



Ingeniería Eléctrica
FACULTAD DE CIENCIAS
FÍSICAS Y MATEMÁTICAS
UNIVERSIDAD DE CHILE

Proyecto Final Área Imágenes: *Content Based Image Retrieval*

EL5206 – Laboratorio de Inteligencia Computacional y Robótica

Nombre:	Sebastián Parra
Profesor:	Claudio Pérez
Auxiliar:	Juan Pérez
Ayudante:	Leonardo Cament
Fecha:	17 de noviembre de 2018

Desarrollo del Proyecto

Para el desarrollo del sistema de búsqueda de imágenes similares basado en su contenido se utilizó principalmente la implementación en C++ de la librería OpenCV, la cual proporciona estructuras de datos y funciones que facilitan enormemente la realización de operaciones con imágenes, tales como la aplicación de filtros mediante convolución, o la extracción de histogramas de imágenes multicanal.

La arquitectura del buscador de imágenes se puede dividir en tres etapas principales, las cuales se tratarán en detalle a continuación.

Cálculo de vectores de características

La primera etapa del sistema desarrollado consistió en la extracción de vectores de características para cada una de las imágenes de la base de datos, junto con las imágenes de consulta a utilizar. En particular, en este proyecto se utilizaron histogramas de color en el espacio HSV como descriptores, cuya implementación consistió de los siguientes pasos:

1. Se transformó cada imagen al espacio HSV mediante la función `cvtColor()` de OpenCV.
2. Se dividió la imagen HSV en una cierta cantidad de regiones. Para esta parte, se implementaron 4 configuraciones distintas para la división: 9 celdas rectangulares de igual tamaño, 16 celdas rectangulares de igual tamaño, 36 celdas rectangulares de igual tamaño, o bien 5 regiones no-rectangulares distribuidas como se muestra en la figura 1.
3. Luego, se calcularon histogramas de color para cada una de las regiones obtenidas en el paso anterior, utilizando 8 bins para el canal H, 12 para el canal S, y 3 para el canal V. Esta sección se implementó utilizando la función `calcHist()` que provee OpenCV.
4. Finalmente, se normalizaron y concatenaron los histogramas obtenidos en cada región, obteniendo un vector de características de tamaño $23[\text{bins/región}] * (\# \text{ regiones})$. Este procedimiento se pudo implementar fácilmente mediante las funciones `normalize()` y `hconcat()` de OpenCV.



Figura 1: División de 5 regiones no-rectangulares usada en el proyecto.

Determinación de una Medida de Similitud

La segunda etapa del sistema consistió en la implementación de alguna medida de similitud entre histogramas, para así poder comparar el vector de características de la imagen de consulta, con los

vectores de la base de datos. Para el caso de este proyecto, se optó por utilizar la distancia Chi-cuadrado como métrica, cuya fórmula se puede apreciar en la ecuación 1.

$$d(H_1, H_2) = 2 \cdot \sum_I \frac{(H_1(I) - H_2(I))^2}{H_1(I) + H_2(I)}$$

Ecuación 1: Distancia Chi-cuadrado entre dos histogramas

Donde cabe notar que esta métrica posee una cota inferior en cero cuando los histogramas son idénticos, pero no posee una cota superior, por lo que estas mediciones sólo cobran sentido relevante cuando son comparadas entre sí.

En cuanto a la implementación de esta etapa, ésta fue relativamente sencilla gracias a la función `compareHist()` de OpenCV.

Ordenamiento de Resultados por Relevancia

Finalmente, la tercera y última etapa del sistema consistió en la implementación de un esquema de comparación de las imágenes de la base de datos, dado una imagen de consulta, para así luego poder ordenar los resultados por relevancia. Para esto, se optó simplemente por calcular la distancia entre la imagen de consulta y cada una de las imágenes de la base de datos, y posteriormente ordenar estas distancias de menor a mayor. Una vez terminada la parte anterior, la implementación de esta sección no fue compleja, dado que sólo se trató de ordenar un vector de menor a mayor, la cual es una operación realizable con librerías estándar de C++.

