DESARROLLO DE MÉTODOS DE BÚSQUEDA EN BD DE IMÁGENES CON JAVAVIS

AUTORA
MARTA FERNÁNDEZ DÍAZ

TUTOR
MIGUEL ANGEL CAZORLA QUEVEDO

DEPARTAMENTO
CIENCIA DE LA COMPUTACIÓN E INTELIGENCIA ARTIFICIAL

CURSO
2004-2005

Índice

1. Introducción

JavaVis es una librería de visión computacional en Java, la cual usa una interfaz gráfica con el fin de mostrar los resultados de aplicar diferentes métodos y algoritmos de visión artificial.

El origen de esta librería surge siguiendo la filosofía de trabajo de la librería Vista, la cual aunque estaba escrita en C estándar pretendía simular la programación orientada a objetos. JavaVis aunque sigue esta filosofía orientada a objetos se ha implementado sobre Java, es una librería que sigue evolucionando desarrollándose en un proyecto introducido en SourceForge donde podemos encontrar toda la documentación sobre la misma (http://javavis.sourceforge.net/index.html).

Partiendo de esta base se han implementado y estudiado diversos métodos de búsqueda en bases de datos de imágenes.

2. Modelos de color

Inicialmente se ha partido del estudio de los distintos tipos de imágenes planteados como son el RGB, el YCbCr y el HSI.

2.1 Modelo de color RGB

Nuestros ojos perciben el color a través de la simulación de tres pigmentos visuales en los conos de la retina, comparando las intensidades en una fuente de luz, se percibe el color de la luz; esta teoría es la base para desplegar la salida de color en un monitor de video mediante el uso de los tres colores primarios, conocido como *modelo de color RGB*.

Cada color aparece en sus componentes primarias espectrales rojo, verde y azul. El modelo está basado en un sistema de coordenadas cartesiano, tal y como podemos ver en la figura 1.

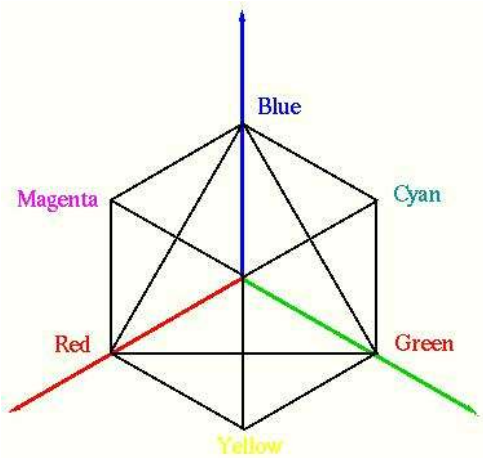

Figura 1. Cubo de color

Todos los colores han sido normalizados de forma que el cubo es unitario, es decir, se supone que todos los valores de rojo, verde y azul están en el rango [0, 1].

Este esquema de color es un modelo aditivo: las intensidades de los colores primarios se suman para producir otros colores. Cada punto de color contenido en los límites del cubo puede representarse como la tríada (R,G,B).

Las imágenes en el modelo de color RGB están formadas por tres planos de imágenes independientes, cada una de los colores primarios. Cuando se introducen en un monitor RGB, las tres imágenes se combinan en la pantalla de fósforo para producir una imagen de color compuesta. Por tanto, se utiliza el modelo RGB para el procesamiento de imágenes si las imágenes se expresan en términos de los tres planos de colores. La mayoría de las cámaras en

color que se usan para adquirir imágenes digitales utilizan el formato RGB.

A continuación se muestra una imagen en color de verduras y frutas junto con sus tresbandas de color rojo, verde y azul. El mayor nivel de gris en cada banda corresponde con el color predominante en cada objeto (fruta o verdura).

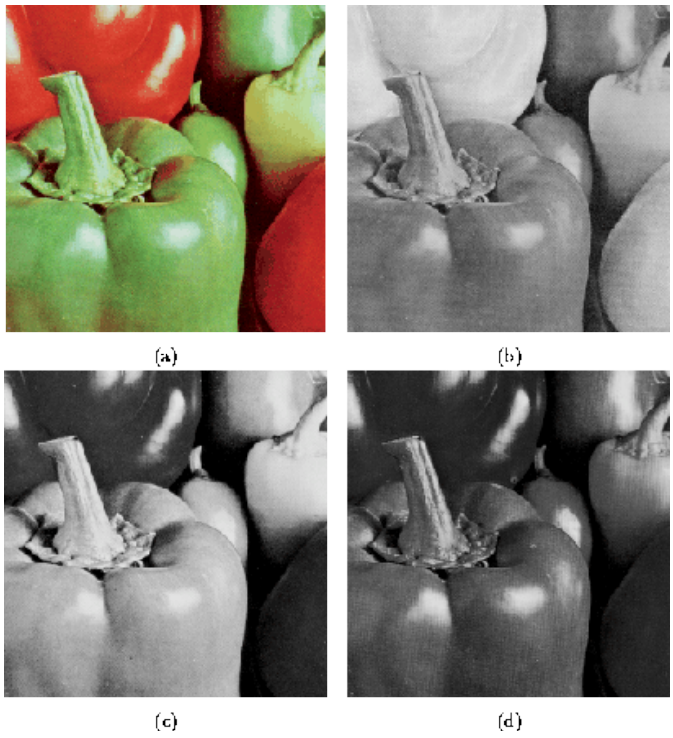

Figura 2. a)imagen original en color b)banda roja c)banda verde d)banda azul

Sin embargo este modelo de color no siempre es el mejor para el procesamiento de imágenes, puesto que no describe de forma eficiente las tres componentes de color (R,G,B). El igual ancho de banda da lugar a la misma profundidad del píxel y resolución del dispositivo para cada componente y esto es debido a que la sensibilidad de la componente de color del ojo humano es menor que la de la luminancia.

