

ANÁLISIS DE EFICIENCIA EN ALGORITMOS DE RECONOCIMIENTO DE IMÁGENES DIGITALES APLICABLES A DISPOSITIVOS MÓVILES BAJO LA PLATAFORMA ANDROID

Miguel Ñauñay Ilbay¹, Luis Tipantuña Córdova², Geovanny Raura³, Tatiana Gualotuña⁴
Facultad de Ingeniería en Sistemas e Informática, Escuela Politécnica del Ejército, Sangolquí, Ecuador
¹mfnaunay@hotmail.com, ²lgtipantuna@gmail.com, ³jgraura@espe.edu.ec, ⁴tatimga5@gmail.com

RESUMEN

Los algoritmos de reconocimiento de imágenes son ampliamente utilizados en aplicaciones de visión por computador, durante los últimos años ha existido un creciente interés en el enfoque basado en la descripción de un objeto, utilizando descriptores locales.

Existen varios trabajos relacionados al estudio comparativo de los descriptores locales para el reconocimiento de imágenes, los cuales se enfocan en varios aspectos como: la precisión de las correspondencias o el número de puntos claves encontrados; sin embargo no existe una comparación que considere parámetros de eficiencia y eficacia de estos descriptores aplicados a dispositivos móviles. Este documento analiza el consumo de batería, el uso de memoria, tiempo de ejecución y desempeño de los descriptores locales SIFT, SURF y ORB sobre dispositivos móviles basados en la plataforma Android; los resultados muestran el algoritmo que mejor rendimiento presenta en base a los parámetros indicados.

Palabras Claves:

Descriptores Locales, SIFT, SURF, ORB, Eficiencia, Eficacia

ABSTRACT

The image recognition algorithms are widely used in vision applications by computer, in recent years there has been a growing interest in the approach based on the description of an object, using local descriptors.

There are several related works to the comparative study of local descriptors for image recognition, which focus on various aspects such as the precision of correspondences or the number of key points found, but there is not a comparison that considers parameters of efficiency and effectiveness of these descriptors applied to mobile phones. This document analyzes the battery consumption, memory usage and performance of local descriptors SIFT, SURF and ORB on mobile devices based on Android platform, the results show the algorithm that has better performance based on the parameters indicated.

KeyWords:

Local Features, SIFT, SURF, ORB, Efficiency, Performance

1. INTRODUCCIÓN

Uno de los principales problemas que enfrentan los teléfonos móviles independientemente de su sistema operativo, es la cantidad limitada de recursos de CPU, memoria RAM y batería [1] (Carroll, 2010). Por lo tanto, el uso eficiente de los mismos se convierte en un elemento clave en la evaluación de una aplicación móvil especialmente si se trata de un aplicativo que demanda de un elevado procesamiento computacional como es el caso de los algoritmos para reconocimiento de imágenes.

El reconocimiento de imágenes o reconocimiento de objetos, es una de las aplicaciones más importantes de la visión por computador y aunque existe una gran cantidad de literatura sobre el tema, aun no existe un algoritmo cien por ciento capaz de responder de manera precisa a la gama de variaciones que puede afectar a una imagen [2] (YasirMalik, 2012) como cambios de iluminación, oclusión o puntos de vista diferentes.

Durante los últimos años, ha existido un creciente interés en el enfoque basado en la descripción de un objeto, utilizando “descriptores locales” [2] (YasirMalik, 2012). Los descriptores locales permiten detectar estructuras o puntos significativos en la imagen y obtener una descripción discriminante de estas estructuras a partir de sus alrededores, con el objetivo de ser comparados con otros descriptores usando medidas de similitud.

Uno de los descriptores locales que más influencia ha tenido en el campo de la visión por computador es *ScaleInvariantFeatureTransform*(SIFT) [5] (David Lowe, 1999) propuesto por David Lowe. Aunque el detector y descriptor de puntos clave SIFT ha sido puesto a prueba en varias investigaciones y es ampliamente utilizado en aplicaciones como: composición panorámica, reconocimiento de objetos o la navegación visual; el alto procesamiento computacional que requiere este descriptor ha sido el punto de partida para la creación de nuevos descriptores tales como *Speeded-Up FeatureTransform*(SURF) y *Oriented FAST and Rotated BRIEF* (ORB) los cuales se enfocan en mejorar la eficiencia y la precisión de correspondencias entre puntos clave con respecto a SIFT.

