



# Máster Universitario de Investigación en Ingeniería de Software y Sistemas Informáticos

Itinerario de Sistemas Informáticos (Cód. 31105132)

---

ESCALADO DE IMAGEN MEDIANTE LA INTERPOLACIÓN POR MÍNIMOS CUADRADOS Y  
COMPARACIÓN CON LOS MÉTODOS CLÁSICOS.

---

*Autor: Javier Colomé Abril*  
*Dirección: Carlos Cerrada Somolinos*



CURSO 2014 – 2015  
CONVOCATORIA SEPTIEMBRE 2015



**UNED. Universidad Nacional de Educación a Distancia.**

Escuela Técnica Superior de Ingeniería Informática.

*Departamento de Ingeniería de Software y Sistemas Informáticos.*

**Máster Universitario en Investigación en Ingeniería de Software y Sistemas Informáticos.**

Itinerario de Sistemas Informáticos (Código de Asignatura 31105132)

*Trabajo Final de Máster (TFdM)*

***Escalado de imagen mediante la interpolación por mínimos cuadrados y  
comparación con los métodos clásicos.***

Autor: Javier Colomé Abril

Dirección: Carlos Cerrada Somolinos

Septiembre 2015

**Escalado de imagen mediante la interpolación por mínimos cuadrados y comparación con los métodos clásicos.**



**Escalado de imagen mediante la interpolación por mínimos cuadrados y comparación con los métodos clásicos.**



**Escalado de imagen mediante la interpolación por mínimos cuadrados y comparación con los métodos clásicos.**





IMPRESO TFdM05\_AUTOR  
AUTORIZACIÓN DE PUBLICACIÓN  
CON FINES ACADÉMICOS



**Impreso TFdM05\_Autor. Autorización de publicación  
y difusión del TFdM para fines académicos**

## Autorización

Autorizo/amos a la Universidad Nacional de Educación a Distancia a difundir y utilizar, con fines académicos, no comerciales y mencionando expresamente a sus autores, tanto la memoria de este Trabajo Fin de Máster, como el código, la documentación y/o el prototipo desarrollado.

Firma del/los Autor/es

Juan del Rosal, 16  
28040, Madrid

Tel: 91 398 89 10  
Fax: 91 398 89 09

[www.issi.uned.es](http://www.issi.uned.es)

*Escalado de imagen mediante la interpolación por mínimos cuadrados y comparación con los métodos clásicos.*



## Abstracto

El presente trabajo persigue el estudio e implementación práctica de los métodos interpoladores clásicos de imágenes digitales por vecino más próximo, bilineal y bicúbica, así como la prospección, estudio e implementación de un nuevo sistema interpolador propio, mediante la estimación de valores de intensidad de píxel, haciendo uso de líneas de tendencia calculadas por mínimos cuadrados, con el que poder comparar científicamente los resultados entre los métodos de interpolación definidos en el trabajo.

Se establece adicionalmente el estudio e implementación de un sistema para llevar a cabo esta comparación de imágenes que determine de forma cuantitativamente y objetivamente la calidad de las mismas. En el trabajo se define el uso del método PSNR (Peak signal-to-noise ratio) para llevar a cabo esta operación de comparación. Alternativamente se ha explorado el uso de la extracción de bordes mediante máscaras de Sóbél para identificar la posibilidad de relacionar la calidad de una imagen con este sistema y realizar una comparación con el método PSNR.

## Palabras clave

<i>Interpolación</i>	<i>Interpolation</i>
<i>Vecino Más Próximo</i>	<i>Nearest Neighbor</i>
<i>Bicúbica</i>	<i>Bicubic</i>
<i>Bilineal</i>	<i>Bilinear</i>
<i>Error Cuadrático Medio</i>	<i>Mean squared error</i>
<i>Imagen Digital</i>	<i>Digital Image</i>
<i>Procesamiento digital de imagen</i>	<i>Digital Image Processing</i>
<i>Escalado</i>	<i>Scale</i>
<i>Mínimos Cuadrados</i>	<i>Least Squares</i>
<i>Relación Señal a Ruido de Pico</i>	<i>Peak signal-to-noise ratio</i>
<i>Línea de tendencia</i>	<i>Trendline</i>
<i>PSNR</i>	<i>Sobel Operator</i>
<i>ImageJ</i>	

*Escalado de imagen mediante la interpolación por mínimos cuadrados y comparación con los métodos clásicos.*



## Índice de Contenidos

<b>1. Lista de figuras y de tablas .....</b>	<b>11</b>
<b>2. Objetivos del TFdM .....</b>	<b>15</b>
<b>3. Introducción.....</b>	<b>16</b>
<b>4. Conceptos preliminares .....</b>	<b>17</b>
4.1. Tasa de Muestreo de Nyquist .....	17
4.2. Aliasing.....	19
4.3. Interpolación.....	20
4.4. Distancia Euclídea .....	21
4.5. Interpolación Lineal .....	22
4.6. Programación del sistema interpolación en Java como un componente para ImageJ .....	24
<b>5. Interpolaciones Clásicas.....</b>	<b>25</b>
5.1. Introducción.....	25
5.2. Interpolación por vecino más próximo.....	27
5.2.1. Características principales .....	27
5.2.2. Cálculos de la interpolación .....	27
5.3. Interpolación Bilineal .....	31
5.3.1. Características principales .....	31
5.3.2. Cálculos de la interpolación .....	31
5.4. Interpolación Bicúbica .....	34
5.4.1. Características principales .....	34
<b>6. Interpolación mediante la aproximación por mínimos cuadrados .....</b>	<b>35</b>
6.1. Línea de tendencia mediante el cálculo de Mínimos Cuadrados .....	35
6.1.1. Conceptos preliminares .....	35
6.1.2 Cálculo de la línea de tendencia mediante Mínimos Cuadrados.....	37
6.2. Aplicación práctica del escalado mediante la Interpolación .....	40
6.2.1. Preparación de la imagen original .....	40
6.2.2. Espaciado de los píxeles conocidos en un mapa x2.....	40
6.2.3. Estimación mediante mínimos cuadrados de los píxeles ubicados en las intersecciones (p) .....	41
<i>Sistema 1 – Cálculo mediante la línea de tendencia de dos diagonales. ....</i>	<i>42</i>

<i>Sistema 2 - Cálculo mediante la línea de tendencia de dos diagonales y promedio de vecinos.</i> .....	45
6.2.4. Estimación de los píxeles cardinales (n,s,e,o) con respecto a los píxeles de intersección (p).....	46
<i>Sistema A - Cálculo de los píxeles cardinales mediante el promedio de los dos vecinos más próximos.</i> .....	47
<i>Sistema B - Cálculo de los píxeles cardinales mediante las líneas de tendencia en sus coordenadas</i> .....	48
<i>Sistema C - Cálculo de los píxeles cardinales mediante un sistema de pesos</i> .....	50
<i>Sistema D - Cálculo de los píxeles cardinales en doble iteración, mediante un sistema de pesos entre la vecindad de los píxeles discretos y los píxeles estimados mediante líneas de tendencia.</i> .....	53
<b>7. Interpolación en color</b> .....	<b>55</b>
<b>8. Plugin Interpolador para ImageJ</b> .....	<b>57</b>
8.1. Características.....	57
8.2. Compilación del plugin.....	57
8.3. Ejecución del plugin .....	59
<i>Interpolación “Vecino más Próximo”.</i> .....	59
<i>Interpolación “Bilineal”.</i> .....	59
<i>Interpolación “Bicúbica”.</i> .....	60
<i>Interpolación propia mediante “Mínimos Cuadrados”.</i> .....	60
<b>9. Método comparador de resultados</b> .....	<b>65</b>
9.1. Número de píxeles de borde mediante convolución por máscaras de SOBEL.....	65
9.2. Peak signal-to-noise ratio (PSNR) .....	66
<i>MAX<sub>i</sub> y el máximo nivel de intensidad</i> .....	67
<i>El Error Cuadrático Medio (MSE)</i> .....	68
<b>10. Plugin comparador cuantitativo de imágenes para ImageJ</b> .....	<b>69</b>
10.1. Características.....	69
10.2. Compilación del plugin.....	69
10.3. Ejecución del plugin .....	71
<b>11. Juego de pruebas y análisis</b> .....	<b>74</b>
<b>12. Conclusiones</b> .....	<b>80</b>
<b>13. Trabajo a Futuro</b> .....	<b>81</b>
<b>14. Dedicación y horas</b> .....	<b>81</b>
<b>15. Bibliografía</b> .....	<b>83</b>
<b>16. Lista de Siglas</b> .....	<b>83</b>

## 1. Lista de figuras y de tablas

<b>Figura o Tabla</b>	<b>Pág.</b>
Figura 1. Primera imagen digital escaneada. 1957.	16
Figura 2. Realizar un escalado al doble de tamaño implica el cálculo del valor de intensidad del 75% de los píxeles.	17
Figura 3. Ejemplos de reconstrucción de una señal senoidal mediante diferentes frecuencias de muestreo.	18
Figura 4. Imagen correctamente muestreada.	19
Figura 5. Imagen con efecto aliasing.	19
Figura 6. Corrección básica del efecto aliasing.	19
Figura 7. Señal con pérdida de información.	20
Figura 8. Señal reconstruida por interpolación.	20
Figura 9. Representación $d_E$ entre el punto 'P' y los puntos 'i' y 'j'.	21
Figura 10. Representación de ocho píxeles horizontales.	22
Figura 11. Definición gráfica para el cálculo del píxel 'X' según el gradiente entre los píxeles 'A' y 'B'	22
Figura 12. Representación del nivel de mezcla de color para el píxel 'X'.	23
Figura 13. Representación del nivel de mezcla para todos los píxeles.	23
Tabla 1. Representación RGB del ejemplo.	23
Tabla 2. Representación RGB completa del ejemplo.	23
Figura 14. Logotipo de la aplicación ImageJ.	24
Figura 15. Coordenadas para seis píxeles	25
Figura 16. Imagen de 2x3 píxeles.	25
Figura 17. Imagen de 4x6 píxeles objetivo del escalado x2.	25
Figura 18. Imagen 2x3 original superpuesta en la imagen 4x6.	26
Figura 19. Coordenadas asociadas a la superposición anterior.	26
Figura 20. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.	27
Figura 21. Imagen 2x3 original superpuesta en la imagen 4x6.	27
Tabla 3. Valores de los mapas superpuestos.	27
Tabla 4. Cálculo de la Distancia Euclídea y el núcleo de interpolación para los píxeles de ejemplo.	28
Tabla 5. Cálculo del núcleo de interpolación para los píxeles 'j', 'k', 'o' y 'p'.	28
Tabla 6. Cálculo del nivel de intensidad del píxel 'j'.	29
Tabla 7. Cálculo del nivel de intensidad del píxel 'k'.	29
Tabla 8. Cálculo del nivel de intensidad del píxel 'o'.	29
Tabla 9. Cálculo del nivel de intensidad del píxel 'p'.	29
Tabla 10. Cálculo del nivel de intensidad en los mapas.	30
Figura 22. Representación del nivel los píxeles 'j', 'k', 'o' y 'p', una vez calculados.	30
Figura 23. Representación de la interpolación una vez finalizada.	30
Figura 24. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.	31
Tabla 11. Cálculo de la Distancia Euclídea y el núcleo de interpolación para los píxeles de ejemplo.	32

<b>Figura o Tabla</b>	<b>Pág.</b>
Tabla 12. Cálculo del nivel de intensidad del píxel 'j'.	32
Tabla 13. Cálculo del nivel de intensidad del píxel 'k'.	32
Tabla 14. Cálculo del nivel de intensidad del píxel 'o'.	32
Tabla 15. Cálculo del nivel de intensidad del píxel 'p'.	33
Tabla 16. Cálculo del nivel de intensidad en los mapas.	33
Figura 25. Representación del cálculo de los 'j', 'k', 'o' y 'p'.	33
Figura 26. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.	34
Tabla 17. Datos discretos conocidos.	35
Figura 27. Línea de tendencia por mínimos cuadrados de puntos dispersos en un espacio.	35
Figura 28. Siete píxeles donde 'B', 'D' y 'F' son píxeles con nivel de gris desconocidos.	36
Figura 29. Representación isométrica de los siete píxeles, teniendo en cuenta que la dimensión de la altura corresponde con el nivel de intensidad del píxel.	36
Figura 30. Línea de tendencia para la recta de píxeles con niveles de gris que no siguen un gradiente.	37
Tabla 18. Datos discretos conocidos.	38
Figura 31. Representación en un gráfico de dispersión de los puntos conocidos.	38
Tabla 19. Cálculos representativos para 'b' y 'm'.	38
Figura 32. Línea de tendencia calculada por Mínimos Cuadrados para el conjunto de puntos discretos en el gráfico de dispersión.	39
Figura 33. Explosión de los píxeles originales de la imagen de 4x4 píxeles, en un nuevo mapa de 8x8 píxeles.	41
Figura 34. Píxeles 'p' en las intersecciones de píxeles desconocidos.	41
Figura 35. Píxel 'p' de muestra interpolado mediante el uso de dos diagonales de tendencia sobre ese punto.	41
Figura 36. Imagen de 4x4 píxeles.	42
Figura 37. Nueva imagen de 8x8 píxeles con los píxeles espaciados.	42
Figura 38. Píxel de intersección 'p' a calcular, mediante las dos diagonales que lo cruzan.	43
Figura 39. Representación isométrica de una de las diagonales con el píxel 'p' central a calcular.	44
Figura 40. Una de las dos diagonales en la que el píxel 'p' se encuentra en la cuarta posición.	44
Tabla 20. Cálculos representativos para 'b' y 'm'.	44
Figura 41. Ejemplo para nuestra matriz de cálculo.	46
Figura 42. Píxeles 'n', 's', 'e' y 'o' pertenecientes a las coordenadas de un píxel de intersección 'p'.	46
Figura 43. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.	47
Tabla 21. Representación de los nueve píxeles y cálculo de los píxeles cardinales 'n', 's', 'e' y 'o'.	47
Figura 44. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.	48
Figura 44 bis. Representación de las dos horizontales y las dos diagonales de píxeles.	48
Tabla 22. Cálculos representativos para 'b' y 'm'.	49
Tabla 23. Cálculos representativos para 'b' y 'm'.	49
Tabla 24. Cálculos representativos para 'b' y 'm'.	49
Figura 45. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.	50
Tabla 25. Tabla de representación de pesos según el ejemplo mostrado.	51
Figura 46. Píxel de intersección 'w'.	52
Figura 47. Píxel n' temporal.	52
Figura 48. Píxel de intersección 'p'.	52

<b>Figura o Tabla</b>	<b>Pág.</b>
Figura 49. Imagen original 4x4.	53
Figura 50. Imagen espaciada.	53
Figura 51. Imagen con píxeles de intersección 'p'.	53
Figura 52. Píxeles cardinales a calcular para un píxel de intersección 'p' dado.	54
Figura 53. Ejemplo ilustrativo de la interpolación a color propuesta.	56
Figura 54. ImageJ y procedimiento de compilación del plugin.	58
Figura 55. Mensaje informativo tras la compilación del plugin.	58
Figura 56. Ubicación del plugin una vez compilado.	58
Figura 57. Pantalla que muestra la opción de interpolación mediante vecino más próximo.	59
Figura 58. Pantalla que muestra la opción de interpolación bilineal.	59
Figura 59. Pantalla que muestra la opción de interpolación bicúbica.	60
Figura 60. Pantalla que muestra la opción de interpolación por mínimos cuadrados.	60
Figura 61. Pantalla de configuración de la interpolación por mínimos cuadrados.	61
Tabla 26. Tabla de configuraciones posibles del interpolador por mínimos cuadrados.	62
Figura 62. Pantalla de ayuda del plugin interpolador.	63
Figura 63. Imagen original 512x512 píxeles en color.	63
Figura 64. Detalle de la imagen original.	63
Figura 65. Imagen original 512x512 convertida a escala de grises.	64
Figura 66. Detalle de la imagen en escala de grises.	64
Figura 67. Nuevo mapa escalado al doble de dimensión (1024x1024) con los píxeles de la imagen original espaciados y lista para ser interpolada.	64
Figura 68. Detalle de la imagen espaciada.	64
Figura 69. Imagen escalada definitiva, una vez interpolada y con dimensión 1024x1024.	64
Figura 70. Detalle de la imagen interpolada.	64
Figura 71. Original.	65
Figura 72. Verticalidad.	65
Figura 73. Horizontalidad.	66
Figura 74. Ambas dimensiones.	66
Figura 75. Imagen de Lenna convolucionada con máscara vertical y horizontal de Sobel y resultado definitivo de los bordes absolutamente blancos con valor de nivel de gris 255.	66
Figura 76. Imagen original.	68
Figura 77. Imagen Interpolada.	68
Figura 78. ImageJ y procedimiento de compilación del plugin.	70
Figura 79. Mensaje indicador que el plugin ha sido compilado.	70
Figura 80. Ubicación del plugin una vez compilado	70
Figura 81. Ventana principal del plugin y selección de la imagen de referencia original.	71
Figura 82. Ventana principal del plugin y selección de la imagen interpolada para ser comparada con la original de referencia.	72
Figura 83. Ventana principal con el resultado del cálculo de la calidad entre la imagen original y la interpolada.	72
Figura 84. Ventana principal con el resultado del cálculo de la calidad de varias imágenes.	73
Figura 85. Ventana de ayuda del plugin de comparación de resultados.	73

<b>Figura o Tabla</b>	<b>Pág.</b>
<i>Figura 86. Baboon. 500x480</i>	74
<i>Figura 87. Bárbara. 720x 576</i>	74
<i>Figura 88. Cameraman. 256x256</i>	74
<i>Figura 89. Lenna. 512x512</i>	74
<i>Tabla 27. Juego de pruebas a ejecutar en cada una de las cuatro imágenes.</i>	74
<i>Tabla 28. Resultaado de la comparación de las imágenes para los métodos de interpolación y sus configuraciones.</i>	75
<i>Tabla 28. Resultaado de la comparación de las imágenes para los métodos de interpolación y sus configuraciones.</i>	76
<i>Figura 90. Top 5 PSNR Baboon.</i>	77
<i>Figura 91. Top 5 PSNR Bárbara</i>	77
<i>Figura 92. Top 5 PSNR Cameraman.</i>	77
<i>Figura 93. Top 5 PSNR Lenna.</i>	77
<i>Tabla 29. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen 'Baboon'.</i>	78
<i>Tabla 30. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen 'Bárbara'.</i>	78
<i>Tabla 31. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen 'Cameraman'.</i>	78
<i>Tabla 32. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen 'Lenna'.</i>	78
<i>Tabla 33. Comparación Top-5 del método PSNR con el método de bordes por máscaras de Söbel.</i>	79

## 2. Objetivos del TFdM

Durante el transcurso de la formación del “**Máster Universitario en Investigación en Ingeniería del Software y Sistemas Informáticos**” me interesé en gran medida por la visión por computador, siendo este trabajo el resultado de la propuesta formal realizada durante este curso académico **2014-2015**.

Para llegar a este punto, las asignaturas superadas en el máster han sido:

- ✓ *Sistemas Difusos de Apoyo a la Toma de Decisiones.*
- ✓ *Modelado y Simulación de Robots.*
- ✓ *Sistemas de Percepción Visual.*
- ✓ *Gestión y Mejora de Procesos Software.*
- ✓ *Computación Ubicua.*

Tras revisar los algoritmos clásicos de interpolación y habiendo revisado la posibilidad de investigar sobre esos conceptos, la idea definitiva consiste en definir y elaborar un nuevo algoritmo que pretenda cubrir algunas de las ideas que me han ido surgiendo, con el ánimo de implementar un algoritmo de interpolación propio.

La idea fundamental sería el uso líneas de tendencia mediante el cálculo de mínimos cuadrados para pronosticar los píxeles desconocidos, fruto de un incremento de la resolución de una imagen.

El uso de este método surge de la asignatura del máster de toma de decisiones y en cómo representar una preferencia. En este caso particular de visión por computador, la toma de decisión viene dada por la respuesta sobre qué valor desconocido del nivel de gris de un píxel se debe establecer, según el criterio de un conjunto de expertos, en una interpolación propia. Teniendo en cuenta que cada experto es un cálculo matemático representado desde un punto de vista diferente.

Según este punto de vista, esta toma de decisión dado un método de interpolación propio pero finalmente recreando un estudio y una solución a un problema de visión por computador, asume el requisito de no profundizar exclusivamente en un aspecto de una asignatura en concreto, sino en la adquisición de competencias para relacionar los contenidos de las asignaturas del máster.

Esto incluye la implementación práctica del algoritmo propio descrito y también la implementación de los algoritmos clásicos, para realizar una comparación de los resultados entre los métodos. La idea sería implementar un Plug-in en Java para poder ser compilado y ejecutado con la herramienta ImageJ, tanto para el contexto de la interpolación en sí como en el sistema de comparación de resultados, siguiendo un método científico.

Adicionalmente comentar que la idea pasa por una inmersión con imágenes en escala de grises pero también asumir imágenes en color o cómo afrontar esa necesidad.

A efectos prácticos, los puntos que cubre el proyecto son:

- ✓ Prospección, estudio e implementación práctica de los métodos clásicos de interpolación.
- ✓ Investigación de un método propio de interpolación basado en el uso de líneas de tendencia, mediante el cálculo por mínimos cuadrados.
- ✓ Comparación de los resultados obtenidos mediante los diferentes métodos de interpolación propuestos, haciendo uso de métodos científicos, en concreto PSNR.
- ✓ Análisis y discusión de los resultados.
- ✓ Enfoque en imágenes en escala de grises y color.

### 3. Introducción

El uso de **imágenes digitales**, como representaciones bidimensionales de un modelo real o imaginario, en un formato de matriz numérica o mapa de bits, es un concepto hoy en día ampliamente conocido en muchos aspectos.

Aunque la primera cámara digital patentada se remonta a 1975, esta no tomó la primera imagen digital de la historia con sus 0,01 megapíxeles de la época.

El acontecimiento se produjo dos décadas antes, en 1957, cuando Russell Kirsch ideó un dispositivo que permitía escanear imágenes, creando una primera imagen digital de 176x176 píxeles con la imagen de su hijo de tres meses.

Esta tecnología fue ya ampliamente utilizada en la década de los años 60, siendo la exploración espacial un claro ejemplo de ello.

Uno de los primeros retos que podemos afrontar haciendo uso de imágenes digitales es la necesidad de reducir o ampliar su tamaño. Esta tarea se conoce como **escalado** y corresponde con una transformación geométrica de la imagen.



Figura 1. Primera imagen digital escaneada. 1957.

Ya que una imagen digital es discreta no existen valores de intensidad intermedios que puedan atribuirse con las intersecciones de los valores horizontales y verticales, de tal forma que, si realizamos una transformación de la imagen, es preciso calcular el valor de los nuevos píxeles en función de los que se han transformado.

En el caso de la ampliación de una imagen digital obtendremos siempre un conjunto de nuevos píxeles desconocidos que convendrá calcular haciendo uso de alguna técnica matemática.

Por ejemplo, si partimos de una imagen digital de 256x256 píxeles y queremos realizar un escalado al doble de su dimensión, obtendremos una imagen de 512x512 píxeles e implicará que  $\frac{3}{4}$  partes de los píxeles ahora visibles en la imagen de mayor tamaño no existían en la imagen original.

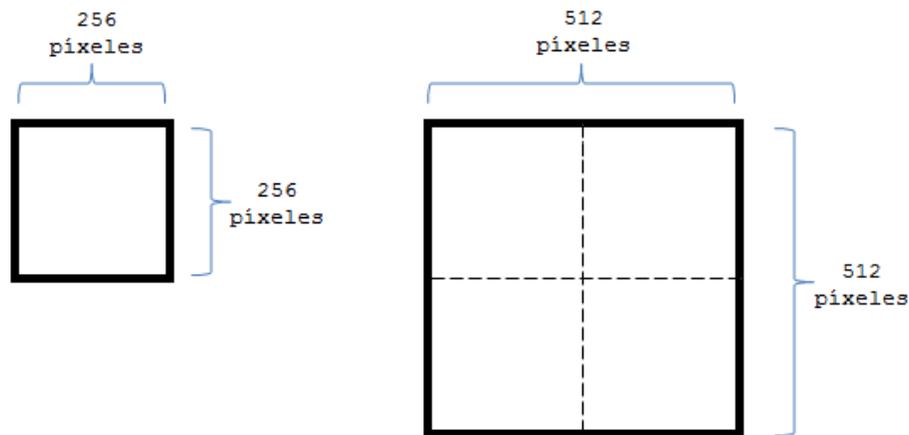


Figura 2. Realizar un escalado al doble de tamaño implica el cálculo del valor de intensidad del 75% de los píxeles.

Para considerar el cálculo de los nuevos píxeles podemos utilizar técnicas de superresolución, en caso de tener más de una imagen de referencia de entrada, o bien hacer uso de técnicas de **interpolación**, que permiten calcular el valor de intensidad de un píxel, en una posición cualquiera, mediante una función que implique los píxeles que le rodean.

Existen diferentes métodos de interpolación clásicos para el tratamiento geométrico de imágenes, entre los que se destacan los de **vecino más próximo**, **bilineal** y **bicúbico**, no obstante podemos idear otros métodos de interpolación.

