

Diseño, Implementación y Evaluación de algoritmos de procesamiento de imagen

Con el crecimiento constante de contenido multimedia en general y de imágenes en particular (web, ordenadores personales, etc.) se necesitan nuevas formas de gestionar este contenido. Este trabajo consiste en diseñar, implementar y evaluar algoritmos simples de “recuperación de imágenes basados en el contenido” (*Content-Based Image Retrieval* o CBIR en inglés).

El objetivo de estas técnicas es permitir la recuperación (o identificación) de imágenes similares a una imagen específica I_0 que juega el papel de pregunta. La estrategia consiste en prescindir de anotaciones manuales (costosas y lentas porque implican personas) para caracterizar el contenido y utilizar únicamente algoritmos automáticos de extracción de características visuales de las imágenes.

El primer algoritmo que vamos a desarrollar caracteriza las imágenes mediante histogramas. Lo vamos a utilizar para “indexar” una base de datos de 2000 imágenes. La base de datos proviene de U. Kentucky Benchmark Image Database. Referencia: D. Nistér and H. Stewénus. *Scalable recognition with a vocabulary tree*. In IEEE Conference on Computer Vision and Pattern Recognition (CVPR), volume 2, pages 2161-2168, June 2006. Contiene imágenes de 500 objetos o escenas. Cada objeto o escena está representado por 4 imágenes tomadas con varios puntos de vista e iluminaciones.

Ejemplo de objeto:



Ejemplo de escena:



Los ficheros de las imágenes están numerados de forma secuencial y las imágenes correspondientes a los mismos objetos o escenas aparecen de forma contigua en esta lista. La base de datos se puede descargar de [soft/Prog/UKentuckyDatabase.zip](#)

Breve descripción de la arquitectura del sistema

- El primer trabajo que tenemos que hacer es “**indexar**” la base de datos; es decir calcular y asignar a cada imagen su(s) descriptor(es). Este cálculo puede ser costoso debido al número de imágenes en la base de datos y se hace una única vez.
- En la fase de “**recuperación**” de contenido o “**búsqueda**”, el usuario presenta una imagen concreta (*query*) que puede pertenecer a la base de datos (o no). El algoritmo de búsqueda calcula el descriptor (o los descriptores) sobre la imagen *query* y calcula una medida de similitud entre el descriptor de la *query* y todos los descriptores de las imágenes de la base de datos.
- En un problema de clasificación, se definiría un valor de umbral sobre la medida de similitud para determinar cuál es el conjunto de imágenes de la base de datos que “se parecen” a la *query* del usuario. Pero en las aplicaciones de búsqueda de contenido en general y de imagen en particular (ver por ejemplo la búsqueda de imágenes en Google), se suele devolver al usuario una lista ordenada de las N imágenes más similares a la *query* (el número N depende en la práctica del tamaño y resolución de la pantalla del usuario).

Evaluación de la calidad de los algoritmos:

Para poder comparar la calidad de distintos algoritmos de indexación y búsqueda o para optimizar los parámetros de un algoritmo concreto, se necesita una evaluación objetiva de las prestaciones. Un enfoque clásico consiste en dibujar una curva **Precision** versus **Recall** calculada a partir de la lista ordenada.

Analicemos un ejemplo: Supongamos que la imagen *query* del usuario sea la siguiente:



Query

y que el sistema de recuperación de información devuelve la lista ordenada siguiente (10 imágenes):



Imagen 1



Imagen 2



Imagen 3



Imagen 4



Imagen 5



Imagen 6



Imagen 7

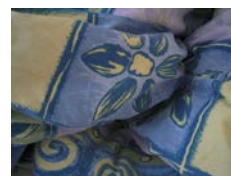


Imagen 8



Imagen 9



Imagen 10

En esta lista, encontramos tres aciertos en las posiciones 1, 2 y 4. Para calcular los valores de *Precision* y *Recall*, vamos a analizar esta lista con varios valores de N. “*Precision*” representa el número de aciertos respecto al número de imágenes en la lista, haciendo variar la longitud N de la lista de 1 a 10:

N	1	2	3	4	5	6	7	8	9	10
Aciertos	1	2	2	3	3	3	3	3	3	3
<i>Precision</i>	1	1	0,67	0,75	0,6	0,5	0,43	0,38	0,33	0,3

“*Recall*” es el número de aciertos respecto al número de imágenes del mismo objeto que están en la base de datos. En el caso de la base de datos de la Universidad de Kentucky, sabemos que cada objeto aparece exactamente 4 veces en la base de datos.

