Suavizado de Bordes en Imágenes (Marzo 2010)

Iván López Espejo

Este documento recoge una forma de abordar el planteamiento y construcción de un sistema para el suavizado (especialmente de bordes) de imágenes digitales.

I. Introducción y Desarrollo

Este documento recoge una forma de abordar el planteamiento y construcción de un sistema para el suavizado (especialmente de bordes) de imágenes digitales.

La idea del sistema es bien sencilla y está basada en minimización de gradiente. El método se encarga de iterar el siguiente algoritmo:

1. Para el píxel *i-ésimo* de la imagen (con $i = 1,2,\dots,F \times C$) almacenamos en un vector el valor de las muestras diferenciales calculadas como la diferencia del nivel de gris del píxel *i-ésimo* con cada uno de los niveles de gris de los píxels de su vecindad.
2. Mediante comparación, para cada uno de los valores del anterior vector construido (en valor absoluto), estudiamos si alguno de ellos excede el valor diferencial máximo impuesto por el parámetro de entrada, donde este valor diferencial máximo se obtiene como

$$dif_{max} \equiv (1 - parámetro) \times (Niveles - 1),$$

donde *Niveles* hace referencia al número de niveles de gris de la imagen (por defecto *Niveles* = 256) y *parámetro* es un valor real en el intervalo [0,1], donde 0 indicaría suavizado mínimo (ya que establecería la diferencia máxima de niveles de gris entre dos píxels contiguos en 255) y 1 haría lo propio referenciando suavizado máximo (ya que no permitiría diferencia de niveles de gris ninguna en la imagen, lo que se traduce en una imagen final de un nivel de gris continuo).

3. Si ninguno de los valores del vector, en valor absoluto, excede el diferencial máximo de niveles de gris, se vuelve al paso 1 hasta completar el recorrido por la imagen. Si alguno sí supera esta cota, calculamos el nuevo valor de gris del píxel a partir del píxel vecino que provoca la máxima diferencia (gradiente), de la forma,

$$I(i) = I(i) + signo \times (dif_{max} - gradiente),$$

donde *signo* controla el signo de la resta entre el píxel *i-ésimo* y uno de los vecinos.

4. Una vez se hayan recorrido todos los píxels de la imagen con este procedimiento, se repetirá el algoritmo tantas veces como sea necesario hasta que en la *n-ésima* iteración no se haya producido ningún cambio en la imagen.

De lo anterior se deduce que el algoritmo ha de ser convergente en un número finito de pasos. No se ha procedido a su demostración matemática, aunque parece intuitivo que esto sea así. El caso que sí podría poner en peligro la convergencia del método, sería el de una imagen que no tuviese bordes, como se daría en la situación de una imagen continua proyectada sobre una superficie esférica.

Notar que el concepto de vecindad se basa en un 8-entorno, así como que el comportamiento de la implementación en los bordes pasa por considerar únicamente los píxels reales contiguos (es decir, en el caso de un píxel en una esquina, el entorno estaría compuesto por 3 píxels).

El sistema puede resultar interesante en el caso de querer alisar imágenes ligeramente pixeladas o con bordes muy pronunciados, como puede ser el caso de un texto. Como ejemplo, se generó el siguiente texto en Paint:

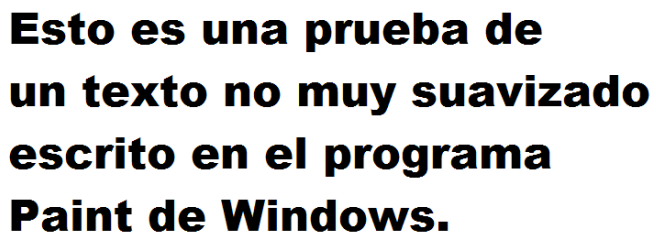

Fig. 1. Texto de ejemplo algo pixelado en los bordes.

A continuación mostramos el efecto de suavizado con un coeficiente de 0.9. Notar el curioso efecto potencialmente aplicable a otros menesteres con coeficientes de suavizado altos:

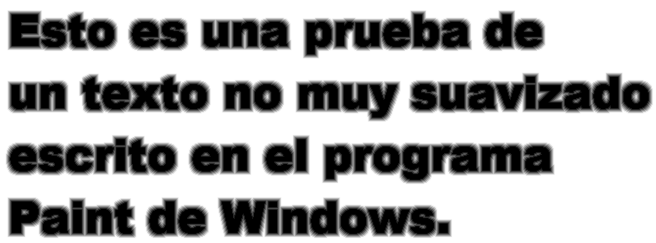

Fig. 2. Texto de ejemplo suavizado con coeficiente 0.9.