En este escenario, el presente artículo presenta un estudio empírico del comportamiento de los algoritmos SIFT, SURF y ORB para la detección de objetos, mediante la utilización de criterios y métricas de eficiencia y eficacia. Los experimentos fueron realizados en un teléfono móvil con sistema operativo Android, a diferencia de investigaciones previas como la evaluación del desempeño de los métodos de extracción para procesamiento de imágenes 3D [3] (Dwarkanath D, Eichhorn A, Halvorsen P, Griwodz C, 2012), en donde se analiza el desempeño de los citados algoritmos pero orientado a un sistema computacional que procesa imágenes o también la evaluación del método de reconocimiento SIFT en Robots móviles [4] (Ramisa A, Vasudevan A, Aldavert D, Toledo R, Lopez R, 2009), que analiza el desempeño del algoritmo SIFT aplicado a Robots.

El artículo se ha organizado de la siguiente manera: en la sección 2 se estudia los descriptores locales SIFT, SURF y ORB. La sección 3 detallará los criterios y métricas de evaluación. La sección 4 presenta los resultados experimentales obtenidos en la evaluación. Y por último las conclusiones del análisis y los trabajos futuros serán mencionados en la sección 5.

2. DESCRIPTORES LOCALES

Esta sección detalla de forma resumida los descriptores locales a estudiar.

SIFT

Publicado por David Lowe en 1999 [5] propone un algoritmo para la extracción de características de una imagen, que permitan describir los objetos contenidos en ella.

El algoritmo descrito por Lowe consta de 4 etapas:

- Detección de extremos en la escala: La primera etapa del algoritmo realiza una búsqueda sobre las diferentes escalas y dimensiones de la imagen identificando posibles puntos de interés, invariantes a los cambios de orientación y escalado. Esto se lleva a cabo mediante la función DoG (Difference-of-Gaussian).
- Localización de keypoint: los puntos claves se eligen en base a medidas de estabilidad (se eliminan los puntos clave con bajo contraste o si se encuentran localizados en los bordes).
- Asignación de la orientación: La invarianza respecto de la rotación se consigue mediante la asignación a cada uno de los puntos una orientación basada en las propiedades locales de la imagen y representando el descriptor respecto de esta orientación.
- Descriptor de keypoints: Los gradientes locales de la imagen se miden en la región que rodea al punto clave. Éstos son transformados mediante una representación que permite medir niveles de distorsión y cambios en la iluminación de forma local

SURF

Desarrollado por Herbert Bay, Tinne Tuytelaars y Luc Van Gool (2008) [6], como un detector y descriptor de puntos de interés para el reconocimiento de objetos. Siendo una de sus ventajas sobre otros métodos, el mejoramiento de la velocidad de cálculo y la robustez ante transformaciones de las imágenes.

Estas mejoras se consiguen mediante la reducción de la dimensionalidad y complejidad en el cálculo de los vectores de características de los puntos de interés obtenidos, mientras continúan siendo suficientemente particulares e igualmente repetitivos.

A continuación se listan las etapas en las que se divide:

- Detección de puntos de interés o keypoints
- Asignación de la orientación
- Extracción de los descriptores

ORB

El descriptor local ORB [7] (Rublee. E, Rabaud. V, Konolige. K, Bradski, K, 2012) fue publicado en el ICCV (International Conference on Computer Vision) del 2011 como una alternativa a los descriptores SIFT y SURF, siendo una de sus ventajas sobre estos la velocidad de cómputo.

ORB utiliza el algoritmo *Features from Accelerated Segment Test* (FAST) para la detección de puntos claves y *BinaryRobust Independent Elementary Features* (BRIEF) para la extracción de los descriptores.

