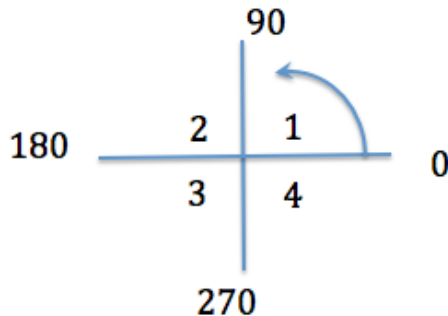


[Forums / Semana 4](#)[Help Center](#)

## Diversas preguntas HOG

[Subscribe for email updates.](#)☒ RESOLVEDNo tags yet. [+ Add Tag](#)Sort replies by: [Oldest first](#) [Newest first](#) [Most popular](#)[Francesc Sala](#) · 2 months ago 

1. En el tema de los ángulos ( $0 - 180$  o  $0 - 360$ ) yo estaba acostumbrado a ver el 0 empezando en la derecha (los cuadrantes iban del 1 al 4 en sentido anti-horario)



Pero entiendo que esto en realidad no tiene mayor importancia: puedo usar un criterio u otro siempre que lo use consistentemente. ¿Es así?

2. Para la contribución por pesos de la orientación, entiendo que el último intervalo debo combinarlo con el primero, tanto si hago  $0 - 360$  como si hago  $0 - 180$ . ¿Es así?

3. Dado un tamaño de celda  $n1 \times n2$ , los pesos de los píxeles para la interpolación especial son siempre los mismos en todas las celdas. Podría entonces tenerlos ya precalculados: tener ya precalculado para una celda "template" el producto de los pesos  $W_{ij}^{super x} * W_{ij}^{super y}$

Para ahorrar cálculos. ¿Tiene sentido?

4. También podría intentar lo mismo para las orientaciones del gradiente. Dado si trabajo con signo o sin signo y el número de intervalos, podría tener tabulado para todas las orientaciones posibles a qué dos intervalos contribuyen y con qué pesos. Tendría que precalcular  $256 \times 256$  posibles valores.

O bien ir guardando a medida que me fueran apareciendo ...

¿Tiene sentido hacerlo?

5. Veo que hacemos muchos cálculos en el HOG. Muchas veces sumando muchos numeros pequeños (porque multiplico por pesos  $< 1$ , porque además a menudo el gradient va a ser pequeño (en la imagen del gradient de la señora ya vemos que hay mucho negro: muchas zonas, la mayoría, donde la magnitud del gradient es muy pequeña) y además luego se normaliza ...

¿Puede introducirse un cierto error controlado al calcular el HOG para hacer menos cálculos?

Se entiende que:

- a) La pregunta quiere decir si se puede hacer eso sin perjudicar la calidad (exactitud y precisión no?) del clasificador resultante.
- b) Y que si hiciéramos esa introducción de error lo haríamos tanto en la fase de aprendizaje como en la de evaluación y en la posterior de explotación del sistema final. O sea, que el HOG lo calcularíamos siempre igual. Como fuera, pero siempre igual.

Simplificaciones que pensaba, del tipo:

- Si la longitud del gradiente es menor que un cierto threshold  $T$ , lo ignoramos. Ya no calculamos pesos ni nada, simplemente no lo tenemos en cuenta.
- Si por ejemplo trabajamos con 9 intervalos de dirección del gradiente, pues dividir cada uno de ellos en 4 y calcular cuales serían los pesos en el centro de esos 36 intervalos. Lo calculo una sola vez al inicio y lo guardo. Luego, para cada gradiente (para cada pixel) en lugar de calcular los pesos exactos, ver en cual de los 36 subintervalos cae y aplicar los pesos de aquel subintervalo (son una aproximación a los pesos exactos).

¿Tiene algún sentido o es más bien absurdo?

Muchas gracias. Cordialmente. f

↑ 0 ↓ · flag

**Ernest Valveny** INSTRUCTOR · 2 months ago 🔒

Hola Francesc,  
intento ir contestando punto por punto:

1. Tienes razón que normalmente estamos acostumbrados a dibujar los ángulos al revés. Es indiferente mientras seamos consistentes, como tú dices.
2. Sí, los ángulos de un extremo se enlazan con los del otro extremo.

3. Tiene todo el sentido del mundo precalcular el template que dices y aplicarlo para todas las celdas si queremos optimizar el cálculo

4. Podría tener sentido también para las orientaciones. No acabo de entender exactamente de dónde sale el número de valores que dices que deberías precalcular. Si entiendo bien lo que dices, serían 360 orientaciones por 9 intervalos.

5. Cualquier aproximación que hagas, como bien dices, habría que aplicarla por igual tanto al entrenar el modelo como al aplicarlo. No tener en cuenta gradientes por debajo de cierto threshold es una práctica utilizada en algunos casos. La discretización de la orientación del gradiente, no sé si tiene tanta utilidad porque si entiendo bien lo que planteas, igualmente para cada pixel hay que determinar en qué subintervalo cae y eso vendría a ser más o menos equivalente a calcular el peso tal como lo hemos explicado.

En los vídeos hemos explicado los conceptos del descriptor, con el foco puesto en intentar que se comprendan, sin entrar en detalles y trucos de implementación que existen para conseguir un cálculo eficiente. Algunos se basan en las ideas que tú planteas.

Un saludo

↑ 0 ↓ · flag

[Francesc Sala](#) · 2 months ago 🔒

Hola. Moltes gràcies Ernest.

En 4 pensaba en  $256 \times 256$  por las posibles diferencias en  $x$  i en  $y$  ( $dx$  i  $dy$  de cada pixel, calculado como  $dx(i,j) = I(i+1,j) - I(i-1,j)$  y el  $dy$  de forma similar). Siendo la imagen de 0 a 255 valores de intensidad posible. Vamos que propiamente ya no pensaba en cada pixel calcular el arco tangente, porque tabulaba las  $256 \times 256$  casos posibles de las diferencias.

Si no, pues tabularia 360 casos y en los pixeles si que calcularia el arco tangente.

Bueno, no sé si ahora se me entiende mejor. Es que explicarlo por escrito se me hace un poco difícil.

Gracias de nuevo por tus respuestas tan detalladas.

↑ 0 ↓ · flag

---

[+ Comment](#)

[Ernest Valveny](#) INSTRUCTOR · 2 months ago 🔒

Hola,

ahora lo entiendo mejor y sí, permitiría optimizar algo el tiempo a cambio de ocupar más memoria.

Habría que evaluar exactamente el impacto que tiene en el tiempo de cálculo.

↑ 0 ↓ · flag

---

[+ Comment](#)

New post

To ensure a positive and productive discussion, please read our [forum posting policies](#) before posting.

<b>B</b>	<i>I</i>			Link	<code>&lt;code&gt;</code>	Pic	Math		Edit: Rich ▼	Preview
<div></div>										

This thread is marked as resolved. Staff are no longer monitoring this thread. If the problem is not fixed, please start a new thread [here](#) to let staff know that there is still a problem.

- ☐ Make this post anonymous to other students
- ☒ Subscribe to this thread at the same time

Add post

