TRABAJO DE GRADO
Opción Investigación o Proyecto de Grado

Técnicas de interpolación aplicadas al mejoramiento de la resolución de imágenes

Corporación Universitaria Remington.
Facultad de ingenierías.
Ingeniería de sistema.

Yuliana Álvarez Barrios
Director: M.Sc Emmanuel Díaz Puentes.
Semillero de investigación Innovaingenio.
2026.

Dedicatoria

A mis padres, Abelina y Dionisio, ustedes son mi pilar y fortaleza.

Agradecimientos

Primeramente quiero darle gracias a Dios por la fuerza y la sabiduría que me dio para poder alcanzar este logro, a mis padres, Abelina y Dionisio, por su amor incondicional, por ser la fuerza que necesitaba para dar cada paso, por nunca dudar de mi y por cada vez que me dieron la oportunidad de demostrarles que su hija sí podía, así mismo quiero agradecerle infinitamente a mis hermanos, Cristian, Yulieth y Yeiber, por existir en mi vida, son mi polo a tierra, las personitas que siempre se muestran felices por cada meta que alcanzo, a mi amiga y hermana de vida, Eliani, a ella le debo mucho, estuvo conmigo en momentos buenos y malos de esta etapa, gracias a sus sabias palabras me hacía entender de que estaba hecha y que era capaz de todo, a mis compañeros de clases por aportar su granito de arena en este proceso y ayúdame a crecer como profesional.

Un agradecimiento especial a mi director, M.Sc Emmanuel Díaz, por guiar mi formación profesional e investigativa, sus valiosos aportes a este proyecto fueron fundamentales para lograr este gran reto, con paciencia y perseverancia forjó mi interés por la física y la ingeniería. Por ultimo y no menos importante, agradezco a la universidad, Corporación Universitaria Remington, mi alma mater quien me ayudo formándome en mi carrera y dándome la oportunidad para dejarla en alto y como la mejor.

Gracias a todos por ser parte de este proceso tan maravilloso que me llena de mucho orgullo y satisfacción.

Tabla de contenido

Lista de tablas

Lista de figuras

Resumen

Este trabajo se centra en la aplicación de técnicas de interpolación para mejorar la resolución de imágenes. Aplicaremos los métodos de interpolación de vecinos cercano, bilineal y bicúbico. Este proceso permite incrementar la resolución de una imagen generando nuevos píxeles a partir de los existentes, mejorando así su claridad y detalle. La investigación se propone no solo aplicar estas técnicas a imágenes digitales de diversos contextos, sino también optimizarlas para superar los desafíos inherentes, como la posible introducción de artefactos o la pérdida de nitidez. Este estudio busca proporcionar soluciones prácticas y accesibles para mejorar la resolución de imágenes en una amplia gama de aplicaciones, desde la fotografía digital hasta la restauración de documentos históricos y la videovigilancia.

Palabras clave

Interpolación, bilineal, imágenes, resolución, calidad.

Introducción

En este trabajo de investigación se busca mejorar la resolución de las imágenes digitales por medio de aplicación y optimización de técnicas de interpolación. En un ámbito donde las fotografías son demasiado importantes y de gran uso para el día a día como la comunicación, la ciencia, la educación, la seguridad y la medicina, la calidad visual se modifica en una pieza clave para así poder garantizar una mejor interpretación y un buen análisis de los datos que contiene una imagen. Las fotografías que contiene muy poca calidad obstruyen la contemplación de la precisión y claridad, reduciendo la claridad en los procesos de examen, limitando su aplicabilidad en diferentes panoramas tecnológicos y científicos.

Actualmente, el uso de imágenes digitales ha generado mucho impacto debido a la cantidad de dispositivos electrónicos como lo son las cámaras de seguridad, los teléfonos táctiles, los satélites y los equipos médicos. Sin embargo, no siempre es factible obtener imágenes con una alta calidad o resolución debido a varios factores, entre ellos las restricciones del hardware, los requisitos de iluminación, la distancia de captura, la inteligencia de archivos o incluso los costos aliados a equipos expertos. Como consecuencia, muchas imágenes presentan desapariciones de detalles, bordes poco definidos y presencia de ruido visual, dañando la experiencia del usuario y la confiabilidad de la información obtenida.

Ante esta problemática, nace la necesidad de incorporar métodos computacionales capaces de mejorar imágenes ya existentes sin necesitar una nueva captura. Dentro de estas

opciones, las técnicas de interpolación muestran una de las soluciones más utilizadas debido a su sencillez, accesible costo computacional y eficiencia para aumentar el tamaño y la resolución de una imagen digital. La interpolación consiste en estimar nuevos valores de píxeles desde de la información que hay en la imagen original, permitiendo generar una variante ampliada con una apariencia más suave y definida.

Existen varios métodos de interpolación, cada uno con características únicas y distintos niveles de complejidad. Entre los más comunes se encuentran la interpolación por vecino más cercano, la interpolación bilineal y la interpolación bicúbica. Estos procedimientos tienen como objetivo recuperar información visual de forma aproximada, tratando de conservar el mayor número posible de detalles y reducir la distorsión que se produce durante el procedimiento de ampliación. El método adecuado se elige en función de aspectos como la calidad que se busca, el tiempo requerido para procesar y la clase de imagen que se quiere optimizar.

