



Ciencia y Tecnología

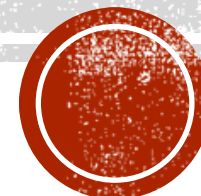
Secretaría de Ciencia, Humanidades, Tecnología e Innovación



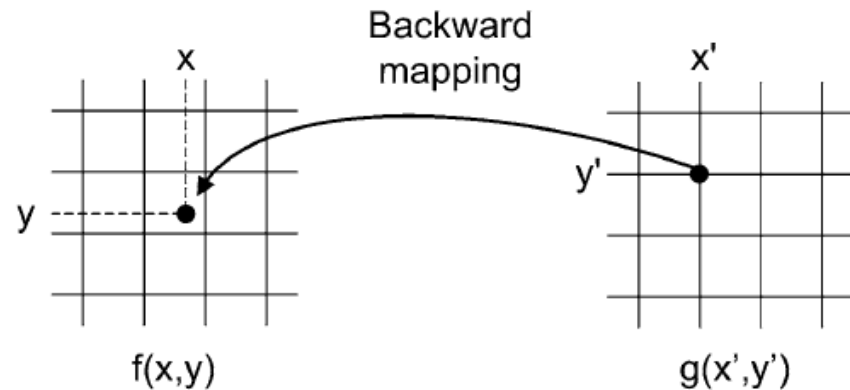
CIMAT
UNIDAD MÉRIDA

INTERPOLACIÓN

Dr. Francisco J. Hernández López
SECIHTI – CIMAT-Mérida
fcoj23@cimat.mx, www.cimat.mx/~fcoj23



MAPEO INVERSO

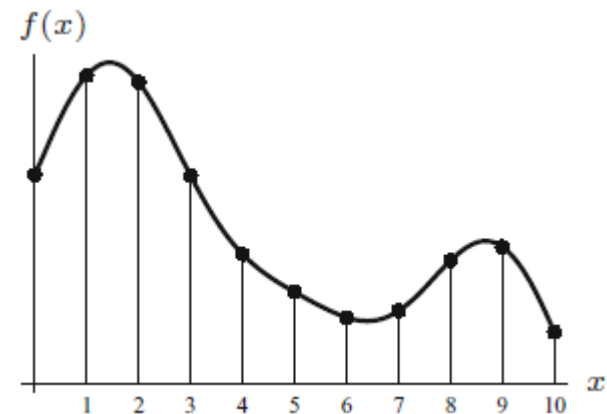
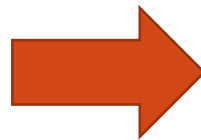
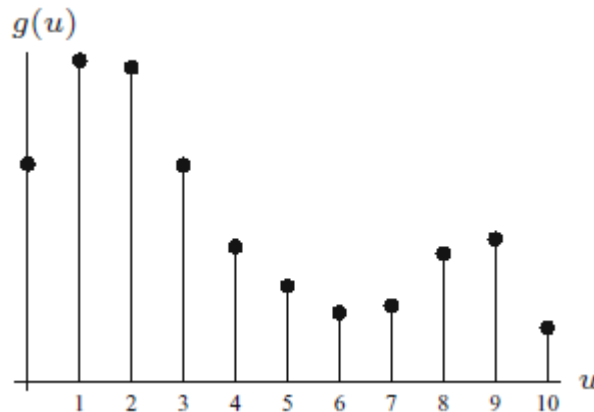


```
Image_Width_Output = Image_Width_Input * Sx;  
Image_Height_Output = Image_Height_Input * Sy;  
for (y_output = 0; y_output < Image_Height_Output;  
    y_output++)  
{  
    for (x_output = 0; x_output < Image_Width_Output;  
        x_output++)  
    {  
        x_input = 1/Sx * x_output;  
        y_input = 1/Sy * y_output;  
        g(x_output, y_output) = f(x_input, y_input);  
    }  
}
```

Si no son posiciones enteras,
podríamos necesitar una interpolación

INTERPOLACIÓN

- Estimar el valor de intensidad de un pixel en localidades no enteras a partir de las localidades cercanas que si tienen un valor de intensidad definido en la imagen discreta
- Dada una función discreta g , el objetivo es estimar los valores de la función original f en cualquier posición $x \in \mathbb{R}$