En nuestro caso, este trabajo tratará el escalado de imágenes digitales mediante el uso de un método propio de interpolación por mínimos cuadrados y la comparación de los resultados con los tres métodos clásicos descritos anteriormente.

## 4. Conceptos preliminares

En este apartado estudiaremos los conceptos necesarios para la comprensión inicial de las bases que forman el sistema propuesto en el presente trabajo.

### 4.1. Tasa de Muestreo de Nyquist

¿A qué velocidad debemos tomar muestras de una señal para poder reconstruirla? Según el Teorema de Muestreo de Nyquist-Shannon, la velocidad mínima de esta **Tasa de Muestreo** debe ser de por lo menos **dos veces la frecuencia máxima** de la señal a capturar.

A efectos prácticos, en una señal de frecuencia homogénea, como sería por ejemplo una señal senoidal de frecuencia fija a 1 Hz, debemos tomar muestras a una velocidad mínima de 2 Hz para poder reconstruir la señal mediante esas muestras, aunque sea una reconstrucción de poca calidad.

En el siguiente ejemplo podemos observar que la señal de la izquierda (señal original que queremos reconstruir mediante un conjunto de sus muestras) está siendo muestreada a diferentes velocidades.

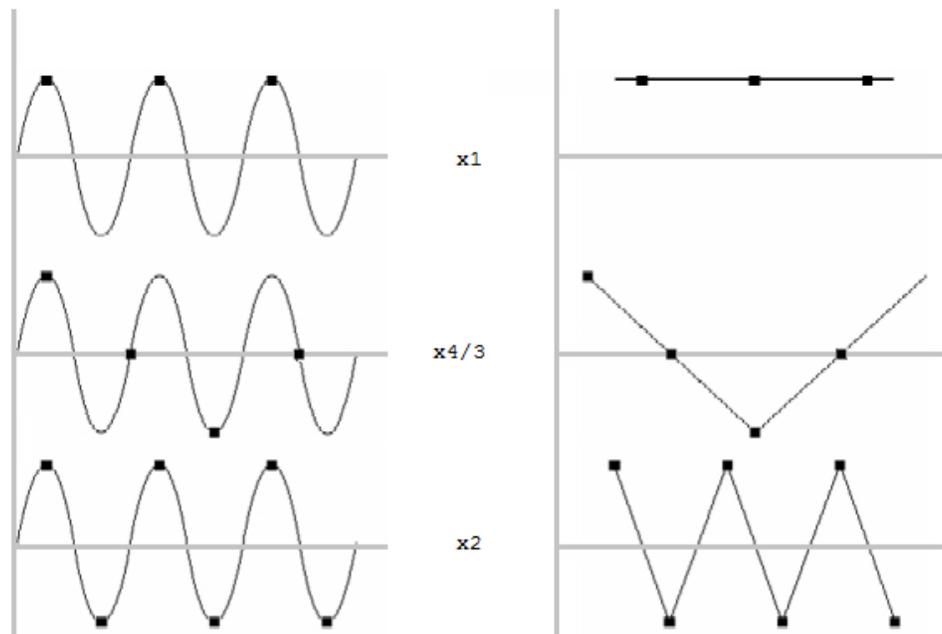


Figura 3. Ejemplos de reconstrucción de una señal senoidal mediante diferentes frecuencias de muestreo.

- ✓ En el primero de los casos se están captando datos a una velocidad igual a la frecuencia de la propia señal. El resultado es que no podemos reconstruir la señal original, quedando una señal reconstruida de amplitud fija.
- ✓ En el segundo de los casos se están captando datos a 1,3 veces la frecuencia de la señal original, una velocidad de muestreo insuficiente para poder reconstruir la señal original mediante las muestras recolectadas. El resultado es la representación de una señal que no se ajusta a la realidad de frecuencia de la señal capturada.
- ✓ En el tercero de los casos se están captando datos al doble de la frecuencia de la señal, que según el teorema de Nyquist es la mínima velocidad con la que podemos reconstruir la señal original. El resultado es que ciertamente la señal reconstruida mantiene su fisonomía pero la calidad de la reconstrucción es mínima, observando una baja resolución.

## 4.2. Aliasing

El **aliasing** es un efecto que provoca que ciertas señales aparezcan de forma indistinguibles cuando han sido muestreadas digitalmente.

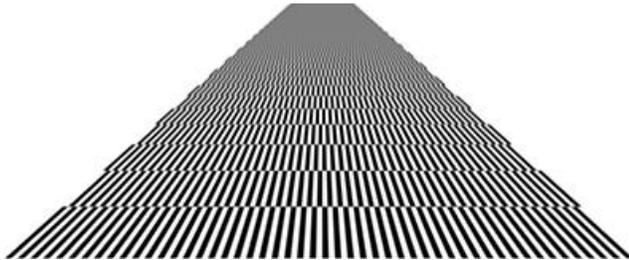


Figura 4. Imagen correctamente muestreada.

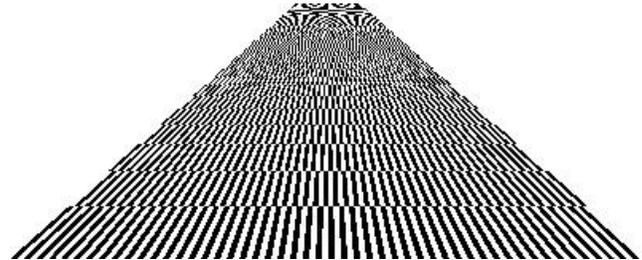


Figura 5. Imagen con efecto aliasing.

Si pretendemos, mediante píxeles de color binario, dibujar una línea en diagonal podemos hacerlo como un conjunto de píxeles (#) dispuestos de la manera expresada en el ejemplo de la derecha (línea diagonal descendente de izquierda a derecha).



Esto nos ofrecerá el efecto de aliasing en el borde de la línea, puesto que existe un corte explícito en la tendencia de la línea dibujada.

Para reducir el impacto de este efecto podemos **suavizar** o difuminar las transiciones de los píxeles, utilizando píxeles de tonalidad diferente, por ejemplo medio tono (\*) o un tercio de tono (.) tal y como se muestra a continuación.



El siguiente ejemplo representa, en la parte superior una línea en diagonal, utilizando el blanco y negro como colores binarios y la corrección mediante varios tonos de gris que suavizan las transiciones.



Figura 6. Corrección básica del efecto aliasing.

### 4.3. Interpolación

Conociendo la información de una señal es posible recuperar la pérdida de datos, provocada en el momento de su adquisición, siendo el grado de ésta pérdida dependiente del muestreo o resolución del dispositivo que ha capturado la señal.

Para alcanzar este objetivo se utiliza el concepto de **Interpolación** que nos permite la **reconstrucción de nuevos datos** partiendo de un conjunto de datos reales y aislados ya conocidos, denominados **datos discretos**.

Tal y como ya hemos avanzado anteriormente en la introducción, la interpolación pretende calcular el valor desconocido de intensidad de un píxel, haciendo uso del valor de intensidad conocido de los píxeles que lo rodean.

A modo de ejemplo veamos una señal con pérdida de información y su reconstrucción mediante el concepto de interpolación.

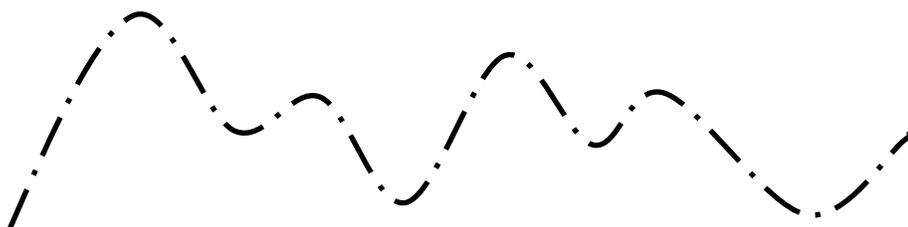


Figura 7. Señal con pérdida de información.

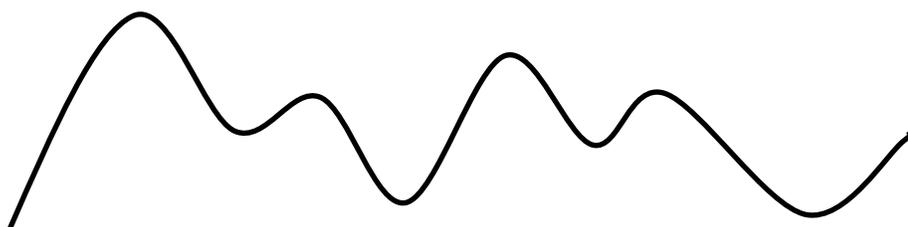


Figura 8. Señal reconstruida por interpolación.

Como condición del teorema de muestreo de **Nyquist-Shannon**, la reconstrucción sería posible a partir de datos obtenidos mediante una tasa de muestreo dos o más veces más veloz a la frecuencia máxima de la señal capturada. Tal y como hemos visto anteriormente, esta frecuencia de muestreo que ofrece una mínima garantía de reconstrucción se denomina Tasa de Muestreo de Nyquist.

## 4.4. Distancia Euclídea

La **distancia euclídea** es el recorrido entre dos puntos de un espacio euclídeo calculada a partir del teorema de Pitágoras. En un espacio bidimensional, la distancia euclídea entre dos puntos P y Q de coordenadas cartesianas  $(x_1, y_1)$  y  $(x_2, y_2)$  respectivamente se calcula mediante la siguiente expresión:

$$d_E(P, Q) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Podemos utilizar la distancia euclídea para determinar la distancia que separa un punto P de varios puntos arbitrarios o conocidos del espacio bidimensional.

Por ejemplo, calculemos la distancia euclídea entre el punto P y los puntos i y j del plano descrito en el gráfico de la derecha.

Distancia euclídea  $d_1$ , entre los puntos (P,i):

$$d_1 = d_E(i, P) = \sqrt{(5 - 2)^2 + (3 - 4)^2} = \sqrt{10} = 3,16$$

Distancia euclídea  $d_2$ , entre los puntos (P,j):

$$d_2 = d_E(j, P) = \sqrt{(5 - 6)^2 + (3 - 1)^2} = \sqrt{5} = 2,23$$

Comparando los resultados podemos observar que el punto más cercano a P es el punto j.

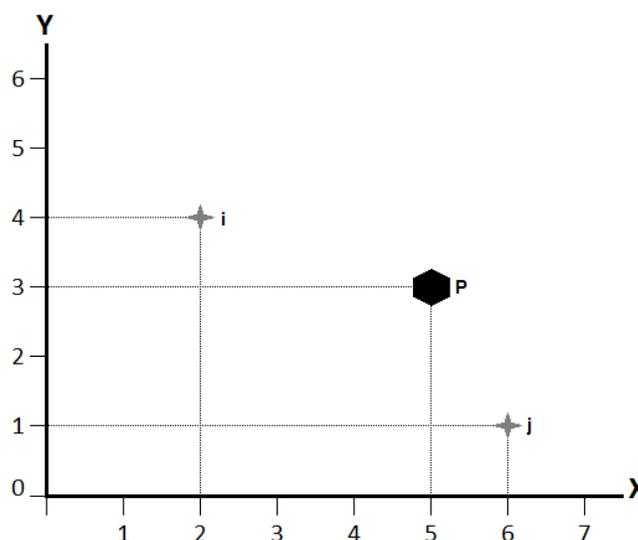


Figura 9. Representación  $d_E$  entre el punto 'P' y los puntos 'i' y 'j'.

## 4.5. Interpolación Lineal

Uno de los principios investigados en los que hemos fundamentado este trabajo es en la idea de cómo conocer el valor de intensidad de un píxel intermedio entre dos píxeles con valor conocido.

Planteando esta incógnita en una escala de ocho píxeles donde el primer píxel, que denominaremos 'A', es el color rojo (RGB=255,0,0) y el último píxel, que denominaremos 'B', es el color verde (RGB=0,255,0), ¿qué valores deben tomar los seis píxeles intermedios?

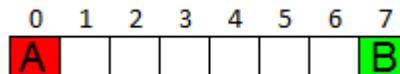


Figura 10. Representación de ocho píxeles horizontales.

Mediante el uso de una **interpolación lineal**, podemos plantear una posible solución mediante una de las formas más básicas de estimación, basada en nuestro caso en la propiedad del color de cualquier punto arbitrario desconocido entre dos o más puntos conocidos.

Por ejemplo, ¿cómo calcularemos el valor de la mezcla del color rojo y verde en el quinto píxel que marcamos en el siguiente gráfico como 'X'?

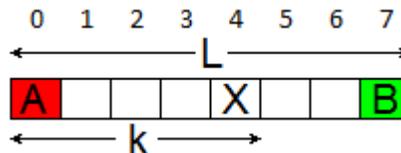


Figura 11. Definición gráfica para el cálculo del píxel 'X' según el gradiente entre los píxeles 'A' y 'B'

La expresión con la que se apoya la interpolación lineal viene dada por:

$$\frac{X - A}{k} = \frac{B - A}{L} \quad X = A + k \frac{(B - A)}{L}$$

De este modo para el caso del ejemplo descrito anteriormente, podemos calcular los valores intermedios de mezcla de rojo y verde, para el conjunto de píxeles desconocidos y ubicados entre los píxeles 'A' y 'B'.

Píxel	k	Rojo	Verde	Azul
A	0	255	0	0
	1			0
	2			0
	3			0
X	4	109	146	0
	5			0
	6			0
B	7	0	255	0

Tabla 1. Representación RGB del ejemplo.

$$X_{Rojo} = A + k \frac{(B - A)}{L} = 255 + 4 \frac{(0 - 255)}{7} = 109,28 \sim 109$$

$$X_{Verde} = A + k \frac{(B - A)}{L} = 0 + 4 \frac{(255 - 0)}{7} = 145,71 \sim 146$$

El resultado de la mezcla de color rojo y verde en el quinto píxel queda definida con un nivel de intensidad de Rojo del 109 y un nivel de intensidad de Verde de 146.

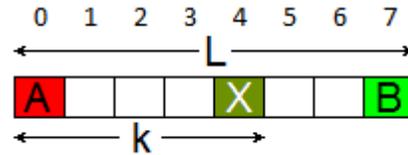


Figura 12. Representación del nivel de mezcla de color para el píxel 'X'.

De este modo, formalizando el cálculo del resto de píxeles desconocidos, siguiendo la expresión descrita obtenemos la transición entre el color rojo y el color verde para los píxeles desconocidos.

Píxel	k	Rojo	Verde	Azul
A	0	255	0	0
	1	219	36	0
	2	182	73	0
	3	146	109	0
X	4	109	146	0
	5	73	182	0
	6	36	219	0
B	7	0	255	0

Tabla 2. Representación RGB completa del ejemplo.

La matriz de píxeles quedaría construida con el siguiente degradado, entre el color rojo y el color verde.

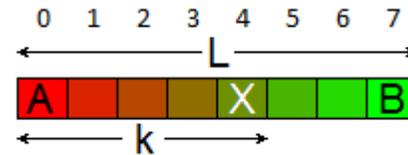


Figura 13. Representación del nivel de mezcla para todos los píxeles.

## 4.6. Programación del sistema interpolación en Java como un componente para ImageJ

Para llevar a cabo prácticamente los cálculos del método de interpolación propuesto se ha optado por realizar una aplicación Java que pueda ser ejecutada como un **Plugin de ImageJ** <http://imagej.nih.gov/ij>



Figura 14. Logotipo de la aplicación ImageJ.

ImageJ es una aplicación de procesamiento de imágenes digitales de dominio público, programado también en **Java** por el **National Institutes of Health (NIH)**, un grupo de instituciones del gobierno de Estados Unidos, cuyos objetivos se centran en la investigación médica.

**ImageJ** es una aplicación utilizada por la comunidad científica e investigadora para resolver problemas de procesado y análisis de imágenes principalmente médicas pero también asociadas a otras disciplinas.

Aprovecharemos la posibilidad que ofrece esta aplicación para incorporar aplicaciones personalizadas programadas en Java bajo sus **librerías de complementos (plugins)** y poder ejecutarlas sobre una o varias imágenes abiertas desde ImageJ.

Mediante el uso de Java programaremos no sólo los cálculos técnicos de la interpolación, sino las transformaciones necesarias y las ventanas gráficas del plugin.

Definimos pues el método de Interpolación mediante el uso de Mínimos Cuadrados en diferentes pasos del plugin para ImageJ, que veremos a continuación.

## 5. Interpolaciones Clásicas

En este capítulo estudiaremos las transformaciones geométricas de interpolación clásicas que, por su uso habitual en **aplicaciones gráficas de mercado**, conviene tener en cuenta para poder avanzar en la comprensión de la interpolación propia que presentamos en este trabajo y que veremos en capítulos sucesivos.

### 5.1. Introducción

Partimos de un mapa de 3x2 píxeles.

Esos seis píxeles (A,B,C,D,E,F) se encuentran en las siguientes coordenadas del mapa:

x	y	Px
0	0	A
1	0	B
2	0	C
0	1	D
1	1	E
2	1	F

Figura 15. Coordenadas para seis píxeles

La representación gráfica de ese mapa de píxeles, siguiendo las coordenadas descritas es el siguiente:

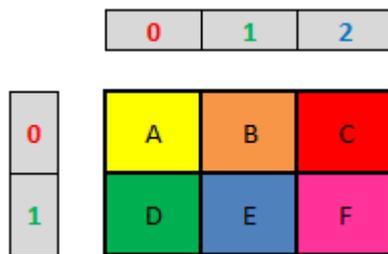


Figura 16. Imagen de 2x3 píxeles.

Los píxeles tendrán cierto nivel de intensidad, en nuestro caso y para los ejemplos que veremos más tarde se definen los siguientes:

A = 1	D = 7
B = 4	E = 6
C = 8	F = 9

Si pretendemos escalar este mapa de píxeles al doble de su dimensión bidimensional, obtendremos un mapa de 6x4 píxeles:

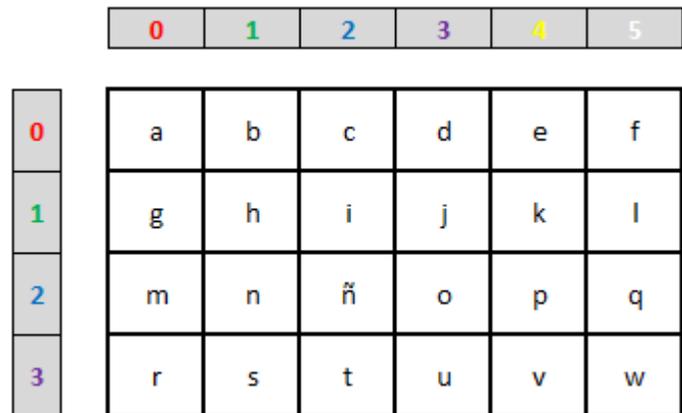


Figura 17. Imagen de 4x6 píxeles objetivo del escalado x2.

Este mapa ya no tendrá seis píxeles y habrá pasado a tener veinticuatro, de forma que si mantenemos los seis píxeles originales, sólo conoceremos  $\frac{1}{4}$  parte del total de píxeles de este nuevo mapa.

Para comprender el concepto de interpolación es preciso tener en mente que tras una transformación geométrica, los píxeles del mapa o rejilla original (en el ejemplo la matriz de 3x2 píxeles), no tienen por qué coincidir con los de la rejilla destino (en el ejemplo la matriz de 6x4 píxeles).

¿Qué valores numéricos corresponderán a estos veinticuatro nuevos píxeles? La respuesta la encontraremos en el cálculo de una interpolación, haciendo uso de los píxeles de la matriz original de la que hemos partido para escalar su dimensión.

Para ello es preciso mantener dos ideas básicas:

- ✓ Por cada píxel del mapa original se han formado cuatro píxeles en el mapa destino.
- ✓ Esta explosión de cada píxel original en cuatro píxeles en el mapa destino, implica una visión superpuesta de ambos mapas de píxeles a modo de dos capas. Esto nos facilitará la comprensión del cálculo de la interpolación

Estas ideas pueden plasmarse gráficamente de la siguiente manera:

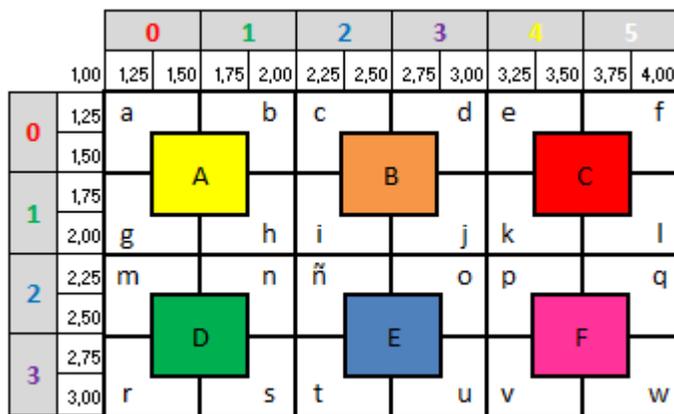


Figura 18. Imagen 2x3 original superpuesta en la imagen 4x6.

De este modo, el mapa original de 3x2 con los píxeles (A,B,C,D,E,F) se adapta de sus coordenadas cartesianas originales (x,y) a las nuevas coordenadas cartesianas del nuevo mapa de 6x4 (x',y'):

x	y	Px	x'	y'
0	0	A	1,5	1,5
1	0	B	2,5	1,5
2	0	C	3,5	1,5
0	1	D	1,5	2,5
1	1	E	2,5	2,5
2	1	F	3,5	2,5

Figura 19. Coordenadas asociadas a la superposición anterior.

A partir de este momento ya estamos en disposición de interpolar el valor de intensidad de los píxeles del nuevo mapa escalado y calcular el valor de esos veinticuatro píxeles que lo forman (a ,b ,c ,d ,e ,f ,g ,h ,i ,j ,k ,l ,m ,n ,ñ ,o , p ,q , r , s , t , u , v , w).

A continuación veremos cómo proceden las interpolaciones clásicas por vecino más próximo (Nearest-neighbor interpolation), bilineal (Bilinear interpolation) y bicúbica (Bicubic interpolation) en el cálculo de estos píxeles desconocidos y mostraremos cómo se formaliza el cálculo para cuatro de los píxeles (j, k, o, p) de esta nueva matriz, a modo de ejemplo.

## 5.2. Interpolación por vecino más próximo

Uno de los métodos más sencillos para calcular el valor de intensidad de un píxel desconocido, es tomar el mismo valor de intensidad que el píxel más cercano de los cuatro que le rodean.

Para el cálculo del píxel más cercano es posible hacer uso de la **distancia euclídea**.

El núcleo de interpolación  $h(x)$  quedaría definido de la siguiente manera:

$$h(x) = \begin{cases} 1 & \text{si } 0 < |x| < 0,5 \\ 0 & \text{de otro modo} \end{cases}$$

### 5.2.1. Características principales

- ✓ Se trata de un método básico de interpolación.
- ✓ No requiere de grandes tiempos y recursos de procesado.
- ✓ Este tipo de interpolación puede provocar un efecto *aliasing* en la imagen resultante puesto que no existe transición en los niveles de intensidad de los píxeles escalados.
- ✓ A efectos prácticos esta interpolación aumenta el tamaño de cada píxel.

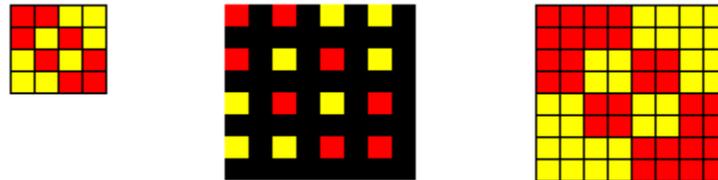


Figura 20. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.

### 5.2.2. Cálculos de la interpolación

A modo de ejemplo calcularemos los valores de intensidad de los píxeles (j, k, o, p) del mapa de 6x4 teniendo en cuenta los cuatro píxeles más cercanos del mapa de 3x2 (B, C, E, F).

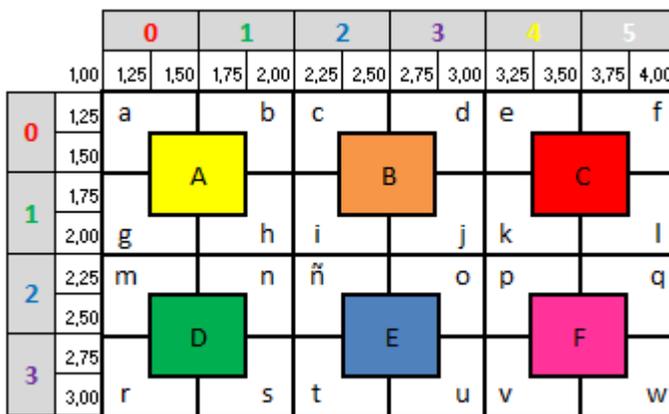


Figura 21. Imagen 2x3 original superpuesta en la imagen 4x6.

Observando la matriz podemos conocer la posición (X,Y) de todos los píxeles implicados:

Mapa	Píxel	X	Y	Nivel de Intensidad
6x4	j	2,75	1,75	Valor a calcular.
	k	3,25	1,75	Valor a calcular.
	o	2,75	2,25	Valor a calcular.
	p	3,25	2,25	Valor a calcular.
3x2	B	2,5	1,5	4
	C	3,5	1,5	8
	E	2,5	2,5	6
	F	3,5	2,5	9

Tabla 3. Valores de los mapas superpuestos.