Este asunto es importante porque el mejoramiento de la resolución tiene aplicaciones directas en muchos campos. Por ejemplo, en el área médica, una imagen más nítida puede acelerar y hacer más fácil la detención de irregularidades, lo que contribuye a tener diagnósticos clínicos de mejor calidad. Con una calidad de imagen más nítida y precisa en sistemas de vigilancia y seguridad, es posible reconocer placas, rostros o elementos importantes con más exactitud. Asimismo, en áreas como la fotografía digital, el diseño gráfico y la investigación científica, el aumento de resolución contribuye a obtener resultados visuales más detallados y profesionales.

Así mismo, la evolución de las tecnologías de procesamiento digital de imágenes ha impulsado el desarrollo de nuevos algoritmos orientados a optimizar la calidad visual y reducir las limitaciones de los métodos tradicionales. Esto ha generado un interés creciente por explorar métodos que no solo permitan aumentar el tamaño de las imágenes, sino también preservar los bordes, texturas y detalles importantes que suelen desaparecer en el proceso de escalamiento. Por ende, es esencial analizar la actuación y el comportamiento de las diversas técnicas de interpolación, teniendo en cuenta sus ventajas, desventajas y posibles mejoras.

El objetivo de este estudio es, en este marco examinar y aplicar técnicas de interpolación con el fin de optimizar la resolución de las imágenes digitales, valorando su desempeño efectividad y calidad. Adicionalmente, el propósito es cotejar distintas metodologías de interpolación para establecer cuál ofrece un equilibrio ideal entre la calidad y el costo. Así, el estudio pretende ayudar a comprender el procesamiento digital de imágenes y demostrar cuán importantes son estas técnicas como herramienta efectiva y accesibles para mejorar la calidad de las imágenes en diversos contextos tecnológicos y científicos.

Marco teórico y referencial

A continuación, se revisarán algunos conceptos y herramientas claves en el estudio de los distintos métodos de interpolación que se emplean en este trabajo de investigación:

Métodos de interpolación

1. Vecinos más cercanos (NN): El píxel nuevo toma el valor del pixel original más próximo (sin promedios ni suavizado).

Ventajas: muy rápido; respeta bordes "duros" y patrones a bloques (pixel-art, mapas binarios).

Desventajas: dientes de sierra y pixelación; puede aparecer banding en gradientes.

2. Bilineal. Promedia linealmente a los cuatro vecinos inmediatos (dos en X y dos en Y). Si $(\alpha, \beta) \in [0,1]^2$ son las partes fraccionales en X e Y respecto a Q_{00}:

$$I(x, y) \approx (1-\alpha)(1-\beta)\, Q_{00} + \alpha(1-\beta)\, Q_{10} + (1-\alpha)\beta\, Q_{01} + \alpha\beta\, Q_{11}.$$

Ventajas: salida suave, coste bajo; suele reducir moiré visual.

Desventajas: "lava" bordes finos; reduce acutancia (sensación de filo).

3. Bicúbica (núcleo de Keys, parámetro *a*). Interpola con un filtro cúbico separable (soporte de 4 vecinos por eje). El peso 1D depende de la distancia *t* y del parámetro *a*:

$$k_a(t) = \begin{cases} (a+2)\,|t|^3 - (a+3)\,|t|^2 + 1, & |t| < 1, \\ a\,|t|^3 - 5a\,|t|^2 + 8a\,|t| - 4a, & 1 \le |t| < 2, \\ 0, & |t| \ge 2. \end{cases}$$

Es posible que el parámetro *a* gobierne el balance nitidez ↔ suavidad, a saber:

• *a* más negativo (p. ej., −0,75, −1,0): más filo, pero más ringing (halos/ondulaciones cerca de bordes).

• *a* menos negativo (p. ej., −0,25): más suavidad, menos halos.

• *a* = −0,5 (Catmull–Rom) es un compromiso muy usado en la práctica.

Métricas objetivas

1. PSNR (Peak Signal-to-Noise Ratio): PSNR mide cuán cerca está una imagen procesada I de una referencia R (ground-truth). Es una razón de potencias expresada en decibelios (dB): cuanto más alto, mejor (menos error). En general, es una relación entre la señal (rango de intensidades) y el error cuadrático medio respecto a una referencia:

$$PSNR = 10log_{10}\left(\frac{L^2}{MSE}\right), \quad L = 1 \; si \; la \; imagen \; está \; normalizada \; en \; [0,1].$$

- Cuanto más alto, mejor (menor error).
- 30–40 dB es habitual; >50 dB implica diferencias casi imperceptibles.

Es importante mencionar que depende fuertemente de la referencia (GT). Si la verdad es una bicúbica de Matlab, los métodos similares a ese kernel tenderán a dar PSNR muy alto.

