AVANCES
EN CIENCIAS E INGENIERÍAS
ARTÍCULO/ARTICLE SECCIÓN/SECTION C

Detección y Agrupación de Logos

Luis Miguel Prócel[1*] **y Vicent Casellas**[2]

[1]*Colegio de Ciencias e Ingenierías –El Politécnico–, Universidad San Francisco de Quito*
Diego de Robles s/n y Vía Interoceánica, Quito, Ecuador
[2]*Departamento de Tecnologías de la Información y las Comunicaciones, Universitat Pompeu Fabra*
Tánger 122-140, Edifici Tànger (Campus de la Comunicació-Poblenou), Barcelona-España
[*]*Autor principal/Corresponding author, e-mail: lprocel@usfq.edu.ec*

Editado por/Edited by: R. Játiva, M.Sc.
Recibido/Received: 03/25/2012. Aceptado/Accepted: 05/19/2012.
Publicado en línea/Published on Web: 06/30/2012. Impreso/Printed: 06/30/2012.

Abstract

In this paper we develop an algorithm for logo detection and grouping in images. For logo detection, the"Scale-Invariant Feature Transform" (SIFT) descriptor is used, which is one of the most studied and used in pattern recognition in the fields of image analysis and computer vision. We have developed a geometric algorithm for grouping and counting the detected logos. This algorithm is based on the so-called "Geometric Hashing" algorithm. Finally, we perform some tests in order to analyze the robustness of the algorithm.

Keywords. Logo detection and grouping, SIFT descriptor, Geometric Hashing algorithm

Resumen

En el presente trabajo se desarrolla un algoritmo para la detección y agrupación de logos en imágenes. Para la detección de logos se usa el descriptor "Scale-Invariant Feature Transform" (SIFT) que es uno de los más estudiado y usado en la detección de patrones en los campos de análisis de imágenes y visión por computadora (computer vision). Luego, se desarrolla un algoritmo geométrico para la agrupación y el conteo de los logos detectados. Este algoritmo se basa en el algoritmo llamado "Geometric Hashing". Finalmente, se realizan pruebas para analizar la robustez del algoritmo.

Palabras Clave. Detección y agrupamiento de logos, descriptor SIFT, algoritmo Geometric Hashing.

Introducción

La detección de patrones es uno de los temas más estudiados en los campos de análisis de imágenes y visión por computadora. En particular, la detección de logos es una herramienta que puede ser usada por empresas audiovisuales y de publicidad. Por ejemplo, dichas empresas pueden facturar por el tiempo de aparición de un logo en un partido de futbol o en una conferencia de prensa.

En general el proceso para la detección de patrones consiste en:

1. Detectar regiones de interés en una imagen de prueba (donde se quiere detectar los patrones). Se asocia a cada región un punto de interés o *keypoint*.
2. Asociar un descriptor a cada región detectada
3. Comparar y parear los descriptores de la imagen de prueba con descriptores de patrones que queremos detectar. En el presente trabajo los patrones que queremos detectar son logos comerciales.

La importancia de elegir un detector y un descriptor radica en la robustez de estos para detectar regiones que han sufrido transformaciones de similitud (rotación, traslación y escalamiento) y afines (se pierden los ángulos rectos pero se mantiene el paralelismo). Cuando solamente existen transformaciones de similitud en una imagen, un buen método para detección de logos es el uso de momentos segundo orden invariantes a este tipo de transformación como los de Hu [1] [2].

Para transformaciones afines, existen algunos tipos detectores que son invariantes a este tipo transformación como son: *Harris-Affine* y *Hessian-Affine* [3], *Edge-Based Region* [3], *Intensity Extrema-based Region* [3], *Maximally Stable Extremal Region* [3], *Salient Region* [3]

ISSN 1390-5384

y *Scale-Invariant Feature Transform Detector* (detector SIFT) [4]. Uno de los descriptores más robusto a este tipo de transformaciones es el *Scale-Invariant Feature Transform Descriptor* (descriptor SIFT) [4]. En el presente trabajo se usa algunos de los detectores arriba mencionados y sobre los *keypoints* de cada región de interés, se aplica el descriptor SIFT.