Obtención de Métricas de Desempeño

Una vez completado el sistema de búsqueda de imágenes, se aplicaron métricas de desempeño para evaluar su comportamiento. En el caso de este proyecto, se aplicaron las métricas de ranking y ranking normalizado, cuyas fórmulas se pueden apreciar en las ecuaciones 2 y 3.

$$Rank = \frac{1}{N_{rel}} \sum_{i=1}^{N_{rel}} R_i$$

Ecuación 2: Ranking de un sistema de búsqueda de imágenes basado en contenido

$$\widetilde{Rank} = \frac{1}{N \times N_{rel}} \left(\sum_{i=1}^{N_{rel}} R_i - \frac{N_{rel}(N_{rel} + 1)}{2} \right)$$

Ecuación 3: Ranking normalizado de un sistema de búsqueda de imágenes basado en contenido

Donde N_{Rel} corresponde al número de imágenes relevantes en la base de datos (o sea, el número de imágenes en este conjunto que son de la misma clase que la imagen de consulta), R_i corresponde al

ranking obtenido por la i -ésima imagen relevante en el vector de distancias, y N corresponde al número total de imágenes en la base de datos. Además, si se analiza con mayor detalle las ecuaciones, se puede notar que el Ranking es una métrica que no posee cota superior, por lo que escala según el tamaño de la base de datos, mientras que el Ranking Normalizado siempre tomará valores entre 0 (desempeño perfecto) y 1 (desempeño perfectamente erróneo).

Implementación de Otro Tipo de Características (Edge Histogram Descriptor)

Luego de obtener métricas de desempeño para el sistema basado en histogramas de color HSV, se procedió a investigar sobre otros tipos de caracterizaciones comúnmente utilizadas para tareas de búsqueda de imágenes basada en contenido, encontrando que además del descriptor basado en color que se utilizó, existen también algunos basados en texturas y formas ^[1], de los cuales se decidió implementar uno de los basados en texturas: El Descriptor de Histograma de Bordes. El funcionamiento de este descriptor se resume en los siguientes pasos ^[2]:

1. La imagen original se convierte a escala de grises (como este es un descriptor de texturas, el color no importa realmente)
2. La imagen monocromática se divide en 16 regiones rectangulares de igual tamaño (grilla de 4x4)
3. Cada región se subdivide en una matriz de $M \times N$ bloques, como se puede apreciar en la figura 2. La idea de este paso es que las dimensiones de esta matriz son constantes sin importar el tamaño de la imagen, por lo que las dimensiones de cada bloque se deben escalar de manera adecuada para hacer calzar los tamaños. Además, es necesario que cada una de las dimensiones de los bloques sea par (i.e. $2N \times 2M$), por lo que es posible que se tengan que ignorar algunos píxeles.

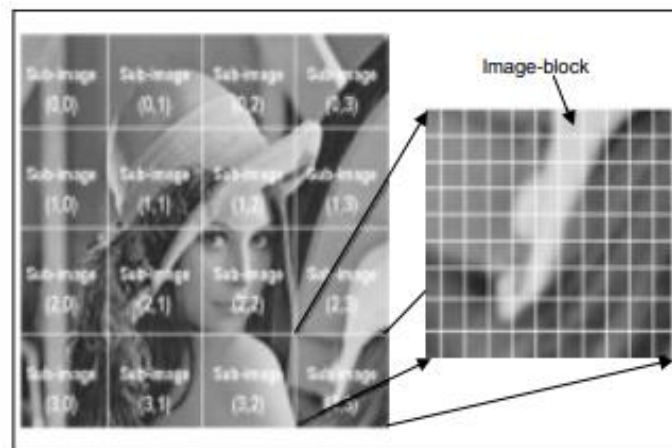


Figura 2: División de la imagen monocromática en 16 regiones y posterior subdivisión en una matriz de bloques.

4. Cada uno de los bloques se vuelve a dividir en 4 sub-bloques (grilla de 2x2), como se puede apreciar en la figura 3, donde el valor de cada sub-bloque será igual a la magnitud promedio de la imagen en esa sub-región.

