

Métodos Numéricos. Proyecto Integrador de  
conocimientos. Interpolación de píxeles

Ana Rodríguez Jaraba  
Blanca López-Jorrín Pérez  
Jorge Ibinarriaga Robles  
Miguel Ángel Huamani  
Victoria Guillén de la Torre

April 2024

# Contents

1.	Introducción . . . . .	2
2.	Principales métodos de interpolación 2D . . . . .	2
2.1	Interpolación Bilineal . . . . .	2
2.2	Interpolación Bicúbica . . . . .	3
2.3	Interpolación del vecino más cercano . . . . .	4
3.	Diagramas de Voronoi . . . . .	4
4.	Conceptos básicos sobre el procesamiento de imágenes . . . . .	6
4.1	Tipos de imágenes digitales: . . . . .	6
4.2	RGB: . . . . .	6
4.3	Resolución de imágenes . . . . .	7
5.	Relevancia de los métodos de interpolación multivariante en el tratamiento de imágenes . . . . .	7
6.	Implementación de los algoritmos . . . . .	9
6.1	Interpolación bilineal . . . . .	9
6.2	Interpolación bicúbica . . . . .	10
6.3	Interpolación del Vecino más Cercano . . . . .	12
6.4	Diagramas de Vonroi . . . . .	12
7.	Comparación de los métodos expuestos . . . . .	12
7.1	Comparación con imágenes . . . . .	13
7.2	Comparación con funciones . . . . .	14
8.	Más métodos de interpolación multivariantes . . . . .	17
9.	Bibliografía . . . . .	18

## 1. Introducción

En el ámbito del procesamiento de imágenes, la interpolación en 2D desempeña un papel crucial para mejorar la calidad y la resolución de las imágenes digitales. Este proyecto se centra en la exploración y aplicación de diversos métodos de interpolación 2D, destacando su relevancia en escenarios reales.

La tarea principal consiste en explicar y comparar los principales métodos de interpolación 2D, tales como la Interpolación Bilineal, la Interpolación Bicúbica y la Interpolación del Vecino más Cercano. Además, se aborda la comprensión de los Diagramas de Voronoi, una herramienta esencial en Geometría Computacional, y su relación con la interpolación del vecino más cercano. Además, se presenta una breve introducción a los conceptos fundamentales del procesamiento de imágenes.

En resumen, este proyecto tiene como objetivo principal aplicar los conceptos y técnicas aprendidas en el curso de Métodos Numéricos para comprender su relevancia y utilidad en el procesamiento práctico de imágenes digitales.

Dadas  $n$  ternas  $(x_i, y_i, z_i)$  se busca encontrar una función  $f: D \rightarrow \mathbb{R}$  tal que  $f(x_i, y_i) = z_i$ . En muchas aplicaciones, para la interpolación en dos dimensiones se usa un regular grid para simplificar el proceso de muchos métodos de interpolación. Es una disposición estructurada de puntos donde los puntos están espaciados de forma preestablecida, no necesariamente uniforme. Tienen aplicaciones en estadística, meteorología y tratamiento de imágenes.

## 2. Principales métodos de interpolación 2D

### 2.1 Interpolación Bilineal

La interpolación bilineal es una extensión de la interpolación lineal la cual se emplea para interpolar funciones de dos variables. El objetivo principal en el que se basa esta técnica es encontrar los 4 puntos más cercanos conocidos, en nuestro caso hablamos de los 4 píxeles más cercanos, a un determinado punto  $P$  de coordenadas  $x$  e  $y$ . La imagen de la izquierda muestra gráficamente lo que se acaba de comentar.

El valor inferido de  $P$  se obtiene de la siguiente manera:

Primero hacemos una interpolación en la dirección x,

$$f(R_1) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21})$$

donde  $R_1 = (x, y_1)$ ,

$$f(R_2) \approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22})$$

donde  $R_2 = (x, y_2)$ .