Finalmente, los *keypoints* de la imagen de prueba que fueron pareados con el logo, deben ser agrupados para concluir que efectivamente existe la presencia de dicho logo. Para la agrupación de los *keypoints* se desarrolla un algoritmo geométrico basado en el llamado *Geometric Hashing Algorithm* [5].

Metodología

El detector y descriptor SIFT

El detector SIFT

Para aplicar el detector SIFT se usa el llamado espacio de escalas en una imagen (*scale space*) el cual se define como la siguiente convolución:

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \tag{1}$$

donde $I(x, y)$ la imagen y $G(x, y, \sigma)$ es una función Gaussiana bidimensional:

$$G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} exp\left(- \frac{x^2 + y^2}{2\sigma^2} \right) \tag{2}$$

Para obtener *keypoints* estables en el espacio de escalas se usa la Diferencia de Gaussianas (*difference-of-Gaussian* o DOG):

$$\begin{aligned} D(x, y, \sigma) &= (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \\ &= L(x, y, k\sigma) - L(x, y, \sigma) \end{aligned} \tag{3}$$

donde k es un valor constante.

Las imágenes convolucionadas son agrupadas en octavas. Cada octava está separada por escalas de 2σ. Para definir las imágenes en octavas superiores se muestrea la imagen tomando los pixeles pares e impares en cada fila y columna [4].
Con el set de imágenes definido en cada octava, se localiza puntos extremos. Para esto, cada punto se compara con sus 8 vecinos en la misma escala y con sus nueve vecinos en las escalas superior e inferior. Si los puntos son más grandes o pequeños que todos sus vecinos, se ha detectado un candidato a *keypoint*. Luego de una interpolación de la posición y evaluación del contraste de los candidatos, se obtienen los *keypoints* definitivos [4]. Finalmente se determina para cada pixel vecino de cada *keypoint* en su escala, el valor del gradiente y su orientación usando las siguientes expresiones:

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \tag{4}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L(x, y+1) - L(x, y-1)}{L(x+1, y) - L(x-1, y)} \right) \tag{5}$$

La orientación de cada vecino es almacenada en un histograma ponderado con una gaussiana circular de σ igual a 1.5 veces la escala del *keypoint*. El pico máximo es medido y asiganado al *keypoint* como su orientación. Este tipo de computo hace que los *keypoints* sean invariantes a transformaciones afines [4].

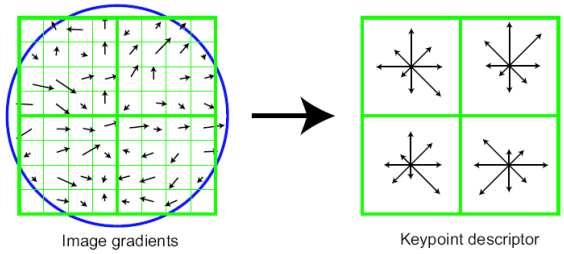

Figura 1: Ejemplo de cómo computar el descriptor SIFT. La magnitud y orientación del gradiente son ponderados por una ventana circula Gaussiana (círculo en azul) en la imagen de la izquierda. En este caso, la región es dividida en 4 subregiones. El histograma de la orientación del gradiente se muestra en la imagen derecha. El tamaño del descriptor es $4 \times 8 = 32$. Esta figura fue tomada de [4].

El descriptor SIFT

Una vez determinada la posición, la escala y la orientación de cada *keypoint*, la magnitud y orientación del gradiente de cada pixel alrededor del *keypoint* es computado. Estas mediciones se realizan en la escala del *keypoint* [4]. Una función Gaussiana con σ igual a 0.5 veces la escala del *keypoint* es usada para ponderar estos valores. La ventana formada por esta Gaussiana es dividida en 16 subregiones [4]. En cada subregión, un histograma de las orientaciones ponderado por sus magnitudes es calculado. El tamaño de este histograma es 8 bins. Al final se obtiene un vector de $16 \times 8 = 128$ elementos conformado por estos histogramas [4]. La figura 1 muestra un ejemplo de este descriptor. Este descriptor puede ser usado con otros detectores afines aplicando un proceso de normalización sobre las regiones de interés [6].