Por estas razones, muchos métodos de codificación de imágenes y sistemas de transmisión utilizan la luminancia y dos diferencias de colores, de ahí el uso del formato YCbCr.

2.2 Modelo de color YcbCr

YCbCr es acrónimo de luminancia y dos cromancias, también se conoce como datos YUV. El ojo humano es mucho más sensible a la información sobre luminancia que a las crominancias, y si se separan es posible comprimir más estás últimas sin mermar en la calidad.

2.3 Modelo de color HSI

HSI es acrónimo de Hue, Saturation, Intensity o lo que es lo mismo Tono, Saturación e Intensidad. El tono es un atributo cromático que describe la pureza de un color (amarillo puro, naranja puro, etc), mientras la saturación proporciona una medida del grado en que un color puro esta diluído en luz blanca. La utilidad del modelo de color HSI se debe principalmente a estas dos características:

- La componente de intensidad se separa de la información de color

- Las otras dos componentes están muy relacionadas con la forma en la que el ser humano percibe el color.

Gracias a estas propiedades este modelo es muy apropiado para el desarrollo de algoritmos de procesamiento de imágenes basados en propiedades del sistema de visión humano. En la siguiente figura podemos ver como se representa este modelo de color.

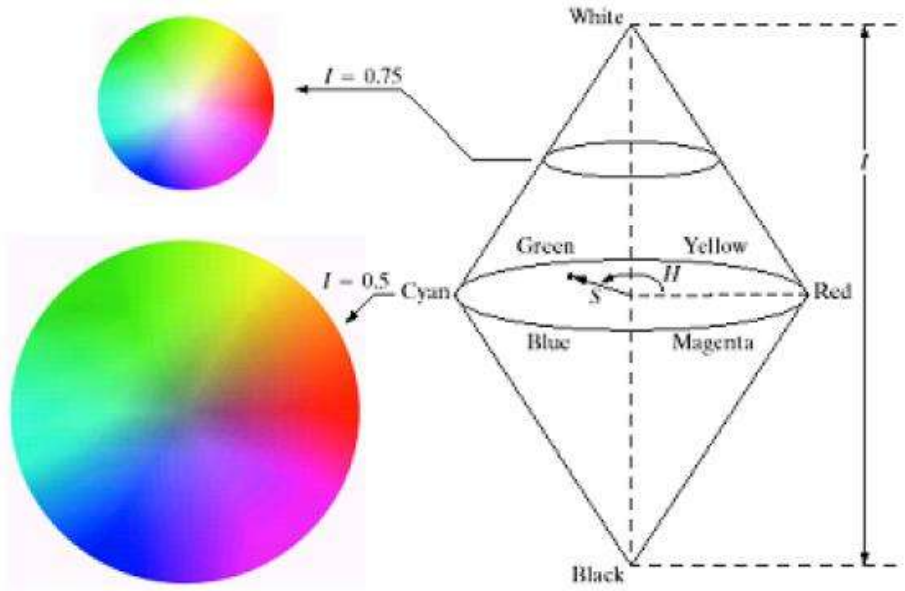

Figura 3. Representación del formato HSI

A lo largo del proyecto se ha trabajado con estos tres tipos de imágenes, aunque inicialmente todas las imágenes parten del formato RGB, por eso se han creado las funciones FRGBToHSI y FRGBToYCbCr.

2.4 FRGBToHSI

Esta función lo que hace es transformar una imagen del modelo de color RGB al modelo de color HSI tras aplicar la siguiente transformación:

$$H = \begin{cases} \theta & \text{if } B \le G \\ 360 - \theta & \text{if } B > G \end{cases}$$

$$\theta = \cos^{-1}\left\{\frac{\frac{1}{2}\left[(R-G)+(R-B)\right]}{\left[(R-G)^2+(R-B)(G-B)\right]^{1/2}}\right\}. \qquad \textit{Ecuación 1}$$

$$S = 1 - \frac{3}{(R+G+B)}\left[\min(R,G,B)\right].$$

$$I = \frac{1}{3}(R+G+B).$$

Podemos observar una prueba de funcionamiento de esta función en la siguiente página, la figura 4 nos muestra la imagen original en RGB, y en la figura 5 podemos ver claramente las tres bandas, el tono (H), la intensidad (I) y la saturación (S).