Posteriormente una interpolación en la dirección y,

$$f(P) \approx \frac{y_2 - y}{y_2 - y_1} f(R_1) + \frac{y - y_1}{y_2 - y_1} f(R_2).$$

Lo cual nos da la siguiente estimación de  $f(x, y)$ ,

$$\begin{aligned} f(x, y) &\approx \frac{f(Q_{11})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y_2 - y) + \\ &\frac{f(Q_{21})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y_2 - y) + \\ &\frac{f(Q_{12})}{(x_2 - x_1)(y_2 - y_1)} (x_2 - x)(y - y_1) + \\ &\frac{f(Q_{22})}{(x_2 - x_1)(y_2 - y_1)} (x - x_1)(y - y_1) \\ &= \frac{1}{(x_2 - x_1)(y_2 - y_1)} [f(Q_{11})(x_2 - x)(y_2 - y) + \\ &f(Q_{21})(x - x_1)(y_2 - y) + f(Q_{12})(x_2 - x)(y - y_1) + \\ &f(Q_{22})(x - x_1)(y - y_1)]. \end{aligned}$$

Finalmente se puede obtener el valor de  $f(P)$ .

## 2.2 Interpolación Bicúbica

La interpolación bicúbica es una extensión de la interpolación cúbica empleada en el procesamiento de imágenes y gráficos computacionales para estimar valores desconocidos en una cuadrícula bidimensional a partir de los valores conocidos de los puntos adyacentes.

Esta interpolación puede ser obtenida mediante el empleo de polinomios de Lagrange, splines cúbicos o algoritmos de convolución cúbica. A diferencia de la

interpolación bilineal, que solo tiene en cuenta 4 píxeles ( $2 \times 2$ ), la interpolación bicúbica considera 16 píxeles ( $4 \times 4$ ).

### 2.3 Interpolación del vecino más cercano

La interpolación por el vecino más cercano es el método más sencillo de interpolación en 2 dimensiones.

Dado un punto  $(x,y)$  arbitrario a interpolar y dadas  $n$  tuplas  $(x_i, y_i, z_i)$ , el valor de la función interpolada en el punto  $(x,y)$  es el valor del punto más cercano,  $z_i$ . En el contexto del tratamiento de imágenes, si se tiene una matriz de píxeles  $n \times m$  y se quiere extender a una matriz  $n' \times m'$  con mayor número de píxeles. El valor de cada nuevo píxel se obtendría con el valor del vecino más cercano a la matriz de píxeles original como se puede observar en la Figura 1

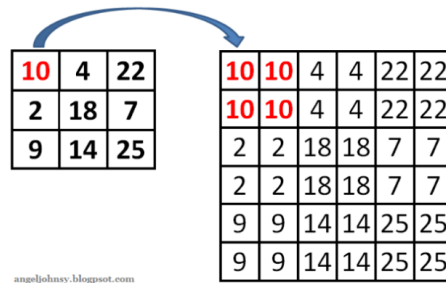


Figure 1: Ejemplo del funcionamiento de la interpolación del vecino más cercano

## 3. Diagramas de Voronoi

Los diagramas de Voronoi son una herramienta de las matemáticas en el campo de la geometría computacional muy utilizada para la interpolación en 2 dimensiones usando la interpolación por el vecino más cercano. En lugar de calcular el promedio de los píxeles circundantes, se toma el valor del píxel más cercano.

Idea: Dados  $n$  puntos en el plano, llamados semillas, se busca hacer una partición del espacio para cada semilla donde la partición sea el conjunto de puntos del espacio que tengan a dicha semilla como el punto más cercano. A dicha partición se le denomina celda de Voronoi.

La aplicación a la Interpolación por el vecino más cercano se debe a que dado un punto arbitrario del espacio, para encontrar el vecino más cercano de

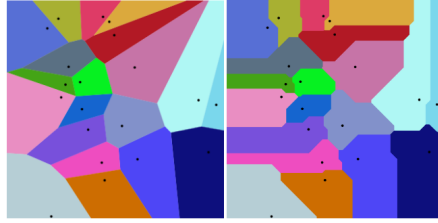


Figure 2: Diagrama de Vonroi con 20 puntos. A la derecha se ha empleado la distancia de Manhattan y a la derecha la distancia euclídea.

un conjunto de puntos dados (los puntos a interpolar) basta con ver a qué celda de Voronoi pertenece.

**Notal:** la zona de la frontera entre dos regiones se llama mediatriz y es lugar geométrica de los puntos que equidistan de ambas semillas (generadas por las dos regiones).

### Ejemplos de Diagramas de Voronoi



Figure 3: .

