

Tarea 02

David Rojas

September 2019

1 Resumen

Se implementaron 3 algoritmos de histogramas de orientaciones; Un algoritmo clásico (HO), un algoritmo basado en orientaciones locales (HELO) y un algoritmo basado en orientaciones locales con interpolación de peso de los píxeles sobre las localidades (S-HELO).

El propósito de este trabajo es entender como se pueden utilizar histogramas de orientaciones con el fin de realizar búsquedas en imágenes a partir de bosquejos.

2 Introducción

Este proyecto consiste en crear histogramas de orientaciones con el fin de realizar búsquedas de imágenes a partir de bosquejos.

Los algoritmos a utilizar para generar los histogramas son HO, HELO y S-HELO. Estos algoritmos nos permiten generar histogramas que luego podemos usar para comparar las formas de las imágenes stock y la imagen bosquejada.

Los detalles de los algoritmos en cuestión:

1. HO: Calcula la orientación y magnitud por píxel, luego genera un histograma con las orientaciones de los píxeles con valores ponderados por sus respectivas magnitudes.
2. HELO: También calcula un histograma, pero a diferencia de HO, HELO utiliza los valores por cuadrantes en vez de por píxel.
3. S-HELO: Similar a HELO, pero en vez de utilizar solo los valores de los cuadrantes, utiliza una ponderación de los píxeles cercanos a cada cuadrante en base a la distancia del píxel al centro de dicho cuadrante.

Nuestro objetivo es comparar los algoritmos bajo diferentes condiciones, por lo que luego usaremos un algoritmo de mAP para comparar los resultados de los algoritmos.

3 Desarrollo

Cada algoritmo tiene su propia implementación, las cuales se detallan a continuación:

1. HO: Histograma de orientaciones simple. Se calculan orientaciones para cada píxel utilizando un filtro Sobel para estimar la verticalidad y horizontalidad de la imagen en determinados píxeles.

Luego se calcula el ángulo de cada píxel utilizando un arcotangente entre ambas componentes calculadas. También se calcula la magnitud de la orientación calculando el modulo generado por vector formado por ambas componentes.

Acto seguido, creamos un histograma con un numero K de espacios y dividimos los píxeles en sus lugares correspondientes dependiendo de su orientación. Sumamos al histograma las magnitudes, ya que de este modo se da mayor importancia a píxeles con una orientación de alta magnitud.

Para terminar, se normaliza el histograma.

2. HELO: Este algoritmo se maneja de forma similar a HO, pero en vez de usar cada píxel para calcular el histograma, dividimos la imagen en B cuadrantes. Para cada cuadrante se suman las contribuciones de todos los píxeles dentro del cuadrante al gradiente para formar un tipo de promedio usando el procedimiento especificado en el paper del algoritmo HELO.

Para calcular las sumas en los bloques, lo ideal sería utilizar imagen integral, pero el intento de implementar resulto en errores que no pudieron corregirse, por lo que se implemento un método provisional en el que se calcula una mascara de unos del tamaño del bloque para convolucionar con toda la imagen. Luego simplemente se toma el valor del centro de cada bloque que contiene la suma de todos los valores del bloque.

Luego el algoritmo sigue de forma similar a HO, pero utilizando los valores de cada bloque en vez de cada píxel.

3. S-HELO: El algoritmo se explica en su paper y en general tiene una implementación similar a la de HELO. Existen sin embargo algunas diferencias fundamentales.

El algoritmo de S-HELO utiliza los valores de no solo los píxeles en el bloque, sino que también de los píxeles en bloques contiguos con una ponderación basada en distancia al centro del bloque.

Para calcular este, se creo una mascara con la ponderación de las distancias para cada píxel y se utilizó esta mascara para en lugar de la mascara de unos del algoritmo HELO.

Para poder realizar todas las pruebas fácilmente y poder hacer ajustes al algoritmo de comparación sin tener que volver a crear nuevamente los histogramas cada vez, se crearon todos los histogramas a la vez y se guardaron como archivos.

