4.2 CIE.....	3
1.4.3 Mathworks.....	3
1.4.4 IEEE	3
1.4.5 Base de datos de Berkeley.....	3
1.4.6 Tesis doctorales y proyectos.....	4
1.4.7 Google Academics	4
2 Desarrollo	5
2.1 Marco Teórico	5
2.1.1 El color.....	5
2.1.1.1 Parámetros básicos en el tratamiento del color	6
2.1.1.2 Teoría tricromática	7
2.1.1.3 Iluminantes	8
2.1.2 Modelos de color.....	9
2.1.2.1 Modelo RGB.....	9
2.1.2.2 Modelo CIELAB	10
2.1.3 Distancias de color	12
2.1.4 Segmentación de imágenes en color	14
2.1.4.1 Segmentación basada en agrupamiento (<i>clustering</i>).....	14
2.1.4.2 Segmentación basada en la optimización de una función de energía	16
2.2 Bases de datos de imágenes.....	22
2.3 Imágenes utilizadas y formas de evaluación.....	22
2.3.1 Imágenes utilizadas	22
2.3.1.1 Imágenes test	22
2.3.1.2 Imágenes reales.....	23
2.3.2 Formas de evaluación	24
2.3.2.1 Bordes	24
2.3.2.2 Regiones.....	27

3	Resultados	33
3.1	<i>Algoritmos utilizados.....</i>	33
3.1.1	Algoritmo de K-means.....	33
3.1.2	Algoritmo de Continuous Max-Flow	34
3.2	<i>Experimentos realizados para la consecución de los resultados.....</i>	35
3.2.1	Kmeans	35
3.2.2	K-means + C.Max-Flow.....	35
3.3	<i>Resultados.....</i>	35
3.3.1	Imágenes Squares.....	36
3.3.2	Imágenes Reales	45
3.3.3	Valores promedio	54
4	Conclusiones y líneas futuras.....	55
4.1	<i>Conclusiones.....</i>	55
4.2	<i>Líneas Futuras</i>	57
	Referencias	58

ÍNDICE DE TABLAS

Tabla 1. Resultados de la imagen 1.bmp.	37
Tabla 2. Resultados de la imagen 2.bmp.	38
Tabla 3. Resultados de la imagen 10.bmp.	39
Tabla 4. Resultados de la imagen 11.bmp.	40
Tabla 5. Resultados de la imagen 12.bmp.	41
Tabla 6. Resultados de la imagen 13.bmp.	42
Tabla 7. Resultados de la imagen 14.bmp.	43
Tabla 8. Resultados de la imagen 15.bmp.	44
Tabla 9. Resultados de la imagen 3096.jpg.	46
Tabla 10. Resultados de la imagen 8068.jpg.	47
Tabla 11. Resultados de la imagen 12003.jpg.	48
Tabla 12. Resultados de la imagen 35070.jpg.	49
Tabla 13. Resultados de la imagen 118035.jpg.	50
Tabla 14. Resultados de la imagen 124084.jpg.	51
Tabla 15. Resultados de la imagen 207056.jpg.	52
Tabla 16. Resultados de la imagen 238011.jpg.	53
Tabla 17. Valores promedio de las imágenes.	54

ÍNDICE DE FIGURAS

Figura 1. Espectro de luz visible.	6
Figura 2. Absorción espectral (normalizada) de los conos.	6
Figura 3. Mezcla aditiva de los tres colores primarios Rojo, Verde y Azul.	8
Figura 4. Arriba a la izq.: iluminante D65, luz de medio día. Arriba a la der.: iluminante A, bombilla incandescente. Abajo a la izq.: iluminante F2, Tubo fluorescente. Abajo a la der.: iluminante D50, luz de día.	8
Figura 5. Representación Tridimensional del espacio RGB.	9
Figura 6. Representación Tridimensional del espacio de color $L^*a^*b^*$.	12
Figura 7. Arriba a la izq.: imagen original. Arriba a la der.: segmentación realizada mediante K-means con la distancia euclídea y 4 etiquetas. Abajo a la izq.: segmentación realizada mediante K-means con la distancia CIEDE94 y 4 etiquetas. Abajo a la der.: segmentación realizada mediante K-means con la distancia CIEDE2000 y 4 etiquetas.	14
Figura 8. Agrupamiento de muestras aleatorias según dos etiquetas.	16
Figura 9. Representación del algoritmo de Max-Flow en tiempo discreto (izq.) y tiempo continuo (der.).	17
Figura 10. Representación de Max-Flow en tiempo Continuo para 2 etiquetas (izq.) y n etiquetas (der.).	19
Figura 11. Imágenes utilizadas como test, squares.	23
Figura 12. Imágenes utilizadas como test, reales.	23
Figura 13. A la izq. la imagen de squares. A la der. su imagen de bordes óptima obtenida con nuestro algoritmo.	24
Figura 14. Representación de la imagen real con sus cinco <i>ground truth</i> .	25
Figura 15. A la izq. la imagen de bordes obtenida de una imagen tipo squares. A la der. una imagen de bordes obtenida de una imagen tipo real.	26
Figura 16. Imágenes de bordes de muestra. Arriba a la izq.: se muestra la imagen de <i>ground truth</i> . Arriba a la der.: se encuentra la imagen de bordes resultante de la segmentación. Abajo, la imagen diferencia.	27
Figura 17 Imágenes squares utilizadas como <i>ground truth</i> en el método de regiones.	28
Figura 18. Imagen de <i>ground truth</i> utilizada para las imágenes del grupo reales	28
Figura 19. A la izq. la imagen original. A la der. la imagen tras ordenarse las etiquetas de mayor a menor.	29
Figura 20. Arriba a la izq.: imagen <i>ground truth</i> de regiones. Arriba a la der.: imagen de bordes segmentada con el algoritmo de Continuous Max-Flow. Abajo: imagen diferencia de regiones.	31
Figura 21. Imagen 8068.jpg con el mismo número de etiquetas que el <i>ground truth</i> (4 etiquetas) y segmentada con el algoritmo C.Max-Flow y distancia CIEDE 00.	55
Figura 22. Imagen 35070.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2 etiquetas.	56
Figura 23. Imagen 118035.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2	

etiquetas.

56

Figura 24. Imagen 207056.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2 etiquetas.

57

Notación

\equiv	Equivalente
$=$	Igual
$>$	Mayor que
$<$	Menor que
\geq	Mayor o igual que
\leq	Menor o igual que
e.o.c	En otro caso
$\sum_b^c a$	Sumatorio sobre a desde b hasta c
\cup	Unión
\cap	Intersección
\emptyset	Conjunto vacío
Ω	Conjunto entero
$ $	Módulo
\subset	Subconjunto de
\forall	Para todo
sin	Función de seno
\tan^{-1}	Función de tangente inversa
\in	Pertenece a
\int	Integral
$[]$	Intervalo cerrado

1 INTRODUCCIÓN

“There is nothing worse than a sharp image of a fuzzy concept”

- Ansel Adams -

Este capítulo pretende servir de introducción para explicar detalladamente la justificación, el porqué de la realización de este trabajo, el objetivo, qué queremos conseguir con la elaboración de este proyecto, el estado del arte, qué estudios e investigaciones existen en relación a este tema y la metodología, el trabajo y la búsqueda de recursos que apoyan este proyecto. En primer lugar, se detallará la justificación del trabajo. En segundo lugar, se marcarán unos objetivos antes de comenzar con el desarrollo. La consecución o no de estos, será un tema que se tratará en capítulos siguientes. En tercer lugar, se mostrarán todos los estudios e investigaciones que se han ido desarrollando estos últimos años en relación al tema que en este proyecto se hace referencia. Por último, se describirá la metodología que se ha llevado a cabo a lo largo del trabajo.

1.1 Justificación

La segmentación de imágenes es un proceso clave en el análisis y comprensión de las mismas. Sin embargo, llevar a cabo este proceso no es del todo trivial. Esto es debido a que en el ámbito de la segmentación de imágenes existen muchos factores y muy diversos como pueden ser el ruido, las características del objeto, texturas,...

El proceso de dividir una imagen digital en múltiples regiones (grupo de píxeles), que guardan algún tipo de característica común, es lo que se denomina segmentación de imágenes.

El resultado de la segmentación es un grupo de píxeles que cubren toda la superficie de la imagen. Todos los píxeles de una misma región guardan entre sí características similares, como pueden ser el color, textura, intensidad... Las regiones adyacentes son significativamente diferentes con respecto a las mismas características.

Como ya he comentado anteriormente, este proceso no es fácilmente aplicable debido a la gran variedad de imágenes (no existe un método de segmentación estándar). Además de ello, existen otros grandes problemas a la hora de la segmentación de imágenes como puede ser la carga computacional necesaria para realizarla.

Es por todo esto, que cada vez más se llevan a cabo investigaciones y avances sobre nuevos algoritmos de segmentación que sean capaces de solventar estos problemas.

Este trabajo se centra precisamente en el estudio de resultados del uso de dos algoritmos altamente conocidos en el ámbito de la segmentación de imágenes en color, como son K-means y Continuous Max-Flow, para satisfacer la búsqueda del etiquetado óptimo de cada pixel o nodo, con respecto a una función de energía.

El primer algoritmo, K-means, se centra en el agrupamiento de los píxeles siguiendo una característica en

común. En este caso, el color. También se denominan técnicas de segmentación basadas en *clustering* (agrupamiento).

El segundo de ellos, Continuous Max-Flow, se fundamenta en la segmentación de imágenes basada en la optimización de una función de energía.

En los apartados siguientes se explicará de forma más detallada el funcionamiento de ambos algoritmos y la actuación conjunta de ambos.

1.2 Objetivos

Mediante la actuación coordinada de los dos algoritmos anteriormente citados, se buscará y analizará si mediante este proceso se puede mejorar la segmentación sobre algunas imágenes a color, que por sí solo el algoritmo de K-means no es capaz de realizar correctamente. También se buscará reducir los costes computacionales, reduciendo el número de iteraciones de K-means y por tanto el tiempo de segmentación.

Además, este trabajo también nos servirá para establecer cuáles son las diferencias aportadas entre el uso de las diferentes distancias que la CIE propone y que en este trabajo ponemos en práctica (véase el punto 2.1.3).

1.3 Estado del arte

En la actualidad, existen numerosos estudios e investigaciones relacionados con el campo del procesamiento de imágenes que se centran en intentar resolver los problemas de minimización de una función de energía a través de métodos basados en corte de grafos, como Min-Cut o Max-Flow. Algunos de esos recientes estudios se basan en la búsqueda de soluciones para: segmentación de imágenes [20,27,28], reconstrucción de escenas tomadas desde varias cámaras [29], reconstrucciones de modelos en 3D y *shape fitting* [30,31,32], síntesis de imágenes y fotomontajes [33], etc.

Normalmente, los problemas de minimización de energía en tiempo discreto se basan en la búsqueda del corte mínimo sobre un apropiado grafo. Igualmente, pueden ser resueltos de una manera eficiente mediante la búsqueda del máximo flujo correspondiente al mismo grafo. Esto es lo que se denomina la teoría del Min-Cut y Max-Flow. Ha habido muchísimas investigaciones en relación a este tema en los últimos años [19, 20].

Recientes estudios en el campo de la investigación de segmentación de imágenes muestran, que formular los problemas de Min-Cut en tiempo continuo configurado de una manera correcta, evita una conversión predefinida y permite encontrar de manera más rápida unas soluciones globales a través de *Convex Optimization*.

G.Strangs [21, 22] fue el primero en estudiar los problemas de Min-Cut y Max-Flow en un dominio temporal continuo. Existen estudios relacionados [27,34] donde Appleton y colaboradores propusieron una mínima aproximación superficial basada en bordes en el dominio continuo para la segmentación de objetos 2D y 3D. Además, Chan y colaboradores fueron capaces de formular la segmentación de imágenes con dos regiones como la resolución de un problema de minimización convexa [23].

Sin embargo, en contraste con la dualidad entre Max-Flow y Min-Cut en tiempo discreto, donde los algoritmos de Min-Cut son definidos de la forma de Max-Flow, en tiempo continuo, el modelado de Max-Flow como una formulación dual de (2.37) está todavía inmerso en investigaciones y desarrollos.

1.4 Metodología

En este apartado especificaremos la metodología que hemos seguido a lo largo de todo el proyecto para la búsqueda de la información que contiene. Así, iremos nombrando las diferentes fuentes de información a las que hemos acudido y daremos algunos detalles de cómo se ha realizado la búsqueda para llegar a los diferentes títulos de las referencias.

1.4.1 Base de datos fama

Fama es el catálogo donde se encuentran todos los recursos de lectura de los que dispone la Universidad de Sevilla, libros, artículos, monografías...

Esta base de datos ha sido una de las más recurridas debido a que, desde ella, se puede acudir a numerosos artículos y libros, incluso hace referencia a artículos de los que dispone la Universidad pero que se encuentran alojadas en otra base de datos.

Para la búsqueda de información sobre los conceptos teóricos que se expondrán más adelante fue bastante útil puesto que realizando una búsqueda mediante palabras claves tales como “space color”, “color”, “luz”... fue suficiente para encontrar unas muy buenas referencias.

Sin embargo, para algunos temas tales como “segmentation” “clustering” o “K-means” los resultados a los que nos llevaban no eran del todo concretos sobre lo que buscábamos. Así que decidimos que en lugar de buscar por palabras claves que tuviesen que ver con este proyecto, buscar por autores que supiésemos nosotros que utilizaban muchos de los términos que aquí aplicamos.

1.4.2 CIE

CIE, *la comisión internacional del color*, dispone en su página web de una aplicación donde se pueden encontrar la lista de términos más conocidos en el ámbito del color y su definición normalizada.

Además, dispone de un gran conjunto de artículos publicados, de los cuales, muchos de ellos tienen que ver con el color.

Es por ello, que acudimos a la página web en busca de recursos que nos fuesen útiles para poder desarrollar con fundamento la parte teórica del proyecto.

Utilizamos, por tanto, su buscador de términos para poder dar con criterio definiciones en este proyecto.

Además, también hicimos uso de su gran lista de artículos gratuitos publicados para que nos sirvieran, junto con otras publicaciones, a la hora de realizar una explicación básica sobre conceptos teóricos más profundos que la mera descripción de un término, como por ejemplo, en la explicación de los espacios de color.

1.4.3 Mathworks

Mathworks, además de ser conocidos por el programa Matlab, también dispone de un buscador en el cual, las personas que van realizando algoritmos en código Matlab las van aportando a un pequeño foro interno.

Además de incluir los algoritmos en código matlab, suelen incluir una pequeña referencia a un artículo donde ha sido usado.

Así pues, una de las búsquedas que realizamos en ésta página fue continuous Max-Flow algorithm, justo el algoritmo que utilizamos en este proyecto.

Al descargar el contenido, además del algoritmo, se incluía un documento al cual hemos hecho referencia en el proyecto.

1.4.4 IEEE

IEEE, *el instituto de ingenieros electrónicos y eléctricos*, es un referente en cuanto a normativa se refiere. Sin embargo, no acudimos de forma directa a su buscador de normas, sino que fue a través de fama donde encontramos los artículos que se hacían referencia en otros artículos.

Por lo tanto, el haber hecho uso del buscador de normas de la que dispone, fue como resultado de haber leído distintos artículos en los cuales se hace referencia a muchas de las normas de las que nosotros mismos ahora hacemos referencia.

1.4.5 Base de datos de Berkeley

La base de datos de Berkeley es una de las bases de datos de imágenes más usadas en el ámbito de procesamiento de imagen, puesto que en ella, se encuentran una gran cantidad de imágenes de las que se puede

hacer uso para la pruebas de distintos algoritmos de segmentación. Posteriormente, se explicará más en profundidad.

El uso de esta base de datos es el resultado de ver que en muchos de los artículos de segmentación de imágenes, se hacía uso de las imágenes que en dicha base se encuentran [14]. Es por ello, que me pareció correcto utilizarla para mi proyecto, puesto que, es usada en el ámbito de la investigación a la que yo mismo estoy haciendo referencia con mi proyecto.

1.4.6 Tesis doctorales y proyectos

Como se ha comentado antes, el uso que se le ha dado al catálogo fama en este proyecto fue sobre todo para encontrar artículos relacionados con la segmentación de imágenes.

A pesar de ello, el uso que se le ha dado ha sido también para encontrar trabajos de fin de carrera y tesis doctorales que guardasen relación con lo que en este proyecto se habla.

1.4.7 Google Academics

Aunque google como buscador sobre documentos de investigación puede no tener muy buena fama, puesto que, los resultados que muestra no suelen ser fuentes muy fiables, google dispone de un buscador específico donde los únicos resultados que verterá serán artículos académicos.

2 DESARROLLO

“As with sound, images are subjective. You and I may not see the same color as red, but we will probably agree that the image on the screen is a digital image or film image, based on contrast, bit depth and refresh rate”

- John Dykstra-

En este capítulo se explicará en profundidad el desarrollo de todo el trabajo. Se introducirá un marco teórico necesario para el buen seguimiento del proyecto. Se detallarán los resultados conseguidos y además, se describirá el desarrollo para conseguirlos. Por último, se explicarán las conclusiones a las que hemos llegado gracias a los datos conseguidos y compararlos con los objetivos que nos habíamos marcado en secciones anteriores.

2.1. Marco Teórico

Esta sección pretende ser una introducción a una serie de conceptos básicos para poder entender el desarrollo del proyecto. Se introducirá, en primer lugar, el concepto básico de color. Luego, se introducirá un término muy ligado al anterior, como son los modelos representativos de color y las distancias utilizados en los mismos. Después, se dará un breve repaso sobre la segmentación de las imágenes y los métodos más populares. Por último, se nombrarán las bases de datos de imágenes que han servido de apoyo a este trabajo.

2.1.1 El color

Según el vocabulario internacional de la luz, la definición de la misma es “Cualquier radiación capaz de causar una sensación visual” (CIE, 1987). La luz visible es una radiación electromagnética comprendida en un rango de 380 a 780 nanómetros.

Para Wyszecki & Stiles (1982), “*Color is that aspect of visual perception by which an observer may distinguish differences between two structure-free fields of views of the same size and shape, such as may be caused by differences in the spectral composition of the radiant energy concerned in the observation*”.

“*El color es el aspecto de la percepción visual por el cual un observador es capaz de distinguir diferencias entre dos campos de visión del mismo tamaño, forma y estructura causadas por las diferencias en la composición espectral de la energía radiante implicada en la observación*”

El color no es más que la respuesta del sistema visual humano a la excitación recibida de la radiación electromagnética comprendida en el rango del espectro de luz visible. Al incidir en la retina del ojo, la radiación es captada e interpretada como un color. Dependiendo de la longitud de onda se caracterizará de una manera u otra. La distribución espectral de la luz visible se representa en la siguiente figura (Fig.1).