Diagrama de Voronoi contruido a partir de las capitales de las capitales de las provincias españolas. Para encontrar la ciudad más cercana en cada punto del país

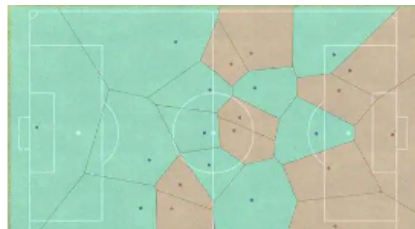


Figure 4: .

Diagrama de Voronoi en el fútbol: se construyen dos diagramas de Voronoi para cada equipo con cada jugador para ver la influencia de cada equipo sobre el terreno de juego. Podemos como el equipo azul tiene claramente mayor dominio en el campo y al tener las celdas conectadas, se favorecen los pases entre el equipo azul, no como en el equipo rojo.

## 4. Conceptos básicos sobre el procesamiento de imágenes

Una imagen digital es la codificación de una imagen en una matriz, mapa de bits o un vector para posteriormente poderla modificar, distribuir o visualizar.

### 4.1 Tipos de imágenes digitales:

- **Raster (bitmap):** representación bidimensional de una imagen a través de una matriz, donde cada elemento de la matriz es un píxel. La matriz tiene un número de columnas ( $m$ ) y un número de filas ( $n$ ) fijo. Es el formato más común. Formatos Ráster: JPG, JPEG, PNG.
- **Imágenes vectores:** Las imágenes vectoriales se basan en la representación de finitas coordenadas geométricas  $(x,y)$  en un sistema de coordenadas cartesianas  $p = (x,y)$ . Como una imagen contiene figuras que para representarlas se necesita un número infinito de puntos, se toman un conjunto finito de coordenadas y se trata de interpolar dichos puntos a través de primitivas geométricas. Estas primitivas pueden ir desde implementaciones sencillas como splines lineales o cuadráticos, hasta más complejas como curvas paramétricas o fractales. También hay distintos tipos como GIF (imágenes en movimiento), imágenes 3D o íconos.

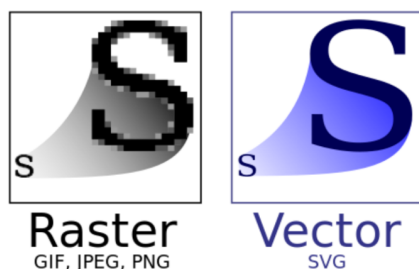


Figure 5: Comparación de los dos tipos de imágenes digitales expuestos.

### 4.2 RGB:

Cada píxel se representa con tres canales (Red, Green, Blue) donde cada canal representa la tonalidad para cada color. Cada canal, al igual que con la escala de grises, se representa con un 8 bits (255 tonalidades). La combinación de estos tres canales da lugar a todos los colores distinguidos para el ojo humano, con un total de 6.777 millones combinaciones posibles. Por esto en el contexto de álgebra lineal, RGB se entiende como una base tridimensional del espacio vectorial de colores,  $R \times G \times B$ .

En la imagen 6 se puede ver la imagen original en sus 3 canales distintos.

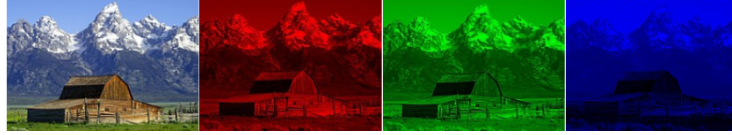


Figure 6: Imagen original en sus tres canales

### 4.3 Resolución de imágenes

El nivel de resolución es el nivel de detalle en la imagen. Un aumento en la resolución supone un aumento en la calidad de la imagen, pero a coste de mayor espacio de almacenamiento.

- En las imágenes ráster la resolución se mide por el número de píxeles.
- En las imágenes vectoriales el contexto de resolución no es aplicable por la naturaleza de su construcción, ya que estas imágenes no dependen de píxeles y pueden escalarse a cualquier dimensión sin perder calidad.

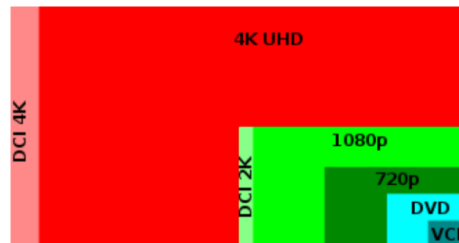


Figure 7: cuanto mayor es el área, mayor es la resolución y, por lo tanto, mejor es la calidad de imagen. Se muestra cómo las diferentes resoluciones se comparan en términos de tamaño y claridad, con el 4K siendo mucho más detallado y con más píxeles que las resoluciones más bajas.