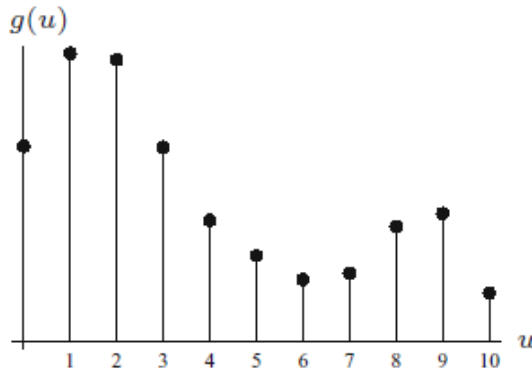
Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer.
ISBN: 978-1-84628-968-2

Interpolación. Francisco J. Hernández-López

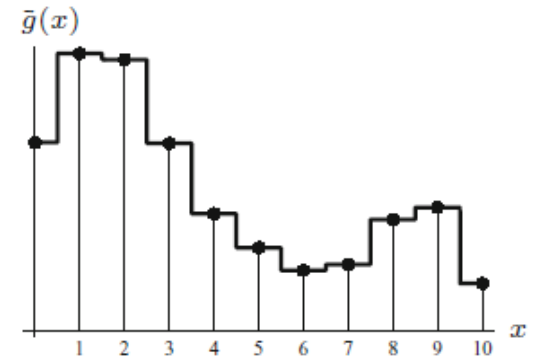
Ene-Jun 2025

MÉTODOS DE INTERPOLACIÓN SIMPLE 1D

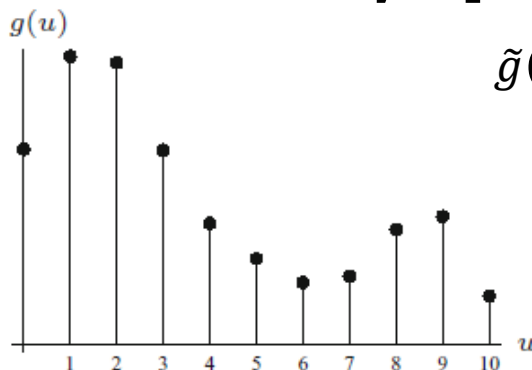
- **Vecinos más cercanos (*nearest-neighbor*).** Redondear las coordenadas al entero más cercano



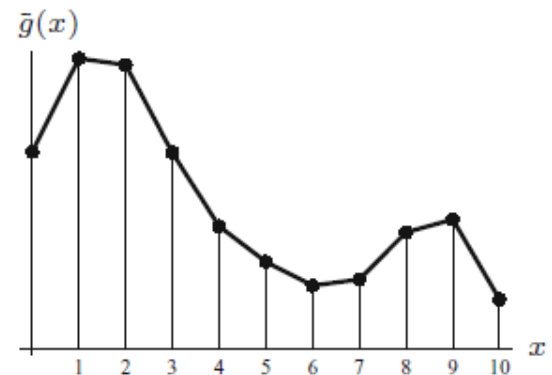
$$\begin{aligned}\tilde{g}(x) &\leftarrow g(u_x) \\ u_x &= \text{round}(x) \\ &= \lfloor x + 0.5 \rfloor\end{aligned}$$



- **Lineal.** Calcular la suma de los dos valores más cercanos dando un mayor peso al valor que está más cerca de x



$$\begin{aligned}\tilde{g}(x) &= (1 - (x - u_x))g(u_x) \\ &\quad + (x - u_x)g(u_x + 1) \\ u_x &= \lfloor x \rfloor\end{aligned}$$



Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer. ISBN: 978-1-84628-968-2

Interpolación. Francisco J. Hernández-López

Ene-Jun 2025

MÉTODOS DE INTERPOLACIÓN SIMPLE 2D

- **Vecinos más cercanos (*nearest-neighbor*).** Redondear las coordenadas al entero más cercano

$$\tilde{I}(x, y) = I(u_x, v_y)$$

$$u_x = \text{round}(x) = \lfloor x + 0.5 \rfloor$$

$$v_y = \text{round}(y) = \lfloor y + 0.5 \rfloor$$

- **Bilinear.** Calcular la suma pesada de los cuatro valores más cercanos dando un mayor peso al valor que está más cerca de (x, y)

