

# Histogramas de Orientaciones

Joaquín Pérez Araya  
Departamento de Ciencias de la Computación  
Universidad de Chile  
Santiago, Chile  
joaquin.perez.a@ug.uchile.cl

## Resumen—

### INTRODUCCIÓN

Para el estudio de imágenes una de los descriptores de imagen útiles es el histograma de orientaciones, es decir, un histograma de las orientaciones internas de la imagen, los ángulos internos de ésta. En este documento se mencionarán 3 formas distintas de histogramas de orientación, uno simple...

Una de las utilidades de los histogramas de orientación es la recuperación de imágenes por dibujos, que consiste en comparar...

### DISEÑO E IMPLEMENTACIÓN

La implementación realizada se divide en dos partes, los histogramas y las consultas.

#### Histogramas

Para el diseño e implementación de los tres histogramas se creó la función auxiliar *ConvolveSobel* que calcula el gradiente en ambas direcciones utilizando la máscara Sobel. La implementación utiliza el siguiente algoritmo:

---

#### Algoritmo 1: ConvolveSobel

---

**Data:** *image* un arreglo que representa la imagen.

**Result:** Gradientes en X e Y de *image*.

**begin**

$Sobel_x \leftarrow \{-1, 0, 1\}, \{-2, 0, 2\}, \{-1, 0, 1\}$

$Sobel_y \leftarrow transpose(Sobel_x)$

$G_x \leftarrow convolve(image, Sobel_x)$

$G_y \leftarrow convolve(image, Sobel_y)$

**return**  $G_x, G_y$

Notar que las funciones *transpose()* y *convolve()*, son de los paquetes *numpy* y *skimage* respectivamente.

---

Histograma de Orientaciones Simple: Este se basa en la obtención directa de los ángulos de los vectores gradiente utilizando el arcotangente de la división de ambos, para luego dividir dichos ángulos en los diferentes bins y votar por ellos según la magnitud de los vectores del ángulo.

Histograma de Orientaciones de Bordes Locales (HELO): Calcula los ángulos

Histograma de Orientaciones de Bordes Locales Suave (SHELO):

---

#### Algoritmo 2: Histograma de Orientaciones Simple

---

**Data:** *image* un arreglo que representa la imagen, la cantidad de bins  $k$ .

**Result:** Histograma de orientaciones de *image*.

**begin**

$h \leftarrow zeros(k)$

$G_x, G_y \leftarrow ConvolveSobel(imagen)$

$angles \leftarrow arctan(\frac{G_x}{G_y})$

**for** *angle* in *angles* **do**

**if** *angle* < 0 **then**

$angle \leftarrow angle + \pi$

$mag \leftarrow \sqrt{G_x^2 + G_y^2}$

$index \leftarrow \lfloor \frac{angles}{\pi} (k - 1) \rfloor \bmod k$

**for**  $i = 0$  to  $k$  **do**

$r, c \leftarrow indexes$  where  $index = i$

$h[i] \leftarrow \sum mag[r, c]$

$h \leftarrow normalize(h, 2)$

**return**  $h$

---

#### Consultas

Para las consultas se aplica el siguiente algoritmo:

#### EXPERIMENTACIÓN

#### CONCLUSIÓN

#### REFERENCIAS

- [1] Cheddad, A., Condell, J., Curran, K., & Mc Kevitt, P. (2010). Digital image steganography: Survey and analysis of current methods. *Signal Processing*, 90(3), 727–752.

#### ANEXO

---

**Algoritmo 3:** Histograma de Orientaciones de Bordes Locales

---

**Data:** *image* un arreglo que representa la imagen, la cantidad de bins  $k$ , tamaño de los bloques  $B$ .

**Result:** Histograma de orientaciones de *image*.

**begin**

```
h ← zeros(k)
Gx, Gy ← ConvolveSobel(imagen)
angles ← arctan( $\frac{G_x}{G_y}$ )
for angle in angles do
  if angle < 0 then
    angle ← angle +  $\pi$ 
mag ←  $\sqrt{G_x^2 + G_y^2}$ 
index ←  $\lfloor \frac{angles}{\pi}(k-1) \rfloor \bmod k$ 
for i = 0 to k do
  r, c ← indexes where index = i
  h[i] ←  $\sum mag[r, c]$ 
h ← normalize(h, 2)
return h
```

---

---

**Algoritmo 4:** Histograma de Orientaciones de Bordes Locales

---

**Data:** *image* un arreglo que representa la imagen, la cantidad de bins  $k$ , tamaño de los bloques  $B$ .

**Result:** Histograma de orientaciones de *image*.

**begin**

```
h ← zeros(k)
Gx, Gy ← ConvolveSobel(imagen)
angles ← arctan( $\frac{G_x}{G_y}$ )
for angle in angles do
  if angle < 0 then
    angle ← angle +  $\pi$ 
mag ←  $\sqrt{G_x^2 + G_y^2}$ 
index ←  $\lfloor \frac{angles}{\pi}(k-1) \rfloor \bmod k$ 
for i = 0 to k do
  r, c ← indexes where index = i
  h[i] ←  $\sum mag[r, c]$ 
h ← normalize(h, 2)
return h
```

---

---

**Algoritmo 5:** Histograma de Orientaciones de Bordes Locales

---

**Data:** *image* un arreglo que representa la imagen de búsqueda, la cantidad de bins  $k$ , tamaño de los bloques  $B$ .

**Result:** Histograma de orientaciones de *image*.

**begin**

```
database ← ReadDatabase()
histograms ← zeros((length(database), k))
distances ← zeros(length(database))
Gx, Gy ← ConvolveSobel(imagen)
angles ← arctan( $\frac{G_x}{G_y}$ )
for angle in angles do
  if angle < 0 then
    angle ← angle +  $\pi$ 
mag ←  $\sqrt{G_x^2 + G_y^2}$ 
index ←  $\lfloor \frac{angles}{\pi}(k-1) \rfloor \bmod k$ 
for i = 0 to k do
  r, c ← indexes where index = i
  h[i] ←  $\sum mag[r, c]$ 
h ← normalize(h, 2)
return h
```

---