El proceso de correspondencia con el descriptor SIFT

Para cada descriptor en la imagen de prueba, se calcula su distancia con respecto a todos los descriptores en la imagen del logo. Se selecciona la distancia más pequeña y la segunda más pequeña y se computa el radio entre ellas. Este radio es comparado con un valor umbral y se rechaza los pareamientos mayores este valor. El valor umbral sugerido por [4] es 0.8. Esto se puede representar como:

$$\begin{cases} \frac{smallest}{next-smallest} > 0{,}8 & \text{se rechaza el pareamiento} \\ \frac{smallest}{next-smallest} \leq 0{,}8 & \text{se acepta el pareamiento.} \end{cases} \tag{6}$$

El algoritmo de Geometric Hashing

El algoritmo de *Geometric Hashing* enfoca el reconocimiento de objetos como un problema de correspondencia [5]. Este algoritmo es del tipo hipótesis y prueba. Se necesita tres pasos para su aplicación:

- Generar una hipótesis acerca de la correspondencia entre las características de la imagen y una colección de características de objetos. Usualmente se realiza una hipótesis acerca de la proyección del objeto en la imagen de prueba.
- Usar esta hipótesis para formar el objeto que deseamos buscar. Este paso se llama proyección hacia atrás.
- Comparar el objeto formado con la hipótesis, con las proyecciones del objeto en la imagen de prueba para rechazar o aceptar la hipótesis.

Invariantes afines para puntos coplanares

Se asume que el modelo está compuesto por un set de puntos coplanares. Si se escogen tres de estos puntos $\mathbf{P_0}$, $\mathbf{P_1}$ y $\mathbf{P_2}$ como un sistema de referencia, cualquier otro punto $\mathbf{P_i}$ puede ser expresado en términos de este sistema usando la siguiente combinación lineal [5]:

$$\mathbf{P_i} = \mathbf{P_0} + u_{i1}(\mathbf{P_1} - \mathbf{P_0}) + u_{i2}(\mathbf{P_2} - \mathbf{P_0}) \tag{7}$$

donde u_{ij} son las coordenadas en el nuevo sistema.

Los puntos de este modelo, $\mathbf{P_i}$, pueden proyectarse para obtener los puntos de la imagen $\mathbf{p_i}$. Si se define a la cámara como ζ, se tiene que la relación entre los puntos es: $\mathbf{p_i} = \zeta\mathbf{P_i}$. Entonces se puede aplicar para los puntos del objeto proyectado en la imagen [5]:

$$\mathbf{p_i} = \mathbf{p_0} + u'_{i1}(\mathbf{p_1} - \mathbf{p_0}) + u'_{i2}(\mathbf{p_2} - \mathbf{p_0}) \tag{8}$$

Esto significa que este tipo de medidas son invariantes afines.

Algoritmo invariante afín

En este algoritmo, las entradas de datos son puntos pareados entre la imagen del objeto (patrón) y la imagen de prueba. Un punto específico en la imagen del objeto, $\mathbf{P_i}$, puede tener más de una correspondencia en la imagen de prueba, $\mathbf{p_{ij}}$, como se puede ver en la figura 2.

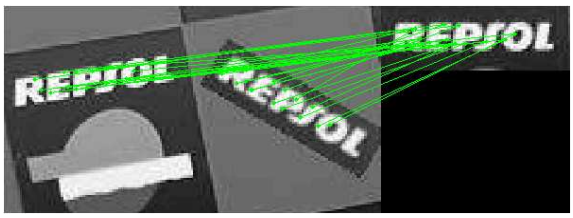

Figura 2: Entrada de datos para el Algoritmo de Geometric Hashing. Izquierda: imagen del logo. Derecha: imagen de prueba

El algoritmo de Geometric Hashing es un algoritmo de votos que consiste en [5]:

→ Para todas las combinaciones de tres puntos en la imagen del objeto, $\mathbf{P_i}$, $\mathbf{P_j}$ y $\mathbf{P_k}$. Estos puntos serán los sistemas de referencia.