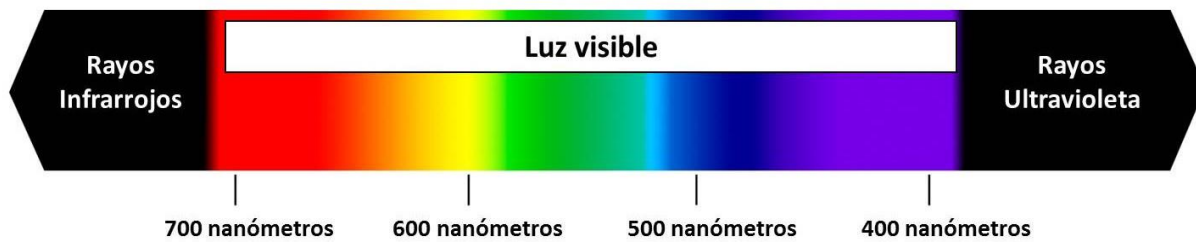


Figura 1. Espectro de luz visible.

En la retina del ojo humano se encuentran los dos tipos de células fotorreceptoras responsables de convertir esa radiación en color. El primero de ellos son los bastones, se encargan de la visión escotópica (blanco y negro). El segundo grupo son los conos, encargados de la visión fotópica (visión en color). Siendo este último grupo el menos numeroso en la retina.

Existen tres tipos de conos que responden a diferentes curvas espectrales, rojo, azul y verde, tal y como se aprecia en la siguiente figura (Fig. 2)

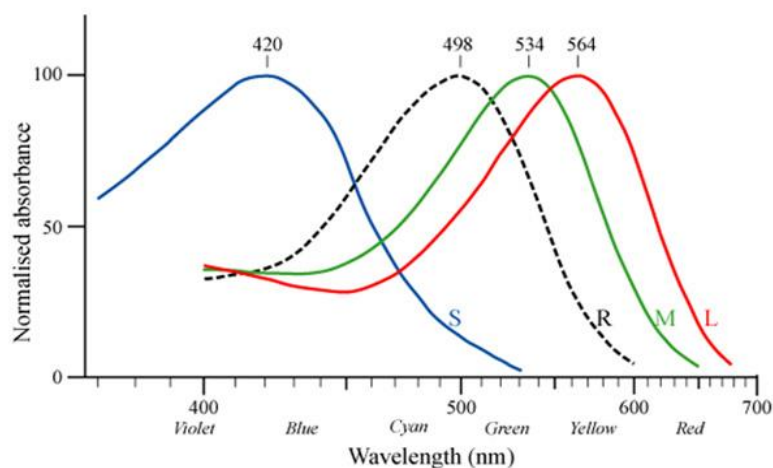


Figura 2. Absorción espectral (normalizada) de los conos.

2.1.1.1 Parámetros básicos en el tratamiento del color

En el tratamiento de imágenes a color existen algunos parámetros básicos que hay que tener en cuenta:

- Intensidad (I):** es una medida del flujo de potencia que atraviesa una superficie perpendicular a la dirección de propagación, y se expresa en unidades de vatios por metro cuadrado.
- Brillo (Br):** “es un atributo relacionado con la sensación visual en relación a que un área parezca que emita más o menos luz”. (CIE)
- Luminancia (Y):** es la potencia radiada ponderada por una función de sensibilidad espectral característica de la superficie sobre la que incide. En nuestro caso, esa ponderación vendrá dada por la caracterización de la visión humana.
- Luminosidad (L*):** es la respuesta perceptiva no lineal a la luminancia que tiene la visión humana. La no linealidad es prácticamente logarítmica. Debido a esta característica encontramos que dos áreas

igualmente iluminadas dan sensación de distinto color.

- e) **Tono o matiz (H):** “Atributo de la percepción visual por el cual un área se parece más a uno de los siguientes colores: rojo, amarillo, verde y azul, o una combinación de pares adyacentes de estos colores considerados en un anillo cerrado” (CIE). Así pues, la caracterización de un objeto al decir que es azul o amarillo lo especifica su tono, puesto que es la longitud de onda dominante percibida por el observador.
- f) **Saturación (S):** Relación del colorido de un área en relación con su proporción de brillo”(CIE). Así pues, si atribuimos que un color más blanco emite más luz, la saturación consistiría en la relación de pureza de un color mezclado con blanco. Por lo que, un color puro estaría totalmente saturado al no contener luz blanca.

2.1.1.2 Teoría tricromática

A la hora de hablar del color, es necesario saber mediante qué fenómeno se forma y cómo se estudian los fundamentos básicos del color. La ciencia que se encarga de cuantificar y describir el color de la percepción humana a través de modelos matemáticos se denomina colorimetría.

Primero Newton (1672) y luego Young (1802), ya establecieron las bases de la colorimetría y la teoría tricromática, al enunciar que muchos estímulos de color se pueden igualar mediante la mezcla aditiva de tres estímulos denominados primarios (Fig. 3).

La teoría tricromática se fundamenta en una serie de 4 leyes básicas enunciadas por el físico alemán Grassmann (1853), de la trivarianza, de la luminancia, de la proporcionalidad y de la aditividad, pero se pueden resumir en estas dos:

La primera, base de la colorimetría, establece que la potencia radiante de un estímulo Q es equivalente a la suma de las potencias radiantes de los tres estímulos primarios (R, G, B), siempre que se mezclen en las proporciones adecuadas.

$$qQ \equiv rR + gG + bB \quad (2.1)$$

La segunda, enuncia que la suma de dos colores cualesquiera es equivalente a la suma de los primarios de referencia que igualan los estímulos en cuestión.

$$q_1Q_1 \equiv r_1R + g_1G + b_1B \quad (2.2)$$

$$q_2Q_2 \equiv r_2R + g_2G + b_2B \quad (2.3)$$

Entonces:

$$q_3Q_3 \equiv q_1Q_1 + q_2Q_2 \text{ si } Q_3 \equiv (r_1 + r_2)R + (g_1 + g_2)G + (b_1 + b_2)B \quad (2.4)$$

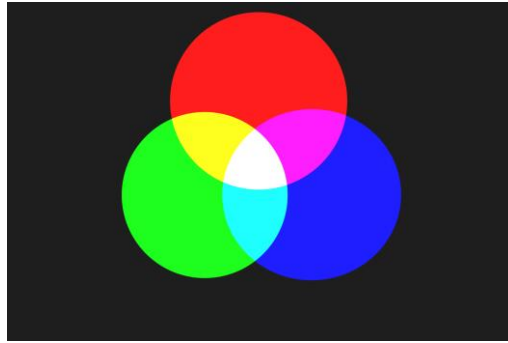


Figura 3. Mezcla aditiva de los tres colores primarios Rojo, Verde y Azul.

2.1.1.3 Iluminantes

En el tratamiento de imágenes hay que tener en cuenta que uno de los parámetros más importantes e influyentes a la hora de tratar una imagen es el iluminante. Esto es debido a que dependiendo del iluminante (condiciones de luminosidad con las que se toma una imagen), las características del color anteriormente descritas pueden variar. Así pues, es necesario definir unos estándares de iluminación para que, a la hora de realizar investigaciones, los resultados sean válidos. La CIE en [15] define una serie de distribuciones espectrales de energía radiante denominadas “Iluminantes Estándares CIE”. En la siguiente figura se muestran los iluminantes más populares así como la distribución espectral de su energía (Fig. 4).

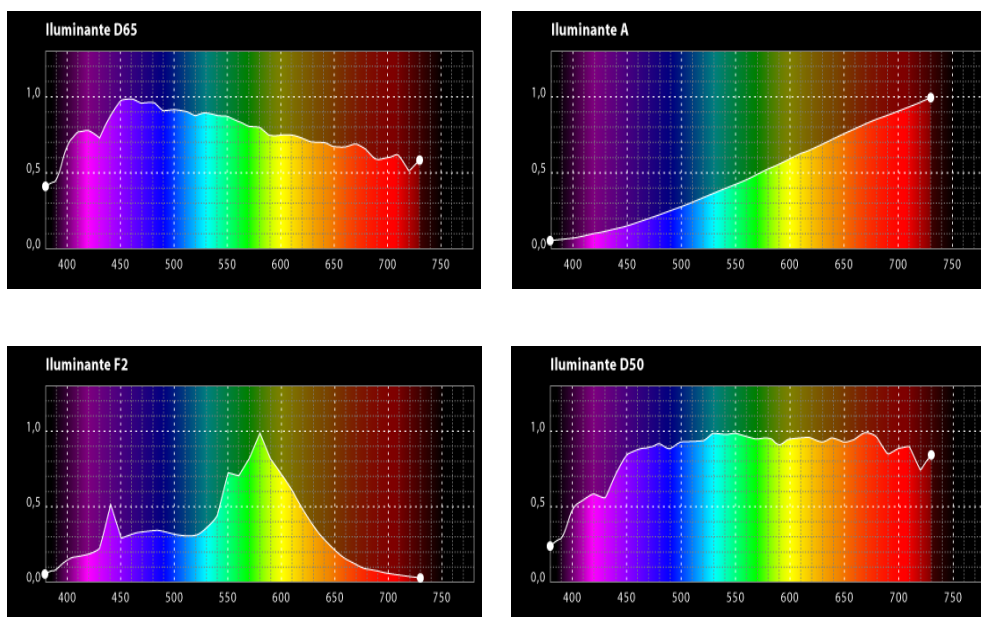


Figura 4. Arriba a la izq.: iluminante D65, luz de medio día. Arriba a la der.: iluminante A, bombilla incandescente. Abajo a la izq.: iluminante F2, Tubo fluorescente. Abajo a la der.: iluminante D50, luz de día.

2.1.2 Modelos de color

Según la Comisión Internacional de Iluminación (CIE), un espacio de color es una representación geométrica de colores en el espacio. Normalmente, este espacio tiene una base de 3 vectores, dichos vectores son los llamados *color matching functions*. La combinación lineal de las bases da lugar a todo el conjunto de colores que se pueden representar en el espacio de color.

Normalmente, los espacios de color tratan de agrupar la mayoría de colores posibles. Sin embargo, existen algunos modelos que sólo contienen un grupo de colores específicos.

Unos ejemplos de modelo de color pueden ser: escala de grises (una dimensión), RG (dos dimensiones), RGB (tres dimensiones)...

Nos centraremos en explicar los modelos de color que se han utilizado en el desarrollo del proyecto. En este caso, el modelo RGB y el modelo LAB.

2.1.2.1 Modelo RGB

El espacio de color RGB es el resultado de la representación de las componentes tricromáticas primarias de la luz (rojo, verde y azul), en un sistema de ejes cartesianos. Este modelo fue el primer espacio de color que se definió en la CIE. Mediante la combinación adecuada de las tres componentes primarias, se pueden obtener casi todos los colores.

Para el cálculo de las coordenadas de cromaticidad en el sistema RGB, se aplican las siguientes ecuaciones:

$$r = \frac{R}{R + G + B} \quad g = \frac{G}{R + G + B} \quad b = \frac{B}{R + G + B} \quad r + g + b = 1 \quad (2.5)$$

La combinación de los valores anteriores, da lugar a un espacio de color tridimensional (Fig. 5).

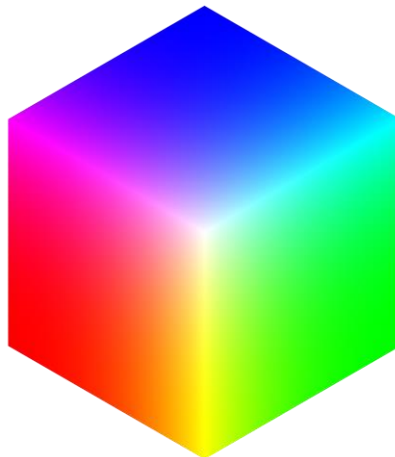


Figura 5. Representación Tridimensional del espacio RGB.

Sin embargo, este modelo de color tiene un problema fundamental. Algunos valores de las componentes cromáticas pueden tomar valores negativos en el eje de coordenadas para poder representar un color. Esto no quiere decir que haya que quitar esa cantidad de primario de la mezcla, sino que su valor tiene que ser negativo.

Para poder solucionar este problema de valores negativos, la CIE decidió adoptar otro espacio de color, XYZ.

2.1.2.2 Modelo CIELAB

Si la posibilidad de representar un color mediante la combinación de tres valores primarios es muy importante, no lo es menos la posibilidad de la existencia de un espacio de color que provoque una relación recíproca entre las diferencias de color percibidas y la distancia que separa las mezclas cromáticas en un espacio de color.

Esto da lugar a la creación de espacios de color uniformes, siendo estos espacios uno de los más utilizados.

En el año 1976 la CIE define el modelo $L^*a^*b^*$ (CIELAB). Es un espacio de color uniforme, donde los estímulos de color se identifican mediante las letras L^* , a^* , b^* . La definición de las componentes matemáticamente son las siguientes:

- Claridad (L^*), representa la cantidad de luz y depende del triestímulo. Puede tomar los valores de $[0,100]$. Esta coordenada se define igual que el sistema CIELUV:

$$L^* = 116 f\left(\frac{Y}{Y_n}\right) - 16 \quad \text{para } \frac{Y}{Y_n} > 0,008856 \quad (2.6)$$

Valores muy bajos del cociente Y/Y_n pueden dar lugar a claridades negativas; por ello se recomienda utilizar una modificación propuesta por Pauli (1976) para este tipo de casos.

$$L_m^* = 903,3 \left(\frac{Y}{Y_n}\right) \quad \text{para } \frac{Y}{Y_n} \leq 0,008856 \quad (2.7)$$

- Componentes de cromaticidad (a^*, b^*):

$$a^* = 500 \left[f\left(\frac{X}{X_n}\right) - f\left(\frac{Y}{Y_n}\right) \right] \quad (2.8)$$

$$b^* = 200 \left[f\left(\frac{Y}{Y_n}\right) - f\left(\frac{Z}{Z_n}\right) \right] \quad (2.9)$$

Donde:

$$f\left(\frac{X}{X_n}\right) = \sqrt[3]{\frac{X}{X_n}} \quad \text{para } \frac{X}{X_n} > 0,008856 \quad (2.10)$$

$$f\left(\frac{X}{X_n}\right) = 7,787 \frac{X}{X_n} + \frac{16}{116} \quad \text{para } \frac{X}{X_n} \leq 0,008856 \quad (2.11)$$

$$\left(\frac{Y}{Y_n}\right) = \sqrt[3]{\frac{Y}{Y_n}} \quad \text{para } \frac{Y}{Y_n} > 0,008856 \quad (2.12)$$

$$f\left(\frac{Y}{Y_n}\right) = 7,787 \frac{Y}{Y_n} + \frac{16}{116} \quad \text{para } \frac{Y}{Y_n} \leq 0,008856 \quad (2.13)$$

$$f\left(\frac{Z}{Z_n}\right) = \sqrt[3]{\frac{Z}{Z_n}} \quad \text{para } \frac{Z}{Z_n} > 0,008856 \quad (2.14)$$

$$f\left(\frac{Z}{Z_n}\right) = 7,787 \frac{Z}{Z_n} + \frac{16}{116} \quad \text{para } \frac{Z}{Z_n} \leq 0,008856 \quad (2.15)$$

Los valores X_n, Y_n, Z_n se corresponde con el estímulo considerado como blanco. Para el iluminante C y el observador estándar CIE 1931:

$$X_n = 98,041 \quad Y_n = 100 \quad Z_n = 118,103 \quad (2.16)$$

Gracias a ello, se consigue una normalización de los valores al relacionarlo con un blanco de referencia.

La diferencia de color del modelo CIELAB queda definida por:

$$\Delta E_{ab}^* = \sqrt{(\Delta L^*)^2 + (\Delta a^*)^2 + (\Delta b^*)^2} \quad (2.17)$$

Donde la diferencia queda calculada mediante la distancia euclídea entre dos puntos situados en un espacio tridimensional. Así, si dos puntos son coincidentes, la diferencia cromática entre los dos puntos (estímulos) es igual a cero.

El espacio CIELAB es uno de los más usados en el tratamiento de imágenes. No sólo por su característica de uniformidad sino porque abarca varios atributos perceptivos del color como son: cantidad de luz, saturación y tono.

Mediante una relación de las coordenadas de cromaticidad es muy fácil obtener los valores de tono (C) y saturación (H):

$$C_{ab}^* = \sqrt{a^{*2} + b^{*2}} \quad H_{ab}^* = \tan^{-1}\left(\frac{b^*}{a^*}\right) \quad (2.18)$$

La representación del espacio de color tridimensional es (Fig. 6):

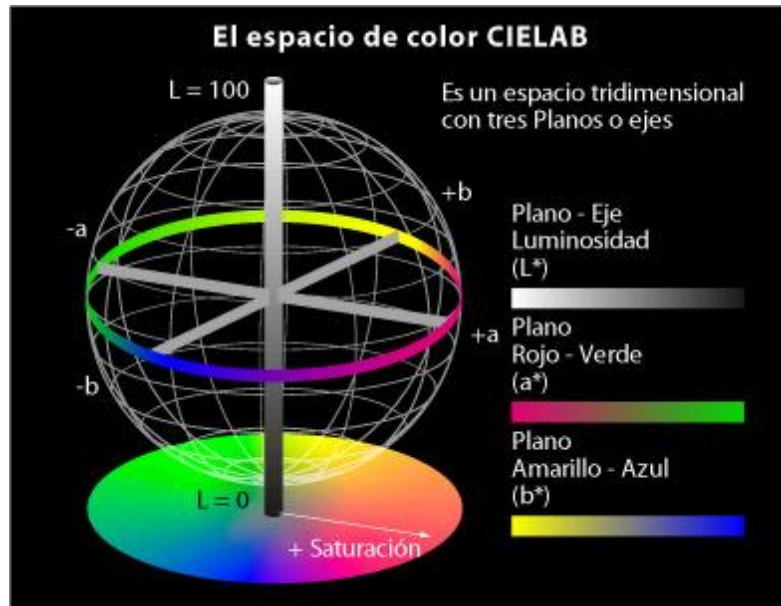


Figura 6. Representación Tridimensional del espacio de color $L^*a^*b^*$.