FAST y BRIEF son algoritmos que mejoran notablemente la velocidad de cómputo, sin embargo carecen de un operador de orientación e invariancia a la rotación respectivamente [1] (Carroll, 2010). Para solventar estos dos problemas ORB propone:

- oFAST (FAST KeypointOrientation): Al algoritmo FAST se le agrega un componente de orientación para los keypoints detectados.
- rBRIEF (Rotation-AwareBrief): Al algoritmo BRIEF se le agrega invarianza a la rotación y mejora la velocidad de cálculo.

3. CRITERIOS Y MÉTRICAS DE EVALUACIÓN

Los criterios y métricas utilizadas en este estudio para el análisis de los descriptores SIFT, SURF y ORB se encuentran divididas en dos grupos: eficiencia y eficacia.

MÉTRICAS DE EFICIENCIA:

Permite medir el uso de recursos del teléfono móvil por parte de los descriptores.

Numero de keypoints

Se define la variable K_t , la cual representa el número promedio de keypoints extraídos de una imagen de entrada.

Tiempo de ejecución

Se define al tiempo de ejecución como el intervalo de tiempo que tarda el descriptor local en procesar una imagen hasta obtener sus características. Esta dada por la formula (1):

$$T_{total} = T_{detector} + T_{extractor} \quad (1)$$

Dónde: $T_{detector}$ representa el tiempo utilizado para detectar los puntos de interés y $T_{extractor}$ el tiempo utilizado para la extracción del vector de características.

Consumo de batería

Sea B_c el porcentaje de batería consumida por un descriptor local para procesar un conjunto de n imágenes de muestra, se define por la formula (2):

$$B_t = \frac{B_c}{n} \quad (2)$$

Dónde: B_t es el porcentaje de batería que consume el descriptor local para procesar una imagen.

Pesos de los descriptores

Este aspecto es muy importante para aplicaciones que necesiten almacenar el vector de características generado por el descriptor para uso futuro. El tamaño de este vector varía dependiendo del descriptor utilizado y el número de características generadas. Esta métrica se define por la formula (3):

$$P_t = KB \text{ necesarios para almacenar un vector de características} \quad (3)$$

MÉTRICAS DE EFICACIA:

Permite medir el desempeño de los descriptores. Se consideran los siguientes parámetros [9] (Mikolajczyk. K, Schmid.C, 2005)

Precisión vs recall

Este enfoque se basa en el número de aciertos y fallos respecto de las correspondencias establecidas entre dos imágenes. A continuación se describe sus componentes:

- *correct matches*: son aquellas imágenes que tras el proceso de comparación han sido identificadas como relacionadas con la imagen original de forma correcta, pues pertenecen a la misma escena.
- *false matches*: son aquellas imágenes que tras el proceso de comparación han sido identificadas como relacionadas con la imagen original de forma incorrecta, pues no pertenecen a la misma escena.
- *correspondences*: representa el número de imágenes que están relacionadas con la imagen original, es decir, que pertenecen a la misma escena.

Precisión (P)

Porcentaje de las imágenes que verdaderamente están relacionadas a la imagen original, se la calcula mediante la fórmula (4)

$$P = \frac{\# \text{ correct matches}}{\# \text{ correct matches} + \# \text{ false matches}} \quad (4)$$

Recall(R)

El porcentaje de imágenes que fueron identificadas como relacionadas con la imagen original respecto al número total de imágenes analizadas, se la define mediante la fórmula (5).

$$R = \frac{\# \text{ correct matches}}{\# \text{ correspondences}} \quad (5)$$

Medida-F1 ($F1$)

Es un valor ponderado entre las dos métricas anteriores y será utilizada para determinar la precisión de los descriptores, se la calcula mediante la fórmula (6).

$$F1 = \frac{2PR}{P+R} \quad (6)$$

4. RESULTADOS EXPERIMENTALES

Esta sección presenta los resultados obtenidos en las pruebas de eficiencia y eficacia. Para la ejecución de las pruebas se utilizaron 100 logotipos destinados al entrenamiento y se ejecutó 10 veces cada prueba, con el objetivo de determinar si los algoritmos cambian su comportamiento según el estado del dispositivo móvil, a continuación se detalla el escenario de pruebas.