$$A = I(u_x, v_y), \quad B = I(u_x + 1, v_y),$$

$$C = I(u_x, v_y + 1), \quad D = I(u_x + 1, v_y + 1),$$

$$u_x = \lfloor x \rfloor, v_y = \lfloor y \rfloor,$$

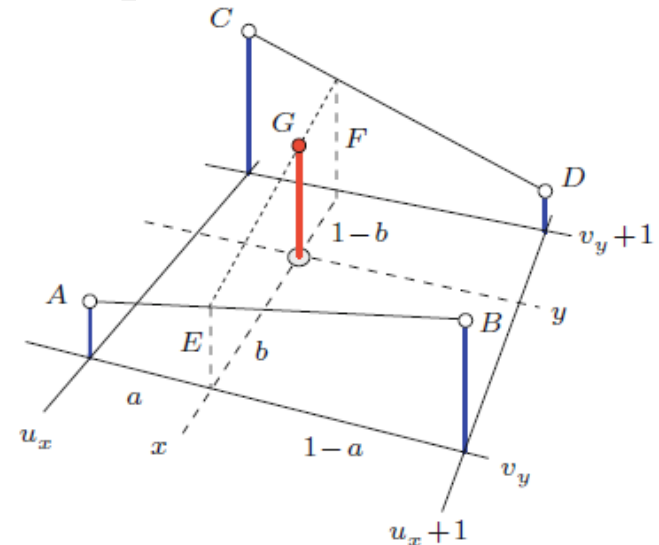
$$a = x - u_x, b = y - v_y,$$

$$E = A + a(B - A), F = C + a(D - C),$$

$$\tilde{I}(x, y) = G = E + b(F - E)$$

$$= (1 - a)(1 - b)A + a(1 - b)B$$

$$+ (1 - a)bC + abD$$



Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer.
ISBN: 978-1-84628-968-2

INTERPOLACIÓN CÚBICA

- Se puede definir como una convolución con el siguiente kernel:

$$w_{cub}(x, a) = \begin{cases} (a + 2)|x|^3 - (a + 3)|x|^2 + 1 & \text{para } 0 \leq |x| < 1, \\ a|x|^3 - 5a|x|^2 + 8a|x| - 4a & \text{para } 1 \leq |x| < 2, \\ 0 & \text{para } |x| \geq 2 \end{cases}$$

[Keys, 1981] mostró que con $a = -1/2$ se alcanza una convergencia de tercer orden con respecto al intervalo de muestreo de la función original, para cualquier otra elección, se alcanzará a lo más una aproximación de primer orden

$$w_{cub}\left(x, -\frac{1}{2}\right) = \begin{cases} \frac{3}{2}|x|^3 - \frac{5}{2}|x|^2 + 1 & \text{para } 0 \leq |x| < 1, \\ -\frac{1}{2}|x|^3 + \frac{5}{2}|x|^2 - 4|x| + 2 & \text{para } 1 \leq |x| < 2, \\ 0 & \text{para } |x| \geq 2 \end{cases}$$

Keys, R. (1981). Cubic convolution interpolation for digital image processing. IEEE transactions on acoustics, speech, and signal processing, 29(6), 1153-1160.

INTERPOLACIÓN BICÚBICA

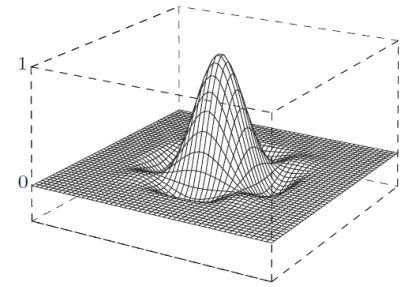
- Se puede definir como una convolución con el siguiente kernel:

$$W_{bic}(x, y) = w_{cub}(x)w_{cub}(y)$$

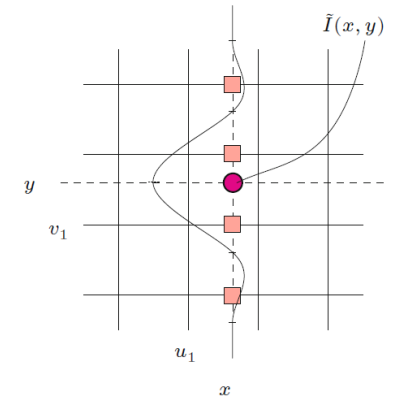
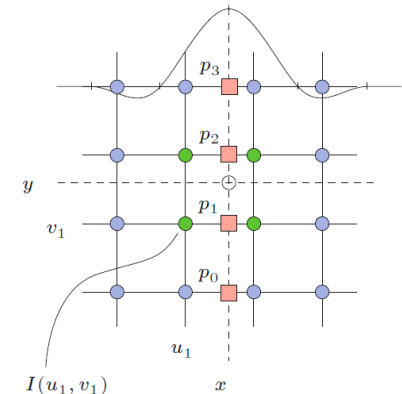
- El cálculo es separable en x y y :

$$\begin{aligned} \tilde{I}(x, y) &= \sum_{v=\lfloor y \rfloor - 1}^{\lfloor y \rfloor + 2} \left[\sum_{u=\lfloor x \rfloor - 1}^{\lfloor x \rfloor + 2} I(u, v) W_{bic}(x - u, y - v) \right] \\ &= \sum_{j=0}^3 \left[w_{cub}(y - v_j) \sum_{i=0}^3 I(u_i, v_j) w_{cub}(x - u_i) \right] \end{aligned}$$

con $u_i = \lfloor x_0 \rfloor - 1 + i$, $v_j = \lfloor y_0 \rfloor - 1 + j$.



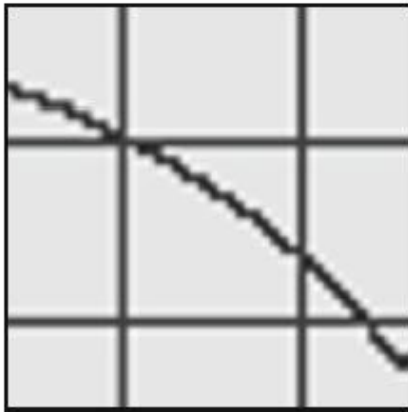
(b) W_{bic}



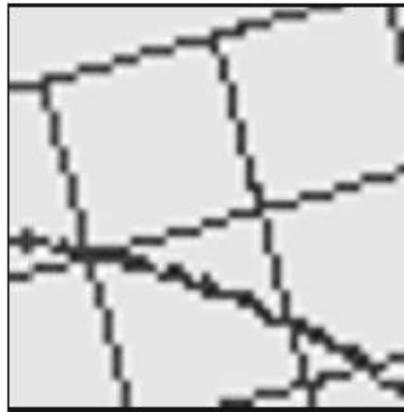
ALGORITMO DE INTERPOLACIÓN BICÚBICA

```
1: BicubicInterpolation( $I, x, y, a$ )  
   Input:  $I$ , original image;  $x, y \in \mathbb{R}$ , continuous position;  $a$ , control parameter. Returns the interpolated image value at position  $(x, y)$ .  
2:  $q \leftarrow 0$   
3: for  $j \leftarrow 0, \dots, 3$  do ▷ iterate over 4 lines  
4:      $v \leftarrow \lfloor y \rfloor - 1 + j$   
5:      $p \leftarrow 0$   
6:     for  $i \leftarrow 0, \dots, 3$  do ▷ iterate over 4 columns  
7:        $u \leftarrow \lfloor x \rfloor - 1 + i$   
8:        $p \leftarrow p + I(u, v) \cdot w_{\text{cub}}(x - u, a)$   
9:      $q \leftarrow q + p \cdot w_{\text{cub}}(y - v, a)$   
10: return  $q$ 
```

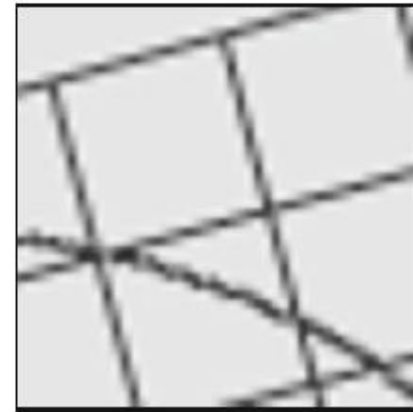
COMPARACIÓN (LÍNEAS)