2.1.3 Distancias de color

En el apartado anterior hemos destacado que ciertos modelos como CIELAB y CIELUV se consideraban espacios de color uniformes. Sin embargo, a pesar del supuesto perceptual uniforme, los últimos estudios e investigaciones demuestran que esa teoría no es del todo cierta [16].

Tal y como se especificó en el apartado anterior, la diferencia de color de CIELAB se realiza mediante la distancia euclídea entre dos puntos. A pesar de ello, se descubrió que considerar esta distancia para describir la diferencia perceptual entre dos puntos no era del todo equiparable con el sistema visual humano.

Así pues, surgen fórmulas que permiten adaptar el espacio de color CIELAB aún mejor a la percepción del sistema visual humano.

Las distancias propuestas por la CIE son CIEDE2000 [17] y CIEDE94 [18], basadas en el espacio de color CIELAB. La más reciente de todas ellas publicada en 2001 es CIEDE2000, la recomendada por la CIE.

En nuestro trabajo, haremos una comparativa de las 3 distancias, la distancia por defecto de CIELAB (euclídea), la distancia CIEDE94 y la distancia CIEDE2000.

Para llegar a las distancias de color propuestas, es necesario acudir a las relaciones que explicamos en el apartado anterior, donde a partir de las coordenadas de croma a^* y b^* , se pueden obtener parámetros tales como tono y saturación.

1. Relación de los valores de CIELAB con los valores de CIEDE2000 y CIEDE94:

$$a' = (1 + G)a^* \quad \text{donde } G = 0,5 \left(1 - \sqrt{\frac{C_{ab}^{*7}}{C_{ab}^{*7} + 25^7}}\right) \quad (2.19)$$

$$b' = b^* \quad L' = L^*$$

2. Cálculo de C' (tono) y h' (saturación) :

$$C' = \sqrt{a'^2 + b'^2}$$

$$h' = \begin{cases} 0, & b' = a' = 0 \\ \tan^{-1}\left(\frac{b'}{a'}\right), & e. o. c \end{cases} \quad (2.20)$$

3. Cálculo de las componentes de diferencia de color entre las dos muestras:

$$\Delta L' = L'_1 - L'_2$$

$$\Delta C' = C'_1 - C'_2$$

$$\Delta h' = h'_1 - h'_2$$

$$\Delta H' = 2\sqrt{C'_1 C'_2} \sin\left(\frac{\Delta h'}{2}\right)$$

4. Cálculo de la fórmula de la CIEDE2000:

$$\Delta E_{00} = \sqrt{\left(\frac{\Delta L'}{K_L S_L}\right)^2 + \left(\frac{\Delta C'}{K_C S_C}\right)^2 + \left(\frac{\Delta H'}{K_H S_H}\right)^2 + R_T \left(\frac{\Delta C'}{K_C S_C}\right)^2 \left(\frac{\Delta H'}{K_H S_H}\right)^2} \quad (2.22)$$

5. Cálculo de la fórmula de la CIEDE94:

$$\Delta E_{94} = \sqrt{\left(\frac{\Delta L^*}{K_L S_L}\right)^2 + \left(\frac{\Delta C_{ab}^*}{K_C S_C}\right)^2 + \left(\frac{\Delta H_{ab}^*}{K_H S_H}\right)^2} \quad (2.23)$$

En la siguiente figura se muestra a modo de ejemplo, una imagen segmentada mediante el algoritmo de K-means con las tres distancias explicadas anteriormente (euclídea, CIEDE94 y CIEDE2000) y 4 etiquetas (Fig. 7).

¹ Los valores que no se han definido y que se encuentran en la fórmula, se pueden encontrar en la referencia [17]

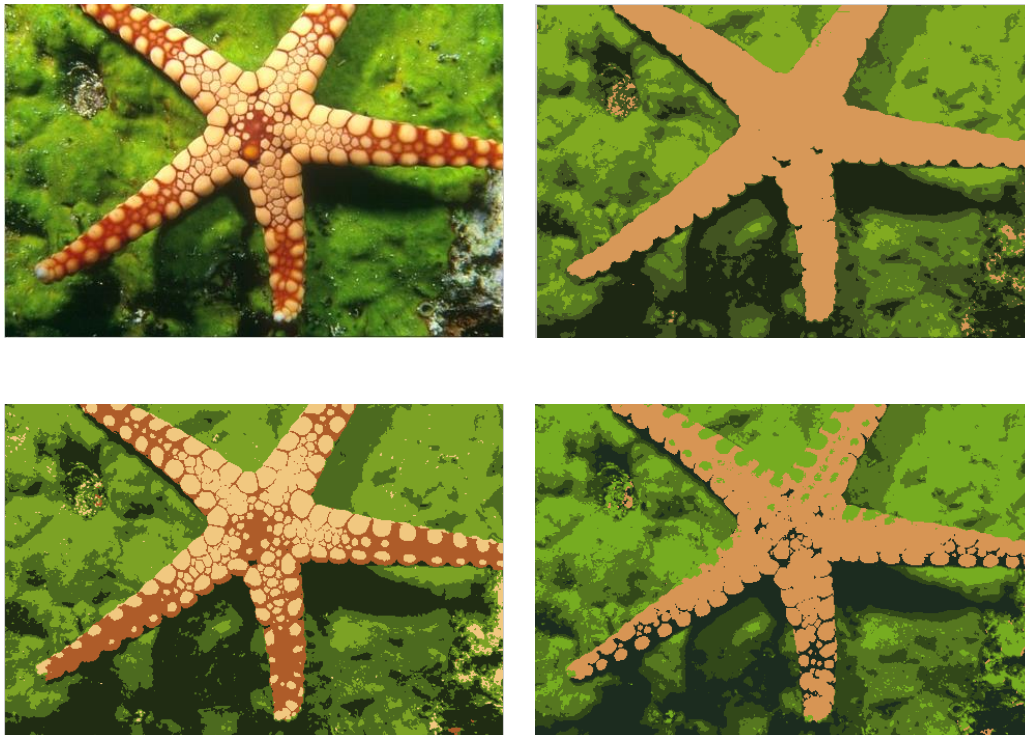


Figura 7. Arriba a la izq.: imagen original. Arriba a la der.: segmentación realizada mediante K-means con la distancia euclídea y 4 etiquetas. Abajo a la izq.: segmentación realizada mediante K-means con la distancia CIEDE94 y 4 etiquetas. Abajo a la der.: segmentación realizada mediante K-means con la distancia CIEDE2000 y 4 etiquetas.

2.1.4 Segmentación de imágenes en color

La segmentación de una imagen se define como la división o separación de dicha imagen en diferentes regiones con atributos similares. Dependiendo de qué atributo sea al que queremos dar importancia, ya que, según la necesidad queremos uno u otro, obtendremos una forma de segmentación diferente en cada caso mediante la utilización de diferentes algoritmos.

La segmentación de imágenes es un proceso clave en el análisis y comprensión de una imagen. Debido a ello, es un apartado muy importante siempre que, la caracterización y diferenciación sea primordial para el objeto de estudio (visión artificial, análisis médicos...).

Existen cinco categorías en la segmentación de imágenes: segmentación basada en pixel, segmentación basada en regiones, segmentación basada en bordes, segmentación híbrida entre región y bordes y, por último, segmentación basada en agrupamiento. Además, existen otras técnicas de segmentación muy utilizadas basadas en la optimización de una función de energía generada por la imagen.

En este apartado, trataré de ahondar en aquellos métodos de segmentación más relevantes para el desarrollo de este proyecto.

2.1.4.1 Segmentación basada en agrupamiento (*clustering*)

Los algoritmos de segmentación basados en *clustering* clasifican los datos de las imágenes y los agrupan en función de una serie de características anteriormente definidas.

Estos tipos de algoritmos son no supervisados, es decir, no requieren el conocimiento previo de aquellos objetos que se van a segmentar. Tan sólo requieren las características previas en base a las que van a realizar la segmentación. Por ello, resultan muy convenientes para aquellos ámbitos de aplicación en los que a priori, no se tiene mucha información, como en el caso de imágenes médicas o en el caso de segmentación de objetos

sobre un fondo.

Hay una gran variedad de algoritmos de este tipo como pueden ser: K-means, Fuzzy K-means, C-means...

Nosotros nos centraremos en el primero de ellos, ya que es el que hemos requerido. Los otros dos, son variantes del mismo.

2.1.4.1.1 Algoritmo de K-means

Como se ha explicado anteriormente, K-means es un algoritmo basado en *clustering*. Lo que quiere decir que su principal objetivo es optimizar la partición de la imagen en áreas conforme a unas características dadas. Este tipo de algoritmos ha sido usado comúnmente para una segmentación básica de imágenes ya que existen ciertas restricciones como la necesidad de que la imagen sea generalmente homogénea, sin texturas [6].

La restricción de texturas a la hora de usar K-means viene dado por dos problemas. El primero de ellos tiene que ver con la condición de comienzo, es decir, la inicialización de los *clusters* iniciales. El segundo de ellos tiene que ver con el hecho de que no se aplica una cohesión espacial a lo largo del proceso de segmentación.

Antes de comenzar a aplicar el algoritmo es necesario realizar una transformación en el espacio de color que se vaya a utilizar. Es necesario crear uno independiente del dispositivo en el que se realice, con el fin de evitar diferencias de color. Uno de los más usuales suele ser el espacio de color RGB.

El algoritmo K-means que se muestra más adelante es el algoritmo aplicado a imágenes en escala de grises. Para el algoritmo aplicado a imágenes en color se extrapolan esas ecuaciones a los planos que contiene el espacio de color correspondiente. Así pues, si utilizamos el espacio de color RGB, los vectores contendrían tres valores en lugar de uno como en el caso de imágenes en escala de grises.

Según [7], el algoritmo de K-means básico consiste en cuatro pasos, que se describen de manera resumida a continuación.

1. Inicialización de (K) centros. Se generan un número de *clusters* de manera aleatoria en función de las condiciones iniciales dadas:

$$Center_i^0 = GL_{min} + \left(i - \frac{1}{2}\right) \frac{GL_{max} - GL_{min}}{K} \quad (2.24)$$

$$i = 1, 2, \dots, K,$$

Donde $Center_i^0$ es el centroide inicial de la i ésima clase, GL_{max} y GL_{min} son el máximo y el mínimo valor de gris (GL) en el espacio muestra respectivamente.

2. Generación de nuevas particiones asignando cada punto a su centro más cercano. El criterio de asignación de un punto a un centro está basado en la distancia euclídea. Actualmente existen nuevas distancias como CIEDE94 y CIEDE2000. Continuando la notación matemática con el uso de la distancia euclídea:

$$Distance_{i,j} = abs(GL_j - Center_i)$$

$$i = 1, 2, \dots, K; j = 1, 2 \dots N. \quad (2.25)$$

Donde $Distance_{i,j}$ es la distancia desde el punto j al centroide de la clase i . N es el número total de puntos pertenecientes al espacio muestra.

3. Cálculo de los (K) nuevos centros. Se calculan conforme a la media de los puntos que han sido

asignados al centro anterior según el paso anterior:

$$Center_i^m = \frac{1}{N_i} \sum_{j=1}^{N_i} GL_j, \quad i = 1, 2, \dots, K \quad (2.26)$$

Donde N_i es el número total de puntos asignados a la clase i ésima en el paso número 2.

4. Se repite el paso número 2 si alguno de los centros asignados cambia del paso 2 al 3. Si no, se acaba el algoritmo.

En la siguiente figura se muestra una versión básica del agrupamiento entre muestras. En este caso, el ejemplo se ha realizado con dos etiquetas (Fig. 8)

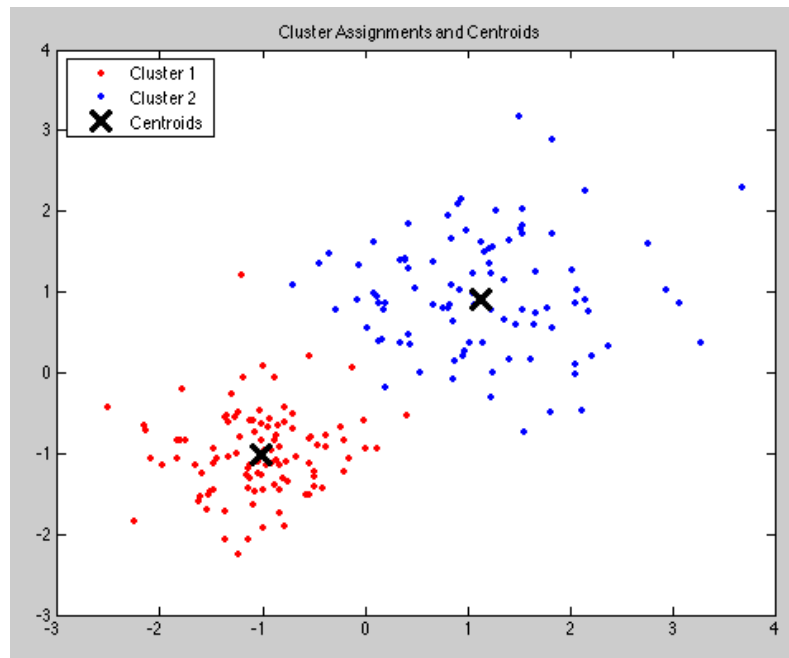


Figura 8. Agrupamiento de muestras aleatorias según dos etiquetas.

El algoritmo de K-means no es sólo aplicable a imágenes, sino a matrices en general. Además, como se ha mencionado anteriormente, se trata de un algoritmo de aprendizaje no supervisado en el que el valor de cada uno de los píxeles que componen la imagen es tomado como un vector y sus componentes son tratadas sin tener en cuenta a qué hacen referencia. Es por ello que, el algoritmo no requiere de modificaciones en caso de usarse distintos espacios de color, tomando más o menos planos. Los elementos del vector pueden ser valores de luminancia, tono o saturación y la eficacia de ellos dependerá de la finalidad que tenga la segmentación.

En nuestro caso en concreto, el algoritmo de K-means se aplicará sobre el espacio de color L^*a^*b (véase 2.1.2.2) con las distancias euclídea, CIEDE94 y CIEDE00 (véase 2.1.3).

2.1.4.2 Segmentación basada en la optimización de una función de energía

Como se ha comentado en la introducción de esta sección, existen muchos métodos para la segmentación de imágenes. Un grupo muy importante de éstos, consiste en la optimización de la función de energía.

Existen dos grupos de este tipo de segmentación. Por un lado, tenemos aquellas funciones definidas en un espacio continuo donde la función a optimizar da lugar a una ecuación diferencial donde para resolverla se

suelen utilizar métodos como el descenso del gradiente. En segundo lugar, tenemos funciones que se definen en un conjunto discreto donde la segmentación de éste da lugar a la resolución de un problema de minimización de un espacio finito.

Nuestro trabajo en concreto se basa en la utilización de un algoritmo novedoso basado en Max-Flow en el espacio continuo.

Experimentos en la segmentación de imagen, tanto supervisado como sin supervisión, muestran que el algoritmo que utilizamos en el proyecto basado en Continuous Max-Flow, da lugar a unos mejores resultados en términos de eficiencia y rendimiento que anteriores aproximaciones en el mismo campo [12].

En primer lugar, se hará una breve introducción de los conceptos básicos de Max-Flow en el espacio discreto ya que muchos de los conceptos utilizados en éste son muy parecidos en el espacio continuo. Aparte, su explicación es mucho más fácil de entender y su visión conceptual también. Continuaremos explicando los conceptos claves y relaciones de lo explicado anteriormente con el algoritmo de Max-Flow en un espacio continuo. Por último, introduciremos el algoritmo que se utiliza en este proyecto, una variante de Max-Flow en espacio continuo, y la relación que existe con el algoritmo basado en clústering (K-means).

Apoyándonos en los artículos [10] [12] y [24], seremos capaces de comprender y explicar los algoritmos de Min-Cut y Max-Flow tanto en tiempo discreto como en continuo. Además de definir el algoritmo que utilizamos, ya que [10] y [24] son artículos escritos conforme a la creación del nuevo algoritmo de Max-Flow.

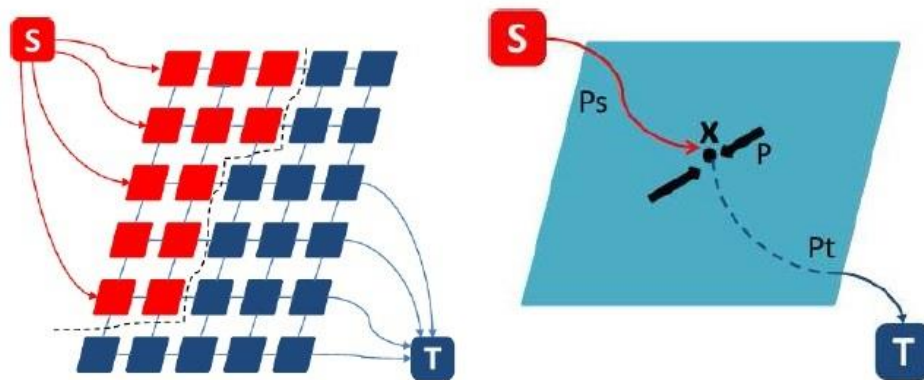


Figura 9. Representación del algoritmo de Max-Flow en tiempo discreto (izq.) y tiempo continuo (der.).

2.1.4.2.1 Algoritmo de Min-Cut / Max-Flow espacio discreto

En relación a lo expuesto anteriormente, Greig, Porteous y Seheult [8], descubrieron que mediante algoritmos de Min-Cut /Max-Flow utilizados en optimización de combinatorias, se podían usar para minimizar una gran cantidad importante de funciones de energía en problemas de visión. En ese artículo en concreto, demostraron que el corte de grafos se podía usar para la restauración de imágenes binarias. Futuros trabajos de investigación estaban centrados en la utilidad de los grafos para la segmentación de imágenes, restauración...

Sea un grafo unidireccional $G = \{V, E\}$ definido por un conjunto de nodos (vértices V), con dos nodos especiales, uno llamado fuente, s y otro llamado sumidero, t , y un conjunto de bordes unidireccionales que conectan esos nodos (E). El conjunto de bordes está compuesto por dos tipos de bordes: bordes espaciales $e_n = (r, q)$, donde $r, q \in V \setminus \{s, t\}$, que conectan entre sí los nodos r y q a excepción de la fuente y el sumidero, s y t respectivamente; los bordes terminales $e_s = (s, r)$ o $e_t = (r, t)$, donde $r \in V \setminus \{s, t\}$ conecta el nodo s o t a otro nodo. Cada borde $e \in E$, tiene asignado un coste no negativo, $C(e) \geq 0$.