## 5. Relevancia de los métodos de interpolación multivariante en el tratamiento de imágenes

Los métodos de interpolación multivariante tienen una gran relevancia en el contexto del tratamiento de imágenes debido a sus numerosas aplicaciones. Estos métodos no sólo se aplican al escalado de imágenes, donde al aumentar el



número de píxeles, se interpola el valor de los píxeles desconocido.

También, estos métodos son importantes para otras técnicas del tratamiento de imágenes como la rotación de imágenes, ya que cuando una imagen en un marco rectangular es rotada, los píxeles no se alinean perfectamente con la imagen rotada y la interpolación es necesaria para calcular los valores de los píxeles en la nueva rotación.

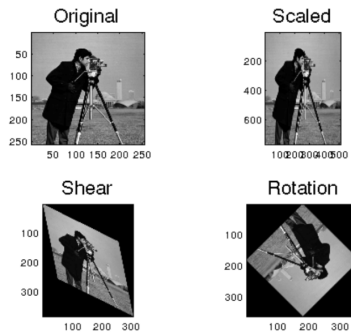


Figure 8: Ejemplo de interpolación aplicado al tratamiento de imágenes: escalado, rotación y cambio de perspectiva

También se puede aplicar a la superresolución, que trata de aumentar la resolución de una imagen. Para aumentar la resolución de una imagen, como hemos visto anteriormente, es necesario aumentar el número de píxeles y para ello es necesario interpolar los nuevos píxeles añadidos. Cabe destacar de que este método difiere del aumento de imágenes, ya que en la superresolución no se está aumentando el tamaño de la imagen, simplemente se está dividiendo el margen original en un mayor número de píxeles.



Figure 9: Comparación de una misma imagen con dos resoluciones distintas

## 6. Implementación de los algoritmos

### 6.1 Interpolación bilineal

En este apartado, se abordará la implementación del algoritmo de interpolación bilineal en el contexto específico del procesamiento de imágenes, detallando cómo se ha llevado a cabo la programación y optimización de este método para mejorar la calidad visual de las imágenes resultantes.

Primero se ha definido una función la cual, dadas las coordenadas de un punto a interferir y los 4 puntos conocidos más cercanos junto con sus valores, se obtiene siguiendo la ecuación mostrada en el apartado 2.2, el valor inferido de dicho punto.

Para la obtención de los 4 puntos más cercanos, se ha realizado otra función. En dicha función se obtienen las coordenadas de estos 4 puntos en la matriz ya ampliada, es decir, si tengo una matriz 3x3 y la quiero ampliar xZ donde z=2 en este caso, los puntos se obtendría en la matriz 6x6 resultante. Cabe destacar que estos puntos deben ser conocidos, por lo que se buscan de Z en Z. Por último, una vez obtenidas las coordenadas en la matriz ampliada, se establece la siguiente política para no solo obtener dichas coordenadas en la matriz original sino para también obtener su valor:

$$x_{\text{original}} = \frac{x_{\text{ampliada}} + z - 1}{z}$$

$$y_{\text{original}} = \frac{y_{\text{ampliada}} + z - 1}{z}$$

Donde x-original e y-original se corresponden con las coordenadas de la matriz dada (píxeles de la imagen) y x-ampliada e y-ampliada serían las coordenadas de la matriz ampliada. z se corresponde con la ampliación de la imagen.

Todo lo dicho se puede observar de forma gráfica en Figura ??

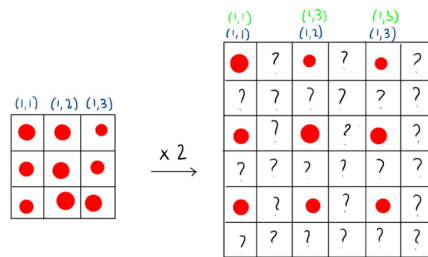


Figure 10: Representación gráfica de la distribución de los píxeles originales en la nueva imagen ampliada donde z=2

Finalmente se diseña la función principal donde se combinan las funciones mencionadas anteriormente para la obtención de la imagen ampliada, cabe destacar que esta función es empleada para los 3 métodos, se utiliza uno u otro en función de la variable ‘tipo’ introducida.