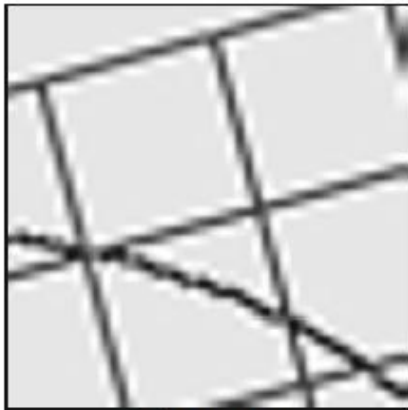
(a) Original



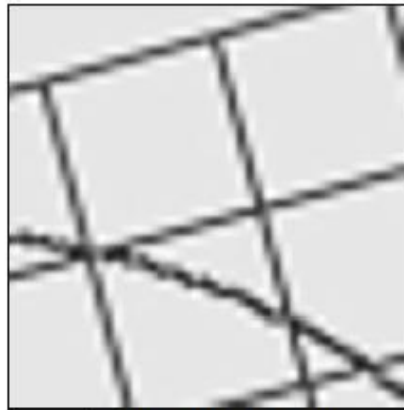
(b) Nearest-neighbor



(c) Bilinear



(d) Bicubic



(e) Catmull-Rom



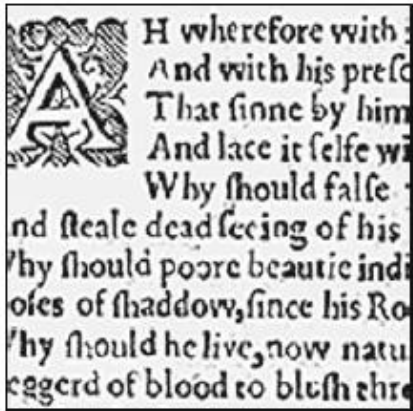
(f) Cubic B-spline

Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer.
ISBN: 978-1-84628-968-2

Interpolación. Francisco J. Hernández-López

Ene-Jun 2025

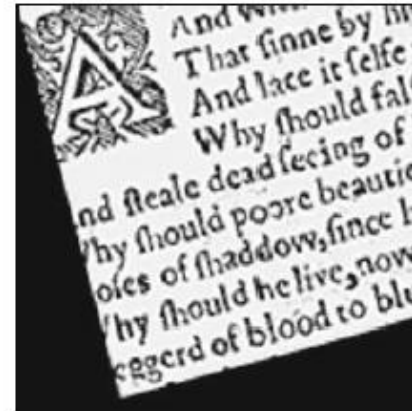
COMPARACIÓN (TEXTO)



(a) Original



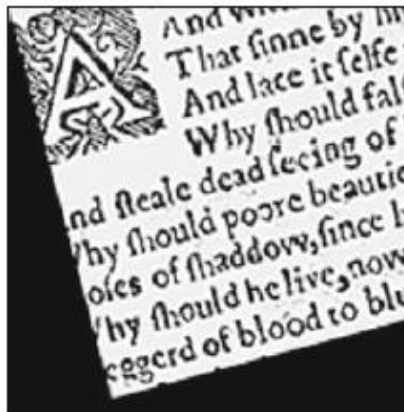
(b) Nearest-neighbor



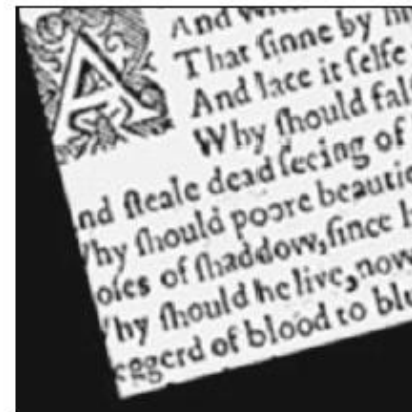
(c) Bilinear



(d) Bicubic



(e) Catmull-Rom



(f) Cubic B-spline

Burger, W., Burge, M. J. (2008). Digital Image Processing - An Algorithmic Introduction using Java. Springer. ISBN: 978-1-84628-968-2

Interpolación. Francisco J. Hernández-López

Ene-Jun 2025

GRACIAS POR SU ATENCIÓN

Francisco J. Hernandez-Lopez

fcoj23@ciimat.mx

WebPage:

www.ciimat.mx/~fcoj23