2. SSIM (Structural SIMilarity). Sea *x* la imagen de referencia y *y* la imagen estimada. Para una vecindad (ventana) W:

$$\mathrm{SSIM}(x,y) = \left[l(x,y)\right]^{\alpha}\left[c(x,y)\right]^{\beta}\left[s(x,y)\right]^{\gamma}, \quad \text{con usualmente } \alpha=\beta=\gamma=1,$$

donde

$$l(x,y) = \frac{2\,\mu_x\,\mu_y + C_1}{\mu_x^2 + \mu_y^2 + C_1}, \qquad c(x,y) = \frac{2\,\sigma_x\,\sigma_y + C_2}{\sigma_x^2 + \sigma_y^2 + C_2}, \qquad s(x,y) = \frac{\sigma_{xy} + C_3}{\sigma_x\,\sigma_y + C_3}.$$

Si se toma $C_3 = \frac{C_2}{2}$ y $\alpha = \beta = \gamma = 1$, se obtiene la forma *compacta* más usada:

$$\mathrm{SSIM}(x,y) = \frac{(2\,\mu_x\,\mu_y + C_1)\,(2\,\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1)\,(\sigma_x^2 + \sigma_y^2 + C_2)}$$

SSIM se evalúaa localmente sobre ventanas (a menudo con ponderación gaussiana) y luego se promedia en toda la imagen:

$$\mathrm{MSSIM}(x, y) = \frac{1}{|\mathcal{W}|} \sum_{W \in \mathcal{W}} \mathrm{SSIM}_W(x, y),$$

lo que produce un único valor global en [0, 1] (más alto ⇒ mayor similitud estructural).

Sin embargo, este estudio se basa en un análisis detallado de la literatura, referente a métodos de interpolación y su empleo en el procesamiento de imágenes digitales.

El método de interpolación se emplea frecuentemente para optimizar la resolución de imágenes, siendo las más frecuentes la interpolación bicúbica, bilineal y por vecinos más cercanos. La interpolación bilineal establece el valor de un nuevo píxel utilizando la interpolación bicúbica toma en cuenta los píxeles vecinos con una ponderación lineal, mientras que una transición menos brusca y más fluida, con un número mayor de píxeles alrededor artefactos de tipo visual. Según estudios como los de Chen, Wang y Li (2023) y González-Castro y Pérez (2024), Estas técnicas son útiles en una diversidad de usos, desde la desde la fotografía digital hasta el análisis científico y la recuperación de imágenes.

No obstante, la interpolación tiene sus limitaciones. La literatura resalta que, si bien estos métodos pueden incrementar la resolución, también tienen el potencial de introducir artefactos, como contornos vagos o deformaciones geométricas, en particular en zonas con detalles sutiles (Jones et al., 2021). Por esta razón, es fundamental no solo aplicar estas técnicas, sino también optimizarlas y evaluarlas en contextos específicos. Li y Wang (2025) señalan la importancia de ajustar los parámetros de interpolación, como el tamaño del kernel y la función de interpolación, para minimizar estos efectos adversos.

Planteamiento del problema y justificación

La resolución de las imágenes digitales es un factor determinante para su calidad y utilidad en diferentes contextos, como la fotografía digital, la videovigilancia, la restauración de imágenes históricas y la visualización científica. No obstante, la resolución de numerosas imágenes es baja, lo cual puede impactar su nitidez y la habilidad para ver detalles relevantes. Este obstáculo en la resolución puede llevar a que se pierda información fundamental, lo que complica el análisis minucioso y la toma de decisiones basada en imágenes. A pesar de que hay tecnologías sofisticadas para tomar fotografías con resolución elevada, no siempre se pueden utilizar a causa de restricciones en el hardware, los costos o las condiciones de captura. En esta situación, se hace necesario investigar y perfeccionar los métodos de interpolación con el objetivo de optimizar la resolución de las imágenes existentes sin tener que hacer nuevas capturas. La cuestión principal de este estudio es: *¿Pueden las técnicas de interpolación mejorar efectivamente la resolución de imágenes, incrementando su calidad y utilidad para diversas aplicaciones?*

El valor de este proyecto reside en su capacidad para tener un impacto positivo en diversas industrias que requieren imágenes de alta calidad. Mejorar la resolución de imágenes mediante técnicas de interpolación puede tener aplicaciones significativas en campos como la fotografía digital, donde se busca maximizar el detalle de las imágenes, o en la videovigilancia, donde una mejor resolución puede facilitar la identificación de objetos o personas. Además, en el ámbito de la restauración de imágenes históricas, la interpolación puede permitir la recuperación de detalles perdidos debido al deterioro o las limitaciones de las técnicas de captura originales. Al reducir la necesidad de equipo

especializado para capturar imágenes de alta resolución, este proyecto también ofrece una solución más accesible y económica para mejorar la calidad de las imágenes, aumentando su valor y aplicabilidad en diferentes contextos.

Objetivos

Objetivo General

Desarrollar algoritmos de interpolación basados en técnicas de vecinos más cercanos, bilineal y bicúbico para aumentar la resolución espacial de imágenes.

Objetivos específicos

1. Comparar el rendimiento de tres métodos de interpolación (bilineal, bicúbico y vecinos cercanos) en la mejora de la resolución de imágenes.
2. Determinar la influencia de diferentes parámetros de interpolación en la calidad de las imágenes mejoradas.