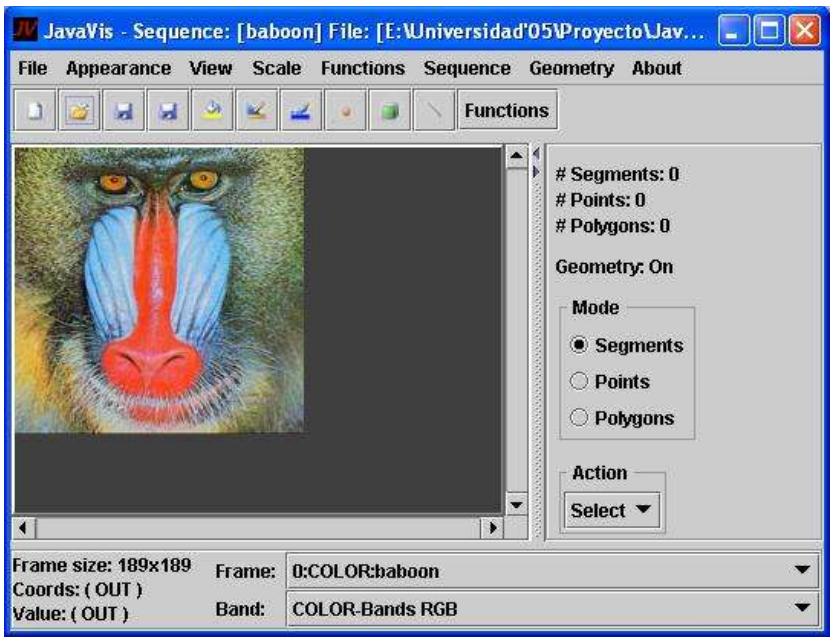

Figura 4. Imagen RGB

Figura 5. Imagen HSI

2.5 FRGBToYCbCr

Esta función es similar a la anterior, pero en este caso transforma la imagen RGB en su homóloga en YcbCr, para ello se ha aplicado la siguiente transformación:

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} + \begin{bmatrix} 65{,}481 & 128{,}553 & 24{,}966 \\ -37{,}797 & -74{,}203 & 112 \\ 112{,}0 & -93{,}786 & -18{,}214 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix}$$

Ecuación 2

En la siguiente figura podemos observar una prueba de funcionamiento de esta función teniendo la misma imagen RGB que en el caso de la transformación a HSI. Podemos diferenciar las tres bandas claramente y comprobamos que la intensidad tanto en el modelo de color anterior como en el actual coinciden.

Figura 6. Imagen YCbCr

3. Métodos de búsqueda en BD de imágenes

Los métodos de búsqueda de imágenes en bases de datos se basan en los histogramas de las imágenes, es decir, se calculan los histogramas de todas las imágenes, tanto las que tenemos de prueba como de la propia base de datos, ese es el principio de los métodos de búsqueda. Debido al que se trabaja con histrogramas se ha escogido cuatro medidas que calculan la distancia entre histogramas, las dos primeras son distancias heurísticas, L1 y L2 (p = 1 y p = 2)

- *Minkowski-form distance Lp*

$$D(X, Y) = (\sum i \mid f(i;X) - f(i;Y) \mid p)\ 1/p \qquad 1 \le p \le \infty \qquad \textit{Ecuación 3}$$

Las dos siguientes distancias se basan en una alternativa bastante interesante que son las divergencias de información teóricas.

- *Kullback-Leibler divergence*

$$D(X, Y) = \sum i\ f(i;X) \log (f(i;X)/f(i;Y)) \qquad \textit{Ecuación 4}$$

- *Jeffrey-divergence*

$$D(X, Y) = \sum i\ f(i;X) \log (f(i;X)/f(i;Y)) + f(i;Y) \log (f(i;Y)/f(i;X)) \qquad \textit{Ecuación 5}$$

3.1 Fase 1: Comparando distancias

En esta fase lo que se ha hecho ha sido implementar la función *FCompareImages*, a esta función le pasamos como parámetro una base de datos de imágenes, una carpeta con imágenes de prueba y un modelo de color, y, se encarga de buscar la imagen que más se parece a cada una de las imágenes de prueba. Para ello utiliza como funciones auxiliares *FCalcHistoColor* – para calcular el histograma de cada imagen de prueba -; *FCalcHistoBD* – para calcular los histogramas de la base de datos de las imágenes -; *FSearchImage* – para buscar cogiendo el histograma de la imagen y los histogramas de la BD la distancia mínima entre histogramas.

Estas funciones auxiliares necesitan de más parámetros de entrada, cómo:

- el valor de discretización – número de bins -
- el porcentaje – variación con respecto a la distancia mínima -
- tipo de imagen

- tipo de distancia calculada

Por lo tanto para poder realizar una comparación en cuanto a las diferentes distancias, se ha ido probando con un porcentaje constante del 10%, con un valor de discretización variable desde 5 hasta 30 bins, con cada uno de los modelos de color, los resultados obtenidos los podemos ver reflejados en las siguientes gráficas:

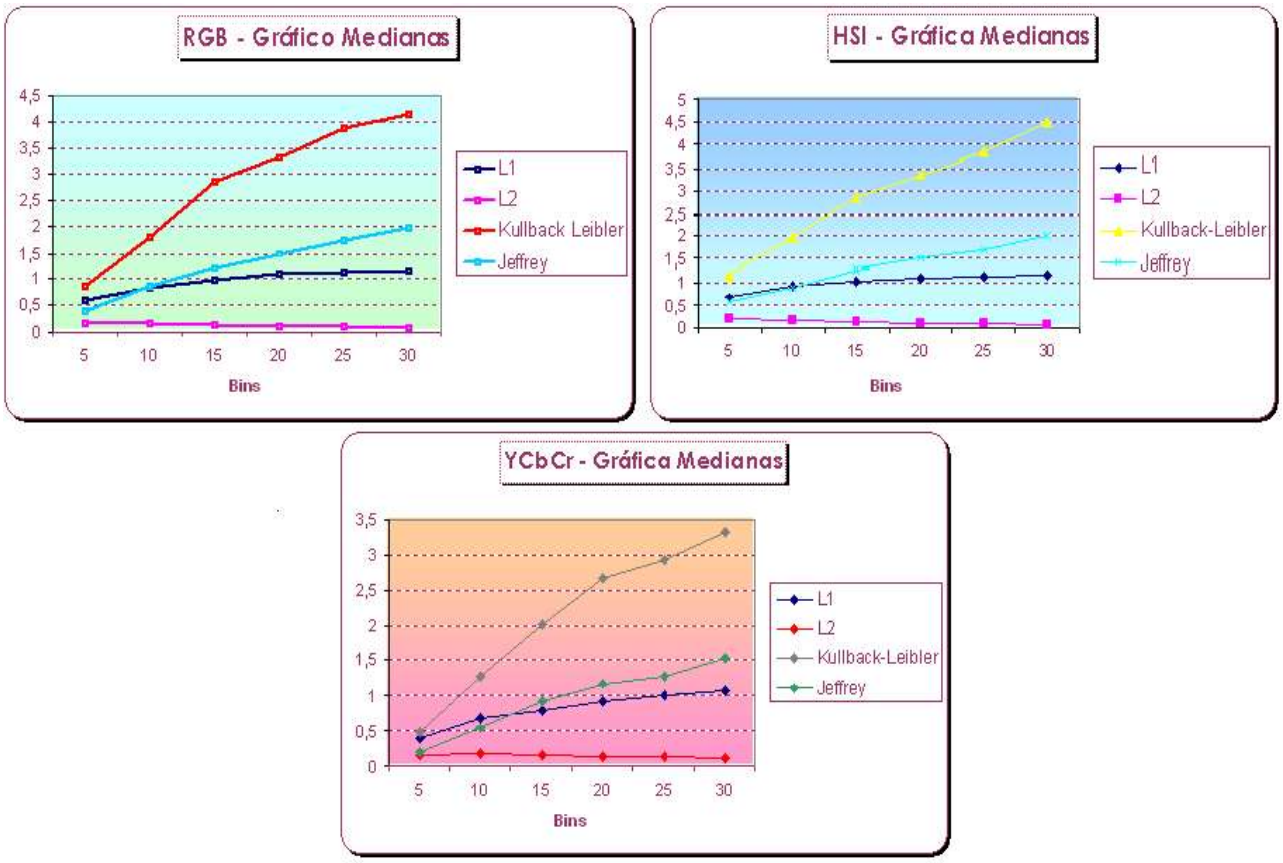

Figura 7. a) gráfica medianas formato RGB b) gráfica medianas formato HSI c) gráfica medianas formato YCbCr

Los resultados obtenidos para los tres modelos de color tienen unas gráficas muy similares, se puede ver claramente que la norma L2 es la que encuentra las distancias con una valor muy pequeño y no existe una gran diferencia entre las distancias calculadas, al aumentar el número de bins las distancias se van reduciendo. La norma L1 está dentro de un rango no muy amplio de valores, pero al ir aumentando el valor de discretización, las distancias van aumentando ligeramente.

Sin embargo, al calcular las divergencias, se observa que Kullback-Leibler empieza con valores pequeños y aumenta mucho tal cual se va incrementando el número de bins, tiene un crecimiento prácticamente exponencial, por lo que esta distancia no nos sirve muy bien para buscar la imagen más parecida. Por otro lado, la divergencia de Jeffrey, aumenta al ir aumentando los bins, pero su crecimiento en cuanto a las distancias calculadas va más o menos a la par que la norma L1.

En la figura 8 tenemos las gráficas de las varianzas de los cálculos realizados, podemos observar como están de dispersos los datos, hay que indicar que tanto al aplicar la norma L1 como la norma L2 se obtienen unas distancias en un rango muy pequeño, por eso los datos apenas están dispersos, son próximos a cero.

También observamos que con la divergencia de Kullback-Leibler los datos están muy dispersos, el rango de distancias es muy grande, varían entre 0 y 9, se confirma que las

imágenes que podemos obtener con esta distancia no van a ser las mismas que si aplicamos la norma L2. Por otro lado, la divergencia de Jeffrey tiene un rango de dispersión variable entre 0 y 2, no es muy amplio, pero sí más amplio que las normas de Minkowski, por lo que se puede decir que ante estos resultados es mejor utilizar las normas L1 y L2 frente a las divergencias con un número de bins de 20.