N	1	2	3	4	5	6	7	8	9	10
Aciertos	1	2	2	3	3	3	3	3	3	3
<i>Recall</i>	0,25	0,5	0,5	0,75	0,75	0,75	0,75	0,75	0,75	0,75

Con estos datos podemos dibujar una curva representando el valor del *Recall* por cada valor de *Precision*. Para tener resultados fiables, tenemos que promediar estos valores de *Precision* y *Recall* para muchas imágenes (idealmente todas las imágenes de la base de datos!). Nótese que los buenos sistemas son los que tienen una curva que se acerca al punto *Precision* → 1 y *Recall* → 1.

SISTEMA 1: Sistema de indexación de imagen con histogramas de niveles de gris.

El primer sistema que vamos a implementar consiste en indexar las imágenes con su histograma de nivel de gris (vamos a olvidarnos en un primer paso que las imágenes tienen información de color). Se pide:

1. Implementar un algoritmo (algo1) que indexe las imágenes de la base de datos con histogramas de niveles de gris. El parámetro más importante para este algoritmo es probablemente el número de *bins* utilizado en el histograma. Si es demasiado pequeño, el histograma no representará bien la imagen, pero si es muy grande, los costes de carga computacional y de almacenamiento serán altos.
2. Implementar una función (algo2) que mida la similitud entre los histogramas de niveles de gris de dos imágenes. Esta medida de similitud entre histogramas es una elección importante.
3. Implementar un algoritmo (algo3) que calcule el histograma de una imagen *query* propuesta por el usuario (como en 1) y mida la similitud entre dicho histograma y los histogramas de las imágenes de la base de datos (empleando 2). Finalmente, el algoritmo tiene que devolver una lista ordenada de las imágenes más similares a la imagen *query*.
4. Evaluar y optimizar el sistema mediante curvas de *Precision & Recall*. Para la optimización de los algoritmos se puede considerar que el mejor algoritmo es el que obtiene
 - a. las mejores curvas *Precision & Recall* (criterio de valoración 1)
 - b. con el tiempo de cálculo más pequeño posible (criterio 2)
 - c. y generando el menor volumen posible de descriptores (criterio 3)

Para que todos los grupos puedan comparar sus resultados, las curvas *Precision & Recall* se van a calcular promediando los resultados obtenidos con un subconjunto de imágenes *query* para la evaluación de los algoritmos que el profesor propondrá el día de entrega del trabajo. Un ejemplo de subconjunto de imágenes *query* podría estar formado por las 8 imágenes inferiores. En un fichero de texto *input.txt* se proporcionarán los nombres de las 8 imágenes propuestas para la evaluación, y el sistema implementado por cada grupo deberá leer dicho fichero, ejecutar el algoritmo 3 y proporcionar el resultado para cada *query* con la misma sintaxis que el fichero de ejemplo *output.txt*. Los ficheros ejemplo *input.txt* y *output.txt* están en *soft/Prog/example*



ukbench00004.jpg



ukbench00040.jpg



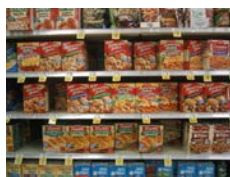
ukbench00060.jpg



ukbench00320.jpg



ukbench00588.jpg



ukbench00840.jpg



ukbench00996.jpg



ukbench01562.jpg

5. Finalmente, cada grupo deberá escribir un informe:
 - a. razonando y justificando las elecciones hechas
 - b. presentando los resultados obtenidos:
 - i. Lista ordenada de las imágenes recuperadas para las 8 imágenes pregunta
 - ii. Curva Precision & Recall (criterio 1)
 - iii. Tiempo de cálculo* en obtener resultados para una imagen *query* (criterio 2)
 - iv. Volumen (MB) de todos los descriptores de la base de datos (criterio 3)
 - c. analizando y comentando los ejemplos destacables de éxito y de fracaso
 - d. finalmente presentando ideas para mejorar el sistema

* Calculado en los ordenadores del laboratorio