El objetivo será calcular los niveles de intensidad de los píxeles (j, k, o, p).

A modo de ejemplo mostraremos prácticamente cómo calcular la distancia euclídea y el núcleo de interpolación teniendo en cuenta el píxel de intensidad desconocida 'j' y la relación de éste con los píxeles de intensidad conocida (B, C, E, F):

Distancia Euclídea (dE)	Núcleo de Interpolación h(j)
$d_E(B, j) = \sqrt{(2,75 - 2,5)^2 + (1,75 - 1,5)^2} = 0,35$	Como $0 <  d_E(B, j)  < 0,5$ entonces $h = 1$
$d_E(C, j) = \sqrt{(2,75 - 3,5)^2 + (1,75 - 1,5)^2} = 0,79$	Como $ d_E(C, j)  \geq 0,5$ entonces $h = 0$
$d_E(E, j) = \sqrt{(2,75 - 2,5)^2 + (1,75 - 2,5)^2} = 0,79$	Como $ d_E(E, j)  \geq 0,5$ entonces $h = 0$
$d_E(F, j) = \sqrt{(2,75 - 3,5)^2 + (1,75 - 2,5)^2} = 1,06$	Como $ d_E(F, j)  \geq 0,5$ entonces $h = 0$

Tabla 4. Cálculo de la Distancia Euclídea y el núcleo de interpolación para los píxeles de ejemplo.

De este modo, calculando todos los valores y trasladándolos en una tabla obtendremos los siguientes resultados:

Cálculos para el píxel 'j':

Distancia Euclídea	Núcleo de interpolación h(j)
$dE(B, j) = 0,35$	1
$dE(C, j) = 0,79$	0
$dE(E, j) = 0,79$	0
$dE(F, j) = 1,06$	0

Cálculos para el píxel 'k':

Distancia Euclídea	Núcleo de interpolación h(k)
$dE(B, k) = 0,79$	0
$dE(C, k) = 0,35$	1
$dE(E, k) = 1,06$	0
$dE(F, k) = 0,79$	0

Cálculos para el píxel 'o':

Distancia Euclídea	Núcleo de interpolación h(o)
$dE(B, o) = 0,79$	0
$dE(C, o) = 1,06$	0
$dE(E, o) = 0,35$	1
$dE(F, o) = 0,79$	0

Cálculos para el píxel 'p':

Distancia Euclídea	Núcleo de interpolación h(p)
$dE(B, p) = 1,06$	0
$dE(C, p) = 0,79$	0
$dE(E, p) = 0,79$	0
$dE(F, p) = 0,35$	1

Tabla 5. Cálculo del núcleo de interpolación para los píxeles 'j', 'k', 'o' y 'p'.

Con la información calculada y el valor de intensidad de los píxeles conocidos de la matriz 3x2 podemos proceder con el cálculo del valor de intensidad de los píxeles (j, k, o, p) de la matriz 6x4.

El nivel de intensidad de cada píxel desconocido será el sumatorio del producto del nivel de intensidad de cada píxel conocido y el núcleo de interpolación para cada caso.

Distancia Euclídea	Núcleo de interpolación $h(j)$	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$dE(B,j) = 0,35$	1	$B = 4$	$h(j) \times B = 1 \times 4 = 4$
$dE(C,j) = 0,79$	0	$C = 8$	$h(j) \times C = 0 \times 8 = 0$
$dE(E,j) = 0,79$	0	$E = 6$	$h(j) \times D = 0 \times 6 = 0$
$dE(F,j) = 1,06$	0	$F = 9$	$h(j) \times E = 0 \times 9 = 0$
<b>Nivel de Intensidad del píxel 'j'</b>			$4 + 0 + 0 + 0 = 4$

Tabla 6. Cálculo del nivel de intensidad del píxel 'j'.

Distancia Euclídea	Núcleo de interpolación $h(k)$	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$dE(B,k) = 0,79$	0	$B = 4$	$h(k) \times B = 0 \times 4 = 0$
$dE(C,k) = 0,35$	1	$C = 8$	$h(k) \times C = 1 \times 8 = 8$
$dE(E,k) = 1,06$	0	$E = 6$	$h(k) \times D = 0 \times 6 = 0$
$dE(F,k) = 0,79$	0	$F = 9$	$h(k) \times E = 0 \times 9 = 0$
<b>Nivel de Intensidad del píxel 'k'</b>			$0 + 8 + 0 + 0 = 8$

Tabla 7. Cálculo del nivel de intensidad del píxel 'k'.

Distancia Euclídea	Núcleo de interpolación $h(o)$	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$dE(B,o) = 0,79$	0	$B = 4$	$h(o) \times B = 0 \times 4 = 0$
$dE(C,o) = 1,06$	0	$C = 8$	$h(o) \times C = 0 \times 8 = 0$
$dE(E,o) = 0,35$	1	$E = 6$	$h(o) \times D = 1 \times 6 = 6$
$dE(F,o) = 0,79$	0	$F = 9$	$h(o) \times E = 0 \times 9 = 0$
<b>Nivel de Intensidad del píxel 'o'</b>			$0 + 0 + 6 + 0 = 6$

Tabla 8. Cálculo del nivel de intensidad del píxel 'o'.

Distancia Euclídea	Núcleo de interpolación $h(p)$	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$dE(B,p) = 1,06$	0	$B = 4$	$h(p) \times B = 0 \times 4 = 0$
$dE(C,p) = 0,79$	0	$C = 8$	$h(p) \times C = 0 \times 8 = 0$
$dE(E,p) = 0,79$	0	$E = 6$	$h(p) \times D = 0 \times 6 = 0$
$dE(F,p) = 0,35$	1	$F = 9$	$h(p) \times E = 1 \times 9 = 9$
<b>Nivel de Intensidad del píxel 'p'</b>			$0 + 0 + 0 + 9 = 9$

Tabla 9. Cálculo del nivel de intensidad del píxel 'p'.

Una vez calculado los niveles de intensidad de los píxeles (j, k, o, p) podemos completar la tabla con sus valores.

Mapa	Píxel	X	Y	Nivel de Intensidad
6x4	j	2,75	1,75	4
	k	3,25	1,75	8
	o	2,75	2,25	6
	p	3,25	2,25	9
3x2	B	2,5	1,5	4
	C	3,5	1,5	8
	E	2,5	2,5	6
	F	3,5	2,5	9

Tabla 10. Cálculo del nivel de intensidad en los mapas.

Nótese que, como era de esperar en una interpolación por vecino más próximo, el nivel de intensidad del píxel 'j' es el mismo que el del píxel 'B', el de 'k'=C', el de 'o'=E y el de 'p' = 'F'.

De esta forma, la representación del nivel de intensidad de los píxeles (j, k, o, p) quedaría representada de la siguiente manera:

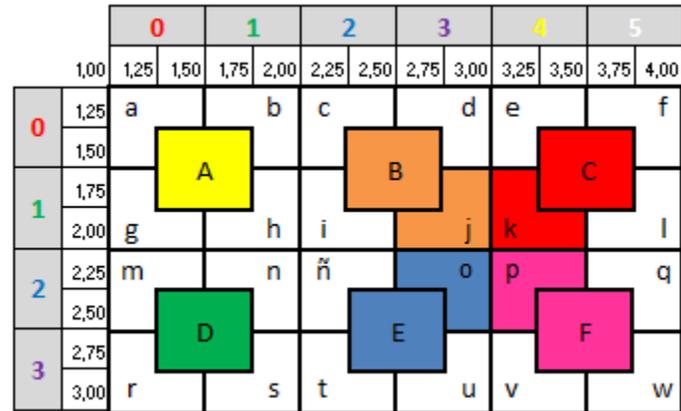


Figura 22. Representación del nivel los píxeles 'j', 'k', 'o' y 'p', una vez calculados.

Por otra parte, formalizando el cálculo para todos los píxeles de la matriz 6x2, obtendríamos la interpolación final:

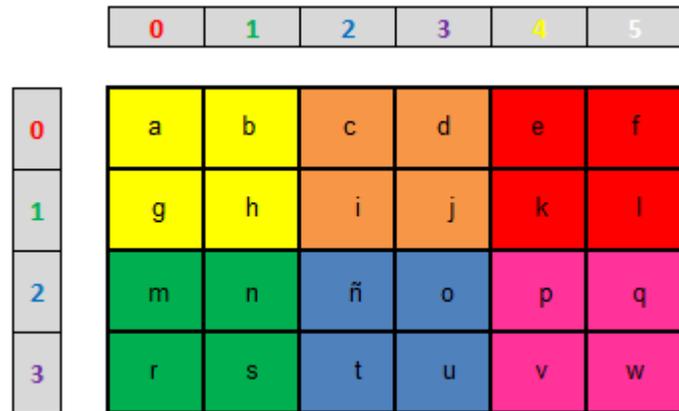


Figura 23. Representación de la interpolación una vez finalizada.

## 5.3. Interpolación Bilineal

Otro de los métodos clásicos que permiten interpolar con un resultado más satisfactorio, en detrimento del coste computacional, es el conocido como la interpolación bilineal.

Este método asigna al píxel de intensidad desconocida el valor medio ponderado de las intensidades de los píxeles que le rodean.

La función de interpolación  $h(x)$  quedaría definida de la siguiente manera:

$$h(x) = \begin{cases} 1 - |x| & \text{si } 0 < |x| < 1 \\ 0 & \text{de otro modo} \end{cases}$$

### 5.3.1. Características principales

- ✓ Se tiene en cuenta los valores de intensidad de los píxeles conocidos que rodean a cada píxel desconocido en una vecindad de los 2x2 píxeles más cercanos.
- ✓ Se calcula el promedio ponderado de esos cuatro píxeles conocidos para calcular el valor interpolado de un píxel desconocido dado.
- ✓ Típicamente es una interpolación que ofrece una mejor calidad en los resultados de escalado, con respecto a la interpolación del vecino más próximo.
- ✓ Requiere de un mayor tiempo y recursos de procesado que la interpolación por vecino más próximo.
- ✓ Habitualmente provoca un suavizado de los detalles.

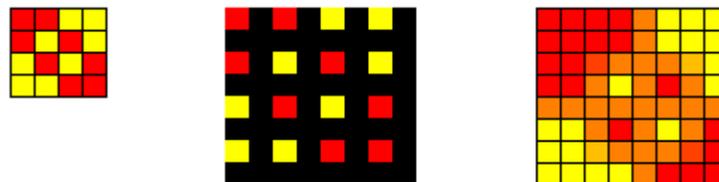


Figura 24. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.

### 5.3.2. Cálculos de la interpolación

El objetivo será calcular los niveles de intensidad de los píxeles (j, k, o, p) según el núcleo de interpolación descrito.

A modo de ejemplo, el núcleo de interpolación teniendo en cuenta el píxel de intensidad desconocida 'j' y la relación de éste con los píxeles de intensidad conocida (B, C, E, F).

Distancia Euclídea	Núcleo de Interpolación h(j)
$d_E(B, j) = 0,35$	Como $0 <  d_E(B, j)  < 1$ entonces $h = 1 -  d_E(B, j)  = 1 - 0,35 = 0,65$
$d_E(C, j) = 0,79$	Como $0 <  d_E(C, j)  < 1$ entonces $h = 1 -  d_E(C, j)  = 1 - 0,79 = 0,21$
$d_E(E, j) = 0,79$	Como $0 <  d_E(E, j)  < 1$ entonces $h = 1 -  d_E(E, j)  = 1 - 0,79 = 0,21$
$d_E(F, j) = 1,06$	Como $ d_E(F, j)  > 1$ entonces $h = 0$

Tabla 11. Cálculo de la Distancia Euclídea y el núcleo de interpolación para los píxeles de ejemplo.

Con la información calculada y el valor de intensidad de los píxeles conocidos de la matriz 3x2 podemos proceder con el cálculo del valor de intensidad de los píxeles (j, k, o, p) de la matriz 6x4:

El nivel de intensidad de cada píxel desconocido será el sumatorio del producto del nivel de intensidad de cada píxel conocido y el núcleo de interpolación para cada caso, redondeando los decimales puesto que sólo podemos establecer técnicamente un nivel de intensidad de píxel mediante un valor entero.

Distancia Euclídea	Núcleo de interpolación h(j)	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$d_E(B, j) = 0,35$	0,65	B = 4	$h(j) \times B = 0,65 \times 4 = 2,6$
$d_E(C, j) = 0,79$	0,21	C = 8	$h(j) \times C = 0,21 \times 8 = 1,68$
$d_E(E, j) = 0,79$	0,21	E = 6	$h(j) \times D = 0,21 \times 6 = 1,26$
$d_E(F, j) = 1,06$	0	F = 9	$h(j) \times E = 0 \times 9 = 0$
<b>Nivel de Intensidad del píxel 'j'</b>			$2,6 + 1,68 + 1,26 + 0 = 5,54 \sim 6$

Tabla 12. Cálculo del nivel de intensidad del píxel 'j'.

Distancia Euclídea	Núcleo de interpolación h(k)	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$d_E(B, k) = 0,79$	0,21	B = 4	$h(k) \times B = 0,21 \times 4 = 0,84$
$d_E(C, k) = 0,35$	0,65	C = 8	$h(k) \times C = 0,65 \times 8 = 5,2$
$d_E(E, k) = 1,06$	0	E = 6	$h(k) \times D = 0 \times 6 = 0$
$d_E(F, k) = 0,79$	0,21	F = 9	$h(k) \times E = 0,21 \times 9 = 1,89$
<b>Nivel de Intensidad del píxel 'k'</b>			$0,84 + 5,2 + 0 + 1,89 = 7,93 \sim 8$

Tabla 13. Cálculo del nivel de intensidad del píxel 'k'.

Distancia Euclídea	Núcleo de interpolación h(o)	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$d_E(B, o) = 0,79$	0,21	B = 4	$h(o) \times B = 0,21 \times 4 = 0,84$
$d_E(C, o) = 1,06$	0	C = 8	$h(o) \times C = 0 \times 8 = 0$
$d_E(E, o) = 0,35$	0,65	E = 6	$h(o) \times D = 0,65 \times 6 = 3,9$
$d_E(F, o) = 0,79$	0,21	F = 9	$h(o) \times E = 0,21 \times 9 = 1,89$
<b>Nivel de Intensidad del píxel 'o'</b>			$0,84 + 0 + 3,9 + 1,89 = 6,63 \sim 7$

Tabla 14. Cálculo del nivel de intensidad del píxel 'o'.

Distancia Euclídea	Núcleo de interpolación $h(p)$	Nivel de Intensidad conocido matriz 3x2	Cálculo Nivel de Intensidad matriz 6x4
$dE(B,p) = 1,06$	0	B = 4	$h(p) \times B = 0 \times 4 = 0$
$dE(C,p) = 0,79$	0,21	C = 8	$h(p) \times C = 0,21 \times 8 = 1,68$
$dE(E,p) = 0,79$	0,21	E = 6	$h(p) \times D = 0,21 \times 6 = 1,26$
$dE(F,p) = 0,35$	0,65	F = 9	$h(p) \times E = 0,65 \times 9 = 5,85$
<b>Nivel de Intensidad del píxel 'p'</b>			$0 + 1,68 + 1,26 + 5,85 = 8,79 \sim 9$

Tabla 15. Cálculo del nivel de intensidad del píxel 'p'.

Una vez calculado los niveles de intensidad de los píxeles (j, k, o, p) podemos completar la tabla con sus valores.

Mapa	Píxel	X	Y	Nivel de Intensidad
6x4	j	2,75	1,75	6
	k	3,25	1,75	8
	o	2,75	2,25	7
	p	3,25	2,25	9
3x2	B	2,5	1,5	4
	C	3,5	1,5	8
	E	2,5	2,5	6
	F	3,5	2,5	9

Tabla 16. Cálculo del nivel de intensidad en los mapas.

De esta forma, la representación del nivel de intensidad de los píxeles (j, k, o, p) quedaría representada de la siguiente manera:

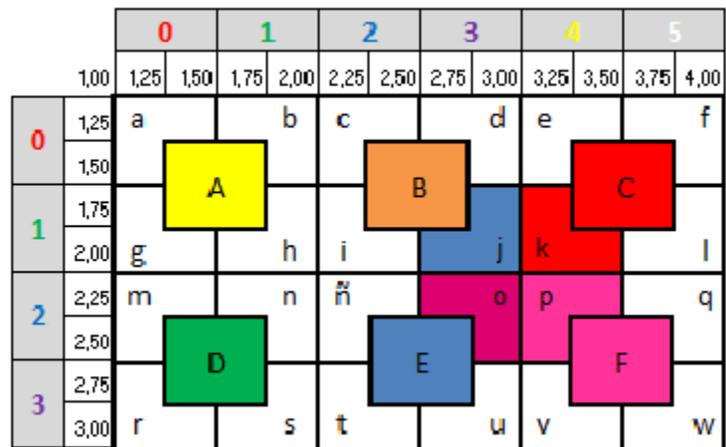


Figura 25. Representación del cálculo de los 'j', 'k', 'o' y 'p'.

Nótese que respecto a la interpolación por vecino más próximo, los valores calculados para los píxeles (j, k, o, p) difieren en su valor de intensidad de gris, en tanto en cuanto el píxel 'j' ahora ha tomado el valor del píxel conocido 'E' y el píxel 'o' ha tomado un valor de intensidad promedio entre el píxel 'C' y 'F'.

## 5.4. Interpolación Bicúbica

Una tercera forma de formalizar una interpolación con un resultado más satisfactorio incluso que la interpolación bilineal es la interpolación bicúbica.

La expresión del núcleo de interpolación  $h(x)$  para este sistema es:

$$h(x) = \begin{cases} 1 - 2|x|^2 + |x|^3 & \text{si } 0 < |x| < 1 \\ 4 - 8|x| + 5|x|^2 - |x|^3 & \text{si } 1 < |x| < 2 \\ 0 & \text{de otro modo} \end{cases}$$

### 5.4.1. Características principales

- ✓ Se tiene en cuenta los valores de intensidad de los píxeles conocidos que rodean a cada píxel desconocido en una vecindad de los 4x4 píxeles más cercanos.
- ✓ Al encontrarse a distancias distintas del píxel de valor desconocido, se calcula su valor otorgando un mayor peso a los píxeles más cercanos.
- ✓ Ofrece una mejor calidad que los sistemas de vecino más próximo y bilineal.
- ✓ Las imágenes resultantes son más nítidas.
- ✓ Requiere de un mayor tiempo y recursos de procesamiento que la interpolación bilineal.
- ✓ Suele ser el procedimiento habitual utilizado en aplicaciones de edición de imágenes.

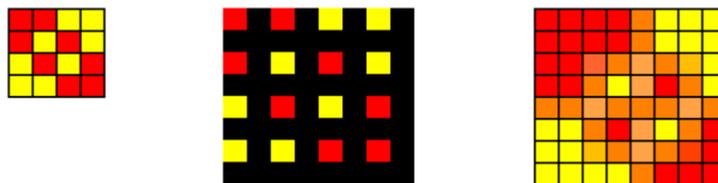


Figura 26. Representación de la interpolación. Imagen original, imagen x2 espaciada e imagen escalada e interpolada.

## 6. Interpolación mediante la aproximación por mínimos cuadrados

Como punto clave de este trabajo de investigación hemos querido razonar un nuevo sistema de interpolación, que nos permita obtener una imagen de salida de mayor tamaño a través de una única imagen de entrada, y posteriormente comparar el resultado con los métodos de interpolación clásicos vistos anteriormente.

El objetivo que persigue este ejercicio es investigar, identificar y elaborar un sistema propio que permita interpolar los píxeles de una imagen de entrada para obtener una imagen de salida dos veces mayor.

El sistema ideado concede una gran representación en el uso de la técnica de análisis numérico por Mínimos Cuadrados.

### 6.1. Línea de tendencia mediante el cálculo de Mínimos Cuadrados

En el caso de no estar sujeto al cálculo de un píxel ubicado entre dos píxeles de valor conocido, sino en el deseo de tener en cuenta más píxeles de su entorno con diferentes intensidades, podemos tener en cuenta algún sistema de estimación.

#### 6.1.1. Conceptos preliminares

Mediante el análisis numérico por **Mínimos Cuadrados** podemos pronosticar el valor de puntos desconocidos sobre una línea de tendencia, conociendo previamente un conjunto de puntos discretos dados.

Veamos un ejemplo, dado un conjunto de puntos conocidos (X,Y), establecemos sus valores en un gráfico de dispersión, calculando la línea de tendencia mediante el uso de Mínimos Cuadrados.

X	Y
0	0
0	1
1	1
2	2
2	3
3	3
3	4

Tabla 17. Datos discretos conocidos.

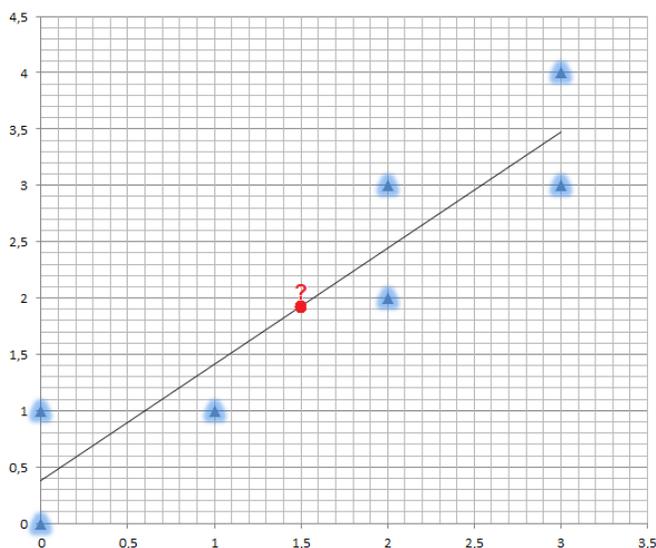


Figura 27. Línea de tendencia por mínimos cuadrados de puntos dispersos en un espacio.

Vemos que la línea de tendencia por Mínimos Cuadrados establece el ajuste del camino de la representación de los puntos, en este caso ascendente.

De ese modo podemos estimar todos los puntos de la recta de tendencia para cualquier valor del eje de las abscisas 'X'. Por ejemplo, ¿qué valor del eje de ordenadas 'Y' se determinaría en la posición  $X=1,5$  sobre la línea de tendencia? El cálculo de los mínimos cuadrados nos define que el valor de 'Y' será 1,9265.

**Asociando el eje de abscisas 'X' como la posición de los píxeles en una recta y el eje de ordenadas 'Y' como el nivel de gris de los píxeles, podemos estimar mediante líneas de tendencia el nivel de gris de un píxel desconocido ubicado en una posición 'X' dada. El cálculo de estas líneas de tendencia se formaliza mediante el cálculo de Mínimos Cuadrados.**

Esta definición podemos representarla de forma gráfica mediante un ejemplo de siete píxeles (A, B, C, D, E, F) en la escala de grises (0 a 255), pero donde sólo conocemos el nivel de gris en cuatro de los siete píxeles (A, C, E, G). Los píxeles (B, D, F) tienen un nivel de gris desconocido.

A	0
B	?
C	128
D	?
E	191
F	?
G	217

Figura 28. Siete píxeles donde 'B', 'D' y 'F' son píxeles con nivel de gris desconocidos.

Si representamos de forma isométrica estos píxeles teniendo en cuenta la definición descrita anteriormente, obtendremos la recta de píxeles y en la dimensión de su altura el nivel de gris de cada uno de ellos.

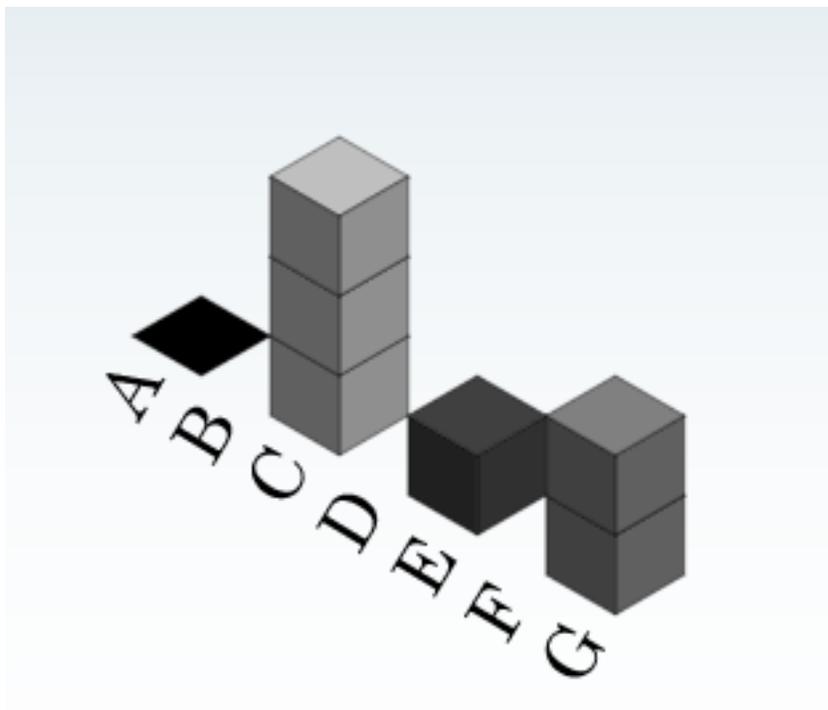


Figura 29. Representación isométrica de los siete píxeles, teniendo en cuenta que la dimensión de la altura corresponde con el nivel de intensidad del píxel.