## 6.2 Interpolación bicúbica

La interpolación bicúbica es una extensión de la interpolación bilineal con la diferencia de que en lugar de emplear únicamente los 4 píxeles más cercanos al punto objetivo, considera 16.

Este algoritmo consta de 5 pasos principales: definir una función interpoladora, en este caso, spline cúbico, establecer una región de interpolación, es decir, para interpolar un punto específico es necesario definir una región alrededor de él, que incluya los valores de los píxeles conocidos.

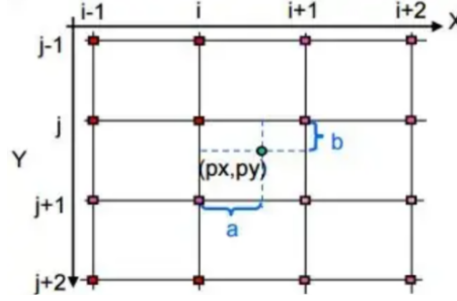


Figure 11: Región de interpolación definida

Una vez que se tiene la región de interpolación 17, se calculan los coeficientes de la función interpoladora bicúbica, y finalmente se evalúa la función interpoladora en el punto deseado. Este proceso se repite para cada punto que se desea interpolar.

Ahora formalicemos esto empleando las fórmulas matemáticas pertinentes y finalmente explicando el algoritmo diseñado en matlab. Como hemos dicho, para cualquier punto (x,y), cuyo valor se quiera hallar mediante interpolación bicúbica se necesitarán los 16 coeficientes  $a_{ij}$ , pues el valor interpolado no será más que la siguiente igualdad:

$$p(x, y) = \begin{bmatrix} 1 & x & x^2 & x^3 \end{bmatrix} \begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} \begin{bmatrix} 1 \\ y \\ y^2 \\ y^3 \end{bmatrix} \quad (1)$$

Por tanto, el problema se reduce a hallar dichos coeficientes. Para ello, definiendo el cuadrado unitario  $[0,1] \times [0,1]$ , debemos tener en cuenta que:

$$p(x, y) = \sum_{i=0}^3 \sum_{j=0}^3 a_{ij} x^i y^j \quad (2)$$

Además, si imponemos condiciones naturales de los splines cúbicos, pues no queremos cortes marcados en la imagen resultante, nos queda:

$$\begin{bmatrix} a_{00} & a_{01} & a_{02} & a_{03} \\ a_{10} & a_{11} & a_{12} & a_{13} \\ a_{20} & a_{21} & a_{22} & a_{23} \\ a_{30} & a_{31} & a_{32} & a_{33} \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -3 & 3 & -2 & -1 \\ 2 & -2 & 1 & 1 \end{bmatrix} \begin{bmatrix} f(0,0) & f(0,1) & f_y(0,0) & f_y(0,1) \\ f(1,0) & f(1,1) & f_y(1,0) & f_y(1,1) \\ f_x(0,0) & f_x(0,1) & f_{xy}(0,0) & f_{xy}(0,1) \\ f_x(1,0) & f_x(1,1) & f_{xy}(1,0) & f_{xy}(1,1) \end{bmatrix} \begin{bmatrix} 1 & 0 & -3 & 2 \\ 0 & 0 & 3 & -2 \\ 0 & 1 & -2 & 1 \\ 0 & 0 & -1 & 1 \end{bmatrix}$$

Por ello, tendremos que hallar para cada punto, las derivadas respecto de  $x$ ,  $y$  y también  $xy$ . No obstante, y de manera evidente, no tenemos la función por lo que no podemos calcular los valores exactos de las derivadas, sino que nos veremos obligados a trabajar con valores aproximados.

Luego, para la implementación del algoritmo, lo primero que hemos hecho ha sido calcular para todos los puntos conocidos, las aproximaciones de sus derivadas, y almacenarlos en matrices de iguales dimensiones a la matriz imagen. De esta forma, se facilitará el acceso a dichos valores y no será necesario calcularlos varias veces. Para la aproximación, hemos empleado la fórmula de interpolación de tres puntos, por lo que la derivada en el punto central vendrá dada por:

$$f'(x_0) = \frac{f(x_1) - f(x_{-1})}{2h} \quad (3)$$

Ya tendríamos todo lo necesario para calcular los valores de los nuevos píxeles. No obstante, cabe destacar el último problema a resolver, la equivalencia entre los índices antiguos y los de la nueva matriz. Para resolverlo, es necesario construir una matriz de transformación inversa, es decir, una matriz que emplearemos para cambiar el índice del punto que queremos hallar en nuestra matriz ampliada al punto en la matriz original, de manera proporcional en cada eje.