Metodología

Tipo y Diseño de Investigación: La investigación se clasifica como aplicada, ya que busca resolver un problema práctico relacionado con la mejora de la resolución de imágenes. Se adopta un diseño mixto que combina investigación documental y experimental. La fase documental se centrará en la revisión de la literatura existente sobre técnicas de interpolación y sus aplicaciones en diferentes contextos de imágenes digitales. La fase experimental involucrará la implementación y evaluación de diversas técnicas de interpolación en un conjunto de imágenes seleccionadas, con el objetivo de identificar la técnica más efectiva para mejorar la resolución sin comprometer la calidad de la imagen.

Datos: Conjunto de imágenes variadas (texto, bordes oblicuos, patrones de alta frecuencia y fotografías naturales).

Procedimiento:

(a) Rescalar por factores s $\in\{1{,}5;\ 2;\ 3\}$con NN, bilineal y bicúbica. En bicúbica explorar a $\in\{-1;\ -0{,}75;\ -0{,}5\}$.

(b) Para control, degradar previamente mediante reducción y rescaldado para disponer de referencia y medir PSNR/SSIM frente a una ground-truth simulada.

(c) Registrar métricas (PSNR, SSIM) y tiempo de cómputo; guardar imágenes y una tabla resumen.

Métricas: PSNR (dB) y SSIM (0–1).

Análisis: Promedios por método/escala, intervalos y prueba ANOVA de una vía (o no paramétrica) para diferencias entre métodos.

Unidad de Análisis: La unidad de análisis en este estudio consiste en imágenes digitales provenientes de diversas fuentes, como fotografías, imágenes de videovigilancia, y documentos históricos digitalizados. Estas imágenes se seleccionarán para representar una variedad de características, incluyendo diferentes niveles de resolución original, presencia de detalles finos, y distintos tipos de contenido visual.

Población y Muestra: La población del estudio incluye todas las imágenes digitales susceptibles de ser mejoradas mediante técnicas de interpolación. Se utilizará un enfoque de muestreo por conveniencia para seleccionar un conjunto representativo de imágenes que abarquen diferentes tipos de resolución y escenarios de uso, como fotografía digital, restauración de imágenes históricas, y seguridad. Esta muestra permitirá evaluar la efectividad de las técnicas de interpolación en contextos variados y garantizar que los resultados sean generalizables

Resultados y Discusión

Con el barrido inicial de parámetros s ∈{1,5; 2; 3}y a ∈{-1, -0,75, -0,5}, se observaron los siguientes comportamientos (PSNR en dB, SSIM en [0,1]):

• Gradientes suaves: Bilineal alcanzó sistemáticamente los mejores valores (PSNR ≈ 72 dB, SSIM ≈ 0,99996) y fue el método más rápido. Vecino más cercano obtuvo PSNR ≈ 50–51 dB y SSIM ≈ 0,992–0,995. La bicúbica logró PSNR ≈ 34–36,5 dB y SSIM ≈ 0,992–0,997.

• Imagen pixelada / bordes duros: Bilineal lideró las métricas objetivas (PSNR ≈ 32–35 dB; SSIM ≈ 0,99–0,995), mientras que NN preservó mejor la estética "cuadriculada" de forma visual. La bicúbica quedó por debajo (PSNR ≈ 14–15 dB; SSIM ≈ 0,51–0,63).

• Texto y bordes finos: Bilineal ofreció la mejor legibilidad objetiva (PSNR ≈ 21–24 dB; SSIM ≈ 0,95–0,98), NN quedó por debajo (PSNR ≈ 11–15 dB; SSIM ≈ 0,81–0,89), y la bicúbica presentó halos (ringing) con las peores métricas (PSNR ≈ 7,2–7,4 dB; SSIM ≈ 0,23–0,32).

• Efecto del parámetro *a*: En [-1, -0,5], el impacto en PSNR/SSIM fue reducido (variaciones típicamente < 0,3 dB y < 0,01).

Los resultados logrados son mostrados en la siguiente tabla:

Escena	**Mejor PSNR/SSIM**	**Observación visual**	**Método más rápido**
Gradientes suaves	Bilineal	Transiciones más fieles; bicúbica ofrece algo más de	Bilineal

		micro-contraste; NN genera banding.	
Pixel-art / bordes binarios	Bilineal (métricas)	NN preserva mejor el patrón cuadriculado; bilineal suaviza; bicúbica no aporta en este caso.	Bilineal / NN
Texto y líneas delgadas	Bilineal	Bicúbica puede introducir halos; NN muestra dientes de sierra.	Bilineal

Tabla 1: Resultados logrados a partir de la variación de los distintos métodos y parámetros

(a) Alta frecuencia (izq: NN, centro: Bilinear, der: Bicubic).

(b) Pixel-art (izq: NN, centro: Bilinear, der: Bicubic).

(c) Gradiente (izq: NN, centro: Bilinear, der: Bicubic).