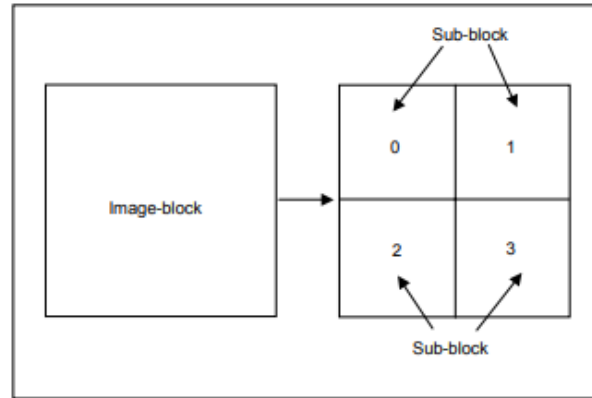


Figura 3: División de cada bloque en 4 sub-bloques de igual tamaño.

5. Se aplican 5 filtros distintos de 2x2 a cada matriz de sub-bloques: Uno para bordes horizontales, otro para verticales, otro para diagonales en 45°, otro para diagonales en 135°, y uno isotrópico (bordes no homogéneos), los cuales se muestran en la figura 4. Como las matrices de sub-bloques también son de 2x2, los resultados de cada filtro son un único número.

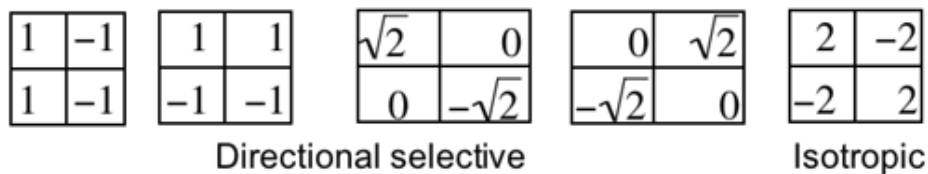


Figura 4: Filtros usados para la detección de bordes.

6. Para cada matriz de sub-bloques, se toman los 5 resultados obtenidos luego de aplicar cada filtro, se selecciona el mayor (o sea, el tipo de borde preponderante en el bloque), y si éste es mayor a un umbral T, entonces al bloque completo se le asigna un valor igual a un indicador del tipo de borde que predomina (0 si es horizontal, 1 si es vertical, 2 si es diagonal en 45°, etc.). En caso contrario, no se asigna ningún número al bloque (o bien, se le asigna un número que equivale a considerarlo no relevante)
7. Una vez obtenido el valor de todos los bloques dentro de una región, se calcula un histograma de 5 bins, donde cada uno representa cuántas veces ha predominado cada tipo de borde, como se puede observar en la figura 5. Como existen bloques que no son asignados a ninguna categoría, es posible que la suma de todos los bins del gráfico sea menor a M*N.

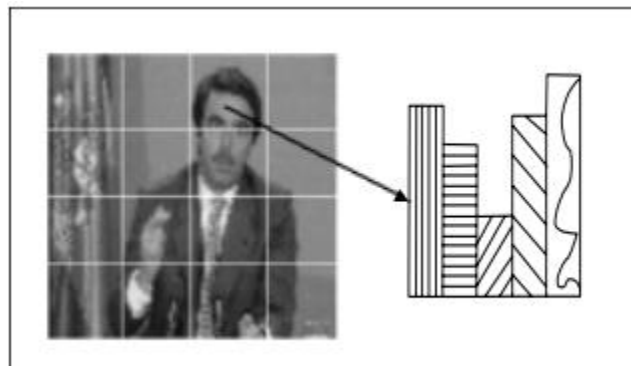


Figura 5: Obtención de histograma de preponderancia de tipo de borde.

8. Se repite este proceso para las 16 regiones y el vector de características resultante será la concatenación de todos los histogramas.

Teniendo este esquema en mente, se implementó este descriptor en el sistema de búsqueda utilizando una matriz de bloques de 15x15 y un umbral de 15, y se midieron las mismas métricas de desempeño, con el objetivo de comparar ambos métodos de extracción de características.

