

# Content Based Image Retrieval(CBIR) Aplicando descriptor ORB

Cabero, Gerardo Antonio, Barboza, Hector Emanuel  
{cabero,emabarboza}@exa.unsa.edu.ar

Departamento de Informatica (DI)- Facultad de Ciencias Exactas (FCE)-  
Universidad Nacional de Salta(UNSa)

**Abstract.** Por cada segundo se suben centenares de imágenes que hacen referencia a artículos para el consumo a través de sitios especializados de e-commerce como por ejemplo eBay o Alibaba. Las personas realizan búsquedas sobre artículos de consumo diario como ropa, zapatos, jeans, o cualquier tipo de prenda que se usa cotidianamente. Habitualmente los motores de búsqueda de imágenes de estas empresas y de otros motores como Google, Bing o DuckDuckGo, buscan imágenes basadas en textos que las etiquetan. Las consultas se realizan a través del ingreso de alguna palabra “clave”, y éste devuelve una serie de imágenes relacionadas a la misma. El objetivo del presente trabajo es crear un motor de búsqueda y recuperación de imágenes, aplicando técnicas de minería de datos y procesamiento de imágenes en donde no se realiza la utilización de texto para realizar una consulta sino una imagen.

**Keywords:** Recuperacion de Imagen-Procesamiento de Imágenes-ORB-Distancia de Hamming.

## 1 Introducción

Hoy en día es muy común ver una prenda de vestir de cualquier tipo e ingresar a internet y realizar una búsqueda con la intención de encontrar una similar, todo este esfuerzo puede llevar horas o incluso días.

Frente a este problema se presenta este sistema de búsqueda y recuperación de imágenes en base al contenido, en contraposición a los tradicionales como los metadatos, las palabras claves, etiquetas o descripciones asociadas a una determinada imagen. Al hablar del contenido de la imagen se hace referencia a su color, textura, formas, entre otras características.

También podemos decir que las palabras claves limitan el alcance de las consultas y precisamente no ofrecen una garantía de que las primeras imágenes que se encuentren sean de relevancia para aquel que realiza la consulta. El desarrollo del trabajo se organiza de la siguiente forma, en la sección 2 se propone una breve descripción sobre el estado del arte: Sistemas de Recuperación de Imágenes Basadas en Contenido (CBIR), en la sección 3 se plantea los DataSets empleados y las características de los mismos, en la sección 4 se abordara las herramientas usadas, en la sección 5 se describe el algoritmo ORB, en la sección 6 dedicado a

las métricas de similitud utilizadas, en la sección 7 se expone los resultados de los experimentos junto a las pruebas computacionales y finalmente en la sección 8 esta dedicada a la conclusiones y trabajos futuros.

## 2 Sistemas de Recuperación de Imágenes Basadas en Contenido (CBIR)

Un CBIR (Content-based Image Retrieval) [5] es un sistema de búsqueda para recuperar imágenes en base a su contenido como ser color, formas, texturas u otra información de relevancia en la imagen.

Las recuperaciones de imágenes basadas en metadatos son poco prácticas cuando se trata de bases de datos grandes, dado a que esta clase de sistemas lo que hacen es basarse en pistas textuales como anotaciones realizadas a mano y etiquetas junto con sugerencias contextuales automatizadas, como el texto que aparece cerca de la imagen en una página web o el mismo nombre de la imagen. En el presente trabajo se implementó una técnica de consulta, que implica primero dar una imagen al sistema CBIR, luego se desarrolla una búsqueda de manera interna aplicando un algoritmo para después obtener información propia de la imagen, luego se compara con cada información de cada una de las imágenes de la base de datos, y finalmente dar resultado en base a un criterio, como por ejemplo en nuestro caso la distancia. Un claro ejemplo de CBIR es TinEye, que es un motor de búsqueda de imágenes inversas.