4.1 ESCENARIO DE PRUEBA:

Como se puede observar en la figura 1, para la ejecución de las pruebas se desarrolló un aplicativo móvil sobre la plataforma Android que permite obtener las métricas descritas en la sección 3.

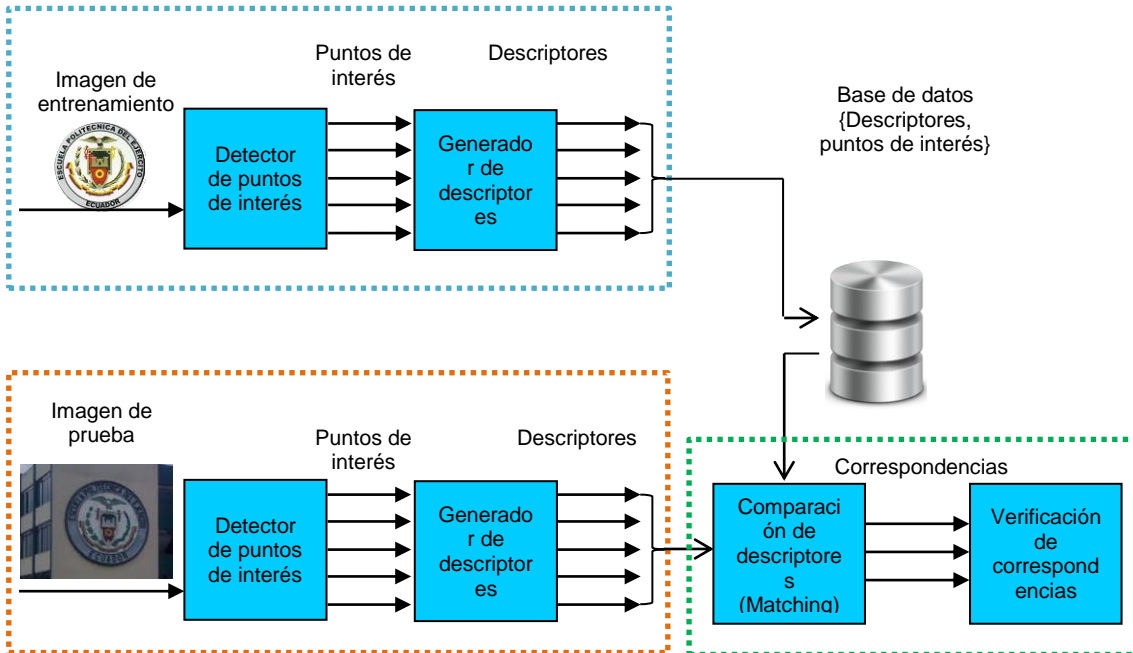


Figura 1: Componentes del escenario de prueba

A continuación se detallan los componentes del aplicativo:

- Base de datos prototipo: base de datos que almacena una tupla (secuencia ordenada de objetos) compuesta por un vector de características y un vector de puntos de interés.
- Componente de entrenamiento: permite almacenar en una base de datos el vector de característica y los puntos de interés generados por uno de los descriptores (SIFT, SURF u ORB) a partir de una imagen de entrada.
- Componente de prueba: permite obtener el vector de característica y los puntos de interés generados por uno de los descriptores (SIFT, SURF u ORB) a partir de una imagen de entrada para su posterior comparación.
- Componente de búsqueda de correspondencias (matching): permite obtener un conjunto de correspondencias entre la tupla obtenida por el componente de prueba y una de las tuplas de la base de datos prototipo.

Para la implementación de los descriptores se utilizó la librería OpenCV disponible para el sistema operativo Android.

La base de imágenes para este análisis ha sido creada a partir de una galería de 200 imágenes (100 logotipos diferentes, es decir, dos imágenes por logotipo) de 640x480 píxeles y con una profundidad de color de 24 bits. Utilizando 100 logotipos para el entrenamiento y 100 logotipos para la fase de pruebas o comprobación.

Los 100 logotipos designados para la fase de comprobación se dividieron en cuatro grupos con las siguientes transformaciones: individuales, escala, rotación e iluminación.