Min-Cut

Definimos como corte a un subgrupo tal que la fuente y el sumidero quedan separados $C \subset E$, también llamado s-cut (Fig. 9 izquierda). Por lo tanto, la imagen queda claramente dividida en dos partes, una relacionada con la fuente, S y otra con el sumidero, T .

$$V = V_s \cup V_t, \quad V_s \cap V_t = \emptyset \quad (2.27)$$

Para cada corte, la energía es definida como la suma de los costes $C(e)$ de cada borde $e \in E_{st} \subset E$, cuyos puntos finales pertenecen a las dos particiones. Por lo tanto, el problema de Min-Cut es encontrar las dos particiones de los vértices tal que el correspondiente corte de energía sea mínimo:

$$\min_{E_{st} \subset E} \sum_{e \in E_{st}} C(e) \quad (2.28)$$

Max-Flow

Por otro lado, cada borde $e \in E$ se puede ver como una tubería y cada coste $C(e)$ puede ser considerado como la capacidad de esa tubería, para la cual el máximo flujo está permitido. Para cada red de “tuberías” tenemos las siguientes restricciones de flujos:

1. Capacidad de los flujos espaciales p : para los bordes unidireccionales espaciales $e_n = (r, q) \in E, r, q \in V\{s, t\}$, los flujos espaciales $p(e_n)$ vienen restringidos por:

$$|p(e_n)| \leq C(e_n) \quad (2.29)$$

2. Capacidad de flujo de la fuente p_s : para el borde $e_s(v): s \rightarrow v$ que conecta la fuente s a un nodo $v \in V\{s, t\}$, el flujo de la fuente $p_s(v)$ es directo desde s a v . Su capacidad $C_s(v)$ indica que:

$$0 \leq p_s(v) \leq C_s(v) \quad (2.30)$$

3. Capacidad de flujo del sumidero p_t : para el borde $e_t(v): v \rightarrow t$ que conecta un nodo $v \in V\{s, t\}$ al sumidero t , $p_t(v)$ es directo desde v a t . Su capacidad $C_t(v)$ indica que:

$$0 \leq p_t(v) \leq C_t(v) \quad (2.31)$$

4. Conservación de los flujos: Para nodo $v \in V\{s, t\}$, los flujos entrantes deben estar equilibrados con los flujos salientes. En otras palabras, todos los flujos que pasan a través de v , incluyendo los flujos espaciales, los flujos de la fuente y los flujos del sumidero vienen restringidos por:

$$\left(\sum_{q \in N(v)} p((q, v)) \right) - p_s(v) + p_t(v) = 0 \quad (2.32)$$

Donde $q \in N(v)$ es el conjunto de nodos vecinos de v .

En este caso, el problema de máximo flujo se basa en encontrar la mayor cantidad de flujo permitido para pasar de la fuente s al sumidero t , en otras palabras:

$$\max_{p_s} \sum_{v \in V\{s,t\}} p_s(v) \quad (2.33)$$

Sujeto a las restricciones (2.29) (2.30) (2.31) y (2.32)

Relación Min-Cut/Max-Flow

Es bien sabido que el problema de Max-Flow (2.33) es equivalente al problema de Min-Cut (2.28), donde los flujos están saturados uniformemente en los cortes de los bordes, en otras palabras, el flujo total está limitado por “las tuberías saturadas”. Por la terminología de graph-cut, cuando un flujo $p(e)$ en el borde $e \in E$ alcanza su correspondiente capacidad $C(e)$ dada por (2.29), (2.30), (2.31), diremos que está saturado; en otro caso, no saturado.

2.1.4.2.2 Algoritmo de Min-Cut / Max-Flow espacio continuo

Min-Cut / Max-Flow con 2 etiquetas

Antes de introducir el modelo de Max-Flow en el espacio continuo con n etiquetas, introduciremos el reciente estudio de Max-Flow en el espacio continuo con 2 etiquetas propuesto por los autores [12] el cual es dual a Min-Cut en el espacio continuo.

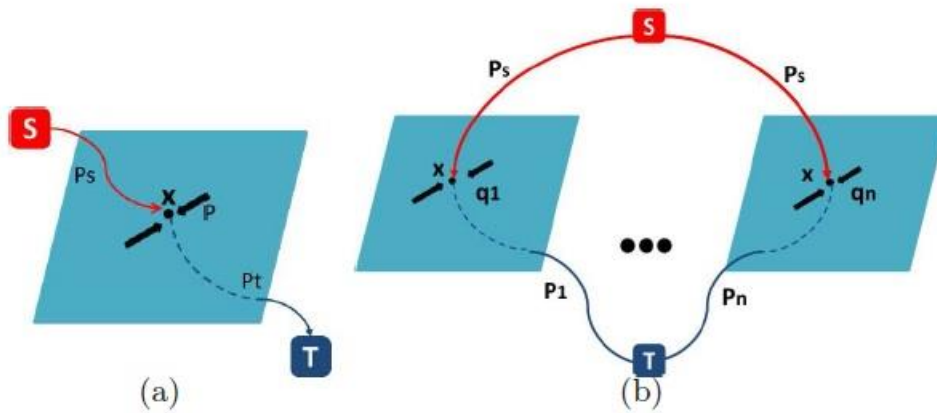


Figura 10. Representación de Max-Flow en tiempo Continuo para 2 etiquetas (izq.) y n etiquetas (der.).

Análogamente a Max-Flow y Min-Cut basado en grafos (Sec 2.1.4.2.1): dado el dominio temporal de la imagen Ω , asumimos que hay dos terminales, la fuente s y el sumidero t (Fig. 10 (a)). Además, asumimos que para cada posición de la imagen $x \in \Omega$ intervienen tres tipos de flujos: el flujo de la fuente $p_s(x) \in \mathbb{R}$ directamente desde la fuente s a x , el flujo del sumidero $p_t(x) \in \mathbb{R}$ directamente desde x al sumidero t y el flujo espacial $p(x) \in \mathbb{R}^2$. Los tres flujos tienen una serie de restricciones:

$$\begin{aligned} |p_s(x)| &\leq C_s(x), & \forall x \in \Omega; \\ |p_t(x)| &\leq C_t(x), & \forall x \in \Omega; \\ |p(x)| &\leq C(x), & \forall x \in \Omega; \end{aligned} \quad (2.34)$$

Además, todos los flujos se conservan para $\forall x \in \Omega$:

$$p_t(x) - p_s(x) + \text{div } p(x) = 0, \quad \forall x \in \Omega; \quad (2.35)$$

Donde $\text{div } p$ evalúa la cantidad total de flujo espacial entrante localmente alrededor de x , lo cual es una analogía al operador suma (2.32) para el tiempo discreto.

En este caso, las restricciones del flujo de la fuente $p_s(x)$ (2.34) y el flujo del sumidero $p_t(x)$ (2.34) son diferentes en comparación con las restricciones de Max-Flow en el dominio temporal discreto (2.30) (2.31). Esto es debido a que la positividad de los flujos $p_s(x)$ y $p_t(x)$ no son necesarias puesto que son flujos directos y su valor indica cómo el flujo se distribuye desde s hasta el punto x o desde x hasta t . De la misma manera, $C_s(x)$ y $C_t(x)$ tampoco es necesario que sean positivos. Por lo tanto, esto extiende la aplicación de Max-Flow y Min-Cut en el dominio temporal continuo.

Análogamente con la formulación del problema de Max-Flow en tiempo discreto (2.33), podemos formular el correspondiente problema de Max-Flow en tiempo continuo mediante la maximización del flujo total desde la fuente:

$$\max_{p_s, p_t, p} \int_{\Omega} p_s \, dx \quad (2.36)$$

Sujeto a las restricciones (2.34) y (2.35)

Además, Yuan y otros [12] probaron que la formulación del problema de máximo flujo en el dominio continuo es equivalente al problema de Min-Cut de la siguiente manera:

$$\min_{u(x) \in [0,1]} \int_{\Omega} (1-u)C_s \, dx + \int_{\Omega} uC_t \, dx + \int_{\Omega} C|\nabla u| \, dx \quad (2.37)$$

De hecho, la ecuación (2.37) tan sólo nos da el modelo dual de (2.36) y la función de etiquetado $u(x)$ es el multiplicador de la condición de conservación de flujo (2.35). Por lo tanto, un modelo eficiente y fiable puede ser construido en base al algoritmo (2.36).

Min-Cut / Max-Flow con n etiquetas

Basándose en las observaciones de lo comentado anteriormente, Yuan et al establecieron una configuración para un modelo de Max-Flow en espacio continuo con n etiquetas [24] (Fig. 10 (b)).

1. Tenemos n copias de Ω_i , $i = 1 \dots n$, del dominio de la imagen Ω ;
2. Para cada posición $x \in \Omega$, el flujo de la fuente $p_s(x)$ intenta ser enviado desde la fuente s hasta x para cada copia Ω_i , $i = 1 \dots n$ de Ω . El flujo de la fuente es el mismo para cada Ω_i , $i = 1 \dots n$, en otras palabras, $p_s(x)$ es único.
3. Para cada posición $x \in \Omega$, el flujo del sumidero $p_i(x)$, $i = 1 \dots n$ es enviado de forma directa desde la posición x en la i -ésima copia Ω_i hasta el sumidero t . Los n flujos $p_i(x)$, $i = 1 \dots n$, pueden ser diferentes;
4. Los flujos espaciales $q_i(x)$, $i = 1 \dots n$ vienen definidos dentro de cada copia Ω_i , $i = 1 \dots n$. Pueden ser diferentes el uno del otro.

Para una configuración dada, establecemos las condiciones de restricción para los flujos $p_i(x)$ y $q_i(x)$, donde $x \in \Omega$, de la siguiente manera:

$$\begin{aligned} |q_i(x)| &\leq C_i(x), \quad i = 1 \dots n; \\ p_i(x) &\leq \rho(l_i, x), \quad i = 1 \dots n; \\ p_i(x) - p_s(x) + \operatorname{div} q_i(x) &= 0, \quad i = 1 \dots n; \end{aligned} \quad (2.38)$$

La función $\rho(l_i, x)$, $i = 1 \dots n$, evalúa la actuación de asignar la etiqueta l_i a una posición específica x .

Por lo tanto, se puede formular el modelo de Max-Flow en tiempo continuo de la siguiente manera:

$$\max_{p_s, p, q} \{P(p_s, p, q) := \int_{\Omega} p_s dx\} \quad (2.39)$$

Sujeto a las restricciones (2.38)

2.1.4.2.3 Algoritmo de Min-Cut / Max-Flow dirigido

En los dos apartados anteriores, hemos introducido teóricamente el algoritmo de Max-Flow y su equivalente Min-Cut tanto en un espacio discreto como continuo. Como ya dije anteriormente, en nuestro proyecto, el algoritmo utilizado se basa en el algoritmo basado en un espacio continuo. Sin embargo, existen algunas pequeñas diferencias entre éste y el utilizado realmente.

En comparación con el algoritmo de Max-Flow y Min-Cut en espacio continuo, el algoritmo aquí propuesto calcula una partición óptima sujeta a unas restricciones dadas de las configuraciones de las regiones, en otras palabras, algunos píxeles de la imagen son etiquetados previamente. Es aquí donde se establece la conjunción del algoritmo de K-means, con el algoritmo de Max-Flow en tiempo continuo.

Según J.Yuan y colaboradores [12], el problema de particionado dirigido de las imágenes puede ser modelado según el siguiente problema de Min-Cut dirigido:

$$\min_{\{\Omega_i\}_{i=1}^n} \sum_i^n \int_{\Omega_i} C_i(x) dx + \alpha \sum_i^n |\partial\Omega_i| \quad (2.40)$$

Sujeto a:

$$\bigcup_{i=1}^n \Omega_i = \Omega; \quad \Omega_k \cap \Omega_l = \emptyset, \quad \forall k \neq l \quad (2.41)$$

Donde $|\partial\Omega_i|$, mide el perímetro de cada subdominio disjunto Ω_i , $i = 1 \dots n$. La función $C_i(x)$, $i = 1 \dots n$, evalúa los costes de asignar a una posición específica $x \in \Omega$ a la región Ω_i .

Además, según propusieron J.Yuan et al [12] y como ya se ha comentado en el apartado anterior, el problema de Max-Flow se puede modelar según:

$$\max_{p_s, p, q} \{P(p_s, p, q) := \int_{\Omega} p_s dx\} \quad (2.42)$$

Sujeto a las mismas restricciones (2.38)

Observamos entonces que resolver el problema de Max-Flow dirigido es exactamente igual a resolver el problema de Max-Flow en espacio continuo, ya que, la ecuación de Max-Flow es exactamente la misma y por lo tanto, su equivalente dual, Min-Cut también lo es.

2.2 Bases de datos de imágenes

Este apartado trata de mostrar la importancia de las bases de datos de imágenes para la realización de proyectos y estudios de investigación sobre un determinado campo.

Las bases de datos no son más que una herramienta donde se encuentran numerosas imágenes que sirven de apoyo y verificación. En mi caso, este trabajo se ha apoyado en una de las mejores bases de datos de imágenes segmentadas, la base de datos de la universidad de Berkeley [13].

Existen muchos trabajos de investigación que han hecho referencia a esta base de datos y se han apoyado de las imágenes que allí se encuentran para el uso del trabajo de investigación [14].

En ella, se pueden encontrar numerosas imágenes de todo tipo y lo que es de mayor interés para este trabajo, imágenes segmentadas por una media de cinco sujetos cada una, lo que se denomina *ground truth*.

Es por todo ello, que en el desarrollo de este trabajo se utilizarán algunas de las imágenes recogidas en esta B.D.

Además de esta base de datos, se usará una pequeña bases de datos de imágenes de test que me han sido proporcionadas por mi tutor del trabajo y que ha sido elaborada por una profesora del departamento de la escuela.

2.3 Imágenes utilizadas y formas de evaluación

En esta sección introduciremos las imágenes utilizadas y las formas de evaluación que se han usado en el desarrollo de este trabajo. En primer lugar, comentaremos las imágenes que se han utilizado en la realización de las pruebas para definir la actuación de los algoritmos en la segmentación de imágenes. En segundo lugar, explicaremos cuáles han sido los métodos de evaluación que se han utilizado para la consecución de resultados numéricos a través de una imagen.

2.3.1 Imágenes utilizadas

Para la realización de este trabajo se han utilizados dos series de imágenes. La primera de ellas, consiste en una serie de imágenes de test, las cuales han servido para comprobar si los algoritmos que se estaban utilizando y tal y como se habían programado funcionaban correctamente, así como, si los algoritmos utilizados para la obtención de valores numéricos de evaluación eran correctos o no. En segundo lugar, y tal y como se comentó en puntos anteriores de este trabajo, se han utilizados imágenes reales obtenidas de la base de datos de Berkeley, para mostrar valores en imágenes reales y además, poder comparar los resultados numéricos que se obtienen con los resultados numéricos que se encuentran en esta base de datos obtenidos a través de otro tipo de algoritmos.

2.3.1.1 Imágenes test

Estas imágenes han sido aportadas por mi tutor de trabajo de fin de grado. Consiste en una serie de imágenes de rectángulos en los que en todas ellas existen 4 rectángulos. En algunas imágenes, a simple vista somos capaces de identificar esos 4 rectángulos. Sin embargo, en otras muchas la percepción visual no es suficiente y apenas diferenciamos 3. Así pues, estas imágenes han servido como test para comprobar dos cosas: la primera

de ellas ha sido la capacidad del algoritmo de K-means y las diferentes distancias para ser capaz de identificar esos 4 rectángulos. Comprobaremos más adelante en el apartado de resultados numéricos si la actuación de las diferentes distancias manifiestan cambios significativos o no. La segunda cosa que se puede comprobar es si el algoritmo de Continuous Max-Flow ha sido capaz de segmentar correctamente las imágenes tal y como lo haría K-means o incluso mejorarlas.

Un ejemplo de este tipo de imágenes es el siguiente (Fig. 11):



Figura 11. Imágenes utilizadas como test, squares.

En la figura 11 se muestran dos imágenes utilizadas para lo manifestado anteriormente. A la izquierda los 4 rectángulos se pueden apreciar fácilmente. Sin embargo, para poder distinguir los 4 rectángulos de la figura de la derecha hay que tener una visión muy aguda.

Se han utilizado un total de cuarenta y siete (47) imágenes de este tipo. A partir de ahora, a este tipo de imágenes las llamaremos “squares”.

2.3.1.2 Imágenes reales

Aparte de las imágenes utilizadas como test como se ha comentado en el apartado anterior, se han utilizado un total de 8 imágenes reales de la base de Berkeley. Las imágenes de esta base de datos son las mejores en cuanto a imágenes referenciadas en otros artículos. Además, en esta base de datos, no sólo se cuentan con las imágenes tal cual, sino que también cuenta con el *ground truth* utilizado posteriormente para los algoritmos de evaluación.

Sin embargo, a pesar de que esta base de datos cuenta con casi quinientas (500) imágenes, estas imágenes están preparadas para que su análisis sea a través de bordes y no a través de regiones de color. Por ello, ha sido muy complicado (y lo es) encontrar imágenes concretas para este tipo de análisis. A pesar de ello, he tratado de coger las ocho (8) imágenes en las que se diferenciaba más el color y en la que el *ground truth* hacía referencia mayoritariamente a regiones en lugar de a todos los bordes de la imagen.

Un ejemplo de este tipo de imágenes es el siguiente (Fig. 12):



Figura 12. Imágenes utilizadas como test, reales.

En la figura 12 se muestran dos tipos de imágenes usadas para ver la actuación de los algoritmos sobre imágenes reales. En este trabajo, haremos referencia a este tipo de imágenes como “reales”.

2.3.2 Formas de evaluación

En la evaluación de las imágenes, tanto las imágenes de squares como las imágenes reales, se han considerado dos metodologías diferentes para la obtención de valores numéricos comparables. Hay que saber que para poder obtener una serie de valores numéricos es necesaria una imagen de referencia con la que comparar la resultante, a la que llamaremos *ground truth*. Esta imagen no es más que el resultado óptimo que se tendría que obtener en caso de que el algoritmo fuese totalmente efectivo. Dependiendo de las imágenes que hemos utilizado (squares o reales) y de la metodología aplicada (bordes o regiones) este *ground truth* cambia.

La primera metodología que hemos aplicado se basa en la comparación de los bordes de las regiones. La segunda de ellas se basa en el interior de las regiones. En los subapartados siguientes se explicará de forma más clara.

En la evaluación de nuestras imágenes, tanto squares como reales, no sólo se han evaluado las imágenes obtenidas con la conjunción del algoritmo de Continuous Max-Flow con K-means, sino que además también se han evaluado las imágenes obtenidas con tan sólo el algoritmo de K-means para ver si realmente se producía alguna mejora.

