

Tarea 1 - Procesamiento y Análisis de Imágenes

Sebastián Cassone

*Departamento de Ingeniería Informática
Universidad de Santiago de Chile*

Motivación—El procesamiento de imágenes emplea operadores como Canny, Sobel y DoG para detectar bordes, áreas donde la intensidad cambia abruptamente. Herramientas como Photoshop utilizan estos detectores para crear efectos artísticos, como bocetos. El operador DoG, que implica la diferencia entre versiones borrosas de una imagen, es comúnmente utilizado. Una extensión conocida como XDOG permite una variedad de efectos artísticos adicionales en las imágenes.

La tarea consiste en implementar en MATLAB el operador XDOG junto con una binarización de la imagen.

I. SOLUCIÓN PROPUESTA

Para resolver el problema planteado, se emplean las ecuaciones provistas en el enunciado de la tarea, tal como se muestra en la Figura I. La primera ecuación calcula la diferencia Gaussiana entre la imagen convolucionada con un filtro Gaussiano de parámetro sigma, que indica la desviación estándar de la gaussiana utilizada en la detección de bordes. Posteriormente, esta imagen convolucionada se resta de una versión filtrada con un sigma multiplicado por una variable K, que ajusta la anchura de los bordes detectados.

Luego, a esta imagen resultante se le aplica la segunda ecuación de la Figura I, que implica Phi, un umbral para la detección de bordes. Ajustar estos valores altera la sensibilidad del detector de bordes; un valor más alto de phi conduce a una detección de bordes más sensible. También se menciona Epsilon, con valores negativos como en este caso (-0.1), que pueden aumentar el contraste de los bordes, contribuyendo así a la imagen final con el filtro XDOG.

Finalmente, para binarizar la imagen, se utiliza como umbral el promedio de todos los píxeles, y se binariza la imagen por debajo de este umbral.

$$D_0(\hat{x}, \sigma, k, I) = G(\hat{x}, \sigma, I) - \Gamma.G(\hat{x}, \sigma, k, I) \quad (4)$$

$$E(X(\sigma, k, \Gamma, \epsilon, \Phi)) = \begin{cases} 1 & \text{if } D_0(\sigma, k, \Gamma, I) < \epsilon \\ 1 + \tanh(\Phi.(D_X(\sigma, k, \Gamma, I))) & \text{otherwise} \end{cases}$$

Figura 1: Ecuaciones propuestas

II. EXPERIMENTOS REALIZADOS

Se consideraron dos imágenes propuestas en la tarea junto con dos imágenes obtenidas de un motor de búsqueda para fines de prueba. Se mantuvieron constantes los valores de gamma, epsilon y k para asegurar condiciones consistentes en los experimentos.

En el primer experimento, con gamma = 0.98; phi = 150; epsilon = -0.1; k = 1.6; sigma = 0.9, se observó que aumentar

el valor de Phi generaba una mayor sensibilidad de borde, pero deterioraba la calidad de la imagen.

En el segundo experimento, con gamma = 0.98; phi = 300; epsilon = -0.1; k = 1.6; sigma = 0.7, al aumentar Phi y reducir Sigma, la imagen del presidente mejoraba progresivamente.

En el tercer experimento, con gamma = 0.98; phi = 5; epsilon = -0.1; k = 1.6; sigma = 0.8, debido al tipo de imagen, no se encontró un valor óptimo para Phi, ya que valores altos detectaban objetos no deseados. Variar Sigma también afectaba la percepción de la imagen, destacando el entorno sobre el objeto principal, por lo que se optó por valores bajos.

En el último experimento, con gamma = 0.98; phi = 2; epsilon = -0.1; k = 1.6; sigma = 0.6, se enfrentó al mismo problema que en el tercer experimento debido al fondo de las imágenes. En comparación con las primeras imágenes que no presentaban este inconveniente.

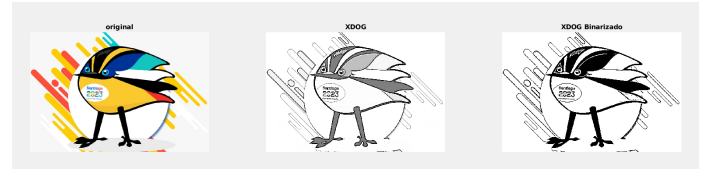


Figura 2: Experimento 1



Figura 3: Experimento 2

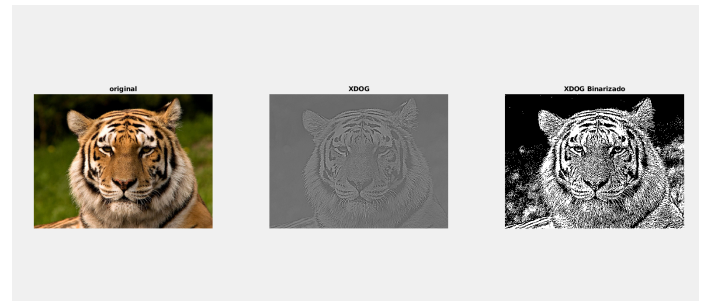


Figura 4: Experimento 3

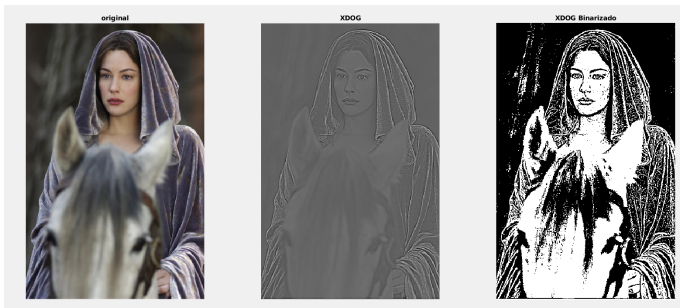


Figura 5: Experimento 4

III. CONCLUSIONES

En conclusión, el procesamiento de imágenes ofrece una variedad de herramientas y técnicas para detectar bordes y crear efectos artísticos. El operador XDoG, una extensión del operador DoG, es especialmente útil para esta tarea. Implementar este operador junto con la binarización de la imagen en MATLAB ofrece una forma eficaz de explorar y aplicar estos conceptos en proyectos de procesamiento de imágenes.