→ Para todas las combinaciones de tres puntos, $\mathbf{p_{ir}}$, $\mathbf{p_{is}}$ y $\mathbf{p_{kt}}$, en la imagen de prueba que corresponden a los tres puntos de la imagen del objeto, $\mathbf{P_i}$, $\mathbf{P_j}$ y $\mathbf{P_k}$. Estos puntos serán los sistemas de referencia en la imagen de prueba.

→ Los votos son reiniciados a cero para este sistema de referencia.

→ Se computa las coordenadas u_{i1} y u_{i2}.

→ Se computa las coordenadasu'_{ir1} y u'_{ir2}.

→ Si el valor u_{i1} y u'_{ir1} son similares y lo mismo sucede para u_{i2} y u'_{ir2}, se incrementa el número de votos.

→ Se cuenta el número de votos para el sistema de referencia. Si hay suficientes votos, el sistema $\mathbf{p_{ir}}$, $\mathbf{p_{is}}$ y $\mathbf{p_{kt}}$ y todos los puntos que tienen coordenadas similares pertenecen al mismo objeto.

Modificaciones geométricas al algoritmo de Geometric Hashing

Para mejorar la detección y agrupamiento de objetos, algunas consideraciones gemétricas se incluyen en el algoritmo de *Geometric Hashing*. Estas condiciones son: radios de áreas, factor de escalamiento entre longitudes y posición del baricentro.

Condición del radio entre áreas

Si dos regiones A' y B' son obtenidas por la aplicación de la misma transformación afín a dos regiones A y B se tiene:

$$\frac{area(A')}{area(B')} = \frac{area(A)}{area(B)} \tag{9}$$

En el algoritmo modificado, se escoge dos triángulos en la imagen del logo (estos triángulos pueden compartir el mismo lado o un vértice). El primer triángulo está formado por los puntos $\mathbf{P_i}$, $\mathbf{P_j}$ y $\mathbf{P_k}$, mientras que el segundo por los puntos $\mathbf{P_l}$, $\mathbf{P_m}$ y $\mathbf{P_n}$. En la imagen de prueba, el triángulo $\mathbf{p_{ir}}$, $\mathbf{p_{js}}$, $\mathbf{p_{kt}}$ corresponde al primer triángulo y el triangulo $\mathbf{p_{lu}}$, $\mathbf{p_{mv}}$, $\mathbf{p_{nw}}$ tiene correspondencia con el segundo. Se aceptan triángulos que cumplan con:

$$abs\left(\frac{area(\mathbf{P_i},\mathbf{P_j},\mathbf{P_k})}{area(\mathbf{P_l},\mathbf{P_m},\mathbf{P_n})} - \frac{area(\mathbf{p_{ir}},\mathbf{p_{js}},\mathbf{p_{kt}})}{area(\mathbf{p_{lu}},\mathbf{p_{mv}},\mathbf{p_{nw}})}\right) \leq \epsilon_1 \tag{10}$$

donde $\epsilon_1 = 0{,}1$ en este trabajo.

Condición de longitud

Para mejorar la correspondencia, se necesita definir un factor de escala entre las longitudes de los lados de los triángulos debido a que el tamaño del logo puede ser diferente en la imagen de prueba. El factor de escala está definido por:

$$F = \sqrt{\frac{1}{2}\frac{area(\mathbf{P_i},\mathbf{P_j},\mathbf{P_k})}{area(\mathbf{P_l},\mathbf{P_m},\mathbf{P_n})} + \frac{1}{2}\frac{area(\mathbf{p_{ir}},\mathbf{p_{js}},\mathbf{p_{kt}})}{area(\mathbf{p_{lu}},\mathbf{p_{mv}},\mathbf{p_{nw}})}} \tag{11}$$