El valor de nivel de gris debemos establecer en los píxeles (B,D,F) puede ser entonces estimado mediante una línea de tendencia entre los píxeles con valor conocido (A, C, E, G) calculándola mediante Mínimos Cuadrados.

De esta forma la línea de tendencia marcará la altitud de cada píxel desconocido, que en nuestra dimensión corresponde con su nivel de gris.

Esta variación de la magnitud del nivel de gris podemos asociarla entonces en función de la distancia de los píxeles en el plano, para así obtener los valores de gris de los píxeles desconocidos.

De esta forma, para el ejemplo que nos ocupa, la línea de tendencia quedaría representada como sigue y el nivel de gris desconocido del píxel 'D' se estimaría asociándolo al punto que le corresponde en esa línea.

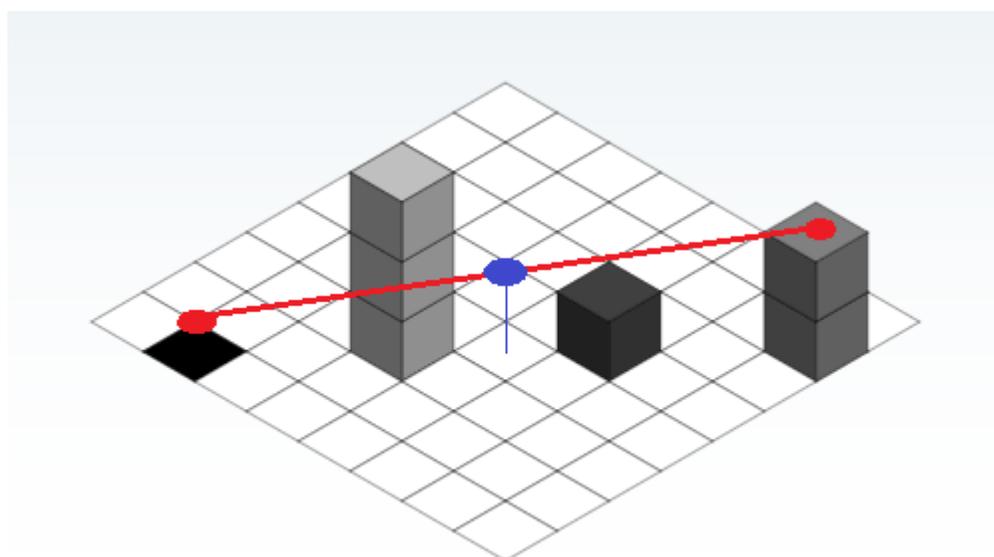


Figura 30. Línea de tendencia para la recta de píxeles con niveles de gris que no siguen un gradiente.

Esta sería la manera de encontrar una función continua que mejor se aproxima a los datos discretos, obteniendo un mejor ajuste según el criterio del mínimo error cuadrático. Veamos en el próximo punto cómo calcular esta línea de tendencia.

### 6.1.2 Cálculo de la línea de tendencia mediante Mínimos Cuadrados

Conociendo un valor arbitrario del eje de las abscisas 'X' calcularemos el valor del eje de ordenadas 'Y', que en nuestro caso corresponderá con el nivel de gris del píxel alojado en 'X'. Para ello utilizaremos la expresión:

$$y = m x + b$$

Para el cálculo de las constantes 'b' (ordenada en el origen) y 'm' (pendiente) utilizaremos las expresiones siguientes.

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x}$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x}$$

El valor de 'N' corresponde con el número de parejas de valores discretos (x,y) conocidos.

Veamos cómo proceder prácticamente a su cálculo, volviendo al ejemplo introducido en el punto anterior.

Para un conjunto de puntos conocidos:

x	y
0	0
0	1
1	1
2	2
2	3
3	3
3	4

Tabla 18. Datos discretos conocidos.

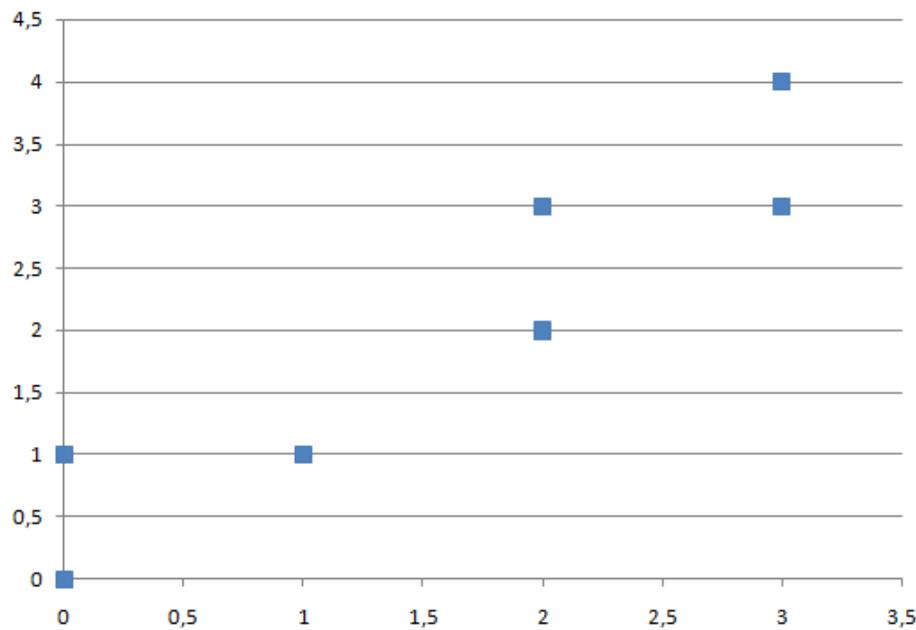


Figura 31. Representación en un gráfico de dispersión de los puntos conocidos.

Procedemos a formalizar los cálculos intermedios necesarios para obtener el valor de 'b' y 'm':

x	y	x <sup>2</sup>	xy
0	0	0	0
0	1	0	0
1	1	1	1
2	2	4	4
2	3	4	6
3	3	9	9
3	4	9	12
$\sum x = 11$	$\sum y = 14$	$\sum x^2 = 27$	$\sum xy = 32$

Tabla 19. Cálculos representativos para 'b' y 'm'.

De este modo, y teniendo en cuenta que tenemos siete puntos (x,y) conocidos y por lo tanto N=7, calculamos las constantes 'b' y 'm':

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x} = \frac{14 * 27 - 11 * 32}{7 * 27 - 11 * 11} = \frac{26}{68} = 0,3824$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x} = \frac{7 * 32 - 11 * 14}{7 * 27 - 11 * 11} = \frac{70}{68} = 1,0294$$

Finalmente podemos construir la función cerrada que nos proporcionará el valor de 'y' asociado a la línea de tendencia para un valor 'x' dado.

$$y = m x + b = 1,0294x + 0,3824$$

Ahora para cualquier valor de 'x' podremos calcular el valor de 'y' según la línea de tendencia de los valores discretos conocidos, por ejemplo:

$$\text{Para } x = 1,5 \rightarrow y = 1,0294 * 1,5 + 0,3824 = 1,9265$$

$$\text{Para } x = 2,6 \rightarrow y = 1,0294 * 2,6 + 0,3824 = 3,0588$$

Así, para cualquier valor de 'x' podemos calcular el valor de 'y', formando la línea de tendencia:

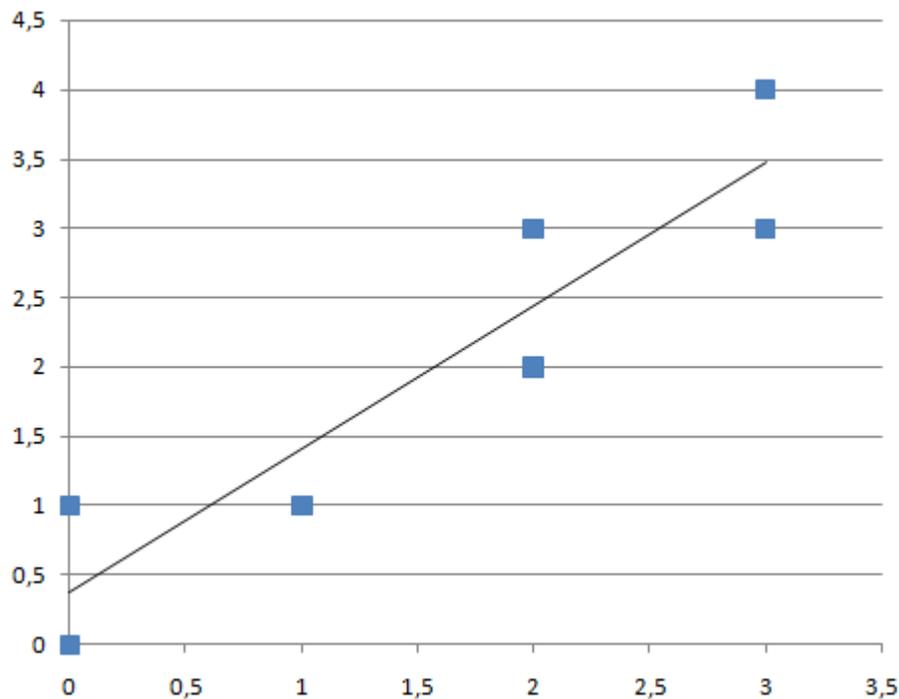


Figura 32. Línea de tendencia calculada por Mínimos Cuadrados para el conjunto de puntos discretos en el gráfico de dispersión.

## 6.2. Aplicación práctica del escalado mediante la Interpolación

El proyecto contempla actualmente el escalado x2 de la imagen, siendo posible su adaptación para realizar otros tipos de escalado.

Tras revisar diferentes lenguajes de programación hemos seleccionado **Java**, como **plugin de ImageJ**, para implementar la aplicación que permite realizar las interpolaciones y también la comparación de los resultados entre diferentes métodos de interpolación, como veremos más adelante.

### 6.2.1. Preparación de la imagen original

Como primer paso, antes de proceder con las técnicas de interpolación, el sistema determina qué es lo que ha solicitado el usuario, mediante la aplicación gráfica del plugin.

- ✓ Tipo de interpolación.
- ✓ Interpolar la imagen en color o bien interpolarla en escala de grises.
- ✓ Mostrar o no al usuario las imágenes temporales generadas durante el proceso de interpolación.
- ✓ Configuración del cálculo la interpolación.
- ✓ Calibraciones específicas.

En función de algunas opciones seleccionadas por el usuario, se procederá con la preparación de la imagen original.

Esta preparación corresponde con la conversión de la imagen a escala de grises de 8-bit si el usuario desea interpolar una imagen en color como blanco y negro, o bien realizar una extracción de las tres componentes de color RGB de una imagen si el usuario desea interpolarla en color.

### 6.2.2. Espaciado de los píxeles conocidos en un mapa x2

Los píxeles de la imagen original a escalar son píxeles conocidos y forman una imagen concreta. El tamaño de esa imagen se corresponde al producto de un conjunto de píxeles horizontales por un conjunto de píxeles verticales (X,Y).

El sistema presentado genera el escalado de esa imagen al doble de tamaño de forma que, tal y como vimos en la introducción de este trabajo, en esa nueva imagen resultante el 75% de los píxeles son totalmente desconocidos.

El método de interpolación calculará esos píxeles desconocidos formando una imagen el doble de tamaño a la original, pero antes de proceder con la interpolación el programa realiza un espaciado de los píxeles de la imagen original en un nuevo mapa con el doble de tamaño, quedando cada píxel conocido ahora envuelto entre píxeles desconocidos. Podemos denominar esta operación como una explosión de los píxeles.

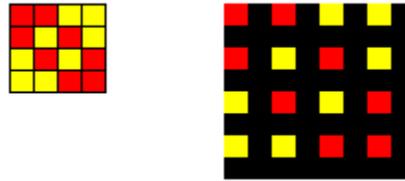


Figura 33. Explosión de los píxeles originales de la imagen de 4x4 píxeles, en un nuevo mapa de 8x8 píxeles. Obsérvese que el 75% de los píxeles de este nuevo mapa son píxeles desconocidos en color negro.

### 6.2.3. Estimación mediante mínimos cuadrados de los píxeles ubicados en las intersecciones (p)

El sistema de interpolación se encargará de calcular el valor de estos píxeles desconocidos mediante el uso de líneas de tendencia, calculadas por mínimos cuadrados de los valores de los píxeles conocidos. Nótese que todas las columnas y filas pares del nuevo mapa están formadas por píxeles desconocidos. Adicionalmente de todas las columnas y filas impares sólo conocemos el 50% de los píxeles.

Por otra parte, puede observarse que la intersección entre las columnas y filas pares forman encrucijadas donde no hay ningún píxel conocido en ninguna de sus cuatro coordenadas básicas (arriba, abajo, izquierda y derecha). Denominaremos estos píxeles como píxeles de intersección 'p'.

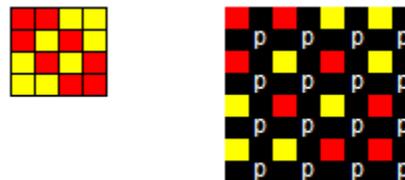


Figura 34. Píxeles 'p' en las intersecciones de píxeles desconocidos.

Si bien estos últimos píxeles desconocidos no mantienen píxeles conocidos en sus posiciones verticales y horizontales sí que existen píxeles conocidos en las diagonales de los mismos.

La interpolación mediante el uso de líneas de tendencia por mínimos cuadrados se iniciará precisamente con el cálculo de estos píxeles de intersección 'p' y la estimación que tendrán según las dos diagonales de cuatro píxeles conocidos que atraviesan cada píxel 'p'.



Figura 35. Píxel 'p' de muestra interpolado mediante el uso de dos diagonales de tendencia sobre ese punto.

Ejemplo, donde el píxel de intersección 'p' desconocido marcado en el centro de una intersección, es calculado mediante la estimación en ese punto 'p' de la línea de tendencia de la diagonal descendente, promediado con la estimación en ese mismo punto 'p' de la línea de tendencia de la diagonal ascendente de los píxeles conocidos en esas diagonales.

El plugin en ImageJ permite calcular estos píxeles de intersección mediante dos sistemas diferentes que hemos denominado 'Sistema 1' y 'Sistema 2'.

### Sistema 1 – Cálculo mediante la línea de tendencia de dos diagonales.

Tal y como hemos visto, tras espaciar los píxeles conocidos de una imagen, y dejando un conjunto de huecos de píxeles desconocidos, podemos pronosticar el valor de estos mediante el uso de líneas de tendencia, calculadas mediante Mínimos Cuadrados.

Por ejemplo, dada la siguiente imagen en escala de gris, formada por 4x4 píxeles.

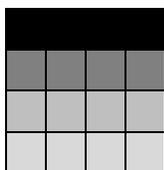


Figura 36. Imagen de 4x4 píxeles.

Separamos los píxeles y obtenemos una imagen con el doble de superficie, pero con  $\frac{3}{4}$  partes de sus píxeles desconocidos:

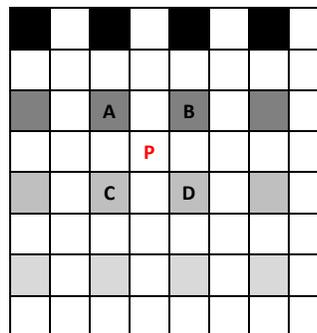


Figura 37. Nueva imagen de 8x8 píxeles con los píxeles espaciados.

En la nueva imagen, ahora con los píxeles conocidos de la imagen original espaciados, encontraremos diferentes intersecciones de píxeles desconocidos, como por ejemplo el píxel de intersección 'p' que hemos marcado en la figura anterior.

Nuestro algoritmo utilizará la base teórica de los Mínimos Cuadrados para pronosticar nuevos píxeles, como por ejemplo el píxel de intersección 'p' que ha sido generado tras la separación de sus cuatro píxeles colindantes con un nivel de gris conocido (marcados como A, B, C y D en la figura anterior).

El píxel de intersección 'p' y todos sus homónimos, creados de una separación cuádruple de píxeles serán calculados de la misma forma, veamos cómo. Identifiquemos los píxeles que utilizaremos para calcular el píxel 'p':

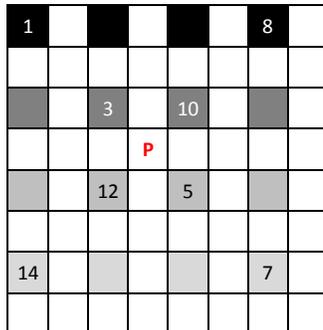


Figura 38. Píxel de intersección 'p' a calcular, mediante las dos diagonales que lo cruzan. A la derecha ambas diagonales.

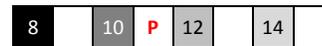
Para pronosticar el valor del píxel de intersección 'p' tendremos en cuenta la estimación del nivel de gris en ese punto desde el punto de vista de las dos diagonales de 45° que pasan sobre él y haciendo uso de los píxeles con nivel de gris conocidos.

En el caso de nuestro ejemplo:

- ✓ **Diagonal 1:** Píxeles 1, 3, 5, 7.



- ✓ **Diagonal 2:** Píxeles 8, 10, 12, 14.



La posición del píxel de intersección 'p' dentro de las dos diagonales corresponde con el cuarto píxel, de tal forma que con esa información podemos estimar qué nivel de gris tendrá ese píxel sobre cada una de las dos líneas de tendencia.

Una vez obtenido el nivel de gris estimado para el píxel de intersección 'p', según cada una de las dos diagonales, practicaremos un promedio de ambos valores y estableceremos el valor del píxel 'p'.

Es decir, la dimensión del eje de las abscisas 'X' para el cálculo de los mínimos cuadrados será precisamente la ubicación espacial del píxel en el plano. En nuestro caso las posiciones primeras, terceras, quintas y séptimas de cada una de las dos diagonales son píxeles conocidos fruto del espaciado de la imagen original, pero los píxeles intermedios segundos, cuartos, sextos y octavos son desconocidos. Como hemos visto, el cuarto píxel en cada una de las dos diagonales corresponde al píxel de intersección 'p' que deseamos calcular.

La dimensión de las ordenadas 'Y' para el cálculo de los mínimos cuadrados será el valor de gris de los píxeles que conocemos.

De esta forma podemos representar isométricamente, a modo de ejemplo ilustrativo, una de las dos diagonales, según lo descrito anteriormente.

En la imagen hemos representado también la línea de tendencia de esa diagonal y la ubicación del píxel 'p' a calcular como incógnita a resolver.

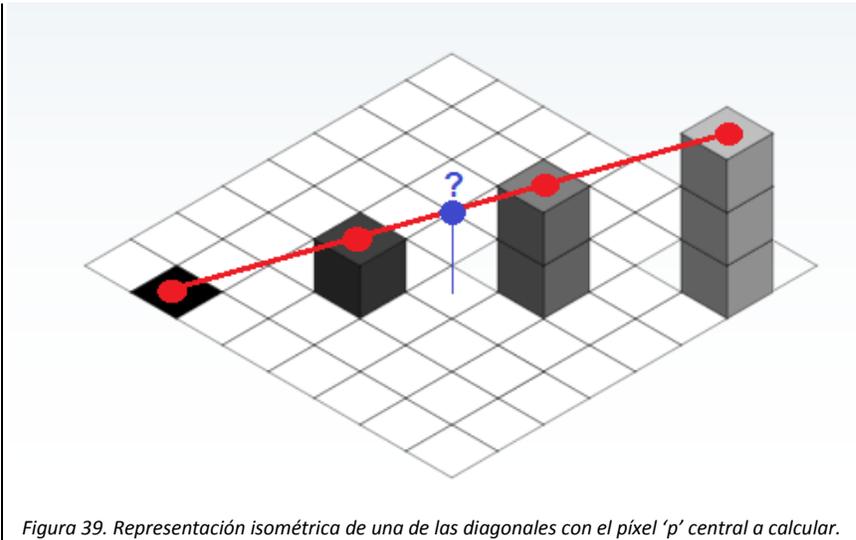


Figura 39. Representación isométrica de una de las diagonales con el píxel 'p' central a calcular.

El cálculo para este caso de ejemplo sería el siguiente.

Ya que en nuestro caso ambas diagonales mantienen los mismos niveles de gris en sus píxeles con nivel de gris conocido, nos bastará realizar el cálculo para una de las diagonales.

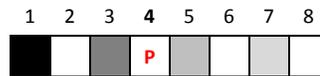


Figura 40. Una de las dos diagonales en la que el píxel 'p' se encuentra en la cuarta posición.

Los niveles de gris de los píxeles conocidos, o las N=4 muestras discretas (1, 3, 5, 7), son los siguientes (0, 128, 191, 217), siendo el nivel de gris 0 el color negro. De esta forma podemos ubicar en el espacio estos valores.

Número de píxel	x	y	x <sup>2</sup>	xy
Primero	1	0	1	0
Tercero	3	128	9	384
Quinto	5	191	25	955
Sexto	7	217	49	1519
	∑ x = 16	∑ y = 536	∑ x <sup>2</sup> = 84	∑ xy = 2858

Tabla 20. Cálculos representativos para 'b' y 'm'.

Con esta información procedemos a calcular las constantes 'b' (ordenada en el origen) y 'm' (pendiente):

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x} = \frac{536 * 84 - 16 * 2858}{4 * 84 - 16 * 16} = -8,8$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x} = \frac{4 * 2858 - 16 * 536}{4 * 84 - 16 * 16} = 35,7$$

De este modo podemos calcular el nivel de gris de un píxel ubicado en la posición  $x=4$ , que sería el píxel 'p':

$$y_{Diagonal\ 1} = m x + b = 35,7 * 4 - 8,8 = 134$$

En nuestro caso de ejemplo, como se trata de una imagen en la que ambas diagonales del píxel 'p' mantienen los mismos niveles de gris, el nivel estimado en el píxel 'p' es el mismo para la segunda diagonal:

$$y_{Diagonal\ 2} = m x + b = 35,7 * 4 - 8,8 = 134$$

El valor definitivo de nivel de gris del píxel 'p' sería el promedio de la estimación del nivel de gris en la posición del píxel de intersección 'p' desde el punto de vista de cada una de las dos diagonales que lo atraviesan, de tal forma que el valor definitivo del nivel de gris del píxel 'p' sería:

$$Nivel\ de\ gris\ del\ píxel\ de\ intersección\ 'p' = \frac{y_{Diagonal\ 1} + y_{Diagonal\ 2}}{2} = \frac{268}{2} = 134$$

A tener en cuenta que el plugin en ImageJ que hemos desarrollado tiene en cuenta el método de cálculo descrito anteriormente, que en el plugin puede seleccionarse como '**Sistema 1**'.

## Sistema 2 - Cálculo mediante la línea de tendencia de dos diagonales y promedio de vecinos.

Por otra parte hemos elaborado una mejora al respecto en la que los píxeles de intersección (píxeles 'p') se calcularán siguiendo la misma estimación del 'Sistema 1', pero incorporando el promedio de los cuatro píxeles más próximos a cada píxel 'p' según la siguiente ecuación:

$$Nivel\ de\ gris\ del\ píxel\ de\ intersección\ 'p' = \frac{y_{Diagonal\ 1} + y_{Diagonal\ 2} + P_1}{3}$$

Donde:

$y_{Diagonal1}$  = Estimación del píxel 'p' mediante mínimos cuadrados para la diagonal de píxeles discretos (1, 3, 5, 7).

$y_{Diagonal2}$  = Estimación del píxel 'p' mediante mínimos cuadrados para la diagonal de píxeles discretos (8, 10, 12, 14).

$P_1$  = Promedio de los píxeles colindantes al píxel 'p' (3, 10, 12, 5).

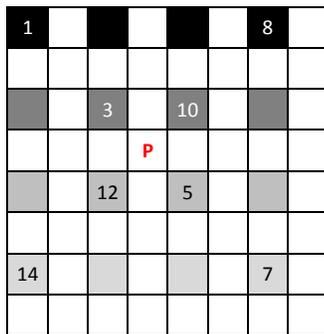


Figura 41. Ejemplo para nuestra matriz de cálculo.

En nuestra imagen de ejemplo, recordemos que el nivel de gris de los píxeles es de más oscuro a más claro (0, 128, 191, 217), por lo que el nivel de gris de los píxeles 3 y 10 es **128** y el nivel de gris de los píxeles 12 y 5 es **191**. De ese modo podemos calcular el promedio entre esos cuatro píxeles:

$$P_1 = \frac{128 + 128 + 191 + 191}{4} = 159,5$$

Con esta información podemos calcular el nivel de gris definitivo para el píxel de intersección 'p':

$$p = \frac{134 + 134 + 159,5}{3} = 142,5 \sim 142$$

De este modo conseguimos un mejor ajuste del nivel de gris del píxel de intersección ubicado en la posición 'p' de la matriz expandida de píxeles, teniendo en cuenta la tendencia de los píxeles discretos que lo atraviesan en sus dos diagonales de cuatro píxeles y los cuatro píxeles discretos que lo rodean de forma inmediata.

Este método de calcular el píxel 'p' podremos seleccionarlo desde el plugin de ImageJ mediante la opción de 'Sistema 2'.