4.2 RESULTADOS

EFICIENCIA

Numero de keypoints

La figura 2, resume el número promedio de *keypoints* detectados en una imagen de entrada por parte de los descriptores, observándose que tanto el descriptor ORB y SURF son los que detectan el mayor número de keypoints (432 y 431 respectivamente), en comparación con el descriptor SIFT que detecta 222 puntos claves.

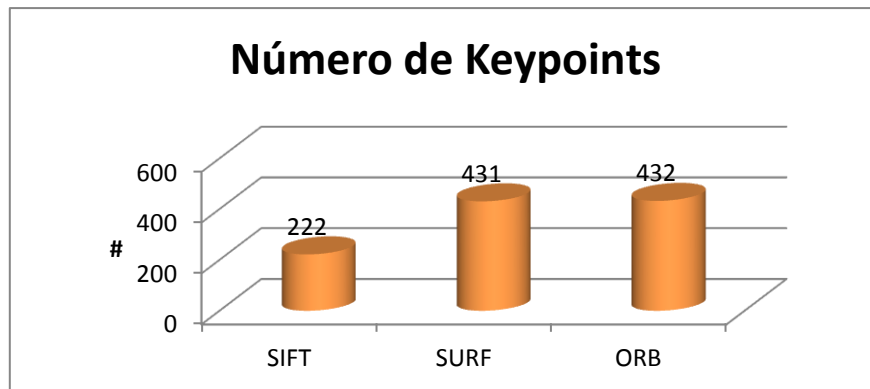


Figura 2: Comparación del número de keypoints (k_t) detectados entre los descriptores SIFT, SURF y ORB

Tiempo de ejecución

En la figura 3 se observa que el tiempo de computo empleado por el descriptor SIFT (3.0 s) es mayor que el tiempo empleados por el descriptor SURF y ORB con aproximadamente 2.06 s y 2.84 s respectivamente. Esto conduce a determinar que el descriptor SIFT no es una buena opción para aplicaciones que realicen reconocimiento en tiempo real, como sería el caso de reconocimiento de imágenes en una trama de video.

Al contrario, el descriptor ORB necesita un tiempo promedio de 0.16 s para procesar una imagen, convirtiéndolo en una mejor alternativa para aplicaciones de tiempo real.

Si bien el tiempo promedio de ejecución del descriptor SURF es de 0.94 s, sigue siendo lento si lo comparamos con el descriptor ORB.

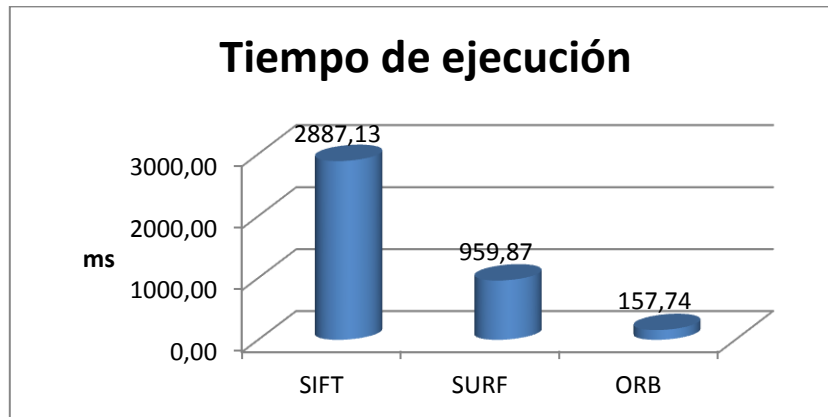


Figura 3: Comparación de tiempo de ejecución (T_{total}) entre los descriptores SIFT, SURF y ORB

Consumo de batería

En lo que se refiere al consumo de batería del dispositivo móvil, en la figura 4 se muestra los niveles consumidos por cada descriptor. Observándose que el porcentaje de batería utilizado por cada descriptor para procesar una imagen es proporcional al tiempo de ejecución analizado en el apartado anterior.

Así, se obtuvo que el porcentaje de batería consumida por ORB es del 0,01% sobre el total disponible, en cambio SURF consume el doble (0,02%) y finalmente SIFT consume cuatro veces más con 0,04%.