Por último, solo quedaría recorrer la matriz entera, encontrando para cada punto los 4 puntos que lo rodean y resolver la ecuación de la figura X.

### 6.3 Interpolación del Vecino más Cercano

Este enfoque considera, para cada nuevo punto que debe ser insertado en la imagen, el valor del píxel más cercano en la imagen original. Dicho en términos prácticos, se asigna al nuevo píxel el valor del píxel vecino más próximo de la imagen que se está escalando.

Para este algoritmo se creó una función que obtiene el píxel más cercano a un punto dado. Una vez más como el punto está en coordenadas de la matriz ampliada, es preciso establecer una política que determine cuáles con las coordenadas en la imagen original del píxel más cercano al dado. La política es la siguiente:

$$x_{\text{original}} = \frac{x_{\text{ampliada}} + z - 1}{z}$$
$$y_{\text{original}} = \frac{y_{\text{ampliada}} + z - 1}{z}$$

Donde x-original e y-original se corresponden con las coordenadas de la matriz dada (píxeles de la imagen) y x-ampliada e y-ampliada serían las coordenadas de la matriz ampliada. z se corresponde con la ampliación de la imagen. Y floor indica un redondeo hacia abajo.

### 6.4 Diagramas de Vonroi

**Construcción de Diagramas de Voronoi. Algoritmo de Fortune** Algoritmo de línea de barrido para generar un diagrama de Voronoi, es uno de los más eficientes para construir diagramas de Voronoi. Complejidad  $O(n \log n)$ , con n el número de semillas.

El algoritmo consiste en un barrido en línea que barre el plano de izquierda a derecha que va calculando el diagrama de Vonoi sólo con la parte del plano que ya ha sido barrida, ignorando los puntos de la derecha.

Para cada punto de la izquierda, se puede definir el conjunto de puntos que equidistan de la línea de barrido (eje) y del propio punto, resultando en una parábola, denominado como línea de varado. Los vértices de la línea de varado en el que se cruzan las parábolas, trazan las líneas del diagrama de Voronoi. Se puede observar la primera iteración de este algoritmo en la Figura 100

## 7. Comparación de los métodos expuestos

Para poder tener una idea más clara de la funcionalidad de los 3 métodos de interpolación implementados, se ha decidido comparar tanto los resultados usando la interpolación para funciones predefinidas así como para imágenes (no se sabe la función pero sí el  $f(x, y)$ ).

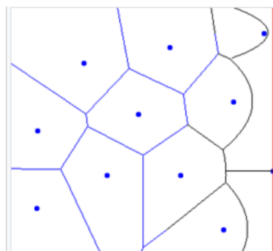


Figure 12: Primera iteración del algoritmo de Fortune

## 7.1 Comparación con imágenes

La imagen que se ha decidido comparar es la mostrada en la Figura 13



Figure 13: Imagen objeto de comparación

Se puede observar que el que método que mejor funciona es la interpolación bicúbica. Este método emplea un función cúbica que se ajusta a los píxeles más cercanos, lo que significa que obtiene más información sobre estos. Esta interpolación da lugar a una mayor suavidad en las transiciones y bordes, así como una mejor preservación de los detalles finos de la imagen original.

La interpolación bilineal también tiene en cuenta una gran cantidad de información de los píxeles vecinos, sin embargo, su resultados son peores que la bicúbica debido a que la complejidad de su algoritmo es menor.

Finalmente los peores resultados se obtienen con la interpolación del vecino más cercano, pues este simplemente toma el valor del píxel más próximo al punto de interpolación. Se observa como las transiciones de un píxel al otro es muy notoria y poco suave. Cabe destacar que, pese a los resultados ligeramente peores de este algoritmo, es el más rápido y fácil de implementar.



## 7.2 Comparación con funciones

Todos los métodos expuestos anteriormente nos permiten rellenar la información incompleta con la que contamos, en este caso, de una imagen. Es decir, estos métodos nos permiten añadir píxeles con los que no contamos para poder expandir una imagen, consiguiendo una mayor resolución de esta.