2.3.2.1 Bordes

La evaluación de los bordes de la imagen es un método que ha consistido en comparar una imagen de bordes óptima llamada *ground truth*, con la imagen de bordes obtenida de la imagen que ha sido segmentada por nuestros algoritmos. Una vez se tienen las dos imágenes, éstas se comparan y se obtiene un resultado numérico. El proceso de evaluación de este tipo ha seguido una serie de pasos que dependiendo de si la imagen era squares o real ha tenido ciertas variantes. A continuación, se mostrarán esas dos fases que se han seguido para obtener los valores.

2.3.2.1.1 Obtención de las imágenes de *ground truth*

Como se ha comentado en los subapartados anteriores, lo primero que se debe considerar es la obtención de un *ground truth*. Una imagen de bordes que sepamos que es 100 % óptima.

Squares

Este tipo de imágenes ya mostradas anteriormente, son imágenes formadas por 4 rectángulos de diferentes tamaños y color. Por lo tanto, sabemos que tienen 4 etiquetas y que su forma es uniforme, es decir, siempre van a ser cuadrados.

Aprovechándonos de esta circunstancia, elaboramos un algoritmo para la detección de los bordes de los cuadrados. Partiendo de la imagen real, se le aplica un algoritmo que sea capaz de detectar los bordes de esa imagen. Así, el resultante será un *ground truth* de la imagen real, es decir, es la que estaremos buscando como modelo y por tanto será la imagen óptima (Fig. 13)



Figura 13. A la izq. la imagen de squares. A la der. Su imagen de bordes óptima obtenida con nuestro algoritmo.

Reales

En comparación con las imágenes anteriores, este tipo de imágenes no son uniformes y, a priori, el número de etiquetas está definido por nosotros pero puede variar para obtener mejores resultados. Sin embargo, como ya dijimos, estas imágenes están sacadas de la base de datos de imágenes de Berkeley. Además de tener las imágenes reales como tal, estas imágenes vienen acompañadas de un *ground truth*. En cada imagen, existen 5 *ground truth* diferentes realizado por sujetos distintos. Por lo tanto, tan sólo es necesario convertir ese *ground truth* que está en formato matriz al formato de una imagen. Gracias al programa Matlab, este cometido es muy sencillo y el resultado es que para cada imagen real contamos un cinco *ground truth* diferentes. La utilización de uno u otro dependerá de cuál se ajuste más a nuestra división por colores en lugar de por regiones (Fig. 14).



Figura 14. Representación de la imagen real con sus cinco *ground truth*.

2.3.2.1.2 Cálculo de *Recall* y *Precision*

Una vez tenemos todas las imágenes de *ground truth* necesarias, se procede a la obtención de otro factor necesario para poder realizar la comparación, las imágenes de bordes de las imágenes obtenidas mediante los algoritmos de K-means y K-means+Continuous Max-Flow.

Para las imágenes squares, a partir de los resultados de las imágenes segmentadas mediante los algoritmos, nos ayudamos del mismo algoritmo que se ha utilizado para la obtención de las imágenes de bordes de las imágenes originales de squares.

Sin embargo, para las imágenes reales al no ser uniforme este método no es aplicable. Es por ello que para conseguir las imágenes de bordes a partir de las imágenes segmentadas mediante los algoritmos, nos ayudamos de un algoritmo de detección de bordes de imágenes que se encuentra en matlab como es “edge” con el método “Canny” ya que es el que mejor resultados reporta.

Así pues, cada vez que se realiza la segmentación de una imagen, bien sea por el método de K-means o por el método de Max-Flow, es necesario utilizar el algoritmo de detección de bordes comentado anteriormente para la obtención de las imágenes de bordes y así poder obtener resultados numéricos.

En la siguiente figura se pueden ver dos imágenes de bordes. La primera de ellas es el resultado de la segmentación mediante el método de K-means y la posterior utilización del algoritmo “edge” de matlab. La segunda imagen es el resultado de la aplicación del algoritmo “edge” para la detección de bordes a una imagen segmentada mediante el algoritmo de Continuous Max-Flow con K-means (Fig. 15).



Figura 15. A la izq. la imagen de bordes obtenida de una imagen tipo squares. A la der. Una imagen de bordes obtenida de una imagen tipo real.

Ahora que ya tenemos todas las imágenes de bordes necesarias, se puede proceder a la comparación y evaluación de las imágenes de *ground truth* con las obtenidas.

En la Fig. 16 se muestran tres imágenes. La imagen arriba a la izquierda es la imagen utilizada como referencia. Esta imagen se obtiene de uno de los observadores de la base de datos de Berkeley. La imagen de arriba a la derecha es la imagen de bordes como resultado de aplicar el algoritmo “edge” a una imagen previamente segmentada mediante K-means de color con 3 etiquetas. La imagen que se encuentra en la parte de abajo es la imagen diferencia de la imagen de arriba a la izquierda menos la imagen de arriba a la derecha. El resultado, si las dos imágenes fuesen exactamente iguales, sería una imagen completamente en negro. Sin embargo, esta imagen contiene píxeles en blanco, los cuales se corresponden a las zonas de no similitud entre ambas.

Para la evaluación se consideran dos parámetros llamados *Recall* y *Precision* y un valor que compara ambos entre sí llamado *F*. Son valores muy frecuentemente utilizados en el campo de la segmentación de imagen a la hora de realizar una comparación con imágenes de bordes [13,14].

Precision

Para el cálculo de *precision* se realiza la siguiente operación:

$$R = \frac{\text{Número de píxeles correctos}^2}{\text{Número total de píxeles de bordes de la imagen segmentada}} \quad (2.43)$$

Recall

Para el cálculo de *recall* se realiza la siguiente operación:

$$P = \frac{\text{Número de píxeles correctos}}{\text{Número total de píxeles de bordes de la imagen de ground truth}} \quad (2.44)$$

² Se consideran píxeles correctos aquellos cuyo valor coincidan en la imagen segmentada y en la imagen de referencia (ground truth).

F

F es un parámetro que se utiliza para aunar los conceptos anteriores en uno solo. La relación es la siguiente:

$$F = \frac{2PR}{P + R} \quad (2.45)$$

Es más importante tener en cuenta el valor que relaciona a ambos, *F*, que separados entre sí. Esto que debido a que por ejemplo, si tenemos una sobre segmentación, el *recall* será muy alto pero en cambio *precision* será muy bajo. Esta relación sólo se vería reflejada en el valor de *F*.

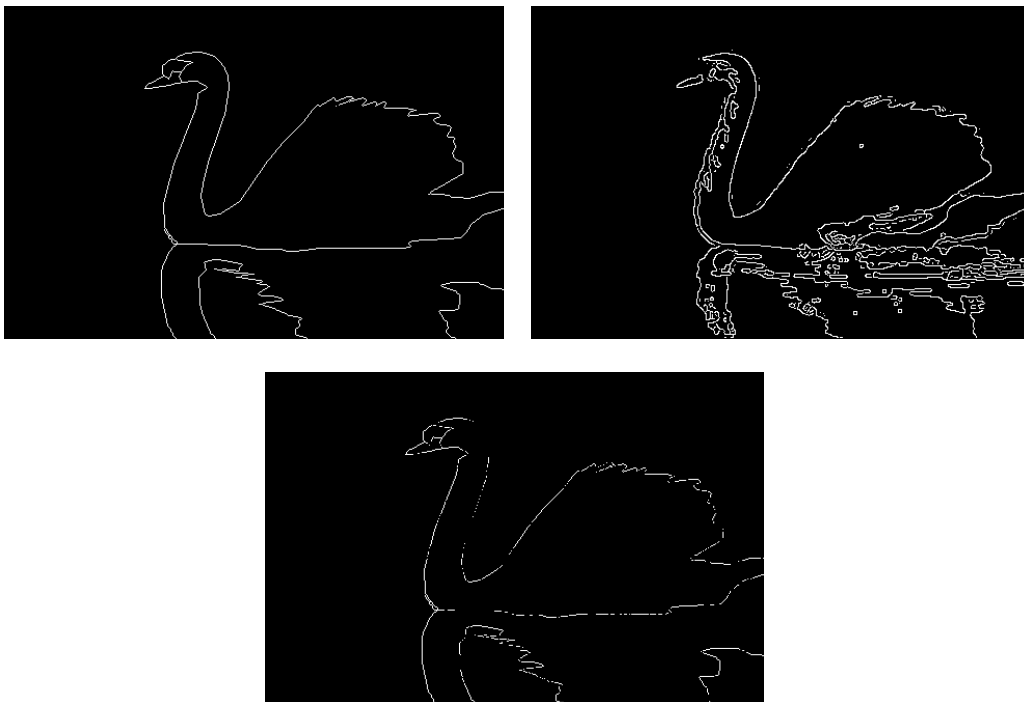


Figura 16. Imágenes de bordes de muestra. Arriba a la izq.: se muestra la imagen de *ground truth*. Arriba a la der.: se encuentra la imagen de bordes resultante de la segmentación. Abajo, la imagen diferencia.

2.3.2.2 Regiones

Al contrario que en el método de bordes, donde sólo se tienen en cuenta los píxeles que conforman el borde de las regiones, en el método de regiones los píxeles a tener en cuenta son todos aquellos que conforman cada una de las regiones segmentadas. Es por ello que, a priori, este método es mucho más real a la hora de evaluar la segmentación, puesto que, la forma de evaluación se adapta más a nuestras necesidades.

Para proceder el método es el mismo que en el anterior caso. En primer lugar es necesario obtener un *ground truth*. En segundo lugar se evalúa la imagen que se obtiene de la segmentación con ese *ground truth*.

Al igual que en el anterior caso, se numerarán y explicarán los distintos procedimientos que se siguen para la evaluación.

2.3.2.2.1 Obtención del *ground truth*

Es la imagen con la que se van a comparar las imágenes obtenidas de nuestros algoritmos. Dependiendo del tipo de imagen (squares o reales), la forma de obtenerla cambia.

Squares

Como es un método que cuenta el número de píxeles de cada región, no es necesario realizar ningún tipo de transformación a la imagen puesto que estas imágenes ya vienen etiquetadas en 4 regiones diferentes, donde cada rectángulo pertenece a uno de ellos.

Así pues, la imagen de *ground truth* es la imagen en sí y no será necesario realizar ninguna modificación en este paso (Fig. 17).



Figura 17 Imágenes squares utilizadas como *ground truth* en el método de regiones.

Reales

En el caso de estas imágenes es diferente puesto que las imágenes no vienen etiquetadas de ninguna manera. Somos nosotros mismos los que definimos el número de etiquetas que creemos debe tener la imagen y a raíz de ello ser capaz de diseñar un *ground truth* de la imagen original.

Para realizarlo, partimos de la imagen de bordes que tenemos de *ground truth* de bordes de cada imagen real (Fig. 14). A partir de esa imagen de bordes y mediante un programa de edición de imágenes como Gimp [35] rellenamos cada región de un color distinto para poder distinguirlo (Fig. 18).



Figura 18. Imagen de *ground truth* utilizada para las imágenes del grupo reales

2.3.2.2.2 Algoritmo de evaluación de regiones

Una vez ya tenemos el *ground truth* de todas las imágenes, procedemos a las comparaciones de píxeles entre regiones.

El método que utilizamos se basa en comparar los píxeles que pertenecen a cada región y evaluarlos según true positive, true negative, false positive o false negative. Más adelante se explican las características que tienen cada uno de ellos.

Como se puede intuir y dado que lo que se compara es la pertenencia de los píxeles a una región dada, es necesario que las regiones del *ground truth* y de la imagen segmentada estén etiquetadas de la misma

manera, ya que, si las etiquetas no fuesen iguales estaríamos comparando dos regiones distintas y por lo tanto los resultados numéricos no serían fiables puesto que los píxeles que se comparan no serían de las mismas regiones.

Para ello, el proceso de comparación se compone de dos pasos:

1. Se ordenan las etiquetas de la misma manera tanto en el *ground truth* como en la imagen obtenida.

El algoritmo que se ha desarrollado para realizar esta tarea consiste básicamente en evaluar el número de píxeles que existen de cada región e ir asignándole de manera descendente el número de etiqueta a cada región. Así si tuviésemos dos regiones donde una de ellas tiene doscientos (200) píxeles y otra de ellas quinientos (500), a la región de 500 píxeles se le asignaría la etiqueta número 1 y a la región de 200 píxeles la etiqueta número dos.

Con esto conseguimos que ambas imágenes (*ground truth* e imagen segmentada) tengan las mismas etiquetas para las mismas regiones, ya que las diferencias de regiones no serán abismales y este método funcionará (Fig. 19).



Figura 19. A la izq. la imagen original. A la der. La imagen tras ordenarse las etiquetas de mayor a menor.

2. Se aplica el algoritmo para evaluar las regiones.

Este algoritmo consiste en la búsqueda de los píxeles que componen cada región y evaluarlos.

En primer lugar el algoritmo separa la imagen en tantos planos como regiones diferentes tenga, tanto la imagen de *ground truth* como la imagen segmentada. Así, si la imagen tiene 4 etiquetas, se crearán 4 planos. Cada plano contendrá una región y 1's donde se sitúe cada región de píxeles.

Una vez tenemos las dos imágenes separadas por planos se comparan las imágenes plano por plano según los siguiente criterios.

Se considerarán píxeles true positive (TP) cuando los píxeles de una misma región coincidan en ambas imágenes. Esto es, si tenemos el primer plano y un píxel es etiquetado en ese plano como perteneciente a la región que hace referencia en ambas imágenes, será true positive.

De manera contraria, se considerarán píxeles true negative (TN) cuando los píxeles de un plano, no estén etiquetados como pertenecientes a dicho plano en ambas imágenes.

Se considerarán píxeles false positive (FP) cuando un píxel de la imagen segmentada esté etiquetado como perteneciente a una región, pero en la imagen de *ground truth* ese píxel no sea perteneciente a dicha región.

Por último, se tienen los píxeles false negative (FN) cuando un píxel de la imagen segmentada esté etiquetado como no perteneciente a una región pero sí lo esté en la imagen de *ground truth*.

Cuando ya tenemos todos los píxeles clasificados según estos cuatro grupos, procedemos a obtener un valor numérico que nos muestre cómo de buena ha sido nuestra clasificación (segmentación).

Sensitivity

Mide la proporción de píxeles correctamente identificados. Se calcula como:

$$\frac{TP}{TP + FN} \quad (2.46)$$

Specifity

Mide la proporción de píxeles que han sido identificados de manera incorrecta. Se calcula como:

$$\frac{TN}{TN + FP} \quad (2.47)$$

Accuracy

Mide la precisión con la que se ha segmentado la imagen. Es la medida más completa ya que aúna la cantidad de los 4 tipos de píxeles. Se calcula como:

$$\frac{TP + TN}{TP + TN + FP + FN} * 100 (\%) \quad (2.48)$$

En la siguiente figura (Fig. 20) se muestran tres imágenes. La imagen de arriba a la izquierda es la imagen de *ground truth* de regiones. La imagen de arriba a la derecha es la imagen de regiones de la segmentación realizada con el algoritmo de Continuous Max-Flow y K-means con la distancia CIE2000 y 4 etiquetas. La imagen de abajo se corresponde a la imagen diferencia de ambas. Si las imágenes fuesen totalmente iguales el resultado debería ser una imagen totalmente negra. Sin embargo, al existir diferencias hay zonas que quedan de color blanco que representan los píxeles de no coincidencia entre las imágenes.

Como se podrá comprobar más adelante, los valores al realizar este tipo de cálculo es mucho mejor y más realista que en el caso de los bordes, ya que aquí se tienen en cuenta cómo se etiquetan los píxeles y no la forma que adquiere la imagen.



Figura 20. Arriba a la izq.: imagen *ground truth* de regiones. Arriba a la der.: imagen de bordes segmentada con el algoritmo de Continuous Max-Flow. Abajo: imagen diferencia de regiones.

3 RESULTADOS

“Do things that have never been done before”

-Russell Kirsch -

Este capítulo servirá para introducir los resultados que hemos obtenidos al poner en práctica la conjunción de los algoritmos propuestos en el capítulo anterior. En primer lugar, se detallará el uso de los algoritmos utilizados para poner en práctica lo que se propuso de forma teórica (véase 2.1.4.2.3). En segundo lugar, se comentarán los diferentes experimentos que se han llevado a cabo para la consecución de los resultados numéricos. Por último, se mostrarán los resultados más representativos que se han obtenido ordenados de una manera clara mediante tablas.

3.1 Algoritmos utilizados

En este apartado explicaremos de forma resumida los algoritmos más importantes utilizados para alcanzar los resultados que hemos obtenido. Todos los algoritmos que se han utilizado en este proyecto están escritos en código Matlab, perfectamente comentados, y se pueden solicitar a través en mi dirección de correo electrónico³.

3.1.1 Algoritmo de K-means

El algoritmo para la segmentación en *clusters* de imágenes en color utilizando el espacio de color L^*a^*b , K-means, ha sido utilizado como método de etiquetado para la posible combinación con el algoritmo de Continuous Max-Flow. El algoritmo que hemos utilizado se encuentra en [25] y nos ofrece la posibilidad de etiquetar la imagen en color en tantos *clusters* como queramos. Sin embargo, este algoritmo tan sólo nos ofrecía la posibilidad de usar la distancia euclídea. Es por ello, que el algoritmo verdaderamente utilizado contiene algunas modificaciones con respecto al citado anteriormente.

Las modificaciones básicas que se le han hecho con respecto al algoritmo original son las siguientes:

1. Hemos añadido la posibilidad de utilizar otras distancias. Además de la distancia euclídea, como se ha comentado a lo largo del proyecto, también nos interesa utilizar las distancias CIEDE00 y CIEDE94. Es por ello, que se ha modificado el código para tener la posibilidad de elegir la distancia que necesitemos en cada caso. Como es obvio, también contiene las fórmulas necesarias para llevar a cabo el cálculo de dichas distancias.

³ Dirección de correo electrónico: luisfpintos@gmail.com

2. Además, también se han almacenado los vectores de distancia de los distintos píxeles a los centroides tanto en la primera iteración como en la última. Esto es clave, puesto que, son valores que deberemos incorporar al algoritmo de Continuous Max-Flow para que pueda realizar la segmentación de las imágenes.
3. Por último, también se ha tenido en cuenta otro factor como es el tiempo de computación. Se han recogido tanto el tiempo computacional en la primera iteración de K-means como en la última (véase 3.2.2).