Los principales componentes [1] de un sistema CBIR son:

- **Interfaz de usuario:** Permitir realizar las consultas y visualizar los resultados obtenidos de las mismas. Las consultas pueden ser realizadas enviando imágenes de referencia al sistema. Esta permite la interacción del usuario para refinar el proceso de búsqueda a partir de la realimentación de valores para obtener el resultado deseado.
- **Análisis de imágenes:** El componente se encarga de realizar el procesamiento de las imágenes, su análisis y extracción de las características mediante técnicas de visión. En donde se calcula y cuantifica la información relativa a color, texturas o formas presentes en la imagen.
- **Comparación y funciones de similitud:** Este es un aspecto fundamental y se relaciona con la definición de los vectores de características que fueron extraídos en el Análisis de Imágenes. Las métricas o funciones de similitud permiten comparar el vector de características de la imagen de referencia (Imagen de consulta) con los vectores de características de las imágenes almacenadas en la base de datos de contenido.

De esta forma aplicando un CBIR, se resuelven algunos problemas como ser el tiempo empleado en indicar textualmente cada imagen, y los problemas de inconsistencia que deriva de alguna característica de una imagen a través de las palabras.

### 3 DataSets

Para el desarrollo del siguiente trabajo, se han desarrollado dos DataSets y se ha descargado uno de internet, los cuales se describen a continuación:

1. DataSet de 100 imágenes extraídas de manera manual de internet relacionada a abejas junto a flores, las cuales están distribuidas en diferentes lugares con diferentes posiciones.
2. DataSet de 127 imágenes relacionadas a remeras extraídas desde Google en donde en la parte del pecho de la remera se encuentran imágenes de diferentes cantantes, famosos, dibujos animados, figuras públicas, entre otras.
3. DataSet de 999 imágenes turísticas relacionadas a viajes, paisajes, cultura, comida, playas, edificios, autobuses, entre otras. Este DataSet ha sido descargado de internet.

### 4 Herramientas Utilizadas

El lenguaje que se utilizó, para realizar la implementación y desarrollo del CBIR fue Python, el cual es un lenguaje de programación cuya filosofía hace hincapié en una sintaxis que favorezca un código legible. Se trata de un lenguaje de programación multi-paradigma. Python cuenta con un amplio conjunto de librerías para resolver problemas que requieren una capacidad de cómputo mayor al de los lenguajes tradicionales.

OpenCV y Matplotlib son librerías para el procesamiento de imágenes dentro de Python. La primera posee un conjunto de herramientas bastante interesantes que ha permitido el desarrollo del trabajo. Para graficar se utilizó la segunda librería que permite realizar gráficos como histogramas, gráficos de distribuciones y más.

### 5 Oriented Fast y Rotativo Brief (ORB)

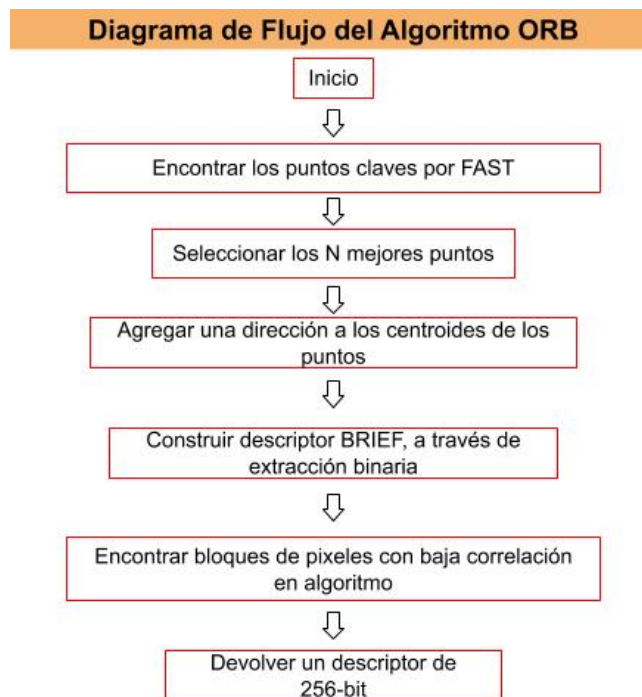
ORB [4] es un algoritmo desarrollado en OpenCV Labs, que hemos adoptado como base para nuestro trabajo, si bien existen otros algoritmos para extraer características como ser SIFT [2], que se caracteriza por trabajar con imágenes invariables de escala.