En el ejercicio se nos propone comparar las distintas técnicas de interpolación con las que hemos trabajado a lo largo de esta práctica final para ser capaces de sacar conclusiones sobre el funcionamiento de cada una de ellas cuando las aplicamos a una función de la cual conocemos su valor en cada punto.

En la Figura 14 podemos observar de forma visual las imágenes de las funciones  $f(x,y)$  y  $g(x,y)$ , las cuales modificaremos aplicando los métodos ya mencionados para, así, poder comprobar cómo la información que rellenan estos métodos, es decir, los píxeles que añaden a las imágenes, son correctos. Esto podemos comprobarlo ya que, en este caso, trabajamos con funciones de las que conocemos todos sus datos y nuestro objetivo es llevar a cabo el proceso contrario que hemos seguido a lo largo de la práctica: interpretar la imagen obtenida con las funciones como la imagen ya interpolada y obtener la imagen original aplicando las técnicas comentadas, reduciendo la resolución, es decir, reduciendo la cantidad de píxeles en dicha imagen.

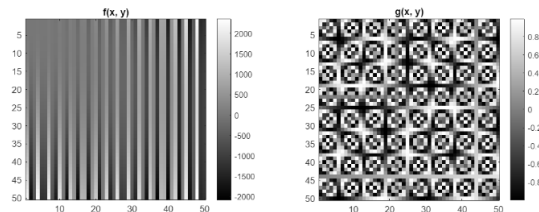


Figure 14

### Interpolación del vecino más cercano

Tras mostrar la imagen (Figura 15) de ambas funciones después de haberlas interpolado por el vecino más cercano, somos capaces de obtener algunas conclusiones, partiendo de la base de que la interpolación afecta a la resolución de las imágenes.

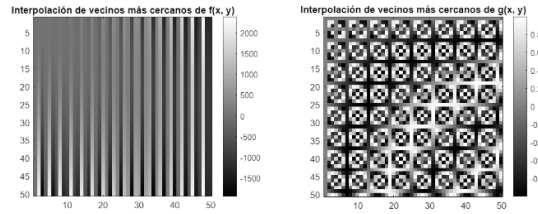


Figure 15

En el caso de la función  $f(x, y)$ , observamos que, en las zonas de alto contraste no son tan prevalentes, es decir, los cambios de blanco a negro no predominan tanto como el resto. Este método de interpolación, aplicado a la función  $f$ , puede no devolver un resultado muy diferente del original debido a que, al asignar el valor del píxel más cercano a los nuevos píxeles, puede no añadir diferencia apreciables a simple vista. Al asignar el valor de los píxeles más cercanos, esta interpolación puede resultar en la agrupación de píxeles que parecen continuos.

Por otra parte, si nos fijamos en el resultado obtenido al interpolar la función  $g(x, y)$  con el método de los vecinos, observamos que la diferencia entre la imagen original y la obtenida es mayor. Esto se debe a que la imagen de esta función presenta una mayor complejidad y mayor cantidad de detalles, con importantes cambios a lo largo de la dimensión de la matriz. Al interpolar usando el método del vecino más cercano y reducir su resolución, los detalles más precisos se pierden y, por eso, en el resultado final, no se aprecia la complejidad de la imagen original.

### Interpolación Bilineal

Esta técnica de interpolación, al basarse en un promedio ponderado de los 4 píxeles más cercanos, tiende a producir resultados más suaves que el método anterior, la interpolación de vecinos más cercanos. (Figura

Primero, encontramos la imagen de la función  $f$  una vez interpolada, en la que podemos observar que no hay una gran diferencia si la comparamos con la imagen original de la función. Esta diferencia inapreciable puede deberse a factores como el ancho de las franjas de la imagen o a la uniformidad de estas.