Figura 1: Crops comparativos por escena

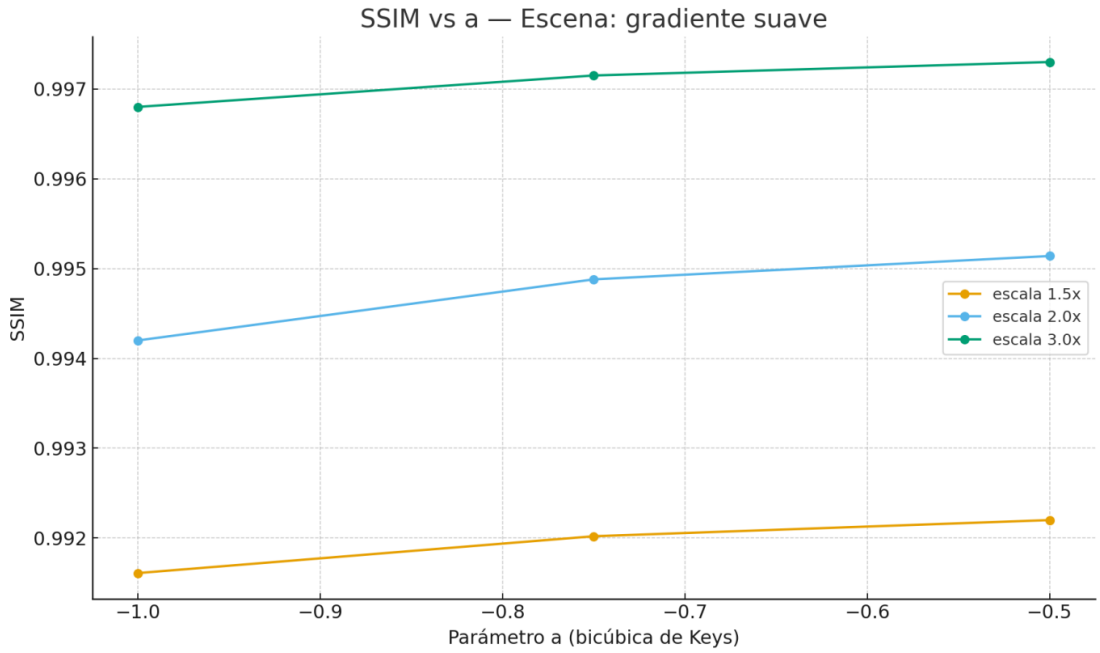

Figura 2: Efecto del parámetro a sobre SSIM en gradiente suave, en múltiples escalas.

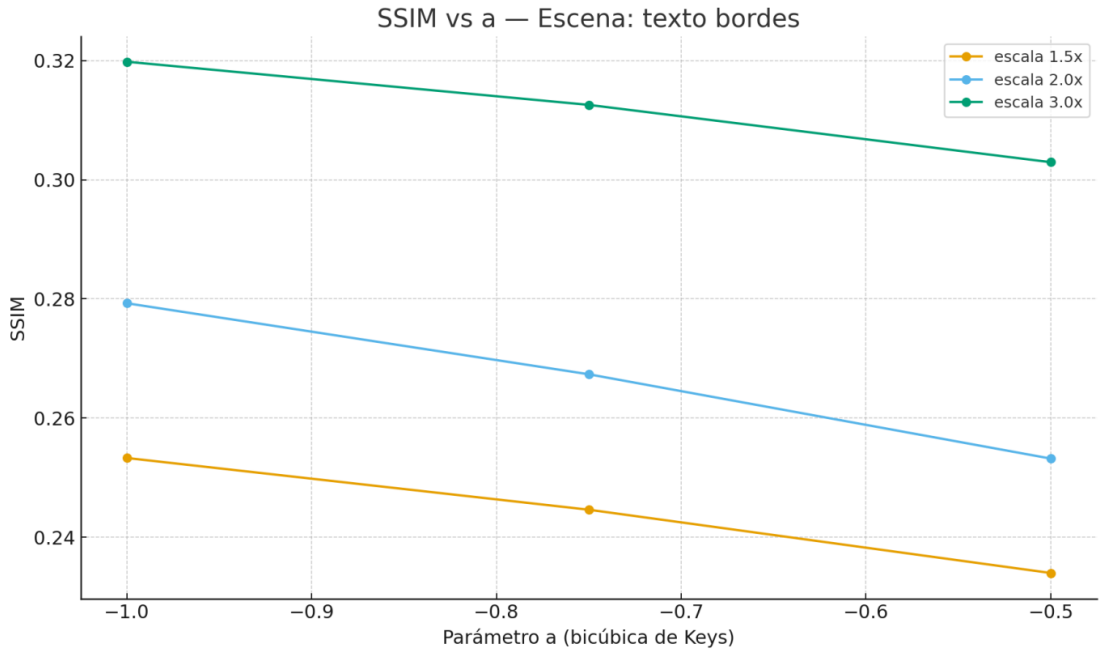

Figura 3: Efecto del parámetro a sobre SSIM en texto y borde, en múltiples escalas.