Resultados Obtenidos

Métricas de Desempeño Según Método de División de Imagen

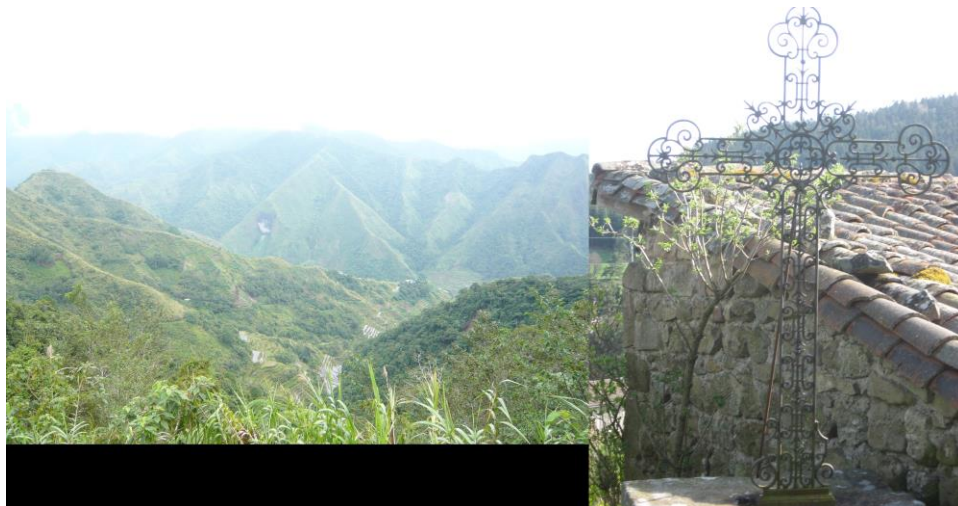
En la tabla 1 se pueden apreciar los valores obtenidos para el Ranking y el Ranking Normalizado, promediados sobre todas las imágenes de consulta, para los métodos de división de imagen de 9 rectángulos, 16 rectángulos, 36 rectángulos, 5 zonas no rectangulares, y usando el descriptor de texturas.

Tabla 1: Métricas de desempeño obtenidas para las variantes implementadas del sistema.

	Ranking Promedio	Ranking Normalizado Promedio
9 Rectángulos	15.1949	0.0328
16 Rectángulos	14.1294	0.0302
36 Rectángulos	14.2046	0.0303
5 Zonas no rectangulares	18.9081	0.0419
Descriptor de histograma de bordes	46.881	0.1106

Obtención de Resultados Cualitativos

En la figura 6 se muestran 10 pares de figuras, las cuales corresponden a los 10 primeros resultados obtenidos cuando la consulta fue la imagen de la izquierda, utilizando la variante del sistema que presentó el mejor ranking normalizado (16 rectángulos).