En la imagen que encontramos a la derecha, apreciamos la imagen de la función  $g(x, y)$  interpolada bilinealmente. Al contrario que el caso anterior, ahora



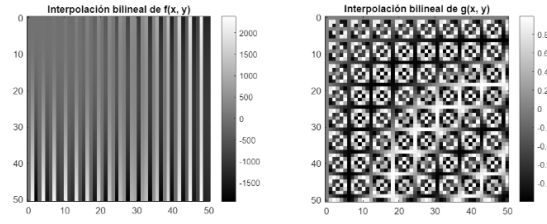


Figure 16

sí que se aprecia una clara diferencia entre la imagen original y la obtenida tras interpolar. Al tratarse de una función cuya imagen presenta una gran complejidad, la interpolación de esta resulta en una menor definición de aquellos detalles que inicialmente eran más precisos. Es por esto mismo que se aprecia mejor esta disminución de la resolución en la imagen de la función  $g$ , ya que los detalles perdidos no se aprecian tanto en la función  $f$ .

### Interpolación Bicúbica

Por último, tras aplicar interpolación bicúbica a ambas funciones,  $f$  y  $g$ , obtenemos las imágenes mostradas en las gráficas. Este método de interpolación suele producir resultados más suaves que la interpolación bilineal, como es el caso anterior. Esta forma de interpolar, más sutil que la interpolación bilineal, resulta en una transición más delicada y menos forzada (Figura 17).

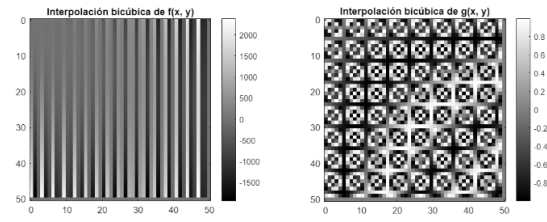


Figure 17

Como se puede observar, la imagen que muestra de la función  $f$ , parece que los bordes se mantienen sin diferencias apreciables a simple vista, mostrando las franjas en estas áreas igual de claras, prácticamente, que en la imagen original. Al tratarse de una imagen en la que encontramos franjas verticales uniformes, este método puede reproducir sin gran dificultad los valores de los píxeles, teniendo en cuenta los valores de los píxeles más próximos. Es por ello por lo que en la imagen obtenida observamos los bordes claramente definidos y transiciones suaves entre las franjas más céntricas.

En el caso de la función  $g$ , al tratarse de una imagen con un patrón más detallado y complejo, se aprecian más diferencias con la imagen original, ya que, en general, esos patrones se mantienen en gran medida. Si bien es cierto, que alguno de esos detalles más desarrollados se han atenuado más de lo que lo hicieron cuando interpolamos la función bilinealmente. Este método de interpolación permite una transición más suave, lo que puede resultar a su vez en la pérdida de algunos detalles, sobre todo de los detalles pequeños que se aprecian en imágenes con una mayor resolución.

## 8. Más métodos de interpolación multivariantes

- **Interpolación por Splines:** los splines que hemos visto en la asignatura de métodos numéricos se pueden extender a varias variables, partiendo de la misma idea. A partir de una partición de un intervalo, se trata de buscar un polinomio (spline) que interpole los puntos en cada subintervalo. En varias variables, la idea es la misma, solo que ahora se obtiene el spline de cada variable por separado y se agregan para interpolar la función conjunto mediante el producto tensorial (TENSOR PRODUCT SPLINES)
- **Interpolación de Barnes:** la interpolación de Barnes se trata de una media ponderada donde las ponderaciones se obtienen a partir de un kernel gaussiano de la distribución de la distancia del punto a interpolar al resto de puntos.

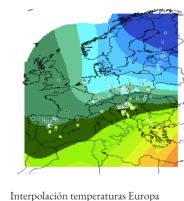


Figure 18: Ejmplos de implementación de la interpolación de Barnes a la vida real prediciendo la temperatura de Europa

- **Delaunay triangulationes** una herramienta de la geometría computacional: divide el espacio de un conjunto de puntos en triángulos cuyos circuncentros no contengan ningún punto, el cual normalmente corresponde con el diagrama de Voronoi. Esto, es útil para computar otros métodos de interpolación como la interpolación por el vecino más cercano. También, se pueden interpolar las puntos, mediante interpolación lineal a través de los vértices del triángulo que contiene al punto.

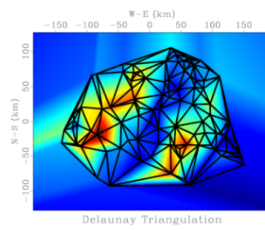


Figure 19: Ejemplo de función interpolada mediante triangulación de Delaunay

## 9. Bibliografía

Wikipedia - Interpolación Bilineal  
 Repositorio UJI - TFG Alba Roncero García