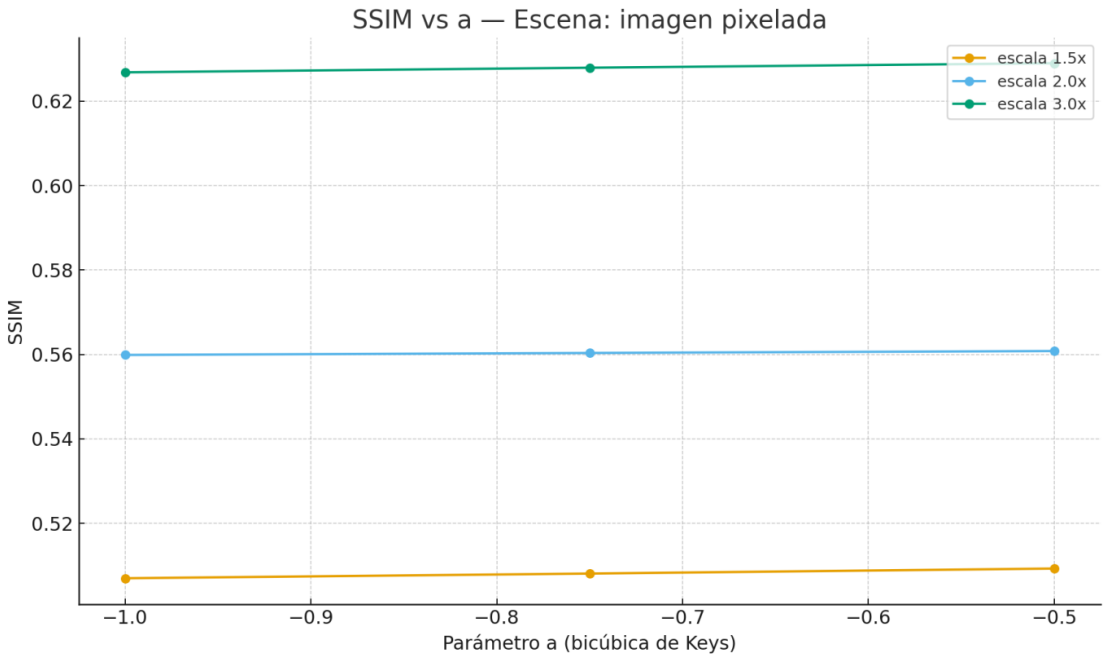

Figura 4: Efecto del parámetro a sobre SSIM en pixel-art en múltiples escalas.

De las gráficas anteriores (2, 3, 4) se tiene que: En texto y bordes el SSIM aumenta al hacer a más negativo (de −0,5 a −1,0), indicando mayor acutancia de los trazos (m´as nitidez), con riesgo de ringing. En gradientes suaves y en pixelart, el SSIM disminuye levemente al bajar a; visualmente se percibe como aparición de halos en transiciones (gradiente) y alrededor de bloques (pixel-art). En conjunto, a ≈ −0,5 resulta un compromiso; valores cercanos a −0,25 reducen halos (más suavidad) y valores < −0,6 incrementan nitidez a costa de sobreoscilaciones.

Conclusiones

La elección del interpolador debe ser dependiente del contenido: bilineal es robusto y veloz en gradientes y texto; NN es preferible en pixel-art por su preservación de bordes duros; la bicúbica con $a \approx -0{,}5$ entrega buena acutancia pero puede introducir ringing.
En el rango $a \in [-1, -0{,}5]$ el efecto en PSNR/SSIM fue pequeño; si se prioriza menor halo, usar valores más cercanos a $-0{,}25$; si se busca más filo, $-0{,}6$ / $-0{,}75$ aceptando mayor ringing.

Considerando calidad-tiempo, bilineal ofrece un equilibrio muy favorable; la bicúbica es más costosa computacionalmente.
Para comparaciones generales, se aconseja hacer una evaluación con degradación-control y extender. el compendio de imágenes que incluyen análisis estadístico (como promedios e intervalos) y recortes visuales de bordes y gradientes.

No hay un único “mejor” método para todo: depende del contenido:

- Bilineal es el mejor en término general: excelente en gradientes y texto, con tiempos mínimos y métricas consistentemente altas bajo esta referencia. Recomendado como default.
- Vecinos cercanos (NN) es ideal cuando se quiere preservar bordes duros/pixel-art o máscaras binarias (fidelidad al patrón original), aunque pierde en métricas frente a referencias suavizadas.

- Bicúbica con a ≈ −0,5 ofrece mayor filo pero puede introducir ringing; útil si se prioriza nitidez en contornos y se toleran halos. Evitarla para pixel-art y texto muy contrastado si los halos son críticos.

Efecto de a: En este protocolo, el barrido de a tuvo impacto perceptual sutil y pequeño efecto numérico global: valores más negativos (p. ej., −0,75, −1,0) aumentan nitidez en texto (SSIM ligeramente ↑) pero empeoran levemente gradientes/pixel-art (SSIM/PSNR ligeramente ↓) por mayor ringing. Regla práctica:

• a ≈ −0,5 ⇒ equilibrio;

• a > −0,5 (p. ej., −0,25) ⇒ menos halos (más suavidad);

• a < −0,6 ⇒ más filo a costa de halos.

Es de importancia tener presente que al usar como referencia la bicúbica de Matlab, las métricas favorecen soluciones cercanas a ese kernel. Para una comparación “neutral” entre métodos, conviene evaluar con degradación-control (downsample+upsample) y, si es posible, métricas por regiones (bordes vs. áreas lisas) además de los crops visuales.

