DEPARTAMENTO DE COMPUTACION
Facultad de Ciencias Exactas y Naturales - UBA

Trabajo Práctico III

Marche un telebeam Don Niembraaaaaa...

Métodos Numéricos
Primer Cuatrimestre - 2015

Integrante	LU	Correo electrónico

Facultad de Ciencias Exactas y Naturales
Universidad de Buenos Aires
Ciudad Universitaria - (Pabellón I/Planta Baja)
Intendente Güiraldes 2160 - C1428EGA
Ciudad Autónoma de Buenos Aires - Rep. Argentina
Tel/Fax: (54 11) 4576-3359
http://www.fcen.uba.ar

Índice

Resumen

En este trabajo se utilizaran distintas técnicas para obtener un re-escalamiento de imágenes. Se utilizara vecino más cercano, interpolación de polinomios bilineal, splines cúbicos, y distintas variantes de los métodos anteriormente mencionados. Se implementaran algoritmos para los mismos, dando la posibilidad de re-escalar las imágenes en distintos tamaños(siempre mayor al original). Se llevara a cabo una experimentación con su respectivo análisis. Como las imágenes obtenidas, no contienen integramente información original, se utilizaran las métricas de Error Cuadrático Medio (ECM) y Peak to Signal Noise Ratio (PSNR) para estudiar en forma cuantitativa la calidad de las mismas. También se considerara la calidad subjetiva, y el tiempo de computo.

Palabras Clave: re-escalamiento imágenes, interpolación, ECM, PSNR

1. Intodución

En el presente trabajo se nos plantea como objetivo el re-escalamiento de imágenes, específicamente ampliarlas.

Se trabajara con imágenes en escala de grises por lo que dada una imagen de mxn, contendrá mxn pixels cada uno con un valor entre 0-255.

Para ampliar las imágenes, a partir de un valor $k \in \mathbb{N}_{>0}$ insertaremos entre la fila i y la fila i +1, k filas con i = 1, ..., m -1, y de manera análoga para las columnas. De esta forma obtendremos una nueva imagen con $(m-1)*k+m$ filas y $(n-1)*k+n$ columnas.

El problema que ahora se genera es ¿que valores asignar a los pixels de las columnas y filas agregadas?. Para esto, emplearemos distintos criterios de asignación de valor a partir de ciertos métodos.

En primera instancia consideraremos el método de vecino mas cercano, para esto el valor de un píxel nuevo sera igual a aquel cuya distancia a otro píxel, en una vecindad definida, sea la mínima. En particular la vencidad tomada será de aquellos cuatro valores más cercano con respecto a los pixels originales (pixels de la imagen sin ampliar, $imagen\ original$).

Otro método empleado fue mediante la interpolación de polinomios, esta consiste en que dada una terna de puntos $(x_0,\ y_0)$, $(x_1,\ y_1)$, $\ldots$ $(x_n,\ y_n)$ se busca obtener un polinomio P que los interpole es decir, que verifique $p(x_0) = y_0$, $p(x_1) = y_1$, $\ldots$ $p(x_n) = y_n$. En general, la interpolación de una serie de puntos es usada para aproximar una función continua en un cierto intervalo. Dado que siempre existe un polinomio interpolador para $n+1$ puntos, de grado a lo sumo n que los interpole[1] , una de la forma de obtenerlo es mediante el método de interpolación de Lagrange, el cual se basa en construir primero los polinomios $L_{n,k}$ definidos como se indica en la ecuación 1.

$$L_{n,k} = \prod_{\substack{i=0\\ i\neq k}}^{n} \frac{(x-x_i)}{(x_k-x_i)} \tag{1}$$

El polinomio de grado a lo sumo n que interpola los $n+1$ puntos se construye según la ecuación 2.

$$P(x) = \sum_{k=0}^{n} y_k L_{n,k}(x) \tag{2}$$

Por lo que el segundo método empleado consiste en usar interpolación bilineal entre dos puntos $(x_0,\ y_0)$ y $(x_1,\ y_1)$, en nuestro caso entre dos pixels, por lo que el polinomio interpolador de Lagrange sera de grado a lo sumo uno es decir, será una recta que pasa por dos puntos. En la ecuación 3

$$P(x) = L_{1,0}(x)y_0 + L_{1,1}(x)y_1 = \frac{x-x_1}{x_0-x_1}y_0 + \frac{x-x_0}{x_1-x_0}y_1 \tag{3}$$

o equivalentemente obtenemos una formula mas clara para el mismo, donde además podemos distinguir la pendiente y la ordenada al origen.

