

# CC5508: Procesamiento y Análisis de Imágenes

## Tarea 2: Gradiente y Orientaciones Locales

Prof. José M. Saavedra Rondo

Octubre 2020

### 1. Objetivo

Familiarizarse con el operador gradiente en imágenes y el cálculo de orientaciones locales.

### 2. Descripción

El gradiente de una imagen (Fig. 1) es un operador muy importante en diversas tareas de procesamiento y análisis de imágenes. Por ejemplo, podemos ocupar la orientación del gradiente en tareas de reconstrucción de imágenes de huellas dactilares o en la representación de imágenes para reconocimiento. El gradiente es un operador aplicado en cada pixel de la imagen, y a partir de él, se estima la orientación en cada punto. Sin embargo, en muchos casos la orientación o el gradiente tiene un comportamiento más bien local. Así, resulta más conveniente estimar una orientación representativa por regiones (celdas) en las que una imagen se puede dividir. Uno de los métodos para estimar orientaciones locales es el método de *gradiente de ángulo doble y magnitud cuadrada*, tal como discutimos en cátedra (clase 7). Ejemplos de campos de gradientes locales estimados a través de este método se muestran en las Figuras 1, 2 y 3.

$$\nabla I = \left( \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right) \quad (1)$$

El método se describe en la Clase 7. Además, se encuentra disponible una implementación en el github del ramo. Un problema con el método descrito es que regiones o celdas con alta variación de ángulos pueden generar orientaciones poco representativas. Esto se debe, principalmente, a que todos los puntos que caen en una región participan de igual manera en la estimación del gradiente local. Una mejora sería que puntos cercanos al centro de la región tengan una ponderación mayor que aquellos que están lejos del centro. Además, puntos que están cerca a las fronteras de la región deberían votar también por regiones adyacentes. Es decir, un punto de la imagen, ya no participará solamente en la estimación del gradiente local de la región en la que cae sino que participa en las 4 regiones más cercanas, 2 respecto del eje x (left, right) y otras 2, respecto del eje y (top, bottom). Esto genera lo que denominamos

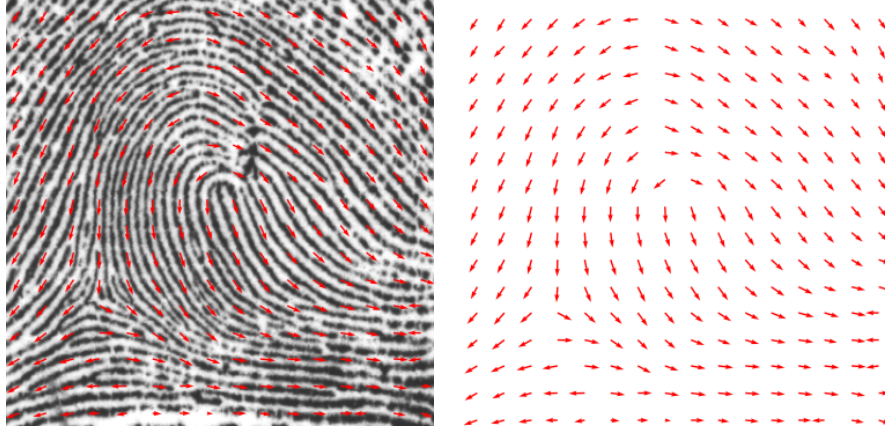


Figura 1: Imagen con su correspondiente campo de gradientes, con división de  $16 \times 16$  celdas

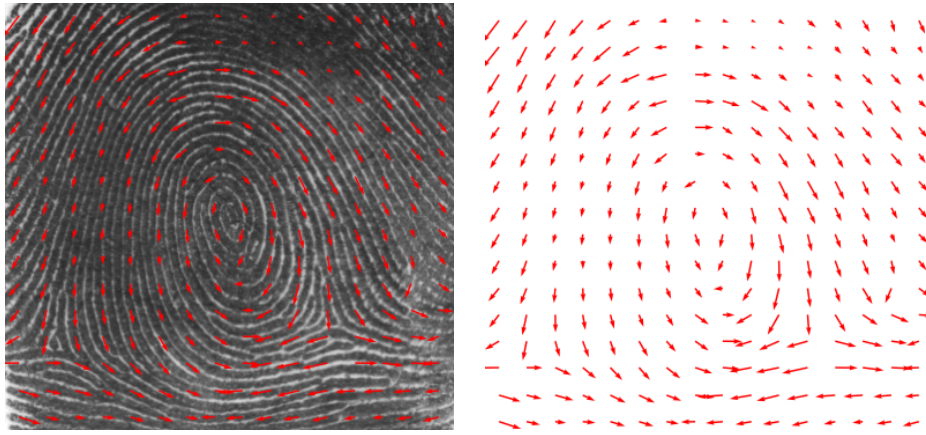


Figura 2: Imagen con su correspondiente campo de gradientes, con división de  $16 \times 16$  celdas

*estimación de gradientes locales en forma suave (soft assignment).* En la clase 7, se describe cómo se realiza la interpolación para realizar el cálculo suave. Adicionalmente, se adjunta un paper (SHELO, Soft Histogram of Edge Local Orientations) que describe el método de interpolación y su aplicación a sistemas de búsqueda.