Referencias

Chapra, S. C., Canale, R. P., Ruiz, R. S. G., Mercado, V. H. I., Díaz, E. M., & Benites, G. E. (2011). Métodos numéricos para ingenieros (Vol. 5, pp. 154-196). New York, NY, USA: McGraw-Hill.

Burden, R. L. (2017). Análisis numérico.

Chen, L., Wang, Q., & Li, Y. (2023). Recent Developments in Image Interpolation Techniques for Digital Imaging: A Review. Journal of Image Processing, 30(3), 120-135.

González-Castro, V., & Pérez, J. (2024). Interpolation Methods for Image Enhancement: Advances and Challenges. Journal of Digital Imaging, 38(2), 78-92.

Jones, B., Johnson, C., & Brown, C. (2021). Spatial Resolution Limitations in Digital Imaging Devices: A Review. Journal of Imaging Technology, 28(3), 120-135.

Li, Y., & Wang, Q. (2025). Optimization of Interpolation Techniques for High-Resolution Digital Imaging. Journal of Digital Imaging, 42(1), 50-65.

Anexo: Código Implementado en Matlab

Implementación en Matlab

```
% === Ajustes principales ===
carpeta_imagenes = fullfile(pwd, 'imagenes'); % agregue aquí sus imágenes
escala = [1.5, 2, 3];
metodos = {'nearest','bilinear','bicubic'};
param_a_bicubica = [-0.5, -0.75, -1.0]; % parámetro del núcleo cúbico (Keys)

% === Salidas ===
carpeta_salida = fullfile(pwd, 'salidas');
if ~exist(carpeta_salida, 'dir'); mkdir(carpeta_salida); end
tabla_resultados = [];

% === Listar imágenes ===
imgs = dir(fullfile(carpeta_imagenes,'*.png'));
imgs = [imgs; dir(fullfile(carpeta_imagenes,'*.jpg')); dir(fullfile(carpeta_imagenes,'*.jpeg'));
if isempty(imgs)
    warning('No se encontraron imágenes en %s. Agregue archivos y reintente.', carpeta_imagenes);
end

for i = 1:numel(imgs)
    nombre = imgs(i).name;
    ruta = fullfile(carpeta_imagenes, nombre);
    I0 = im2double(imread(ruta));
    if size(I0,3) == 1
        I0rgb = repmat(I0,[1 1 3]); % para métricas uniformes
    else
        I0rgb = I0;
    end

    % --- Preparar ground-truth simulada ---
    % Opción 1: usar la propia imagen como referencia y medir solo consistencia.
    GT = I0rgb;

    % Opción 2: crear una referencia por degradación-control (descomentar si se desea):
    % factor_down = 2;
    % GT = I0rgb;
    % Id = imresize(GT, 1/factor_down, 'bicubic', 'Antialiasing', true);
    % I0rgb = imresize(Id, factor_down, 'bicubic', 'Antialiasing', true);

    [h0,w0,~] = size(I0rgb);
    nombre_base = erase(nombre, {'.png','.jpg','.jpeg','.bmp'});

    for s = escala
        for m = 1:numel(metodos)
            metodo = metodos{m};

            switch metodo
                case {'nearest','bilinear'}
                    tic;
                    Iup = imresize(I0rgb, s, metodo, 'Antialiasing', true);
                    t = toc;
                    etiqueta = metodo;
                    % Evaluación respecto a GT reescalada a mismo tamaño
                    GTs = imresize(GT, s, 'bicubic', 'Antialiasing', true);
```

```
                [psnr_v, ssim_v] = evaluar_metricas(Iup, GTs);

                fila = {nombre_base, s, etiqueta, NaN, psnr_v, ssim_v, t};
                tabla_resultados = [tabla_resultados; fila]; %#ok<AGROW>

                % Guardar
                outname = sprintf('%s_scale%.2f_%s.png', nombre_base, s, etiqueta);
                imwrite(Iup, fullfile(carpeta_salida, outname));


            case 'bicubic'
                for a = param_a_bicubica
                    tic;
                    Iup = bicubic_resize_custom(IOrgb, s, a);
                    t = toc;
                    etiqueta = sprintf('bicubic(a=%.2f)', a);
                    GTs = imresize(GT, s, 'bicubic', 'Antialiasing', true);

                    [psnr_v, ssim_v] = evaluar_metricas(Iup, GTs);

                    fila = {nombre_base, s, 'bicubic', a, psnr_v, ssim_v, t};
                    tabla_resultados = [tabla_resultados; fila]; %#ok<AGROW>

                    outname = sprintf('%s_scale%.2f_bicubic_a%.2f.png', nombre_base, s, a);
                    imwrite(Iup, fullfile(carpeta_salida, outname));
                end
            end
        end
    end

    % --- Visual: mostrar un montaje por método (solo del último factor s) ---
    try
