

## PRÁCTICA 7. PROPAGACIÓN DE FRENTES.

Implementar la función

```
void aan_ecuacion_propagacion_frentes(float *red_input, float *green_input, float  
*blue_input, float *red_output, float *green_output, float *blue_output, int width, int  
height, float dt, int Niter, float *F)
```

que realiza Niter iteraciones de la discretización de la propagación de frentes tomando como incremento temporal dt. Es decir hay que implementar para cada canal el esquema numérico:

$$\frac{du}{dt} = F \|\nabla u\|$$

Que, discretizado, queda:

$$u_{i,j}^{n+1} = u_{i,j}^n + F_{i,j} dt \left( \sqrt{\left( (u_x)_{i,j}^n \right)^2 + \left( (u_y)_{i,j}^n \right)^2} \right)$$

Si bien u representa al frente, necesitamos otro canal para fijar la velocidad o fuerza de propagación del mismo. En los ejemplos proporcionados, el frente será el canal rojo y la velocidad la fijará el canal verde.

Hacer al menos dos pruebas:

1. Con una velocidad inversa a la magnitud de un canal. En este caso, el canal verde determina, en cada píxel, cuál es la velocidad de propagación. Si el canal verde está al máximo, no hay propagación:

$$F_{i,j} = k(255 - v_{i,j})/255$$

K sirve para controlar lo rápida que es la propagación en general.

2. Con la velocidad inversa a la magnitud del gradiente. En este caso, calculamos el gradiente usando la práctica 2 (de la imagen completa o sólo de un canal, según el caso) y lo usamos para determinar la velocidad:

$$F_{i,j} = k(1 - V)$$

$$V = \min(1, \|\nabla v\|_{i,j} / M)$$

Donde M determina a partir de qué valor del módulo del gradiente se frena completamente la propagación.