$$P(x) = \frac{y_1-y_0}{x_1-x_0}(x-x_0) + y_0 \tag{4}$$

Ses $f(x_i) = y_i$, en nuestro caso el valor de f depende de dos puntos, por lo que el valor del píxel p_{ij} se obtendrá extendiendo la ecuación 4 a:

$$p(i,j) = \frac{f(i,j_1) - f(i,j_0)}{j_1 - j_0} + f(i,j_0) \tag{5}$$

Otro de los métodos utilizados es el de splines cúbicos. Dada una función f definida en $[a,b]$ y un conjunto de puntos $a = x_0 < x_1 < \ldots < x_n = b$ un trazador cúbico S para f es una función tal que en cada subintervalo $[x_j, x_{j+1}]$ con $j = 0,1,\ldots,n-1$ $S_j(x)$ es un polinomio cubico, y verifica:

- $S(x_j) = f(x_j)\ \forall\ j = 0,1,\ldots,n$
- $S_{j+1}(x_{j+1}) = S_j(x_{j+1})\ \forall\ j = 0,1,\ldots,n-2$

- $S'_{j+1}(x_{j+1}) = S'_j(x_{j+1}) \; \forall \; j = 0, 1, \ldots, n-2$
- $S''_{j+1}(x_{j+1}) = S''_j(x_{j+1}) \; \forall \; j = 0, 1, \ldots, n-2$
- Y una de las siguientes condiciones
  - $S''(x_0) = S''(x_n) = 0 \; \forall \; j = 0, 1, \ldots, n-2$ (condición natural)
  - $S'(x_0) = f'(x_0) y S'(x_n) = f'(x_n)$ (condición sujeta)

Escribiendo a los S_j en la forma $S_j(x) = a_j + b_j(x - x_j) + c_j(x - x_j)^2 + d_j(x - x_j)^3$, y planteando las condiciones anteriormente mencionadas se puede obtener un sistema lineal de $n + 1$ ecuaciones y $n + 1$ incógnitas, donde estas son los c_j. En el caso de la condición natural, que fue el utilizado en este trabajo práctico, mas específicamente se obtiene:

$$A = \begin{bmatrix} 1 & 0 & 0 & \cdots & \cdots & \cdots & 0 \\ h_0 & 2(h_0 + h_1) & h_1 & \ddots & & & \vdots \\ 0 & h_1 & 2(h_1 + h_2) & h_2 & \ddots & & \vdots \\ \vdots & & \ddots & \ddots & \ddots & & \vdots \\ \vdots & & & \ddots & \ddots & \ddots & 0 \\ \vdots & & & & h_{n-2} & 2(h_{n-2} + h_{n-1}) & h_{n-1} \\ 0 & \cdots & \cdots & \cdots & 0 & 0 & 1 \end{bmatrix}$$

$$b = \begin{bmatrix} 0 \\ \frac{3}{h_1}(a_2 - a_1) - \frac{3}{h_0}(a_1 - a_0) \\ \vdots \\ \frac{3}{h_{n-1}}(a_n - a_{n-1}) - \frac{3}{h_{n-2}}(a_{n-1} - a_{n-2}) \\ 0 \end{bmatrix} ; \quad c = \begin{bmatrix} c_0 \\ c_1 \\ \vdots \\ c_n \end{bmatrix}$$

con $h_j = x_{j+1} - x_j$.

donde A es una matriz estrictamente diagonal dominante. Esto último implica que la matriz es invertible y por lo tanto el sistema tiene solución única. Una vez determinado c, se pueden obtener los a_j,b_j y d_j.

A partir de los resultados obtenidos en cada método buscaremos introducir alguna modificación en los mismos con el fin de obtener alguna mejora temporal y/o cualitativa. La forma en que se medirá la cálidad de la imagen obtenida será a través de el *Error Cuadrático Medio* (ECM) y *Peak to Signal Noise Ratio* (PSNR) los mismos se definen como:

$$\texttt{ECM}(I, \bar{I}) = \frac{1}{mn} \sum_{i=1}^{m} \sum_{j=1}^{n} |I_{ij} - \bar{I}_{ij}|^2 \tag{6}$$

y

$$\texttt{PSNR}(I, \bar{I}) = 10 \log_{10} \left(\frac{255^2}{\texttt{ECM}(I, \bar{I})} \right). \tag{7}$$

Con I e $\bar{I}$ la imagen original y la ampliada respectivamente, de dimensiones mxn. Como para útilizar esta métrica es necesario que las imágenes tengan igual dimensiones, aquellas con las que trabajaremos serán reducidas y luego ampliadas, con los métodos con los que trabajemos, a su tamaño original.