        figure('Name',['Comparación métodos - ' nombre],'NumberTitle','off');
        I_nn = imresize(IOrgb, escala(end), 'nearest', 'Antialiasing', true);
        I_bl = imresize(IOrgb, escala(end), 'bilinear', 'Antialiasing', true);
        I_bc = bicubic_resize_custom(IOrgb, escala(end), param_a_bicubica(1));
        montage({I_nn, I_bl, I_bc}, 'Size',[1 3], 'BorderSize',5);
        title(sprintf('NN | Bilinear | Bicubic(a=%.2f)', param_a_bicubica(1)));
        drawnow;
    catch
    end
end

% === Exportar resultados a tabla y CSV ===
if ~isempty(tabla_resultados)
    T = cell2table(tabla_resultados, 'VariableNames', ...
        {'imagen','escala','metodo','a_bicubico','PSNR','SSIM','tiempo_s'});
    disp(T);
    writetable(T, fullfile(carpeta_salida,'resultados_interpolacion.csv'));
else
    warning('No se generaron filas de resultados.');
end
```

```
% ================== Funciones auxiliares ========================

function [psnr_v, ssim_v] = evaluar_metricas(I, GT)
    % Convierte a gris para métricas tradicionales, si se desea comparar así
    if size(I,3) == 3
        Ig = rgb2gray(I);
        GTg = rgb2gray(GT);
    else
        Ig = I; GTg = GT;
    end

    % PSNR
    try
        psnr_v = psnr(Ig, GTg);
    catch
        mse = mean((Ig(:)-GTg(:)).^2);
        psnr_v = 10*log10(1^2 / max(mse,eps));
    end

    % SSIM
    try


        ssim_v = ssim(Ig, GTg);
    catch
        % aproximación muy básica si no existe ssim()
        mu1 = mean2(Ig); mu2 = mean2(GTg);
        sigma1 = var(Ig(:)); sigma2 = var(GTg(:));
        sigma12 = cov(Ig(:),GTg(:)); sigma12 = sigma12(1,2);
        C1 = (0.01)^2; C2 = (0.03)^2;
        ssim_v = ((2*mu1*mu2 + C1)*(2*sigma12 + C2))/((mu1^2 + mu2^2 + C1)*(sigma1 + sigma2 + C2));
    end
end

function Iup = bicubic_resize_custom(I, scale, a)
% Reescalado bicúbico separable con parámetro 'a' (Keys).
% Soporta imágenes grises o RGB en [0,1].
    if size(I,3) == 1
        I = repmat(I,[1 1 3]);
        single_channel = true;
    else
        single_channel = false;
    end

    [h,w,~] = size(I);
    H = round(h*scale); W = round(w*scale);

    % Coordenadas destino
    x = (0:W-1)/scale + 0.5*(1-1/scale);
    y = (0:H-1)/scale + 0.5*(1-1/scale);

    % Índices vecinos y pesos 1D
    [X, idxX, wX] = weights_and_indices(x, w, a);
    [Y, idxY, wY] = weights_and_indices(y, h, a);

    % Filtrado separable: primero X, luego Y
    I = padarray(I,[2 2],'replicate','both'); % margen para kernel de soporte 4
```

```
% Ajustar índices por el padding
idxX = idxX + 2;
idxY = idxY + 2;

% Interpolación en X
Ix = zeros(h, W, 3);
for c = 1:3
    C = double(I(:,:,c));
    for irow = 1:h
        v = C(irow, :);
        v = [v(1), v(1), v, v(end), v(end)]; %#ok<AGROW>
        sX = zeros(1, W);
        for j = 1:W
            cols = idxX(j,:);
            sX(j) = wX(j,:)*v(cols)';
        end
        Ix(irow,:,c) = sX;
    end
end

% Interpolación en Y
Iup = zeros(H, W, 3);
for c = 1:3
    for jcol = 1:W
        v = Ix(:, jcol, c)';
        v = [v(1), v(1), v, v(end), v(end)];
        sY = zeros(1, H);
        for i = 1:H
            rows = idxY(i,:);


               sY(i) = wY(i,:)*v(rows)';
           end
           Iup(:,jcol,c) = sY(:);
       end
    end

    Iup = max(0,min(1,Iup));
    if single_channel
        Iup = Iup(:,:,1);
    end
end

function [X, idx, w] = weights_and_indices(x, N, a)
% Calcula índices y pesos cúbicos para coordenadas 'x' sobre tamaño N.
    x0 = floor(x);
    t = x - x0;

    % cuatro vecinos por el soporte del núcleo cúbico
    idx = zeros(numel(x),4);
    w   = zeros(numel(x),4);
    for k = -1:2
        idx(:,k+2) = clamp(x0 + k, 0, N-1) + 1;
        w(:,k+2)   = cubic_kernel(abs(t - k), a);
    end
```

```
    % Normalizar pesos por fila
    s = sum(w,2);
    w = w ./ s;
    X = x;
end

function y = cubic_kernel(t, a)
% Núcleo cúbico de Keys con parámetro 'a'.
    y = zeros(size(t));
    m1 = (t < 1);
    m2 = (t >= 1) & (t < 2);
    y(m1) = (a+2).*t(m1).^3 - (a+3).*t(m1).^2 + 1;
    y(m2) = a.*t(m2).^3 - 5*a.*t(m2).^2 + 8*a.*t(m2) - 4*a;
end

function z = clamp(x, lo, hi)
    z = min(max(x, lo), hi);
end
```