ORB básicamente trabaja de la siguiente forma, detecta puntos de puntos claves FAST (Features from Accelerated Segment Test) y construye el descriptor BRIEF (Binary Robust Independent Elementary Feature).

FAST es un método de detección de esquinas, que permite extraer puntos de características, que luego se utilizan para rastrear y mapear objetos, para cada pixel compara de manera rápida el brillo de ese pixel frente a 16 pixeles, los cuales forman un círculo alrededor del pixel  $p$ . Los pixeles en el círculo se clasifican en tres clases (más claras, más oscuras o similares), en caso de haber más de 8 pixeles más oscuros que  $p$  se selecciona como punto de clave.

Los puntos claves o puntos de interés de una imagen son los puntos más relevantes de la misma, que a pesar de realizar alguna transformación no van a cambiar.

Una limitación de FAST es que no tiene un componente de orientación o característica multiescala. Así es que ORB usa la pirámide de imágenes multiescala, que es prácticamente la misma imagen en diferentes tamaños. Luego de detectar los puntos claves, ORB asigna una orientación a cada punto clave, según cómo cambien los niveles de intensidad alrededor del punto clave. BRIEF se construye seleccionando todos los puntos claves encontrados de la imagen por el algoritmo rápido FAST y los convierte en un vector de característica binario, definiendo de esta manera un objeto.

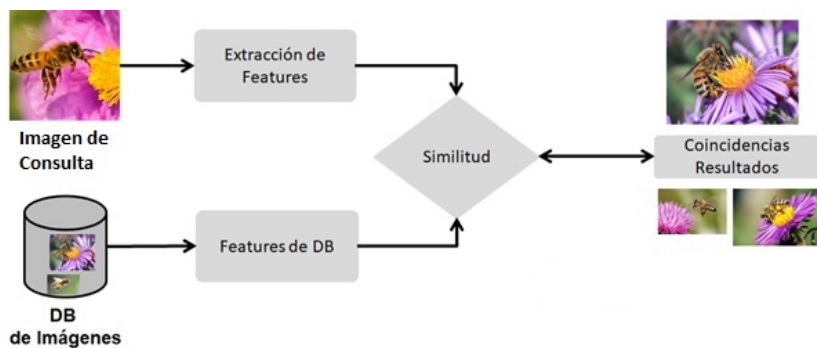


**Fig. 1.** Diagrama de flujo del algoritmo ORB

## 6 Métrica de Similitud

Una métrica de similitud es una función matemática que asocia un valor numérico entre un par de descriptores. La búsqueda por similitud requiere la utilización de un método que permita comparar dos imágenes a través de sus descriptores de

forma rápida. Para ello, podemos mencionar la existencia de diversos métodos para obtención de similitud, como por ejemplo similitud de Jaccard o Arcoseno. En el siguiente trabajo se utilizó la distancia Hamming [3] y también la distancia euclidea.



**Fig. 2.** Modelo de CBIR Aplicado

La distancia de Hamming es la diferencia entre los elementos de dos palabras codificadas  $U$  y  $V$  que son vectores, indicando a la diferencia como  $d(U, V)$ . En teoría de la información, la distancia de Hamming es la efectividad de los códigos de bloque y depende de la diferencia entre una palabra de código válida y otra. Se define un número de bits que tienen que cambiarse para transformar una palabra de código válida, en otra palabra de código válida. Si es mayor esta diferencia, menor es la posibilidad de que un código válido se transforme en otro código válido por una serie de errores. Si dos palabras de código difieren en una distancia  $d$ , se necesitan de errores para convertir una en la otra. Las puntuaciones se realizan a través de la selección de aquellas imágenes que tienen mayor cantidad de puntos en coincidencia y a su vez sus distancias fuera menor.