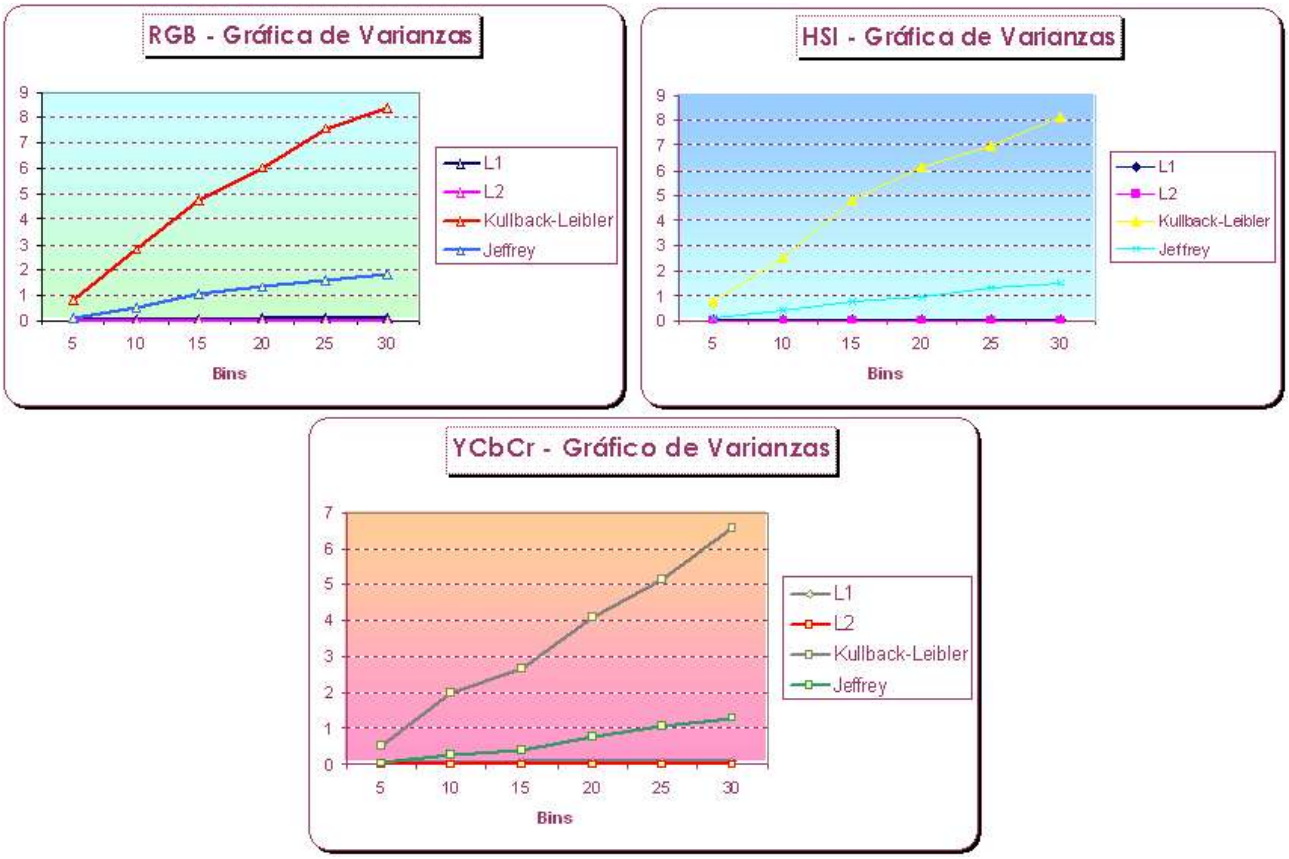

Figura 8. a) gráfica varianzas formato RGB b) gráfica varianzas formato HSI c) gráfica varianzas formato YCbCr

3.2 Fase 2: Agrupando imágenes

En esta segunda fase se ha implementado FGroupImages, esta función también utiliza las funciones auxiliares FCalcHistoBD e FCalcHistoColor. Tiene como parámetros de entrada:

- formato de la imagen: lo probamos con los tres formatos: RGB, HSI, YCbCr
- distancias: L1, L2, Kullback-Leibler, Jeffrey
- directorio con las imágenes a procesar
- número de bins: va a ser constante, vamos a usar 20
- porcentaje: 10%

El objetivo de esta parte es observar como las distancias agrupan las imágenes, es decir, tenemos un número de imagenes, por ejemplo 180, pues vamos una por una comparándolas con todo el grupo de imágenes, incluso consigo misma, y obtenemos las distancias de cada una de esas comparaciones en función de los parámetro de entrada ya comentados. Con estos resultados lo que hacemos es crear una imagen que nos pinte un punto por cada distancia, de forma que un punto negro nos indicará que es más próximo a cero (distancia mínima) y un punto blanco más lejando a cero.

Para realizar las pruebas se ha escogido una base de datos de 180 imágenes ya agrupadas en subcarpetas con el fin de observar si coinciden los grupos obtenidos con los que ya tenemos, en total tenemos 12 grupos. En la siguiente serie de figuras podemos comprobar los resultados obtenidos al aplicar la función FcompareImages:

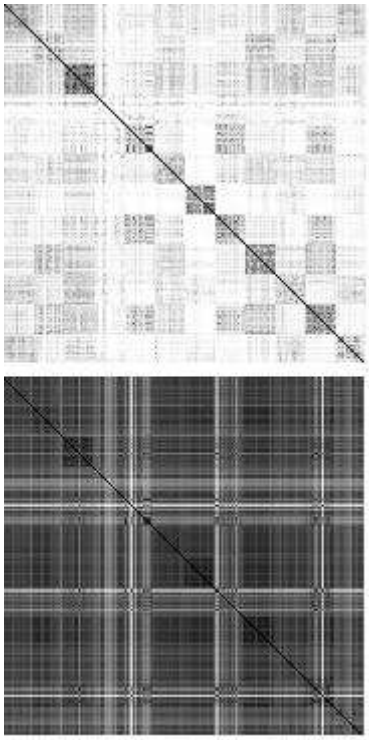

Figura 9. a) Formato RGB, norma L1

b) Formato RGB, norma L2

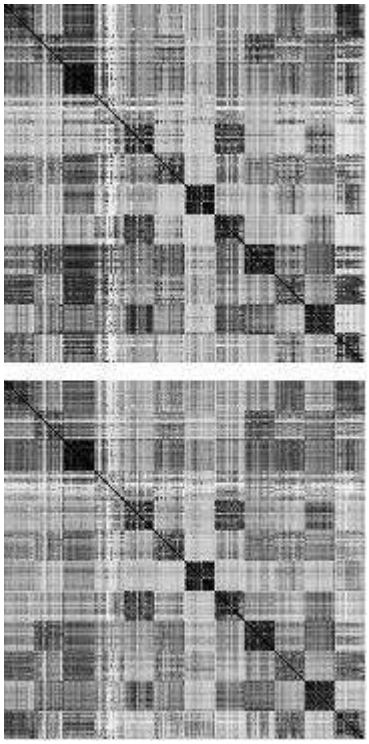

Figura 10. a) Formato RGB, Kullback-Leibler

b) Formato RGB, Jeffrey

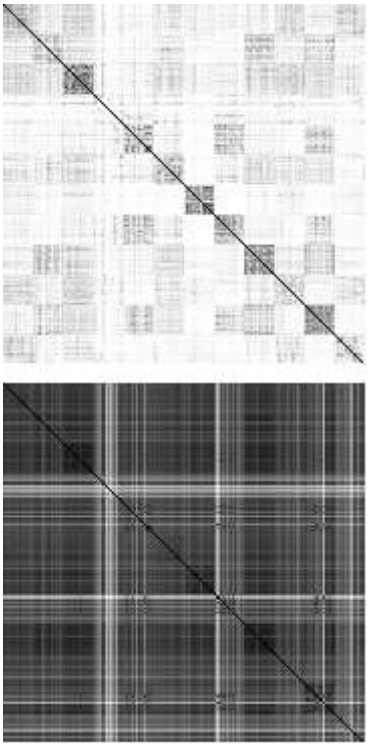

Figura 11. a) Formato HSI, norma L1

b) Formato HSI, norma L2

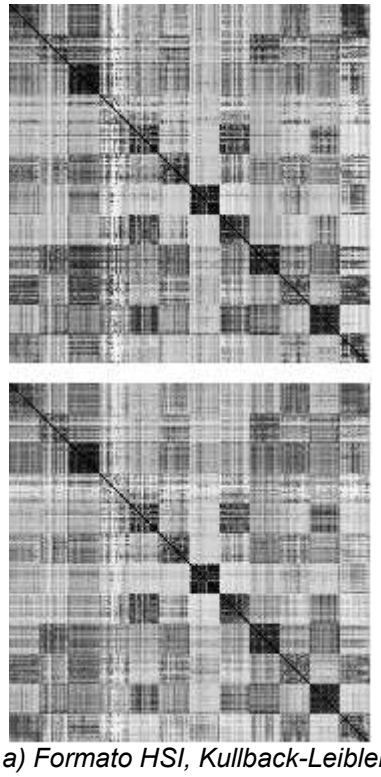

Figura 12. a) Formato HSI, Kullback-Leibler

b) Formato HSI, Jeffrey

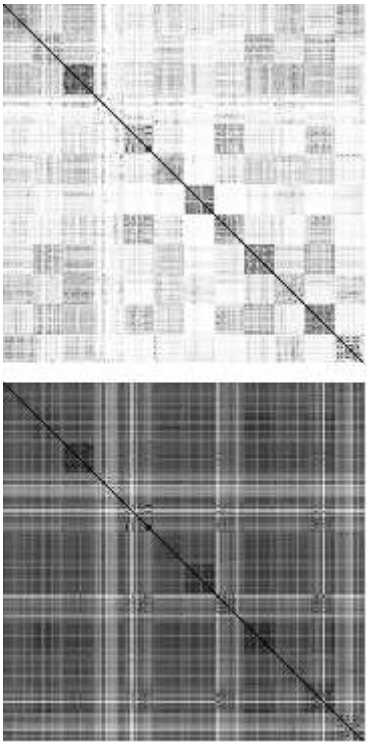

Figura 13. a) Formato YCbCr, norma L1

b) Formato YCbCr, norma L2

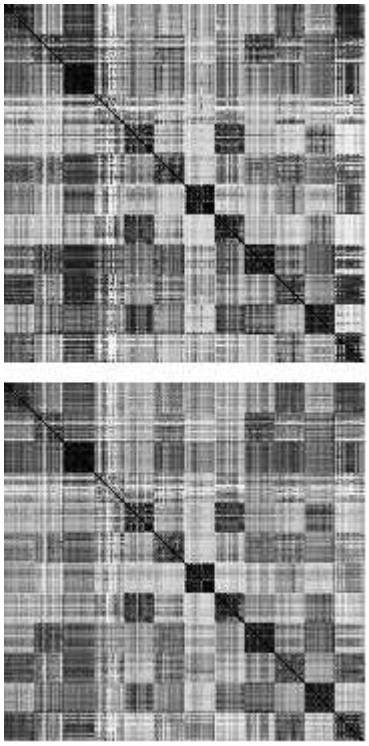

Figura 14. a) Formato YCbCr, Kullback-Leibler

b) Formato YCbCr, Jeffrey

Las imágenes nos indican con su diagonal pintada en negro, la distancia mínima obtenida entre una imagen consigo misma, y alrededor de esta diagonal nos encontramos con los grupos que forman la base de datos de imágenes, en total los 12 grupos que esperábamos encontrar. El resto de agrupaciones que se encuentran fuera de la diagonal no nos aportan ningún tipo de información con respecto a los grupos, podemos encontrar similitudes o totales diferencias, pero no nos interesa para este tipo de pruebas.

Se observa que los resultados obtenidos no varían mucho en cuanto al modelo de color usado. En este caso las divergencias obtienen resultados muy similares, pero con las distancias de Minkowski no ocurre lo mismo. La norma L2 que nos daba resultados muy buenos en la fase anterior, en este caso no nos sirve, esto es debido a que al obtener valores tan pequeños (entre cero y uno) y tener que realizar la raíz cuadrada, se hacen más pequeños y no se llegan a poder distinguir, aunque se pueden discernir algunos grupos en la diagonal, no nos aporta información clara como sí que lo hacen el resto de distancias.