#### 6.2.4. Estimación de los píxeles cardinales (n,s,e,o) con respecto a los píxeles de intersección (p)

El sistema de interpolación permite, además de estimar los píxeles de intersección 'p' del mapa escalado, calcular los píxeles cardinales que se encuentran arriba, abajo, izquierda y derecha de esos píxeles de intersección. Llamaremos estos cuatro píxeles de nivel de gris desconocido, píxeles cardinales (n, s, e, o).

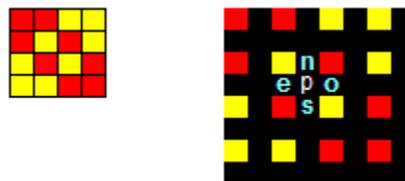


Figura 42. Píxeles 'n', 's', 'e' y 'o' pertenecientes a las coordenadas de un píxel de intersección 'p'.

Tal y como puede apreciarse, un píxel 'n' en relación a un píxel 'p' será también un píxel 's' de otro píxel 'p' ubicado sobre el primero. Así mismo el píxel 'e' de un píxel 'p' será 'o' de otro píxel 'p' alojado a la izquierda del primero.

El plugin en ImageJ permite estimar estos cuatro píxeles mediante cuatro sistemas diferentes que hemos elaborado y que hemos denominado 'Sistema A', 'Sistema B', 'Sistema C' y 'Sistema D'.

### Sistema A - Cálculo de los píxeles cardinales mediante el promedio de los dos vecinos más próximos.

Sin duda este primer sistema es el más básico y permite calcular los cuatro píxeles cardinales, asociados a un píxel de intersección 'p', haciendo uso del promedio del valor de nivel de gris de los dos píxeles que están alojados en la vecindad más próxima a cada incógnita.

De ese modo, el cálculo del nivel de gris de los píxeles cardinales a un píxel de intersección 'p' se define como:

- ✓ Píxel 'n', alojado arriba del píxel 'p':

$$n_p = \frac{A + B}{2}$$

- ✓ Píxel 's', alojado abajo del píxel 'p':

$$s_p = \frac{C + D}{2}$$

- ✓ Píxel 'e', alojado a la izquierda del píxel 'p':

$$e_p = \frac{A + C}{2}$$

- ✓ Píxel 'o', alojado a la derecha del píxel 'p':

$$o_p = \frac{B + D}{2}$$

Donde los píxeles A, B, C y D quedan definidos en las siguientes posiciones, en relación al píxel de intersección 'p':

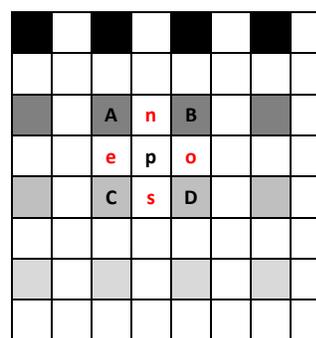


Figura 43. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.

A efectos prácticos, el cálculo de los píxeles cardinales para el ejemplo en cuestión sería el siguiente, teniendo en cuenta que el nivel de gris de los píxeles A y B es 128 y el de los píxeles C y D es 191.

A = 128	$n_p = \frac{128 + 128}{2} = 128$	B = 128
$e_p = \frac{128 + 191}{2} \sim 159$	'p'	$o_p = \frac{191 + 128}{2} \sim 159$
C = 191	$s_p = \frac{191 + 191}{2} = 191$	D = 191

Tabla 21. Representación de los nueve píxeles y cálculo de los píxeles cardinales 'n', 's', 'e' y 'o'.

### Sistema B - Cálculo de los píxeles cardinales mediante las líneas de tendencia en sus coordenadas

Mediante este sistema, es posible calcular los píxeles cardinales 'n', 's', 'e' y 'o' mediante el uso de líneas de tendencia, calculadas por mínimos cuadrados.

Se establecen en el cálculo cuatro líneas, dos horizontales y dos verticales, donde las líneas horizontales suceden por los píxeles desconocidos 'n' y 's', mientras que las líneas verticales suceden por los píxeles desconocidos 'e' y 'o'.

Para cada una de esas cuatro líneas se establece qué valor de gris tendrá un píxel ubicado sobre la misma teniendo en cuenta los valores discretos conocidos sobre la línea. De la misma manera que estimamos el valor del píxel 'p' en su ubicación respecto a sus dos diagonales, ahora calcularemos la estimación de los píxeles 'n', 's', 'e' y 'o' según sus verticales y horizontales.

De ese modo, para calcular los píxeles cardinales 'n', 's', 'e' y 'o' utilizaremos el valor del nivel de gris de los doce píxeles discretos que los atraviesan con sus líneas horizontales y verticales.

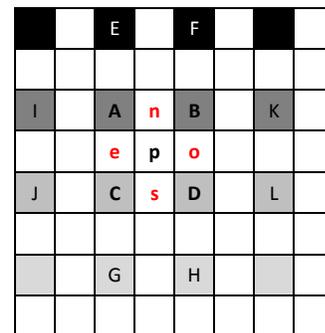


Figura 44. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.

En el caso de nuestro ejemplo descubrimos las siguientes líneas:

- ✓ **Horizontal 1:** Píxeles I, A, B, K para el cálculo del píxel cardinal 'n':



- ✓ **Horizontal 2:** Píxeles J, C, D, L para el cálculo del píxel cardinal 's':



- ✓ **Vertical 1:** Píxeles E, A, C, G para el cálculo del píxel cardinal 'e':



- ✓ **Vertical 2:** Píxeles F, B, D, H para el cálculo del píxel cardinal 'o':



Figura 44 bis. Representación de las dos horizontales y las dos diagonales de píxeles.

Podemos observar que en las posiciones primera, tercera, quinta y séptima disponemos de valores de nivel de gris discretos asociados a los doce píxeles conocidos y que la cuarta posición coincide con el valor de un píxel cardinal 'n', 's', 'e' u 'o'.

De ese modo podemos estimar, en cada una de las cuatro líneas, qué valor tendrá el píxel ubicado en la cuarta posición, mediante el uso de líneas de tendencia, calculadas con mínimos cuadrados.

**Horizontal 1 - Cálculo del píxel cardinal 'n'.**

Número de píxel	x	y	x <sup>2</sup>	xy
Primero	1	128	1	128
Tercero	3	128	9	384
Quinto	5	128	25	640
Sexto	7	128	49	896
	Σ x = 16	Σ y = 512	Σ x <sup>2</sup> = 84	Σ xy = 2048

Tabla 22. Cálculos representativos para 'b' y 'm'.

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x} = \frac{10240}{80} = 128$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x} = \frac{0}{80} = 0$$

En la cuarta posición se estima el valor de nivel de gris del píxel 'n':

$$n = m x + b = 0 * 4 + 128 = 128$$

**Horizontal 2 - Cálculo del píxel cardinal 's'.**

Número de píxel	x	y	x <sup>2</sup>	xy
Primero	1	191	1	191
Tercero	3	191	9	573
Quinto	5	191	25	955
Sexto	7	191	49	1337
	Σ x = 16	Σ y = 764	Σ x <sup>2</sup> = 84	Σ xy = 3056

Tabla 23. Cálculos representativos para 'b' y 'm'.

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x} = \frac{15280}{80} = 191$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x} = \frac{0}{80} = 0$$

En la cuarta posición se estima el valor de nivel de gris del píxel 's':

$$s = m x + b = 0 * 4 + 191 = 191$$

**Vertical 1 y Vertical 2 - Cálculo de los píxeles cardinal 'e' y 'o'.**

Número de píxel	x	y	x <sup>2</sup>	xy
Primero	1	0	1	0
Tercero	3	128	9	384
Quinto	5	191	25	955
Sexto	7	217	49	1519
	Σ x = 16	Σ y = 536	Σ x <sup>2</sup> = 84	Σ xy = 2858

Tabla 24. Cálculos representativos para 'b' y 'm'.

$$b = \frac{\sum y \sum x^2 - \sum x \sum xy}{N \sum x^2 - \sum x \sum x} = \frac{-704}{80} = -8,8$$

$$m = \frac{N \sum xy - \sum x \sum y}{N \sum x^2 - \sum x \sum x} = \frac{2856}{80} = 35,7$$

En la cuarta posición se estima el valor de nivel de gris de los píxeles 'e' y 'o':

$$e = o = m x + b = 35,7 * 4 - 8,8 = 134$$

Según los cálculos de línea de tendencia se estima el nivel de gris de los píxeles cardinales (n, s, e, o):

$$n = 128 \quad s = 191 \quad e = 134 \quad o = 134$$

## Sistema C - Cálculo de los píxeles cardinales mediante un sistema de pesos

La particularidad de este sistema radica en el cálculo temporal de píxeles, cada vez que nos posicionamos sobre un píxel de intersección 'p', para aprovechar una única iteración sobre la imagen. La interpolación se realiza en cuatro pasos.

- ✓ **Primero**, calcularemos un conjunto de píxeles de intersección 'p' ubicados en la anterior y siguiente intersección de orden cardinal de un píxel 'p' dado. Esos píxeles de intersección 'p' temporales los denominaremos píxeles 'w', 'x', 'y' y 'z' y se calcularán de la siguiente manera:

$w$  = Promedio del pronóstico por mínimos cuadrados de las líneas de tendencia (F,A,J) y (E,B,L).  
 $x$  = Promedio del pronóstico por mínimos cuadrados de las líneas de tendencia (E,B,L) y (K,D,G).  
 $y$  = Promedio del pronóstico por mínimos cuadrados de las líneas de tendencia (K,D,G) y (H,C,I).  
 $z$  = Promedio del pronóstico por mínimos cuadrados de las líneas de tendencia (H,C,I) y (F,A,J).

Una vez calculados estos valores temporales, establecemos qué peso tendrán (en %) estos píxeles en el resultado final. Denominaremos este peso como "%P±1".

- ✓ **Segundo**. Una vez calculados los píxeles 'p' en intersección cardinal  $\pm 1$ , calcularemos otro conjunto de valores temporales que denominaremos píxeles  $n'$ ,  $e'$ ,  $s'$  y  $o'$ .

Además, estableceremos qué peso tendrán esos píxeles (en %) en el resultado final. Denominaremos este peso como "%HóV".

M		E		F		N	
			w				
I		A	n	B		K	
		z	e	p	o	x	
J		C	s	D		L	
			y				
Q		G		H		R	

Figura 45. Píxeles cardinales 'n', 's', 'e' y 'o' asociados a un píxel de intersección 'p'.

$n'$  = Pronóstico por mínimos cuadrados de la línea de tendencia (I,A,B,K).  
 $e'$  = Pronóstico por mínimos cuadrados de la línea de tendencia (E,A,C,G).  
 $s'$  = Pronóstico por mínimos cuadrados de la línea de tendencia (J,C,D,L).  
 $o'$  = Pronóstico por mínimos cuadrados de la línea de tendencia (F,B,D,H).

- ✓ **Tercero**. Calcularemos el valor del píxel de intersección 'p' en el que estamos posicionados de la manera habitual, según el método del Sistema 1 que hemos visto en capítulos anteriores. A efectos prácticos, el nivel de gris del píxel 'p' será:

$p$  = Promedio del pronóstico por mínimos cuadrados de las líneas de tendencia (M,A,D,R) y (N,B,C,Q).

Además, estableceremos qué peso tendrá (en %) este tipo de valor en el resultado final. Denominaremos este peso como “%P”.

- ✓ **Cuarto.** En función del peso que establezcamos a cada uno de los tres cálculos, calcularemos el valor del nivel de gris de los píxeles (n,s,e,o) mediante el siguiente sistema de pesos.

$$\begin{aligned}
 n &= (w * \%P_{\pm 1}) + (n' * \%HóV) + (p * \%P) \\
 e &= (x * \%P_{\pm 1}) + (e' * \%HóV) + (p * \%P) \\
 s &= (y * \%P_{\pm 1}) + (s' * \%HóV) + (p * \%P) \\
 o &= (z * \%P_{\pm 1}) + (o' * \%HóV) + (p * \%P)
 \end{aligned}$$

A modo de ejemplo, configurando los siguientes pesos, que deben sumar la unidad,

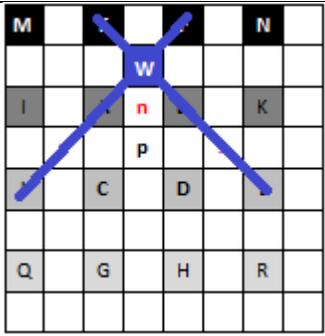
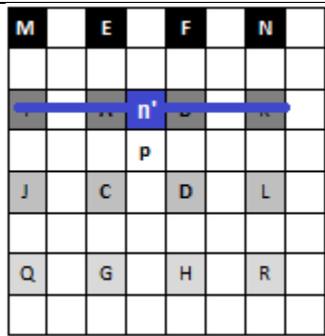
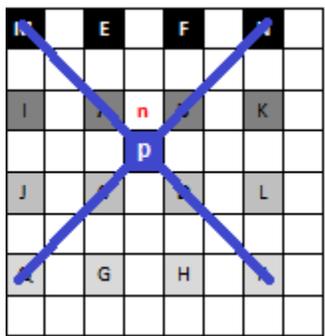
- ✓ %P±1 = 0,3
- ✓ %HóV = 0,5
- ✓ %P = 0,2

Estamos indicando que:

El píxel ‘n’	El píxel ‘s’	El píxel ‘e’	El píxel ‘o’
Tendrá un <b>30%</b> de la tonalidad calculada en la posición del píxel de intersección ‘w’, calculado mediante el promedio de la estimación de las dos diagonales de tendencia que lo atraviesan.	Tendrá un <b>30%</b> de la tonalidad calculada en la posición del píxel de intersección ‘y’, calculado mediante el promedio de la estimación de las dos diagonales de tendencia que lo atraviesan.	Tendrá un <b>30%</b> de la tonalidad calculada en la posición del píxel de intersección ‘z’, calculado mediante el promedio de la estimación de las dos diagonales de tendencia que lo atraviesan.	Tendrá un <b>30%</b> de la tonalidad calculada en la posición del píxel de intersección ‘x’, calculado mediante el promedio de la estimación de las dos diagonales de tendencia que lo atraviesan.
Tendrá un <b>50%</b> de la tonalidad calculada en esa misma posición mediante la línea de tendencia <b>horizontal</b> que lo atraviesa.		Tendrá un <b>50%</b> de la tonalidad calculada en esa misma posición mediante la línea de tendencia <b>vertical</b> que lo atraviesa.	
Tendrá un <b>20%</b> de la tonalidad calculada en la posición del píxel de intersección ‘p’, calculado mediante el promedio de la estimación de las dos diagonales de tendencia que lo atraviesan.			

Tabla 25. Tabla de representación de pesos según el ejemplo mostrado.

Para el caso del cálculo del píxel cardinal 'n', calcularemos:

<p>1 - El píxel de intersección 'w', mediante el promedio de la estimación en ese punto, de la línea de tendencia de las dos diagonales de tres píxeles discretos que lo atraviesan.</p> <p style="text-align: center;"><b>w = 59</b></p> <p>Estableceremos un peso para este nivel de gris que denominaremos <b>%P±1</b>, por ejemplo el <b>30%</b>.</p>	 <p style="text-align: center;"><i>Figura 46. Píxel de intersección 'w'.</i></p>
<p>2 - Posteriormente calcularemos qué valor se estimaría en la posición 'n' mediante la línea de tendencia de píxeles discretos que lo atraviesa, denominando este valor temporal n'.</p> <p style="text-align: center;"><b>n' = 128</b></p> <p>Estableceremos un peso para este nivel de gris que denominaremos <b>%HóV</b>, por ejemplo el <b>50%</b>.</p>	 <p style="text-align: center;"><i>Figura 47. Píxel n' temporal.</i></p>
<p>3 - Finalmente tendremos el cálculo del nivel de gris del píxel de intersección 'p', calculado mediante el promedio de la estimación en ese punto, de la línea de tendencia de las dos diagonales de cuatro píxeles discretos que lo atraviesan (se trata del "Sistema 1" descrito en capítulos anteriores).</p> <p style="text-align: center;"><b>p = 142.</b></p> <p>Estableceremos un peso para este nivel de gris que denominaremos <b>%P</b>, por ejemplo el <b>20%</b> restante.</p>	 <p style="text-align: center;"><i>Figura 48. Píxel de intersección 'p'.</i></p>
<p>4- Finalmente estableceremos el nivel de gris del píxel cardinal 'n' en función de los tres valores calculados anteriormente y los pesos que les hayamos otorgado en el resultado final.</p> <p style="text-align: center;"><b>n = 0,3 * 59 + 0,5 * 128 + 0,2 * 142 = 110</b></p>	

A efectos prácticos, con este sistema podemos configurar que tipo de cálculo le otorgamos más valor para definir el nivel de gris definitivo en un píxel cardinal, de forma que si ofrecemos un mayor peso a:

- ✓ %P±1 → Estamos indicando que confiamos más en el valor calculado del píxel de intersección temporal.
- ✓ %HóV → Estamos indicando que confiamos más en el valor calculado mediante la línea de tendencia que atraviesa el píxel cardinal a calcular.
- ✓ %P → Estamos indicando que confiamos más en el valor calculado del píxel de intersección 'p' al cual estamos referenciados para calcular el píxel cardinal.

**Sistema D - Cálculo de los píxeles cardinales en doble iteración, mediante un sistema de pesos entre la vecindad de los píxeles discretos y los píxeles estimados mediante líneas de tendencia.**

Una de las principales características de este algoritmo es que realiza dos iteraciones sobre la imagen. Una para calcular los píxeles de intersección 'p' y otra para calcular los píxeles cardinales 'n', 's', 'e' y 'o' en función de los píxeles de intersección 'p' y los píxeles discretos conocidos de la imagen original.

El cálculo de los píxeles de intersección 'p' se calculan mediante el 'Sistema 1' o 'Sistema 2', seleccionado en la configuración del plugin. El procedimiento es el mismo que el descrito en capítulos anteriores.

M	E	F	N
I	A	B	K
J	C	D	L
Q	G	H	R

Figura 49. Imagen original 4x4.

La imagen original es trasladada a una nueva imagen, con el doble de tamaño y realizando un espaciado de los píxeles, formando un 75% de filas y columnas de píxeles desconocidos.

M		E		F		N	
I		A		B		K	
J		C		D		L	
Q		G		H		R	

Figura 50. Imagen espaciada.

Imagen con el espaciado aplicado y con un tamaño el doble que la original, lista para calcular los píxeles de intersección 'p'.

M		E		F		N	
	p		p		p		p
I		A		B		K	
	p		p		p		p
J		C		D		L	
	p		p		p		p
Q		G		H		R	
	p		p		p		p

Figura 51. Imagen con píxeles de intersección 'p'.

Imagen con los píxeles de intersección 'p' calculados mediante líneas de tendencia por mínimos cuadrados, según 'Sistema 1' o 'Sistema 2'.

Una vez calculados todos los píxeles de intersección 'p' veremos que restarán por calcular los píxeles desconocidos que hay entre dos píxeles discretos, provenientes de la imagen original, que coincidirán con píxeles desconocidos entre dos píxeles de intersección 'p'. Todos esos píxeles son los píxeles cardinales 'n', 's', 'e' y 'o' de los píxeles de intersección 'p'.

M		E		F		N
			p1			
I		A	n	B		K
	p2	o	p	e	p3	
J		C	s	D		L
			p4			
Q		G		H		R

Figura 52. Píxeles cardinales a calcular para un píxel de intersección 'p' dado.

A modo de ejemplo, en la imagen de la izquierda se nos muestra un píxel de intersección 'p' en concreto, calculado en la primera iteración mediante el 'Sistema 2' o el 'Sistema 2'.

Al realizar una primera iteración ya tenemos calculados todos los píxeles de intersección 'p', como por ejemplo 'p1', 'p2', 'p3' o 'p4' entre otros.

Por otra parte se nos indica los píxeles cardinales 'n', 's', 'e' y 'o' desconocidos asociados a ese píxel de intersección 'p'.

Este sistema se plantea cómo calcular los píxeles restantes en una segunda iteración. En definitiva todos los píxeles cardinales 'n', 's', 'e' y 'o' de los píxeles 'p' calculados en la iteración anterior, mediante mínimos cuadrados.

Hemos basado este cálculo en un sistema de pesos, asociado a los cuatro píxeles vecinos que rodean a cada píxel cardinal dado, otorgando una mayor relevancia en el resultado final para un par de píxeles vecinos en detrimento del otro par.

$$\text{Píxel Cardinal} = \%Peso1 \frac{p + p'}{2} + \%Peso2 \frac{A + B}{2}$$

Donde:

- ✓ **Píxel Cardinal:** Nivel de gris del píxel cardinal 'n', 's', 'e' u 'o' según el caso.
- ✓ **%Peso1:** Relación de peso que le asignaremos al par de píxeles de intersección, vecinos al píxel cardinal a calcular.
- ✓ **p+p':** Suma del nivel de gris de los píxeles de intersección 'p', vecinos al píxel cardinal a calcular.
- ✓ **A+B:** Suma del nivel de gris de los píxeles discretos, provenientes de la imagen original y vecinos al píxel cardinal a calcular.
- ✓ **%Peso2:** Relación de peso que le asignaremos al par de píxeles discretos, provenientes de la imagen original, vecinos al píxel cardinal a calcular.

Esto es, en el ejemplo mostrado anteriormente, para el píxel cardinal 'n', sus vecinos son los píxeles 'A', 'B', 'p1' y 'p', siendo los dos primeros píxeles discretos que provienen de la imagen original y los dos últimos píxeles de intersección, calculados en la iteración anterior mediante líneas de tendencia por mínimos cuadrados.

Podemos indicarle al sistema qué porcentaje de nivel de gris le asignaremos finalmente al píxel 'n' en relación a sus dos pares de píxeles vecinos:

- ✓ A los píxeles de intersección 'p' y 'p1' calculados en la iteración anterior le asignaremos el peso '**%Peso1**'
- ✓ A los píxeles discretos 'A', 'B' que provienen de la imagen original le asignaremos el peso '**%Peso2**'.

De ese modo:  $\text{Píxel Cardinal} = \%Peso1 * \text{Promedio nivel de gris píxeles de intersección} + \%Peso2 * \text{Promedio nivel de gris píxeles discretos}$

En nuestro ejemplo, podremos calcular el nivel de gris del píxel cardinal 'n', sabiendo el nivel de gris de los píxeles vecinos a 'n' y, el valor configurado en el interpolador para que se tenga en cuenta un mayor peso con los píxeles discretos que con los píxeles de intersección (70% vs 30%).

El valor de gris conocido de cada uno de los cuatro píxeles vecinos de 'n' es:

$$\begin{aligned} p1 &= 59 \\ p &= 142 \\ A &= 128 \\ B &= 128 \end{aligned}$$

Calcularemos el nivel de gris del píxel cardinal 'n' según:

$$\begin{aligned} n &= \%Peso1 \frac{p + p1}{2} + \%Peso2 \frac{A + B}{2} = \\ &= 0,3 \frac{142 + 59}{2} + 0,7 \frac{128 + 128}{2} = 119,75 \sim 120 \end{aligned}$$

Queda entonces definido el valor del píxel cardinal 'n' con valor con nivel de gris 120.

## 7. Interpolación en color

Uno de los objetivos del proyecto consiste en poder interpolar en color. Esto es, partiendo de una imagen de entrada en color, escalar su tamaño x2 mediante el uso de la interpolación y mostrar el resultado como una imagen, también en color.

Para realizar este cometido, la forma que hemos pensado ha sido la de extraer de la imagen original en color las tres componentes de color, rojo, verde y azul, para posteriormente practicar el escalado mediante interpolación en cada una de esas tres componentes y finalmente unificar las componentes en una nueva imagen en color, ahora el doble de tamaño que la original.

Este procedimiento puede aplicarse a cualquiera de las interpolaciones descritas en el proyecto, tanto las clásicas como la interpolación propia mediante mínimos cuadrados, no obstante prácticamente únicamente hemos implementado en Java la interpolación en color en nuestro propio sistema. Veamos un ejemplo práctico, donde partiendo una imagen original en color.

- ✓ Extraemos las tres componentes de color, rojo, verde y azul de la imagen en tres mapas diferenciadas.
- ✓ Creamos tres mapas con el doble de dimensión que la imagen original, trasladamos y separamos los píxeles de las tres componentes en cada uno de esos nuevos mapas de píxeles.
- ✓ Aplicamos el algoritmo de interpolación por mínimos cuadrados en cada uno de los tres mapas con píxeles separados, obteniendo ahora las tres componentes ya escaladas.
- ✓ Unificamos los tres mapas de color interpolados en una nueva imagen a color con el doble de dimensión que la original.