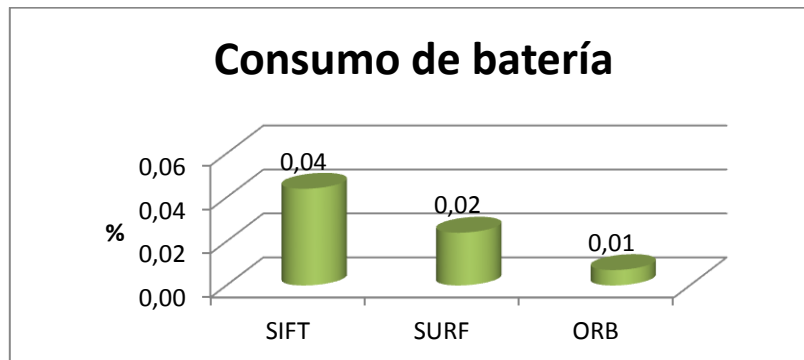


Figura 4: Comparación del (%) de batería consumido (B_t) por los descriptores SIFT, SURF y ORB

Pesos de los descriptores

Con respecto a la cantidad de memoria que necesitan los descriptores para almacenar sus vectores de características, en la figura 5 se observa que tanto SIFT como ORB son los que menos uso de memoria tienen con 102.30KB y 49.66KB respectivamente. En cambio el vector de características generado por el descriptor SURF necesita un promedio de 385.02KB casi 8 veces el peso del vector generado por ORB.

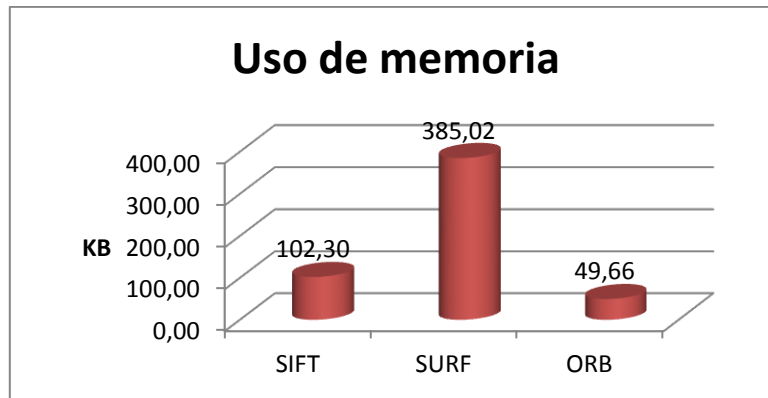


Figura 5: Comparación del peso/tamaño (P_t) que necesitan los descriptores SIFT, SURF y ORB para almacenar sus vectores de características

EFICACIA:

Al calcular el promedio entre los resultados (F1) de los grupos de imágenes analizados, se obtuvo que el descriptor ORB posee una precisión promedio del 41% siendo esta la más alta entre los descriptores evaluados. La Figura 6 muestra el promedio de las precisiones calculadas para cada descriptor.

Se pudo observar que el descriptor SURF tiene mejor desempeño con imágenes que no han sufrido transformaciones y con imágenes con cambios de escala, alcanzando un 65% y 57% respectivamente en cada grupo.

Cabe destacar que la diferencia entre SURF y ORB es apenas del 1% con respecto a cada grupo analizado, el descriptor SIFT es el que menor desempeño presenta al analizar los cuatro grupos de imágenes obteniendo un 33% de precisión.