3.3 Fase 3: Algoritmo Kmedias

En esta últimal fase, se ha implementado el algoritmo Kmedias. Este algoritmo consiste en agrupar las imágenes en un número k de grupos. Inicialmente se escogen estos centros del grupo al azar, entre todas las imágenes de la base de datos. En pasos posteriores se van recalculando estos centros de forma que los nuevos centros serán los histograma medios de cada grupo. El objetivo es encontrar la mejor agrupación, para ello no puede haber ningún grupo con una sola imagen (que sería el centro), la forma de implementarlo ha sido siguiendo los pasos explicados a continuación:

i. Escoge k histogramas de imágenes al azar para actuar como centros

ii. Hasta que los centros permanezcan constantes (no hay cambios en los grupos entre dos iteraciones)

    1. Se asocia cada imagen al centro en función de la distancia mínima

2. .Se calculan los nuevos centros, que serán los histogramas medios de cada grupo

Como se ha comentado en las fases anteriores, en esta también se utilizan las funciones auxiliares FCalcHistoBD y FcalcHistoColor. Los parámetros de entrada de Kmedias son:

- el directorio con las imágenes a agrupar
- número de grupos (K)
- el tipo de imagen
- la distancia (L1, L2, Kullback-Leibler y Jeffrey)
- un boleano que nos indica si ya tenemos los histogramas calculados o tenemos que calcularlos
- otro booleano con el que le indicamos que pare de iterar cuando encuentre un resultado

Las dificultades de este algoritmo son varias, una de ellas es saber escoger los histogramas que van a actuar como centros base, puesto que si se escogen imágenes pertenecientes al mismo grupo, el resultado no va a ser bueno. Y la otra dificultad es escoger el número k de grupos óptimo, para que se creen los grupos necesarios. Por eso podemos decir que se obtienen resultados bueno o resultados aceptables, en función de este tipo de elecciones. A continuación vamos a ver varias pruebas de funcionamiento de este algoritmo.

En esta primera prueba tenemos originalmente estas imágenes:

Figura 15

Tras aplicar Kmedias con k = 2, la distancia L1 y el formato RGB con los resultados obtenidos los podemos observar en la siguiente figura, en la cual se han marcado los dos centros iniciales para la agrupación (con un punto de cada color), como se puede observar ambas imágenes pertenecen al mismo grupo, por lo que los resultados que vamos a obtener serán o aceptables o incorrectos, variará en función de la cantidad de imágenes que se puedan agrupar a dichos centros. En esta primera prueba los resultados son aceptables, puesto que hay un margen de error de una imagen en cada grupo. Además se observa que las distancia tienen valores en un rango pequeño.

30065.JPG Group: 2 Dist: 1.897
30069.JPG Group: 2 Dist: 1.845
30081.JPG Group: 1 Dist: 1.813
30083.JPG Group: 2 Dist: 1.642
345028.JPG Group: 1 Dist: 0.0
345030.JPG Group: 2 Dist: 0.0
345037.JPG Group: 1 Dist: 1.541
345042.JPG Group: 1 Dist: 1.444

Figura 16. Resultados prueba funcionamiento 1

En la siguiente prueba tenemos un mayor número de imágenes, un total de 12 y aplicamos Kmedias con k = 3, la distancia Kullback-Leibler y formato YCbCr en la figura siguiente podemos observar el resultado obtenido:

114017.JPG Group: 1 Dist: 5.418
114022.JPG Group: 1 Dist: 0.0
114047.JPG Group: 1 Dist: 1.925
114055.JPG Group: 1 Dist: 5.454
440005.JPG Group: 2 Dist: 7.914
440007.JPG Group: 2 Dist: 0.0
440012.JPG Group: 2 Dist: 1.409
440027.JPG Group: 2 Dist: 1.184
476002.JPG Group: 3 Dist: 1.815
476010.JPG Group: 3 Dist: 0.0
476016.JPG Group: 3 Dist: 2.627
476026.JPG Group: 3 Dist: 3.964

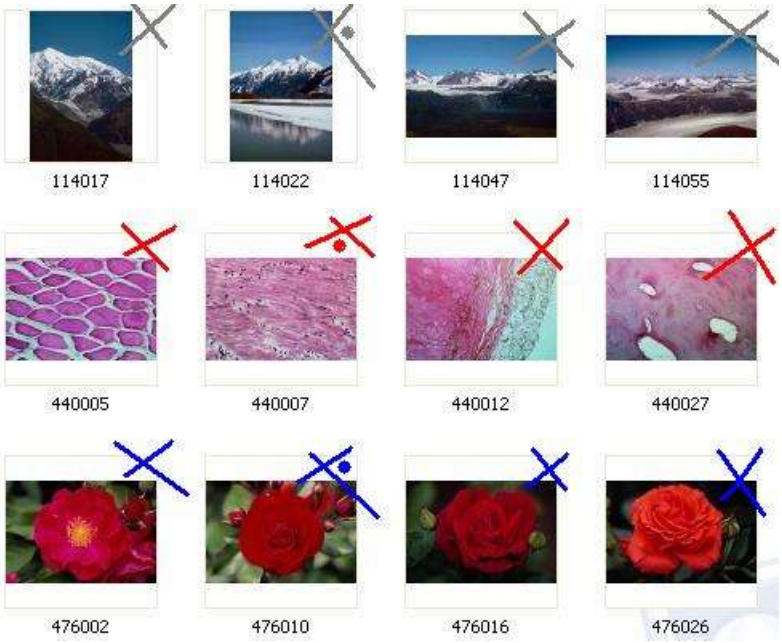

Figura 17. Resultados prueba funcionamiento 2

En este caso en concreto el resultado es correcto, debido a que los centros (señalados con un punto de cada color) se han escogido correctamente, cada uno pertenece a un grupo de imágenes. Podemos observar también las distancias pertenecientes a cada grupo, y vemos que el rango es mayor que en la prueba de funcionamiento anterior, esto es debido a que se ha escogido la divergencia de Kullback-Leibler, la cual tiene una gran dispersión cuanto mayor es el número de bins, y se está trabajando con un valor de discretización de 20, aún así agrupa correctamente.