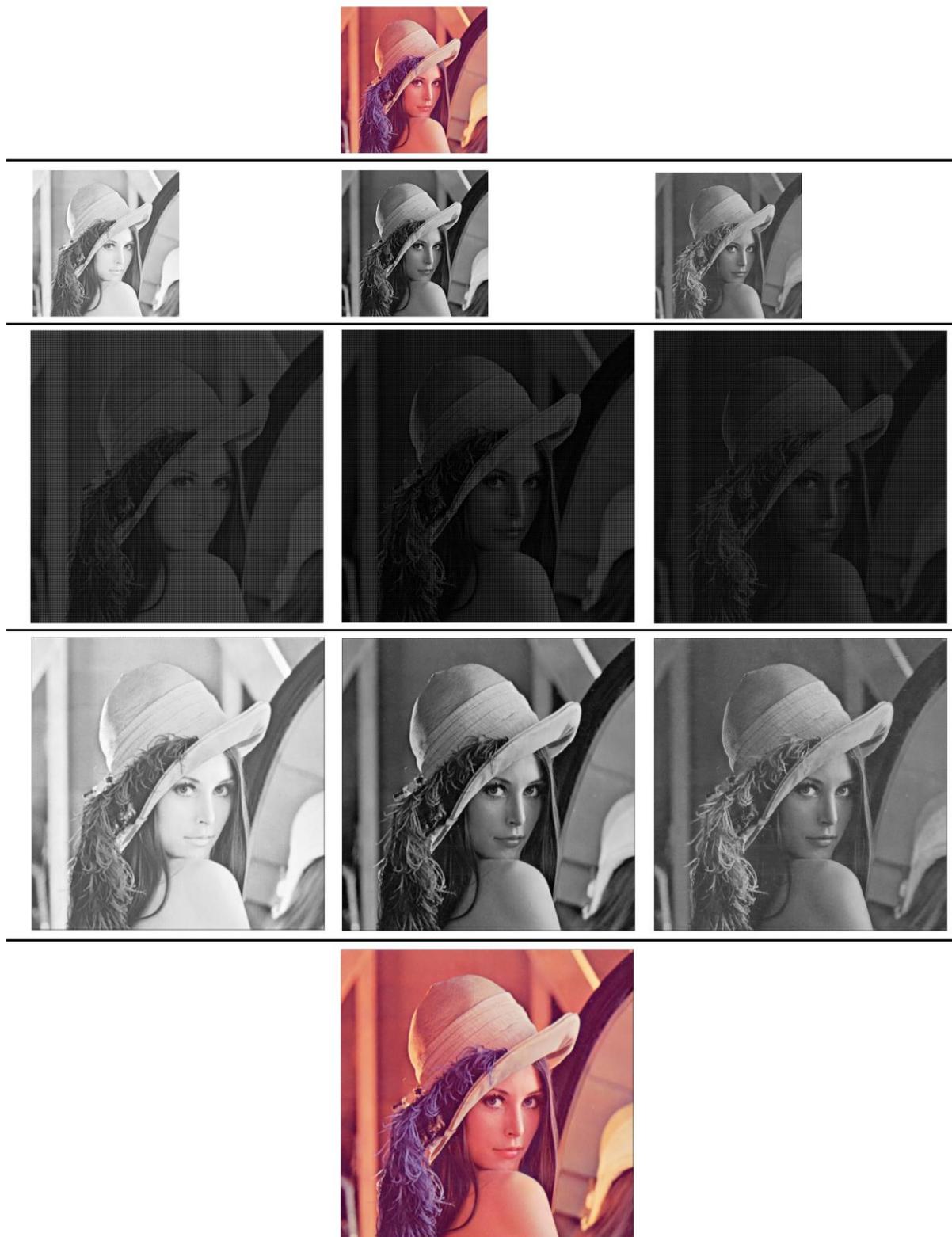


Figura 53. Ejemplo ilustrativo de la interpolación a color propuesta.

## 8. Plugin Interpolador para ImageJ

En este capítulo mostraremos las pantallas del plugin de interpolación que hemos desarrollado y que ofrece la posibilidad de ejecutar todas las opciones que hemos descrito teóricamente en capítulos anteriores.

### 8.1. Características

Las características del plugin son las siguientes:

- ✓ Desarrollado en Java como complemento para ImageJ v1.48 o superior, con unas 1800 líneas de código.
- ✓ Permite el escalado x2 de imágenes mediante los métodos de interpolación clásicos por vecino más próximo, bilineal y bicúbica, en imágenes en escala de grises.
- ✓ Ofrece el escalado x2 de imágenes mediante el método propio por mínimos cuadrados, tanto en imágenes en escala de grises como en color.

### 8.2. Compilación del plugin

Antes de proceder con la ejecución práctica del plugin conviene compilar el código fuente en Java.

Para ello, la herramienta ImageJ ofrece de manera integrada la posibilidad de realizar la compilación del fichero Java. Seguiremos los siguientes pasos:

- ✓ Identificaremos el directorio de instalación de ImageJ. En nuestro caso de ejemplo en la ruta **D:\APPS\ImageJ\**
- ✓ Accederemos al directorio '**plugins**', en nuestro caso ubicado en la ruta **D:\APPS\ImageJ\plugins\**
- ✓ Dentro del directorio '**plugins**' crearemos un directorio para nuestros plugins, por ejemplo en nuestro caso un directorio llamado '**Colome**' que quedará ubicado en la ruta absoluta **D:\APPS\ImageJ\plugins\Colome**
- ✓ Copiaremos dentro del directorio que hemos creado el fichero **COLOME\_Interpolador.java**
- ✓ Ejecutaremos ImageJ, en nuestro caso el ejecutable **D:\APPS\ImageJ\ImageJ.exe**
- ✓ Desplegaremos el menú '**Plugins**' y seleccionaremos la opción '**Compile and Run...**'

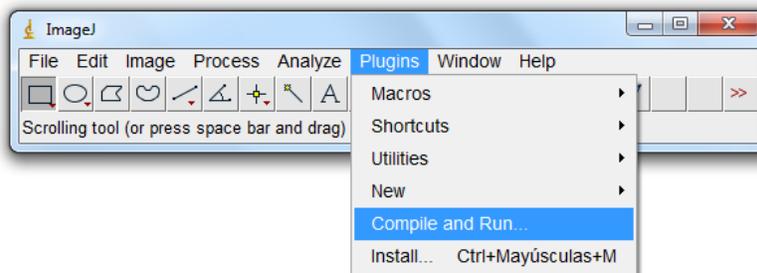


Figura 54. ImageJ y procedimiento de compilación del plugin.

- ✓ Nos aparecerá una ventana desde la que podremos seleccionar ficheros Java. Posicionaremos el foco sobre el directorio donde hemos copiado nuestro plugin y seleccionaremos el fichero **COLOME\_Interpolador.java**
- ✓ ImageJ compilará el programa y nos mostrará una ventana indicando que no hay ninguna imagen abierta. Pulsaremos el botón 'OK'.



Figura 55. Mensaje informativo tras la compilación del plugin.

- ✓ El programa quedará compilado, habiéndose creado en la misma ruta los ficheros de clase "COLOME\_Interpolador\$1.class" y "COLOME\_Interpolador.class".
- ✓ Cerraremos la aplicación ImageJ y la volveremos a ejecutar para que refresque los menús de los plugin y ahora ya podremos encontrar nuestro plugin en el menú 'Plugins', bajo la sección 'Colome' y el nombre de plugin 'COLOME\_Interpolador'.

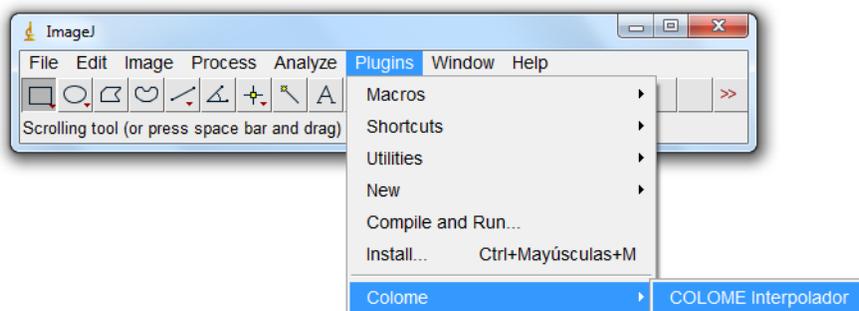


Figura 56. Ubicación del plugin una vez compilado.

### 8.3. Ejecución del plugin

Una vez compilado el plugin podrá ser ejecutado, siempre y cuando tengamos una imagen abierta en ImageJ, desde el menú 'Plugins', sección 'Colome' y opción 'COLOME\_Interpolador'.

Una vez ejecutado el plugin observaremos una pantalla principal denominada "Interpolador de imagen digital" desde donde podremos seleccionar un método de interpolación.

#### Interpolación "Vecino más Próximo".

Si deseamos interpolar la imagen que tenemos abierta desde ImageJ mediante la interpolación clásica por "Vecino más próximo", seleccionaremos la opción "Vecino más próximo" desde el selector de interpolación a aplicar.

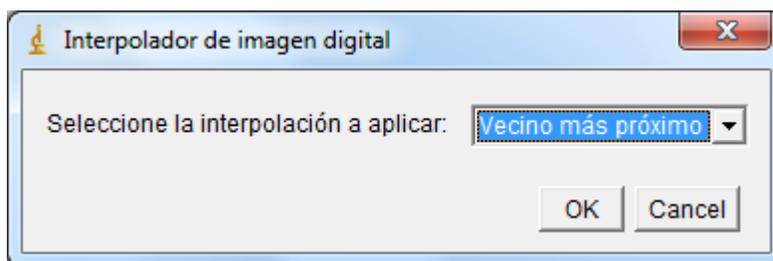


Figura 57. Pantalla que muestra la opción de interpolación mediante vecino más próximo.

Seguidamente se mostrará la imagen ya escalada mediante la interpolación seleccionada en escala de grises y una ventana de log con información relativa al proceso de interpolación.

#### Interpolación "Bilineal".

Si deseamos interpolar la imagen que tenemos abierta desde ImageJ mediante el método "Bilineal" clásico, seleccionaremos la opción "Bilineal" desde el selector de interpolación a aplicar.

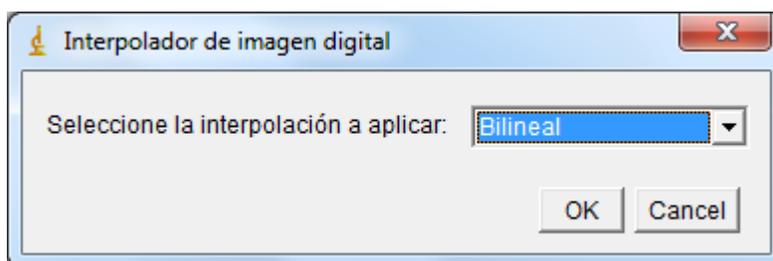


Figura 58. Pantalla que muestra la opción de interpolación bilineal.

Seguidamente se mostrará la imagen ya escalada mediante la interpolación seleccionada en escala de grises y una ventana de log con información relativa al proceso de interpolación.

### Interpolación “Bicúbica”.

Si deseamos interpolar la imagen que tenemos abierta desde ImageJ mediante una interpolación “**Bicúbica**” clásica, seleccionaremos la opción “**Bicúbica**” desde el selector de interpolación a aplicar.

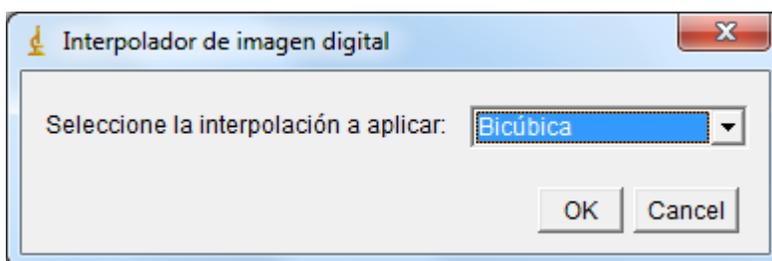


Figura 59. Pantalla que muestra la opción de interpolación bicúbica.

Seguidamente se mostrará la imagen ya escalada mediante la interpolación seleccionada en escala de grises y una ventana de log con información relativa al proceso de interpolación.

### Interpolación propia mediante “Mínimos Cuadrados”.

Si deseamos interpolar la imagen que tenemos abierta desde ImageJ mediante la interpolación propuesta en este proyecto seleccionaremos la opción “**Mínimos Cuadrados**” desde el selector de interpolación a aplicar.

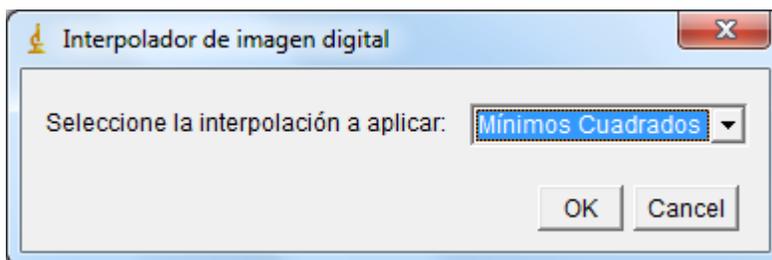


Figura 60. Pantalla que muestra la opción de interpolación por mínimos cuadrados.

Seguidamente se mostrará una pantalla de configuración para este tipo de interpolación.

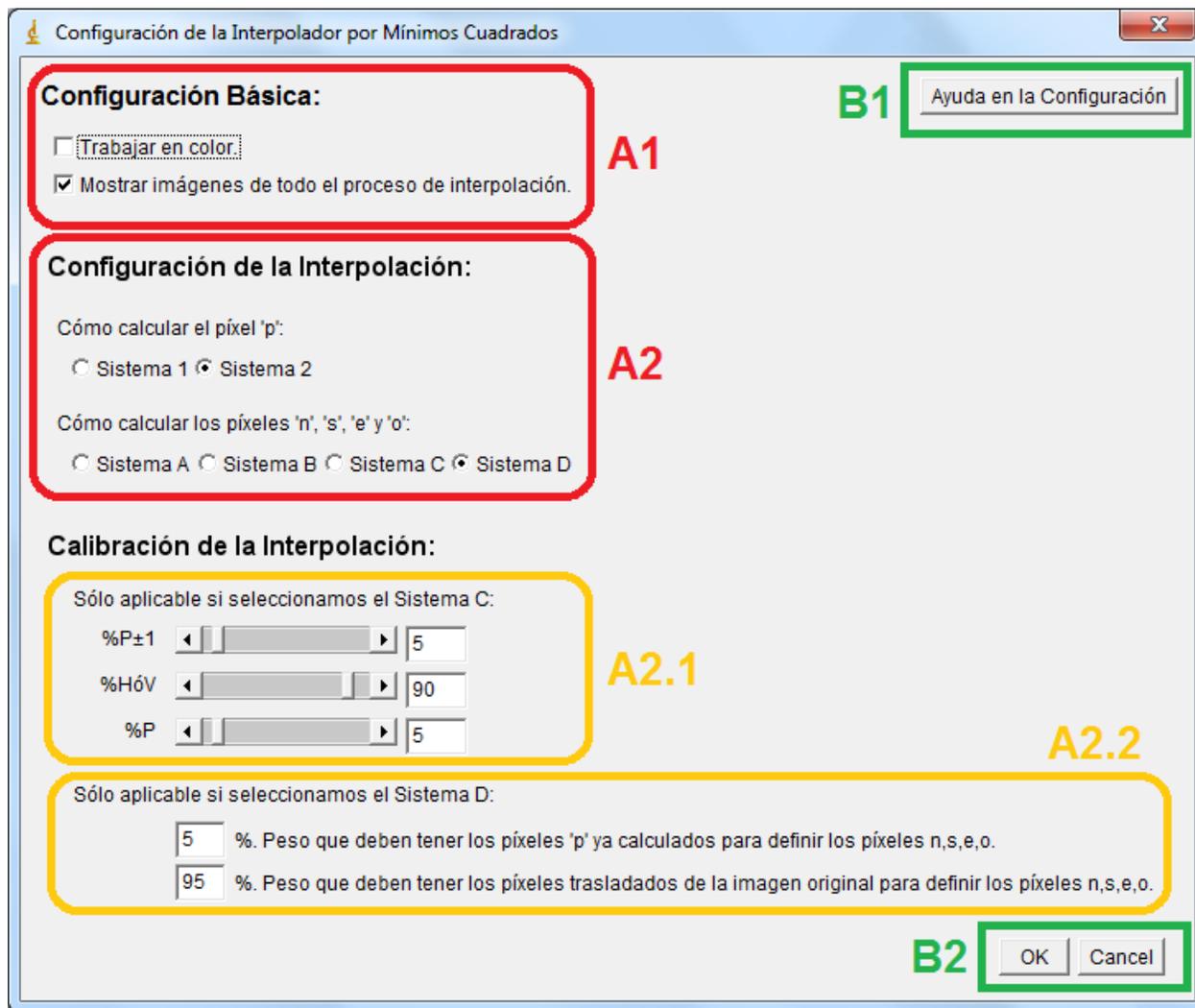


Figura 61. Pantalla de configuración de la interpolación por mínimos cuadrados.

Desde esta pantalla podremos configurar el sistema de interpolación que queramos ejecutar sobre la imagen abierta desde ImageJ.

A efectos prácticos, la sección **A1** corresponde a la configuración básica de la interpolación, como por ejemplo definir trabajar en color o en escala de grises.

Por otra parte la sección **A2** nos permitirá definir qué sistema de cálculo deseamos utilizar para los píxeles de intersección 'p'.

Así mismo desde la sección **A3** podremos seleccionar qué sistema de cálculo deseamos utilizar para los píxeles cardinales 'n', 's', 'e' y 'o'.

La configuración que nos permite esta pantalla queda resumida en la siguiente tabla.

A1	Trabajar en Color.	Permite indicar al método interpolador que utilice una imagen en color y realice el escalado en color, interpolando las tres componentes de color rojo, verde y azul.	
	Mostrar imágenes de todo el proceso de interpolación.	Permite ofrecer por pantalla todas las imágenes temporales que se han generado durante el proceso de la interpolación.	
A2	Cómo calcular el píxel 'p'.	Sistema 1	Permite calcular los píxeles de intersección 'p' según el sistema descrito en el capítulo " <b>Cálculo mediante la línea de tendencia de dos diagonales</b> ".
		Sistema 2	Permite calcular los píxeles de intersección 'p' según el sistema descrito en el capítulo " <b>Cálculo mediante la línea de tendencia de dos diagonales y promedio de vecinos</b> ".
	Cómo calcular los píxeles 'n', 's', 'e' y 'o'.	Sistema A	Permite calcular los píxeles cardinales 'n', 's', 'e' y 'o' según el sistema descrito en el capítulo " <b>Cálculo de los píxeles cardinales mediante el promedio de los dos vecinos más próximos</b> ".
		Sistema B	Permite calcular los píxeles cardinales 'n', 's', 'e' y 'o' según el sistema descrito en el capítulo " <b>Cálculo de los píxeles cardinales mediante las líneas de tendencia en sus coordenadas</b> ".
		Sistema C	Permite calcular los píxeles cardinales 'n', 's', 'e' y 'o' según el sistema descrito en el capítulo " <b>Cálculo de los píxeles cardinales mediante un sistema de pesos</b> ".
		A2.1	Permite establecer los pesos necesarios para formalizar los cálculos del " <b>Sistema C</b> ".
		Sistema D	Permite calcular los píxeles cardinales 'n', 's', 'e' y 'o' según el sistema descrito en el capítulo " <b>Cálculo de los píxeles cardinales en doble iteración, mediante un sistema de pesos entre la vecindad de los píxeles discretos y los píxeles estimados mediante líneas de tendencia</b> ".
A2.2	Permite establecer los pesos necesarios para formalizar los cálculos del " <b>Sistema D</b> ".		
B1	Ayuda en la Configuración	Este botón permite abrir la ventana de ayuda sobre el funcionamiento de las opciones de configuración de la interpolación.	
B2	OK	Este botón permite consolidar las configuraciones realizadas e iniciar el proceso de interpolación que finalizará con la visualización de las imágenes temporales y/o la imagen final escalada.	
	Cancel	Este botón permite salir del plugin sin realizar ninguna acción sobre la imagen.	

Tabla 26. Tabla de configuraciones posibles del interpolador por mínimos cuadrados.

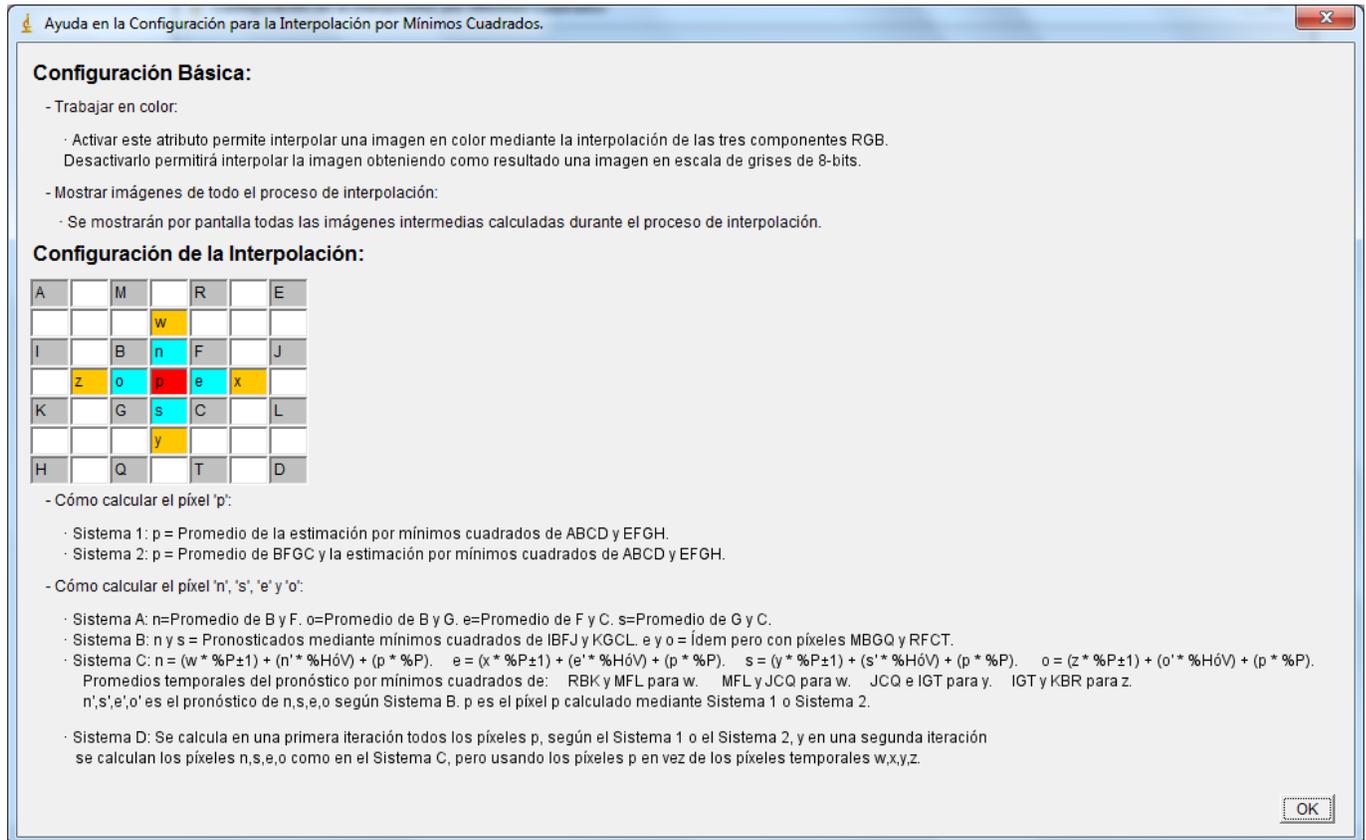


Figura 62. Pantalla de ayuda del plugin interpolador.

Veamos el resultado de una interpolación de ejemplo mediante nuestro método por mínimos cuadrados, sin hacer uso de la interpolación en color, que veremos más adelante.



Figura 63. Imagen original 512x512 píxeles en color.



Figura 64. Detalle de la imagen original.



Figura 65. Imagen original 512x512 convertida a escala de grises.

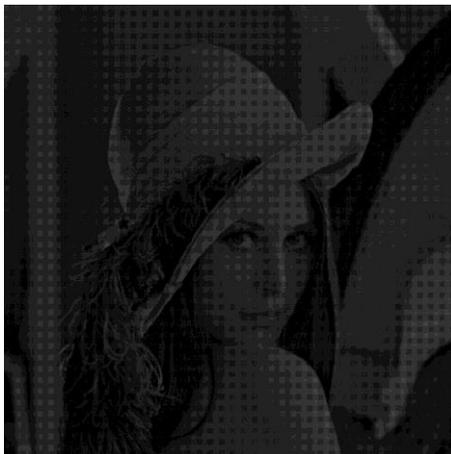


Figura 67. Nuevo mapa escalado al doble de dimensión (1024x1024) con los píxeles de la imagen original espaciados y lista para ser interpolada.



Figura 69. Imagen escalada definitiva, una vez interpolada y con dimensión 1024x1024.



Figura 66. Detalle de la imagen en escala de grises.



Figura 68. Detalle de la imagen espaciada.



Figura 70. Detalle de la imagen interpolada.

## 9. Método comparador de resultados

Uno de los objetivos del TFdM es la exploración de algún método de comparación de resultados para determinar de manera objetiva si el resultado de nuestro interpolador es mejor o peor a los resultados obtenidos mediante la interpolación con los métodos clásicos.

El sistema comparador de resultados debía ser también implementado en Java y como un **plugin para ImageJ**.

La pretensión inicial fue utilizar métodos que permitiesen obtener un valor entero que describiese la nitidez de una imagen, siendo esta calculada por algún método numérico. En relación a este punto hemos implementado un sistema que determina **píxeles de bordes en una imagen** que pretende llevar a cabo este objetivo mediante **máscaras de Sobel**.