Se aceptan triángulos que cumplan con la siguientes condiciones:

$$abs\Big(max(\overline{\mathbf{P_i P_j}}, \overline{\mathbf{P_i P_k}}, \overline{\mathbf{P_j P_k}}) - F \times max(\overline{\mathbf{P_{ir} P_{js}}}, \overline{\mathbf{P_{ir} P_{kt}}}, \overline{\mathbf{P_{js} P_{kt}}})\Big) \leq \epsilon_2$$

y

$$abs\Big(max(\overline{\mathbf{P_l P_m}}, \overline{\mathbf{P_l P_n}}, \overline{\mathbf{P_m P_n}}) - F \times max(\overline{\mathbf{P_{lu} P_{mv}}}, \overline{\mathbf{P_{lu} P_{nw}}}, \overline{\mathbf{P_{mv} P_{nw}}})\Big) \leq \epsilon_2 \quad (12)$$

donde $\epsilon_2 = 5$ en el presente trabajo.

Condición del baricentro

Para asegurarse que el par de triángulos pertenece al mismo logo, se mide la distancia entre los baricentros de los pares de triángulos en el logo y en la imagen de prueba, y se aceptan pares que cumplan con:

$$abs\Big(dist\Big(bar(\mathbf{P_i}, \mathbf{P_j}, \mathbf{P_k}), bar(\mathbf{P_l}, \mathbf{P_m}, \mathbf{P_n})\Big) - dist\Big(bar(\mathbf{p_{ir}}, \mathbf{p_{js}}, \mathbf{p_{kt}}), bar(\mathbf{p_{lu}}, \mathbf{p_{mv}}, \mathbf{p_{nw}})\Big)\Big) \leq \epsilon_2 \quad (13)$$

donde la función $dist(\mathbf{P_1}, \mathbf{P_2})$ entrega la distancia entre los puntos $\mathbf{P_1}$ y $\mathbf{P_2}$, la función $bar(\mathbf{P_i}, \mathbf{P_j}, \mathbf{P_k})$ entrega el baricentro del triángulo $\mathbf{P_i}, \mathbf{P_j}, \mathbf{P_k}$ y $\epsilon_2 = 5$ en este trabajo.

Resultados

Para el presente trabajo se usaron *keypoints* obtenidos por los detectores SIFT [4], Harris Affine [3], Hessian Affine [3] y Maximally Stable Extremal Region [3]. Sobre los *keypoints* calculados se aplica el descriptor SIFT. Luego se realiza el proceso de correspondencia entre los *keypoints* de la imagen de prueba y los *keypoints* de la imagen del logo. Finalmente, se aplica el algoritmo de *geometric Hashing* con las modificaciones geométricas ya explicadas. Para las pruebas de robustez del algoritmo de agrupamiento, a la imagen de prueba se le realiza los siguientes cambios:

- modificación de su histograma para obtener imágenes oscuras y brillantes,
- convolución con una Gaussina para obtener imágenes desenfocadas,
- adición de ruido Gaussiano, y
- transformación afín para cambiar la forma de la imagen.

A continuación se muestra los resultados:

Figura 3: Imagen de prueba original.

Figura 4: Resultado de la aplicación del algoritmo de agrupación en la imágen de prueba original (sin ninguna modificación).

Figura 5: Aplicación del algoritmo sobre la imagen de prueba modificada su histograma.

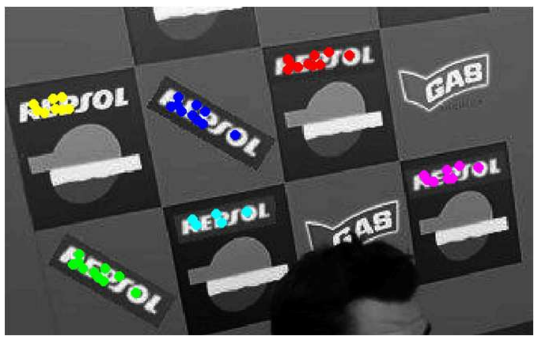

Figura 6: Aplicación del algoritmo sobre la imagen de prueba modificada su histograma.