En la tercera prueba de funcionamiento, se han indicado los mismos parámetros de entrada que en el caso anterior, 12 imágenes, k = 3, distancia Kullback-Leibler y formato YcbCr. El objetivo es demostrar que con los mismos parámetros de entrada se pueden obtener resultados distintos en función de los centros iniciales escogidos al azar, los resultados obtenidos se pueden ver en la figura 18.

114017.JPG Group: 1 Dist: 0.0
114022.JPG Group: 1 Dist: 2.326
114047.JPG Group: 1 Dist: 2.605
114055.JPG Group: 2 Dist: 0.0
440007.JPG Group: 2 Dist: 7.921
440027.JPG Group: 2 Dist: 7.143
440005.JPG Group: 2 Dist: 8.462
440012.JPG Group: 2 Dist: 7.405
476002.JPG Group: 3 Dist: 3.312
476010.JPG Group: 3 Dist: 3.066
476016.JPG Group: 3 Dist: 5.290
476026.JPG Group: 3 Dist: 0.0

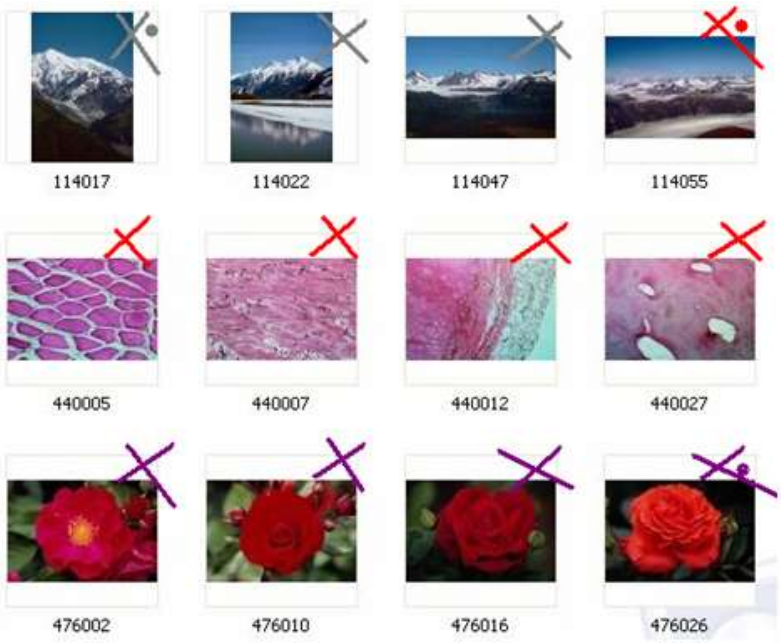

Figura 18. Resultados prueba funcionamiento 3

En esta ocasión han coincidido dos centros con dos imágenes pertenecientes al mismo grupo, y el tercer centro a una imagen perteneciente a otro grupo. Este último centro ha agrupado correctamente las imágenes, pero el resto de imágenes aunque no se han agrupado correctamente nos indican un resultado aceptable, puesto que únicamente una de ellas podríamos inidicar que no está bien agrupada, concretamente la 114055.JPG.

4. Conclusiones

Al trabajar con histogramas los resultados que se obtienen dependen de las gamas de colores de las imágenes, por lo que imágenes distintas con gamas parecidas nos pueden dar resultados erróneos o ruido en los mismos.

El formato de la imagen empleado no es crítico para los resultados, eso sí, han de estar normalizados dentro del rango [0..1].

Cuanto mayor es el número de bins mejor son los resultados, por eso se ha utilizado como valor de discretización 20.

La norma de Minkowski L2 sirve para buscar imágenes, puesto que se obtienen unas distancias pequeñas con muy poca dispersión en los datos, sin embargo cuando el objetivo es agrupar imágenes y debido a esta característica (valores mínimos) no nos sirve, puesto su rango es aún menor (por realizar la raíz cuadrada) y obtenemos resultados poco claros.

En el algoritmo de agrupación de las Kmedias es muy importante la elección correcta de los centros donde se tienen que agrupar las imágenes, puesto que una elección errónea nos puede conducir a resultados erróneos y como mucho aceptables. También es muy importante la elección de k, puesto que si se escoge un k muy grande o un k muy pequeño no se agrupará de forma correcta obteniendo por tanto resultados erróneos.

Después de haber realizado pruebas con las distintas distancias entre histogramas, se puede concluir indicando que se empleará una distancia u otra en función del objetivo a conseguir:

- Búsqueda de imágenes → Minkowski-form distance (L1 y L2)
- Agrupación de imágenes → Minkowski-form distance (L1), Kullback-Leibler divergence, Jeffrey divergence

5. Bibliografía

Computer Vision – A modern approach
David A. Forsyth, Jean Ponce. Editorial: Prentice Hall

Empirical Evaluation of Dissimilarity Measures for Color and Texture
Yossi Rubner, Jan Puzicha, Carlo Tomasi and Joachim M. Buhmann
Computer Vision and Image Understanding 84, 25-43 (2001)

Color Indexing
Michael J. Swain, Dana H. Ballard
International Journal of Computer Vision, 7:1, 11-32 (1991)

Manual JavaVis
http://javavis.sourceforge.net/index.html

Api Java 1.5
http://java.sun.com/j2se/1.5.0/docs/api/index.html