No obstante, y para formalizar una investigación más profunda de los métodos estandarizados utilizados para comparar resultados, finalmente hemos estudiado el método de **Relación Señal a Ruido de Pico** o **PSNR**, de sus siglas en inglés **Peak signal-to-noise ratio**.

Finalmente hemos elaborado un plugin para ImageJ que ofrece el resultado de ambos sistemas cuantitativos.

### 9.1. Número de píxeles de borde mediante convolución por máscaras de SOBEL

El cálculo del número de píxeles de borde de una imagen interpolada fue la primera idea tras la prospección de posibles métodos con los que poder comparar, dos o más imágenes escaladas, mediante distintos métodos de interpolación.

Para conseguir este propósito utilizamos la extracción de bordes de la imagen mediante la convolución por máscaras de Sobel. El proceso pasa por los siguientes puntos:

- ✓ Transformación de la imagen a escala de gris.



Figura 71. Original.

- ✓ Sobre la imagen aplicamos la máscara vertical de Sobel para extraer los bordes verticales.

$$\begin{matrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{matrix}$$



Figura 72. Verticalidad.

- ✓ Sobre la imagen en escala de gris aplicamos la máscara horizontal de Sobel para extraer los bordes horizontales.

$$\begin{matrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{matrix}$$



Figura 73. Horizontalidad.

- ✓ Si en alguna de las dos imágenes de borde horizontal o vertical encontramos un píxel con valor 255 (blanco puro) lo consideramos un borde absoluto y lo contabilizamos, de forma que tras analizar las imágenes obtendremos un valor entero con el número total de píxeles de borde.



Figura 74. Ambas dimensiones.

De este modo obtenemos un valor entero que nos describe los bordes encontrados en una imagen, según nuestro criterio, pudiendo comparar una imagen interpolada con otra, simplemente revisando cuántos píxeles de borde se encuentran en cada una de ellas, según este sistema.



Figura 75. Imagen de Lenna convolucionada con máscara vertical y horizontal de Sobel y resultado definitivo de los bordes absolutamente blancos con valor de nivel de gris 255.

## 9.2. Peak signal-to-noise ratio (PSNR)

El **Peak signal-to-noise ratio (PSNR)** es un sistema utilizado como medida cuantitativa de la calidad de la reconstrucción en el ámbito de la compresión de imágenes.

De este modo se determina un valor, en dB, de la calidad de una imagen reconstruida mediante compresión en relación con una imagen original, siendo un mayor valor de PSNR indicador de una mejor calidad de la imagen.

Podemos aprovechar este sistema para realizar el siguiente método comparativo.

- ✓ Partimos de una imagen original.
- ✓ Reducimos esta imagen a la mitad.
- ✓ Escalamos la imagen reducida haciendo uso de diferentes métodos interpoladores, obteniendo de nuevo diferentes imágenes con el tamaño de la imagen original.
- ✓ Comparamos la imagen original con las diversas imágenes escaladas por métodos interpoladores y observaremos qué método ofrece un mayor valor de PSNR siendo su imagen la que ofrezca una mayor calidad respecto al resto.

El cálculo del PSNR se corresponde con la siguiente expresión:

$$PSNR = 20 \log_{10}(MAX_i) - 10 \log_{10}(MSE)$$

Sabiendo que:

- ✓  $MAX_i = (2^B - 1)$ : Es el valor máximo posible de intensidad que puede alcanzar un píxel.
- ✓  $MSE$  (Mean Square Error) o Error Cuadrático Medio.

### $MAX_i$ y el máximo nivel de intensidad

En relación al  $MAX_i$  conviene tener en cuenta que hay autores que lo calculan de un modo u otro:

- ✓ La definición nos describe que se trata del máximo nivel de intensidad que puede alcanzar un píxel en la imagen a tratar. Esto es, si la imagen es de 8 bits el nivel máximo que puede alcanzar un píxel es 255 (color blanco), ya que 8 bits  $\rightarrow (2^8 - 1) = (2^8 - 1) = 256 - 1 = 255$ , siendo este valor el **máximo teórico** que puede alcanzar un píxel.
- ✓ Por otra parte hay autores que definen  $MAX_i$  como el máximo nivel de intensidad que ha alcanzado un píxel en la imagen. Es decir, si la imagen es de 8 bits, sabemos que teóricamente el mayor nivel de gris que puede alcanzar teóricamente es 255, pero puede ser que nuestra imagen no tenga ningún píxel que alcance el valor 255 (una imagen oscura por ejemplo), siendo  $MAX_i$  el valor **máximo práctico** que ha alcanzado un píxel en esa imagen en concreto.

En nuestro caso, calcularemos el PSNR mediante el uso de máximo nivel teórico que puede alcanzar un píxel, siendo esta opción ampliamente recogida por una gran mayoría de autores.

## El Error Cuadrático Medio (MSE)

El **Error Cuadrático Medio** o **MSE** es un estimador estadístico que mide el promedio de los errores al cuadrado. Según un vector de predicciones  $\hat{y}_i$  y un vector de valores verdaderos  $y_i$  la ecuación del MSE se describe de la siguiente manera.

$$MSE = \frac{1}{n} \sum_{n-1}^n (\hat{y}_i - y_i)^2$$

Siendo  $n$  el número de píxeles de la imagen (Altura x Anchura).

Para su cálculo necesitamos dos imágenes, la imagen original donde sabemos que sus píxeles son verdaderos, y la imagen a comparar con la original, que puede ser una imagen interpolada como en nuestro caso. Los píxeles de esta última imagen son los píxeles calculados o predicciones. Veamos cómo formalizar los cálculos para dos imágenes de 1x4 píxeles, con sus respectivos niveles de intensidad de píxel.

1	2	3	4
0	100	70	50

Figura 76. Imagen original.

1	2	3	4
2	90	100	50

Figura 77. Imagen Interpolada.

En este caso  $n = 4$  puesto que la dimensión de las imágenes es de una fila por cuatro columnas de píxeles, con un total de cuatro píxeles en la imagen.

$$MSE = \frac{1}{n} \sum_{n-1}^n (\hat{y}_i - y_i)^2 = \frac{1}{4} [(2 - 0)^2 + (90 - 100)^2 + (100 - 70)^2 + (50 - 50)^2] = \frac{1004}{4} = 251 \text{ dB}$$

## 10. Plugin comparador cuantitativo de imágenes para ImageJ

En este capítulo describiremos el funcionamiento práctico del comparador de imágenes implementado como plugin de ImageJ.

### 10.1. Características

Las características del plugin son las siguientes:

- ✓ Desarrollado en Java como complemento para ImageJ v1.21a o superior, con unas 600 líneas de código.
- ✓ Permite la comparación de una imagen original en escala de gris con diferentes imágenes en las que hayamos realizado cálculos ofreciendo el nivel de MSE, PSNR y píxeles de contorno.
- ✓ El plugin recorta los píxeles del borde de las imágenes a tratar, ya que nuestro interpolador no calcula esos píxeles y deseamos que el plugin pueda comparar dos imágenes en las mismas condiciones de entrada, sin marcos en ninguna de ellas.

### 10.2. Compilación del plugin

Antes de proceder con la ejecución práctica del plugin conviene compilar el código fuente en Java.

Para ello, la herramienta ImageJ ofrece de manera integrada la posibilidad de realizar la compilación del fichero Java. Seguiremos los siguientes pasos:

- ✓ Identificaremos el directorio de instalación de ImageJ. En nuestro caso de ejemplo en la ruta **D:\APPS\ImageJ\**
- ✓ Accederemos al directorio '**plugins**', en nuestro caso ubicado en la ruta **D:\APPS\ImageJ\plugins\**
- ✓ Dentro del directorio '**plugins**' crearemos un directorio para nuestros plugins, por ejemplo en nuestro caso un directorio llamado '**Colome**' que quedará ubicado en la ruta absoluta **D:\APPS\ImageJ\plugins\Colome**
- ✓ Copiaremos dentro del directorio que hemos creado el fichero **COLOME\_ComparadorPSNR.java**
- ✓ Ejecutaremos ImageJ, en nuestro caso el ejecutable **D:\APPS\ImageJ\ImageJ.exe**
- ✓ Desplegaremos el menú '**Plugins**' y seleccionaremos la opción '**Compile and Run...**'

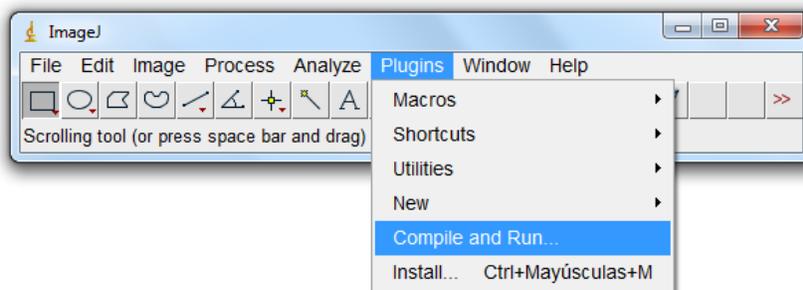


Figura 78. ImageJ y procedimiento de compilación del plugin.

- ✓ Nos aparecerá una ventana desde la que podremos seleccionar ficheros Java. Posicionaremos el foco sobre el directorio donde hemos copiado nuestro plugin y seleccionaremos el fichero **COLOME\_ComparadorPSNR.java**
- ✓ ImageJ compilará el programa y nos mostrará una ventana indicando que no hay ninguna imagen abierta. Pulsaremos el botón 'OK'.



Figura 79. Mensaje indicador que el plugin ha sido compilado.

- ✓ El programa quedará compilado, habiéndose creado en la misma ruta los ficheros de clase "COLOME\_ComparadorPSNR\$Select.class", "COLOME\_ComparadorPSNR\$SNRPane.class", "COLOME\_ComparadorPSNR\$TestImage.class" y "COLOME\_ComparadorPSNR.class".
- ✓ Cerraremos la aplicación ImageJ y la volveremos a ejecutar para que refresque los menús de los plugin y ahora ya podremos encontrar nuestro plugin en el menú 'Plugins', bajo la sección 'Colome' y el nombre de plugin 'COLOME\_ComparadorPSNR'.

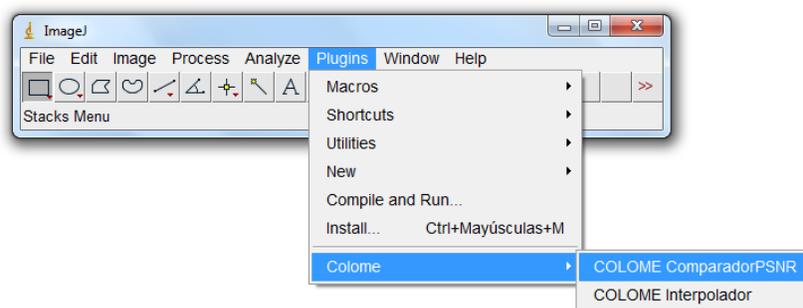


Figura 80. Ubicación del plugin una vez compilado.

### 10.3. Ejecución del plugin

Una vez compilado el plugin podrá ser ejecutado, siempre y cuando tengamos un conjunto de imágenes a comparar en ImageJ, desde el menú 'Plugins', sección 'Colome' y opción 'COLOME\_ComparadorPSNR'. Seguiremos las instrucciones:

- ✓ Conviene tener abiertas en ImageJ la imagen original y las imágenes interpoladas para realizar la comparación entre sí.
- ✓ Una vez ejecutado el plugin observaremos una pantalla principal denominada "Comparador Imágenes Interpoladas mediante PSRN" desde donde debemos clicar en el botón "Seleccionar..." de la sección denominada "Imagen de Referencia Original".
- ✓ Se nos abrirá una pequeña ventana con un listado de todas las imágenes abiertas en ImageJ. Será preciso seleccionar en este punto la imagen original con la que queremos comparar el resto de imágenes abiertas. Una vez seleccionada la imagen de referencia, pulsaremos el botón "Establecer imagen".

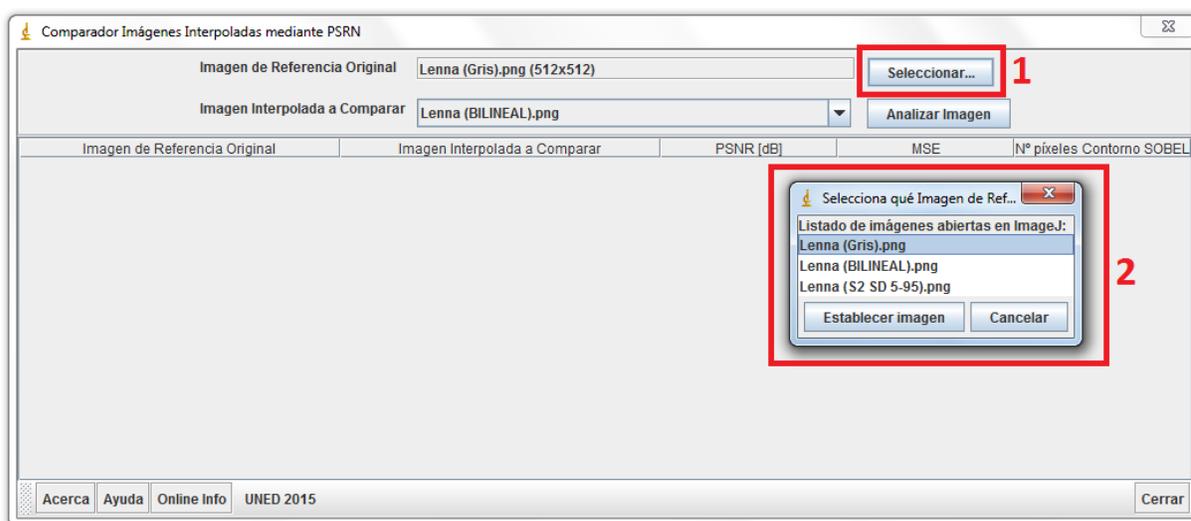


Figura 81. Ventana principal del plugin y selección de la imagen de referencia original.

- ✓ Una vez seleccionada la imagen de referencia, clicaremos en el desplegable de "Imagen Interpolada a Comparar" y podremos observar el nombre de las imágenes abiertas en ImageJ. Seleccionaremos una de las imágenes interpoladas que deseamos comparar con la imagen de referencia original, establecida en el punto anterior. Una vez seleccionada la imagen que queremos comparar con la original, pulsaremos el botón "Analizar imagen".

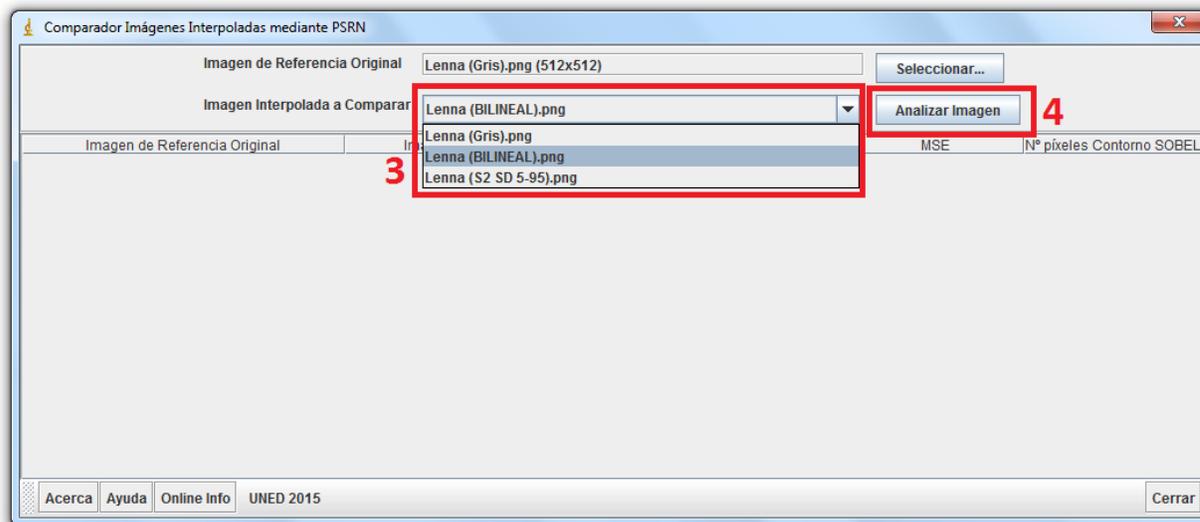


Figura 82. Ventana principal del plugin y selección de la imagen interpolada para ser comparada con la original de referencia.

- ✓ El plugin realizará el cálculo del PSNR, el error cuadrático medio MSE y el número de píxeles detectados como contorno mediante las máscaras de Sobel, mostrando la información en una línea de la ventana principal del plugin.

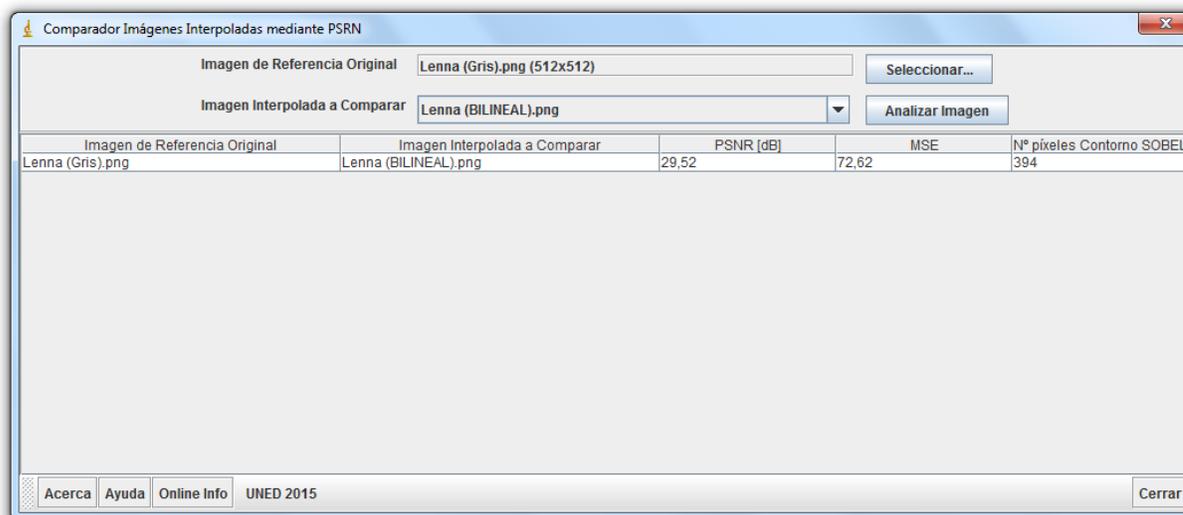


Figura 83. Ventana principal con el resultado del cálculo de la calidad entre la imagen original y la interpolada.

- ✓ Es posible seleccionar del desplegable más imágenes interpoladas abiertas desde ImageJ y compararlas también con la imagen original, acumulando líneas con los resultados de los cálculos.

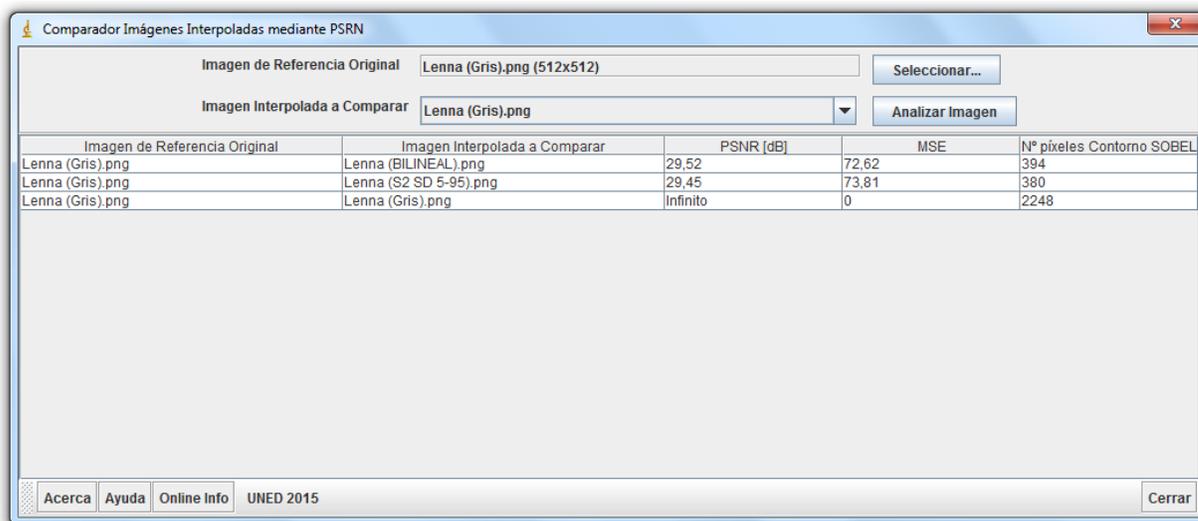


Figura 84. Ventana principal con el resultado del cálculo de la calidad de varias imágenes.

A tener en cuenta que podemos obtener una pequeña ayuda desde el botón “Ayuda” ubicado en la parte inferior izquierda de la ventana principal del plugin.



Figura 85. Ventana de ayuda del plugin de comparación de resultados.

## 11. Juego de pruebas y análisis

Para la realización de pruebas hemos seleccionado cuatro imágenes con la que hemos trabajado y mediante las cuales hemos obtenido las conclusiones finales.

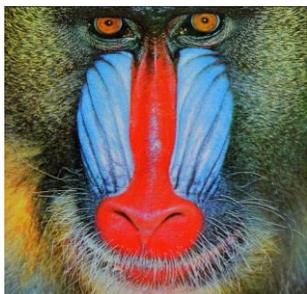


Figura 86. Baboon. 500x480



Figura 87. Bárbara. 720x 576



Figura 88. Cameraman. 256x256

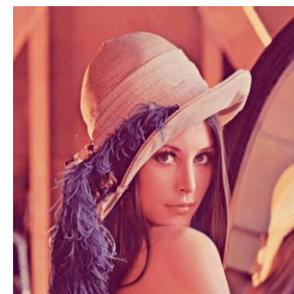


Figura 89. Lenna. 512x512

Hemos ejecutado el plugin que hemos desarrollado con estas imágenes en escala de gris con las siguientes configuraciones de interpolación, con el objetivo de verificar si nuestro algoritmo interpolador por mínimos cuadrados se acerca más o menos al mejor de los sistemas clásicos (Interpolación Bicúbica).

# Prueba	Interpolación	Configuración de la Interpolación
1	Bilineal	No aplica.
2	Bicúbica	No aplica.
3	Mínimos Cuadrados	Sistema 1 + Sistema A
4	Mínimos Cuadrados	Sistema 1 + Sistema B
5	Mínimos Cuadrados	Sistema 1 + Sistema C
6	Mínimos Cuadrados	Sistema 1 + Sistema D (Peso del 5% para píxeles 'p' y 95% para píxeles discretos).
7	Mínimos Cuadrados	Sistema 2 + Sistema A
8	Mínimos Cuadrados	Sistema 2 + Sistema B
9	Mínimos Cuadrados	Sistema 2 + Sistema C
10	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 0% para píxeles 'p' y 100% para píxeles discretos).
11	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 5% para píxeles 'p' y 95% para píxeles discretos).
12	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 15% para píxeles 'p' y 85% para píxeles discretos).
13	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 25% para píxeles 'p' y 75% para píxeles discretos).
14	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 50% para píxeles 'p' y 50% para píxeles discretos).
15	Mínimos Cuadrados	Sistema 2 + Sistema D (Peso del 100% para píxeles 'p' y 0% para píxeles discretos).

Tabla 27. Juego de pruebas a ejecutar en cada una de las cuatro imágenes.

Recordemos que el sistema más elaborado de cálculo del píxel de intersección 'p' es el 'Sistema 2' y que el sistema del que esperamos mejores resultados para el cálculo de los píxeles cardinales 'n', 's', 'e' y 'o' es el 'Sistema D', puesto que realiza dos iteraciones.

En relación al ‘Sistema D’ realizaremos diferentes pruebas haciendo uso de varias configuraciones de pesos para revisar cuál será mejor. Este sistema de pesos asigna más o menos representación al nivel de intensidad de los píxeles ‘p’ o a los píxeles discretos que se trasladaron de la imagen original, todos vecinos al píxel cardinal a calcular.

De ese modo, una configuración de tipo “(Peso del 100% para píxeles ‘p’ y 0% para píxeles discretos)” implica que el cálculo final de los píxeles cardinales se realizará mediante los dos píxeles vecinos de intersección ‘p’ calculados en la iteración anterior, sin tener en cuenta los otros dos píxeles que lo rodean, que provenían directamente de la imagen original.

El método de la prueba consiste en tomar las cuatro imágenes descritas y reducirlas al 50%, interpolando posteriormente esas imágenes para obtener nuevas imágenes al 100% del tamaño original. De este modo posteriormente podremos comparar las imágenes originales con las imágenes escaladas a partir de las imágenes reducidas. Realizando las sesenta interpolaciones descritas mediante nuestro plugin interpolador y pasando las imágenes resultantes por el plugin comparador, obtenemos los siguientes datos.