Por lo anterior, en esta tarea se pide extender el método de orientaciones locales de modo que cada gradiente local sea calculado a través del método de interpolación bi-lineal (soft assignment) explicado en clases (clase 7). Además se pide calcular el histograma de orientaciones locales y evaluar su uso en recuperación de imágenes de huellas dactilares.

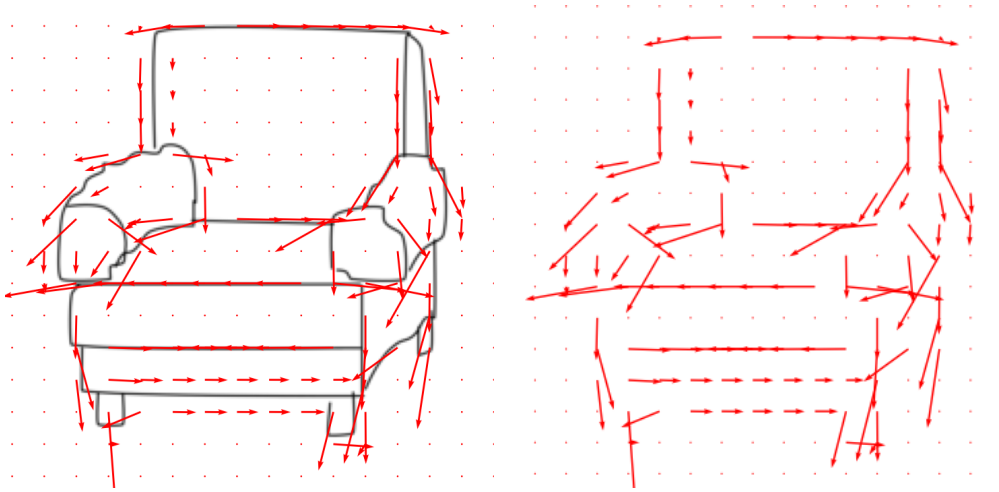


Figura 3: Un sketch con su correspondiente campo de gradientes, con división de  $16 \times 16$  celdas

### 3. Descripción Detallada

1. Implementar el programa *orienta.py*, que reciba una imagen y muestre (show) el campo de gradientes calculados con: a) interpolación bi-lineal y b) sin interpolación (básico). Además el programa de permitir ingresar  $K$ , que indica la granularidad de la división de la imagen, es decir genera una grilla de  $K \times K$  celdas.
2. Implementar el programa *finger.py*, que reciba como entrada una imagen de huella dactilar y muestre las 5 huellas más similares a la imagen de entrada, estas últimas obtenidas del dataset *fingerprints* que se adjunta a esta tarea. Para este fin, cada imagen debe ser representada como un histograma de orientaciones de largo  $L$ , generado por las orientaciones locales. Este método es conocido como SHELO y se describe en el paper adjunto. La similitud entre las imágenes se mide a través de la distancia entre los histogramas correspondientes. Las imágenes más similares a una imagen de entrada son aquellas que generan menor distancia con respecto a la comparación de histogramas con la consulta. Utilice distancia Euclidiana (Eq. 2) entre los histogramas normalizados con norma L2 (Eq. 3).
3. Para el caso anterior, evalúe la calidad de recuperación entre utilizar gradientes locales con interpolación y sin interpolación. Una forma cuantitativa de medir la calidad de recuperación es contando el número de imágenes recuperadas que no tienen relación con la consulta (son de diferentes clases), esto es el *error de recuperación*. En esta caso utilice como consulta cada una de las imágenes del dataset *fingerprints* y recupere las 5 huellas más similares a la entrada, sin considerar la imagen de consulta. Esta técnica de evaluación se denomina *leave-one-out*. Luego cuente el número de imágenes recuperadas que no son de la misma clase que la entrada. Promedie el error para todas

las consultas y eso será el error de recuperación. Muestre el error de recuperación al usar histograma de orientaciones locales con y sin interpolación. Además, realice experimentos para  $K = \{8, 16, 32, 64\}$  y  $L = \{16, 32, 64, 128\}$ .

$$D(H^1, H^2) = \sqrt{\sum_{i=0}^{L-1} (H_i^1 - H_i^2)^2} \quad (2)$$

$$H_{norm} = \frac{H}{\sqrt{\sum_{i=0}^{L-1} H_i^2}} \quad (3)$$

4. EL histograma se construye usando interpolación lineal. Es decir, un ángulo vota por los 2 bins más cercanos en el histograma con sus respectivas ponderaciones(ver Clase 6).
5. Los gradientes en cada punto se estiman con SOBEL.
6. Presente ejemplos de sus resultados.
7. Escriba un reporte de 4-5 páginas sobre el desarrollo de la tarea. Para redactar el informe siga el formato de la siguiente sección.

## 4. Informe

1. **Abstract o Resumen:** es el resumen del trabajo.
2. **Introducción:** se describe el problema y el contexto. Aquí deben definir el término Esteganografía y sus aplicaciones en imágenes. (10 %)
3. **Desarrollo:** se describe el diseño e implementación del programa. (40 %)
4. **Resultados Experimentales y Discusión:** se debe presentar los resultados y hacer un análisis de los mismos. (40 %). En este punto se debe analizar la bondad de utilizar interpolación así como la influencia de los parámetros  $K$  y  $L$ .
5. **Conclusiones** (10 %)

## 5. Entrega

La entrega se realiza por u-cursos hasta el domingo 25 de octubre, 2020, 23:50 hrs. Se debe incluir:

1. Código fuente (en Python)
2. Informe