## 7 Experimentos

Como base del experimento se utilizaron los DataSets descritos en la sección 3, cada una de las imágenes son indexadas de 1 a  $N$ , siendo  $N$  la cantidad de imágenes. Las imágenes son procesadas, obteniendo de esta manera los descriptores referentes a cada imagen aplicando el algoritmo ORB, siendo el resultante una lista de puntos relevantes de cada una las imágenes, luego se organizó cada imagen con su lista de descriptores, para construir un DataSet de descriptores. El trabajo consistió en la aplicación de una imagen de consulta, la misma pasó por un proceso de obtención de descriptores a través del algoritmo ORB.

Otra prueba fue comparar con otra medida de similitud como ser la distancia euclidea. También se construyeron vectores de histogramas RGB para cada una de las imágenes de cada DataSet. ORB aplicado con Hamming se refleja una notable eficiencia en términos de tiempo a diferencia de realizar comparaciones con punto a punto de cada imagen. Los resultados obtenidos fueron muy variados y se reflejan en la tabla 1:

**Table 1.** Resultados de búsqueda en segundos

DataSet	ORB+Hamming	Punto a punto + Distancia Euclidea	Histograma + Distancia Euclidea
DataSet 1	+0,07seg	+1seg	0,05seg
DataSet 2	+8seg	+30min	+5seg
DataSet 3	+17seg	+27min	+10seg

Para medir la eficiencia del algoritmo ORB, primero se analiza la primera imagen del ranking desvuelta y se compara la distancia de similitud de esta imagen recuperada junto con la imagen de consulta. En la tabla 2, se refleja las diferentes modificaciones realizadas a la imagen de consulta como ser cambio de tamaño, cambio de posición y rotación en diferentes grados.

**Table 2.** Eficiencia de Búsqueda

DataSET	DataSet 1	DataSet 2	DataSet 3
Escalado de Imagen de Consulta 50 %	Acertó	Acertó	Acertó
Escalado de Imagen de Consulta 150%	Acertó	Acertó	Acertó
Rotado Derecha 33 grados	Acertó	No acertó	No acertó
Rotado Derecha 120 grados	Acertó	Acertó	Acertó

## 8 Conclusiones y Trabajos Futuros

Durante la elaboración del trabajo, se observó que ORB es un algoritmo de rápida implementación, los resultados obtenidos han sido satisfactorios, no solo por su cantidad de aciertos, sino por el tiempo de proceso de la consulta, no obstante ORB no es óptima en la detección de las variaciones en la imagen de consulta, en casos como por ejemplo cuando la misma era rotada.

Con respecto a la comparación punto a punto, no es una opción viable para un DataSet en donde se supera las 100 imágenes. Así también la distancia Hamming no puede aplicarse cuando los descriptores a comparar son de longitud diferente. Como trabajos a futuro se propone utilizar un algoritmo de extracción de descriptores como SIFT o SURF para construir nuevos descriptores por bordes, como así también explorar otros espacios de color como HSI y optimizar la

herramienta utilizadas para soportar DataSet superiores del orden del millón de imágenes en donde se aplique inteligencia artificial como ser redes neuronales.

## References

1. Cesar Benavides Alvarez, Juan Villegas Cortez, Graciela Roman Alonso, and Carlos Aviles Cruz. Face classification by local texture analysis through cbir and surf points. *IEEE Latin America Transactions*, 14(5):2418–2424, 2016.
2. Ebrahim Karami, Siva Prasad, and Mohamed Shehata. Image matching using sift, surf, brief and orb: performance comparison for distorted images. *arXiv preprint arXiv:1710.02726*, 2017.
3. Mohammad Norouzi, David J Fleet, and Ruslan R Salakhutdinov. Hamming distance metric learning. In *Advances in neural information processing systems*, pages 1061–1069, 2012.
4. Ethan Rublee, Vincent Rabaud, Kurt Konolige, and Gary R Bradski. Orb: An efficient alternative to sift or surf. In *ICCV*, volume 11, page 2. Citeseer, 2011.
5. Yong Rui, Thomas S Huang, Michael Ortega, and Sharad Mehrotra. Relevance feedback: a power tool for interactive content-based image retrieval. *IEEE Transactions on circuits and systems for video technology*, 8(5):644–655, 1998.