3.1.2 Algoritmo de Continuous Max-Flow

El algoritmo en el que se fundamenta el estudio y análisis de este trabajo de fin de grado es un algoritmo que se puede encontrar en [26].

El software que se obtiene incluye un programa que ha sido diseñado de forma eficiente para resolver los problemas de segmentación de imágenes multi-region en 2D/3D. Este software, como es lógico, se fundamenta en las ecuaciones enunciadas en los apartados anteriores (2.40).

Este algoritmo ha sido desarrollado para tres herramientas de programación, matlab, C y GPU. El uso para matlab es muy sencillo de utilizar.

Para que el software pueda resolver los problemas de segmentación multi-región, es necesario que se le introduzcan los valores de las diferentes capacidades C_i , el número de etiquetas $nlab$ y los valores de *penalty*, entre otros. Es en este lugar donde el funcionamiento conjunto del algoritmo de K-means explicado en el apartado anterior y el algoritmo de Continuous Max-Flow tiene sentido. Los valores de las capacidades C_i vienen dados por el vector de distancias de cada uno de los píxeles al centroide (Ec. 3.2). Este vector es el generado por el algoritmo de K-means. El número de etiquetas es el mismo que el que se creyó conveniente utilizar en el algoritmo de K-means. Los valores de *penalty* son utilizados como parámetros de penalización en la detección de bordes y se calcula mediante dos parámetros introducidos *ag* y *bg* como:

$$penalty = \frac{bg}{1 + ag \cdot grad(imagen)} \quad (3.1)$$

Donde *grad(imagen)* indica la función gradiente de matlab realizada a la imagen.

$$\min_{\{\Omega_i\}_{i=1}^n} \sum_i^n \int_{\Omega_i} C_i(x) dx + \alpha \sum_i^n |\partial\Omega_i| \quad (3.2)$$

Los valores de los $C_i(x)$ son los costes de las capacidades que coinciden en este caso con la distancia de cada uno de los píxeles a los centroides correspondientes. Es la aplicación de la Ec. 2.40 a nuestro caso en concreto, donde se combinan el algoritmo de Max-Flow y K-means.

En este caso, no ha sido necesario realizar ningún tipo de modificación al software original. Tan sólo hemos tenido que construir un script que realizase las modificaciones necesarias a la imagen que se va a introducir en el algoritmo y llame al software que realiza la segmentación.

3.2 Experimentos realizados para la consecución de los resultados

A continuación, explicaremos los distintos experimentos que se han llevado a cabo para poder obtener los resultados numéricos. Por un lado, hemos realizado la segmentación de las imágenes en color mediante el algoritmo de K-means únicamente. Por otro lado, la segmentación realizada mediante la combinación del algoritmo de K-means y C.Max-Flow.

3.2.1 Kmeans

La segmentación de las imágenes en color mediante el algoritmo de K-means (véase 3.1.1) se ha realizado para poder comparar los resultados que se obtienen mediante este método y mediante la actuación combinada del algoritmo de K-means y C.Max-Flow y poder ver así si los resultados mejoran las segmentaciones de las imágenes en color.

A pesar de que como se explicó en apartados anteriores, donde el *ground truth* tiene definidas unas etiquetas que nosotros creemos que son las mejores para su segmentación, el algoritmo de K-means segmenta las imágenes reales a color de 2 a 5 etiquetas cada una de ellas. Con ello, podremos comprobar cuál es la influencia del número de etiquetas en las distintas imágenes en los resultados de la segmentación.

3.2.2 K-means + C.Max-Flow

Como ya se ha explicado en apartados anteriores, para poder realizar la segmentación mediante el algoritmo de C.Max-Flow es necesario un vector de distancia de los píxeles a cada uno de los centroides. En relación a esto se han realizado dos tipos de experimentos.

En primer lugar, se ha ejecutado el algoritmo de C.Max-Flow con el vector de distancias que el algoritmo de K-means devuelve en la primera iteración. Con ello, dejamos que el algoritmo de C.Max-Flow realice la mayor carga de trabajo en la segmentación.

En segundo lugar, se ha ejecutado el algoritmo con el vector de distancias que el algoritmo de Kmeans devuelve en la última iteración. Este experimento se utiliza para ver si el algoritmo realiza alguna mejora notoria sobre la segmentación que K-means realiza en las imágenes a color (véase **Observación 1**).

Observación 1: se ha comprobado que el hecho de introducir al algoritmo de Continuous Max-Flow el vector de distancias de la última iteración del algoritmo de K-means no mejora los resultados obtenidos con éste. Por lo tanto, no se mostrarán los resultados numéricos en las tablas, ya que no tiene ninguna relevancia más allá de esta observación.

Por otro lado, el algoritmo de C.Max-Flow admite los valores de *penalty* (véase 3.1.2). En este sentido, se han utilizado 19 valores para *ag* y 19 valores para *bg*, dando lugar a un total de 361 variaciones de los valores de *penalty*.

Por lo tanto, si tenemos que cada imagen ha sido etiquetada por K-means con 2, 3,4 y 5 etiquetas y cada imagen tiene un total de 361 valores distintos de *penalty*, cada una de las imágenes reales en color lleva consigo un experimento de un total de 1444 grupos de valores distintos (*sensitivity*, *specificity* y *accuracy*) que el algoritmo de C.Max-Flow devuelve que hay que comprobar. Además, el experimento se repite dos veces, puesto que también se comprueba el hecho de probar con el vector de distancias en la primera iteración y en la última. Por lo que en total son 2888 conjuntos de valores que el algoritmo de C.Max-Flow devuelve para cada imagen.

Es por ello que, a raíz de la gran cantidad de datos que tenemos para cada una de las imágenes, ha sido necesaria la programación de algoritmos de búsqueda de los valores más óptimos.

3.3 Resultados

En este apartado se expondrán todos los resultados más relevantes que se han obtenido en el desarrollo del

proyecto sin comentar las conclusiones que se pueden obtener (véase 4.1). En primer lugar, se presentarán los resultados que se han obtenido de las imágenes denominadas “squares”. En segundo lugar, se expondrán los resultados de las imágenes denominadas “reales” (véase 2.3.1). Todos los valores se presentarán a modo de tabla para que se puedan comparar los resultados.

3.3.1 Imágenes Squares

Las imágenes de los rectángulos de colores (squares) han sido utilizadas para ver cuál es la diferencia existente entre el uso de las distintas distancias que se han propuesto utilizar en este proyecto.

Como ya se ha dicho, estas imágenes tienen la característica de que muchos de sus colores son prácticamente iguales, de hecho, en muchas de ellas es casi imposible detectar la existencia de cuatro rectángulos a simple vista.

En relación a ese problema, a la hora de ejecutar el algoritmo de K-means, muchas veces nos daba el error “*Empty cluster at iteration 1*”. Esto quiere decir que, en la primera iteración, el algoritmo de K-means no era capaz de encontrar los 4 centroides correctamente, ya que la inicialización de los centroides estaba configurada para que se realice de manera aleatoria.

Para resolver este problema se ejecutó el algoritmo de dos maneras:

1. Por un lado, se ejecutó el algoritmo con cada una de las imágenes tantas veces como fuese necesario para que no diese el error de *cluster* vacío en la primera iteración. (véase **Observación 2**)
2. Por otro lado, se cambió la forma en la que el algoritmo trata el error de *cluster* vacío para que en el caso de que encontrara un *cluster* vacío, en lugar de lanzar el error y que parase la ejecución, crease un nuevo clúster basado en futuras observaciones de su centroide. En el algoritmo, esta opción se modifica mediante el parámetro “*EmptyAction*” con la opción “*singleton*”. El caso anterior tendría la opción “*error*” en su lugar.

Observación 2: el hecho de que el algoritmo de K-means sea ejecutado todas las veces que sea necesario para que el algoritmo no dé error de *cluster* vacío provoca que, todos los resultados obtenidos en relación a este tipo de imágenes sean siempre el valor máximo sea cual sea la distancia y el algoritmo que se aplique. Es por ello que, no se muestran los valores que se han conseguido de esta forma, ya que el algoritmo debe ser capaz de avanzar sin ningún tipo de error y sin supervisión y con esta metodología no lo conseguimos. Aún así, los resultados están disponibles en el paquete donde se incluye el código del proyecto.

Los resultados que se muestran a continuación no son el total de las 47 imágenes, pues no considero relevante el hecho de tener que mostrarlas todas.

La forma en la que se organiza la información es una tabla donde se muestran: la imagen original y las imágenes tomadas como referencia en cada uno de los experimentos⁴, los valores máximos obtenidos correspondientes a cada uno de los experimentos junto con sus imágenes y las distancias correspondientes (Eu, CIEDE94 y CIEDE00). No es necesario indicar el número de etiquetas de cada experimento, puesto que, en este caso es siempre 4. Además, también se incluirán los valores de *penalty* en el caso del algoritmo de C.Max-Flow.

Los valores que se muestran en la tabla son:

-*Recall* (R).

-*Precision* (P).

-Valor F (F).

-*Accuracy* (AC).

-*Specifity* (SP).

-*Sensitivity* (SE).

-Tiempo de computación (Tc segundos).

⁴ En el caso de las imágenes de los cuadrados, la imagen referencia en el cálculo de regiones es la misma que la imagen original.


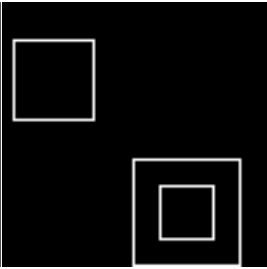
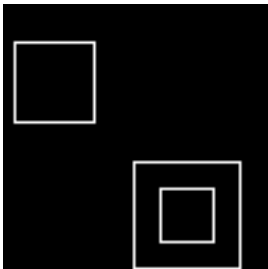
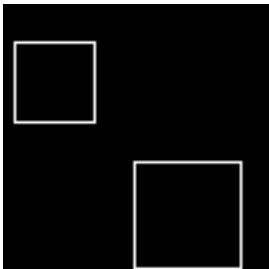

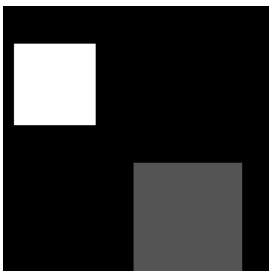
								
		Imagen: 1.bmp						
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>				
Bordes					Todas las distancias		Todas las distancias	
	R	P	F	Tc	R	P	F	Tc
	1	1	1	1,32s	0,778	1	0,875	0,832s
Regiones					EU, CIEDE 94		Todas las distancias	
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	1,32s	97,79	0,985	0,956	0,832s

Tabla 1. Resultados de la imagen 1.bmp.


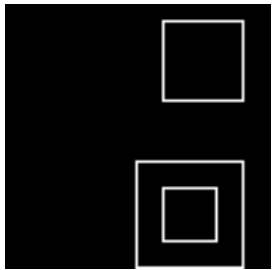
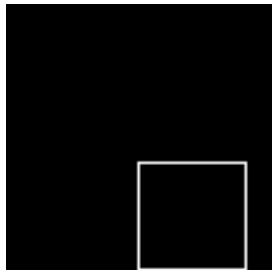


									
		Imagen: 2.bmp							
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>					
Bordes									
		Todas las distancias		Todas las distancias					
		R	P	F	Tc	R	P	F	Tc
		1	1	1	0,0976s	0,444	1	0,615	0,01s
Regiones									
		Todas las distancias		Todas las distancias					
		AC	SP	SE	Tc	AC	SP	SE	Tc
		100	1	1	0,097s	92,99	0,953	0,859	0,01s

Tabla 2. Resultados de la imagen 2.bmp.


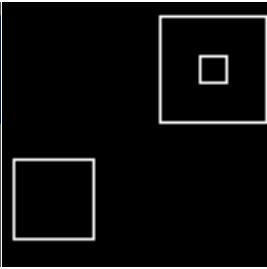
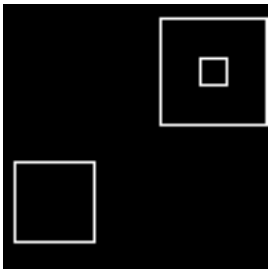
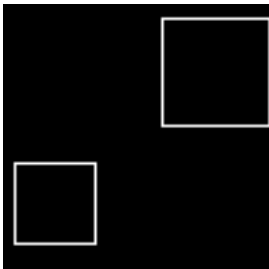


								
		Imagen: 10.bmp						
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>				
Bordes					Todas las distancias		Todas las distancias	
	R	P	F	Tc	R	P	F	Tc
	1	1	1	1,84s	0,875	1	0,933	1,32s
Regiones					Todas las distancias		Todas las distancias	
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	1,84s	99,39	0,996	0,988	1,32s

Tabla 3. Resultados de la imagen 10.bmp.


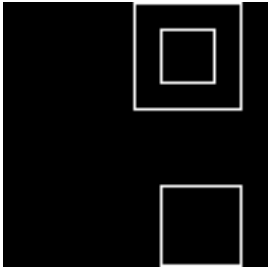
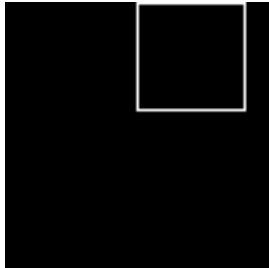


 <p style="text-align: center;">Imagen: 11.bmp</p>								
K-means					K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>			
Bordes	 <p style="text-align: center;">Todas las distancias</p>				 <p style="text-align: center;">Todas las distancias</p>			
	R	P	F	Tc	R	P	F	Tc
	1	1	1	1,48s	0,444	1	0,615	0,778s
Regiones	 <p style="text-align: center;">Todas las distancias</p>				 <p style="text-align: center;">Todas las distancias</p>			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	1,48s	92,99	0,953	0,859	0,778s

Tabla 4. Resultados de la imagen 11.bmp.


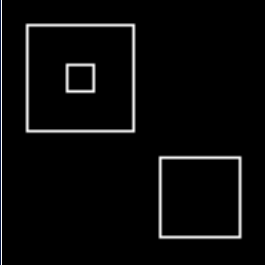
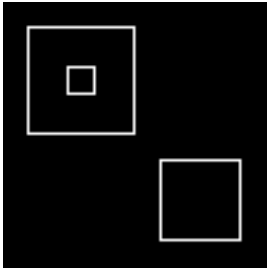
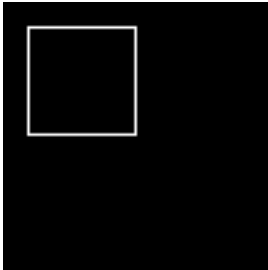

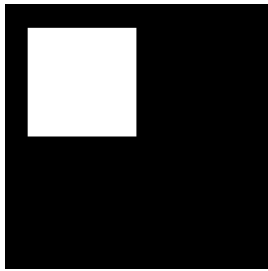
								
		Imagen: 12.bmp						
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>				
Bordes								
			Todas las distancias		Todas las distancias			
	R	P	F	Tc	R	P	F	Tc
	1	1	1	0,09s	0,5	1	0,666	0,01s
Regiones								
			Todas las distancias		Todas las distancias			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	0,09s	94,59	0,964	0,892	0,01s

Tabla 5. Resultados de la imagen 12.bmp.


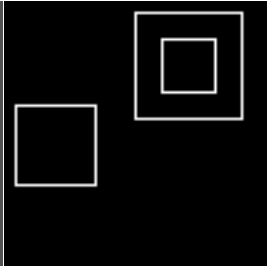
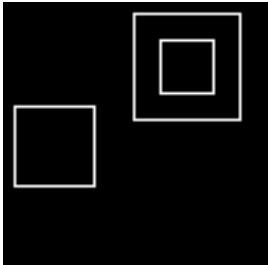
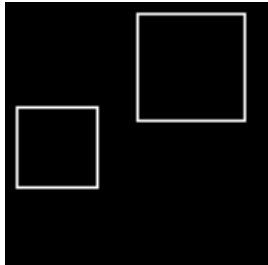

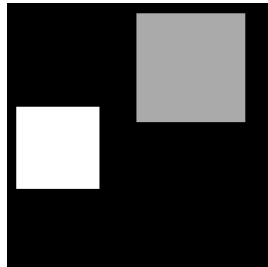
								
		Imagen: 13.bmp						
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>				
Bordes								
			EU,CIEDE 94		CIEDE 00			
	R	P	F	Tc	R	P	F	Tc
	1	1	1	0,937s	0,778	1	0,875	0,473s
Regiones								
			EU,CIEDE 94		CIEDE 00			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	0,937s	97,79	0,985	0,956	0,473s

Tabla 6. Resultados de la imagen 13.bmp.


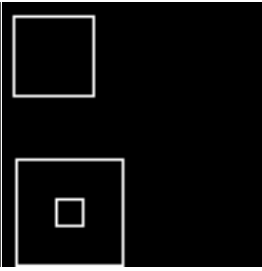
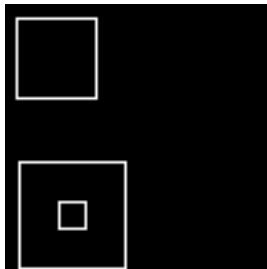
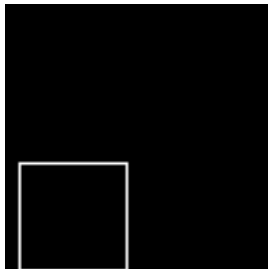

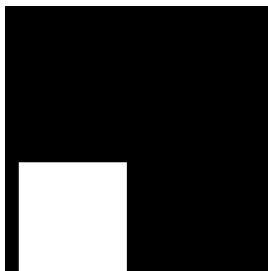
  <p style="text-align: center;">Imagen: 14.bmp</p>								
<p>K-means</p>		<p>K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i></p>						
Bordes	 <p style="text-align: center;">EU</p>		 <p style="text-align: center;">Todas las distancias</p>					
	R	P	F	Tc	R	P	F	Tc
	1	1	1	2,55s	0,5	1	0,667	1,23s
Regiones	 <p style="text-align: center;">EU</p>		 <p style="text-align: center;">Todas las distancias</p>					
	AC	SP	SE	Tc	AC	SP	SE	Tc
	100	1	1	2,55s	94,59	0,964	0,892	1,23s

Tabla 7. Resultados de la imagen 14.bmp.