# Prueba	Imagen	PSNR	MSE	Núm. BORDES	Interpolación	SD - Píxel Intersección 'p'	SD - Píxeles Cardinales 'n','s','e','o'	SD - Configuración peso 'p' vs Originales
1	Baboon	22,38	375,9	5858	Bilineal	No aplica	No aplica	No aplica
2	Baboon	22,26	386,69	7461	Bicúbica	No aplica	No aplica	No aplica
3	Baboon	21,63	446,68	5614	Mínimos Cuadrados	Sistema 1	Sistema A	No aplica
4	Baboon	21,06	509,99	3529	Mínimos Cuadrados	Sistema 1	Sistema B	No aplica
5	Baboon	20,99	518,15	3154	Mínimos Cuadrados	Sistema 1	Sistema C	No aplica
6	Baboon	21,89	420,35	5325	Mínimos Cuadrados	Sistema 1	Sistema D	5% / 95%
7	Baboon	21,92	418,1	5863	Mínimos Cuadrados	Sistema 2	Sistema A	No aplica
8	Baboon	21,31	481,41	3529	Mínimos Cuadrados	Sistema 2	Sistema B	No aplica
9	Baboon	21,25	487,85	3213	Mínimos Cuadrados	Sistema 2	Sistema C	No aplica
10	Baboon	22,19	392,73	5863	Mínimos Cuadrados	Sistema 2	Sistema D	0% / 100%
11	Baboon	22,2	392,04	5621	Mínimos Cuadrados	Sistema 2	Sistema D	5% / 95%
12	Baboon	22,21	391,18	5130	Mínimos Cuadrados	Sistema 2	Sistema D	15% / 85%
13	Baboon	22,2	391,64	4660	Mínimos Cuadrados	Sistema 2	Sistema D	25% / 75%
14	Baboon	22,11	399,7	3545	Mínimos Cuadrados	Sistema 2	Sistema D	50% / 50%
15	Baboon	21,65	444,44	1648	Mínimos Cuadrados	Sistema 2	Sistema D	100% / 0%
1	Bárbara	25,47	184,39	577	Bilineal	No aplica	No aplica	No aplica
2	Bárbara	25,68	175,9	716	Bicúbica	No aplica	No aplica	No aplica
3	Bárbara	24,79	215,6	502	Mínimos Cuadrados	Sistema 1	Sistema A	No aplica
4	Bárbara	24,2	247,09	408	Mínimos Cuadrados	Sistema 1	Sistema B	No aplica
5	Bárbara	24,14	250,37	311	Mínimos Cuadrados	Sistema 1	Sistema C	No aplica
6	Bárbara	24,94	208,25	440	Mínimos Cuadrados	Sistema 1	Sistema D	5% / 95%

7	Bárbara	25,14	199,31	556	Mínimos Cuadrados	Sistema 2	Sistema A	No aplica
8	Bárbara	24,5	230,8	408	Mínimos Cuadrados	Sistema 2	Sistema B	No aplica
9	Bárbara	24,45	233,15	329	Mínimos Cuadrados	Sistema 2	Sistema C	No aplica
10	Bárbara	25,33	190,37	556	Mínimos Cuadrados	Sistema 2	Sistema D	0% / 100%
11	Bárbara	25,31	191,66	521	Mínimos Cuadrados	Sistema 2	Sistema D	5% / 95%
12	Bárbara	25,27	193,11	447	Mínimos Cuadrados	Sistema 2	Sistema D	15% / 85%
13	Bárbara	25,23	194,92	380	Mínimos Cuadrados	Sistema 2	Sistema D	25% / 75%
14	Bárbara	25,1	200,85	272	Mínimos Cuadrados	Sistema 2	Sistema D	50% / 50%
15	Bárbara	24,75	217,67	148	Mínimos Cuadrados	Sistema 2	Sistema D	100% / 0%
1	Cameraman	22,25	387,27	1129	Bilineal	No aplica	No aplica	No aplica
2	Cameraman	22,49	366,13	1397	Bicúbica	No aplica	No aplica	No aplica
3	Cameraman	21,73	436,45	891	Mínimos Cuadrados	Sistema 1	Sistema A	No aplica
4	Cameraman	21,01	514,77	675	Mínimos Cuadrados	Sistema 1	Sistema B	No aplica
5	Cameraman	21,03	512,78	633	Mínimos Cuadrados	Sistema 1	Sistema C	No aplica
6	Cameraman	21,69	440,35	840	Mínimos Cuadrados	Sistema 1	Sistema D	5% / 95%
7	Cameraman	22,28	384,96	1129	Mínimos Cuadrados	Sistema 2	Sistema A	No aplica
8	Cameraman	21,47	463,28	692	Mínimos Cuadrados	Sistema 2	Sistema B	No aplica
9	Cameraman	21,52	458,06	675	Mínimos Cuadrados	Sistema 2	Sistema C	No aplica
10	Cameraman	22,28	384,96	1129	Mínimos Cuadrados	Sistema 2	Sistema D	0% / 100%
11	Cameraman	22,25	387,09	1101	Mínimos Cuadrados	Sistema 2	Sistema D	5% / 95%
12	Cameraman	22,22	389,78	1057	Mínimos Cuadrados	Sistema 2	Sistema D	15% / 85%
13	Cameraman	22,19	393,05	1018	Mínimos Cuadrados	Sistema 2	Sistema D	25% / 75%
14	Cameraman	22,08	402,75	915	Mínimos Cuadrados	Sistema 2	Sistema D	50% / 50%
15	Cameraman	21,81	428,72	696	Mínimos Cuadrados	Sistema 2	Sistema D	100% / 0%
1	Lenna	29,52	72,62	394	Bilineal	No aplica	No aplica	No aplica
2	Lenna	29,85	67,27	571	Bicúbica	No aplica	No aplica	No aplica
3	Lenna	28,73	87,1	357	Mínimos Cuadrados	Sistema 1	Sistema A	No aplica
4	Lenna	27,81	107,77	282	Mínimos Cuadrados	Sistema 1	Sistema B	No aplica
5	Lenna	27,83	107,2	256	Mínimos Cuadrados	Sistema 1	Sistema C	No aplica
6	Lenna	28,67	88,36	341	Mínimos Cuadrados	Sistema 1	Sistema D	5% / 95%
7	Lenna	29,51	72,84	395	Mínimos Cuadrados	Sistema 2	Sistema A	No aplica
8	Lenna	28,42	93,52	282	Mínimos Cuadrados	Sistema 2	Sistema B	No aplica
9	Lenna	28,49	92,03	269	Mínimos Cuadrados	Sistema 2	Sistema C	No aplica
10	Lenna	29,51	72,84	395	Mínimos Cuadrados	Sistema 2	Sistema D	0% / 100%
11	Lenna	29,45	73,81	380	Mínimos Cuadrados	Sistema 2	Sistema D	5% / 95%
12	Lenna	29,42	74,25	361	Mínimos Cuadrados	Sistema 2	Sistema D	15% / 85%
13	Lenna	29,39	74,87	332	Mínimos Cuadrados	Sistema 2	Sistema D	25% / 75%
14	Lenna	29,27	76,94	269	Mínimos Cuadrados	Sistema 2	Sistema D	50% / 50%
15	Lenna	28,96	82,65	116	Mínimos Cuadrados	Sistema 2	Sistema D	100% / 0%

Tabla 28. Resultado de la comparación de las imágenes para los métodos de interpolación y sus configuraciones.

De estos resultados vamos a extraer los cinco mejores métodos de interpolación, según el PSNR, para cada una de las cuatro imágenes.

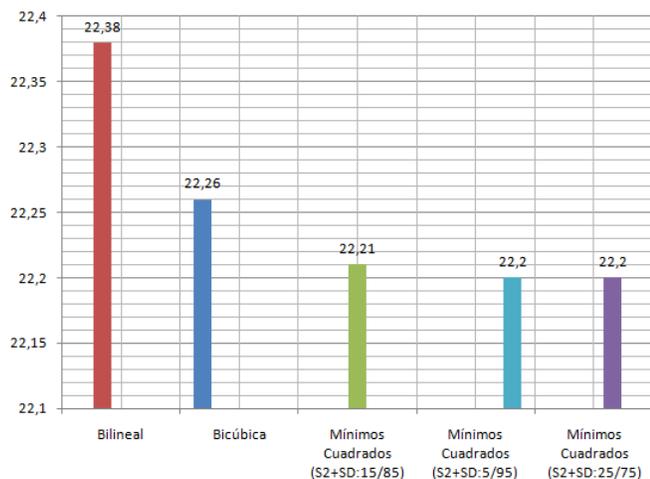


Figura 90. Top 5 PSNR Baboon.

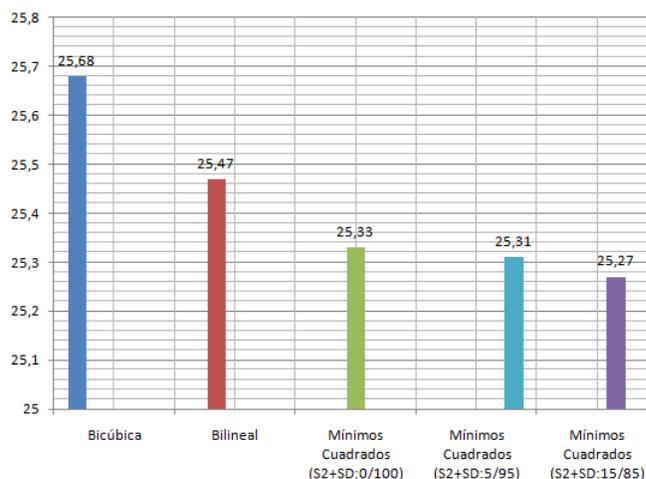


Figura 91. Top 5 PSNR Bárbara

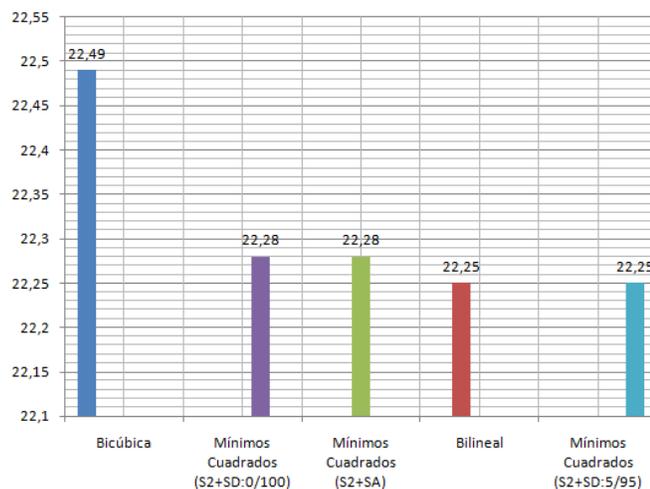


Figura 92. Top 5 PSNR Cameraman.

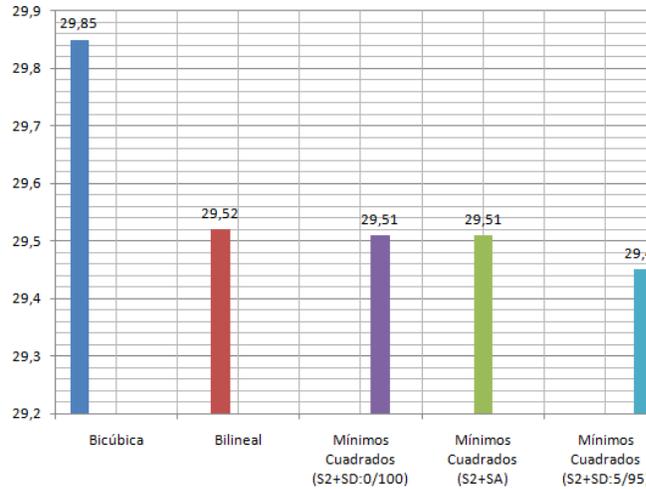


Figura 93. Top 5 PSNR Lenna.

De los gráficos podemos extraer que, según el **cálculo de PSNR**:

- ✓ Todos los métodos de interpolación están en valores de PSNR cercanos, existiendo sólo una desviación de décimas entre los métodos.
- ✓ En tres de las cuatro imágenes el método de interpolación Bicúbica (habitual en aplicaciones de edición de imagen) es el ganador en cuanto a calidad de imagen se refiere. Sólo en una de las imágenes el sistema de interpolación Bilineal ha superado al método Bicúbico.

- ✓ Nuestro método de interpolación propio mediante mínimos cuadrados mantiene el tipo, igualando el valor de PSNR obtenido mediante interpolación Bilineal en la imagen de Lenna y superando a este mismo método en la imagen de Cameraman. En las otras dos imágenes nuestro método se mantiene unas décimas por debajo del método de interpolación Bilineal.
- ✓ Del total de muestras se determina en todos los casos que en nuestro sistema propio, el método '**Sistema 2**' para el cálculo de los píxeles de intersección 'p' es mejor que el '**Sistema 1**' puesto que aparece en todas las marcas de Top-5.
- ✓ Por otra parte, en cuanto al método de cálculo para los píxeles cardinales despunta el '**Sistema D**' con una relación de pesos a favor de los píxeles discretos que rodean al píxel (píxeles que provienen de la imagen original) en detrimento de los píxeles de intersección 'p' calculados en la iteración anterior.
  - Esto tiene cierta lógica, en tanto en cuanto el resultado final es mejor si utilizamos, en el cálculo de los píxeles desconocidos, píxeles que son absolutamente ciertos (provienen de la imagen original) y no han sido calculados o estimados, como es el caso de los píxeles de intersección 'p'.

Por otra parte, si revisamos la extracción del número de píxeles de borde mediante las máscaras de Söbel, aplicando el criterio de a mayor número de píxeles de borde mejor calidad de imagen, observamos que los mejores sistemas interpoladores para cada una de las imágenes son:

Interpolación	Nº Bordes Baboon	Top
Bicúbica	7461	1
Mínimos Cuadrados (S2+SD:0/100)	5863	2
Mínimos Cuadrados (S2+SA)	5863	3
Bilineal	5858	4
Mínimos Cuadrados (S2+SD:5/95)	5621	5

Tabla 29. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen '**Baboon**'.

Interpolación	Nº Bordes Bárbara	Top
Bicúbica	716	1
Bilineal	577	2
Mínimos Cuadrados (S2+SA)	556	3
Mínimos Cuadrados (S2+SD:0/100)	556	4
Mínimos Cuadrados (S2+SD:5/95)	521	5

Tabla 30. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen '**Bárbara**'.

Interpolación	Nº Bordes Cameraman	Top
Bilineal	1129	1
Bicúbica	1397	2
Mínimos Cuadrados (S1+SA)	891	3
Mínimos Cuadrados (S1+SB)	675	4
Mínimos Cuadrados (S1+SC)	633	5

Tabla 31. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen '**Cameraman**'.

Interpolación	Nº Bordes Lenna	Top
Bicúbica	571	1
Mínimos Cuadrados (S2+SA)	395	2
Mínimos Cuadrados (S2+SD:0/100)	395	3
Bilineal	394	4
Mínimos Cuadrados (S2+SD:5/95)	380	5

Tabla 32. Top-5 mejores sistemas interpoladores según el número de píxeles de borde absolutos para la imagen '**Lenna**'.

Incorporando esta información en una tabla de valoraciones en la que comparamos el Top-5 de los métodos de interpolación según el valor de PNSR y el valor de Bordes por máscaras de Söbel, para cada una de las cuatro imágenes, obtenemos:

		Bilineal	Bicúbica	Mínimos Cuadrados (S2+SD:15/85)	Mínimos Cuadrados (S2+SD:5/95)	Mínimos Cuadrados (S2+SD:25/75)	Mínimos Cuadrados (S2+SD:0/100)	Mínimos Cuadrados (S2+5A)	Mínimos Cuadrados (S1+5A)	Mínimos Cuadrados (S1+5B)	Mínimos Cuadrados (S1+5C)
Baboon	PSNR	1	2	3	4	5	No aplica	No aplica	No aplica	No aplica	No aplica
	Bordes	4	1	No aplica	5	No aplica	2	3	No aplica	No aplica	No aplica
Bárbara	PSNR	2	1	5	4	No aplica	3	No aplica	No aplica	No aplica	No aplica
	Bordes	2	1	No aplica	5	No aplica	4	3	No aplica	No aplica	No aplica
Cameraman	PSNR	4	1	No aplica	5	No aplica	2	3	No aplica	No aplica	No aplica
	Bordes	1	2	No aplica	No aplica	No aplica	No aplica	No aplica	3	4	5
Lenna	PSNR	2	1	No aplica	5	No aplica	3	4	No aplica	No aplica	No aplica
	Bordes	4	1	No aplica	5	No aplica	3	2	No aplica	No aplica	No aplica

Tabla 33. Comparación Top-5 del método PSNR con el método de bordes por máscaras de Söbel.

Teniendo en cuenta que el valor '1' es mejor resultado que el valor '5', tanto para el sistema PSNR como para el sistema de máscaras Söbel, observamos lo siguiente.

- ✓ Las coincidencias de las valoraciones del método PSNR con las del método de bordes por máscara de Söbel coinciden en decirnos que el primer y segundo método interpolador de la imagen de Bárbara es el mismo.
- ✓ Por otra parte también coinciden en valorar el primero, tercero y quinto método interpolador en la imagen de Lenna. Es decir, ambos métodos han coincidido en valorar algunos de los mejores métodos de interpolación.
- ✓ Observamos también que existe una desviación considerable en la valoración realizada en la imagen de Cameraman.

Estos últimos datos nos dan una idea en relación al método comparador por máscaras de Söbel, partiendo del buen criterio que el sistema PSNR determina en cada caso, ya que es un método utilizado científicamente en otras disciplinas. Viendo esto podemos determinar que el método por máscaras de Söbel no es un sistema fiable para determinar la calidad de una imagen.

## 12. Conclusiones

Podemos concluir este TFdM con las percepciones que de él se desprenden tras analizar las pruebas realizadas y los valores obtenidos. Las conclusiones son las siguientes.

- ✓ Escalar una imagen incrementando su tamaño implica que habrá una gran cantidad de píxeles nuevos que convendrá calcular siguiendo algún método matemático, siendo los métodos interpoladores los más habituales.
  - Simplemente incrementar dos veces el tamaño de una imagen implica que el 75% de los píxeles de la nueva imagen sean desconocidos.
- ✓ Hemos analizado los diferentes sistemas de interpolación y hemos podido entender que la complejidad de estos métodos no inciden directamente en los resultados.
  - A efectos prácticos nuestro sistema interpolador realiza más cálculos que una interpolación bilineal clásica y normalmente no obtiene mejores resultados.
  - Incluso realizando dos iteraciones sobre la imagen, para calcular un conjunto de píxeles que utilizaremos en la segunda iteración, no implica que el resultado sea más favorable que el mejor de los métodos clásicos con una relación calidad/uso de recursos (método interpolador Bicúbico).
- ✓ Además de considerar el estudio de un método interpolador propio, también hemos ideado un método para comparar cuantitativamente la calidad de las imágenes interpoladas, mediante el uso de máscaras de Söbel para identificar las imágenes con más bordes.
  - Este método, comparado con el de PSNR utilizado por la comunidad científica, ofrece algunos puntos de conexión pero no podemos determinar el mismo resultado haciendo uso de ambos.
  - Es por ello que descartamos identificar una mejor calidad de una imagen haciendo uso del número de los mejores píxeles de borde identificados mediante máscaras de Söbel.
- ✓ Para el cálculo de ciertos píxeles desconocidos en la imagen se tiende a mejorar los resultados cuando se realizan promedios, del nivel de intensidad de los píxeles cercanos, que tendencias entre píxeles más alejados.
  - Esto lo hemos comprobado al certificar que los métodos de cálculo, que utilizan en mayor grado los píxeles más cercanos de la imagen original, al píxel desconocido a calcular, son los que ofrecen un mayor PSNR.
  - Es decir, se obtienen habitualmente mejores resultados configurando el interpolador por mínimos cuadrados con el 'Sistema 1', que calcula píxeles de intersección desconocidos y el 'Sistema D', con una configuración de pesos ventajosa para los píxeles discretos que provienen de la imagen original en detrimento de los píxeles ya calculados.

- ✓ En cuanto a la complejidad técnica que ha conllevado la elaboración práctica de los algoritmos mediante plugins en Java para ImageJ, destacar que tras la consecución de este TFdM he podido obtener una buena formación de habilidades de forma autodidacta, relacionada con el mundo de la programación.
  - En relación al punto anterior hemos profundizado quizá demasiado en el aspecto práctico del TFdM dedicando un gran número de horas a comprender aspectos relacionados con la programación de los plugins en Java.

## 13. Trabajo a Futuro

Tras lograr un conjunto de habilidades relacionadas con la interpolación de imágenes digitales, en sus formas clásicas, tras el estudio y elaboración de un sistema propio y su implementación práctica, pueden identificarse vías de investigación y mejora que conviene tener en cuenta.

En esta línea, los puntos más relevantes que hemos anotado son:

- ✓ Modificar el sistema para que permita un escalado, no solo basado en la ampliación de la imagen sino también permitiendo una reducción de la misma.
- ✓ Mediante pequeños ajustes del algoritmo, permitir el escalado en otras dimensiones, manteniendo o no la relación horizontal con vertical.
- ✓ Aunque se trata de un proceso técnicamente sencillo, no hemos incluido un cálculo de tiempos de procesado del algoritmo, ni ningún estudio comparativo de uso de recursos computacionales con otros métodos de interpolación.
- ✓ En la versión actual no hemos interpolado los bordes de la imagen, obteniendo unos píxeles sin calcular.
- ✓ Siendo foráneo en la programación en Java, el algoritmo puede ser mejorado por expertos en sistemas y estructuras de datos de ese lenguaje.

## 14. Dedicación y horas

Para la consecución de este TFdM hemos dedicado **512 horas** que hemos desglosado en seis actividades.

- ✓ Adquisición de competencias y habilidades (52 horas).
- ✓ Métodos clásicos de interpolación B/N (114 horas).
- ✓ Método propio de interpolación B/N (179 horas).
- ✓ Adaptación de los métodos a imágenes en color (45 horas).
- ✓ Método de comparación de los resultados (90 horas).
- ✓ Finalización del proyecto (32 horas).

El desglose de las tareas para cada una de las actividades queda descrito en el siguiente **diagrama de Gantt**.



## 15. Bibliografía

- ✓ **Visión por Computador, Imágenes digitales y aplicaciones (2ª Edición)**. Gonzalo Pajares Martinsanz, Jesús M. de la Cruz García. *Editorial Ra-Ma*. 2007.
- ✓ **Digital Image Processing. Third Edition**. Rafael C. González, Richard E. Woods. *Pearson Education*. 2011.
- ✓ **Nearest Neighbor Value Interpolation**. Rukundo Olivier, Cao Hanqiang. *IJACSA, Vol 3, Nº4, 2012*.
- ✓ **Performance Analysis Of Different Interpolation Technique Used For Improving PSNR Of Different Images Using Wavelet Transform**. Sudip Kumar, Neelesh Agrawal, Arvind Kumar Jaiswal, Navendu Nitin, Mukesh Kumar. *ECE Department SHIATS-DU Allahabad India. IJERT Vol2, Issue 6, Junio 2013*.
- ✓ **ImageJ User Guide, IJ 1.46r**. Tiago Ferreira, Wayne Rasband. *Octubre 2012*.
- ✓ **Writing ImageJ Plugins (based on IJ 1.36)**. Werner Bailer. *Upper Austria University of Applied Sciences. Dept. of Media Technology and Design- Hagenberg, Austria. Julio 2006*.
- ✓ Primera imagen digital escaneada en 1957, hijo de Russell Kirsch. Kirsch, Russell A., "Russell A. Kirsch", NIST's Museum; SEAC and the Start of Image Processing at the National Bureau of Standards (National Institute of Standards and Technology). This image is in the public domain because it is a work of the United States Federal Government, specifically an employee of the National Institute of Standards and Technology, under the terms of Title 17, Chapter 1, Section 105 of the US Code.

## 16. Lista de Siglas

- ✓ **PSNR**. Peak signal-to-noise ratio. La Relación Señal a Ruido de Pico permite definir la relación entre la máxima energía posible de una señal y el ruido que afecta a su representación fidedigna. Es una técnica utilizada para valorar cuantitativamente la calidad de una imagen digital.
- ✓ **MSE**. Mean Square Error. El Error Cuadrático Medio mide el promedio de los errores al cuadrado, es decir, la diferencia entre el estimador y lo que se estima. Es un parámetro utilizado en el cálculo del PSNR.
- ✓ **TFdM**. Trabajo Final de Máster. Hace referencia al presente documento y a los programas que se han desarrollado para el fin de los objetivos del proyecto.
- ✓ **Mínimos Cuadrados (S1+SA)**. Define el uso de nuestro propio interpolador con la configuración del 'Sistema 1' y el 'Sistema A'.
- ✓ **Mínimos Cuadrados (S2+SD:5/95)**. Define el uso de nuestro propio interpolador con la configuración del 'Sistema 2' y el 'Sistema D', siendo la configuración de pesos de este último un 5% vs 95% en relación a la intensidad de los píxeles de intersección 'p' y a los píxeles discretos de la imagen original respectivamente.
- ✓ **MAXi**. Máximo nivel de intensidad que puede tener un píxel de una imagen.
- ✓ **dB**. Decibelios.
- ✓ **dE**. Distancia Euclídea