Primero se crearon archivos de histogramas para las imágenes de stock (imágenes a buscar) usando Canny y sin usar Canny con valores $K = [36, 72, 96]$ y $B = [36, 72, 96]$.

Cada archivo generado en este paso tiene los histogramas de todas las imágenes relacionadas con un objeto en específico con las especificaciones de los algoritmos utilizados en el nombre del archivo.

Luego se realizó un proceso similar para las imágenes de bosquejos, pero en este caso los archivos no contienen varios histogramas, sino que cada archivo contiene el histograma correspondiente a su imagen.

Para el cálculo de mAP se tomaron todos los histogramas y para cada imagen de consulta se calculó la distancia con todos los histogramas para las imágenes de stock. Luego se ordenaron los resultados en base a la distancia con la imagen de consulta y usando estos resultados se calculó el mAP.

4 Resultados Experimentales y Discusión

Los algoritmos de HELO y S-HELO funcionan como corresponde. A continuación en las imágenes Figure 1 y Figure 2 se puede ver que ambos algoritmos generan resultados bastante prometedores.



Figure 1: Orientaciones calculadas con HELO.



Figure 2: Orientaciones calculadas con S-HELO.

Las mAP para todos los histogramas se detallan en la tabla a continuación:

Algoritmo	Canny	K	B	mAP
HO	No	36	-	0.0422
HO	Si	36	-	0.074
HELO	No	36	25	0.0489
HELO	No	36	12	0.0412
HELO	Si	36	25	0.0693
HELO	Si	36	12	0.0569
S-HELO	No	36	25	0.0519
S-HELO	No	36	12	0.0428
S-HELO	Si	36	25	0.0684
S-HELO	Si	36	12	0.0558
HO	No	72	-	0.0401
HO	Si	72	-	0.074
HELO	No	72	25	0.0471
HELO	No	72	12	0.0422
HELO	Si	72	25	0.0645
HELO	Si	72	12	0.0567
S-HELO	No	72	25	0.0489
S-HELO	No	72	12	0.0374
S-HELO	Si	72	25	0.0652
S-HELO	Si	72	12	0.0535

Table 1: mAP

HO	No	96	-	0.0386
HO	Si	96	-	0.074
HELO	No	96	25	0.0441
HELO	No	96	12	0.0452
HELO	Si	96	25	0.0671
HELO	Si	96	12	0.0538
S-HELO	No	96	25	0.0447
S-HELO	No	96	12	0.0398
S-HELO	Si	96	25	0.0602
S-HELO	Si	96	12	0.0527

Table 2: mAP

5 Conclusión

Podemos ver claramente que los resultados con HELO y S-HELO son muy similares entre si y que consistentemente una reducción en el numero de bloques conlleva una caída en la mAP y en general sucede lo mismo con un aumento en el numero de bins del histograma.

También podemos ver que los resultados utilizando Canny sobre las imágenes son consistentemente mejores para todos los algoritmos.

Algo importante que mencionar es que HELO y S-HELO suelen ser mejores que HO hasta el momento en que aplicamos Canny. A partir de entonces, HO tiene una reacción interesante.

A este punto ya debería estar claro mirando los bajos números de la tabla y los detalles del párrafo anterior que existe algún problema posiblemente en nuestro algoritmo de evaluación o en la generación de los algoritmos, ya que nuestros resultados no reflejan los resultados expuestos en los papers de HELO y S-HELO.

Es necesario volver a la mesa de trabajo para investigar la causa de estas inconsistencias.

6 Referencias

1. HELO: "An Improved Histogram of Edge Local Orientations for Sketch-Based Image Retrieval", por Jose M. Saavedra y Benjamin Bustos.
2. S-HELO: "SKETCH BASED IMAGE RETRIEVAL USING A SOFT COMPUTATION OF THE HISTOGRAM OF EDGE LOCAL ORIENTATIONS (S-HELO)", por Jose M. Saavedra.