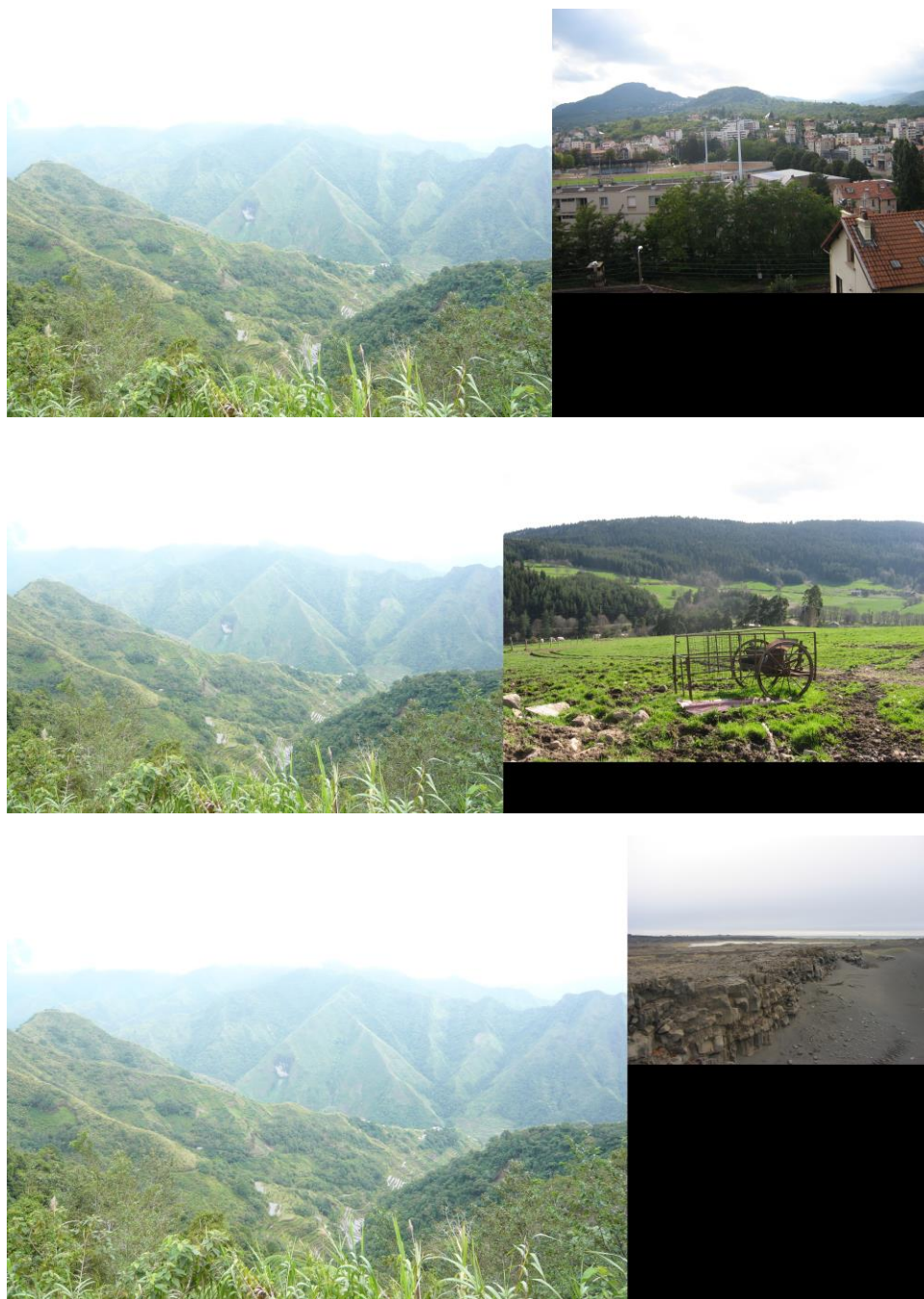


Figura 6: Imagen de consulta (izquierda) y sus 10 primeros resultados de la base de datos (derecha).

Análisis de los Resultados

Como se puede observar en la tabla 1, no sólo pareciera ser que el sistema presenta un mejor desempeño con descriptores de color al usar esta base de datos, sino que los resultados también indican que es más recomendable utilizar regiones rectangulares simétricas, lo cual parecier ser un poco contraintuitivo, dado que la región no rectangular que se utilizó se escogió precisamente porque debiese apuntar a una

separación más coherente de las regiones de la imagen, puesto que cuando se toma una foto, se tiende a enfocar el objeto de interés en el centro. Sin embargo, este fenómeno se puede explicar al notar que la mayoría de las clases de la base de datos que se ocupó consiste de paisajes, los cuales se debiesen ver más beneficiados de una separación de regiones rectangular, a diferencia de, por ejemplo, las imágenes de objetos concretos, los cuales tienden a estar enfocados en el centro de la figura.

Adicionalmente, al observar los resultados mostrados en la figura 6, se puede notar que, incluso los resultados que no pertenecer a la clase correcta, presentan un cierto grado de similitud en color con la imagen de consulta. En este contexto, se podría decir que el buscador tiene un buen desempeño, ya que si bien no siempre muestra en los primeros resultados el paisaje exacto que le es mostrado, siempre muestra ambientes muy similares, lo cual podría incluso ser la intención original del usuario (buscar imágenes parecidas, en vez de calces exactos). Sin embargo, por el otro lado, este descriptor puede traer problemas para búsquedas más específicas, como por ejemplo prendas de vestir, donde podría ocurrir que al consultar con un pantalón rojo, el buscador arroje resultados de prendas rojas en general, en vez de pantalones similares al mostrado. De esta manera, queda en evidencia las complicaciones que surgen al momento de evaluar el desempeño de un sistema de búsqueda de imágenes basado en contenido, puesto que existen muchos criterios distintos.

Referencias

- [1] A. Ahmed Mohamed, C. Abanti Makori and J. Kamau, "A Literature Survey of Image Descriptors in Content Based Image Retrieval", International Journal of Scientific & Engineering Research, vol. 7, no. 3, pp. 919-929, 2016. [Accessed 28 December 2018].
- [2] C. Won, D. Park and S. Park, "Efficient Use of MPEG-7 Edge Histogram Descriptor", ETRI Journal, vol. 24, no. 1, pp. 23-30, 2002. Available: 10.4218/etrij.02.0102.0103.