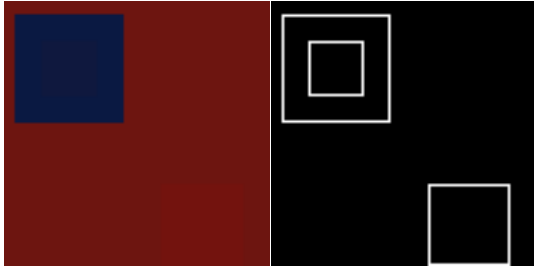
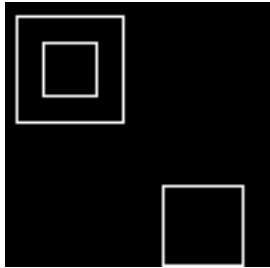
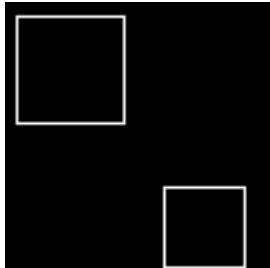

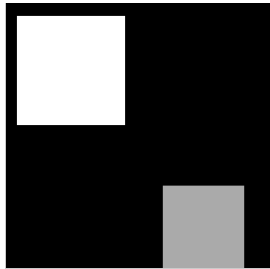
									
		Imagen: 15.bmp							
		K-means		K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>					
Bordes									
		EU,CIEDE 94		Todas las distancias					
		R	P	F	Tc	R	P	F	Tc
		1	1	1	1,27s	0,777	1	0,875	0,775s
Regiones									
		EU,CIEDE94		Todas las distancias					
		AC	SP	SE	Tc	AC	SP	SE	Tc
		100	1	1	1,27s	97,79	0,985	0,956	0,775s

Tabla 8. Resultados de la imagen 15.bmp.

3.3.2 Imágenes Reales

Los resultados que se muestran a continuación son los referidos a las imágenes reales obtenidas de la base de datos de Berkeley [13]. La forma en la que se organizarán los valores presentados sigue la misma línea que el apartado anterior.

En este caso, sí será necesario mostrar el número de etiquetas *n_{labs}* con el que está segmentada la imagen y con el que están etiquetados los resultados que se muestran (*n_{labsBordes}* y *n_{labsRegiones}*), ya que, el número de etiquetas de los resultados más óptimos no tienen por qué coincidir con el número de etiquetas de la imagen tomada como referencia. Sin embargo, el problema que se daba con las imágenes de cuadrados (squares) de “*empty clusters*” no se da en este tipo de imágenes.

(Ver tablas siguiente página)

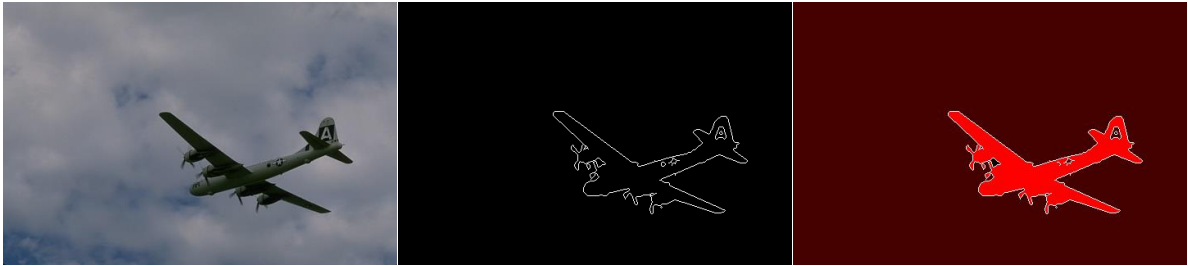
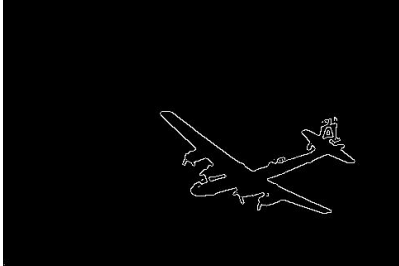



								
<p>Imagen: 3096.jpg nlabs = 2 nlabsBordes = 2 nlabsRegiones = 2</p>								
K-means				K-Means(Iter1)+C.Max-Flow <i>Penalty: ag = 0,01, bg = 0,01</i>				
Bordes	 EU				 CIEDE 00			
	R	P	F	Tc	R	P	F	Tc
	1	0,797	0,887	0,65s	1	0,828	0,906	5,87s
Regiones	 EU				 CIEDE 00			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	99,03	0,993	0,985	0,65s	99,17	0,994	0,988	5,87s

Tabla 9. Resultados de la imagen 3096.jpg.

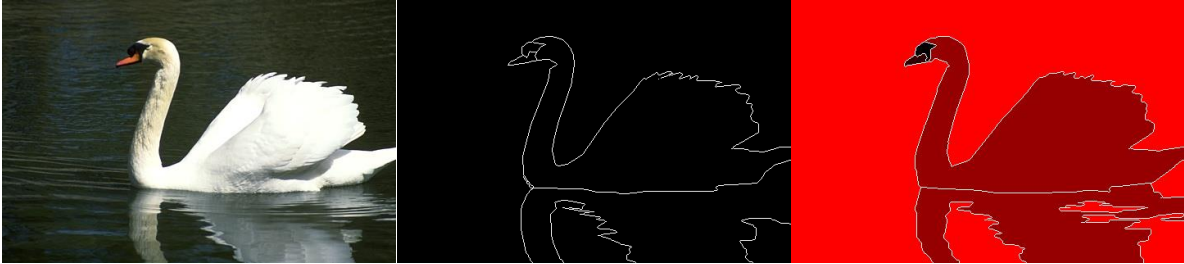
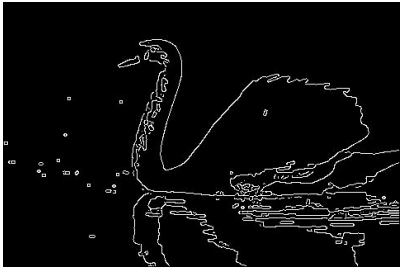
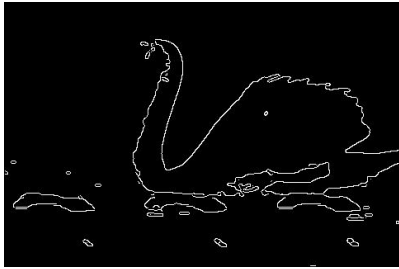
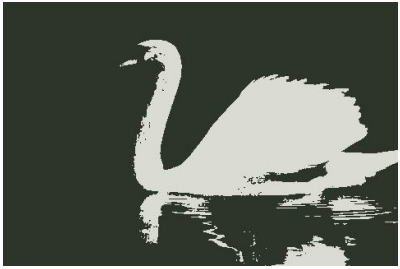
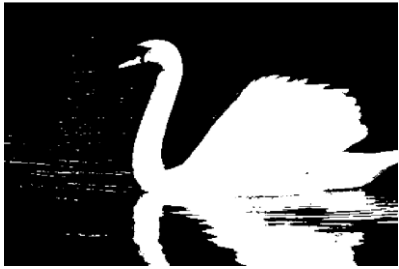
 <p style="text-align: center;">Imagen: 8068.jpg nlabs = 4 nlabsBordes = 3 nlabsRegiones = 2</p>								
K-means				K-Means(Iter1)+C.Max-Flow Penalty Bordes: ag = 0,01, bg = 9,666 Penalty Regiones: ag = 0,01, bg = 0,01				
Bordes	 CIEDE00				 CIEDE 00			
	R	P	F	Tc	R	P	F	Tc
	0,972	0,349	0,513	24,17s	0,724	0,449	0,554	5,44s
Regiones	 CIEDE 00				 EU			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	94,27	0,962	0,886	45,30s	98,33	0,989	0,967	0,20s

Tabla 10. Resultados de la imagen 8068.jpg.

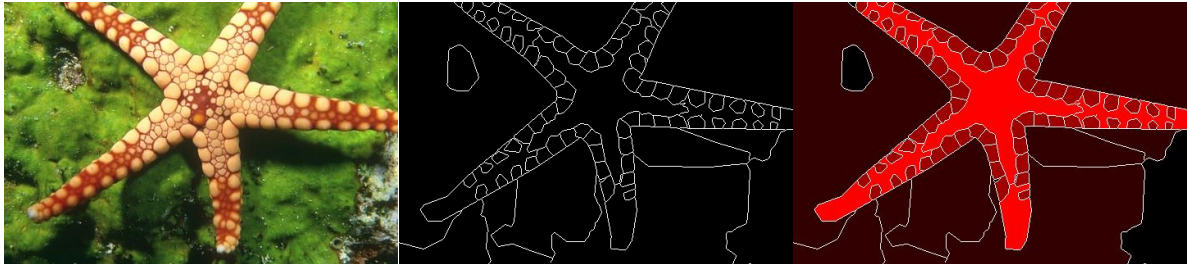
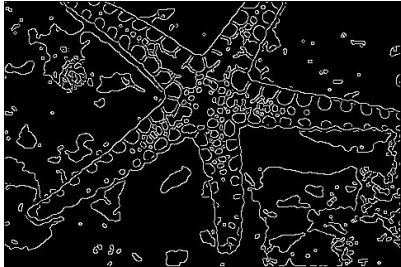


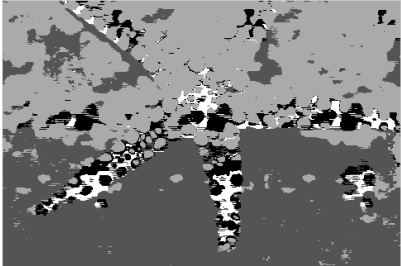
 <p style="text-align: center;">Imagen: 12003.jpg nlabs = 4 nlabsBordes = 4 nlabRegiones = 4</p>								
K-means				K-Means(Iter1)+C.Max-Flow <i>Penalty Bordes: ag = 0,01, bg = 3,2289</i> <i>Penalty Regiones: ag = 17,7138, bg = 8,0572</i>				
Bordes	 EU				 CIEDE 94			
	R	P	F	Tc	R	P	F	Tc
	0,894	0,329	0,481	1,56s	0,913	0,323	0,477	0,19s
Regiones	 EU				 CIEDE 94			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	74,57	0,831	0,491	1,56s	73,33	0,822	0,467	0,19s

Tabla 11. Resultados de la imagen 12003.jpg.

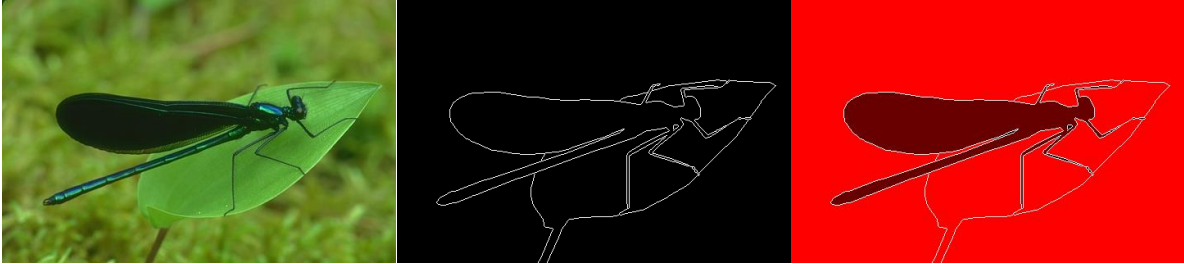
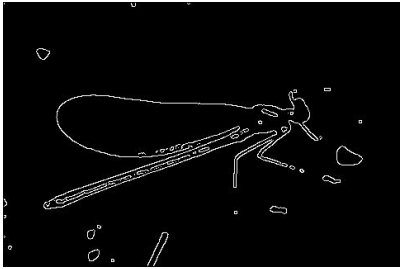
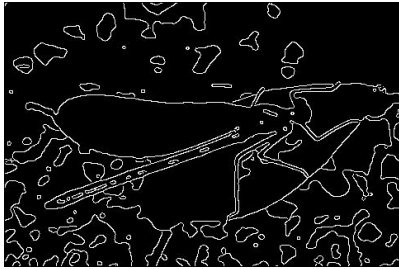

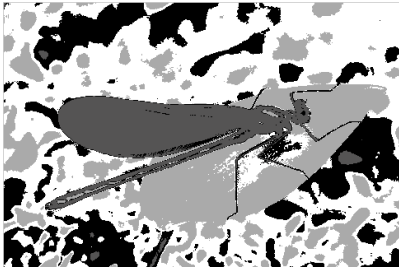
 <p style="text-align: center;">Imagen: 35070.jpg nlab = 2 nlabBordes = 2 (K-means) y 3 (K-means + C.Max-Flow) nlabRegiones = 2 (K-means) y 4 (K-means + C.Max-Flow)</p>								
K-means				K-Means(Iter1)+C.Max-Flow <i>Penalty Bordes: ag = 0,01, bg = 20,9327</i> <i>Penalty Regiones: ag = 0,01, bg = 0,01</i>				
Bordes	 EU				 EU			
	R	P	F	Tc	R	P	F	Tc
	0,798	0,642	0,711	0,861s	1	0,309	0,473	0,03s
Regiones	 EU				 CIEDE 94			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	98,43	0,988	0,976	0,861s	92,53	0,951	0,888	0,12s

Tabla 12. Resultados de la imagen 35070.jpg.








  								
<p>Imagen: 118035.jpg nlabs = 4 nlabsBordes = 3 (K-means) y 2 (K-means+C.Max-Flow) nlabsRegiones = 3 (K-means) y 4 (K-means+C.Max-Flow)</p>								
K-means				K-Means(Iter1)+C.Max-Flow <i>Penalty Bordes: ag = 0,01, bg = 0,01</i> <i>Penalty Regiones: ag = 0,01, bg = 0,01</i>				
Bordes	 <p style="text-align: center;">EU</p>				 <p style="text-align: center;">CIEDE 00</p>			
	R	P	F	Tc	R	P	F	Tc
	1	0,498	0,665	1,08s	0,971	0,455	0,619	5,49s
Regiones	 <p style="text-align: center;">CIEDE 94</p>				 <p style="text-align: center;">CIEDE 94</p>			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	96,52	0,978	0,913	1,64s	95,52	0,972	0,888	0,198s

Tabla 13. Resultados de la imagen 118035.jpg.

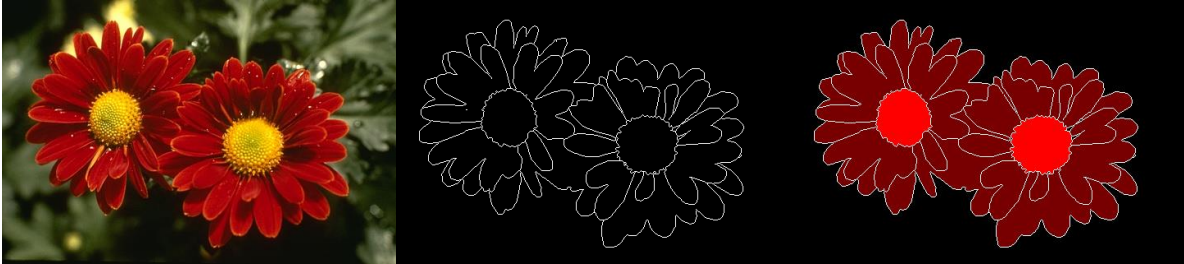
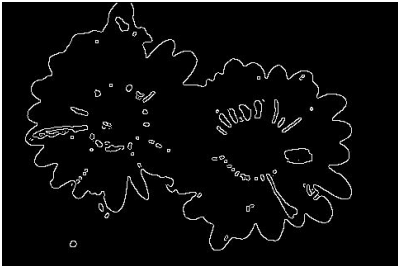
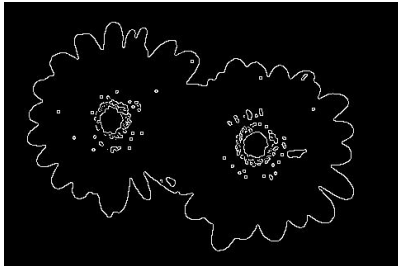
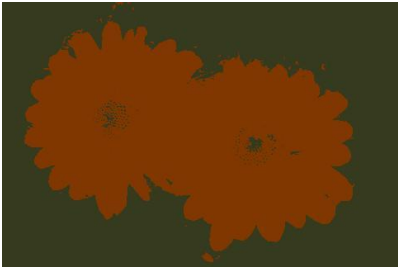

																			
<p>Imagen: 124084.jpg nlabs = 3 nlabsBordes = 2 nlabsRegiones = 2</p>																			
<p>K-means</p>		<p>K-Means(Iter1)+C.Max-Flow <i>Penalty Bordes: ag = 27,3704, bg = 28,9799</i> <i>Penalty Regiones: ag = 4,8383, bg = 12,8855</i></p>																	
Bordes	 EU	 EU																	
	<table border="1"> <thead> <tr> <th>R</th> <th>P</th> <th>F</th> <th>Tc</th> </tr> </thead> <tbody> <tr> <td>0,588</td> <td>0,599</td> <td>0,594</td> <td>0,91s</td> </tr> </tbody> </table>	R	P	F	Tc	0,588	0,599	0,594	0,91s	<table border="1"> <thead> <tr> <th>R</th> <th>P</th> <th>F</th> <th>Tc</th> </tr> </thead> <tbody> <tr> <td>0,551</td> <td>0,629</td> <td>0,587</td> <td>0,15s</td> </tr> </tbody> </table>		R	P	F	Tc	0,551	0,629	0,587	0,15s
	R	P	F	Tc															
0,588	0,599	0,594	0,91s																
R	P	F	Tc																
0,551	0,629	0,587	0,15s																
Regiones	 CIEDE 00	 EU																	
	<table border="1"> <thead> <tr> <th>AC</th> <th>SP</th> <th>SE</th> <th>Tc</th> </tr> </thead> <tbody> <tr> <td>93,36</td> <td>0,950</td> <td>0,9</td> <td>0,84s</td> </tr> </tbody> </table>	AC	SP	SE	Tc	93,36	0,950	0,9	0,84s	<table border="1"> <thead> <tr> <th>AC</th> <th>SP</th> <th>SE</th> <th>Tc</th> </tr> </thead> <tbody> <tr> <td>94,14</td> <td>0,957</td> <td>0,913</td> <td>0,15s</td> </tr> </tbody> </table>		AC	SP	SE	Tc	94,14	0,957	0,913	0,15s
	AC	SP	SE	Tc															
93,36	0,950	0,9	0,84s																
AC	SP	SE	Tc																
94,14	0,957	0,913	0,15s																

Tabla 14. Resultados de la imagen 124084.jpg.