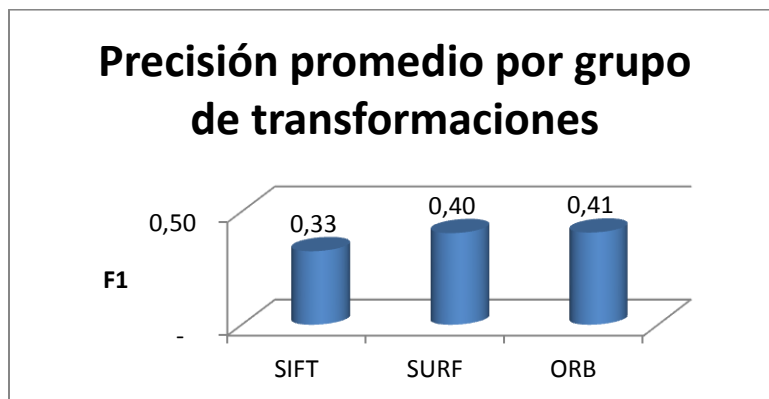


Figura 6: Comparación entre descriptores por cada grupo de transformaciones

5. CONCLUSIONES Y TRABAJO FUTURO

El descriptor ORB muestra ser el que mejor efectividad presenta en cuanto al uso de memoria, tiempo de procesamiento, consumo de batería, y precisión en el reconocimiento de imágenes comparados con los descriptores SIFT, SURF; los resultados obtenidos permiten determinar que las diferencias entre ORB y SURF no tienen una variación significativa, sin embargo el descriptor SIFT se presenta con el menor desempeño en todos los test realizados.

Ninguno de los descriptores analizados tiene un buen desempeño con logotipos, debido a que este tipo de imagen no ofrece una alta cantidad de características visuales como texturas, número de objetos que componen la imagen, y transformaciones. Se ratifica lo expuesto por Juárez (2011) [8] en su estudio sobre el reconocimiento e identificación de logotipos en imágenes con transformada SIFT en un ambiente desktop utilizando el mismo tipo de imágenes aquí analizadas.

Se propone como trabajo futuro el análisis de eficiencia y eficacia de los descriptores SIFT, SURF y ORB en sistemas operativos móviles como: iOS, Windows phone y BlackBerry; además la investigación de tecnologías como OCR que incorpora metadatos en imágenes para conseguir mejoras en la eficiencia de los descriptores y precisión en el reconocimiento.

6. REFERENCIAS

[1]Carroll, A. (2010) *"AnAnalysis of PowerConsumption in a Smartphone"*, disponible en URL: http://static.usenix.org/event/usenix10/tech/full_papers/Carroll.pdf, [consultada 26 de Marzo del 2013]

[2]YasirMalik. M, (2012) *"PowerConsumptionAnalysis of a ModernSmartphone"*, disponible en URL: <http://es.scribd.com/doc/116401727/Power-Consumption-Analysis-of-a-Modern-Smartphone#download>, [consultada 26 de Marzo del 2013]

[3] Dwarakanath D,Eichhorn A, Halvorsen P, Griwodz C.(2012) *"Evaluating Performance of Feature Extraction Methods forPractical 3D Imaging Systems"*

[4]Ramisa A, Vasudevan A, Aldavert D,Toledo R, Lopez,R. (2009) *"Evaluation of the SIFT Object Recognition Method in Mobile Robots"*

[5] Lowe, David G. (1999). *"Reconocimiento de objetos locales de escala invariante características"*. Actas de la Conferencia Internacional sobre la Visión por Computador [consultada 30 de Marzo del 2013]

[6] Bay. H, Tuytelaars. T, Van Gool L. (2008) *"SURF: Speeded Up Robust Features"*, disponible en URL: <http://www.vision.ee.ethz.ch/~surf/eccv06.pdf>, [consultada 31 de Marzo del 2013]

[7] Rublee. E, Rabaud. V, Konolige. K, Bradski, K. (2012) *"ORB: anefficientalternativeto SIFT or SURF"*, disponible en URL: https://willowgarage.com/sites/default/files/orb_final.pdf, [consultada 31 de Marzo del 2013]

[8] Juárez. C, (2011), *"Reconocimiento e identificación de logotipos en imágenes con transformada SIFT"*, disponible en URL: http://e-archivo.uc3m.es/bitstream/10016/13624/2/PFC_Cesar_Juarez_Megias.pdf, [consultada 15 de Marzo del 2013]

[9] Mikolajczyk. K, Schmid.C, *"A performance evaluation of local descriptors"*. IEEE Trans. Pattern Anal. Mach. Intell., 27:1615–1630, October 2005.