2. Desarollo

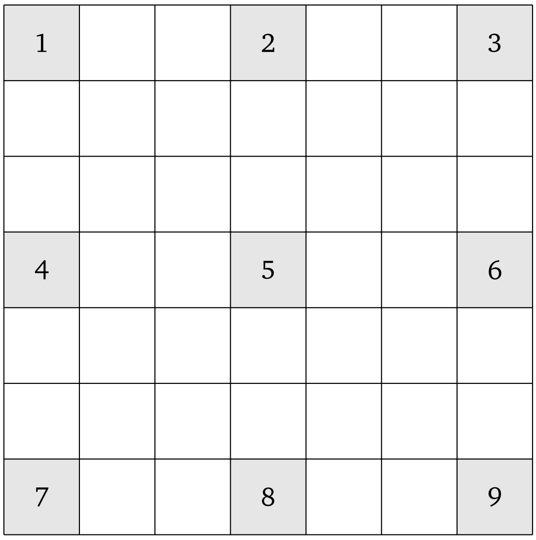

Figura 1: Imagen re-escalada

3. Experimentación

3.1. Spline:

Para el caso anterior el tamaño de los bloques era fijo, si teníamos dos puntos conocidos $P_{i,j}$ y $P_{i,j+1}$, dentro del bloque con el que trabajamos, los valores intermedios serian calculados creando un spline por fila para estos puntos. Pero, ¿Que sucede si el valor de ambos difieren de manera considerable? Esto podría implicar que uno es muy claro y el otro no luego, al momento de ampliar la imagen aquellos nuevos pixels que se encuentren entre los $P_{i,j}$ y $P_{i,j+1}$ se verán afectados por estos dos, pudiendo tener un valor intermedio a estos.

Supongamos que ambos puntos representaban dos objetos distintos o se trataba de sus de bordes, entonces en la imagen ampliada nos gustaría que aun siga existiendo esta distinción, por ejemplo si se pasa de un color blanco a otro negro no nos gustaría que los pixels intermedio queden en alguna otra tonalidad. Con los splines cúbicos no siempre se puede evitar esto, aun cuando el tamaño del bloque no sea el de toda la imagen deberíamos tomar bloque muy pequeños para poder evitarlo. Por lo que para solucionar esto propusimos que el tamaño de un bloque, sea al comenzar, desde el primer punto conocido de la fila $P_{i,0}$ con la que se trabaja, hasta el punto anterior a un $P_{i,t}$ talque $\mid f(P_{i,0}) - f(P_{i,t}) \mid \leqslant$ a (siendo f la función que retorna el valor de un pixel p) con a un valor determinado. Luego, con la misma idea el bloque se tomara desde $P_{i,t}$ hasta un $P_{i,t+r}$ o hasta el final de la fila. Notemos que al igual que en el caso anterior una vez que se completan las filas podemos aplicar la misma idea a las columnas y así obtener completar los valores.

Pero entonces al utilizar spline cúbicos para interpolar por ejemplo desde $P_{i,0}$ a $P_{i,t-(k+1)}$ y otro para $P_{i,t-(k+1)}$ a $P_{i,t+r}$ los valores entre $P_{i,t-(k+1)}$ y $P_{i,t}$ aun no tienen ningún valor asignado, como queremos que en la imagen ampliada estos sean exactamente igual a $P_{i,t-(k+1)}$ o a $P_{i,t)}$ y no un valor intermedio entre estos. Optaremos por completar a los mismos siguiendo la idea de vecino mas cercano definiendo como vecindad a los valores originales mas cercano.

Para este caso el tamaño de los bloques a tomar esta ligado a que valor de a utilizar, notemos que si usamos valores muy chicos, es decir permitimos una diferencia mínima, posiblemente terminemos aplicando el método de vecino mas cercano en casi toda la imagen y perdamos la aplicación de spline.

Mientas que por el contrario, si permitimos diferencias muy grandes entonces caeríamos en la implementación anterior (solo splines). Para comprobar si esto sucede experimentaremos con varios valores y veremos si existe algún valor o rango de valores óptimos el cual hace reducir el ECM y si estas variaciones influyen o no en el tiempo de computo.

4. Conclusiones

Referencias

[1] Burden R. L., Faires J.D.- Numerical Analysis