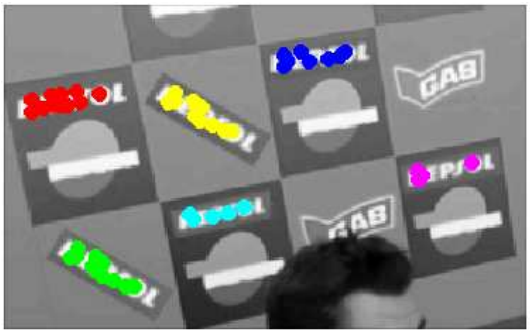

Figura 7: Aplicación del algoritmo sobre la imagen de prueba cambiada su escala. La escala es $0{,}85$ veces el tamaño original.

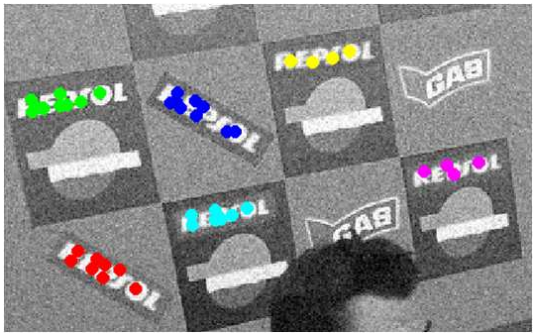

Figura 8: Aplicación del algoritmo sobre la imagen añadida ruido. Se añadió ruido Gaussiano de $\sigma = 15$.

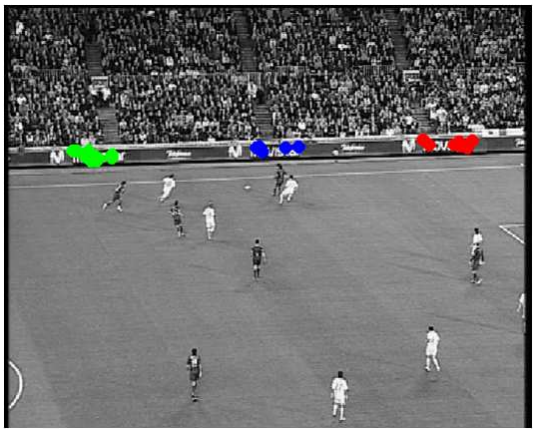

Figura 9: Aplicación del algoritmo sobre un frame de un partido de fútbol. Los 3 logos de "Movistar"fueron detectados.

Conclusiones

Debido a la naturaleza combinatoria del algoritmo, para aplicaciones en tiempo real se debe considerar programar en lenguajes no interpretativos. Este algoritmo es muy sensitivo a la cantidad de *keypoints* detectados. Para mejorar la eficacia del mismo, se debe usar no solamente los *keypoints* que son resultado del detector SIFT, sino de otros detectores afines y a todos ellos aplicarles el descriptor SIFT. La falta de *keypoints* es el motivo más importante para no detectar un logo con el algoritmo presentado. Este inconveniente se presentó cuando se aplicaron condiciones extremas a la imagen de prueba (mucho ruido, desenfoque alto, imagen muy oscura o brillante, etc.). Las pruebas para medir la robustez del algoritmo fueron satisfactorias en condiciones normales.

Referencias

[1] Hu, M. 1962. "Visual Pattern Recognition by Moment Invariants". *IRE Trans. Information Theory*. 2(IT - 8), 179 – 187.

[2] Pratt, W. 2007. *"Digital Image Processing"*, Willey, 4th edition.

[3] Mikolajczyk, K., Tuytelaars, T., Schmid, C., Zisserman, A., Matas, J., Schaffalitzky, F., Kadir, T., and Van Gool, L. 2005. "A Comparison of Affine Region". *International Journal of Computer Vision*. 65(1 - 2), 43 – 72.

[4] Lowe, D. 2004. "Distinctive Image Features from Scale-invariant Keypoints". *International Journal on Computer Vision*. 60(2), 91 – 110.

[5] Forsyth, D. and Ponce, J. 2002. *"Computer Vision: A Modern Approach"*, Prentice Hall.

[6] Mikolajczyk, K. and Schmid, C. 2003. "A Performance Evaluation of Local Descriptors". *Proceedings of IEEE Conference on Computer Vision and Pattern*.