

PRÁCTICA 8. IMPLEMENTAR LA ECUACIÓN DE ONDAS.

Implementar la función

```
void aan_ecuacion_ondas_metodo_explicito(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)
```

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

$$u_{i,j}^{n+1} = 2u_{i,j}^n - u_{i,j}^{n-1} + (dt)^2 (u_{i+1,j+1}^n + u_{i-1,j-1}^n + u_{i+1,j-1}^n + u_{i-1,j+1}^n + u_{i+1,j}^n + u_{i-1,j}^n + u_{i,j+1}^n + u_{i,j-1}^n - 8u_{i,j}^n)$$

Inicialmente conocemos $u_{i,j}^0$ y $u_{i,j}^1$. Para los experimentos supondremos que $u_{i,j}^0 = u_{i,j}^1$.

Las pruebas que se harán son las siguientes :

Test 1: Tomar $dt=0.1$, y 1000 iteraciones tomar una imagen formada por varios cuadrados (o círculos) de distintos colores sobre un fondo de nivel de gris 128. (tomar una imagen de dimensión pequeña (128x128) para que tengamos espacio en el disco para salvar la secuencia. Crear la secuencia de imágenes y crear la película. Hacer lo mismo para $dt=0.01$ y 1000 iteraciones. Hay que tener en cuenta que en este caso, las oscilaciones producidas por la ecuación de ondas pueden provocar que los niveles de gris superen el 255 o bajen por debajo de 0. Por tanto para pasarla al tipo unsigned char antes de guardarlas en disco tendremos que “cortar” las imágenes de tal manera que si el valor del nivel de gris es mayor de 255, le asignamos el valor 255, y si está por debajo de 0, le asignamos 0.

Test 2: Hacer lo mismo tomando una imagen real a ver que es lo que sale.