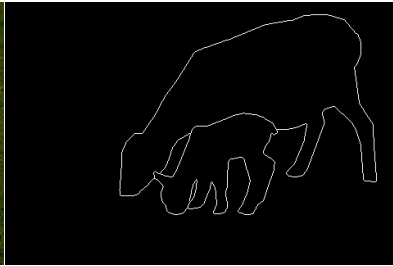
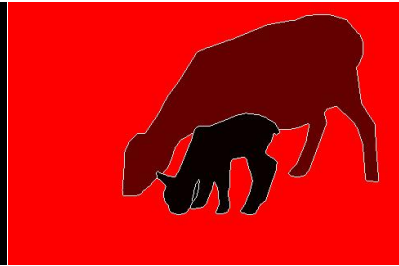
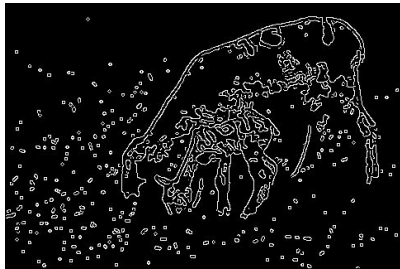



  								
<p align="center">Imagen: 207056.jpg nlabs = 3 nlabsBordes = 4 (K-means) y 2 (K-means+C.Max-Flow) nlabsRegiones = 2</p>								
<p align="center">K-means</p>		<p align="center">K-Means(Iter1)+C.Max-Flow Penalty Bordes: ag = 8.0572, bg = 12.885 Penalty Regiones: ag = 0,01, bg = 14,4949</p>						
Bordes	 <p align="center">CIEDE 00</p>				 <p align="center">EU</p>			
	R	P	F	Tc	R	P	F	Tc
	0,938	0,153	0,26	39,66s	1	0,446	0,617	0,29s
Regiones	 <p align="center">CIEDE 00</p>				 <p align="center">EU</p>			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	91,80	0,945	0,836	15,29s	95,55	0,970	0,911	0,33s

Tabla 15. Resultados de la imagen 207056.jpg.


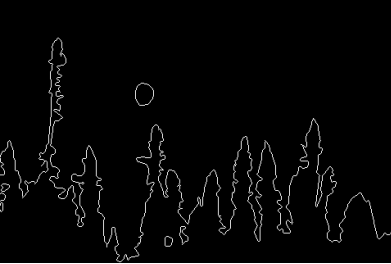
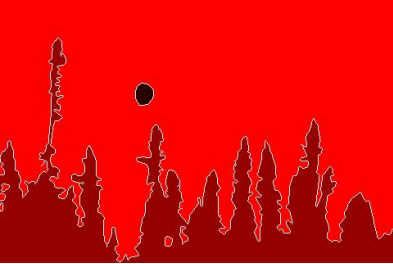
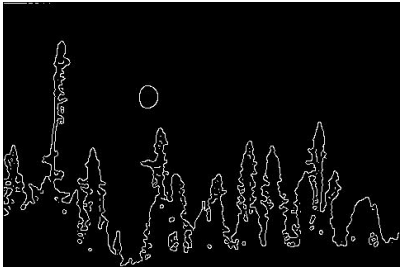
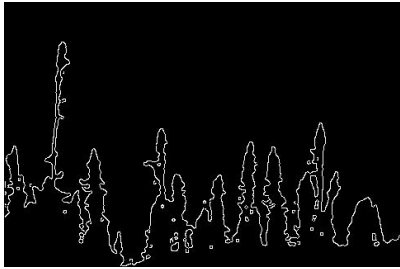


  								
<p>Imagen: 238011.jpg nlabs = 3 nlabsBordes = 4 (K-means) y 2 (K-means+C.Max-Flow) nlabsRegiones = 2</p>								
<p>K-means</p>		<p>K-Means(Iter1)+C.Max-Flow <i>Penalty Bordes: ag = 0,01, bg = 0,01</i> <i>Penalty Regiones: ag = 0,01, bg = 0,01</i></p>						
Bordes	 <p>CIEDE 94</p>				 <p>CIEDE 94</p>			
	R	P	F	Tc	R	P	F	Tc
	1	0,691	0,817	2,92s	0,963	0,695	0,807	0,13s
Regiones	 <p>EU</p>				 <p>CIEDE 94</p>			
	AC	SP	SE	Tc	AC	SP	SE	Tc
	98,22	0,988	0,964	0,66s	98,24	0,988	0,965	0,13s

Tabla 16. Resultados de la imagen 238011.jpg.

3.3.3 Valores promedio

Imágenes Squares							
K-means				K-Means(Iter1)+C.Max-Flow			
R	P	F	Tc	R	P	F	Tc
1	1	1	1,19s	0,637	1	0,766	0,678s
AC	SP	SE	Tc	AC	SP	SE	Tc
100	1	1	1,198s	95,99	0,973	0,919	0,678
Imágenes Reales							
K-means				K-Means(Iter1)+C.Max-Flow			
R	P	F	Tc	R	P	F	Tc
0,899	0,507	0,616	8,97s	0,890	0,516	0,629	2,19s
AC	SP	SE	Tc	AC	SP	SE	Tc
93,275	0,954	0,868	8,35s	93,35	0,955	0,873	0,9s

Tabla 17. Valores promedio de las imágenes.

Los algoritmos han sido ejecutados en un ordenador con las siguientes características:

-Procesador Intel Core i5 2,7Ghz.

-Memoria RAM 4GB 1333 MHz DDR3.

4 CONCLUSIONES Y LÍNEAS FUTURAS

“Things do not have to change the world to be important”

-Steve Jobs -

En este capítulo se presentarán las conclusiones alcanzadas de todo el proyecto gracias a los resultados obtenidos. Se comprobará si los objetivos iniciales pretendidos se han conseguido o no y el porqué de los resultados. Además también se comentarán las posibles líneas futuras que ofrece este trabajo.

4.1 Conclusiones

En este apartado se mostrarán las conclusiones que se han obtenido en relación a los resultados obtenidos en el capítulo anterior. Además, comentaremos si los objetivos iniciales del proyecto se han cumplido o no.

En primer lugar, destacar cómo en la mayoría de las imágenes reales, el número de etiquetas de la imagen segmentada, cuyos valores son máximos, no coincidan con la imagen de referencia (véase tabla 10). Esto es debido a que el *ground truth* no es totalmente realista al color y por lo tanto, existen problemas de coincidencia de píxeles. De hecho, si mostramos un ejemplo de la imagen segmentada con el mismo número de etiquetas, vemos cómo la segmentación con 4 etiquetas es mucho más realista en referencia a la imagen original (Fig. 21).

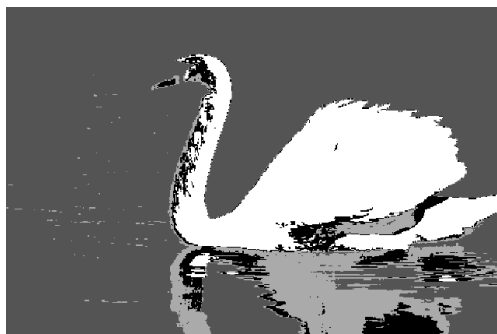


Figura 21. Imagen 8068.jpg con el mismo número de etiquetas que el *ground truth* (4 etiquetas) y segmentada con el algoritmo C.Max-Flow y distancia CIEDE 00.

En segundo lugar, comentar cómo el uso combinado del algoritmo de K-means en la primera iteración y el uso del algoritmo Continuous Max-Flow provocan que exista una mejora bastante notoria en el tiempo de computación, llegando muchas veces a ser incluso la décima parte, y mejorando en algunos casos la

segmentación de la imagen (tabla 9, 10, 13, 14, 15 y 16).

En tercer lugar, en las imágenes mostradas en el apartado de resultados, se ve como el uso del algoritmo de C.Max-Flow resalta más detalles en la segmentación en color que de por sí el algoritmo de K-means no es capaz de encontrar. En las imágenes de la tabla 12, se aprecia como el algoritmo C.Max-Flow es capaz de segmentar la imagen con mucho detalle. Si se compara el resultado de la segmentación con el algoritmo K-Means, se observará que el lujo de detalles es mucho mayor en el algoritmo C.Max-Flow. De hecho, si tenemos la misma imagen mostrada en la tabla 12 pero con muchas menos etiquetas, veremos como el algoritmo sigue siendo capaz de encontrar muchos detalles (Fig. 22). Otros ejemplos también pueden ser las imágenes usadas en la tabla 13, donde se muestran imágenes segmentadas con 4 etiquetas, pero incluso con dos (Fig. 23) se siguen apreciando los detalles.



Figura 22. Imagen 35070.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2 etiquetas.



Figura 23. Imagen 118035.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2 etiquetas.

En general, gracias a los resultados aportados en el apartado anterior y a las imágenes que se han mostrado en éste, se observa como el algoritmo C.Max-Flow es capaz de obtener una segmentación muy detallada incluso con pocas etiquetas, mejorando así la actuación del algoritmo K-means. Así, el algoritmo es capaz de segmentar tanto imágenes muy simples (tabla 9 y tabla 16) como imágenes un poco más complicadas (tabla 12, tabla 13 y tabla 15) y obtener una segmentación en color con un gran lujo de detalles en muy pocas etiquetas.

Como se ha visto, esto puede llegar a ser un gran problema puesto que, los *ground truth* de los que disponemos no se adaptan correctamente a nuestras necesidades y a los detalles que los algoritmos devuelven, provocando que los números que se muestran en el apartado resultados sea muchas veces engañoso acorde a la realidad.

En relación a los objetivos marcados (véase 1.2), podemos concluir que:

1. Se ha comprobado como la actuación combinada del algoritmo de K-means junto con el algoritmo de Continuous Max-Flow mejora notablemente la segmentación en color en comparación al simple uso del algoritmo de K-means, incluso con pocas etiquetas e imágenes con muchos detalles.

2. Se ha demostrado cómo el tiempo de computación de la segmentación de las imágenes se reduce notablemente al utilizar sólo la primera iteración de K-means en el algoritmo C.Max-Flow (véase T_c en las tablas del apartado 3.3.2).
3. Mediante el uso de las imágenes denominadas como square (tablas de las 1 a la 8) se ha intentado ver si existe una diferencia entre el uso de las distintas distancias que propone la CIE. Sin embargo, con este tipo de imágenes, los resultados que se han obtenido no son para nada relevantes puesto que las mejores distancias varían con las imágenes. Aún así, en los resultados ofrecidos por las imágenes reales se ve como existe una distancia que es la que mayor lujo de detalles ofrece, la CIEDE 00, incluso con pocas etiquetas.

En muchas tablas no está indicado que sea esta distancia la que mayores resultados ofrece pero, esto es debido a lo ya explicado en este mismo apartado, donde se indicaba que por causa de la naturaleza de la imagen tomada como *ground truth* no se tenían en cuenta los finos detalles de la segmentación. Aún así, existen imágenes realizadas con la distancia CIEDE00 y el algoritmo de segmentación C.Max-Flow que demuestran que esta distancia, con respecto a las otras, es la que más detalles ofrece sobre todo en imágenes con muchos colores (Fig. 22, Fig 23, Fig.24).



Figura 24. Imagen 207056.jpg segmentada con el algoritmo C.Max-Flow, distancia CIEDE 00 y 2 etiquetas.

4.2 Líneas Futuras

Algunas de las líneas futuras de investigación que este trabajo ofrece son consecuencia de los resultados obtenidos en el mismo. Entre las posibles líneas de trabajo destacan:

1. Búsqueda de nuevas imágenes con un *ground truth* que se adapte mucho mejor a las necesidades de este proyecto. Como se ha comentado, uno de los problemas ha sido la dificultad para desarrollar el *ground truth* de las imágenes y que éste se adaptase a las necesidades que este trabajo tiene, como la gran cantidad de detalles que las imágenes segmentadas ofrecen.
2. Prueba de otros algoritmos de etiquetado. En este proyecto se ha utilizado un algoritmo como es K-means para combinarlo junto con el algoritmo C.Max-Flow. Así pues, pienso que se podrían investigar los resultados que aportaría el uso combinado de otros algoritmos de clústering con C.Max-Flow.
3. Prueba del algoritmo en imágenes reales. Se ha comprobado como este algoritmo es capaz de obtener una gran cantidad de detalles reduciendo la carga computacional notablemente. Es por ello, que pienso se podría probar el utilizar este algoritmo en la segmentación de imágenes médicas o en técnicas de visión artificial.
4. Intentar mejorar las diferencias entre el uso de distancias. Este es un problema que ha existido en este proyecto debido al problema del *ground truth*. Sin embargo se ha podido comprobar como en un principio la distancia CIEDE00 es la mejor a la hora de encontrar detalles. Por ello, se propone que se siga investigando si es realmente esa distancia la que mejor resultados reporta, tal y como aquí se propone.

REFERENCIAS

- [1] CIE, *International Lighting Vocabulary*, CIE. Publication, nº 17(E.11).Paris, Bureau Central De La CIE, 1987.
- [2] CIE, *TermList*, <http://eilmv.cie.co.at/>
- [3] Sabine Süsstrunk, Robert Buckley, Steve Swen, *Standard RGB Color Spaces, Laboratory of audio-visual Communication (EPFL)*, Xerox Architecture Center, Apple Computer Lausanne, Switzerland; Webster, NY; Cupertino, CA.
- [4] CIE, *17-157 CIE 1976 L*a*b* colour space*.
- [5] Joaquín Campos Acosta, *La Especificación del color:Espacios de representación del color*, Serrano,144.28006 Madrid.
- [6] K.Muthukannan, P.Latha, *Clustering Techniques based Crops Image Segmentation*, International Conference on Recent Trends in Computational Methods, Communication and Controls. India, 2012
- [7] ZhongxiangZhu ,Bo Zhao ,Enrong Mao and Zhenghe Song, *Image segmentation based on Ant colony optimization and K-means clustering*, Proceedings of the IEEE international conference on Automation and logistics, Jinan,China, August 18-21,2007.
- [8] Greig, D., Porteous, B. y Seheult, A. *Exact maximum a posteriori estimation for binary images*.Journal of the Royal Statistical Society. Series B (Methodological), 51(2)271-279,1989.
- [9] María del Carmen Tobar Puente, *Optimización de una energía mediante corte de grafos*. Tesis Doctoral, Universidad Politécnica de Madrid, 2014
- [10] Yuri Y. Boykov, Marie-Pierre Jolly, *Interactive Graph Cuts for Optimal Boundary & Region Segmentation of Objects in N-D Images*, En Proceedings, Eighth IEEE International Conference on Computer Vision, (ICCV'01),tomo1, páginas 105-112, IEEE, 2001
- [11] Ford , L. y Fulkerson,D. *Flows in networks*,Princeton Univ. Press, 1962.
- [12] Jing Yuan, Egil Bae, Xue-Cheng Tai y Yuri Boykov, *A study on continuous max-flow and min-cut approaches*, CVPR, San Francisco, 2010.
- [13] Base de datos de Berkeley, <http://www.eecs.berkeley.edu/Research/Projects/CS/vision/grouping/resources.html>
- [14] Aurora Sáez, Carmen Serrano, Begoña Acha ,*Normalized Cut Optimization based on color perception findings. A comparative study*, 31 Julio 2014.
- [15] CIE, *Technical Report Colorimetry*,3rd Edition, 2004.

-
- [16] M.R Luo, G. Cui y B.Rigg, *The Development of the CIE 2000 Color Difference Formula*. CIEDE200, Color Research and Applications, Vol 26, N°5, pag 340-350, 2001.
- [17] Gaurav Sharma, WenCheng Wu y Edul N.Dalai, *The CIEDE200 Color-Difference Formula, Implementation Notes, Supplementary Test Data, and Mathematical Observations*,2004.
- [18] CIE, *Industrial colour-difference evaluation Technical Report CIE 166-1995*, Central Bureau of CIE, ISBN 3-900-734-60-7,Viena, Austria, 1995.
- [19] Yuri Boykov and Vladimir Kolmogorov, *An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision*. *IEEE Transactions on Patter Analysis and Machine Intelligence*, 26:359– 374, 2001.
- [20] Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23:1222 – 1239, 2001.
- [21] Gil. Strang, *Maximal flow through a domain*. *Mathematical Programming*, 26:123–143, 1983. □
- [22] Gil. Strang. *Maximum flows and minimum cuts in the plane*. *Advances in Mechanics and Mathematics*, III:1–11, 2008. □
- [23] Tony F. Chan, Selim Esedoglu, and Mila Nikolova. *Algorithms for finding global minimizers of image segmentation and denoising models*. *SIAM J. Appl. Math.*, 66(5):1632–1648 (electronic), 2006
- [24] Jing Yuan, Egil Bae, Xue-Cheng Tai y Yuri Boykov, *A Continuous Max-Flow Approach to Potts Model*
- [25] Link del código de K-means, http://www.princeton.edu/~kung/ele571/571-MatLab/571BP_Chad/kmeans.m
- [26] Link del código de Max-Flow, <http://www.mathworks.com/matlabcentral/fileexchange/34224-fast-continuous-max-flow-algorithm-to-2d-3d-multi-region-image-segmentation>
- [27] Ben Appleton and Hugues Talbot. *Globally optimal surfaces by continuous maximal flows*. In DICTA, pages 987-996, 2003
- [28] Egil Bae and Xue-Cheng Tai. *Efficient global minimization for the multiphase Chan-Vese model of image segmentation*. In Energy Minimization Methods in Computer Vision and Pattern Recognition (EMMCVPR), pages 28–41, 2009.
- [29] Vladimir Kolmogorov and Ramin Zabih. *Multi-camera scene reconstruction via graph cuts*. In European Conference on Computer Vision, pages 82–96, 2002.
- [30] Victor Lempitsky and Yuri Boykov. *Global optimization for shape fitting*. In CVPR, 2007. 6, 2007.
- [31] Victor S. Lempitsky, Yuri Boykov, and Denis V. Ivanov. *Oriented visibility for multiview reconstruction*. In ECCV’06, pages 226–238, 2006.
- [32] George Vogiatzis, Carlos H. Esteban, Philip H. Torr, and Roberto Cipolla. *Multi-view stereo via volumetric graph-cuts and occlusion robust photo-consistency*. PAMI, 29(12):2241–2246, 2007.

- [33] Vivek Kwatra, Arno Schoedl, Irfan Essa, Greg Turk, and Aaron Bobick. *Graphcut textures: Image and video synthesis using graph cuts*. ACM Transactions on Graphics, SIGGRAPH 2003, 22(3):277–286, July 2003.
- [34] Ben Appleton and Hugues Talbot. *Globally minimal surfaces by continuous maximal flows*. IEEE Trans. Pattern Anal. Mach. Intell., 28(1):106–118, 2006.
- [35] Programa Gimp. <http://gimp.es>